Improving System Maturity Assessments by Incorporating a Design Structure Matrix

by Zuhal Tompkins

B.S. in Microbiology, June 1999, Istanbul University
M.S. in Biotechnology, January 2009, Georgetown University
M.S. in Systems Engineering, January 2016, The George Washington University

A Dissertation submitted to

The Faculty of
The School of Engineering and Applied Science
of The George Washington University
in partial satisfaction of the requirements
for the degree of Doctor of Philosophy

January 10, 2019

Dissertation directed by

Michael Grenn
Professorial of Engineering Management and Systems Engineering

Blake Roberts
Professorial of Engineering Management and Systems Engineering

The School of Engineering and Applied Science of The George Washington University certifies that Zuhal Tompkins has passed the Final Examination for the degree of Doctor of Philosophy as of September 26, 2018. This is the final and approved form of the dissertation.

Improving System Maturity Assessments by Incorporating a Design Structure Matrix

Zuhal Tompkins

Dissertation Research Committee:

Michael Grenn, Professional Lecturer of Engineering Management and Systems Engineering, Dissertation Co-Director

Blake Roberts, Professional Lecturer of Engineering Management and Systems Engineering, Dissertation Co-Director

Shahram Sarkani, Professor of Engineering Management and Systems Engineering, Committee Member

Thomas A. Mazzuchi, Professor of Engineering Management and Systems Engineering & of Decision Sciences, Committee Member

Amir Etemadi, Professional Lecturer of Engineering Management and Systems Engineering, Committee Member

© Copyright 2018 by Zuhal Tompkins
All rights reserved.

Dedication

This dissertation is dedicated to my husband, **Matthew L. Tompkins**, I would probably not be where I am now without him. His love and support have truly been a blessing.

This dissertation is also dedicated to my children, **Caitlyn** and **Clara**. At the time of this writing, they are seven years old and two years old, respectively.

This dissertation is also dedicated to my mother, **Remziye Vildan Alkas**, for her unconditional love, trust, support in every decision I made and for instilling an importance on education in my life.

Finally, this dissertation is also dedicated in memory of my father, **Husamettin Alkas**.

I am lucky to have all of you in my life.

Acknowledgements

As Caroline Ghosn said “Collaboration is like carbonation for fresh ideas. Working together bubbles up ideas you would not have some up with solo which gets you further faster.”

This dissertation is a collection of work that cannot be done solo. I would like to express my sincere thanks to people who have helped in so many ways.

First and most, I would like to express my sincere gratitude to Dr. Mark London for his support, coaching, tutoring and encouragement during this journey.

I would like to acknowledge Dr. Donald York, Mr. Marc Austin, and The International Systems Readiness Assessment Community of Interest (ISRACOI) for their guidance and sharing their knowledge with me to bubble up great ideas.

I also would like to extend my sincere thanks to Dr. Tugrul Daim.

My deepest thanks to all of you.

Abstract

Improving System Maturity Assessments by Incorporating a Design Structure Matrix

Decision makers utilize qualitative and quantitative metrics/measurements to assess system development and to identify technical and programmatic risks. As systems increase in complexity and the number of interfaces increase, it is vital to develop a comprehensive view of the system development status. In order to help decision makers to assess system level development status, Sauser et al. developed System Readiness Level (SRL) as a system development metric. SRL is a function of Technology Readiness Level (TRL) and Integration Readiness Level (IRL) values for the system elements and interrelations of a specific system. Although the SRL was developed by Sauser et al, it is continuing to be assessed and expend by academia and industry.

Although System Readiness Assessment (SRA) methods have been demonstrated to be valuable for understanding the development status of complex systems of developing technologies in the Department of Defense (DoD) and Department of Energy (DoE) applications, current SRA methods present shortcomings when practitioners apply SRA to a larger system in which significant investments are required to assess technologies and integrate them into larger systems. While current SRA approaches may make sense for a single mission or project, it presents drawbacks for use in research and development programs where decision makers need to assess complex integrated systems that include multiple components. Consequently, decision makers need a tool to conduct a SRA for a complex system that includes multiple elements, interfaces and mission.

This dissertation evaluates the design structure matrix (DSM) as a new approach for assessing the system readiness level (SRL) of complex systems. A framework to demonstrate how a current SRL calculation method can be incorporated into the DSM tool is presented to demonstrate the benefits of DSMs for improved confidence in system readiness assessments. A cross-comparison of the modeled system readiness results obtained with a current SRL approach with nesting assumptions and the proposed DSM approach suggests that the DSM approach provides similar results with the SRL calculations without nesting. The application of the DSM approach to a real-world example is also presented to demonstrate how a DSM framework facilitates the decision-making process for system readiness assessment by providing a more evidence-based maturity assessment for complex systems. The DSM approach discussed in this research can be easily adopted by program managers and practitioners for a wide range of complex systems and system architectures with multiple subsystem components.

Table of Contents

Dedication ..... **iv**

Acknowledgements ..... **v**

Abstract ..... **vi**

Table of Contents ..... **viii**

List of Figures ..... **xi**

List of Tables ..... **xiii**

List of Acronyms ..... **xiv**

List of Mathematical Symbols ..... **xvi**

Chapter 1. Introduction 1

1.1. Problem Statement 2

1.2. Research Scope and Limitations 4

1.3. Organization of Dissertation 5

1.4. Significance and Research Contribution 5

Chapter 2. Literature Review 7

2.1. Definition of System 7

2.2. Definition of System Complexity 8

2.3. Definition of Systems of Systems 8

2.4. Systems of Systems Characteristics 9

2.5. DoD Systems of Systems Definition 10

2.6. DoD Acquisition Lifecycle 12

2.7. System Development Metrics 16

2.7.1 Technology Readiness Level (TRL) 17
2.7.2 Integration Readiness Level (IRL) 19
2.7.3 System Readiness Level (SRL) 21
2.7.3.1 SRL Calculation Methods 23
2.7.3.1.1 Sauser SRL Calculation Method 24
2.7.4 Limitation of the Current SRA Approaches 26
2.8. Design Structure Matrix (DSM) 29
2.8.1 DSM Structure 30
2.8.2 DSM Structure 32
2.8.3 Past Applications of DSMs 34
2.8.4 Building and Analyzing a Product Architecture DSM 35
Chapter 3. Research Goals and Hypotheses 38
3.1. Research Motivations and Goals 38
3.2. Research Questions and Hypotheses 38
Chapter 4. Research Method 40
4.1. Monte-Carlo Simulation Data Collection 46
Chapter 5. Data Analysis and Interpretation 50
Chapter 6. Demonstration Study 63
6.1. Description and Assumptions 63
6.2. Process 64
Chapter 7. Conclusions and Future Work 73

Appendix A: Summary Report - The SRL Values for the Sauser SRL Method and DSM Approach for the System Configurations 75

Appendix B: Boxplot of the SRL Values for the Sauser SRL Method and DSM Approach for the System Configurations 78

Appendix C: Normality Results of the SRL Values for the Sauser SRL Method and DSM Approach for the System Configurations With R 80

Appendix D: Minitab Data Analysis Output 82

References 89

List of Figures

Figure 1.1: Process Diagram 3

Figure 2.1: Simple and Complex Systems (Maier, 1998) 8

Figure 2.2: DoD Systems Engineering Considerations (Schwartz, 2016) 12

Figure 2.3: DSM overview: (A) network diagram; (B) base DSM representation; and (C) partitioned DSM representation (A. Yassine & Braha, 2003). 31

Figure 2.4: The four primary types of DSM models (Eppinger & Browning, 2012) 33

Figure 2.5: Example of Multi-level Clustering and Collapsing of a Clustered DSM (Adapted from (Levandowski et al., 2016)) 34

Figure 4.1: Research Methodology 40

Figure 4.2: Network diagram representation of the system from Knaggs et al with the TRL and IRL values included 41

Figure 4.3: DSM of the System from Knaggs et al. 42

Figure 4.4: Comparison results from composite SRL calculations 43

Figure 5.1: Histogram of Mean (Full System SRL) results 52

Figure 5.3: Histogram of Mean (Nested Full System SRL) results 53

Figure 5.4: Histogram of DMS results 53

Figure 5.5: Boxplot of the SRL values for the Sauser SRL method and DSM approach for the system configurations described by Knaggs et al. 54

Figure 6.1: Example of a warfare system interface diagram adopted from (Beiderbecke, 2013) 65

Figure 6.2: Navy ship combat system DSM with interfaces between the elements 66

Figure 6.3: NSWS 1 (left) and NSWS 2 (right) NSWSs with TRL and IRL values before partitioning. 67

Figure 6.4: NSWS 1 (Top) and NSWS 2 (Bottom) NSWSs with TRL and IRL values after partitioning and clustering. 69

Figure 6.5: SRL mapping to defense acquisition (Brian Sauser et al., 2008). 71

List of Tables

Table 2.1: Systems of System Definitions 9

Table 2.2: Comparing Systems and Systems of Systems Source (DAU, 2017) 11

Table 2.3: List of TRL, IRL, SRL Values, Acquisition Phases, and Source 16

Table 2.4: Technology Readiness Level Definitions (DoD, 2009) 18

Table 2.5: Integration Readiness Level Definitions (Brian Sauser et al., 2008) 20

Table 2.6: Systems Readiness Level Definitions (B. J. Sauser et al., 2006) 22

Table 2.7: Systems Readiness Level Definitions from SRA Handbook (Marc F. Austin & York, 2017) 23

Table 2.8: Legacy SRL Model Calculation Methodology and Standard Equation Forms (adapted from (London et al., 2014)) 24

Table 2.9: Basic procedure for building and analyzing a product architecture DSM (Browning, 2001; Eppinger & Browning, 2012; A. Yassine & Braha, 2003; A. A. Yassine, 2004) 36

Table 4.1: Step-by-step comparison of the methodology of Knaggs et al. and the DSM_SRL methodology 43

Table 4.2: Individual SRL Values 45

Table 4.3: Variable definitions and data types 49

Table 5.1:Descriptive statistics summary of the modeled system SRL results for the Knaggs et al. and DSM_SRL methods 50

Table 5.2: Shapiro-Wilk Test of Normality Summary 55

List of Acronyms

Acronym	Description
ACCC	Advanced Combustion and Carbon Capture (ACCC) system
AT&L	Acquisition Technology and Logistics
CTE	Critical Technology Element
DoD	Department of Defense
DSM	Design Structure Matrix
DSM_SRL	Design Structure Matrix System Readiness Level Methodology
EMD	Engineering, Manufacturing, and Development
GAO	Government Accountability Office
GT	Graph Theory
GT_SRL	Graph Theory System Readiness Level
IM_SRL	Incidence Matrix System Readiness Level
INCOSE	International Council on Systems Engineering
IOT&E	Initial Operational Test and Evaluation
IRL	Integration Readiness Levels
MDA	Milestone Decision Authority
NASA	National Aeronautics and Space Administration
NSWS	Navy Ship Warfare System
O&S	Operation & Support
P&D	Production and Development
PN_SRL	Petri Net System Readiness Level
PMI	Project Management Institute

SE	Systems Engineering
SoS	System of Systems
SRA	System Readiness Assessment
SRL	System Readiness Levels
S_SRL	SysDML System Readiness Level
SysDML	Systems Development and Maturity Laboratory
TA	Tropical Algebra
TA_SRL	Tropical Algebra System Readiness Level
TMRR	Technology Maturity and Risk Reduction
TPM	Technology Performance Measure
TRA	Technology Readiness Assessment
TRL	Technical Readiness Levels
TTRL	Technology Transfer Readiness Level
U.S.	United States

List of Mathematical Symbols

Symbol	Description
TRL*i*	TRL of element *i*,
IRL*ij*	Integration among elements *i* and *j*
SRL*i*	Component SRL (Maturity level of element *i)*
SRL	Composite SRL (arithmetic mean of the calculated component SRLs)
n	the number of elements
n_i	the number of integrations of element i

Chapter 1. Introduction

As systems become more complex and resources become more constrained, the availability of efficient and effective methods to measure System of Systems (SoS) maturity and to provide evidence for systems engineering and programmatic decisions in both the defense and commercial sectors is becoming increasingly important (Atwater & Uzdzinski, 2014; Thal, Olson, & Blessner, 2018). During large-scale developments, it is critical to assess the system's maturity at multiple points during the lifecycle of the project. Given the trends within System Readiness Assessment (SRA) and the increased focus on affordability initiatives for acquisition and sustainment, a SoS approach that can measure the maturity of a complex system throughout its lifecycle is needed (Marc F Austin & York, 2016; Marc F. Austin & York, 2017; Pavel Fomin, 2010; P. Fomin, Mazzuchi, & Sarkani, 2009). Complex systems within a Department of Defense (DoD) construct have been defined previously by (AT&L, 2008; Baldwin & Lucero, 2016; Sheard & Mostashari, 2009). Complex systems represent a challenge for maturity assessments because of the component integrations among the many elements that comprise the overall system. The relational impacts of varying development rates, obsolescence and even the cascading effects of adding enhancements at various points in a system cycle must be assessed to overcome the challenges associated with complex system maturity assessments. However, the dependence or sensitivity of the changing maturity for a component within a complex system cannot be easily recognized to mitigate risks for readiness, performance, cost, or sustainment. This Dissertation demonstrates how System Readiness Level (SRL) calculation methods can be evaluated within a Design Structure Matrix (DSM) framework to increase the level of confidence in

SRAs. Our research employed a modeling and simulation approach to compare the DSM results obtained using the most widely accepted SRA method. An experiment performed on a real-world example demonstrates that the DSM framework facilitates the decision-making process for SRA for the systems engineering and program management community by providing a richer, evidence-based maturity assessment for complex systems.

1.1. Problem Statement

Contemporary government acquisition programs become increasingly complex to meet the challenges of the DoD and commercial sector objectives (Malone & Smoker, 2013). Given recent trends in the development of defense systems (moving toward robust defined requirements and more cost-effective solutions) the DoD has been required to devote more attention to affordability and agility earlier in the development cycle of systems (Atwater & Uzdzinski, 2014). The Project Management Institute (PMI) estimates that for every dollar spent on projects and programs, 5.1 percent is wasted because of poor requirement management (GAO). Under these existing pressures, a holistic understanding and evaluation of the performance and readiness for Critical Technology Elements (CTEs) within complex systems is needed as they develop and mature during the acquisition process. SRA methods offer a foundation for evaluating the existing maturity of systems under development; however, SRAs have not been extended to a multitiered technology development process for complex systems. Previous scholars have acknowledged this gap in the literature (Knaggs et al., 2017). There is a corresponding gap within both industry and academia for tools to support models that assess technology maturity and system performance that may foresee the impact of the development of a

technology. Any tool and model to be considered for SoS maturity must be easy to use, be easy to understand and have results that are produced from objective evidence (Atwater & Uzdzinski, 2014). A detailed literature review revealed that a DSM tool that provides a simple, compact, and visual representation of a complex system may provide opportunities to improve current SRA approaches (Atwater & Uzdzinski, 2014; Avnet, 2016; Avnet & Weigel, 2010; Browning, 2001, 2002, 2016; Eppinger & Browning, 2012; P. Fomin et al., 2009; Garg, Eppinger, Joglekar, & Olechowski, 2017). Figure 1.1 depicts a simplified process diagram that was applied in this dissertation.

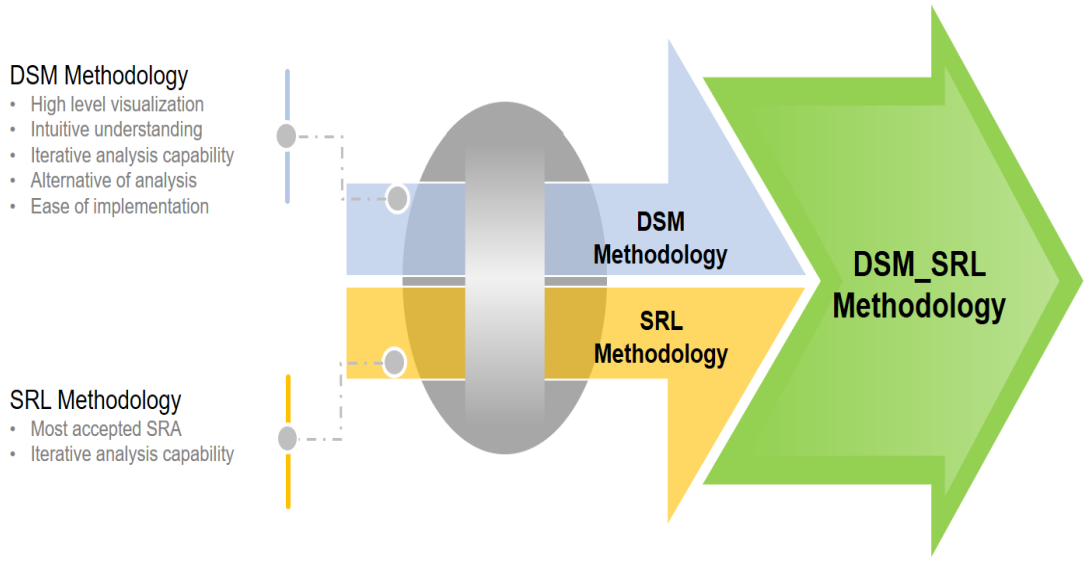


Figure 1.1: Process Diagram

The readily identifiable definition of subsystem boundaries without the use of a confusing and complex network system diagram allows for the effective tracking and monitoring of subsystem readiness metrics over time as the system progresses through the acquisition process. The DSM approach applied in this dissertation provides an effective and intuitive tool for the system engineering and program management

communities for the active tracking of system readiness and maturity through the development and acquisition lifecycle.

1.2. Research Scope and Limitations

According to the traditional measurement, there are two types of measurement properties, mathematical and utility properties (Bedford & Cooke, 2001). This dissertation discusses the utility properties to demonstrate how SRL models can be improved by incorporating DSM tool. This research provides a detailed quantitative analysis of the legacy SRL, but does not examine the mathematical operations and was developed under the following baseline limitations;

- This dissertation accepted the assertation that SRA, and therefore SRLs are being used and are assumed to be helpful to decision makers.
- This dissertation did not suggest enhancements to currently available SRL approaches, and accepted the approaches proposed by Sauser et al., Garrett et al., McConkie et al., London et al., and Thal et al. as they were known in the literature.
- This dissertation does not determine if there is a correlation between SRL values and system cost.

Decision makers utilize qualitative and quantitative metrics/measurements to assess particular system development and to identify technical and programmatic risks. This dissertation aimed to improve SRA process by combining SRLs approaches (quantitative) and Design Structure Matrix (DSM) tool (qualitative).

1.3. Organization of Dissertation

The remainder of this dissertation is structured as follows: First, a brief review of system development metrics, including Technology Readiness Level (TRL), Integration Readiness Level (IRL), and SRL based on the calculation methods in the existing literature is provided. Next, a review of the DSM model approach is presented, followed by the application and integration of a SRA method employing the proposed DSM approach (DSM_SRL). The dissertation also discusses how the DSM approach can be used for SoS maturity assessment and defines the relationships among objects regarding technology and integration that can be used for the model. A system example from (Knaggs et al., 2017) is evaluated using this unique DSM approach that leverages modeling and simulation to highlight the pertinent advantages of this approach. This dissertation also demonstrates the unique application and system maturity estimation benefits of the DSM approach for two real-life examples: 1) an Navy Ship Warfare System (NSWS) that was completed on schedule and on budget with no new technology insertions, "NSWS 1", and 2) an evaluated NSWS with multiple immature critical technology insertions that experienced schedule delays and budget growth, "NSWS 2" (GAO, 2013, 2017). Finally, specific data results and conclusions are presented along with suggestions for future extensions of the current study to better support decision makers for improved confidence in SRAs.

1.4. Significance and Research Contribution

Risk management is one of the most critical measures in the CTE development and integration. Inadequate risk identification can result in unexpected schedule overruns, budget increase, substantial rework, and reduced capability. Although some decision

making tools exist to support decision makers to recognize and mitigate risks, only a few clearly consider the effect of a system's architecture.

This dissertation makes three major contributions to the SRA body of knowledge. The first contribution is methodological. A SRL assessment of complex systems through a DSM model is introduced to be used by decision makers on projects involving development and integration of new technology elements into a complex system or a SoS. The new framework combines analysis with SRL and system architecture concepts to focus attention on high risk elements and interfaces. This work addresses technical risk, which deals with the uncertainty associated with evolving and integrating CTEs into a complex system or a SoS. The second contribution is the validation of the proposed DSM_SRL method through the application of the same system and data set employed by Knaggs et al. (Knaggs et al., 2017). The third contribution to the SRA body of knowledge is the application ofthe proposed model to a real-world DoD example using a Navy Ship Warfare System (NSWS) for two US Navy surface ships. The analytical framework developed in this dissertation was applied to two real-world DoD examples using a NSWS for two US Navy surface ships. NSWS 1 anfd NSWS 2 projects were included in the data set. Observations were compared to post-project lessons learned to develop a general understanding of the relationship between the project structure and the implementation approach for each case.

Chapter 2. Literature Review

This chapter provides an overview of the literature review used as foundation for the research. The section begins with assigning definitions to system, system complexity, and SoS to lay the foundation of the terms will be used throughout this dissertation. Next characteristics of a SoS provided to emphasize that not all complex systems are SoSs. Following the DoD SoS definition and system vs SoS comparison, DoD acquisition life cycle processes are reviewed to align the lifecycle of the project with maturity assessment process. Detailed discussion of maturity metrics used to evaluate the system readiness throughout SRA development are provided along with TRL, IRL, and SRL formulations. Finally, detailed discussion of DSM, DSM structure, and past applications of DSMs are provided.

2.1. Definition of System

There are many definitions of a system, but most commonly accepted definitions of the system proposed by International Council on Systems Engineering (INCOSE) and Dr. Russell Ackoff. While INCOSE defines a system as "A system is a construct or collection of different elements that together produce results not obtainable by the elements alone." (INCOSE, 2015), Dr. Ackoff defines it as "a system is a set of two or more elements that satisfies the following properties: The behavior of each element has an effect on the behavior of the whole, the behavior of each of the elements and their effects on the whole are interdependent, and however subgroups of the elements are formed, each has an effect on the behavior of the whole and none has an independent effect on it." (Ackoff, 2000).

2.2. Definition of System Complexity

The complexity of any system is subject to the number and the types of relationships amongst the elements. Figure 2.1 displays examples of two systems. While system A is a fairly simple system with straight forward relationships amongst its elements, System B, presents more complexity. Even though both systems contain the same number of elements, System B presents more element interactions.

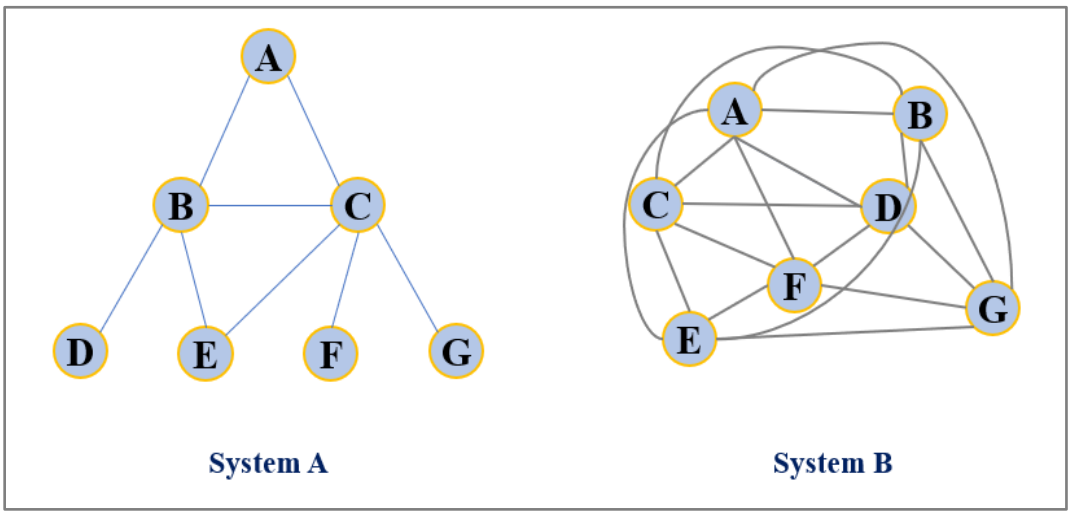


Figure 2.1: Simple and Complex Systems (Maier, 1998)

It is important to mention in this section that not every complex system is a SoS (Maier, 1998). For example, human body is a complex system, but it is not a SoS due to composed of subsystems which are not independent and cannot function by themselves. Following section will provide more details on definition and characteristics of a SoS.

2.3. Definition of Systems of Systems

System of systems describes the interaction between different independent and complex systems in order to achieve a common goal. There are many definitions of SoS depending on the application area and focus (Carlock & Fenton, 2001; Maier, 1998; Sage & Cuppan, 2001). Table 2.1 summarizes most common SoS descriptions.

Table 2.1: Systems of System Definitions

Source	Definition
SE Body of Knowledge	"A SoS is an integration of a finite number of constituent systems which are independent and operatable, and which are networked together for a period of time to achieve a certain higher goal."(Jamshidi, 2008)
INCOSE SE Handbook	"A system-of-interest whose elements are managerially and/or operationally independent systems. These interoperating and/or integrated collections of systems produce results unachievable by the individual systems alone." (INCOSE, 2015)
ISO/IEC/ IEEE 15288: 2015	"A system of systems (SoS) is a system-of-interest (SOI) whose elements are themselves systems. A SoS brings together a set of systems for a task that none of the systems can accomplish on its own. Each constituent system keeps its own management, goals, and resources while coordinating within the SoS and adapting to meet SoS goals." (IEEE, 2015)
US DoD SoS SE Guide	"A set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities." (DAU, 2017)

2.4. Systems of Systems Characteristics

Maier's work on SoS discussions is one of the earlier systematic discussions on SoS. In his study, Maier identified five important characteristics of a SoS (Maier, 1998).

1. *Operational Independence of the Elements*: The subsystems of the SoS must be able to operate independently. The SoS is composed of subsystems which are independent and have an individual role.
2. *Managerial Independence of the Elements*: The subsystems do operate independently. The subsystems are separately developed and integrated but continue a current operational existence independent of the SoS.
3. *Evolutionary Development*: The SoS formation is an evolving process. Its development and presence is evolutionary with capabilities and functions and added, removed, changed, and adapted with knowledge.

4. *Emergent Behavior*: SoS have developing features that only become apparent after the SoS has been created. These features are developing properties of the entire SoS and cannot be present within the subsystems before becoming a SoS.
5. *Geographic Distribution*: The subsystem of a SoS are often geographically scattered across diverse organizations. An externally managed subsystem becomes an integral part of the SoS. As a result, communication between the subsystems becomes difficult.

Examples of SoS usually come from military, transportation, and Network systems.

2.5. DoD Systems of Systems Definition

For the purposes of this dissertation, the definitions from the US Defense Acquisition Guidebook will be used. According to the US Defense Acquisition Guidebook;

> *"An SoS is described as a set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities. For complex SoS, the interdependencies that exist or are developed between and/or among the individual systems being integrated are significantly important and need to be tracked. Each SoS may consist of varying technologies that matured decades apart, designed for different purposes but now used to meet new objectives that may not have been defined at the time the systems were deployed.*
>
> *Both individual systems and SoS conform to the accepted definition of a system in that each consists of parts, relationships and a whole that is greater than the sum of its parts; however, not all systems are SoS. There are distinct*

differences between systems and SoS that should be taken into account in the application of SE to SoS" (DAU, 2017)

According to the DoD Systems Engineering Guide for Systems of Systems not all systems are SoS (DoD, 2008). Therefore, it is essential to understand the differences between a system and SoS per DoD's definition. Table 2.2 provides a summary of distinct differences between systems and SoS per DAU's definition (DAU, 2017).

Table 2.2: Comparing Systems and Systems of Systems (DAU, 2017)

	System	System of Systems (SoS)
Management & Oversight		
Stakeholder Involvement	Clearer set of stakeholders	Two or more levels of stakeholders with mixed, possibly competing interests. The stakeholders represent: 1. the independent and useful systems 2. the aggregation of the independent and useful systems
Governance	Aligned PM and funding. Higher levels of governance such as PEO and AT&L (internal and external governance)	Added levels of complexity due to management and funding for both SoS and systems; No single manager controls all constituent systems in the SoS
Operational Environment		
Operational Focus	Designed and developed to meet operational objectives	Called upon to provide integrated capabilities using systems whose objectives have not been directly derived from current SoS system's objectives
Implementation		
Acquisition/ Development	Aligned to established acquisition process	Multiple system life cycles across acquisition programs, involving legacy systems, systems under development, new developments and technology insertion; Stated capability objectives but may not have formal requirements
Test & Evaluation	Test and evaluation (T&E) of the system is possible	Testing more challenging due to systems' asynchronous life cycles, independence of constituent systems, and the complexity of all the moving parts; Given these challenges, the T&E approach may need to focus on system or subsystem testing in risk areas of the capability and evaluate evidence from SoS level activities or roll-ups of system-level activities
Engineering & Design Considerations		
Boundaries & Interfaces	Focuses on boundaries and interfaces	Focus on identifying systems contributing to SoS objectives and enabling flow of data, control and functionality across and/or between the SoS while balancing needs of systems. The boundaries and interfaces between systems become very important, since they serve as a conduit for data transfer

Performance & Behavior	Ability of the system to meet performance objectives	Performance across the SoS that satisfies SoS user capability needs while balancing needs of the systems
Metrics	Well defined	Difficult to define, agree, and quantify

2.6. DoD Acquisition Lifecycle

The DoD Acquisition Life Cycle is a five-phase cycle with entrance and exit criteria at each phase. Figure 2-2 provides a starting point for SE activities, reviews, and documentation by acquisition phase from Materiel Solution Analysis, Technology Maturation and Risk Reduction, Engineering and Manufacturing Development, Production and Deployment, and Operations and Support (DAU, 2017). It is critical to assess the system's maturity at multiple points during the lifecycle of the project (M. F. Austin & York, 2015; Marc F Austin & York, 2016; Marc F. Austin & York, 2017; DAU, 2017).

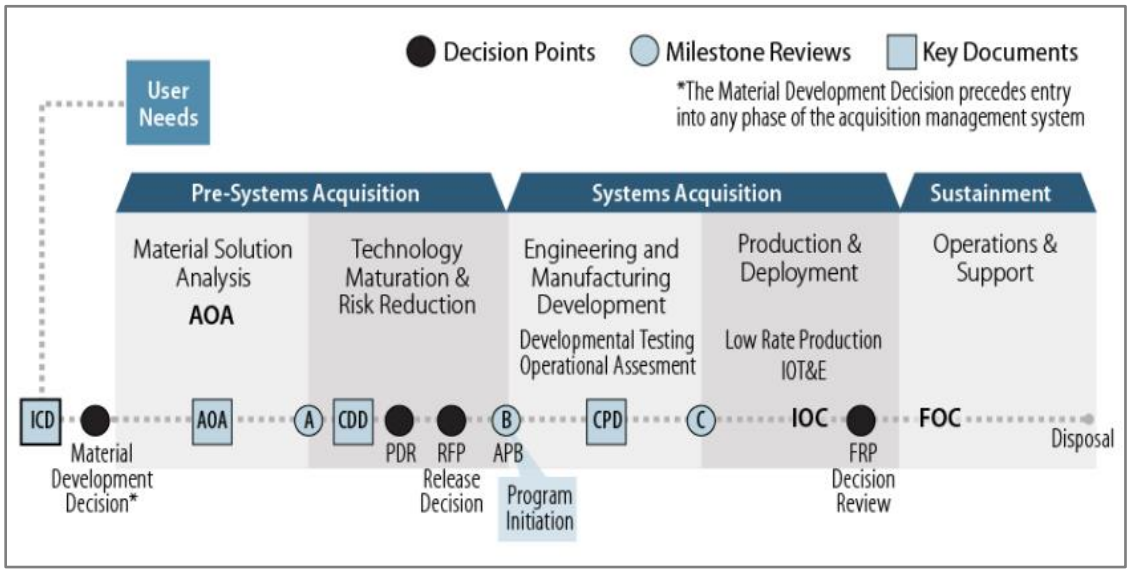


Figure 2.2: DoD Systems Engineering Considerations (Schwartz, 2016)

Phase 1: Materiel Solution Analysis (MSA) During the MSA phase system engineers assess potential materiel solutions. Analysis of Alternatives (AoA), consider the

alternatives in the context of the larger SoS. While conducting AoA, systems engineers assess alternative technologies against the planned solutions for their readiness level and integration risk. AoA should include material cost, total life cycle cost, schedule, concepts of operations, overall risks, and CTEs associated with each proposed alternative. Main systems engineering activities included in this phase; evaluate material solution alternatives against capability requirements, conduct analysis to support preferred material solution, define balanced and achievable performance requirements, develop technical strategy/plans, including potential rapid approaches, and assess technical risk/opportunities; develop mitigation strategies. The MSA phase ends when the AoA is completed and a specific solution is identified which is required to pass through Milestone A. Milestone Decision Authority (MDA) is the official responsible for determining whether a program fulfills the milestone criteria and proceeds.

Phase 2: Technology Maturation and Risk Reduction (TMRR) The TMRR phase is when CTEs are matured to the point that a decision can be made with confidence. During the TMRR phase systems engineers assess CTEs maturity and appropriate risk level associated with the technology. As a part of risk reduction activities, product developers develop prototypes which increases technical confidence and reduces the risk related to integration events. According to DoD 5000.02, If programs have very expensive units, they may be exempt from developing a prototype (DoD, 2015). Main systems engineering activities included in TMRR phase; Complete technology maturation to appropriate risk level, conduct prototyping to reduce technical risk, conduct engineering trades to inform/shape the CDD and program cost/schedule goals, established system architecture (both H/W and S/W), complete preliminary system design to meet

requirements and address design considerations, update risk assessments and technical strategies and plans (DAU, 2017).

Phase 3: Engineering and Manufacturing Development (EMD) A program must pass Milestone B to proceed to the EMD Phase, where a system is designed and developed, all technologies and capabilities are fully integrated into a single system. According to DoD 5000.02 during this phase all sources of risk (including technology maturity) must be mitigated. It is important to note that not all CTEs planned for the system are required to be mature to continue to Milestone B (DoD, 2015). The decision maker may decide to evolutionary acquisition strategy to develop new capabilities in multiple increments (Lorell, Lowell, & Younossi, 2006). evolutionary acquisition strategy requires continues systems maturity assessment. DoD 5000.02 definition of the evolutionary acquisition as follows;

> *"Evolutionary acquisition is an acquisition strategy structured to deliver capability in increments, recognizing, up front, the need for future capability improvements. The objective is to balance needs and available capability with resources, and to put capability into the hands of the user quickly. The success of the strategy depends on phased definition of capability needs and system requirements, and the maturation of technologies that lead to disciplined development and production of systems that provide increasing capability over time".* (DoD, 2015)

Main systems engineering activities included in EMD phase; Complete detailed design to support development/fabrication and test, implement solutions that resolve performance deficiencies, develop production process and plans (both H/W and S/W),

establish/assess product baseline to meet performance and operational needs, update risk assessments and technical strategies and plans, assess production readiness, implements design for exportability (DAU, 2017).

Phase 4: Production and Deployment (P&D) A technology muss pass Milestone C to enter into the P&D phase. In order to pass Milestone C, a technology must satisfy developmental testing, operational assessment, show interoperability with other elements, and MDA must approve the progress. At this stage low rate production starts to demonstrate technologies feasibility. Main systems engineering activities included in P&D phase; Implement solutions to resolve production or operational performance deficiencies, establish and control production process, update risk assessments and technical strategies and plans, initiate sustaining engineering (DAU, 2017).

Phase 5: Operations and Support (O&S) this is the final phase of a system's life. During the O&S phase sustainment and disposal plans are developed. Main systems engineering activities included in O&S phase; Implement changes to resolve operational deficiencies, implement solutions for new capability requirements, execute demilitarization and disposal (DAU, 2017).

To address repeated technology insertion into complex system like weapon systems, system development metrics are needed. The NASA proposed TRL metric to assess the risk associated with technology development (Olechowski, Eppinger, & Joglekar, 2015). Later IRL and SRL metrics introduced to the and academia and industry to address SoS challenges. Next section of this chapter describes System Development Metrics and provides detail calculations.

2.7. System Development Metrics

According to the SRA Engineering Handbook, four metrics are used to evaluate the system readiness throughout SRA development (Marc F. Austin & York, 2017). The first two metrics, TRL and IRL, are assigned using inputs from subject matter experts. The remaining metrics, the component readiness level (CRL) and SRL, are computed by applying various calculation methods. Table 2.3 presents the definitions of TRL, IRL, and SRL as well as the related system development acquisition phases for each metric definition and the literature source for each metric definition.

Table 2.3: List of TRL, IRL, SRL Values, Acquisition Phases, and Source

Scale Level	TRL Hardware Definition (DOD (TRA) 2009)	IRL Definition (Sauser et al. 2010)	SRA Engineering Handbook SRL Definition (Marc F. Austin & York, 2017)	SRL Definition (Sauser et al. 2010 and 2011)	Acquisition Phase Definition (DOD (TRA) 2009)
1	Basic principles observed and reported	Interface defined to characterize component relationship.	System alternative materiel solutions should have been considered	SRL Value: 0.10-0.39 Refine initial concept. Develop system/technology development strategy	Material Solution Analysis (MSA)
2	Technology concept and/or application is formulated.	Characterize technology interface interactions.	System materiel solution should have been identified		
3	Analytical and experimental critical function and/or proof of concept demonstrated.	Compatibility demonstrated between technology elements.	System high-risk immature technologies should have been identified and prototyped		
4	Component validated in a laboratory environment.	Quality assurance of integration quality.	System performance specifications and constraints should have been defined and the baseline has been allocated	SRL Value: 0.40-0.59 Reduce technology risks and determine appropriate set of technologies to integrate into a full system.	Technology Maturation and Risk Reduction (TMRR)
5	Component validated in a relevant environment.	Sufficient control established between technologies to establish, manage, and terminate the integration.	System high-risk component technology development should have been complete; low-risk system components identified		

6	System/subsystem demonstrated in a relevant environment.	Integrated technologies can accept, translate, and structure information.	System component integrability should have been validated	SRL Value: 0.60-0.79 Develop a system or capability increment; reduce integration and manufacturing risk; ensure operational supportability; reduce logistics footprint; implement human systems integration; design for producibility; ensure affordability and protection of critical program information; and demonstrates system integration, interoperability, safety, and utility.	Late Technology Demonstration - or- Engineering & Manufacturing Development (EMD)
7	System prototype demonstrated in a relevant environment.	Integration requirements are validated and verified.	System threshold capability should have been demonstrated at operational performance level using operational interfaces		
8	Actual system completed and qualified through test and evaluation.	Integration completed and mission qualified via test and evaluation.	System interoperability should have been demonstrated in an operational environment	SRL Value: 0.80-0.89 Achieve operational capability that satisfies mission needs.	Production & Deployment (P&D)
9	Actual system proven through successful mission operations.	Integration is mission proven through successful mission operations.	System has achieved initial operational capability and can satisfy mission objectives	SRL Value: 0.90-1.00 Execute a support program that meets operational support performance requirements and sustains the system in the most cost-effective manner over its lifecycle.	Operations & Support (O&S)

The following section will discuss the specific SRA metrics identified in Table 2.3 and demonstrate the general trend of increasingly complex systems under development and the resulting need for metrics to help measure the progression of capability within complex systems.

2.7.1 Technology Readiness Level (TRL)

TRL is a metric to evaluate CTE's maturity progress (DoD, 2009). Technology Readiness Assessment (TRA) Desk book describes a CTE as "A technology element is "critical" if the system being acquired depends on this technology element to meet operational requirements (within acceptable cost and schedule limits) and if the technology

element or its application is either new or novel or in an area that poses major technological risk during detailed design or demonstration." (DoD, 2009). CTE's can be hardware or software.

In the 1970's the NASA developed a seven level TRL metric to address the risk associated with technology development for repeated technology insertion into SoS architectures (Olechowski et al., 2015; B. J. Sauser, Verma, Emmanuel, & Ryan, 2006). By the 1990's this metric had progressed into the nine levels that exist today and has become broadly used across the NASA as an evaluation metric to assess the maturity of a CTE. Following NASA, in 1999, the DoD incorporated a comparable TRL concept. (B. J. Sauser et al., 2006). A nine level TRL levels, definitions, and descriptions DOD, NASA, and other organizations use are provided in Table 2.4.

Table 2.4: Technology Readiness Level Definitions (DoD, 2009)

TRL	Definition	Description
9	Actual system proven through successful mission operations.	Actual application of the technology in its final form and under mission conditions, such as those encountered in operational test and evaluation (OT&E). Examples include using the system under operational mission conditions.
8	Actual system completed and qualified through test and demonstration.	Technology has been proven to work in its final form and under expected conditions. In almost all cases, this TRL represents the end of true system development. Examples include developmental test and evaluation (DT&E) of the system in its intended weapon system to determine if it meets design specifications.
7	System prototype demonstration in an operational environment.	Prototype near or at planned operational system. Represents a major step up from TRL 6 by requiring demonstration of an actual system prototype in an operational environment (e.g., in an aircraft, in a vehicle, or in space).
6	System/subsystem model or prototype demonstration in a relevant environment.	Representative model or prototype system, which is well beyond that of TRL 5, is tested in a relevant environment. Represents a major step up in a technology's demonstrated readiness. Examples include testing a prototype in a high-fidelity laboratory environment or in a simulated operational environment.
5	Component and/or breadboard validation in a	Fidelity of breadboard technology increases significantly. The basic technological components are integrated with reasonably realistic supporting elements to they can be tested in a simulated

	relevant environment.	environment. Examples include "high-fidelity" laboratory integration of components.
4	Component and/or breadboard validation in a laboratory environment.	Basic technological components are integrated to establish that they will work together. This is relatively "low fidelity" compared with the eventual system. Examples include integration of "ad hoc" hardware in the laboratory.
3	Analytical and experimental critical function and/or characteristic proof of concept.	Active R&D is initiated. This includes analytical studies and laboratory studies to physically validate the analytical predictions of separate elements of the technology. Examples include components that are not yet integrated or representative.
2	Technology concept and/or application formulated.	Invention begins. Once basic principles are observed, practical applications can be invented. Applications are speculative, and there may be no proof or detailed analysis to support the assumptions. Examples are limited to analytic studies.
1	Basic principles observed and reported.	Lowest level of technology readiness. Scientific research begins to be translated into applied research and development (R&D). Examples might include paper studies of a technology's basic properties.

Although TRL metric has been used to evaluate the maturity of technology development, it presents shortfalls to provide a complete representation of the difficulty of integration of the subject technology or subsystems into a complex system (B. J. Sauser, Long, Forbes, & McGrory, 2009; B. J. Sauser, Marquez, Henry, & DiMarzio, 2008). Since the TRL cannot evaluate complex systems inter-connections (Azizian et al. 2009 and 2011; Mankins 2002; Sauser et al. 2008, 2009, and 2011), IRLs were subsequently developed by Sauser et al. to address this limitation (B. Sauser, Ramirez-Marquez, Verma, & Gove, 2006).

2.7.2 Integration Readiness Level (IRL)

Sauser et al. proposed IRLs to evaluate the capabilities of system element interfaces to address repeated technology insertion into complex SoS architectures (London, Holzer, Eveleigh, & Sarkani, 2014; Brian Sauser, Dinesh Verma, Jose Ramirez-Marquez, & Ryan

Gove, 2008). Similar to the development of the TRL scale, IRLs were limited to distinct CTEs. IRLs are also limited to considering only the readiness of the involved technology relationships among distinct elements and systems (Long, 2011). The IRL method developed by Sauser et al. is currently the most widely used method found in the literature (Brian Sauser et al., 2008). A nine level IRL levels, definitions, and descriptions are provided in Table 2.5.

Table 2.5: Integration Readiness Level Definitions (Brian Sauser et al., 2008)

IRL	Definition	Description
9	Integration is **Mission Proven** through successful mission operations.	IRL 9 represents the integrated technologies being used in the system environment successfully. In order for a technology to move to the TRL 9, it must first be integrated into the system and then proven in the relevant environment; thus, progressing IRL to 9 also implies maturing the component technology to the TRL 9.
8	Actual integration completed and **Mission Qualified** through test and demonstration in the system environment.	IRL 8 represents not only the integration-meeting requirements, but also a system-level demonstration in the relevant environment. This will reveal any unknown bugs/defects that could not be discovered until the interaction of the two integrating technologies was observed in the system environment.
7	The integration of technologies has been **Verified and Validated** with sufficient detail to be actionable.	IRL 7 represents a significant step beyond IRL 6; the integration has to work from a technical perspective, but also from a requirements perspective. IRL 7 represents the integration meeting requirements such as performance, throughput, and reliability.
6	The integrating technologies can **Accept, Translate, and Structure Information** for its intended application.	IRL 6 is the highest technical level to be achieved; it includes the ability to not only control integration, but to specify what information to exchange, to label units of measure to specify what the information is, and the ability to translate from a foreign data structure to a local one.
5	There is sufficient **Control** between technologies necessary to establish, manage, and terminate the integration.	IRL 5 simply denotes the ability of one or more of the integrating technologies to control the integration itself; this includes establishing, maintaining, and terminating.
4	There is sufficient detail in the **Quality and Assurance** of the integration between technologies.	Many technology-integration failures never progress past IRL 3, due to the assumption that if two technologies can exchange information successfully, then they are fully integrated. IRL 4 goes beyond simple data exchange and requires that the data sent is the data received and there exists a mechanism for checking it.
3	There is **Compatibility** (i.e., common language)	IRL 3 represents the minimum required level to provide successful integration. This means that the two

	between technologies to orderly and efficiently integrate and interact.	technologies are able to not only influence each other, but also to communicate interpretable data. IRL 3 represents the first tangible step in the maturity process.
2	There is some level of specificity to characterize the **Interaction** (i.e., ability to influence) between technologies through their interface.	Once a medium has been defined, a "signaling" method must be selected such that two integrating technologies are able to influence each other over that medium. Since IRL 2 represents the ability of two technologies to influence each other over a given medium, this represents integration proof-of-concept.
1	An **Interface** between technologies has been identified with sufficient	This is the lowest level of integration readiness and describes the selection of a medium for integration.
0	No Integration	No integration between specified components has been planned or intended.

Although the IRL represented an improvement to the TRL by assessing technological system component connections, a systematic method was important for assessing complex systems in their entirety rather than only individual components and their interplays. A new framework was required to assess the maturity of a complex system consisting of involving CTEs and interrelated relationships. This new framework was named the SRL by Sauser et al. (B. J. Sauser et al., 2006).

2.7.3 System Readiness Level (SRL)

The SRL was originally established at the Systems Development Maturity Laboratory (SysDML) to provide system level metrics that allows continuing assessment over the development life cycle into the entire system including its interfaces. (Ramirez-Marquez et al., 2009, Sauser et al. 2012). SRLs mathematically combine element TRL values with IRL values to create a distinct measure of whole system development progress (B. Sauser, Ramirez-Marquez , Nowicki, Deshmukh, & Sarfaraz, 2011; B. J. Sauser et al., 2006). SRL values are computed for each element, and these subsystem element values are used to compute a system-level SRL value. Initial SRL calculation methods use pairwise matrix calculations to characterize the system readiness. Recent SRL studies

have examined the probabilistic distributions of SRL values (Tan, Ramirez-Marquez, & Sauser, 2011; Tan, Sauser, & Ramirez-Marquez, 2011), tropical algebra computational approaches (McConkie, Mazzuchi, Sarkani, & Marchette, 2013), incidence matrix approaches (London et al., 2014), and Petri net approaches (Thal et al., 2018). SRL levels, definitions, and descriptions are provided in Table 2.6 (B. J. Sauser et al., 2006).

Table 2.6: Systems Readiness Level Definitions (B. J. Sauser et al., 2006)

SRL	Definition	Description
0.90 to 1.00	Operations & Support	Execute a support program that meets operational support performance requirement and sustains the system in the most cost-effective manner over its total lifecycle.
0.70 to 0.89	Production	Achieve operational capability that satisfies mission needs.
0.60 to 0.79	System Development & Demonstration	Develop system capability or (increments thereof); reduce integration and manufacturing risk; ensure operational supportability; reduce logistics footprint; implement human systems integration; design for production; ensure affordability and protection of critical program information; and demonstrate system integration, interoperability, safety and utility.
0.40 to 0.59	Technology Development	Reduce technology risks and determine appropriate set of technologies to integrate into a full system.
0.10 to 0.39	Concept Refinement	Refine initial concept; develop system/technology strategy.

In 2015, The International Systems Readiness Assessment Community of Interest (ISRACOI) was formed by group of experts from government, industry, non-profits, and academia. ISRACOI published an SRA Engineering Handbook which serves as a guide for the systems engineerings and program managements for performing SRAs. In this SRA Handbook the authors proposed a SRL Transional model to translate the 0 to 1 calculated SRL values to 1 to 9 scale inline with TRL and IRL scaling for ease of understanding. Table 2.7 provides summary of the SRA Handbook SRL definitions.

Table 2.7: Systems Readiness Level Definitions from SRA Handbook (Marc F. Austin & York, 2017)

SRL	Definition
9	System has achieved initial operational capability and can satisfy mission objectives.
8	System interoperability should have been demonstrated in an operational environment.
7	System threshold capability should have been demonstrated at operational performance level using operational interfaces.
6	System component integrability should have been validated.
5	System high-risk component technology development should have been complete; low-risk system components identified.
4	System performance specifications and constraints should have been defined and the baseline has been allocated.
3	System high-risk immature technologies should have been identified and prototyped.
2	System materiel solution should have been identified.
1	System alternative materiel solutions should have been considered.

Although SRL scale 1 to 9 is under development, academia started to use the scale to align TRL, IRL and SRL scales (Marc F. Austin & York, 2017; Knaggs et al., 2017; Ross, 2016).

2.7.3.1 SRL Calculation Methods

Five main mathematical methodologies for computing SRL values are available in the literature: the Sauser SRL method – S_SRL (B. Sauser et al., 2011; Brian Sauser, Dinesh Verma, Jose Ramirez-Marquez, & Ryan Gove, 2008; Brian Sauser et al., 2008), graph theory approach method – GT_SRL (Garrett, Anderson, Baron, & Moreland), tropical algebra method – TA_SRL (McConkie et al.), incidence matrix approach method – IM_SRL (London et al., 2014), and petri net approach – PN_SRL (Thal et al., 2018). Table 2.8 is expanded from the table in London et al. and succinctly lists the fundamental equation formulations of these existing SRL calculation methods (London et al., 2014).

Table 2.8: Legacy SRL Model Calculation Methodology and Standard Equation Forms
(adapted from (London et al., 2014))

SRL Model	Calculation Methodology	Standard Mathematical Form
Sauser SRL Method	Pairwise Matrix Multiplication	$S_SRL = \frac{1}{n}\sum_{i=1}^{n}\left[k_{SRL}\left[\left(\frac{1}{9}IRL_{SRL}^{n\times n}\right) * \left(\frac{1}{9}TRL_{n,1}\right)\right]\right]$
Graph Theory SRL Method	Pairwise Matrix Multiplication	$GT_SRL = \frac{1}{n}\sum_{i=1}^{n}\left[k_{GTSRL}\left[\left(\frac{1}{9}IRL_{GSRL}^{n\times n}\right) * \left(\frac{1}{9}TRL_{n,1}\right)\right]\right]$
Tropical Algebra SRL Method	Min-Plus Tropical Algebra	$TA_SRL = \oplus_{i=1}^{n}\left[(IRL_{i,1} \otimes TRL_1) \oplus \cdots \oplus (IRL_{i,n} \otimes TRL_n)\right]$
Incidence Matrix SRL Method	Cartesian Product Minimum Value	$ITRL_{\min} = \left[min\left(min\left(\left\{IRL_{i,j}^{T} \rangle 0\right\}_{i=1}^{m} \times \left\{TRL^{T}\right\}\right)\right)\right]$
Petri Net Method	Eigenvalue concept of Incidence Matrices	$SRL_{capacity} = \frac{\sum_{m=1}^{j}\left[\frac{\sum_{i=1}^{n}\left((k\times TRL)+IRL_i\right)/(c+1)}{n}\right]_m}{j}$

The details of these alternate methods will not be discussed here but the interested readers should refer two papers by McConkie et al. (2011) and London et al. (2014) which provide an excellent review of the recent literature on mathematical methodologies for computing SRL values.

2.7.3.1.1 Sauser SRL Calculation Method

In this study, the Sauser SRL calculation method of Sauser et al. (Brian Sauser et al., 2008) were applied. This method will be referred as S_SRL. The descriptions for the TRL and IRL follow those listed in Table 2.3.

According to Sauser et al., the SRL is a function of the TRL and IRL matrices. The TRL matrix is described as a vector with n elements, where TRL*i* presents the TRL of element *i*, as shown in Equation 1.

$$\left[TRL\right]_{n\times 1} = \begin{bmatrix} TRL_1 \\ TRL_2 \\ \cdots \\ TRL_n \end{bmatrix} \text{ (Eq. 1)}$$

The IRL matrix presented in Equation 2 provides a symbolic matrix for the IRL of a system with *n* elements. The integration among elements *i* and *j* is represented by element IRL*ij* in the IRL matrix. The notional integration of element *i* to itself is shown by IRL*ii* and is presumed to be an IRL of value 9.

$$\left[IRL\right]_{n\times n} = \begin{bmatrix} IRL_{11} & IRL_{12} & \cdots & IRL_{1n} \\ IRL_{21} & IRL_{22} & \cdots & IRL_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ IRL_{n1} & IRL_{n2} & \cdots & IRL_{nn} \end{bmatrix} \text{ (Eq. 2)}$$

Next, the TRLs and IRLs equivalent to values from 1 to 9 are normalized. Then, a SRL matrix is produced by generating the matrix product of the IRL and TRL matrices to compute the SRL value from the constituent TRL and IRL values, as presented in Equation 3.

$$\left[SRL\right]_{n\times 1} = \left[IRL\right]_{n\times n} \times \left[TRL\right]_{n\times 1} \text{ (Eq. 3)}$$

Statistically, for a system with *n* technology elements, the SRL is given by Equation 4. TRL*i* represents the distinct TRLs, and the IRL*ij* entries are the distinct IRLs among the elements. SRL*i* characterizes the maturity level of element *i*, thereby representing the maturity of this specific element and all of its connections and interfaces to adjoining elements.

$$[SRL] = \begin{bmatrix} SRL_1 \\ SRL_2 \\ \cdots \\ SRL_n \end{bmatrix} = \begin{bmatrix} IRL_{11}TRL_1 + IRL_{12}TRL_2 + \cdots + IRL_{1n}TRL_n \\ IRL_{21}TRL_1 + IRL_{22}TRL_2 + \cdots + IRL_{2n}TRL_n \\ \cdots \\ IRL_{n1}TRL_1 + IRL_{n2}TRL_2 + \cdots + IRL_{nn}TRL_n \end{bmatrix} \quad \text{(Eq. 4)}$$

To obtain a normalized SRL value, the corresponding SRL*i* for each element *i* is then divided by n_i, as presented in Equation 5. The value n_i represents the number of integrations of element *i* with any element, including the integration of the element with itself (M. F. Austin & York, 2015).

$$ElementSRL = \frac{SRL_i}{n_i} \quad \text{(Eq. 5)}$$

The composite SRL for the system is the arithmetic mean of the calculated element SRLs, as presented in Equation 6, where *n* represents the number of elements.

$$SRL = \frac{\frac{SRL_1}{n_1} + \frac{SRL_2}{n_2} + \cdots + \frac{SRL_2}{n_n}}{n} \quad \text{(Eq. 6)}$$

Equation 7 combines Equations 1-6 into one formula.

$$SRL = \frac{1}{n}\sum_{i=1}^{n}\left[k_{SSRL}\left[\left(\frac{1}{9} IRL_{SSRL}^{n\times n}\right) * \left(\frac{1}{9} TRL_{n,1}\right)\right]\right] \quad \text{(Eq. 7)}$$

2.7.4 Limitation of the Current SRA Approaches

SRA methods have been demonstrated to be valuable for understanding the development status of complex systems of developing technologies in DoD and Department of Energy (DoE) applications (Knaggs et al., 2017). However, current SRA methods present shortcomings when practitioners apply SRA to a larger system in which significant investments are required to assess technologies and integrate them into larger systems (Atwater & Uzdzinski, 2014; Knaggs et al., 2017). These systems are often

decomposed into subsystems that can be controlled independently and whose cooperative specific activities produce the performance of the larger system (M. & A., 2004; Ross, 2016). According to Ross,

> *"the Sauser approach may make sense for a single mission or project, such as the deployment of a new software system. However, it has some serious drawbacks for use in research and development where planners need to decide what technologies to develop for the eventual deployment of a new platform, weapon, or system." (Ross, 2016).*

Knaggs et al. attempted to solve this problem and investigated nesting SRL method to overcome this challenge (Knaggs et al., 2017). Knaggs et al. described nesting as "the treatment of a subsystem of components as a single component technology in an SRA evaluation" (Knaggs et al., 2017). The study applied S_SRL calculations and extended to focus

> *"on the net impact of calculating the system readiness of a subsystem of technology components than extended to calculating an equivalent TRL for the subsystem treated as a single element technology later including this equivalent TRL and subsystem interfaces in a SRA for a larger system" (Knaggs et al., 2017).*

The results of the Knaggs et al. study revealed a significant difference in the SRL scores of the nested and non-nested models (Knaggs et al., 2017). The authors concluded that considerably fewer elements and interfaces were included in the nested model than in the full system model (not a nested model) and suggested further investigations that focus on nesting methods (Knaggs et al., 2017).

Unlike the SRL methods described above, “the DSM is a network modeling tool used to represent the elements comprising a system and their interactions, thereby highlighting the system's architecture (or designed structure)” (Browning, 2001, 2016). The DSM is particularly suitable for applications in the development of complex engineered systems. The DSM has become a promising tool for designers to emphasize system element relationships and is also used to cluster elements into modules that have strong relationships among their elements (AlGeddawy, 2014; Brady, 2002; Thebeau, 2001). Current SRL methods do not focus on the relationships among the system elements. This lack of focus on relationships becomes a problem when decision makers are reviewing a complex system composed of a myriad of elements that results in complex network system diagrams or a myriad of lines of an adjacency matrix offered by the S_SRL methodology.

Application of the DSM tool in the current SRL calculation method may improve system readiness decision assessments. Compared with the current network system diagram methods (such as the SRL methods described above), the main benefit of the DSM is the graphical nature of the matrix presentation of complex systems. This graphical nature provides a compact, simply scalable, and user-friendly view of the system architecture compared to network system diagram methods (Eppinger & Browning, 2012).

The DSM depiction has advantages over a graph or matrix view in that the DSM is more compact, scalable, and readable with increases in the system size while also depicting important system architectural characteristics, such as modules, loops and process iterations (Browning, 2001, 2016). The following section will further discuss the

implementation and application of the DSM to SRAs of systems with complex architectures.

This study proposes a framework to demonstrate how a current SRL calculation method can be incorporated into the DSM tool and is presented to demonstrate the benefits of DSM for improved system readiness decision assessments.

2.8. Design Structure Matrix (DSM)

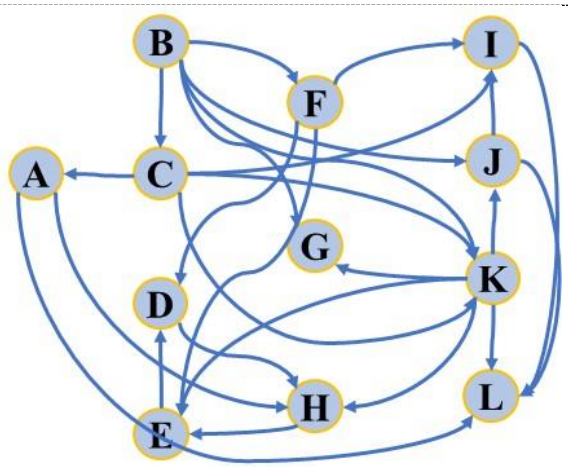
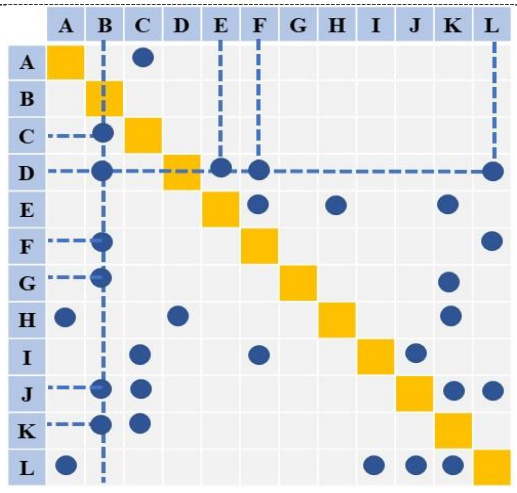
The DSM is a network modeling tool for representing and evaluating system models in a variety of application areas (Eppinger & Browning, 2012). The DSM has the benefits of simplicity and brevity in illustration and can also emphasize significant patterns in system architectures if supported by proper analysis (A. Yassine & Braha, 2003). The DSM offers an additional advantage by providing a practical, scalable, and intuitive solution for representing and analyzing system architectures (Eppinger & Browning, 2012).

The DSM represents a system configuration or system network diagram as a square N×N matrix, showing the relationships between the set of N system elements, with the rows and columns identically labeled and ordered and the off-diagonal elements indicating interactions among the diagonal elements. The regions above and below the diagonal in the matrix denote the directionality of any element connectional relationships, thereby making the basic DSM system representation fundamentally equivalent to a directed graph. In certain scenarios, the DSM depiction has advantages over a graph in that the DSM is more compact, scalable, and readable with increases in the system size while also depicting important system architectural characteristics, such as modules and process iterations (Browning, 2001, 2016). The DSM is a highly flexible tool that has

been used to model many types of systems, as will be discussed below sections 2.8.1-2.8.3.

2.8.1 DSM Structure

Steward developed a priority table displaying the different design activities together with their predecessors. Subsequently, these activities were organized in a square N×N priority matrix with marks in the matrix displaying relationships between the activities (Steward, 1981). Figure 2.3 shows a simple process consisting of twelve elements that are shown as a network diagram on the far left (A), a base DSM representation in the middle (B) and a partitioned DSM representation on the right (C) (A. Yassine & Braha, 2003).

A) Network diagram representation	
B) A base DSM representation	

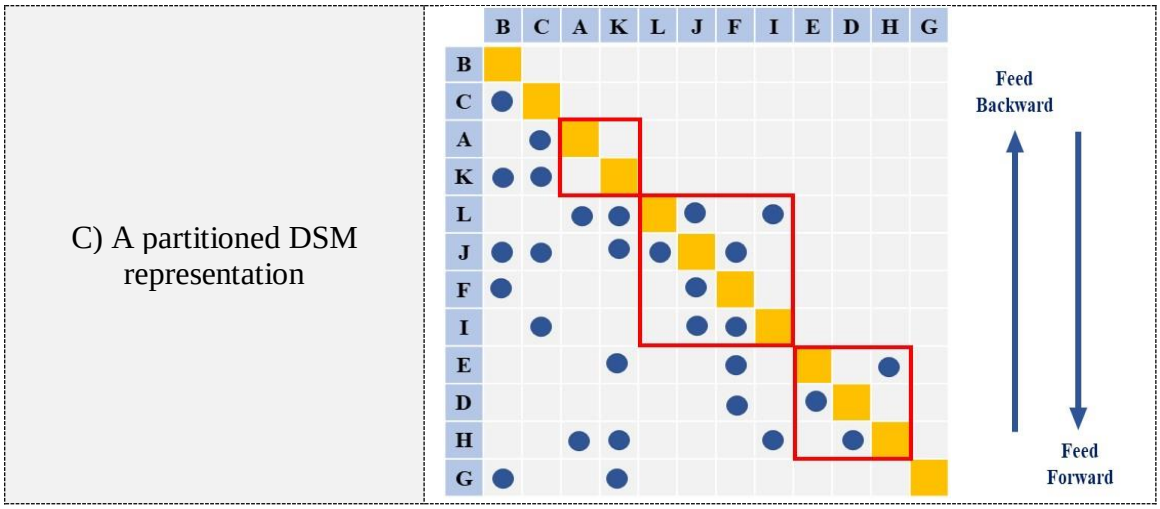


Figure 2.3: DSM overview: (A) network diagram; (B) base DSM representation; and (C) partitioned DSM representation (A. Yassine & Braha, 2003).

According to the marks presented in Figure 2.3, element D provides information to elements B, E, F, and L. The mark in the off-diagonal cells denotes an interface that is both an input and output, contingent on whether one takes the viewpoint of its provider or its dependent (Browning, 2001; Eppinger & Browning, 2012; Garg et al., 2017; Olechowski et al., 2015). Additionally, the DSM representation can be extended to include other features of the interactions, such as strength, impact, importance, and weight. These features might be represented by using numbers, symbols, or colors instead of simply the marks in each cell (Eppinger & Browning, 2012).

Marks below the diagonal illustrate forward information flow that affects a later element. Marks above the diagonal illustrate information flow fed back to an earlier element. In other words, the earlier element should be revisited based on the new information. Partitioning analysis eliminates or reduces the iterations creating rework (A. A. Yassine, 2004).

2.8.2 DSM Structure

Modularity is a design approach that segments a system into subsystems called modules, that can be independently operated. When a system is modularized, the subsystems of its design are rearranged and allocated to modules according to the purpose. From an engineering standpoint, a modularization generally has two drives; be able to manage the complexity and mitigate future uncertainty.

Designing for modularity and utilizing product architecture DSM allows decision makers to plan subsystems, to understand these subsystems' connections, and to assess the risk of new technology (Eppinger & Browning, 2012). According to Eppinger, typical applications of DSM includes "enhancement of product modularity, which determines subsystem boundaries, relates to component sharing across product lines, and affects the difficulty of outsourcing and system integration" (Eppinger & Browning, 2012). While Figure 2.4 depicts four primary types of DSM models it also presents how modular DSMs look.

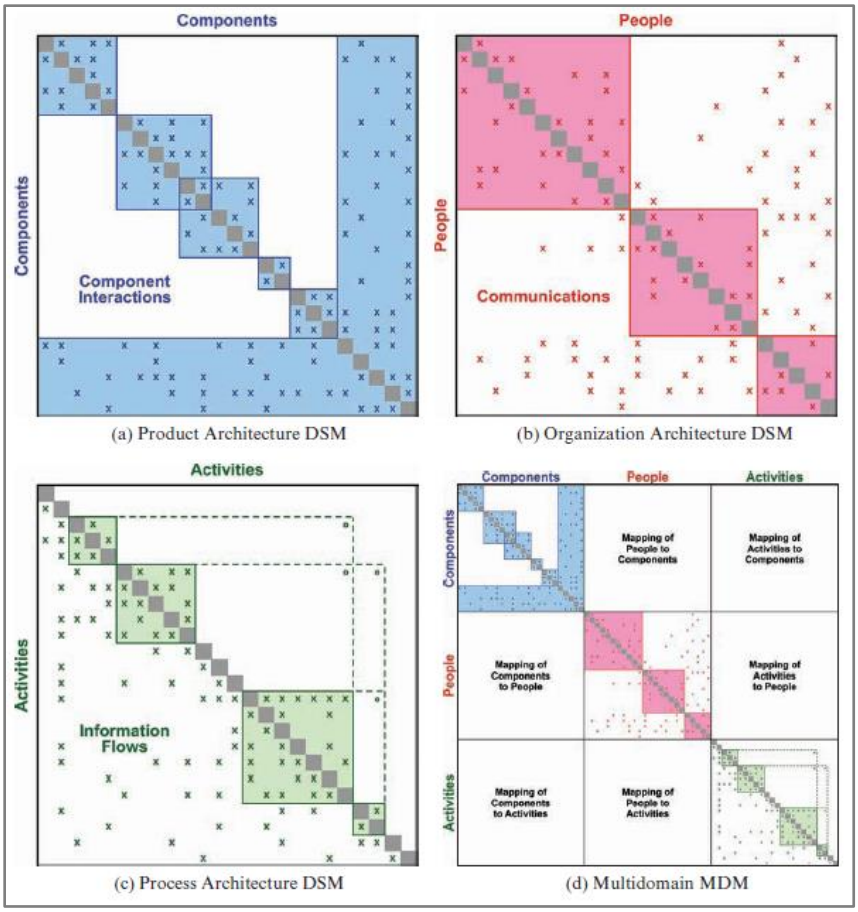


Figure 2.4: The four primary types of DSM models (Eppinger & Browning, 2012)

Additionally, modularity allows DSM user to collapse and expand clusters (subsystems) to facilitate the identification of subsystem boundaries without losing quantitative information. Clusters that share interfaces can be gathered into systems. The clusters can also be collapsed to show only the assembly structure, as shown in Figure 2.5. From the initial clustering, shown in blue boxes, a new cluster, highlighted in green, is formed due to the high connectivity between the two modules (Levandowski, Müller, & Ola, 2016). The orange lines depict the interfaces between the different clusters. A comparison of the initial clustering (first figure) and the partially collapsed state illustrates that the interfaces are preserved and reduced to a single interaction between the

cluster and element. This process can be repeated until the optimized matrix view is formed.

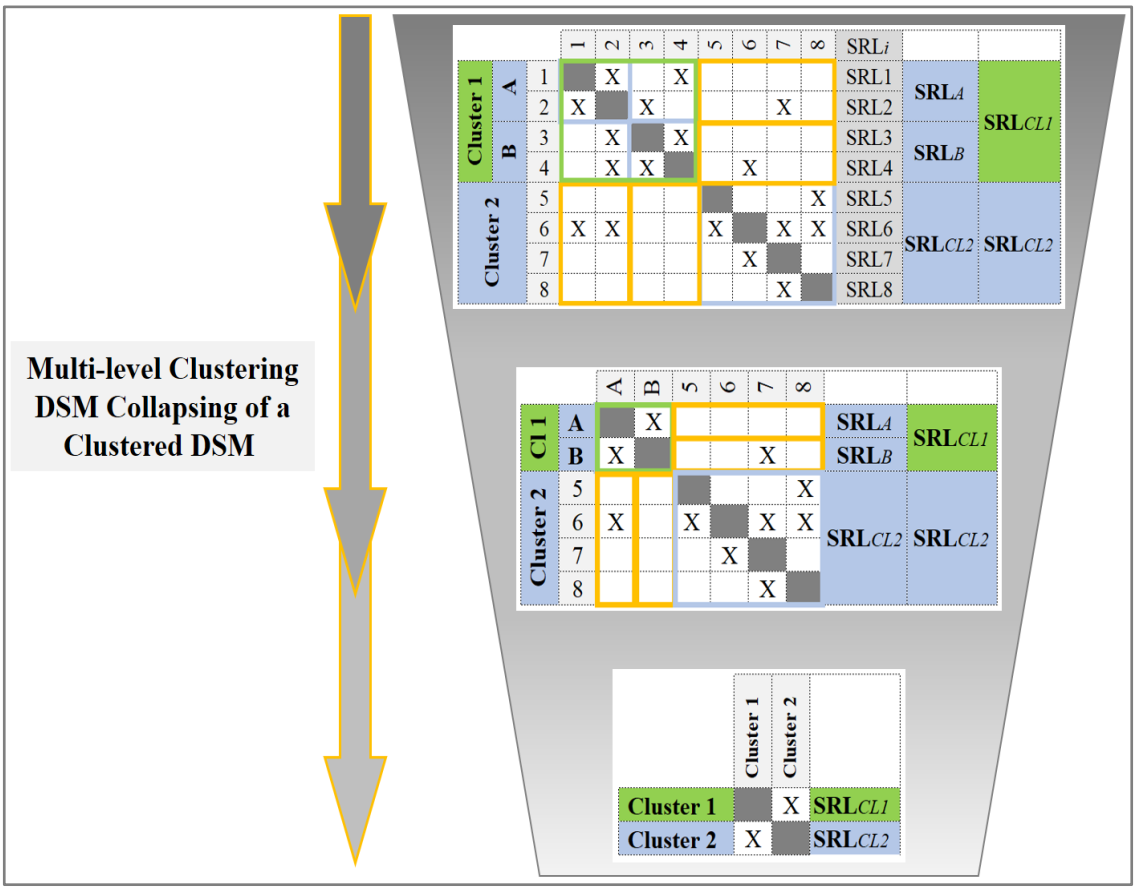


Figure 2.5: Example of Multi-level Clustering and Collapsing of a Clustered DSM
(Adapted from (Levandowski et al., 2016))

2.8.3 Past Applications of DSMs

DSMs have been used in various applications since being introduced by Steward in the early 1960s. However, the use of DSMs did not become widespread until the early 1980s (Steward, 1981). By utilizing the DSM approach to design variables and methods, Steward clarified that the features of the DSM approach would include how design iterations and reviews are handled and how data flows during the design stage. Eppinger et al. further developed Steward's technique of the DSM by applying it as a modeling tool for managing concurrent engineering for design and manufacturing to improve efficiency

and reduce cycle time (Eppinger, Whitney, Smith, & Gebala, 1991; Smith & Eppinger, 1990). In the 1990s, a group of MIT researchers further extended DSM applications beyond Steward's time-based models to include static architectural models, complex organizations, systems of systems, and highly complex products (Browning, 2001, 2016; Eppinger & Browning, 2012).

In 2016, Browning conducted an extensive systematic literature review to identify a set of prominent DSM papers. The results of this study highlighted over 3,000 DSM-related publications in the existing literature at that time. Their study highlighted that DSMs were leveraged within the aerospace, energy, automotive, construction, shipbuilding, healthcare, innovation systems, computer, software development, network system control, and sensor system (large-scale) industries (Browning, 2016). The literature review performed by Browning systematic literature indicated that there are many studies utilizing the DSM methodology to assess element level technology readiness, but the methodology has not been extended to system-level readiness assessments (Browning, 2016).

2.8.4 Building and Analyzing a Product Architecture DSM

This dissertation focusses on a product architecture DSM model involving a static DSM categorized as a component-based DSM. The product architecture DSM provides an effective illustration for elements in a particular product and their connectional relationships (Browning, 2001, 2002). Furthermore, the product architecture DSM also documents both the product decomposition and element interfaces. The DSM can also be analyzed via clustering, which improves architectural understanding (Browning, 2016; Eppinger & Browning, 2012).

The basic procedure for building and analyzing a product architecture DSM is described in Table 2.9.

Table 2.9: Basic procedure for building and analyzing a product architecture DSM
(Browning, 2001; Eppinger & Browning, 2012; A. Yassine & Braha, 2003; A. A. Yassine, 2004)

Steps		Definition
Building a Product Architecture DSM		
1	Decompose	Decompose the overall product or system into its subsystems and/or components. When developing a System of Systems, the system is typically decomposed into smaller components or subsystems that can be handled more easily.
2	Define Boundaries	Define the system boundaries to include all of the relevant components and interactions.
3	Lay Out the DSM	Lay out the square DSM with components, labeling the rows and columns, grouped into subsystems or modules if appropriate.
4	Identify Interactions	Identify the known interactions between the components. Most interactions in product architecture DSMs are symmetric. That is, if component A interacts with component B, then B also interacts with A.
5	Mark Interactions	Represent the interactions using marks or values in the DSM cells.
6	Apply Mathematical Algorithms	Once the DSM is constructed, mathematical algorithms can be applied to suggest changes to the systems.
Analyzing a Product Architecture DSM		
7	Partitioning	Partitioning is the process of rearranging the DSM rows and columns such that the new DSM arrangement contains minimum feedback marks. During partitioning, the following steps should be considered: 1. Identified system elements that can be determined without input from the remaining elements in the matrix. Place those elements in the top of the DSM. 2. Identified system elements (or tasks) that deliver no information to other elements in the matrix. Place those elements in the bottom of the DSM. Once an element is rearranged, it is removed from the DSM. **3.** Marks below the diagonal are synonymous with marks above the diagonal and represent interactions between the components. In other words, clusters contain most of the interactions (A. A. Yassine, 2004)**.**
8	Clustering	This is a form of partitioning analysis that reorders the rows and columns of the DSM to group the components according to some objective. Identified subsets of DSM elements (clusters or modules) are mutually exclusive. Clusters should contain most of the interfaces.

This section established the basic SRL calculation framework present in the literature and the basic methods involved in applying the standard DSM methodology to a given system. The next section discusses the research methodology and data collection for this dissertation.

Chapter 3. Research Goals and Hypotheses

3.1. Research Motivations and Goals

System Readiness Assessment (SRA) methods provide a basis for assessing the current readiness of systems under development yet SRAs have not been extended to multi-tiered technology development process for complex system applications (Knaggs et al., 2017). A detailed literature research (Garrett et al., 2011; Knaggs et al., 2017; London et al., 2014) suggests that Nesting methods, which seek to describe system readiness behavior of cascading system architectures, may provide opportunities to improve current SRA approaches. Recent research (Knaggs et al., 2017) has applied limited nesting approaches to SRA methodology to assess the system readiness for complex systems. This recent research suggests that more comprehensive nesting approaches may yield improved SRA prediction results that are required to solve the complex systems' system readiness. Analysis of the problem led to the conclusion that a new System Readiness Assessment methodology was needed to resolve the limitation of proposed methods used to measure system readiness for multi-tiered complex systems.

3.2. Research Questions and Hypotheses

The overarching research question within this study was as follows:

- **Research Question 1:** Can DSM methodology be utilized for systems readiness assessment?
- **Research Question 2:** Can DSM methodology be utilized for decision making in complex engineering systems?

- **Research Question 3:** Can proposed DSM framework combine system architecture concepts and analysis with system readiness levels (SRLs) to focus improve assessment of attention on high-risk components and interfaces?
- **Research Question 4**: Can a DSM methodology effectively improve a multi-tiered complex system's system readiness compared to Nesting approach (Knaggs et al., 2017)?

The above questions become the foundation for developing the detailed hypotheses listed below:

- **Research Hypothesis 1:** There is significant difference between the system readiness of a multi-tiered complex system evaluated with a Nesting approach and the system readiness evaluated using a DSM approach.
  - The purpose of this demonstration is to compare the risk identification framework produced by Knaggs et al. with the proposed risk identification framework (Knaggs et al., 2017).
- **Research Hypothesis 2:** There is no significant difference between DSM framework that combines system architecture concepts and analysis with system readiness levels (SRLs) and Sauser SRL (Full System SRL).
  - The purpose of this demonstration is to compare/analyze the system readiness estimations generated with DSM approach and Sauser SRL (Full System SRL).

Chapter 4. Research Method

This research employed a modeling and simulation approach similar to the SRL calculation modeling structure employed by London et al. but that differs in that the system previously evaluated by Knaggs et al. and two NSWSs, which are larger in system size and order, are evaluated rather than the simple system configurations evaluated by London et al. (Knaggs et al., 2017; London et al., 2014). Figure 4.1 presents the methodology employed for this study.

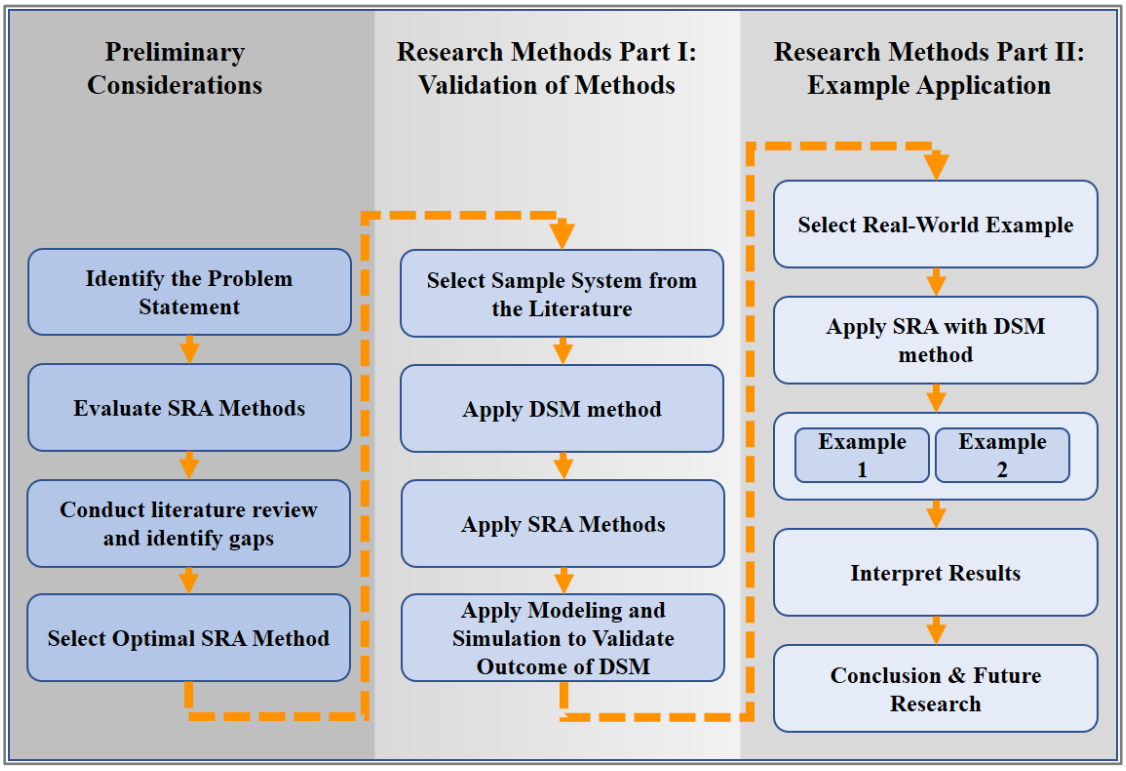


Figure 4.1: Research Methodology

This dissertation included three main phases; Preliminary Considerations, Research Methods Part I: Validation of Methods, and Research Methods Part II: Example Application. The Preliminary Considerations phase identified the initial steps in preparing our research approach (identify the problem statement, evaluate the SRA

methods, conduct a literature review and identify gaps, and select the optimal SRA method). The Research Methods Part I: Validation of Methods validated DSM_SRL methodology (select a sample system from the literature, apply the DSM method, apply the SRA methods, and apply modeling and simulation to validate the data output of the DSM). Finally, Research Methods Part II: Example Application phase applied DSM_SRL methodology to specific example applications (select real-world example, apply SRA with the DSM method to real-world examples, and interpret the results).

The sample system from Knaggs et al. with order n=19 was selected because it provided an example of a system network diagram that contained a nesting algorithm in its formulation. The system network diagram representation of Knaggs et al. with the TRL and IRL values is presented in Figure 4.2 (Knaggs et al., 2017).

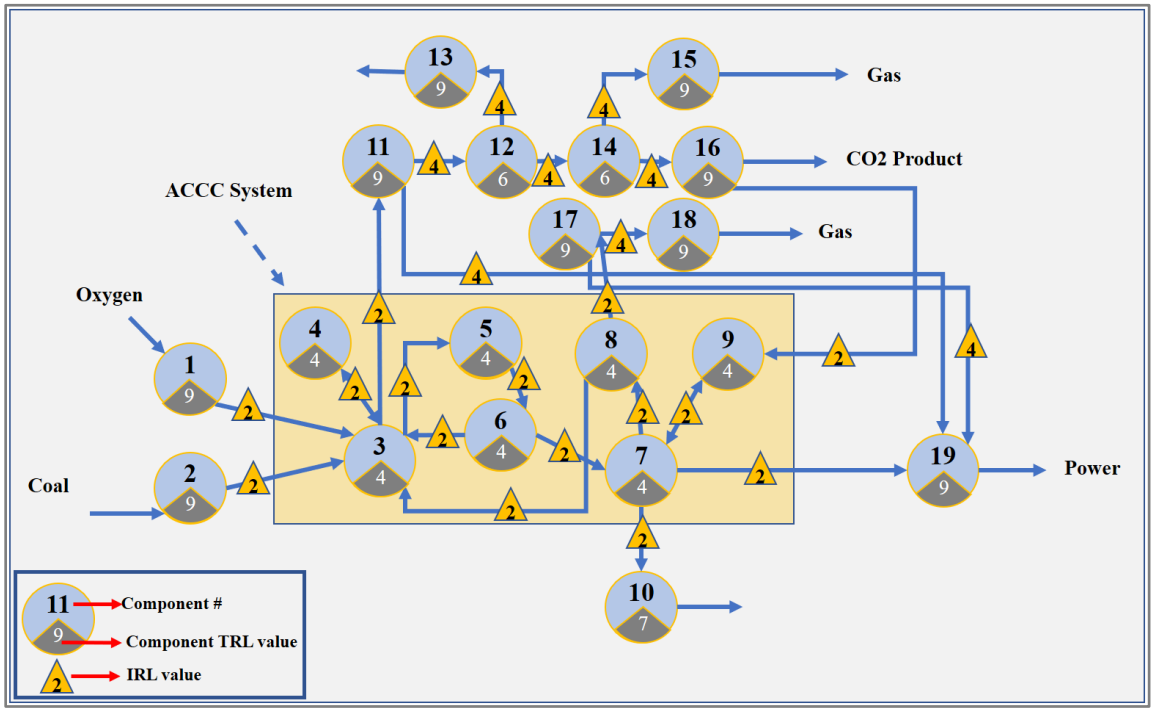


Figure 4.2: Network diagram representation of the system from Knaggs et al with the TRL and IRL values included

The next step was to generate the DSM from the network diagram. Figure 4.3 presents the DSM of the system from Knaggs et al., with the TRL and IRL values and each component's interfaces with other components (Knaggs et al., 2017). Additionally, Figure 4.3 presents both the network diagram and DSM view of the same data set and highlights how nesting features can be evident in either system view.

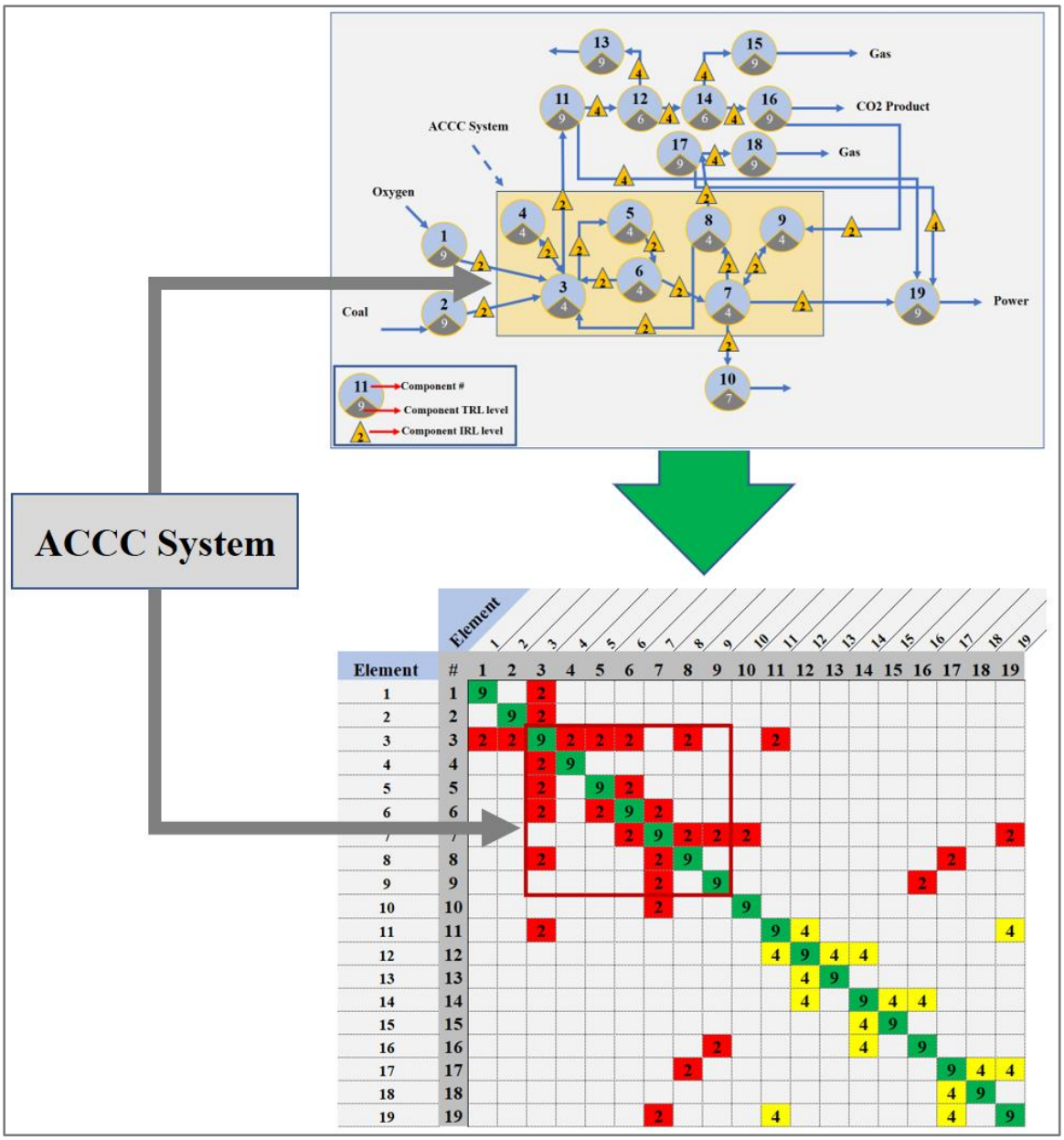
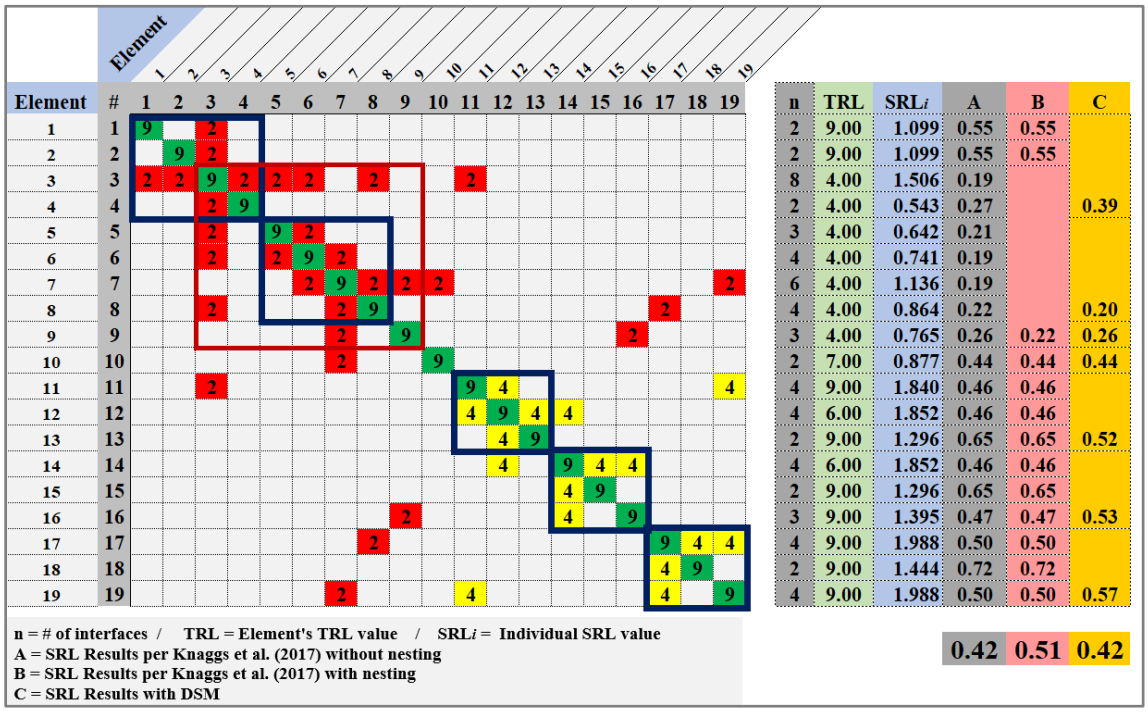


Figure 4.3: DSM of the System from Knaggs et al.

After generating the N×N DSM from the given network diagram, the system was re-evaluated with regard to redesigning it by utilizing a partitioning method to ensure that the rearranged DSM contains minimum feedback marks (Browning, 2001; Eppinger & Browning, 2012; A. Yassine & Braha, 2003; A. A. Yassine, 2004). The partitioning algorithms described in Table 2.9 was applied for this purpose. The DSM clustering analysis results revealed five potential clusters, as shown in the blue boxes in Figure 4.4.



n	TRL	SRL*i*	A	B	C
2	9.00	1.099	0.55	0.55	
2	9.00	1.099	0.55	0.55	
8	4.00	1.506	0.19		
2	4.00	0.543	0.27		0.39
3	4.00	0.642	0.21		
4	4.00	0.741	0.19		
6	4.00	1.136	0.19		
4	4.00	0.864	0.22		0.20
3	4.00	0.765	0.26	0.22	0.26
2	7.00	0.877	0.44	0.44	0.44
4	9.00	1.840	0.46	0.46	
4	6.00	1.852	0.46	0.46	
2	9.00	1.296	0.65	0.65	0.52
4	6.00	1.852	0.46	0.46	
2	9.00	1.296	0.65	0.65	
3	9.00	1.395	0.47	0.47	0.53
4	9.00	1.988	0.50	0.50	
2	9.00	1.444	0.72	0.72	
4	9.00	1.988	0.50	0.50	0.57
			0.42	0.51	0.42

Figure 4.4: Comparison results from composite SRL calculations

Table 4.1 presents the steps followed to calculate a composite SRL for both methods. Next, the detailed calculations demonstrated below was performed.

Table 4.1: Step-by-step comparison of the methodology of Knaggs et al. and the DSM_SRL methodology

Applying Nesting (Knaggs et al., 2017) Methodology	Applying DSM_SRL Methodology
Decompose the overall product or system into its subsystems and/or components.	Decompose the overall product or system into its subsystems and/or components.
Create a network diagram for the system.	
Perform an SRL calculation for the subsystem:	Partition the system (if necessary) described in Table 2.9:

• Develop a directed graph system representation to define the subsystem boundaries. • Calculate a composite SRL score for the ACCC subsystem. Translate the SRL scores into a set of ETRL score ranges by utilizing the SRL translation model*.	• Reorder the rows and columns of the DSM to group the components according to a given objective.
When conducting the nested SRA calculation, the ETRL functions in the same manner as a TRL.	Cluster the system: • The clusters should contain the majority of the interfaces.
Apply the SRL calculations of Sauser et al. to determine the element SRL contributions (Equation 5).	Apply the SRL calculations of Sauser et al. to determine the element SRL contributions (Equation 5).
	Calculate the SRL contributions of each cluster (subsystem) (Equation 6).
Calculate the composite SRL (Equation 6).	Calculate a composite SRL (Equation 6).
	Fold (hide) clusters that do not have any critical integration maturity (IRL ≥ 6).

*A complete explanation of the ETRL score and the calculations is provided in (Knaggs et al., 2017).

The subsequent matrices were generated for the TRL and IRL to demonstrate the steps involved in computing the SRL (Equations 1 and 2).

$$\left[TRL\right]_{n\times 1} = \begin{bmatrix} TRL_1 \\ TRL_2 \\ \cdots \\ TRL_n \end{bmatrix} \quad \text{(Eq. 1)}$$

$$= \begin{bmatrix} 9 & 9 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 7 & 9 & 6 & 9 & 6 & 9 & 9 & 9 & 9 & 9 \end{bmatrix}$$

$$\left[IRL\right]_{n\times n} = \begin{bmatrix} IRL_{12} & IRL_{12} & \cdots & IRL_{1n} \\ IRL_{21} & IRL_{22} & \cdots & IRL_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ IRL_{n1} & IRL_{n2} & \cdots & IRL_{nn} \end{bmatrix} \quad \text{(Eq. 2)}$$

$$=\begin{bmatrix}
9 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 9 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 2 & 9 & 2 & 2 & 2 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 0 & 9 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 0 & 2 & 9 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 2 & 9 & 2 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\
0 & 0 & 2 & 0 & 0 & 0 & 2 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 9 & 4 & 4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 9 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 9 & 4 & 4 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 9 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 4 & 0 & 9 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 4 & 4 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 9 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 9
\end{bmatrix}$$

As showed in the integration matrix, an IRL value of 0 is assigned if there is no interface between a given pair of technology elements. An IRL value of 9 on the matrix diagonal was used to represent a technology element integrated with itself. Next, the TRL and IRL values were normalized (Brian Sauser et al., 2008). Then, SRL values were calculates as follows (Equations 3 and 4):

$$[SRL] = [IRL]_{19\times19} \times [TRL]_{19\times1} = \begin{bmatrix} SRL_1 \\ SRL_2 \\ \cdots \\ SRL_{19} \end{bmatrix} \quad \text{(Eq. 4)}$$

Table 4.2 shows the calculated values for each SRL_i and element SRL.

Table 4.2: Individual SRL Values

	SRL_1	SRL_2	SRL_3	SRL_4	SRL_5	SRL_6	SRL_7	SRL_8	SRL_9	SRL_{10}
SRL_i (0, n_i)	1.099	1.099	1.506	0.543	0.642	0.741	1.136	0.864	0.765	0.877
SRL_i/n_i (0, 1)	0.55	0.55	0.19	0.27	0.21	0.19	0.19	0.22	0.26	0.44
	SRL_{11}	SRL_{12}	SRL_{13}	SRL_{14}	SRL_{15}	SRL_{16}	SRL_{17}	SRL_{18}	SRL_{19}	
SRL_i (0, n_i)	1.840	1.852	1.296	1.852	1.296	1.395	1.988	1.444	1.988	
SRL_i/n_i (0, 1)	0.46	0.46	0.65	0.46	0.65	0.47	0.50	0.72	0.50	

The corresponding SRL_i for each element i was then divided by n_i (Brian Sauser et al., 2008) (Equations 5).

$$ElementSRL = \frac{SRL_i}{n_i} \quad \text{(Eq. 5)}$$

Once the element SRL was calculated for each element as shown in Equation 5, the SRL contributions were calculated for each cluster (subsystem) using Equation 6. Equation 6 was used to calculate the composite readiness for the first cluster (subsystem):

$$SRL = \frac{\frac{SRL_1}{n_1} + \frac{SRL_2}{n_2} + \ldots + \frac{SRL_n}{n_n}}{n} \quad \text{(Eq. 6)}$$

$$= \frac{\frac{1.099}{2} + \frac{1.099}{2} + \frac{1.506}{8} + \frac{0.543}{2}}{4} = 0.39$$

The integral value of four in the denominator of Equation 6 represented the number of elements within this specific cluster. The composite SRLs for the remaining clusters were calculated in a similar manner and are shown in the last column in Fig. 4.4.

In the final calculation step, the cluster composite readiness was averaged to calculate the composite SRL readiness value of the system.

4.1. Monte-Carlo Simulation Data Collection

Using the calculation methodology discussed above, the results of the SRL model of Knaggs et al. were evaluated under the following different configurations:

- Composite SRL calculations for the entire system diagram without nesting. This configuration represented the average SRL of the full system when considered as

the entire un-nested system using the original TRL and IRL values.

- Composite SRL results per Knaggs et al. for the system diagram using a nesting approach. This configuration represented the calculated average SRL of the entire system with the subsystem treated as a nested system with TRL=4 (SRL=0.28 from the original subsystem SRL).
- Composite SRL results with the DSM method. The output of this method represented the calculated average SRL of the entire system with the recommended subsystem clustering approach implemented.

Figure 4.4 presents our results from the composite SRL calculations for the SRL results per Knaggs et al. without nesting (Column A, gray shaded area), the SRL results per Knaggs et al. with nesting (Column B, rose-shaded area), and the SRL results with the DSM (Column C, orange shaded area) (Knaggs et al., 2017). The red boxes in Fig 4.4 represent the nested Advanced Combustion and Carbon Capture (ACCC) system per Knaggs et al., and the blue boxes represent the DSM clustering analysis results (Knaggs et al., 2017).

To implement the SRL calculation methods and DSM implementation of the Knaggs et al. ACCC system, research project-specific MATLAB script files and a complex MS Excel spreadsheet were employed and evaluated the Knaggs et al. system under the following conditions:

1) The Knaggs et al. system was evaluated in both the fully nested configuration (specific elements collapsed into a collective TRL component) and fully expanded configuration (all TRL and IRL components and interactions included).
2) Under both configurations, the subsystem SRL and full system SRL values were

calculated using the standard SRL calculation methodology outlined by (Brian Sauser et al., 2008).

3) Under both configurations, the subsystem and full system DSM values were obtained by employing the basic DSM methodology outlined in Eppinger et al. and presented above (Eppinger & Browning, 2012).

4) Once the SRL and DSM methods were properly implemented in the MATLAB code, the system TRL and IRL values were assigned randomly selected TRL and IRL integer values in the range of [1, 2, … 9]. Each value was selected from a random uniform distribution of integer values ranging from 1 to 9. This model was run for 1,000 iterations to provide a broad selection of TRL and IRL values across the parameter values.

- **Full System SRL –** SRL Calculation of full system without nesting
- **Original Subsystem SRL -** SRL Calculation of nested subsystem (only nested subsystem with 9 elements)
- **Nested Full System SRL –** SRL Calculation of full system model with nested subsystem
- **DSM Results -** SRL Calculation of full system with DSM

The result of the above SRL modeling and calculation process was the generation of a set of data that allowed the system readiness attained by standard S_SRL methods to be compared with the readiness values obtained using a DSM_SRL approach. The set of dependent and independent variables and their corresponding variable ranges and categories are listed in Table 4.3.

Table 4.3: Variable definitions and data types

Variable Type	Name	Range and Value Type
Independent	Calculation Method	S_SRL Method (Categorical) DSM_SRL Method (Categorical)
	System Size (n)	[X, Y, Z…] Interval
	System Order (m)	[X, Y, Z, …] Interval
	TRL_n / IRL_m values	Uniform Integer ({1,2,…,9)
Dependent	SRL Value	[0,1] Continuous (B. J. Sauser et al., 2006)
	SRL Value	[1,2,…9] Ordinal (Marc F. Austin & York, 2017)
	Min (TRL or IRL)	[1,2,…9] Ordinal
	Max (TRL or IRL)	[1,2,…,9] Ordinal
	Med (TRL or IRL)	[1,2,…,9] Ordinal
	Range (TRL or IRL)	[0,2,…8] Ordinal

For each of the four separate system configurations noted above, a randomly generated set of uniformly distributed TRL and IRL integer values was selected according to step (4) above. Each independent model iteration produced SRL calculation results and a DSM_SRL calculation result for estimated system readiness for the system configurations under consideration. The modeling and simulation experimental design chosen was a partial variable space-filling design because of the excessively large set of permutations of system TRL and IRL values. For example, the full Knaggs et al. system includes 19 technology elements. Exhaustively covering the full permutation space of 9 TRL levels for each of these 19 elements would require $9^{19}=1.35\times10^{18}$ permutations, which is an unsuitably large dataset to generate and assess. This consideration also demonstrates the need to leverage advanced complex system modeling approaches when applying the SRL methodology to complex systems of large size and order. The next section discusses the analysis applied to the modeled data collected from the process outlined above.

Chapter 5. Data Analysis and Interpretation

The following analysis was conducted for each SRL calculation methodology (the S_SRL method and DSM_SRL method) and for the system configurations Full System SRL (full system without nesting), Original Subsystem SRL (nested subsystem only), Nested Full System SRL, and DSM Results described above section 4.1. First, descriptive statistics, such as the mean, median, and standard deviation, of the S_SRL output and DSM_SRL model output results described above were calculated using Minitab 18 Statistical Software (Minitab, 2017) . The summary results of this preliminary analysis are presented in Table 5.1.

Table 5.1:Descriptive statistics summary of the modeled system SRL results for the Knaggs et al. and DSM_SRL methods

Variable	Mean	SE Mean	StDev	Variance	Minimum	Q1	Median
Mean (Full System SRL)	0.27406	0.00109	0.03456	0.00119	0.16510	0.25095	0.27414
Mean (Original Subsystem SRL)	0.39323	0.00275	0.08709	0.00759	0.11946	0.33348	0.39009
Mean (Nested Full System SRL)	0.39671	0.00205	0.06494	0.00422	0.21898	0.35201	0.39433
DSM Results	0.27029	0.00110	0.03470	0.00120	0.16510	0.24691	0.27104

An exploratory data analysis of the SRL distribution from Table 5.1 suggests that there is a marked difference in the mean and variance of the distribution of the SRL values among the SRL calculation methods as applied to the Knaggs et al. model configurations and the DSM approach. When using the system readiness modeling approach discussed in chapter 4 above, in Table 5.1, the DSM_SRL approach produces a mean SRL value of 0.27029, which is only 0.01% lower than the mean SRL value of 0.27406 produced by the SRL for the full system configuration.

In addition to the analysis of the differences of the calculated SRL mean, Table 5.1 also demonstrates that the DSM_SRL approach produces a modeled SRL distribution with very close standard deviation and variance (0.004% and 0.008% higher than the next largest standard deviation and variance, respectively) than those produced by the S_SRL method (Brian Sauser et al., 2008; Tan, Ramirez-Marquez, et al., 2011) employed for the original system evaluated in section III. Which supports the second null hypothesis "There is no significant difference between DSM framework that combines system architecture concepts and analysis with system readiness levels (SRLs) and Sauser SRL (Full System SRL)".

Using the model results from section III and the descriptive statistical analysis listed in Table 5.1, histograms (Figure 5.1-5.4) and a boxplot (Figure 5.5) of the distribution of SRL values for both the S_SRL calculation method and DSM approach were developed. These figures were developed for the DSM method and Sauser method for the several system configurations derived from (Knaggs et al., 2017).

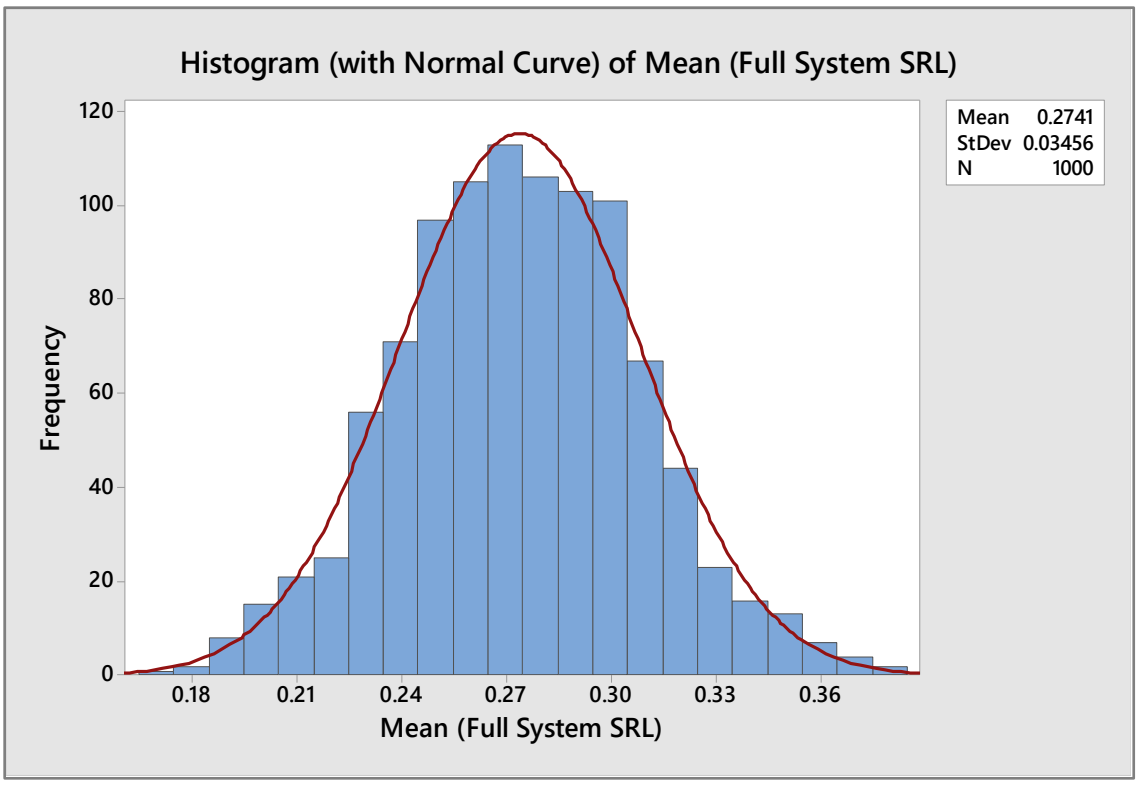


Figure 5.1: Histogram of Mean (Full System SRL) results

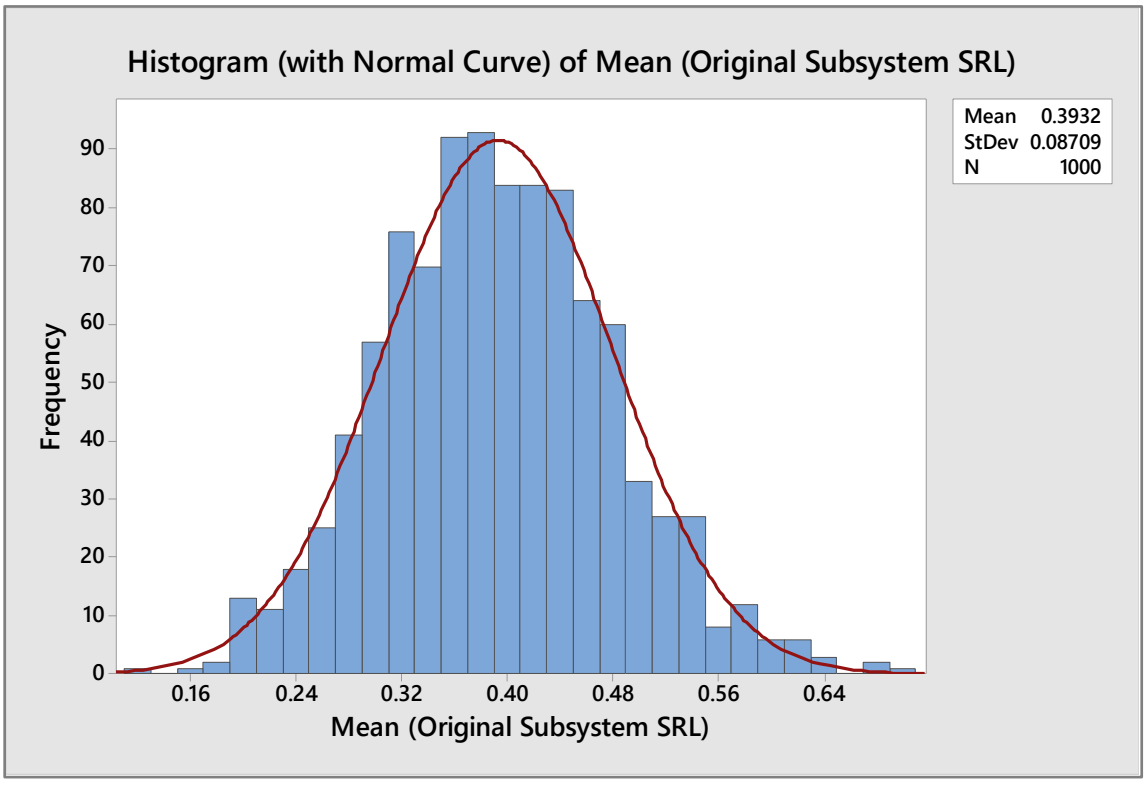


Figure 5.2: Histogram of Mean (Original Subsystem SRL) results

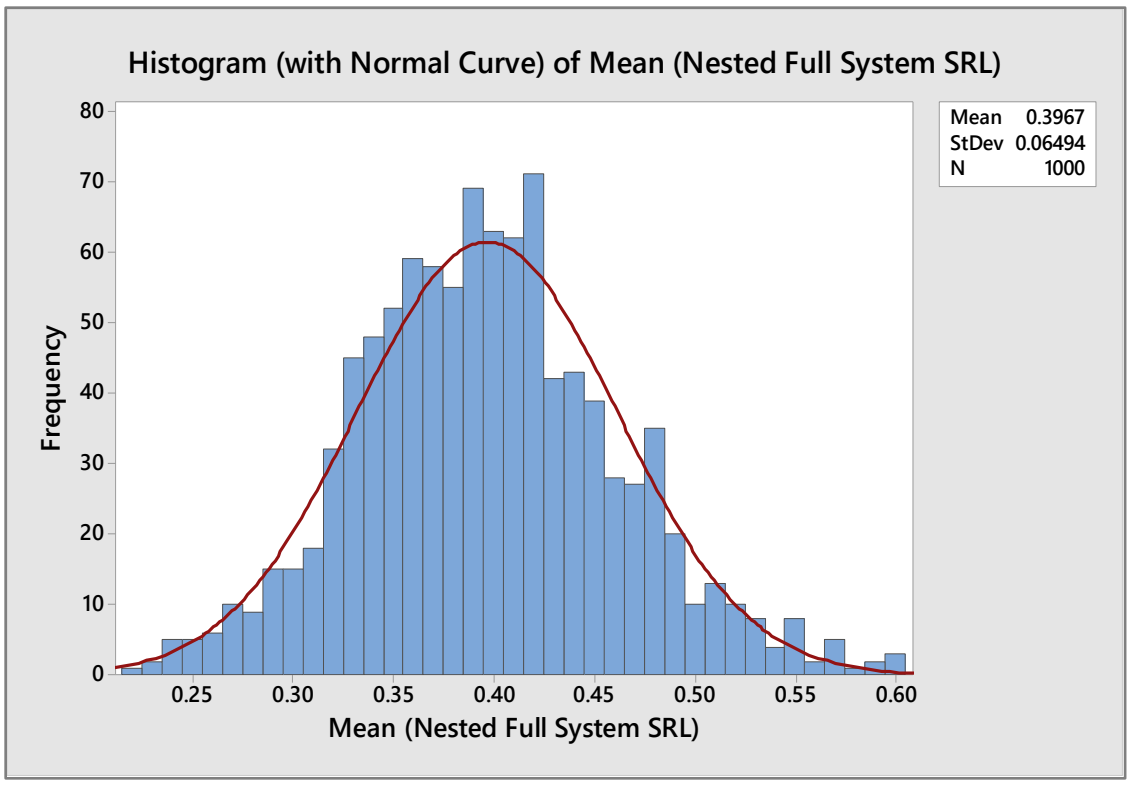


Figure 5.3: Histogram of Mean (Nested Full System SRL) results

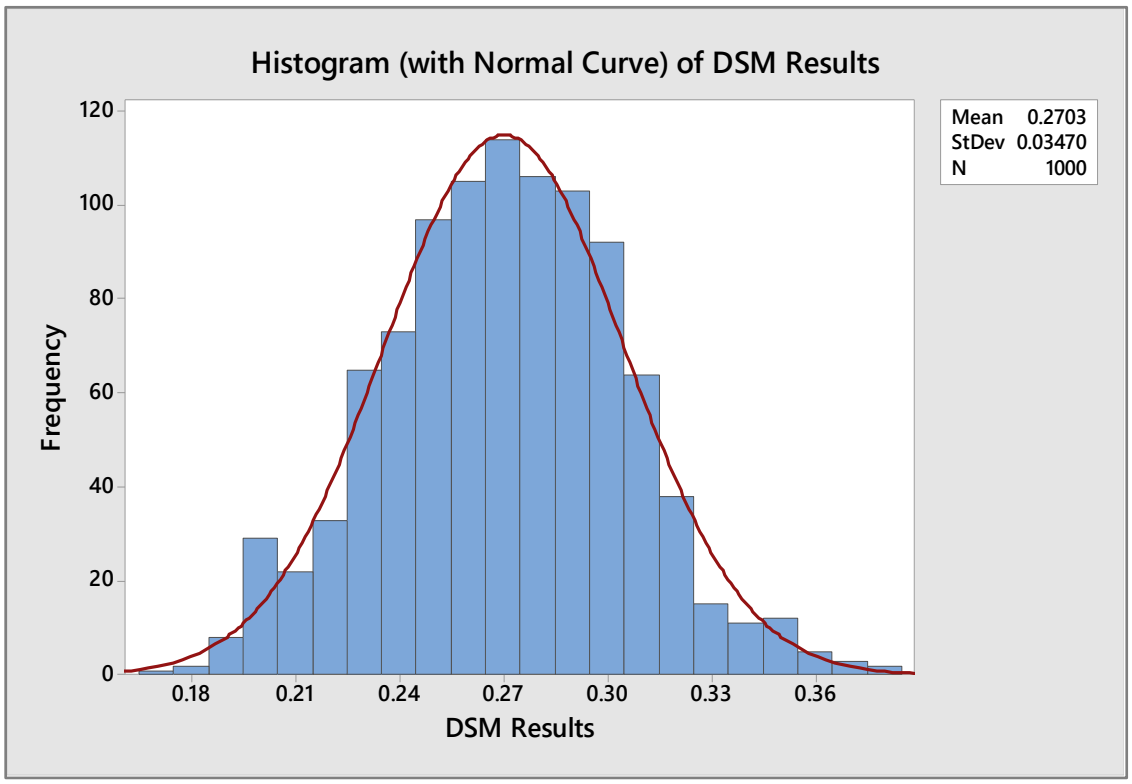


Figure 5.4: Histogram of DMS results

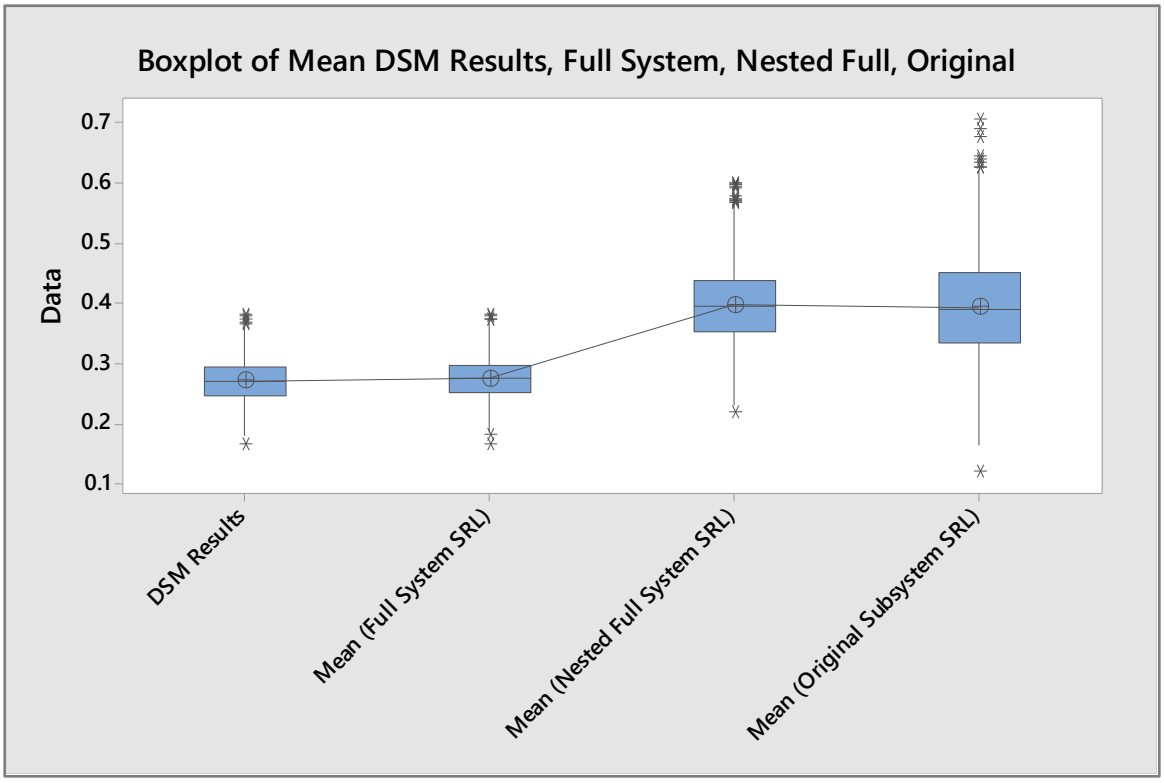


Figure 5.5: Boxplot of the SRL values for the Sauser SRL method and DSM approach for the system configurations described by Knaggs et al.

The graphical histogram and boxplot depictions (Figures 5.1-5.5) supported the study hypothesis that the DSM_SRL system readiness method consistently produces similar mean SRL value of 0.27029, which is only 0.01% lower than the mean SRL value of 0.27406 produced by the S_SRL for the full system configuration and consequently very close to system readiness results produced with the S_SRL approach for Full System SRL. This observation is evidenced by the visually lower average calculated system readiness values and the corresponding narrower standard deviation and variance of the DSM_SRL method results versus the other approaches with nesting under consideration.

Addition to the graphical histogram and boxplot depictions, Shapiro-Wilk Test of Normality applied to detect departures from normality due to skewness. The results showed that while DSM Results, Mean (Full System SRL), and Mean (Original

Subsystem SRL) populations have a normal distribution, Mean (Nested Full System SRL) sample does not have a normal distribution. Table 5.2 depicts the results of the Shapiro-Wilk Test of Normality.

Table 5.2: Shapiro-Wilk Test of Normality Summary

Variable	Shapiro-Wilk (s-w)	p-value
Mean (Full System SRL)	0.99867	0.6708
Mean (Original Subsystem SRL)	0.99703	0.05994
Mean (Nested Full System SRL)	0.99485	0.001757
DSM Results	0.99749	0.1274

Additionally, the normal probability plot graphics (Figure 5.6-5.10) generated to validate our normality findings.

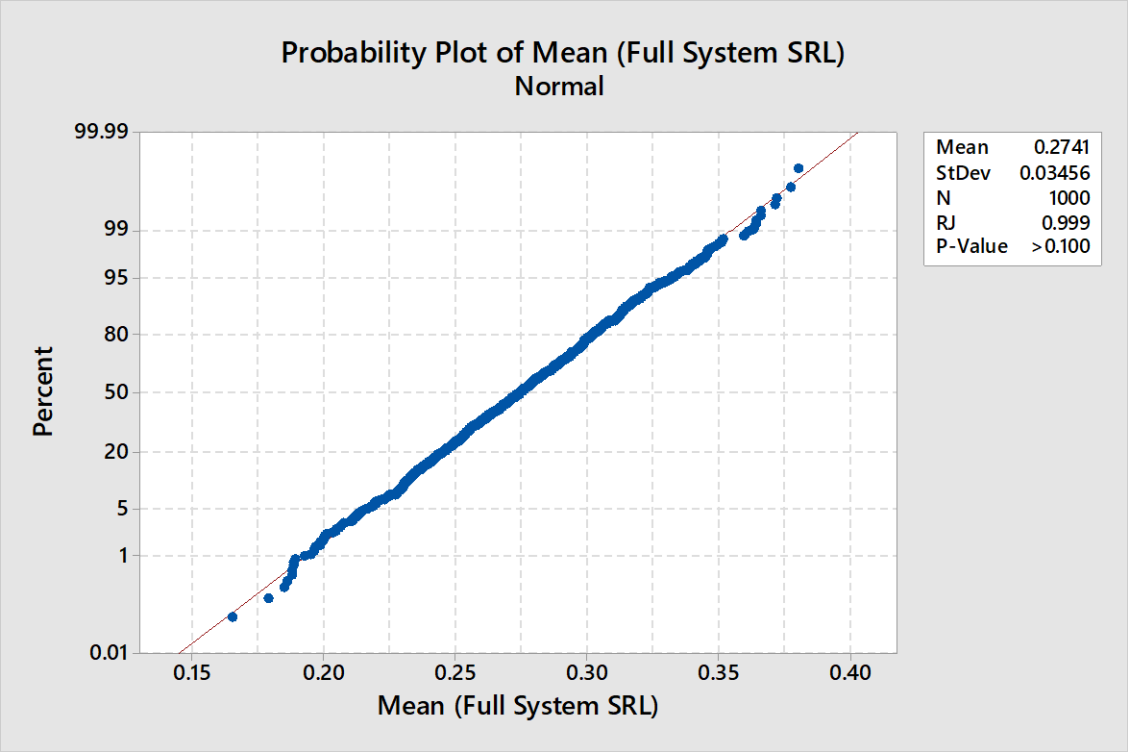


Figure 5.6: Normal Probability Plot of Full System SRL

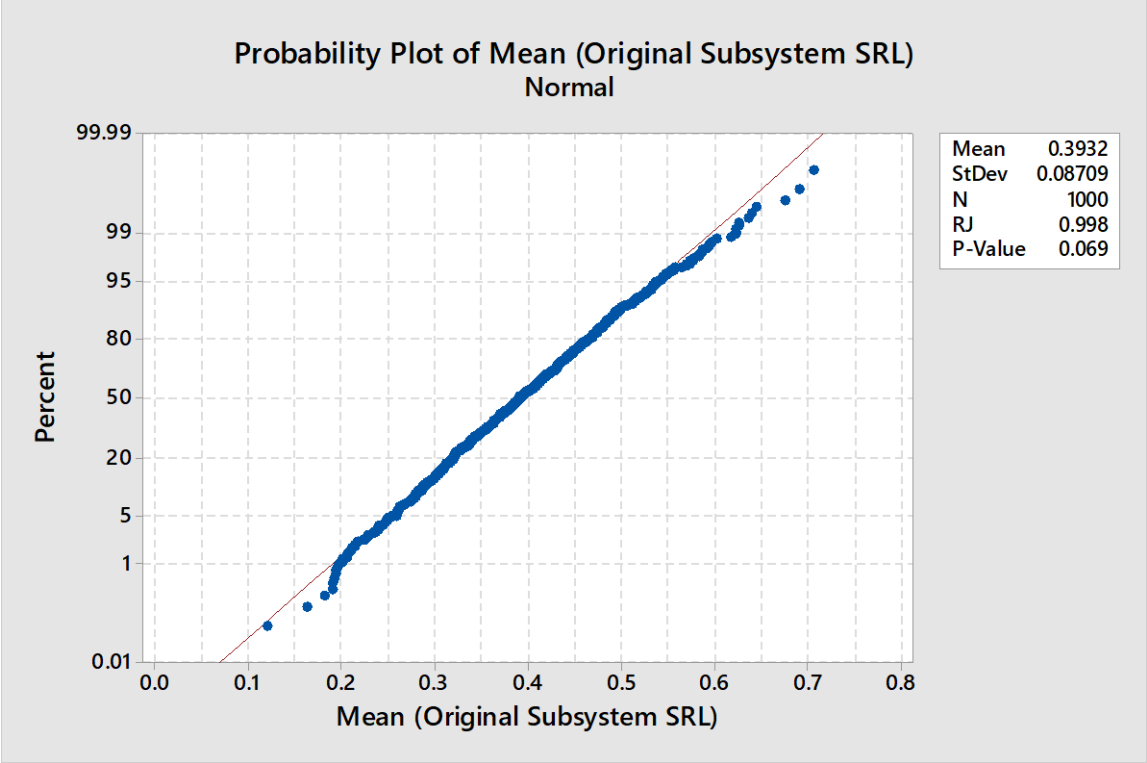


Figure 5.7: Normal Probability Plot of Original Subsystem SRL

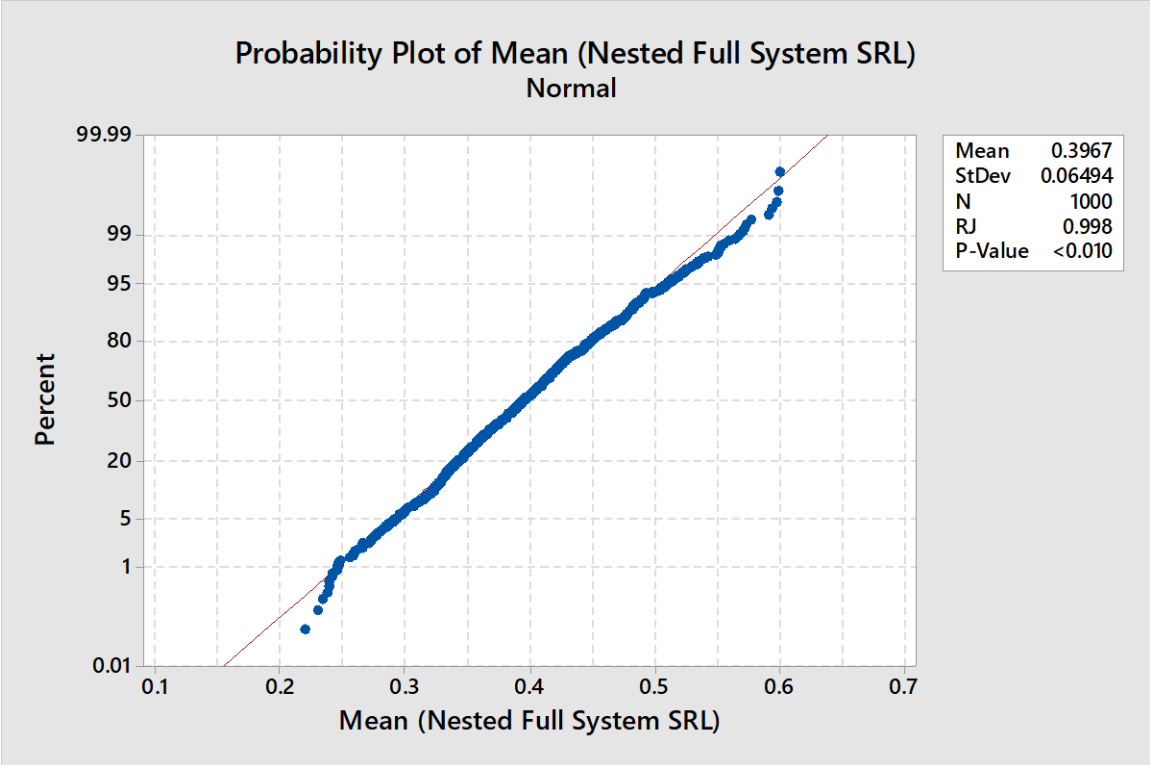


Figure 5.8: Normal Probability Plot of Nested Full System SRL

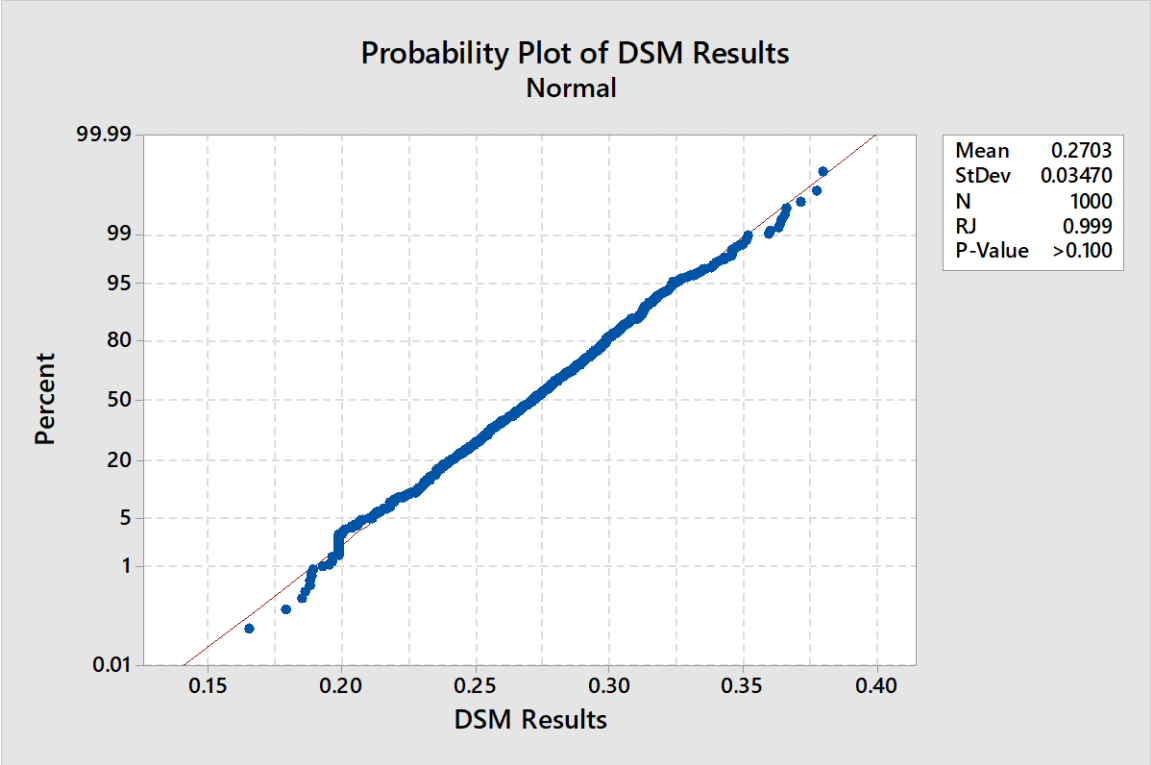


Figure 5.9: Normal Probability Plot of DSM Results

In addition to the descriptive analysis and exploratory analysis provided above, one-way analysis of variance (ANOVA) is utilized to determine whether there are any statistically significant differences between the means of Full System SRL, Original Subsystem SRL, Nested Full System SRL and DSM Results. The ANOVA test was conducted at 95% confidence levels (α=0.05) as evaluated using the different system configurations presented in (Knaggs et al., 2017). The results of One-way ANOVA: DSM Results, Mean (Full System SRL), Mean (Nested Full System SRL), Mean (Original Subsystem SRL) test is provided below.

One-way ANOVA: DSM Results, Mean (Full System SRL), ... ystem SRL)

Method

Null hypothesis	All means are equal
Alternative hypothesis	Not all means are equal
Significance level	α = 0.05

Equal variances were assumed for the analysis.

Factor Information

Factor	Levels	Values
Factor	4	DSM Results, Mean (Full System SRL), Mean (Original Subsystem SRL), Mean (Nested Full System SRL)

Analysis of Variance

Source	DF	Seq SS	Contribution	Adj SS	Adj MS	F-Value	P-Value
Factor	3	15.09	51.55%	15.09	5.03055	1416.95	0.000
Error	3996	14.19	48.45%	14.19	0.00355		
Total	3999	29.28	100.00%				

Model Summary

S	R-sq	R-sq(adj)	PRESS	R-sq(pred)
0.0595842	51.55%	51.51%	14.2153	51.45%

Means

Factor	N	Mean	StDev	95% CI
DSM Results	1000	0.27029	0.03470	(0.26660, 0.27399)
Mean (Full System SRL)	1000	0.27406	0.03456	(0.27037, 0.27776)
Mean (Original Subsystem SRL)	1000	0.39323	0.08709	(0.38954, 0.39692)
Mean (Nested Full System SRL)	1000	0.39671	0.06494	(0.39302, 0.40041)

Pooled StDev = 0.0595842

The ANOVA test demonstrated a statistically significant difference in the mean calculated SRL value between the DSM Results, Mean (Full System SRL), and Mean (Nested Full System SRL), Mean (Original Subsystem SRL) of the S_SRL calculation results. A p-value of 0.000 across all comparisons and across all confidence levels under consideration suggests that there is a statistically significant difference in the calculated system readiness level mean using the DSM approach versus the S_SRL with nesting assumptions methodology employed for both the nested and non-nested system configurations.

Although continued research remains to be completed to determine the specific quantitative causal factors underlying this significant difference of SRL means between the S_SRL and the DSM_SRL method, the presence of subsystem collapse is a likely explanation for this difference. By reorganizing the original system network diagram architecture using a collapsing subsystem approach, the DSM_SRL method effectively compresses the "white space" in the DSM matrix, thereby reducing the impact of the S_SRL calculation step in which empty cells in the adjacency matrix are added to the row calculation. The reduced impact of these additive calculations in the DSM process drives the resultant SRL value toward a more realistic value that treats the overall system as a more representative assembly of its constituent subsystems.

ANOVA test results on the SRL mean across the system configurations of Knaggs et al. support a rejection of the null hypothesis that there is no significant difference in calculated SRL mean between the S_SRL method and the DSM method. A statistically significant difference in the SRL value means was identified at the 95% confidence levels when the DSM values were compared to the Nested Full System SRL and Original Subsystem SRL methods for different system and subsystem configurations. The significantly lower SRL value means obtained from the DSM approach suggests that the DSM method reliably produces a closer SRL value and thereby represents a more conservative estimate of system readiness than the Nested Full System SRL and Original Subsystem SRL methods for a given system architecture composed of one or more defined subsystems.

Addition to, ANOVA which is the method used to test differences between two or more means, paired t-test conducted to analyze Nested Full System SRL/ DSM SRL and

Original Subsystem SRL/ DSM SRL populations which will provide resolution to the study hypothesizes.

- **Hypothesis Resolution for Hypothesis #1**

Paired T-Test and CI: DSM Results, Mean (Nested Full System SRL)

Descriptive Statistics

Sample	N	Mean	StDev	SE Mean
DSM Results	1000	0.27029	0.03470	0.00110
Mean (Nested Full System SRL)	1000	0.39671	0.06494	0.00205

Estimation for Paired Difference

Mean	StDev	SE Mean	95% CI for μ_difference
-0.12642	0.07401	0.00234	(-0.13101, -0.12183)

μ_difference: mean of (DSM Results - Mean (Nested Full System SRL))

Test

Null hypothesis	H_0: μ_difference = 0
Alternative hypothesis	H_1: μ_difference ≠ 0

T-Value	P-Value
-54.02	0.000

The paired t-test comparisons demonstrated a statistically significant difference in the mean calculated SRL value between the DSM approach is compared to the S_SRL with the nested full system configuration. A p-value of 0.000 for the pairwise comparisons and the confidence level under consideration suggests that there is a statistically significant difference in the calculated system readiness level mean using the DSM approach versus the S_SRL with nesting assumptions methodology.

- **Hypothesis Resolution for Hypothesis #2**

Paired T-Test and CI: DSM Results, Mean (Full System SRL)

Descriptive Statistics

Sample	N	Mean	StDev	SE Mean
DSM Results	1000	0.27029	0.03470	0.00110
Mean (Full System SRL)	1000	0.27406	0.03456	0.00109

Estimation for Paired Difference

Mean	StDev	SE Mean	95% CI for μ_difference
-0.00377	0.04795	0.00152	(-0.00675, -0.00080)

μ_difference: mean of (DSM Results - Mean (Full System SRL))

Test

Null hypothesis	H_0: μ_difference = 0
Alternative hypothesis	H_1: μ_difference ≠ 0

T-Value	P-Value
-2.49	0.013

The paired t-test comparisons demonstrated that there is no statistically significant difference between the means and conclude that a significant difference does exist in the mean calculated SRL value between the DSM approach is compared to the S_SRL with the full system configuration. A p-value of 0.013 for the pairwise comparisons and the confidence level under consideration suggests that there is not a statistically significant difference in the calculated system readiness level mean using the DSM approach versus the S_SRL without nesting assumptions methodology.

In addition to promoting more conservative SRAs for a given defined system comparing to Nested Full System SRL and Original Subsystem SRL, the DSM approach also offers qualitative benefits versus the legacy SRL calculation methods in the literature. Additionally, the results showed that the DSM approach offers a intuitively

comprehensible method for complex system evaluation than the SRL methods listed in section 2.4. The DSM system representation more readily defines system and subsystem boundaries in a more easily comprehensible graphical nature than the equivalent system network diagram could produce. This observation is admittedly qualitative. However, the depiction of complex systems in a DSM representation lends itself to straightforwardly defining system and subsystem boundaries. Furthermore, once the DSM method is properly configurated in its initial DSM table condition, the graphical system depiction changes produced by the application of subsystem partitioning methods can easily be tracked and monitored. These progressive changes to the DSM system depiction open additional avenues to developing alternative subsystem partitioning approaches where the system modification characteristics can be reliably monitored and quantitatively evaluated.

After completing the data analysis and interpretation phase, the next step was to apply this methodology to a real-world example to understand the level of effort required for practitioners to adopt such a method.

Chapter 6. Demonstration Study

The foregoing discussion demonstrates the application of a DSM_SRL approach to an established system in the literature and the resulting benefits of the DSM_SRL approach. Use of the Knaggs et al. system also provided a validated system example to demonstrate the SRL calculation improvements that may be realized from employing a DSM_SRL approach to a given complex system. However, the performance of this example demonstrated the need to demonstrate the application of DSM methods to a more complex and real-world system, such as a NSWS, by performing matrix operation and SRAs on a DSM representation of such a system. This example will provide insight into a real-world example that demonstrates the decision-making benefits to be gained by the systems engineering and program management community from employing a DSM approach to system readiness.

6.1. Description and Assumptions

A NSWS was selected to provide a system size that was sufficiently large to demonstrate the potential of the DSM for use on a suitably complex system. In this study two NSWS systems were evaluated. The first NSWS (NSWS 1) was a system that was completed on schedule and budget with no new technology insertions. The second NSWS (NSWS 2) that was evaluated had multiple immature critical technology insertions that produced schedule delays and cost growth (GAO, 2013, 2017). According to the GAO report cost growth and schedule delay defined as (GAO, 2018);

> *"cost growth is measured as the difference between the initial cost estimate reflected in the Navy's budget request documents prior to ship construction (year in which the Navy requested authorization for the ship from Congress) and the*

cost estimate reflected in the Navy's fiscal year budget request documents or the actual cost."

"Schedule delays are measured as the difference between the obligation work limiting date (OWLD) reflected in the Navy's budget request documents prior to ship construction (year in which the Navy requested authorization for the ship from Congress) and the OWLD reflected in the Navy's fiscal year 2018 budget request documents or the actual date. In this report, we refer to OWLD as the date the ship is provided to the fleet. This graphic only includes lead ships that report obligation work limiting dates or an equivalent milestone."

The key assumptions applied to the DSM evaluation method for these two example systems are as follows:

- If two elements have the same interface in the legacy platform, then the TRL and IRL values can be assigned a value of 8 (Daniel Ross Katz, 2014; Daniel R. Katz, Sarkani, Mazzuchi, & Conrow, 2015).
- The sub elements with the lowest TRL value determine the higher-level element overall TRL value (Marc F. Austin & York, 2017; Ross, 2016).
- Due to the principles of mutual dependency, interfacing elements could be at a different TRL, but their IRLs must be assigned the same and equal values as the least ready component (Brian Sauser et al., 2008).
- IRL values cannot exceed the TRL values (Ross, 2016).

6.2. Process

The first step in this example evaluation was to determine how many different NSWSs defined the systems to demonstrate the decision-making improvements offered

by the DSM approach. As described above, two NSWSs were identified for this study objectives.

The next few steps produced the initial DSM. A warfare system interface diagram (WSID) were attained for specific ship platforms to construct feasible architectures. The WSID depicts the detect-control-engage systems that are considered part of the warfare system (Michael & Byers, 2013). Figure 6.1 presents an example of a WSID (Beiderbecke, 2013). In Figure 6.1, each box represents an element, and each arrow or line represents an interface between elements.

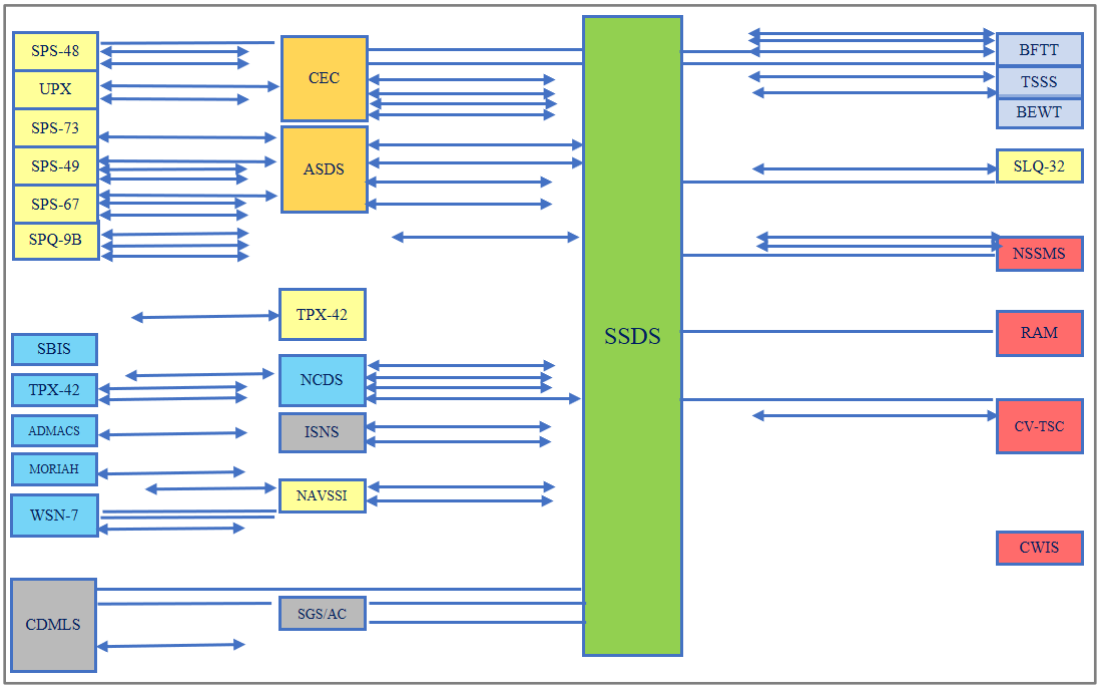


Figure 6.1: Example of a warfare system interface diagram adopted from (Beiderbecke, 2013)

Two platform-specific WSIDs were then decomposed into key combat system elements. The elements were entered into a DSM, and the interfaces between the elements were established using the abovementioned DSM methodologies. At this point, the DSM appeared similar to Figure 6.2, with 'X' marks signifying that an interface existed between the elements.

Component	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
	Component	Detect 1	Detect 2	Detect 3	Control 1	Engagement 1	Control 2	Control 3	Control 4	Control 5	Control 6	Control 7	Control 8	Detect 8	Control 9	Control 10	Detect 4	Detect 5	Control 11	Detect 6	Training 1	Training 2	Detect 7	Engagement 2	Engagement 3	Control 12	Control 13
Detect 1	1	X		X	X						X	X	X	X			X		X		X		X	X	X		
Detect 2	2		X								X										X	X	X				
Detect 3	3	X		X				X				X															
Control 1	4	X			X						X	X	X									X					
Engagement 1	5					X						X													X		
Control 2	6						X							X	X		X										
Control 3	7							X				X							X	X			X				
Control 4	8								X							X											
Control 5	9									X				X	X	X											
Control 6	10	X	X		X						X							X	X								
Control 7	11	X		X	X	X		X				X		X					X		X						
Control 8	12	X			X								X	X					X			X					
Detect 8	13	X					X			X		X	X	X	X									X			
Control 9	14						X			X				X	X		X		X					X			
Control 10	15								X						X	X	X									X	
Detect 4	16	X										X		X		X	X										
Detect 5	17										X							X	X								
Control 11	18	X						X			X	X	X		X			X	X		X		X	X	X	X	X
Detect 6	19							X												X			X				
Training 1	20	X	X									X					X		X		X	X					
Training 2	21		X		X								X								X	X		X			X
Detect 7	22	X	X					X											X	X			X				
Engagement 2	23	X												X	X				X			X		X			
Engagement 3	24	X																	X						X		
Control 12	25															X			X							X	
Control 13	26																		X			X				X	X

Figure 6.2: Navy ship combat system DSM with interfaces between the elements

TRL and IRL values were then assigned to each element and interfaces using the four assumptions mentioned earlier in this section. For NSWS 1, TRL and IRL values of 8 were assigned to each element because of the program's success history (Daniel R. Katz et al., 2015). For NSWS 2, if the element is the same on both platforms, then TRL and IRL values of 8 were again assigned to each element to demonstrate that these elements had achieved a proven maturity demonstration on the NSWS 2 (Daniel Ross Katz, 2014; Daniel R. Katz et al., 2015). TRL and IRL values for the CTEs were obtained from multiple GAO reports (GAO, 2013, 2017). Figure 6.3 shows both platforms with the TRL and IRL values before partitioning.

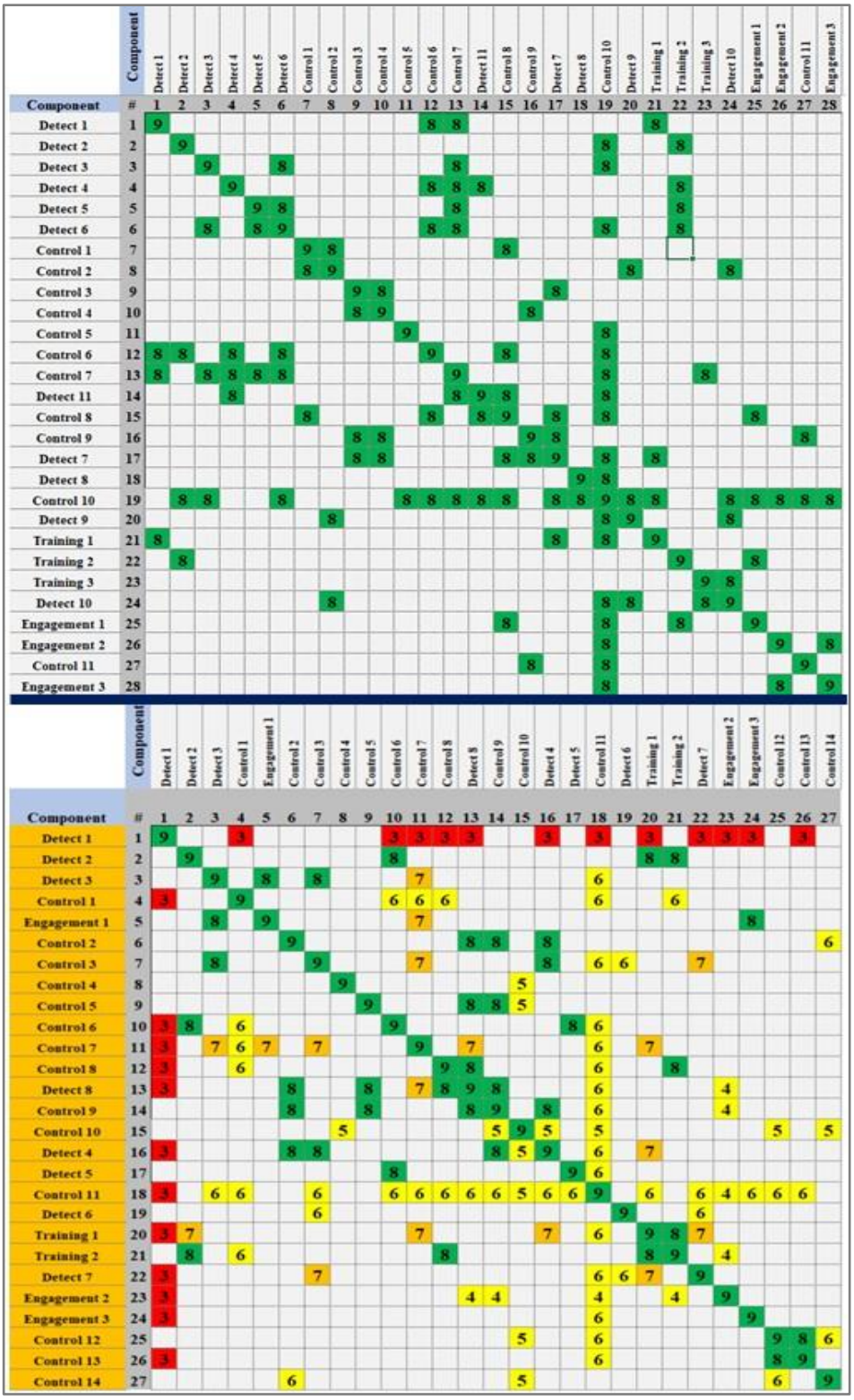


Figure 6.3: NSWS 1 (left) and NSWS 2 (right) NSWSs with TRL and IRL values before partitioning.

Following the identification of the TRL and IRL values, the next step was the application of the DSM partitioning and clustering analysis. The DSM partitioning analysis was applied to create interconnected subsets of the elements. Once partitioning was completed, clustering analysis was applied to both platforms. Figure 6.4 shows the platforms with TRL and IRL values after partitioning and clustering. The element sequence has changed to increase potential grouping of the subsystems. Identified concentrated groups of similar relationships (clusters) are denoted by blue boxes.

The next step was to calculate the element SRL contribution values by applying Equation 5 (Brian Sauser et al., 2008). In our model, the element SRL contribution values are in column A (dark gray shaded, Figure 6.4). Once the element SRL contributions values were calculated, Equation 6 was utilized to calculate each cluster's (subsystem) SRL contributions. The cluster SRL contribution values can be found in column B (orange shaded, Figure 6.4). Then, the composite SRL values for both platforms (bottom right corner) were calculated.

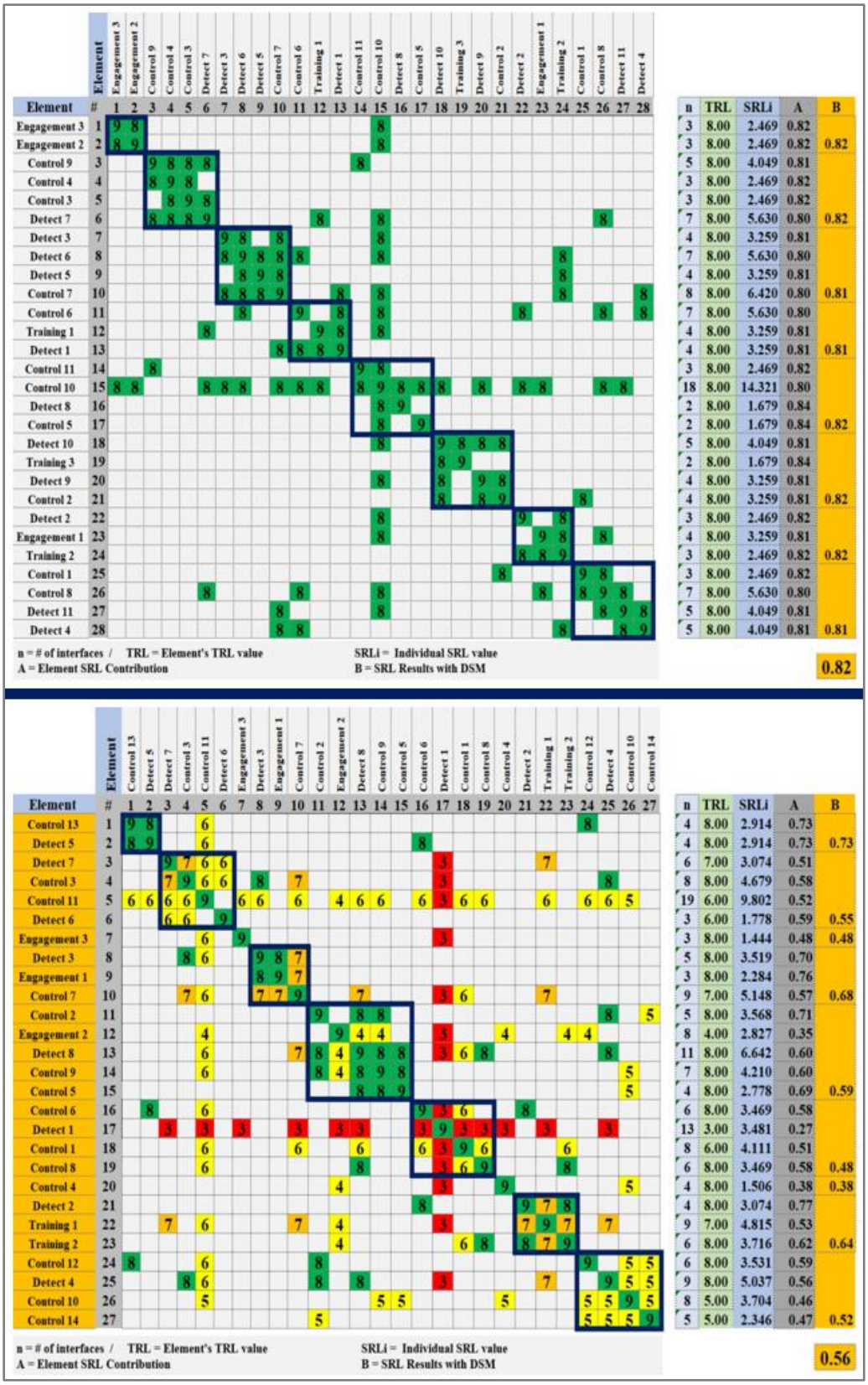


Figure 6.4: NSWS 1 (Top) and NSWS 2 (Bottom) NSWSs with TRL and IRL values after partitioning and clustering.

As noted above, the DSM provides a compact, easily scalable, and user-friendly view of the system architecture. Furthermore, the DSM view can be improved by collapsing the DSM clusters, which are coupled elements, into a single row. The next step was to collapse possible DSM clusters that do not have any low TRL values (TRL≤5) or critical integration maturity values (IRL ≤ 5). By hiding high TRL and IRL components and integrations, a user can more effectively identify high-risk elements to allow decision makers to focus primarily on these high-risk elements and their potential interfaces with other elements. For example, for a NSWS 1, the system does not have any low TRL values (TRL≤5) or critical integration maturity values (IRL ≤ 5), and therefore, the entire combat system can be collapsed into one line. However, in the delayed NSWS system, the DSM clusters cannot be collapsed because of multiple low TRL values (TRL≤5) and/or critical integration maturity values (IRL ≤ 5). This feature of collapsing low-risk subsystem clusters becomes useful when decision makers must consider large systems with multiple maturity levels. This feature will also allow users to determine the readiness of specific subsystem clusters and to track the readiness of subsystems over time to determine the optimal resource allocation needs.

The calculated composite SRL value (Equation 6) of 0.82 for the NSWS 1 indicates that the system should be in the system production and deployment phase, i.e., the system achieved an operational capability that satisfies the mission needs (see Figure 6.5).

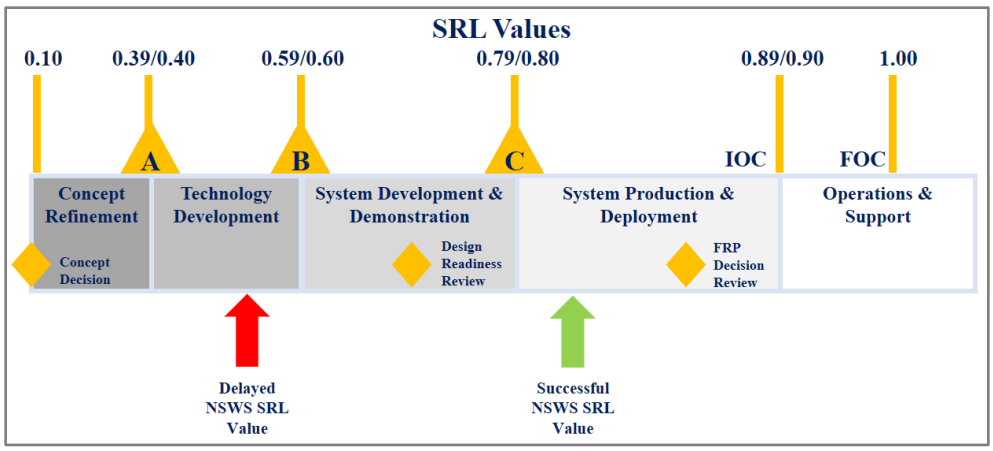


Figure 6.5: SRL mapping to defense acquisition (Brian Sauser et al., 2008).

The calculated composite SRL value of 0.56 for the delayed NSWS indicates that the system should be in the technology demonstration phase and working toward demonstrating that the required system performance and constraints will be defined, the baseline will be allocated, and the engineering team is still working on high-risk system component technology development.

In the subsequent system acquisition phase, the system development and demonstration phase, the system should validate the element integrability and threshold capability at the operational performance level using operational interfaces. According to Sauser et al., the following activities should occur during the system development and demonstration phase: “develop a system or capability increment; reduce integration and manufacturing risk; ensure operational supportability; reduce the logistics footprint; implement human systems integration; design for producibility; ensure affordability and protection of critical program information; and demonstrate system integration, interoperability, safety, and utility” (Brian Sauser et al., 2008).

The analysis of the NSWS system architectures and DSM evaluation results above suggests that comparable average SRL values would be produced by the DSM approach

and S_SRL methods described in Table 5.1 and Figures 5.1-5.6 for the NSWS systems. While the NSWS systems are larger and more complex in terms of system size (number of elements) and order (number of element interfaces), the fundamental architectural integrations do not represent a system structure that in the opinion would be produced significantly different SRL distribution results than those depicted in section III. Thus, the DSM analysis of the NSWS systems demonstrates a lower average calculated SRL value than the S_SRL methods and provides additional benefits of a graphical depiction of system boundaries and easier tracking of subsystem partitioning methods.

Chapter 7. Conclusions and Future Work

This dissertation discussed an application of the DSM for assessing the maturity of complex systems. A system example from the literature was evaluated using both a S_SRL with nesting calculation methodology and the proposed DSM SRA approach. The results from the system readiness cross-comparison model suggested that the DSM approach provides a more conservative solution for assessing system readiness than the S_SRL with nesting calculation method. In addition, the DSM method was applied to a real-world NSWS systems and demonstrated a more flexible approach to handling SRAs of nested system design architectures with complex subsystem architectures. A framework to demonstrate how SRL calculation methods can be incorporated into the DSM tool to improve system readiness decision assessments was also presented. Through real-world examples, this research demonstrated that a DSM framework facilitates the decision-making process for the systems engineer or program manager by providing an improved evidence-based maturity assessment for complex systems. DSM can be easily adopted by program managers and systems engineers to evaluate the readiness of complex systems. The readily identifiable definition of subsystem boundaries without the use of a confusing network system diagram allows for the effective tracking and monitoring of subsystem readiness metrics over time as the system progresses through the acquisition process. The DSM approach provides an effective tool for the system engineering and program management communities for the active tracking of system readiness and maturity through the development and acquisition lifecycle.

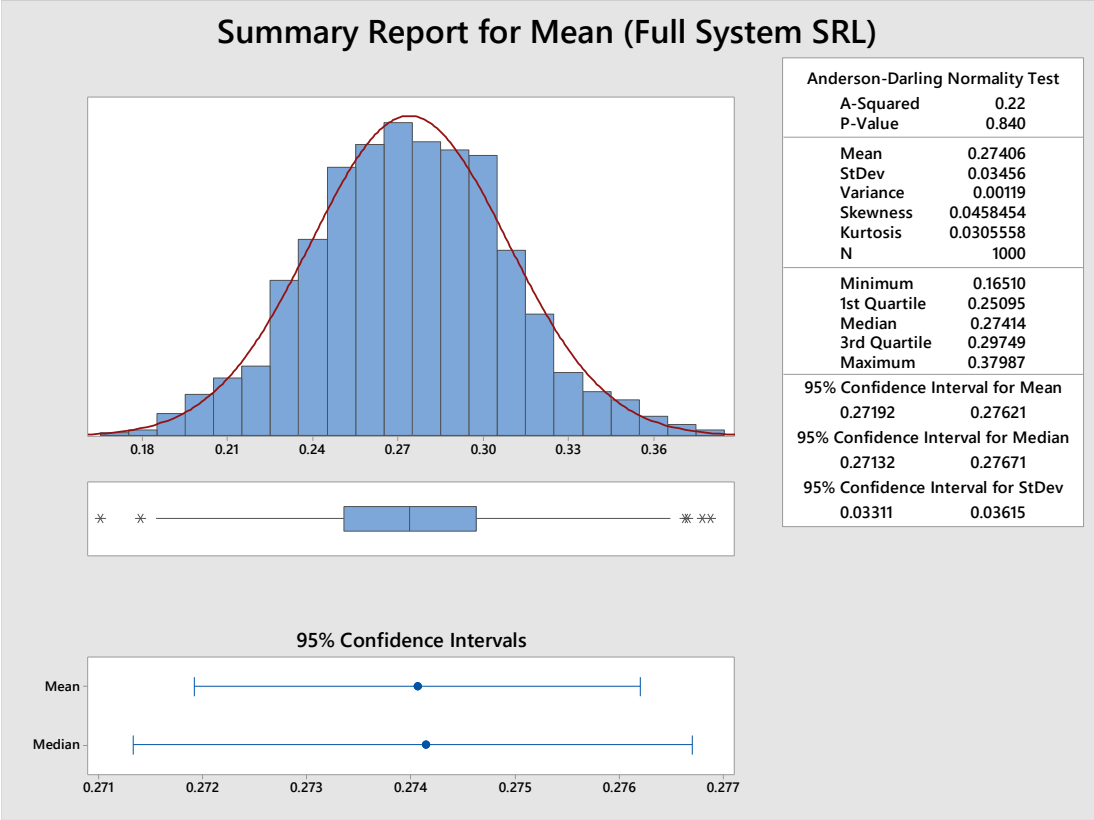
Continued research on the application of DSM methods to SRAs should be pursued in following areas. A comprehensive quantitative evaluation of system readiness results for

systems of larger size, order, and complexity than the system in Knaggs et al. should be produced. Such a study should collectively evaluate the stability and integrity of all the SRL calculation methods in the literature and compare these results to the DSM approach identified in this research.

The current standard system readiness metric is the SRA developed by the ISRACOI. The SRA method employs an adjacency matrix-based method combined with ordinal data inputs to assess the relative readiness of a system using a graphical system network model implementation. A study which evaluates the SRA model using principles from complex network analysis and graph theory to evaluate selected properties of the SRA method and propose suitable extensions should be produced. Specific properties of SRA should be evaluated using vertex and edge degree statistics, graph connectivity measures, measures of centrality such as vertex centrality and closeness, as well as node clustering coefficients. Comparisons of the system network structure used in the SRA method and several proposed network analysis methods should be offered to improve the accuracy and extension of SRA to more complex system model implementations.

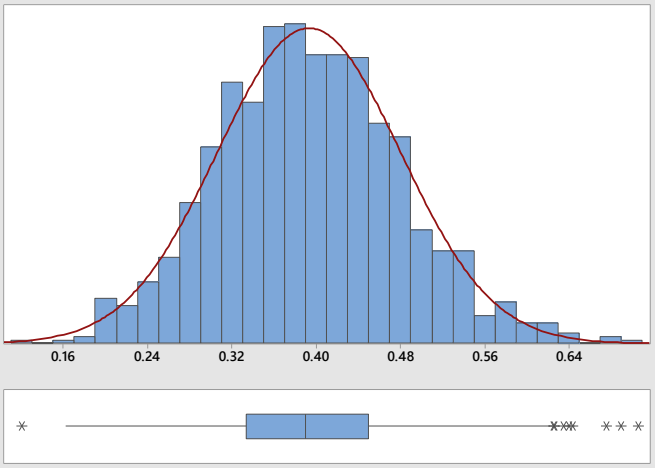
Additionally, the fundamental principles of information utility in the framework of system performance metrics to the development and establishment of the SRA should be investigated. In particular, following researches should seek to verify that the SRA model satisfies standard industry best practices for technical performance metric best practices. Standard information utility measurement principles should be employed to assess the information content of SRA and to explore areas of improving SRA within the information utility domain.

Appendix A: Summary Report - The SRL Values for the Sauser SRL Method and DSM Approach for the System Configurations

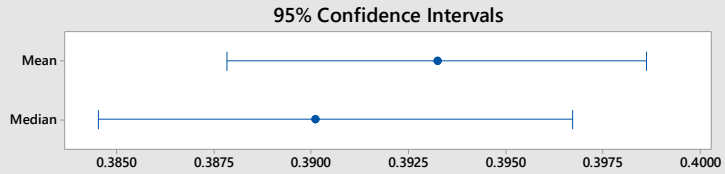


Anderson-Darling Normality Test	
A-Squared	0.22
P-Value	0.840
Mean	0.27406
StDev	0.03456
Variance	0.00119
Skewness	0.0458454
Kurtosis	0.0305558
N	1000
Minimum	0.16510
1st Quartile	0.25095
Median	0.27414
3rd Quartile	0.29749
Maximum	0.37987
95% Confidence Interval for Mean	
0.27192	0.27621
95% Confidence Interval for Median	
0.27132	0.27671
95% Confidence Interval for StDev	
0.03311	0.03615

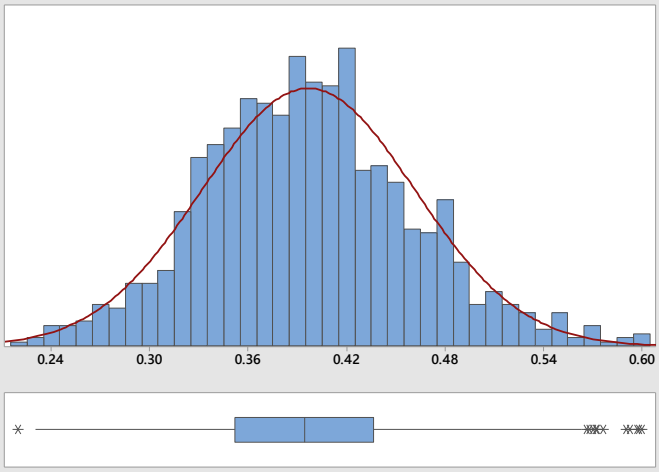
Summary Report for Mean (Original Subsystem SRL)



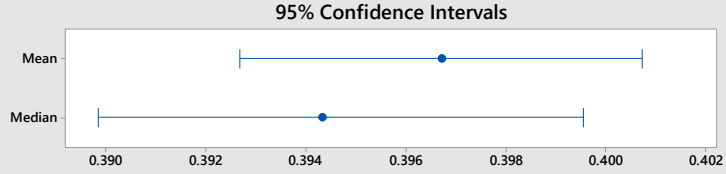
Anderson-Darling Normality Test	
A-Squared	0.48
P-Value	0.238
Mean	0.39323
StDev	0.08709
Variance	0.00759
Skewness	0.201698
Kurtosis	0.153476
N	1000
Minimum	0.11946
1st Quartile	0.33348
Median	0.39009
3rd Quartile	0.44988
Maximum	0.70547
95% Confidence Interval for Mean	
0.38783	0.39864
95% Confidence Interval for Median	
0.38450	0.39672
95% Confidence Interval for StDev	
0.08344	0.09109

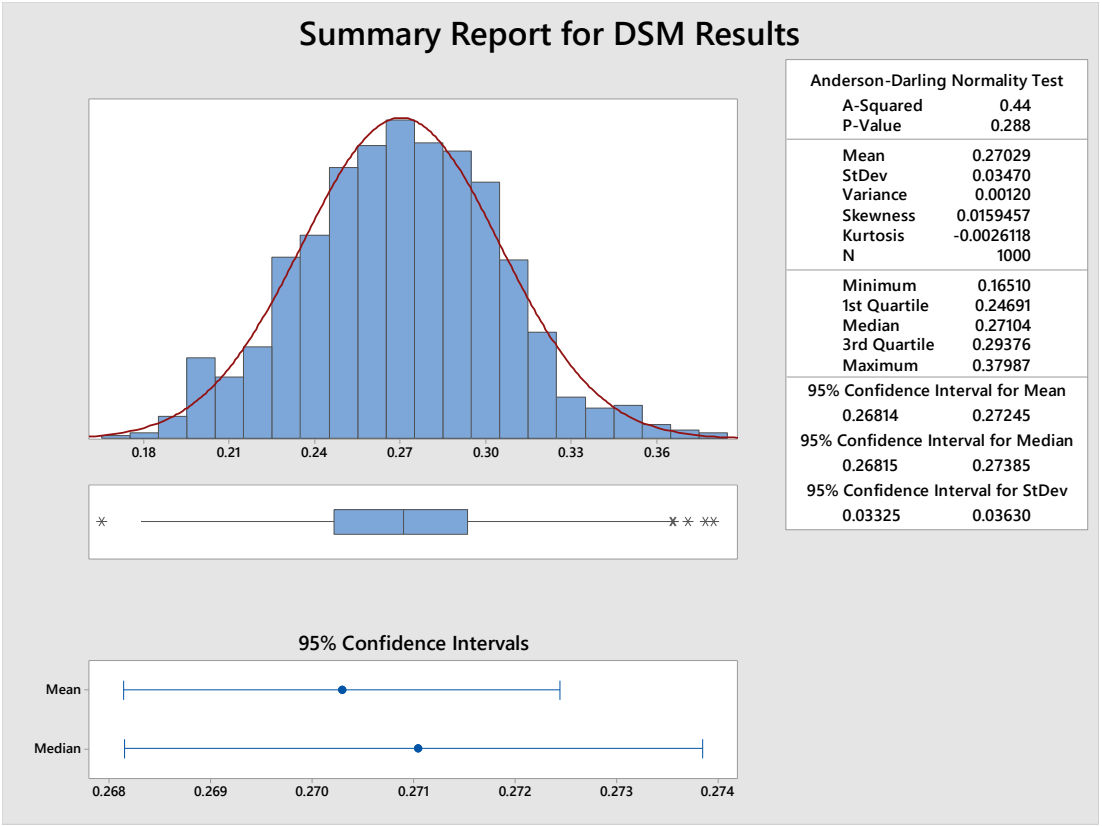


Summary Report for Mean (Nested Full System SRL)

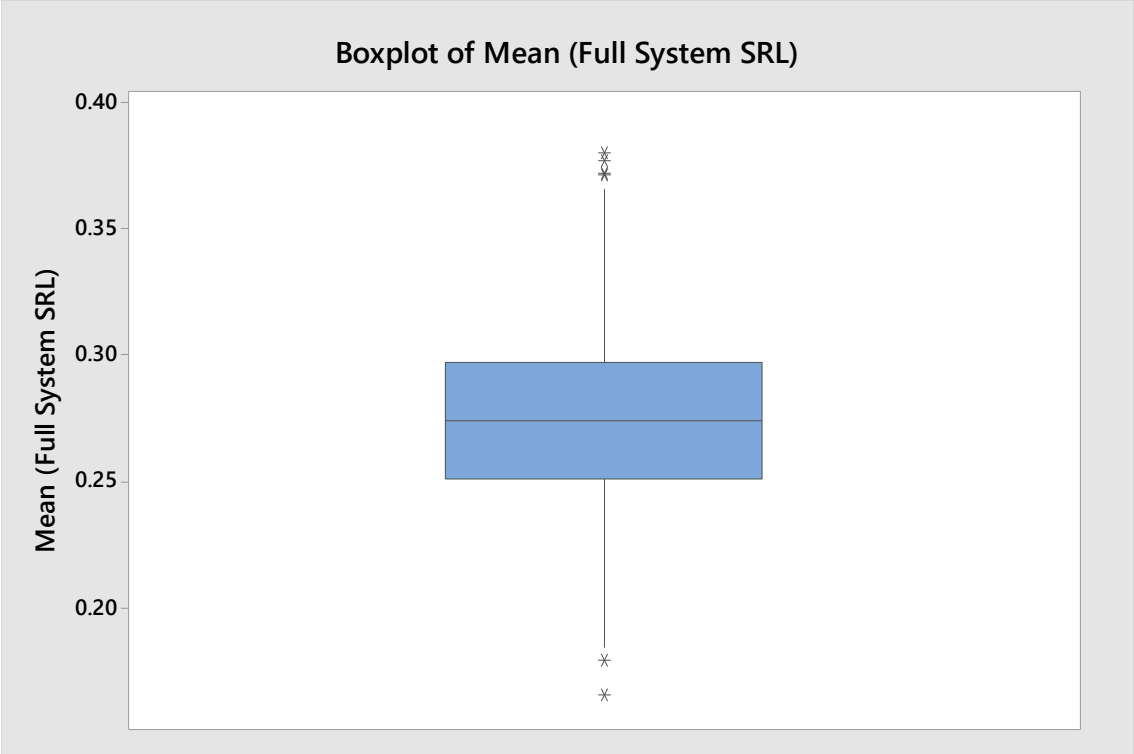


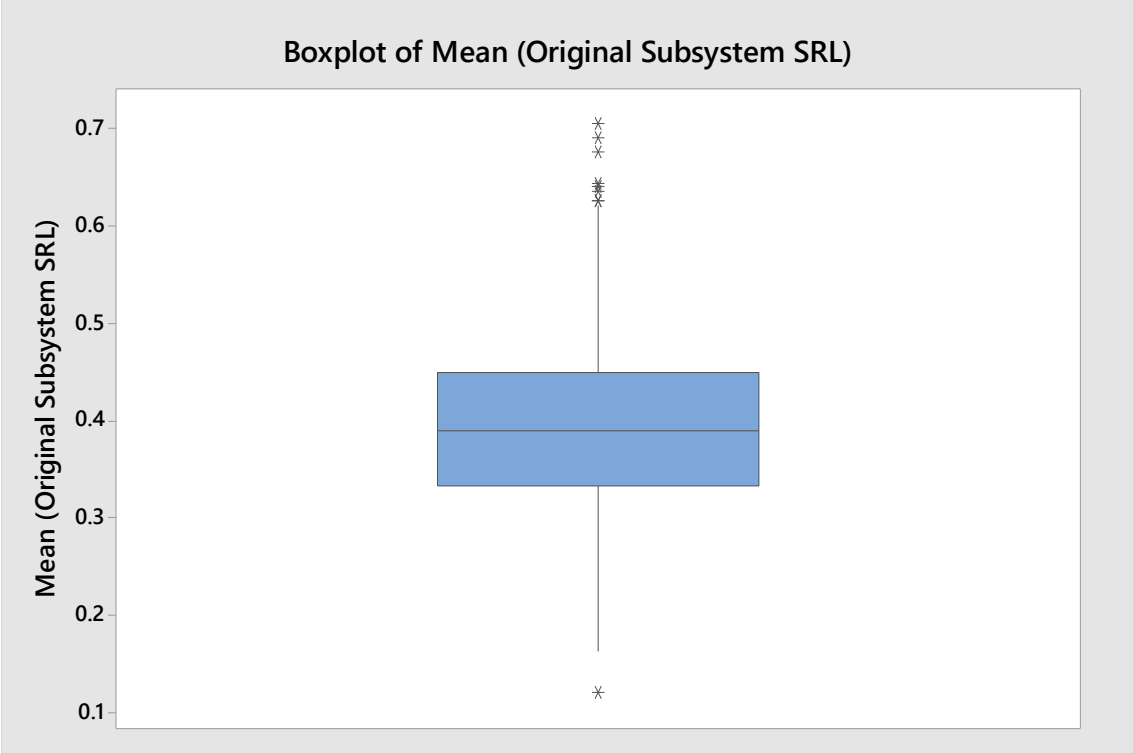
Anderson-Darling Normality Test	
A-Squared	1.18
P-Value	<0.005
Mean	0.39671
StDev	0.06494
Variance	0.00422
Skewness	0.234299
Kurtosis	0.171222
N	1000
Minimum	0.21898
1st Quartile	0.35201
Median	0.39433
3rd Quartile	0.43667
Maximum	0.59968
95% Confidence Interval for Mean	
0.39268	0.40074
95% Confidence Interval for Median	
0.38984	0.39955
95% Confidence Interval for StDev	
0.06221	0.06791

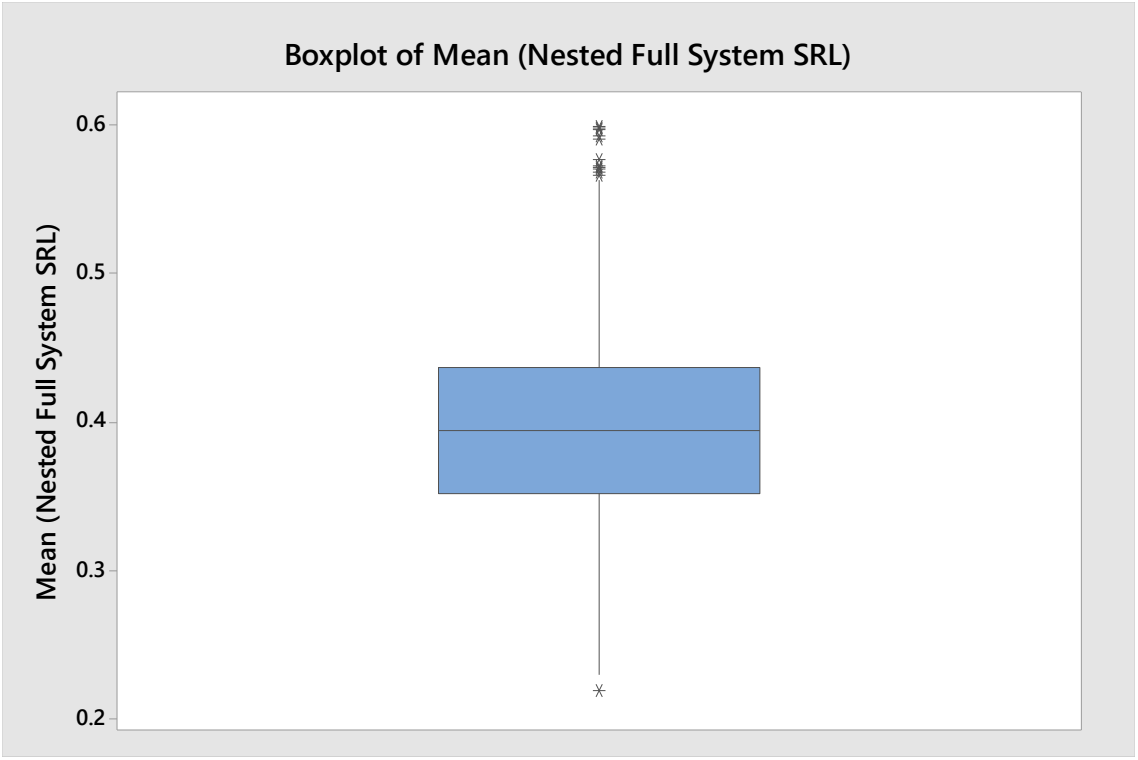


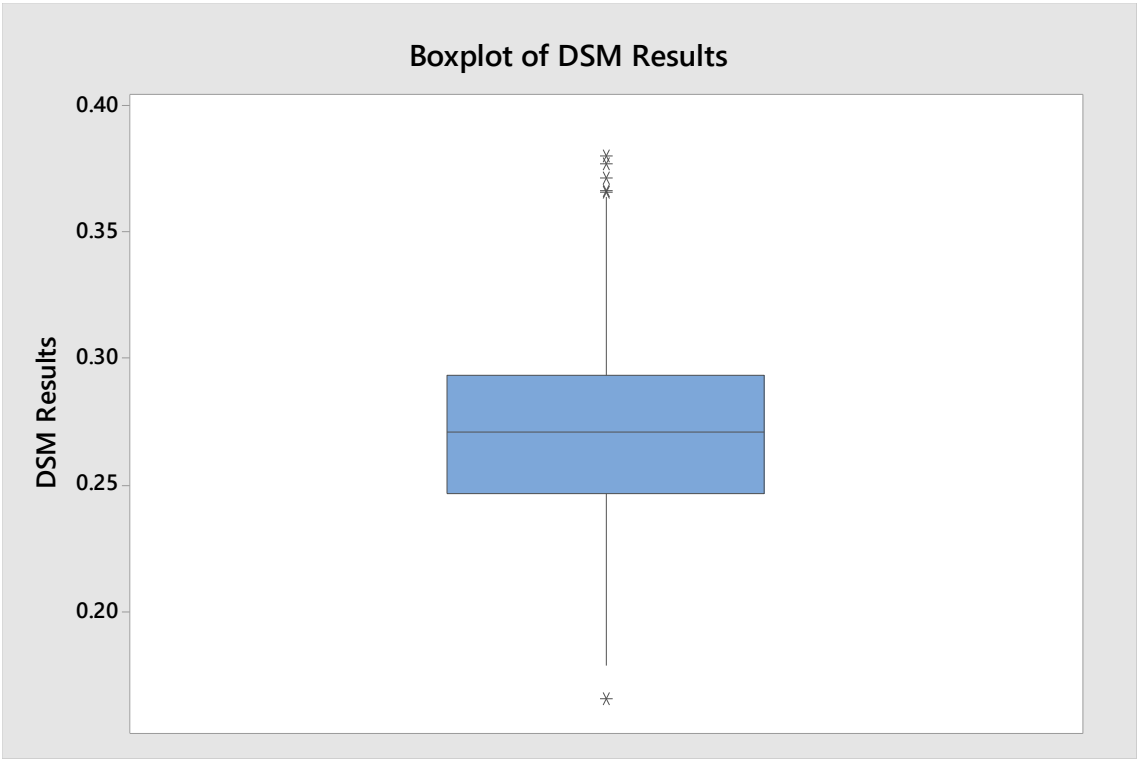

Summary Report for DSM Results
Anderson-Darling Normality Test
A-Squared 0.44
P-Value 0.288
Mean 0.27029
StDev 0.03470
Variance 0.00120
Skewness 0.0159457
Kurtosis -0.0026118
N 1000
Minimum 0.16510
1st Quartile 0.24691
Median 0.27104
3rd Quartile 0.29376
Maximum 0.37987
95% Confidence Interval for Mean
0.26814 0.27245
95% Confidence Interval for Median
0.26815 0.27385
95% Confidence Interval for StDev
0.03325 0.03630
0.18 0.21 0.24 0.27 0.30 0.33 0.36
95% Confidence Intervals
Mean
Median
0.268 0.269 0.270 0.271 0.272 0.273 0.274

Appendix B: Boxplot of the SRL Values for the Sauser SRL Method and DSM Approach for the System Configurations

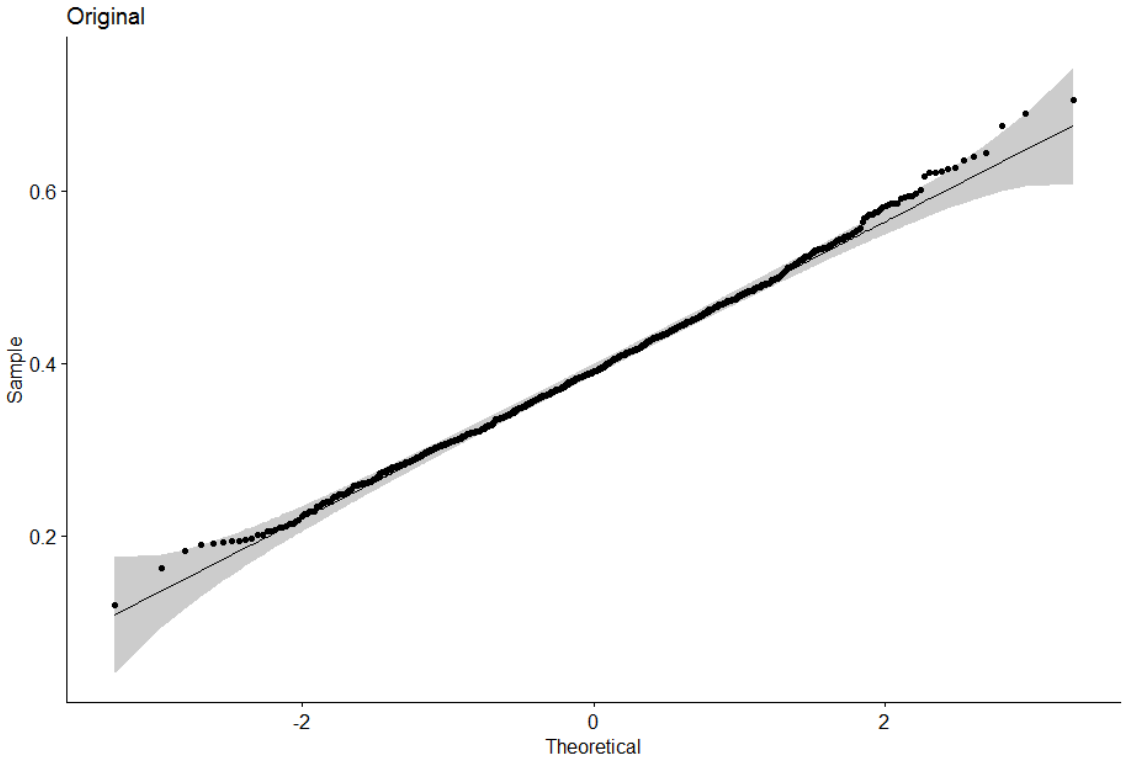


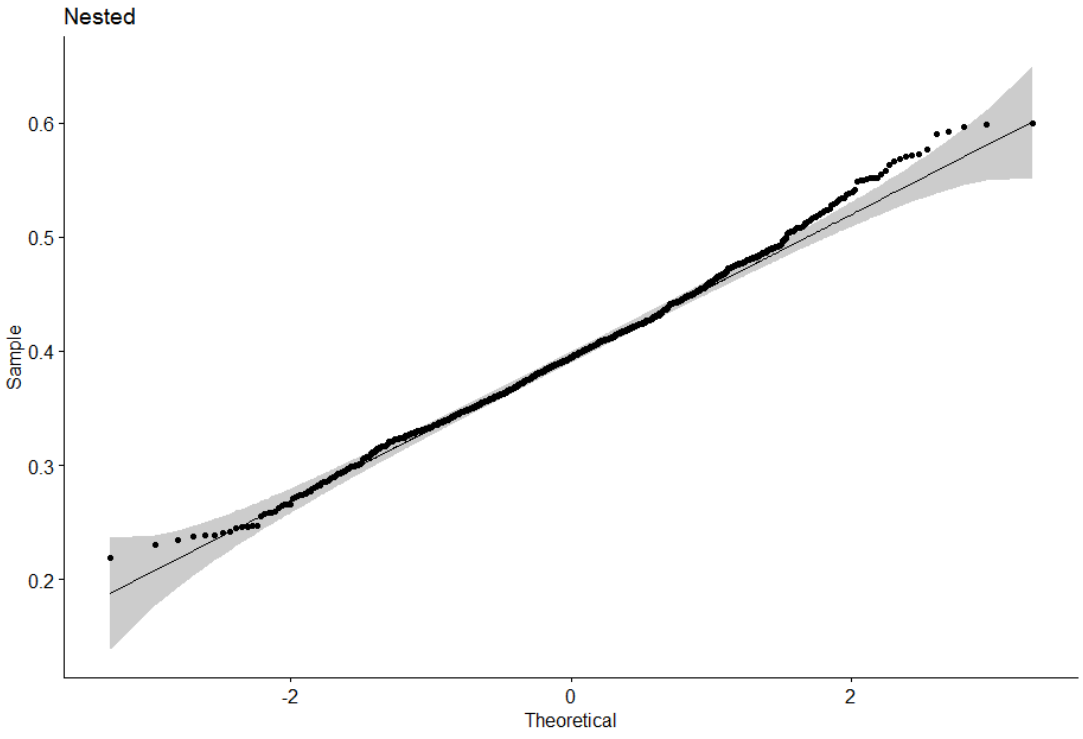


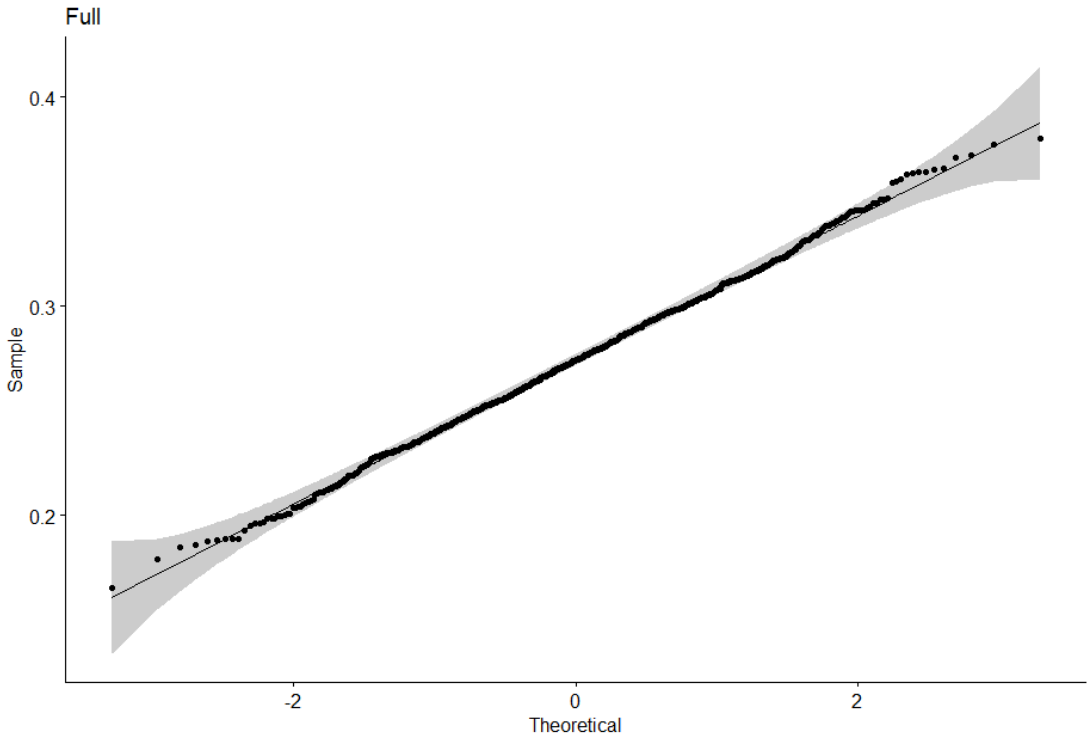

Boxplot of Mean (Nested Full System SRL)
Mean (Nested Full System SRL)
0.6
0.5
0.4
0.3
0.2

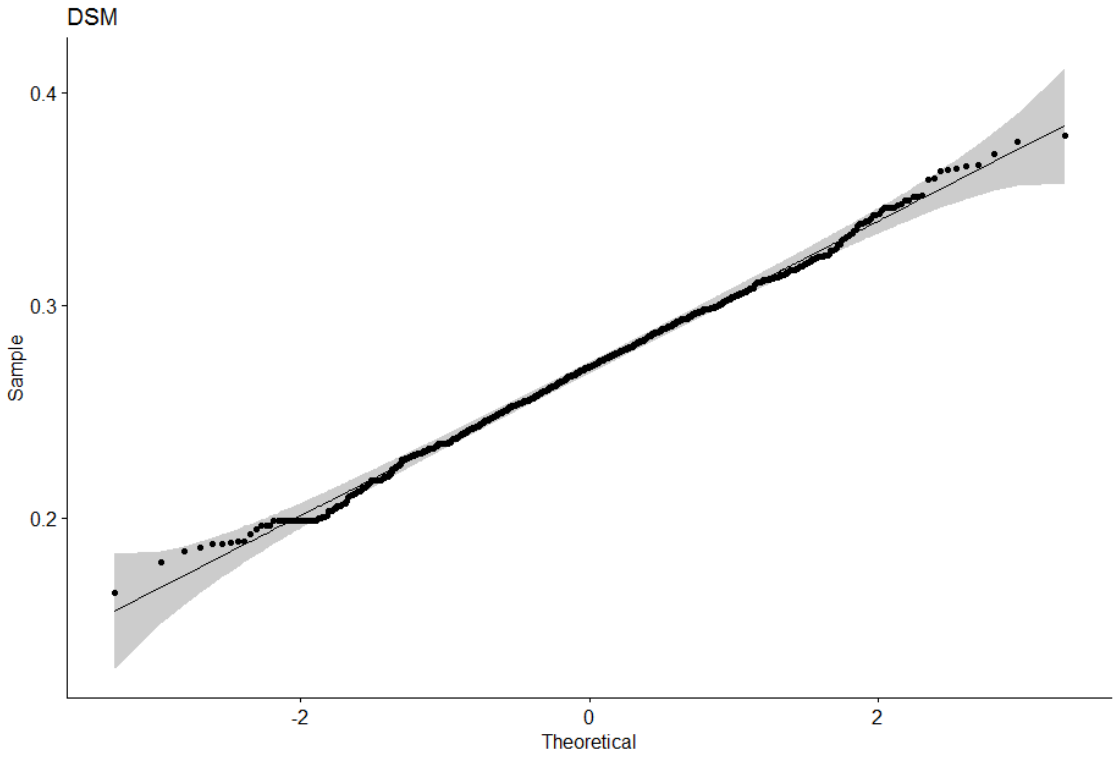

Boxplot of DSM Results
DSM Results
0.40
0.35
0.30
0.25
0.20

Appendix C: Normality Results of the SRL Values for the Sauser SRL Method and DSM Approach for the System Configurations With R






Full
0.4
0.3
0.2
Sample
-2
0
2
Theoretical


DSM
0.4
0.3
0.2
Sample
-2
0
2
Theoretical

Appendix D: Minitab Data Analysis Output

Paired T-Test and CI: DSM Results, Mean (Full System SRL)

Descriptive Statistics

Sample	N	Mean	StDev	SE Mean
DSM Results	1000	0.27029	0.03470	0.00110
Mean (Full System SRL)	1000	0.27406	0.03456	0.00109

Estimation for Paired Difference

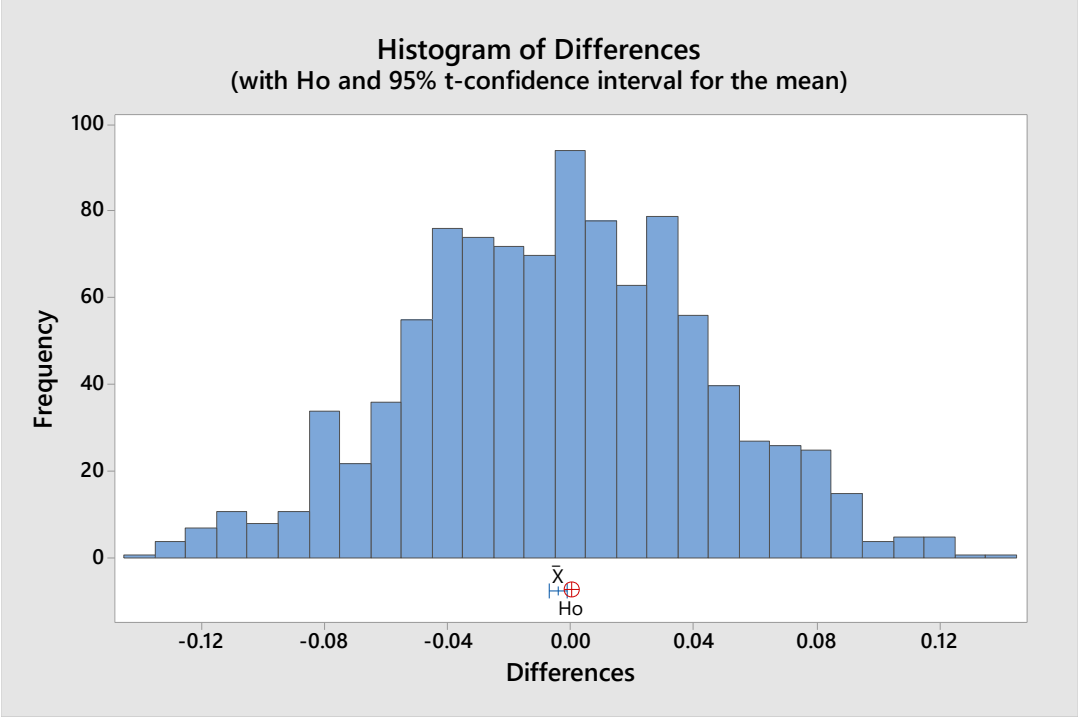
Mean	StDev	SE Mean	95% CI for μ_difference
-0.00377	0.04795	0.00152	(-0.00675, -0.00080)

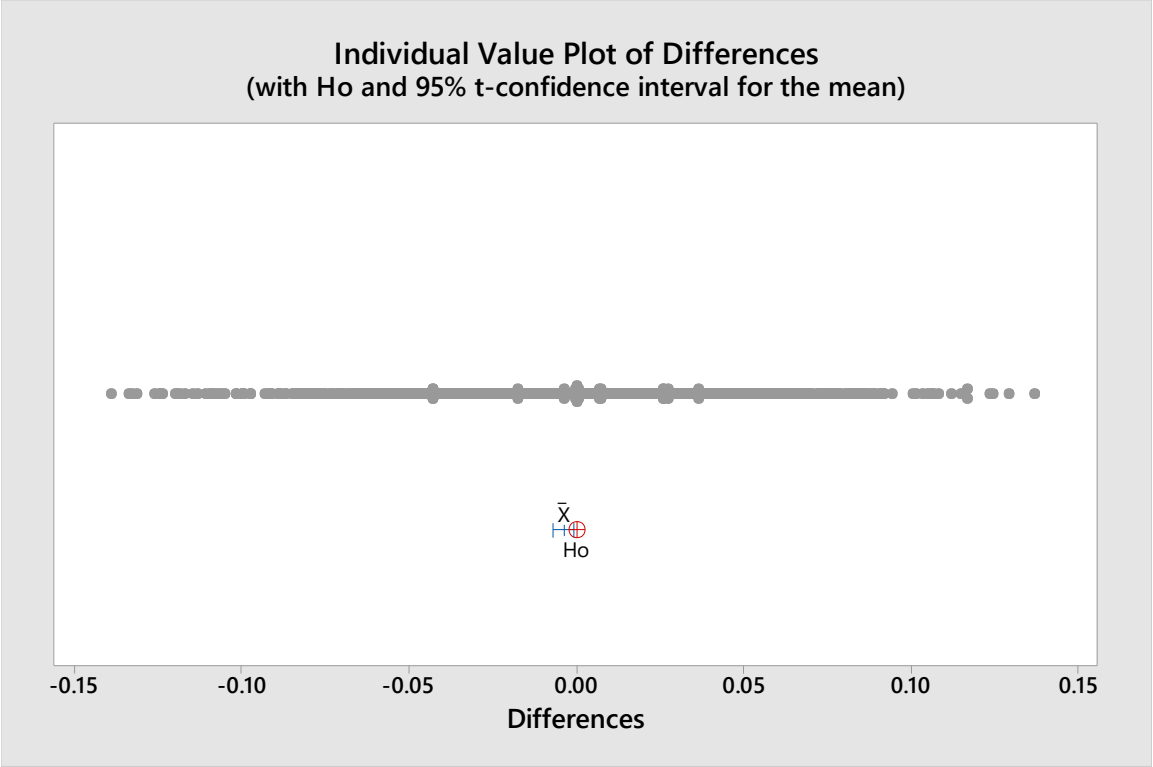
μ_difference: mean of (DSM Results - Mean (Full System SRL))

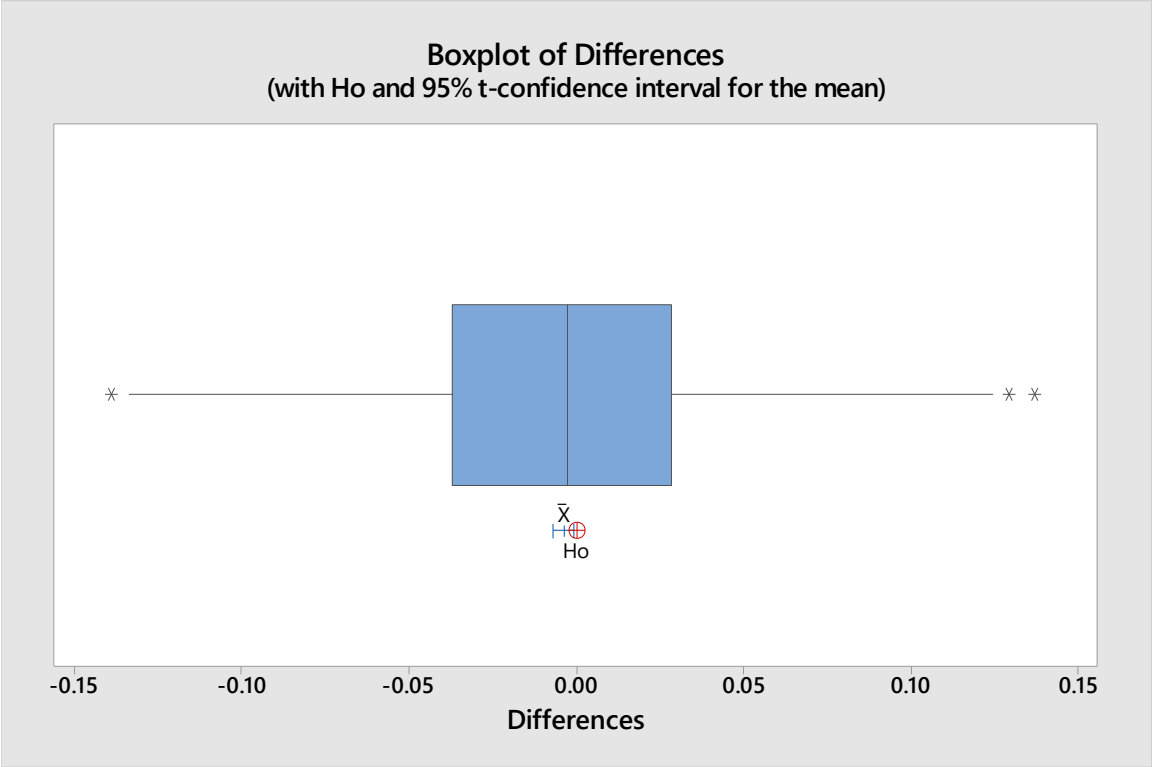
Test

Null hypothesis	H_0: μ_difference = 0
Alternative hypothesis	H_1: μ_difference ≠ 0

T-Value	P-Value
-2.49	0.013




Individual Value Plot of Differences
(with Ho and 95% t-confidence interval for the mean)
X̄
Ho
-0.15
-0.10
-0.05
0.00
0.05
0.10
0.15
Differences


Boxplot of Differences
(with Ho and 95% t-confidence interval for the mean)
X̄
Ho
-0.15
-0.10
-0.05
0.00
0.05
0.10
0.15
Differences

Paired T-Test and CI: DSM Results, Mean (Original Subsystem SRL)

Descriptive Statistics

Sample	N	Mean	StDev	SE Mean
DSM Results	1000	0.27029	0.03470	0.00110
Mean (Original Subsystem SRL)	1000	0.39323	0.08709	0.00275

Estimation for Paired Difference

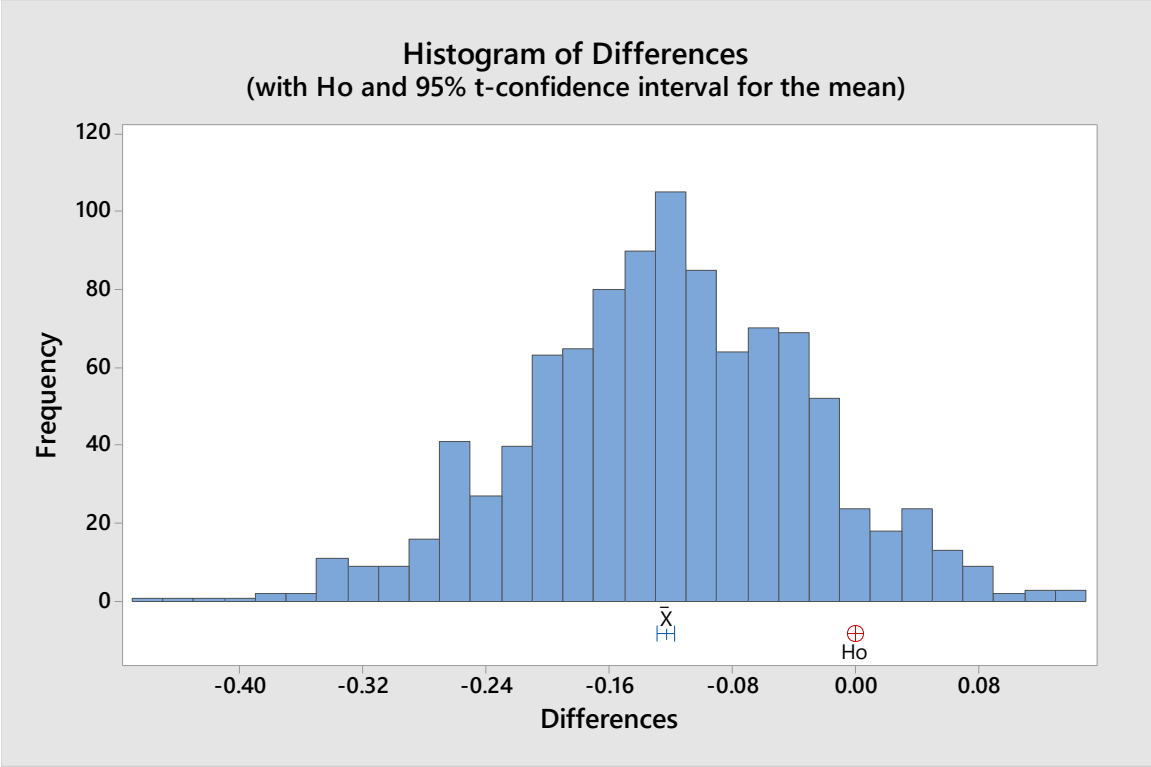
Mean	StDev	SE Mean	95% CI for μ_difference
-0.12294	0.09224	0.00292	(-0.12866, -0.11722)

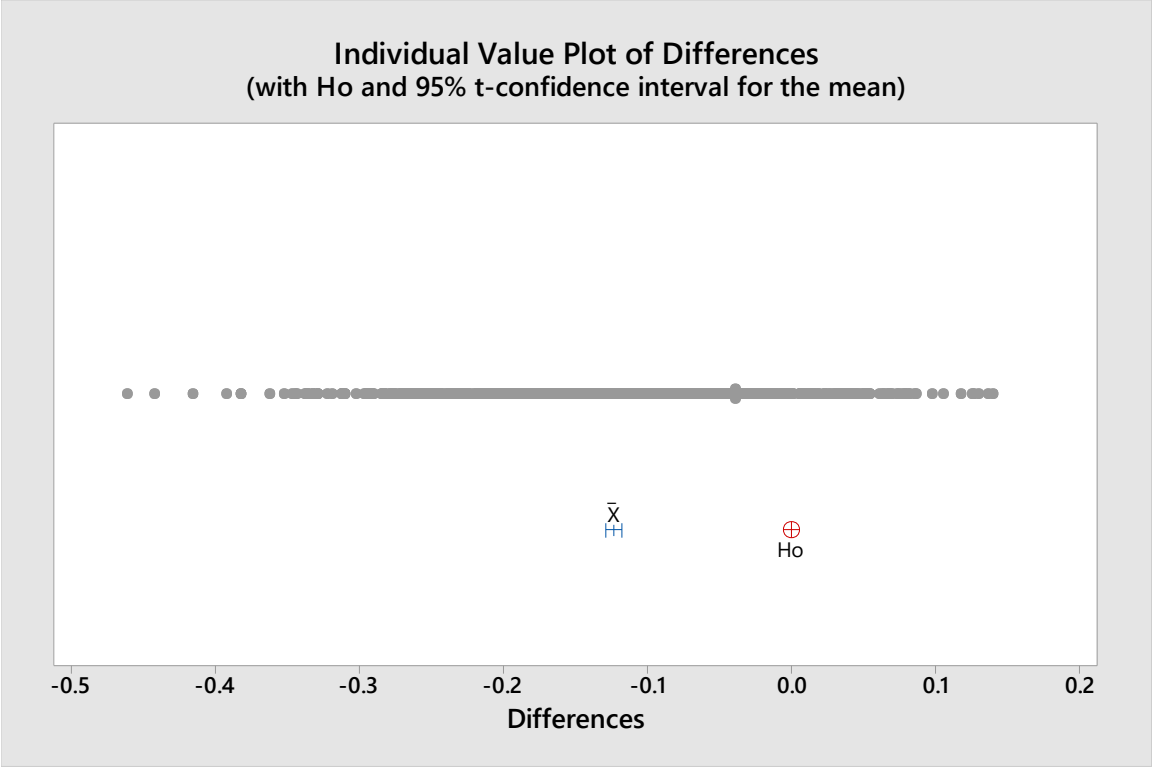
μ_difference: mean of (DSM Results - Mean (Original Subsystem SRL))

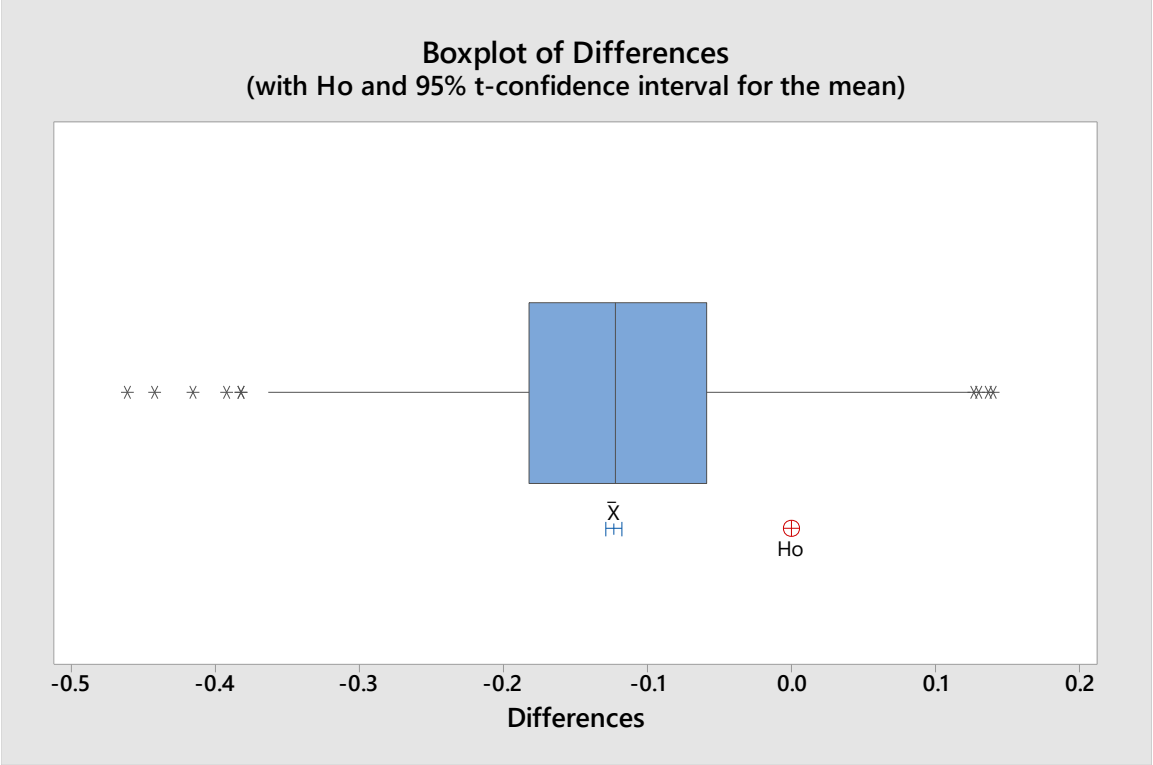
Test

Null hypothesis	H_0: μ_difference = 0
Alternative hypothesis	H_1: μ_difference ≠ 0

T-Value	P-Value
-42.15	0.000




Individual Value Plot of Differences
(with Ho and 95% t-confidence interval for the mean)
X̄
Ho
-0.5
-0.4
-0.3
-0.2
-0.1
0.0
0.1
0.2
Differences


Boxplot of Differences
(with Ho and 95% t-confidence interval for the mean)
X̄
Ho
-0.5
-0.4
-0.3
-0.2
-0.1
0.0
0.1
0.2
Differences

Paired T-Test and CI: DSM Results, Mean (Nested Full System SRL)

Descriptive Statistics

Sample	N	Mean	StDev	SE Mean
DSM Results	1000	0.27029	0.03470	0.00110
Mean (Nested Full System SRL)	1000	0.39671	0.06494	0.00205

Estimation for Paired Difference

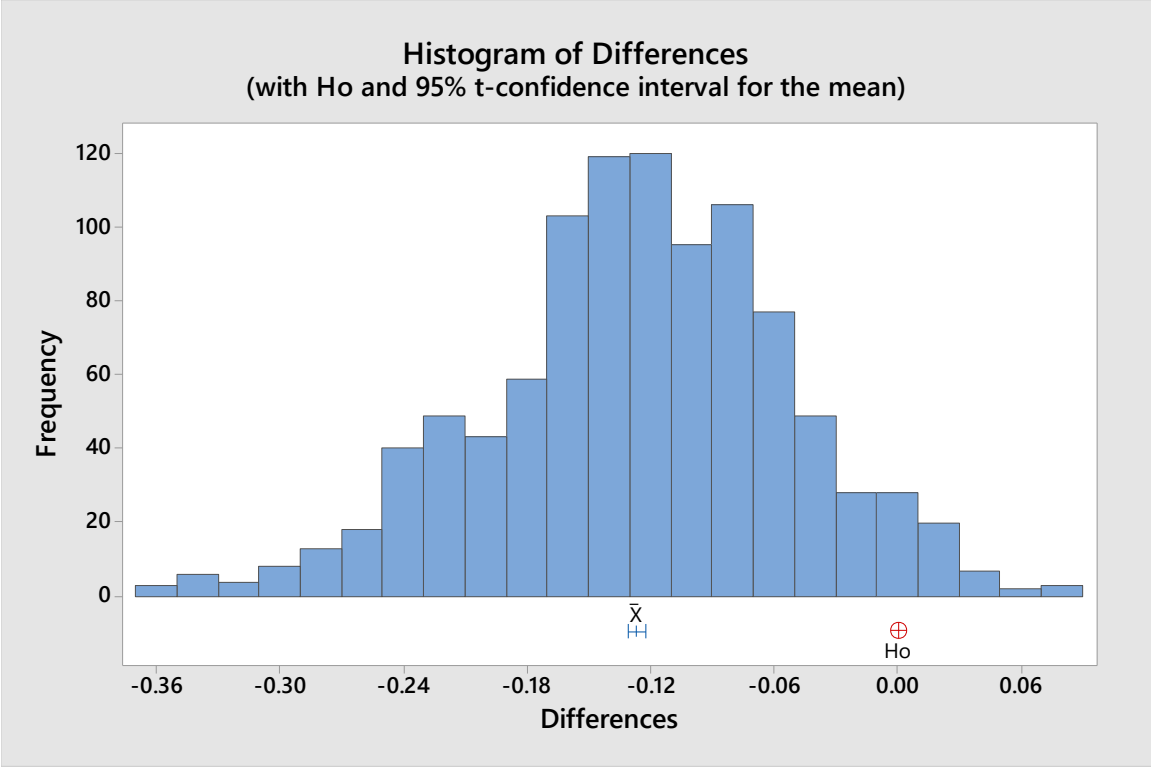
Mean	StDev	SE Mean	95% CI for μ_difference
-0.12642	0.07401	0.00234	(-0.13101, -0.12183)

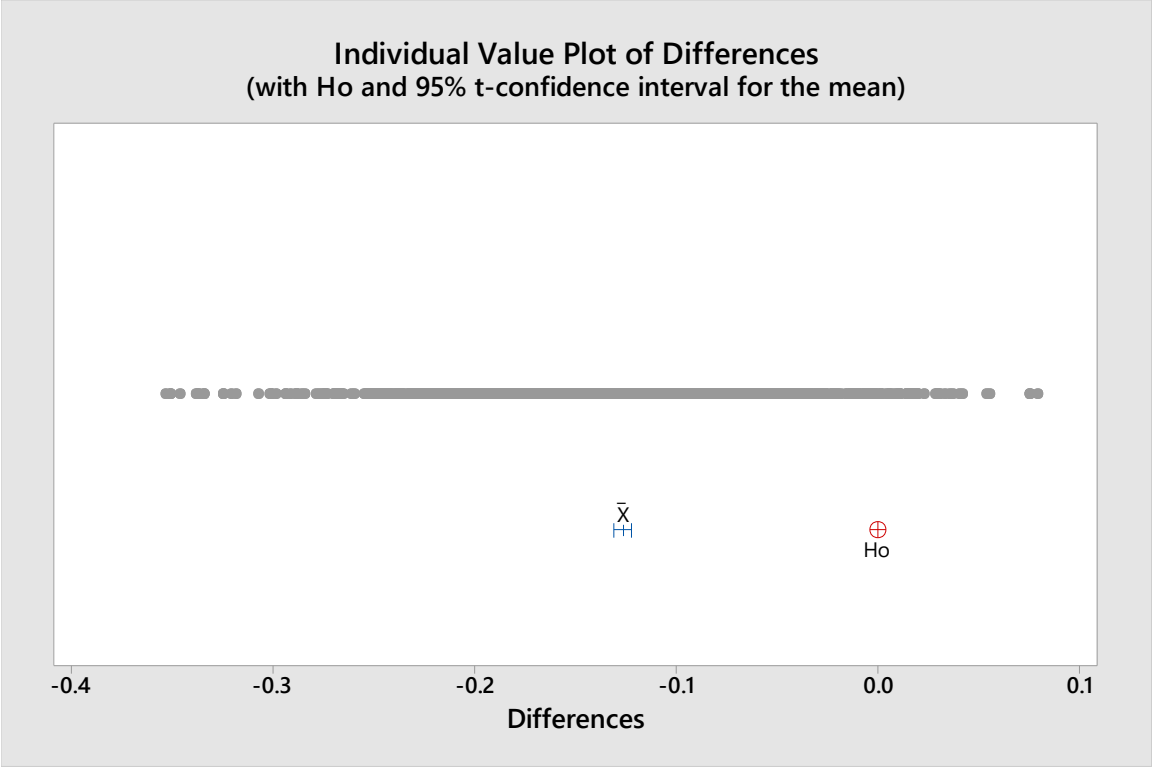
μ_difference: mean of (DSM Results - Mean (Nested Full System SRL))

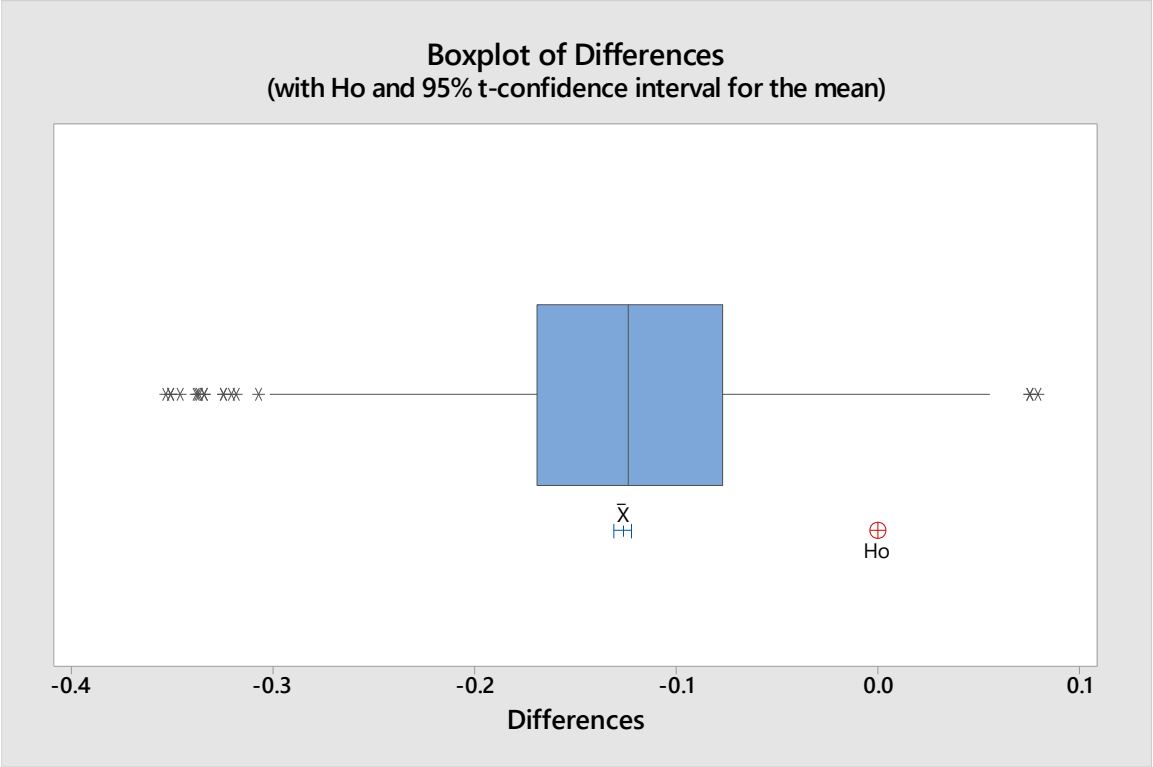
Test

Null hypothesis	H_0: μ_difference = 0
Alternative hypothesis	H_1: μ_difference ≠ 0

T-Value	P-Value
-54.02	0.000




Individual Value Plot of Differences
(with Ho and 95% t-confidence interval for the mean)
X̄
Ho
-0.4
-0.3
-0.2
-0.1
0.0
0.1
Differences


Boxplot of Differences
(with Ho and 95% t-confidence interval for the mean)
X̄
Ho
-0.4
-0.3
-0.2
-0.1
0.0
0.1
Differences

Correlation: DSM Results, Mean (Full System SRL), Mean ... ystem SRL)

Correlations

	DSM Results	Mean (Full Syste	Mean (Nested Ful
Mean (Full Syste	0.042		
	0.188		
Mean (Nested Ful	-0.013	0.000	
	0.692	0.990	
Mean (Original S	0.047	0.015	-0.049
	0.142	0.646	0.120

Cell Contents
Pearson correlation
P-Value

Covariances: DSM Results, Mean (Full System SRL), Mean ... stem SRL)

Covariances

	DSM Results	Mean (Full Syste	Mean (Nested Ful	Mean (Original S
DSM Results	0.00120438			
Mean (Full Syste	0.00004993	0.00119467		
Mean (Nested Ful	-0.00002828	0.00000090	0.00421658	
Mean (Original S	0.00014061	0.00004378	-0.00027830	0.00758546

References

Ackoff, R. L. (2000). Creating the corporate future. *M. Lucas, Understanding Business*.

AlGeddawy, T. (2014). A DSM Cladistics Model for Product Family Architecture Design. *Procedia CIRP, 21*, 87-92. doi:https://doi.org/10.1016/j.procir.2014.03.122

AT&L, O. (2008). *Systems Engineering Guide for Systems of Systems*. Washington D.C. Pentagon Retrieved from https://www.acq.osd.mil/se/docs/se-guide-for-sos.pdf.

Atwater, B., & Uzdzinski, J. (2014). Wholistic Sustainment Maturity: The Extension of System Readiness Methodology across all Phases of the Lifecycle of a Complex System. *Procedia Computer Science, 28*(Supplement C), 601-609. doi:https://doi.org/10.1016/j.procs.2014.03.073

Austin, M. F., & York, D. M. (2015). System Readiness Assessment (SRA) an illustrative example. *Procedia Comput. Sci., 44*, 486-496. doi:10.1016/j.procs.2015.03.031

Austin, M. F., & York, D. M. (2016). System Readiness Assessment (SRA) a Vade Mecum. In *Complex Systems Design & Management* (pp. 53-68): Springer.

Austin, M. F., & York, D. M. (2017). *NSA System Readiness Assessment (SRA) Engineering Handbook*. Ft. George G. Meade, MD: National Security Agency.

Avnet, M. S. (2016). A Network-Based Analysis of Team Coordination and Shared Cognition in Systems Engineering. *Systems Engineering, 19*(5), 395-408. doi:10.1002/sys.21341

Avnet, M. S., & Weigel, A. L. (2010). An application of the Design Structure Matrix to Integrated Concurrent Engineering. *Acta Astronautica, 66*(5-6), 937-949. doi:10.1016/j.actaastro.2009.09.004

Baldwin, K., & Lucero, D. S. (2016). Defense System Complexity: Engineering Challenges and Opportunities *The ITEA Journal of Test and Evaluation*, 10-16.

Bedford, T., & Cooke, R. M. (2001). *Probabilistic Risk Analysis: Foundations and Methods.* Cambridge, UK: Cambridge University Press.

Beiderbecke, B. A. (2013). *Defining and assessing the impact to the combat/weapon systems of a ship post major overhaul.* (Master of Science in Systems Engineering M.S. thesis), Naval Postgraduate School, Monterey, CA.

Brady, T. K. (2002). *Utilization of dependency structure matrix analysis to assess complex project designs.* Paper presented at the ASME 2002 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.

Browning, T. R. (2001). Applying the design structure matrix to system decomposition and integration problems: a review and new directions. *IEEE Transactions on Engineering Management, 48*(3), 292-306. doi:10.1109/17.946528

Browning, T. R. (2002). Process integration using the design structure matrix. *Systems Engineering, 5*(3), 180-193. doi:10.1002/sys.10023

Browning, T. R. (2016). Design Structure Matrix Extensions and Innovations: A Survey and New Opportunities. *IEEE Transactions on Engineering Management, 63*(1), 27-52. doi:10.1109/TEM.2015.2491283

Carlock, P. G., & Fenton, R. E. (2001). System of Systems (SoS) enterprise systems engineering for information-intensive organizations. *Systems Engineering, 4*(4), 242-261.

DAU. (2017). Defense Acquisition University - Defense Acquisition Guidebook. *Chapter 3 Systems Engineering.* Retrieved from https://www.dau.mil/guidebooks/Shared%20Documents%20HTML/Chapter%203%20Systems%20Engineering.aspx#toc20

DoD. (2008). *Systems engineering guide for systems of systems*. Retrieved from Washington, DC:

DoD. (2009). *Technology Readiness Assessment (TRA) Deskbook*. Retrieved from WASHINGTON DC: http://www.dtic.mil/dtic/tr/fulltext/u2/a524200.pdf

DoD. (2015). DoD instruction 5000.02: Operation of the defense acquisition system. In: Author Washington, DC.

Eppinger, S. D., & Browning, T. R. (2012). *Design structure matrix methods and applications* (1 ed. Vol. 1). Cambridge, Mass: MIT Press.

Eppinger, S. D., Whitney, D. E., Smith, R. P., & Gebala, D. A. (1991). Organizing the tasks in complex design projects. In D. Sriram, R. Logcher, & S. Fukuda (Eds.), *Computer-Aided Cooperative Product Development* (pp. 229-252). Berlin, Heidelberg: Springer Berlin Heidelberg.

Fomin, P. (2010). *Growing the house of quality.* (Dissertation/Thesis), ProQuest Dissertations & Theses Global. Retrieved from http://proxygw.wrlc.org/login?url=https://search-proquest-com.proxygw.wrlc.org/docview/820632172?accountid=11243

Fomin, P., Mazzuchi, T. A., & Sarkani, S. (2009). *Incorporating Maturity Assessment into Quality Functional Deployment for Improved Decision Support Analysis, Risk Management, and Defense Acquisition.* Paper presented at the World Congress on Engineering and Computer Science 2009, 20-22 Oct. 2009, Hong Kong, China.

GAO. (2009). *GAO Cost Estimating and Assessment Guide: Best Practices for Developing and ManagingCapital Program Costs*. Retrieved from

GAO. (2013). *FORD-CLASS CARRIERS: lead ship testing and reliability shortfalls will limit initial fleet capabilities*. Retrieved from

GAO. (2017). *Follow-On Ships Need More Frequent and Accurate Cost Estimates to Avoid Pitfalls of Lead Ship*. Retrieved from

GAO. (2018). *Navy Shipbuilding: Past Performance Provides Valuable Lessons for Future Investments*. Retrieved from

Garg, T., Eppinger, S., Joglekar, N., & Olechowski, A. (2017). *Using TRLS and system architecture to estimate technology integration risk.* Paper presented at the 21st International Conference on Engineering Design, ICED 2017, August 21, 2017 - August 25, 2017, Vancouver, BC, Canada.

Garrett, R. K., Anderson, S., Baron, N. T., & Moreland, J. D. (2011). Managing the interstitials, a System of Systems framework suited for the Ballistic Missile Defense System. *Systems Engineering, 14*(1), 87-109. doi:10.1002/sys.20173

IEEE. (2015). ISO/IEC/IEEE International Standard - Systems and software engineering -- System life cycle processes. In *ISO/IEC/IEEE 15288 First edition 2015-05-15* (pp. 1-118).

INCOSE, D. D. W. (2015). Systems engineering handbook: A guide for system life cycle processes and activities. *Fourth Edition*.

Jamshidi, M. (2008). *Systems of systems engineering: principles and applications*: CRC press.

Katz, D. R. (2014). *The Relationship of Technology and Design Maturity to DoD Weapon System Cost Change and Schedule Change During Engineering and Manufacturing Development.* (Dissertation/Thesis), ProQuest Dissertations & Theses Global, Retrieved from http://proxygw.wrlc.org/login?url=https://search-proquest-com.proxygw.wrlc.org/docview/1506156389?accountid=11243

Katz, D. R., Sarkani, S., Mazzuchi, T., & Conrow, E. H. (2015). The Relationship of Technology and Design Maturity to DoD Weapon System Cost Change and Schedule Change During Engineering and Manufacturing Development. *Systems Engineering, 18*(1), 1-15. doi:10.1111/sys.21281

Knaggs, M., Harkreader, D., Unione, A., Oelfke, J., Ramsey, J., Keaims, D., . . . Atwater, B. (2017). *Nesting in the evaluation of system readiness for complex systems of emerging technologies.* Paper presented at the 11th Annual IEEE International Systems Conference, SysCon 2017, April 24, 2017 - April 27, 2017, Montreal, QC, Canada.

Levandowski, C., Müller, J. R., & Ola, I. (2016). *Modularization in Concept Development Using Functional Modeling*. Paper presented at the 2016 ISPE International Conference on Transdisciplinary Engineering.

London, M. A., Holzer, T. H., Eveleigh, T. J., & Sarkani, S. (2014). Incidence matrix approach for calculating readiness levels. *Journal of Systems Science and Systems Engineering, 23*(4), 377-403. doi:10.1007/s11518-014-5255-8

Long, J. M. (2011, 5-12 March). *Integration readiness levels.* Paper presented at the Aerospace Conf., Big Sky, MT.

Lorell, M. A., Lowell, J. F., & Younossi, O. (2006). *Evolutionary Acquisition Implementation Challenges for Defense Space Programs*. Arlington, VA: RAND Corporation.

M., S. D., & A., Y. A. (2004). Characterizing complex product architectures. *Systems Engineering, 7*(1), 35-60. doi:doi:10.1002/sys.10056

Maier, M. W. (1998). Architecting principles for systems-of-systems. *Systems Engineering, 1*(4), 267-284. doi:10.1002/(SICI)1520-6858(1998)1:4<267::AID-SYS3>3.0.CO;2-D

Malone, P., & Smoker, R. (2013, September 10, 2013 - September 12, 2013). *Using system complexity to increase cost estimate accuracy in government procurements.* Paper presented at the AIAA SPACE 2013 Conf. and Expo., San Diego, CA, United states.

McConkie, E., Mazzuchi, T. A., Sarkani, S., & Marchette, D. (2013). Mathematical properties of system readiness Levels. *Syst. Eng., 16*(4), 391-400. doi:10.1002/sys.21237

Michael, J., & Byers, R. (2013). Warfare System Interface Diagrams (WSIDs) in support of combat systems modernization. *Leading Edge*(NSWCDD/MP-13/22).

Minitab. (2017). Minitab 18 Statistical Software. State College, PA. Retrieved from (www.minitab.com)

Olechowski, A., Eppinger, S. D., & Joglekar, N. (2015, August 2, 2015 - August 6, 2015). *Technology readiness levels at 40: A study of state-of-the-art use, challenges, and opportunities.* Paper presented at the Portland Int. Center for

Management of Engineering and Technology, PICMET 2015, Portland, OR, United states.
Ross, S. (2016). Application of System and Integration READINESS LEVELS to Department of Defense RESEARCH and DEVELOPMENT*23*(3), 248. Retrieved from
Sage, A. P., & Cuppan, C. D. (2001). On the systems engineering and management of systems of systems and federations of systems. *Information knowledge systems management, 2*(4), 325-345.
Sauser, B., Ramirez-Marquez, J., Verma, D., & Gove, R. (2006). Determining system interoperability using an integration readiness level. *Stevens Institute of Technology, Systems Engineering and Engineering Management, available at: http://www. dtic. mil/ndia/2006systems/Thursday/sauser. pdf (accessed 29th November 2008)*.
Sauser, B., Ramirez-Marquez , J. E., Nowicki, D., Deshmukh, A., & Sarfaraz, M. (2011). *Development of Systems Engineering Maturity Models and Management Tools*. Retrieved from Systems Engineering Research Center, Hoboken, NJ:
Sauser, B., Verma, D., Ramirez-Marquez, J., & Gove, R. (2008). *System Maturity Indices for Decision Support in the Defense Acquisition Process*. Paper presented at the 5th Annual Acquisition Research Symposium of the Naval Postgraduate School, Monterey, CA.
Sauser, B., Verma, D., Ramirez-Marquez, J., & Gove, R. (2008). A systems approach to expanding the technology readiness level within defense acquisition. *Int. J. Defense Acquisition Manage. (IJDAM), 1*, 39-58.
Sauser, B. J., Verma, D., Emmanuel, R.-M. J., & Ryan, G. (2006). *From TRL to SRL: The Concept of System Readiness Levels*. Paper presented at the Conference on Systems Engineering Research, Los Angeles, CA.
Schwartz, M. (2016). Defense Acquisitions: How DOD Acquires Weapon Systems and Recent Efforts to Reform the Process (Washington, DC: Congressional Research Service), May 2014. In.
Sheard, S. A., & Mostashari, A. (2009). Principles of complex systems for systems engineering. *Systems Engineering, 12*(4), 295-311. doi:10.1002/sys.20124
Smith, R. P., & Eppinger, S. D. (1990). *Modeling Design Iteration, Working Paper*. Cambridge, MA: Sloan School of Management, MIT.
Steward, D. V. (1981). The design structure system: a method for managing the design of complex systems. *IEEE Trans. Eng. Manage., EM-28*(3), 71-74. doi:10.1109/TEM.1981.6448589
Tan, W., Ramirez-Marquez, J., & Sauser, B. (2011). A probabilistic approach to system maturity assessment. *Syst. Eng., 14*(3), 279-293. doi:10.1002/sys.20179
Tan, W., Sauser, B. J., & Ramirez-Marquez, J. E. (2011). Analyzing component importance in multifunction multicapability systems developmental maturity assessment. *IEEE Trans. Eng. Manage., 58*(2), 275-294. doi:10.1109/TEM.2010.2071877
Thal, B., Olson, B., & Blessner, P. (2018). A robust system maturity model for complex systems utilizing system readiness level and Petri nets. *Complex & Intelligent Systems*. doi:10.1007/s40747-018-0073-0

Thebeau, R. E. (2001). *Knowledge management of system interfaces and interactions from product development processes.* (Masters in Science), Massachusetts Institute of Technology, Massachusetts Institute of Technology. Retrieved from http://hdl.handle.net/1721.1/29168 (48272131-MIT)

Yassine, A., & Braha, D. (2003). Complex Concurrent Engineering and the Design Structure Matrix Method. *Concurrent Engineering, 11*(3), 165-176. doi:10.1177/106329303034503

Yassine, A. A. (2004). An Introduction to Modeling and Analyzing Complex Product Development Processes Using the Design Structure Matrix (DSM) Method. *Urbana, 51*(9), 1-17.