DISSERTATION

THE NEXT GENERATION SPACE SUIT: A CASE STUDY OF THE SYSTEMS ENGINEERING CHALLENGES IN SPACE SUIT DEVELOPMENT

Submitted by

Michael A. Cabrera

Department of Systems Engineering

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Fall 2023

Doctoral Committee:

Advisor: Steve Simske

Greg Marzolf
Erika Miller
Maria Delgado

ABSTRACT

THE NEXT GENERATION SPACE SUIT: A CASE STUDY OF
THE SYSTEMS ENGINEERING CHALLENGES IN SPACE SUIT
DEVELOPMENT

The objective for a NASA contractor, the performing organization in this case study, is to develop and deliver the next generation space suit to NASA, the customer in this case study, against a radically different level of customer expectation from previous years. In 2019, the administration had proposed a return to the moon, thus transforming and changing the system context of the current, next generation space suit in addition to pushing schedule expectations forward two years. The purpose of this dissertation will serve as a case study in two specific areas with qualitative and quantitative analyses regarding a new process and approach to (i) project lifecycle development and (ii) requirements engineering with the intent that if utilized, these tools may have contributed to improvements across the project in terms of meeting cost, scope, budget and quality while appropriately accounting for risk management. The procedure entails a research method in which the current state of the project, current state of the art, and the identified systems engineering challenges are evaluated and iterative models are tempered through development by continual improvements by engineering evaluation of engineers on the project. The current results have produced (i) a prototype project lifecycle development method via agile, Lean and Scrum hybrid implementations into a Traditional Waterfall framework and (ii) a prototype requirements engineering scorecard with implementations of FMEA and quantitative analysis to determine root cause identification.

ACKNOWLEDGEMENTS

I would first like to thank my parents, Ralph and Niza Cabrera for supporting me since the beginning in every regard, but especially now for their patience and encouragement during my entire academic career. A special thanks goes to my academic advisor at Colorado State University, Dr. Steve Simske for bringing me into the program, helping me develop my dissertation idea, his guidance during my coursework and teaching me the value of contributing back to the field of engineering. I would like to thank Dr. Maria Delgado, Dr. Erika Miller and Dr. Gregory Marzolf for their patience, input and time spent on the doctoral committee. I would like to thank all of those across NASA, JETS and xEMU, especially in the Systems Engineering Department, GSE Testing Team and Project Management Department for supporting me in my research with exclusive thanks to the following individuals: Vanessa Beene and Vladenka Oliva for assisting me through the proper channels to help me publish and release the study and also their tremendous patience and willingness to support. Brian Neumann for working with me early in my research, spending large amounts of time teaching me the system, inspiring ideas for case studies and for helping me ask the right questions; Jeremy Hemler for inspiring some of the case studies and explaining with sufficient details challenges he faced during the project; Donald Barker for his expertise in lunar dust mitigations and helping a fellow Ram be successful with his PhD and sharing his experiences with his thesis to allow me to be successful; Gregg Weaver, my manager who has had the utmost patience with my project management career and keeping me focused on all things important; Gary Bradley for taking a chance on bringing me to the team to help him manage, develop and groom our engineers and teaching me all things project management; Thomas Smith for his guidance, counsel and ability to clearly and calmly approach the challenges we faced together; David Westheimer for his patience and methodical examination of my dissertation as a technical reviewer. Additionally, I would like to thank the faculty at Colorado State University for their guidance and support during my course path through the program as without the necessary coursework, I would not have been inspired to direct my dissertation efforts in the appropriate direction. In addition to Dr. Steve Simske, another faculty member I give special thanks is to Dr. Ann Batchelor, my Requirements Engineering professor that illustrated and inspired in me the importance of building requirements and their documentation to allow for project and product success, as a large portion of my research is dedicated to those ideas. A huge thanks also goes to the students and faculty of Texas A&M's Manufacturing and Mechanical Engineering Technology program. Finally, I would like to thank Ingrid Bridge for her patience and support during my time before and during my tenure at Colorado State.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. CHAPTER 1: INTRODUCTION

Lede: This chapter identifies the problem to be resolved, what motivated the research, an overview of the solution and the methodical approach to the research. Additionally, the research questions, hypotheses, null hypotheses and products of the research are detailed.

1.1. Statement of the Problem

The current space suit, the Extravehicular Mobility Unit (EMU), is utilized by crew members onboard the International Space Station (ISS) and was last heavily designed in 1984 (Martin, 2021). This suit lacks the ability to sustain both lunar and beyond missions and as a result, the Exploration Extra Vehicular Mobility Unit (xEMU), both the next generation suit and project name, has been in development since 2007, which is an extension of previous project the Advanced Extravehicular Mobility Unit (AEMU). The goal of xEMU is to provide NASA with the next generation space suit. In 2019, Vice President Mike Pence announced at the National Space Council that astronauts would return to the moon by 2024 (Wall, 2019). This announcement provided the division developing the xEMU, the Crew and Thermal Systems Division, with a budget and expedited delivery date applied to their current xEMU efforts but with additional scope to account for areas the xEMU project did not specify in the work break down structure, especially in the areas of lunar dust and extended EVA time. As a result of the progress in terms of scope, cost and budget from 2019 to 2021, Office of the Inspector General of NASA performed an audit and released Report No. IG-21-025 of the status and development of the new space suits for the current and new space stations in addition to the Human Launch Support (HLS) programs. As a result of the audit the Exploration Extravehicular Activities (xEVAS) contract is planning to cancel the xEMU project and transfer the current work completed and subject matter expert (SME) knowledge and transfer that to a new, commercial industry partner that is yet to be awarded and potentially removing the JSC Engineering and Technology Services (JETS), the performing organization in this study, from future work on the next generation space suit. However, there is no implication to suggest that JETS or its supporting entities were direct impactors but due to closing of the project, the benefit of a case study and potential lessons learned exercise is merited.

1.2. Motivation

With the Inspector General's Audit and recommendation, in conjunction with the request for information and proposal NASA put forward in 2021, future work is being transferred to industry instead of the continued JETS contract with NASA, effectively putting an end the xEMU project. The anticipated next generation space suit and collection of work from xEMU however will transfer to a contracting partnership through the xEVAS request for proposal that was announced in 2021 and will be awarded to a commercial organization in early 2022 yet to be announced (Martin, 2021). While there are many motivational factors to consider, many of the key issues and challenges associated with the change of direction from a direct NASA contractor to an industry commercial partner stem from a politically driven motive to perform future work more efficiently and effectively in terms of schedule, budget and resources. As a result of JETS working within the constraints and faculties of a government agency which may have impacted the proposal of work to transfer from JETS to an industry partner, the motivation is to provide a knowledge capture in this dissertation, with an emphasis in areas that could have been optimized, particularly areas of Systems Engineering interest.

1.3. Overview of Solution

The recommendations prompted by this research is to use the xEMU project as an extended case study to determine the root causes of project challenges and optimizations as they are related to systems engineering practices and provide qualitative and quantitative data with a potential recommendation to the new industry partner to improve their ability to successfully deliver a space suit to the moon with the new projected schedule, budget and within scope.

1.4. Research Approach

The approach, research method, methodologies, research question, hypotheses, case studies and forward work described in this section were progressively elaborated throughout the dissertation process. The goal of illustrating the approach at the onset of the dissertation is to guide the reader throughout the subsequent chapters to allow for guidance and illustration of the importance of the preliminary work which supported the research approach. The elaboration was guided by understanding the current state of the project, current state-of-the-art and available case studies which could provide adequate data to develop research questions, form hypotheses and develop artifacts (products and presentations) as an output of the dissertation process.

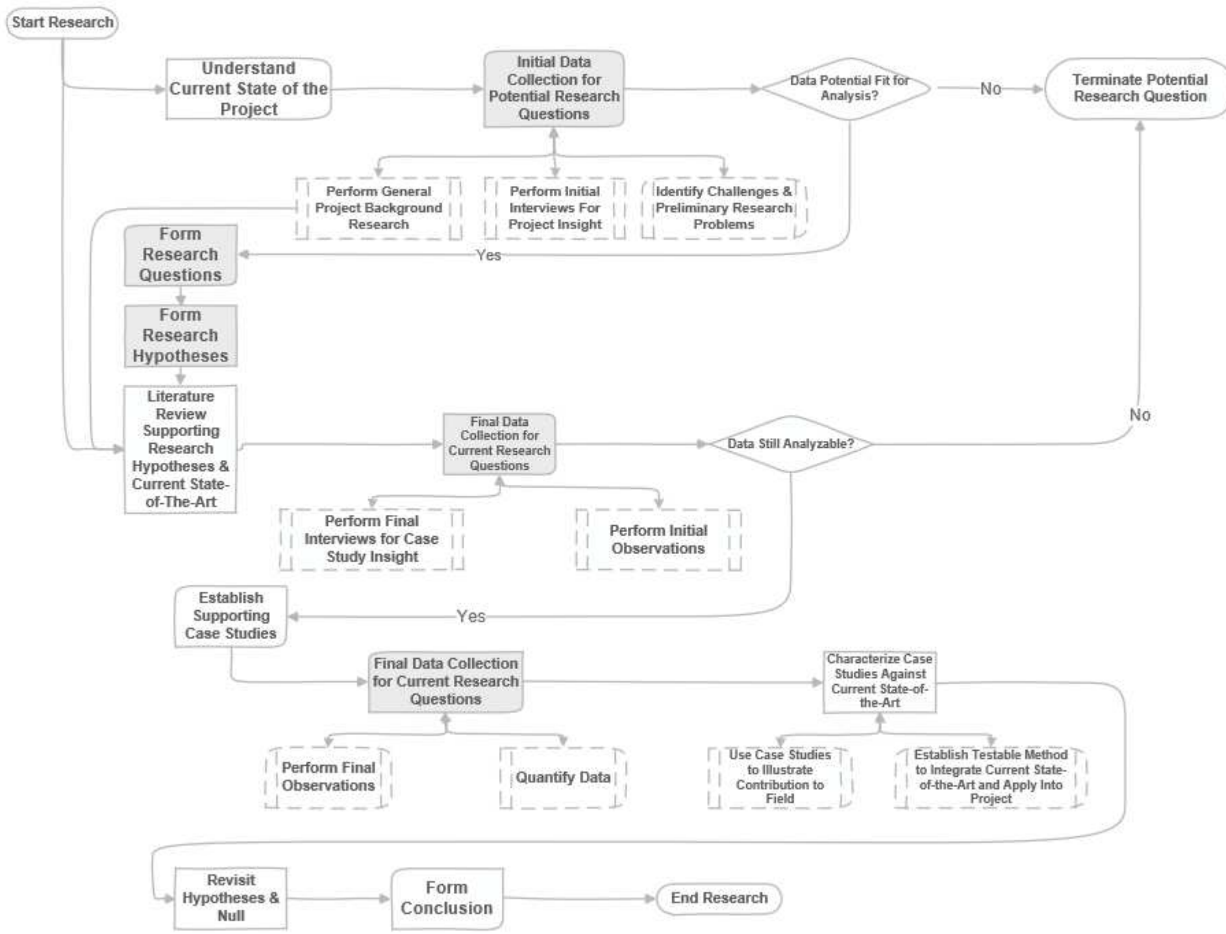


Figure 1 – Research Approach

During the research, two distinct areas of interest in systems engineering were presented as challenges: systems engineering challenge #1: requirements engineering; systems engineering challenge #2: project lifecycle development. The intention in the flowchart is to juxtapose the current vs. proposed project development by detailing how current approaches were hypothesized to be less effective than their potential development. Comparative analysis derived approaches in the current state-of-the-art that could be applicable to potentially improve the conditions of the systems engineering challenges by posing potential solutions to the current state of the project. Approaches that worked on the project would potentially support the proposed development and knowledge gaps were emphasized where both case study research and current state-of-the-art research might potentially close those knowledge gaps in the field of systems engineering.

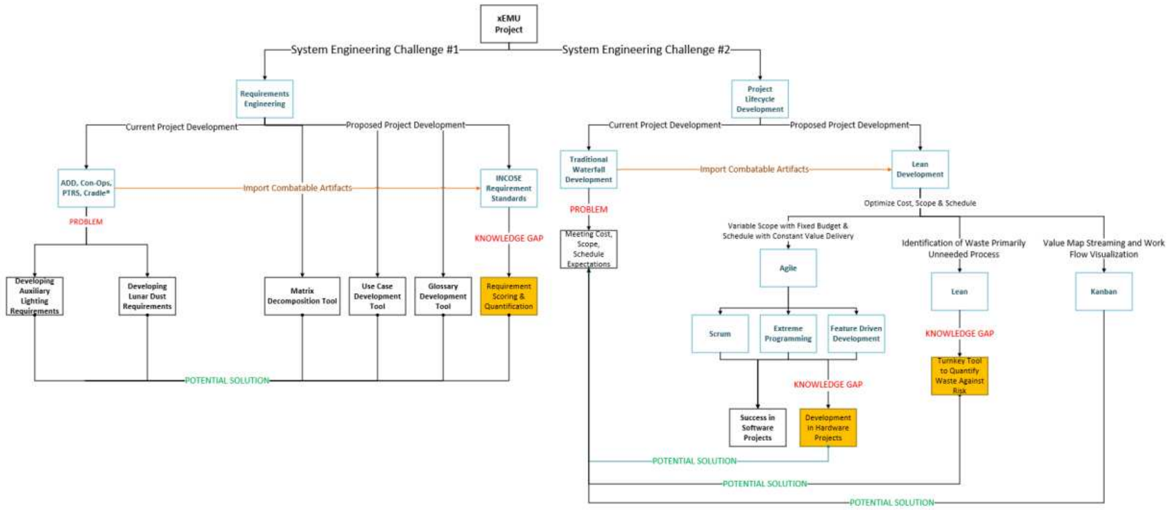


Figure 2 – Knowledge Gaps, Problems & Solutions on xEMU Project

A methodical approach with a brief description details while tabular was by no means a linear method to the research approach. The products of the research were the Modified Agile Concept (MAC) and the Requirements Engineering Scorecard (RES), and their relevance was first dictated in the methodical approach below. The following research questions, hypotheses and deliverables (products and presentation) are detailed.

Step #	Description
Step 1	***Formulation of research questions* (see slides).** Guiding research questions formulated based on interviews, surveys, brainstorming and focus groups.
Step 2	***Selection of project data sources.*** Reference materials were selected from space suit project performance databases, interviews with team members, NASA internal documents, budget reports and schedule performance reviews.
Step 3	***Selection of academic literature.*** Peer reviewed studies and academic texts on project lifecycle development and requirements engineering. Governing bodies (PMI, INCOSE, Scrum Alliance, etc.) were investigated. CSU Primo, Google Scholar were selected for search databases.
Step 4	***Selection of case studies.*** Steps 1-3 allow for challenges to emerge which support the establishment of case studies. Those challenges applicable to systems engineering that have sufficient information to be analyzed with data that is quantifiable were selected. Project lifecycle development (with a study into testing station project development) and requirements engineering (with a study into lunar dust and auxiliary lighting on the space suit) were selected.
Step 5	***Formulation of hypotheses* (see slides).**
Step 6	***Formulation of test method via tempered models* (see slides).**
Step 7	***Data recording.*** ***(1)*** Quantifiable data <u>for current project lifecycle case study</u> will be analyzed by calculation of Earned Value Metrics to demonstrate pre-condition. While a new project cannot be feasibly evaluated or launched against the new process, **the Modified Agile Concept (MAC)**, facets of the model can be evaluated, and data compared pre and post MAC. The tempered model's progression will also have an evaluation on more of a (A|B) comparison and FMEA quantitative/qualitative data. **(2)** Quantifiable data for <u>requirements engineering case studies</u> will be analyzed by evaluating requirements against the **Requirements Engineering Scorecard (RES)**. The scorecard will be used to grade the previous lunar dust and auxiliary lighting requirements as they were originally done, and the scorecard will be used to guide and grade the proposed lunar dust and auxiliary lighting requirements development. Scores will be compared. The tempered model's progression will also have an evaluation on more of a (A|B) comparison.
Step 8	***Reporting Results & Conclusion***. The final goal is to report results holistically, both with respect to the case study developments' products and an overall assessment of the system engineering posture moving forward with the project and any potentially beneficial artifacts to the field of systems engineering.

Figure 3 – Research Method

<u>Research Question #1:</u>

Can the lunar dust and auxiliary lighting suit requirements' challenges be resolved by applying a characterization against INCOSE writing practices to guarantee robustness via current state-of-the-art requirements development methods in a scorecard?

<u>Research Question #2:</u>

Will a modified, agile-hybrid project lifecycle development model applied to waterfall teams develop a superior product within time and schedule constraints in a hardware-intensive environment?

<u>Hypotheses (H_a)</u>

- If the lunar dust requirements including verifications and validations were decomposed via an INCOSE-influenced scorecard process, the project would have better approximated the anticipated product against the customer's expectations.

- If the auxiliary lighting requirements including verifications and validations were decomposed via an INCOSE-influenced scorecard process, the project would have better approximated the anticipated product against the customer's expectations.
- If the testing team developed their team and products against a modified, hybrid-agile methodology for hardware-intensive systems, the schedule and budget would have been better approximated per the customer's expectations.

Hypotheses (H_o)

- The project would not have better approximated the anticipated product against the customer's expectations if the lunar dust requirements including verifications and validations were decomposed via an INCOSE-influenced scorecard.
- The project would not have better approximated the anticipated product against the customer's expectations if the auxiliary lighting requirements including verifications and validations decomposed via an INCOSE-influenced scorecard.
- The schedule and budget would have been better approximated per the customer's expectations if the testing team developed their team and products against a modified, hybrid-agile methodology for hardware-intensive systems.

Product:

- (1) A prototype for a hybrid project lifecycle development method to promote a hypothesized optimization of any aerospace hardware builds in terms of cost, scope, schedule, quality and safety but more specifically for GSE projects in the MAC. This will take the form of a document with supporting templates.

- (2) A prototype for requirements engineering robustness scoring to promote a hypothesized optimization of requirement definition, decomposition to approximate a superior product in the RES. This will take the form of a document with supporting templates.

Presentation

- (3) While the information I have works as a stand-alone product, I find the spirit of the product and communication of issues that arose on the project will be beneficial for others inside and outside the industry. For that reason, presenting this information generally to support others in need of requirements engineering and project lifecycle development is of value. This will take the form of a presentation with supplementary products (1) and (2).

2. CHAPTER 2: PROJECT BACKGROUND

Lede: This chapter provides an exhaustive overview of the current and future space suits, both with regards to their system and subsystem designs and their approach to hierarchical project, organizational and work breakdown development. The purpose was to provide the necessary context to support further sections with the intent of uniting the project work with the systems engineering applicable facets. Additionally, the current cost and schedule are outlined.

2.1. Overview of Space Suit Development

A cursory overview of the past, current and future space suits is given for context to understand decisions made in terms of form, fit, function, quality and safety.

2.1.1. Past and Current EMUs

The EMU is the United States space suit that performs the necessary life functions to sustain human life in a mobilized from in the vacuum of space. Extra-vehicular activity (EVA) dates back to 1965 during the Gemini program. The design changed dramatically and rapidly during the Gemini, Apollo and Shuttle programs whereas existing design modifications were made as revisions during the Skylab and ISS programs (Prouty, 1991). Extensive time was needed to work outside of a pressurized habitat in space and as a result, technology needed to advance to parallel the advancements in pressurized habitats in a vacuum, specifically Shuttle and ISS operations. As a result of unanticipated issues, the Skylab astronauts executed twelve different contingency EVAs (Goodman, 1991). Although these EVAs proved to be advantageous, the Shuttle program was not adequately designed to account for EVAs in the design of the legacy space suits and as a result of this unincorporated design, EMU development was delayed in excess of four years in comparison to the shuttle program. Much like the Shuttle program, the EMU was designed without a developmental phase and initial hardware designs were tested to the standards of flight hardware and certified in parallel with the Shuttle flights. This resulted in 4,000 hours of EMU processing in between Shuttle flights (Peacock, 1991). The result of more EVAs resulted in more complex designs, which provoked more complex EVAs and time in space. In comparison to all available data on EVAs, the EMU has performed 75% of the total EVAs (Jordan, 2006). Although the idea surrounding an EMU was for contingency and limited capacity operations, the current and future demand requires a suit for longer durations and harsher environments.

2.1.2. Future xEMU Development

The xEMU is an extension of the EMU and is designed for operations in low-earth orbit, cis-lunar orbit, the lunar surface and eventually deep space with a mission to Mars. The xEMU is an improvement over the legacy EMU with advancements in planetary and upper suit mobility, Environment Protection Garment (EPG) with dust mitigation, automated suit checkout, integrated communications, informatics display and control, HD video and lights, high speed data communication, one-hour emergency return, maximum variable pressure of 8.2 psi, vacuum regenerative carbon dioxide removal system, membrane evaporative cooling and rear entry ingress/egress.

2.2. Scope of New Space Suit

The new xEMU space suit will be worn on Artemis missions and its development is two-fold: (1) provide one demonstration unit, xEMU Demo, to the ISS and provide two flight units, xEMU, for the Artemis III mission. The highest priority is to deliver the flight units for the Artemis mission, however, the demonstration unit for the ISS is the current priority, with the demo anticipated for 2023 and the flight unit anticipated for 2024 (both dates have since been postponed). The goal is to perform a demonstration of functionality on the ISS to perform extravehicular activities (EVAs) so as to avoid a system integrated test on the lunar surface with the Artemis flight unit. If issues are found during the ISS EVAs, sufficient time has been scheduled for workarounds to employ any correction actions.

The Concept of Operations (Con Ops) for the lunar walk are as follows: (1) pre-deploy the xEMU flight units to Gateway, the new space station, after the crew arrives on Orion, (2) crew will retrieve, assemble and checkout the xEMU in the HLS lander while docked on Gateway, (3) crew will wear the xEMU attached to an umbilical for lunar descent and ascent for up to 12 hours to satisfy longevity walks. The mission will be to investigate the lunar south pole vicinity for a period of approximately 6 days with 5 EVAs.

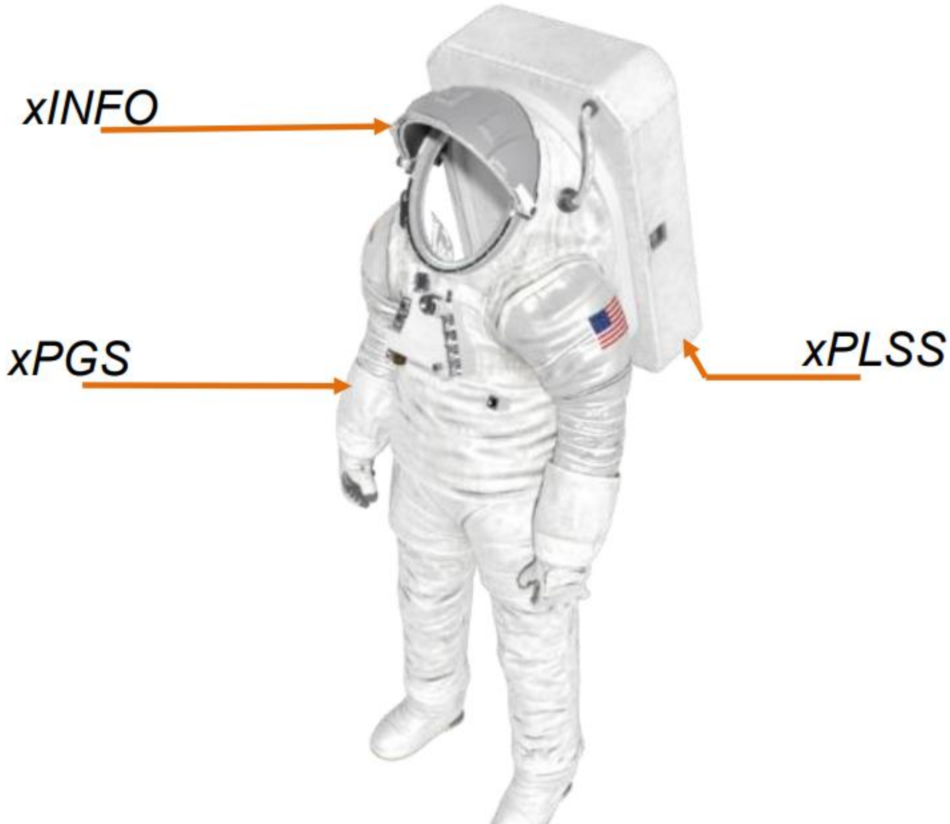


Figure 4 – xEMU Space Suit

xEMU Demo	Characteristic	xEMU
4.3 psi	**Operating Pressure**	8.2 psi
LEO Microgravity	**Design Environment**	Deep Space Microgravity Surface
EMU TMG	**Environ. Protection**	New EPG
1 size EMU LTA, arms, gloves	**xPGS Components**	Fleet of sizes Walking LTA New arms Dust resistant
EDaR HECA	**xINFO**	Upgraded lights and camera Graphical Display INFO controls
Advanced SSER	**xPLSS Capabilities**	Auto checkout Exploration Radio

Figure 5 – xEMU Demo vs. xEMU Characteristics

2.3. System: xEMU

The xEMU, the demo unit in specific, is the system level product and is the first step towards a flight xEMU. The xEMU is comprised of three complimentary systems: (1) the Exploratory Portable Life Support System (xPLSS) which maintains the life support functions for the crew in terms of oxygen, ventilation and thermal functionalities, (2) the Exploratory Pressure Garment System (xPGS) which serves as the pressurized garment to provide environmental protection and thermal interface from the xPLSS, made up of external soft goods, boots and helmet, (3) the Exploratory Informatics System (xINFO) which supports telecommunications and data processing for the entire xEMU (Todd, 2020).

2.3.1. Subsystem: xPLSS

The xPLSS has been in development for over 10 years. The base schematic and technology were chosen in 2007 after a throughout trade study that engaged the EVA community between Glenn Research Center and (GRC) and Johnson Space Center (JSC). The range of environments are as follows, with Martian environments necessitating unidentified additional components (Campbell, 2017):

- Low Earth orbit (LEO)
    - Microgravity.
    - Low radiation.
    - Vacuum ambient.
- Moon – CIS lunar
    - Partial gravity.
    - Elevated radiation.
    - Vacuum ambient.

- Mars
    - Partial gravity
    - Elevated radiation
    - Low pressure carbon dioxide (CO_2) ambient

The current scope of EMUs provides life support functions for pressurization, oxygen ventilation, removal of carbon dioxide, water and trace contaminants, and thermal control. The future scope for the xEMU includes modifications in the following areas, which together perform the critical life support functions that enable autonomous operation for a crew member separate from the pressurized habitat (Campbell, 2017):

- Primary Oxygen Loop
- Secondary Oxygen Loop
- Oxygen Ventilation Loop
- Thermal Control Loop
- Auxiliary Thermal Control Loop
- Vacuum Access Manifolds

The following loops and manifold can be simplified into three sub-subsystems: the oxygen loop, the ventilation loop and thermal loop.

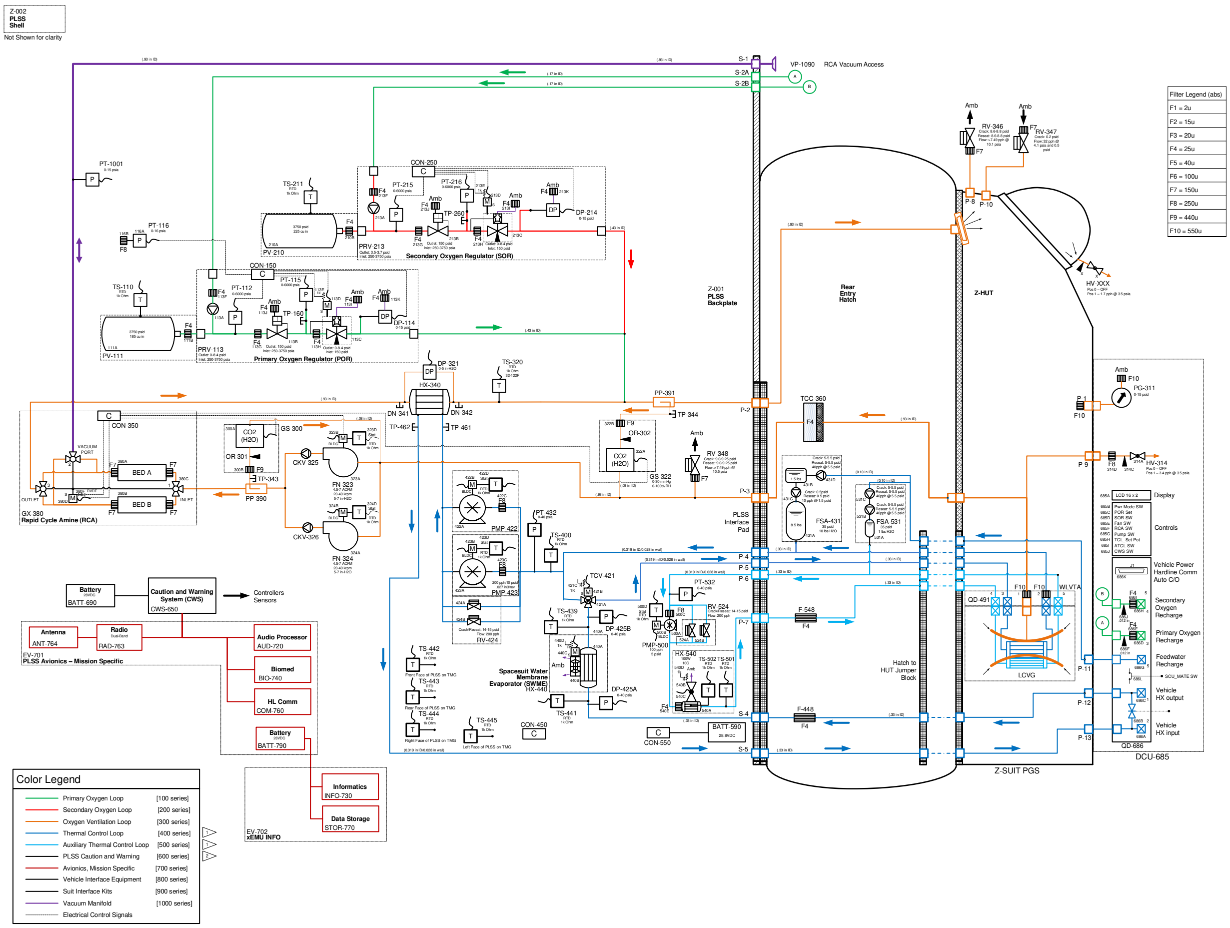


Figure 6 – xPLSS Piping & Instrumentation Diagram (Used with permission from Campbell, 2017)

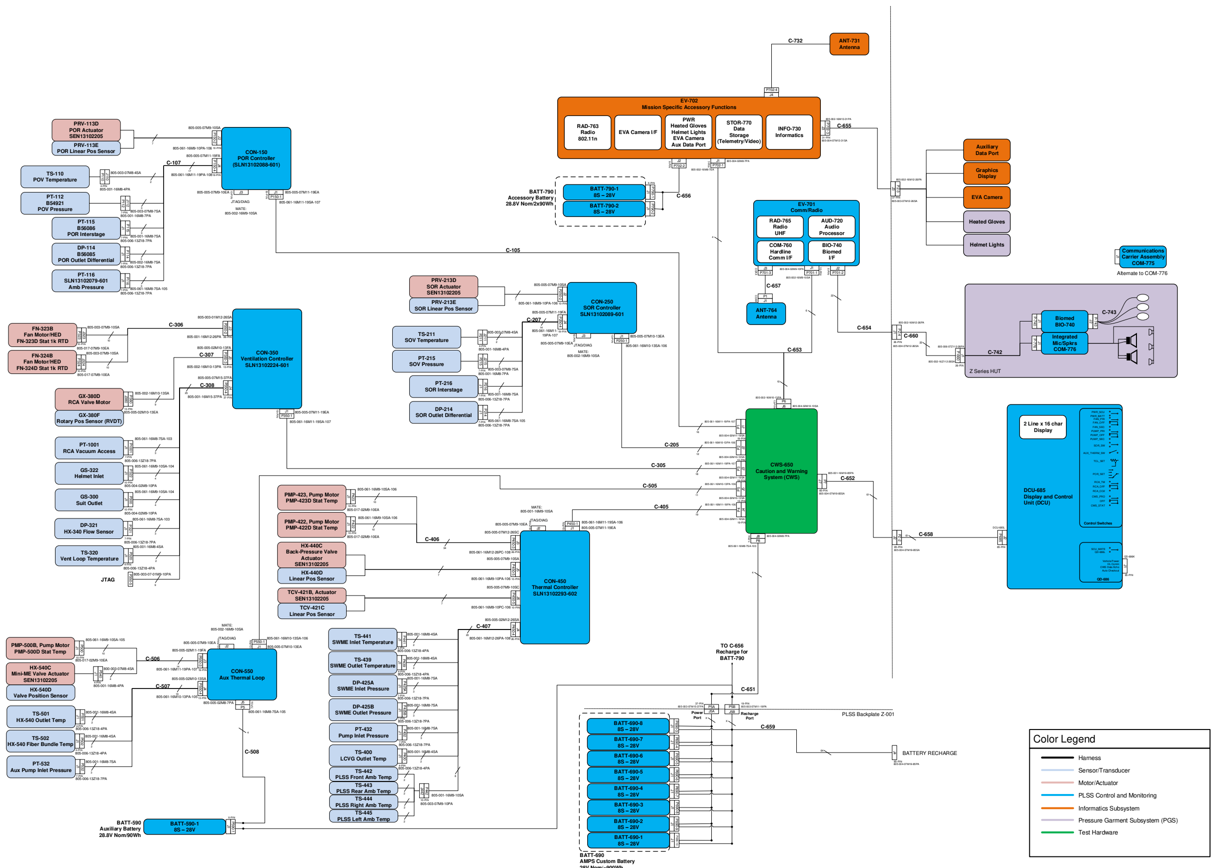


Figure 7 – xPLSS Electrical Block Diagram Schematic (Used with permission from Campbell, 2017)

2.3.1.1. Oxygen Loop

The oxygen loop is comprised of a primary loop and a secondary loop for redundancy. For all intents, both loops are identical. The loops provide oxygen to the xPLSS for pressurization, leakage make-up and metabolic consumption. Additionally, the loops provide sufficient gas flowrates to support denitrogenation purge. The two main components of the loop are the Primary Oxygen Regulator (POR) and the Primary Oxygen Vessel (POV); the secondary loop has complimentary components in the Secondary Oxygen Regulator (SOR) and the Secondary Oxygen Vessel (SOV). The POV stores usable oxygen, which is then fed to the POR, which is preset based on historical operating pressures. The specific set points are dictated by motor controllers, CON-150 for the primary loop and CON-250 for the secondary loop. These set points are managed by the astronaut via the Display and Control Unit (DCU), serves as the primary crew interface for controlling the xEMU space suit. It is centrally located on the front of the torso of the space suit and is the hub for resupplying the space suit with consumables including oxygen and water (Davis, 2019). Check valve allows for protection against reverse leakage. To monitor the behavior of the loops sensors, including the resistance temperature devices (RTDs) and pressure transducers (PTs) are established along the inlets and outlets of the line. The controllers (CON-150/250) monitor the sensors and send telemetry to the Caution and Warning System (CWS), which acts as the central nervous system for the space suit and control the critical life functions via fault detection across all controllers the inputs from DCU and outputs of mechanical-electrical behavior to the DCU LCD. Filters are also included to maintain cleanliness in the lines.

2.3.1.2. Ventilation Loop

The ventilation loop contains components for carbon dioxide washout, fluid movement through the lines, trace contaminant removal, humidity control and PGS inlet gas temperature control. A centrifugal fan moves air at variable flow rates, based on historical operating fan parameters. The loop contains too fans, both identical with one active at a time with the other serving as a redundant device should the primary fan fail. Gas sensors are on either end of the Rapid Cycle Amine (RCA) swing bed to monitor carbon dioxide levels. The RCA is the hallmark of the ventilation loop and a novel technology for EMUs. The RCA utilizes a proprietary formulation of amine beads which operate in a two-fold capacity based on the two-bed configuration.

While one bed is open to the ventilation line, adsorbing carbon dioxide and water from the loop, the other bed desorbs the carbon dioxide and water to vacuum, thus regenerating the amine beads. The oxygen loop ties into the ventilation loop when make-up oxygen is required. The ventilation loop controller, the CON-350, communicates to a rotary stepper motor as a function of the carbon dioxide readings in the suit to switch a rotary valve from one amine bed to another. The DCU also serves as a peripheral to control the swingy bed activity within the xPLSS. The ventilation gas travels to a sensible heat exchanger, HX-340, where heat was added by the fan and translates that heat to the thermal loop. To monitor the behavior of the loops are sensors, including the RTDs and PTs are established along the inlets and outlets of the line.

2.3.1.3. Thermal Loop

The primary thermal control for the space suit is responsible in the thermal control loop. This loop is designed to accommodate electronic and metabolic waste heat and environmental heat leak. A series of lines pass through the PGS liquid ventilation garment system (LVGS) to regulate the body temperature of the astronaut. A feedwater supply assembly (FSA) uses suit ventilation loop pressure to compress the bladder, which pressurizes the loop fluid to the same pressure as the ventilation loop pressure. Redundant pressurization for the thermal loop can elicit activation of the POR or the SOR, which indicates that a loss of primary oxygen supply does not necessitate an open-loop abort as the secondary oxygen supply is capable of supporting the operations of the thermal loop (Campbell, 2017). The usage of water is tracked by the CWS from the spacesuit water membrane evaporator (SWME), a heat-exchanger and de-gasser in the thermal loop. The SWME removes heat with the passage of fluid and de-gasses air that might propagate in the line. The SWME uses hollow fiber technology developed by 3M to perform the de-gassing and heat-exchanging as water passes through the hollow fibers, release heat through an exothermic reaction and degasses as fluid travels passively through the hollow fibers (Makinen, 2014). In similar design to the oxygen and ventilation loops, the thermal loop has an auxiliary fan, feed water supply and miniature version of the SWME in the event the primary loop fails. Additionally, in similar design to the oxygen and ventilation loops, the DCU and CWS allows for control of the subprocesses of the ventilation loop with controllers driving the functionality of the SWME, and miniature SWME in the CON-450 and CON-550, respectively. To monitor the behavior of the loops are sensors, including the RTDs and PTs are established along the inlets and outlets of the line.

2.3.2. Subsystem: xPGS

The xPGS subsystem is unitized into several component-level parts and as the focus of the study is primarily directed towards the xPLSS, only pertinent components to provide context will be identified. The hard upper torso (HUT) is the rigid portion of the suit that covers the crew member's upper torso. The core function of the HUT is to allow the astronaut to work efficiently and safely in space. The hatch is the pressurized compartment serving as the interface between the xPGS and xPLSS that serves as the cover to the portal to enter and exit the suit. The hatch also serves as the inlet and outlet for breathing air and cooling water to enter the space suit. The helmet is designed to incorporate a hemi-ellipsoid shape and includes a protective extravehicular visor assembly (EVVA). Both the helmet and visor are made of polycarbonate to protect against UV radiation and is coated with a scratch resistant hard coating. The integrated communication system (ICS) supports headset-free auditory communication via (3) microphones and (2) speakers. The arm sub-assembly includes the shoulders, lower arm elements and gloves, which are the legacy units used on the EMU. The lower torso assembly contains the boots, legs, waist and thigh assembly, which are the legacy units used on the EMU. The liquid colling and ventilation garment (LCVG) contains primary and auxiliary loop to interface with the xPLSS to regulate the thermal conditions of the suited crew member (Ross, 2019). The environmental protection garment (EPA), a novel technology for next generation space suit development, is used for lunar dust mitigation. The maximum absorbency garment (MAG) is a waste contaminant garment to be worn under the LCVG (Davis, 2019).

2.3.3. Subsystem: xINFO

The xINFO subsystem is unitized into several component-level parts and as the focus of the study is primarily directed towards the xPLSS, only pertinent components to provide context will be identified. The purpose of the xINFO subsystem is to reduce the cognitive load on the crew member when processing data during an EVA by establishing non-critical avionics to support superior EVAs with regards to efficiency and effectiveness. The primary mission is to collect data and its conversion to useful and actionable information by communication to the crew member during an EVA, with the equipment existing on or around the ellipsoid bubble helmet. Components include radio and antenna for communication and a graphical user display to include procedures, high-definition camera video and images, xPLSS consumables, system health and status, messages from ground and warning messages from other crew members. High resolution cameras allow for live feeds of critical science imagery and supports the EVA crew and flight controllers' awareness situationally to promote the ability for verification of task completion and troubleshooting during crew member EVAs. Lights allow for illumination of worksites and light settings are optimized for variable conditions both in microgravity and the lunar surface. Guidance and navigation tools also promote situational awareness. xINFO additionally provides video, images and audio store on the suit to allow for EVA information retention.

2.3.4. Forms of Flight Hardware

The following descriptions for flight hardware are extracted directly from NASA document Johnson Space Center 1281.10A.

- **Class I equipment.** Equipment acceptable for space flight use. (Controlled flight equipment)
- **Class II equipment.** Equipment acceptable for use in ground tests or training in a hazardous environment. (Controlled non-flight equipment).
- **Class III equipment.** Equipment acceptable for non-hazardous training or display purposes. (Non-controlled non-flight equipment)
- **Class IIIW equipment.** Equipment acceptable for use in Water Immersion training in a hazardous environment. (Controlled non-flight equipment)
- **Class 1E equipment.** Class 1E is for experimental hardware

and some commercial off-the-shelf hardware.

- **Ground Support Equipment (GSE).** Non-flight equipment designed and certified with a physical and/or functional interface with flight hardware that is required for the handling, servicing, inspection, testing, maintenance, alignment, adjustment, checkout, repair or overhaul of Class I or Class II products. (Controlled non-flight equipment)
- **Special Test Equipment/Devices (STE/D).** Special Test Equipment/Devices (STE/D). STE/D are similar in function to GSE but are not controlled until time of use. This equipment may be used in support of class I, II, IIIW and GSE checkout and service in limited cases. (Controlled non-flight equipment, see NT-CWI-001 for additional requirements). Note: Items with a flight part number cannot be classified as STE/D.
- **Non-flight equipment.** Equipment used to aid in the processing, maintaining, testing, repairing, etc., of the flight equipment and all its systems. Non-flight equipment is comprised of GSE, Class II, III, IIIW, commercial tools, special test equipment, special test devices, and element tools. (Non-controlled non-flight equipment).

2.4. Project & Resource Breakdown

Stakeholders across xEMU are listed, including performing organization, the customer and all pertinent, peripheral stakeholders. A brief overview of the scope, cost and schedule is also presented.

2.4.1. NASA Customer

NASA is the customer in this case study and is responsible for several unique technological and scientific milestones in aeronautics, space flight, applications and sciences. The organization was first established in 1958 in response to Soviet Union space exploration (Sputnik and post-Sputnik) and over the last 60 years, NASA has continued to forward the frontier of state-of-the-art aeronautics research that has drastically changed the manner in which we live. To specify only one area of focus, NASA has been developing the xEMU system and subsystems since 2007 but rely on contracting agencies for procurement of hardware, testing, quality assurance and control and in certain cases, designs for niche components found on the xEMU.

2.4.2. JETS Performing Organization

Jacobs Technology is an engineering contractor supporting a wide range of projects and operations for NASA under the JETS contract and is the performing organization in this case study. JETS develops and sustains flight and ground supporting hardware in addition to software for the human spaceflight program. Engineers on the contract perform flight systems analyses that include navigation & guidance and control simulations for current and in-process engineering projects. The engineering services supported include thermal, stress, vibration and loads testing, failure analyses, data processing and communications. The service contract additionally provides mission-critical services to ISS, which include sustainment of multiple ISS hardware and software platforms such as exercise equipment, ISS robotic arms for visiting vehicle capture, habitation systems, cameras, robotic systems for berthing and docking spacecraft, avionics and instrumentation, extravehicular activity tool development (EVA) and Capsule Parachute Assembly System (CPAS) development. In addition to these projects and services, xEMU has been a major project in the collection of projects under the JETS contract, currently employing nearly 300 employees of the approximately 1,400 currently on the contract.

2.4.3. Organizational & Work Breakdown Structure

2.4.3.1. NASA Project Management

NASA project management in the xEMU organization is primarily reserved for the system level of xEMU and subsystem levels of xPLSS, xPGS and xINFO. Each of the four areas has a project manager and at times a designated deputy project manager. These NASA civil servants are responsible for allocating the scope to the JETS project managers and approving their contractual budgets and schedule while supervising their progress through periodic schedule and budget meetings. Project

2.4.3.2. ISS, Gateway, Human Landing Systems

ISS, Gateway and the HLS are the three entities that serves as the most pertinent and influential stakeholders to the xEMU project. Although NASA is the customer of the JETS contract, the system and subsystem requirements are derived from these offices and thus drive the requirements in terms of scope. The budget and schedule are byproducts of the government funding and as a result of contractual agreements with performing organizations outside of direct federal control, specifically JETS, are given a schedule and budgetary expectations to deliver a series of suits within a specific time and cost. Gateway, the proposed outpost to orbit the moon and vital part of NASA's future deep space program, the ISS, the current space station currently in orbit and the HLS, the complete system to take crew from lunar orbit to the lunar surface, are all stakeholders that have been elicited information to help complete the proper scope, cost and schedule for xEMU.

2.4.3.3. EVA Office, xEVA Project

The EVA office is responsible for serving as the program management authority under the NASA umbrella and as such is charged with supporting final review and approval for all areas of EVA, including the EMU and xEMU. The Exploratory EVA (xEVA) project is the new transition from xEMU to the xEVAS contract that is the equivalent program management authority as the EVA office is to the EMU and xEMU.

2.4.3.4. JETS Project Management

JETS project management is responsible for executing the customer's scope of work with the intended level of quality to the contractually agreed to budget and schedule. These managers do not exist on one specific organizational level but are convoluted throughout the organization, participating in all of the subsystems, systems engineering group and safety and mission assurance departments. The project managers primary functions are to serve as a conduit between the customer and development team and secure the product within cost, schedule and budget and within quality expectation while properly assessing risk throughout the project lifecycle. In that capacity, each project manager works on sub-projects, with the portfolio of projects existing as the xEMU total project scope. These subprojects include and are not limited to testing, analysis, design and systems engineering. JETS project management, in particular project management on the GSE test and design team will be a focal point of the case study.

2.4.3.5. xEMU SE&I Group (at a glance)

The Systems Engineering & Integration (SE&I) Group is responsible for project support, architecture integration, requirements & verification, analysis integration, flight operations, manufacturing, software integration and assembly, integration and testing. This group is made up of both NASA civil servants and JETS employees. The Systems Engineering Management Plan (SEMP) internal document CTSD-ADV-1495 documents the technical approaches for organizing the resources, products and processes to accomplish the maturation and design definition of the project requirements. The document serves as the bridge between project management and the technical team. The group will further be decomposed in *Section 3.1 xEMU SE&I Group* as their involvement in xEMU will be one of the major focal points of this case study.

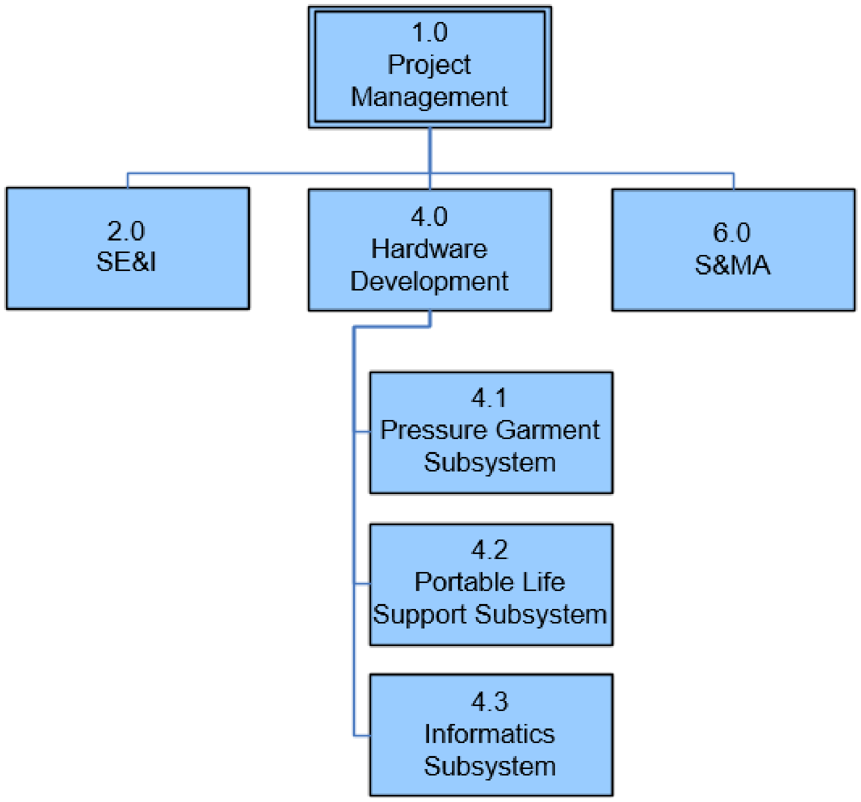


Figure 8 – xEMU Project Management and Systems Engineering Organizational Chart

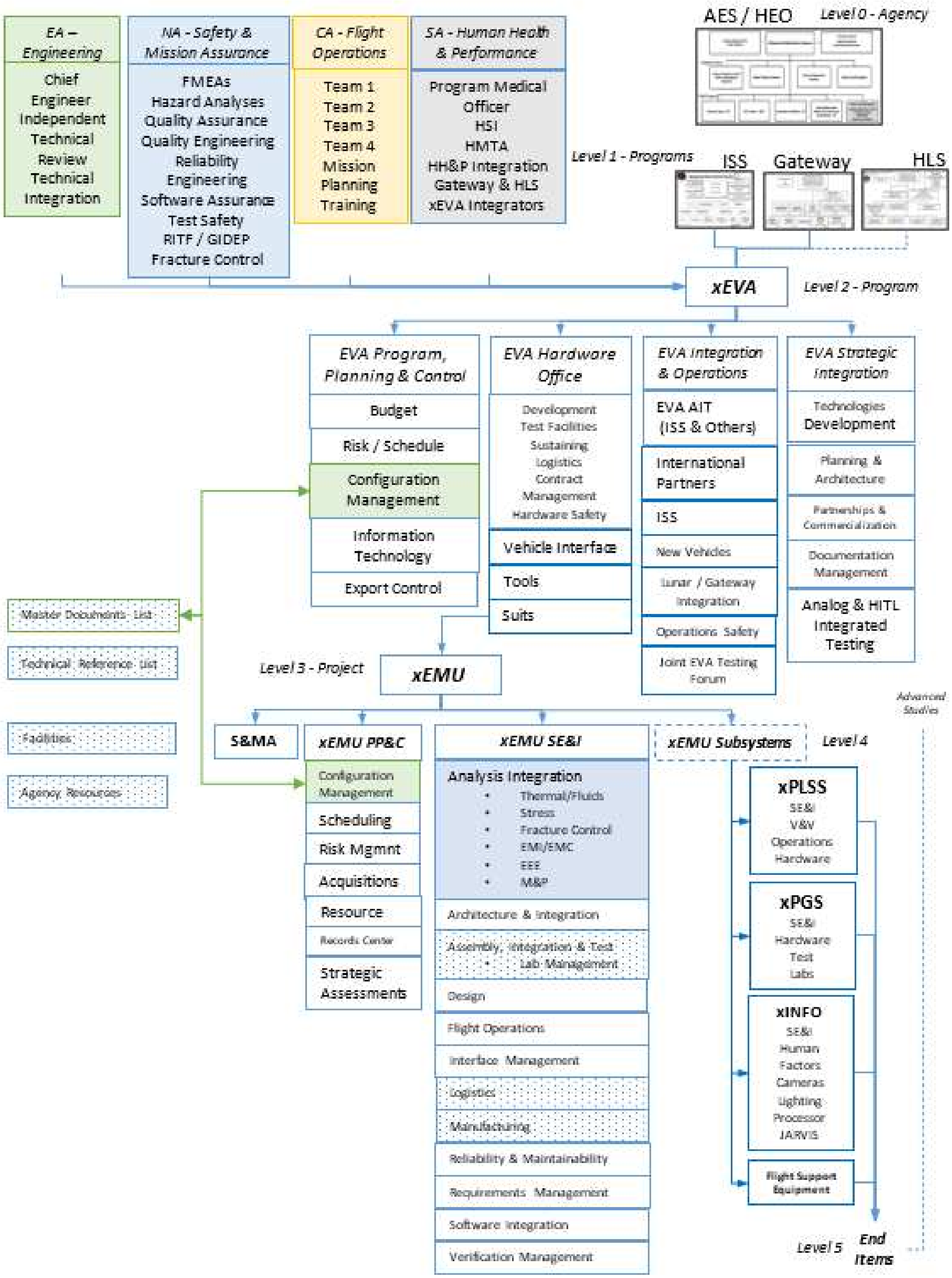


Figure 9 – xEMU Stakeholder Group Cross-Functional Diagram

2.4.3.6. Component Owners

Component owners are a collection of NASA civil servants and JETS employees that are responsible for the management of their components, as each component constitutes the collection of products for each of the subsystems. There are 16 components that comprise the xPGS subsystem, 5 components that comprise the xINFO subsystem and 44 components that comprise the xPLSS subsystem. These include mechanical, electrical and structural components for all subsystems.

2.4.3.7. Verification & Validation

Component owners and/or systems engineers are responsible for assigning verification and validation matrices in the specification and in the system engineering requirements database. These verifications and validations should follow International Council of Systems Engineering (INCOSE) standards and be agreed upon by the pertinent stakeholders and approved by configuration management at the systems engineering and integration forums. Verification will support the performing organization and their best practices when checking systems requirements to verify they have been written correctly, assure the performing organization resource groups have applied best design practices and analysis in design during the design and analysis phase and assure that the product was tested against the appropriate parameters. In comparison with verification, the customer will validate along the project lifecycle if at the systems requirements phase if the team is building the correct product, during the design and analysis phase if the product is being built correctly and in the product release phase validate that the performing organization has built the right product.

2.4.3.8. Lab Operations

The lab operations team consists of technicians that fabricate and maintain hardware, including mechanical and electrical testing stands. These tests stand perform developmental and qualification tests at a component level, subsystem level and system level for the xEMU. The GSE design and test team will be a major focus of the case study.

2.4.3.9. Hardware Management

Hardware is managed at a subsystem level (xPLSS, xPGS, xINFO), with a hardware lead directing the design, analysis, fabrication and risk mitigation efforts. Subgroups managed by the hardware lead include and are not limited to or include all of the following for each hardware subsystem lead team: component owners, design, stress analysis, thermal analysis, integration engineering, project engineering.

2.4.3.10. Project Planning & Controls

Project Planning & Controls (PP&C) is the branch responsible for resource management, strategic assessments, information management, risk management, records collection, public relations, configuration and data management and schedule management across xEMU.

2.4.3.11. Safety & Mission Assurance

Safety & Mission Assurance (S&MA) covers primarily quality facets including quality assurance and quality engineering. Additionally, S&MA software assurance, test safety, reliability assurance, Failure Modes and Effects Analysis (FMEA) and hazards analysis.

2.4.3.12. Risk Management

Risk management on xEMU is designed to ensure that risks are identified, classified and effectively managed. Risks are managed at the subsystem level or project level. If a risk cannot be properly managed, it is elevated to the xEVA office and it that risk cannot be properly managed at the office level, it may be elevated to the Gateway, ISS or HLS offices. Risks are identified by any stakeholders but are primarily managed by project management. A risk management tool is used on xEMU to link, track, manage and qualify risks by grading them according to severity. A risk owner will continuously via a continuous risk management process throughout the project lifecycle. The risk management department defines the practices and owns the risk management plan for xEMU.

2.5. Schedule

The schedule below defines the intended "Boots to Moon" initiative wherein the xEMU project will produce three different deliverables: a Design Verification Testing (DVT) spacesuit unit, a Qualification spacesuit unit and a Flight spacesuit unit The first schedule defines the original planning without the need for a lunar suit. The second schedule defines the accelerated schedule to enable a flight demonstration in FY23.

FY18	FY19	FY20	FY21	FY22	FY23	FY24	FY25
SRR (Jan)	PDR	xEMU dSRR xEMU dPDR		CDR			SAR & Delivery
DVT Build/Assy			DVT Testing	Qual & Flight HW Build		Qual Testing	Acceptance Testing

Terms and Definitions: SRR - System Requirements Review, PDR - Preliminary Design Review, CDR - Critical Design Review, DVT - Design Verification Testing, SAR - Systems Acceptance Review

Figure 10 – Original Delivery Schedule

<table>
<tr><th>FY18</th><th>FY19</th><th>FY20</th><th>FY21</th><th>FY22</th><th>FY23</th></tr>
<tr><td>SRR (Jan)</td><td>PDR</td><td></td><td>CDR</td><td></td><td>SAR & Delivery</td></tr>
<tr><td colspan="2" rowspan="2">DVT Build/Assy</td><td rowspan="2">DVT Testing</td><td colspan="2">Qual & Flight HW Build</td><td rowspan="2">Acceptance Testing</td></tr>
<tr><td></td><td>QualTesting</td></tr>
</table>

Figure 11 – Accelerated Delivery Schedule

2.5.1. Schedule: Design Verification Testing

xEMU testing will be done two-fold with the first testing approaches being that of DVT testing. DVT will be done in order to verify the functional build using Class III hardware both on the test station side and equivalent flight component hardware that is of Class III pedigree. This testing could be thought of as pre-qualification testing with the deliverable being a developmental unit. The ability to build hardware more rapidly than GSE allows for proof of concept and troubleshooting before development of the Qualification units.

2.5.2. Schedule: Qualification Testing & Flight Units

Following (and also fast-tracked alongside) DVT is Qualification Testing. The approach will be to implement the lessons learned from DVT and apply those to the spacesuit units for Qualification, which will verify the corrections and updates after DVT and promote the Flight spacesuit unit build to be tested on the ISS and ultimately tested and utilized on the lunar surface.

2.6. Cost

The cost of the project incorporates the genesis of the xEMU product, dating back to 2017. Between the inception of the project in 2017 to the mid-third quarter of 2021, the total cost of xEMU has risen to approximately $420 million with the anticipation that the continuation of xEMU under xEVAS will anticipate costs of upwards to $1 billion by the year 2025 (Martin, 2021). With respect to the Jacobs GSE projects, those would account for roughly $4 million dollars of that $420 million value.

3. CHAPTER 3: SYSTEMS ENGINEERING DEPARTMENTS

Lede: This chapter outlines the systems engineering departments. Two primary systems engineering groups exist within this dissertation. The first is an external group specific to xEMU with members of both JETS and NASA employees whereas the second is an internal group specific to JETS organized across the JETS contract.

3.1. xEMU SE&I Group

The SE&I team is the system technical integrator for the xEMU project. In an effort to effectively affect thorough and consistent architecture integration by the project, the SE&I team leads and organizes the xEMU project groups in project requirements development, architecture development, planning product development, and in cross-system integration, assessment, test, verification and validation. The lead of the SE&I group reports directly to the xEMU Project Manager and Subsystem Managers, thus supporting integration with internal and external organizations. The SE&I team is organized in the chart below:

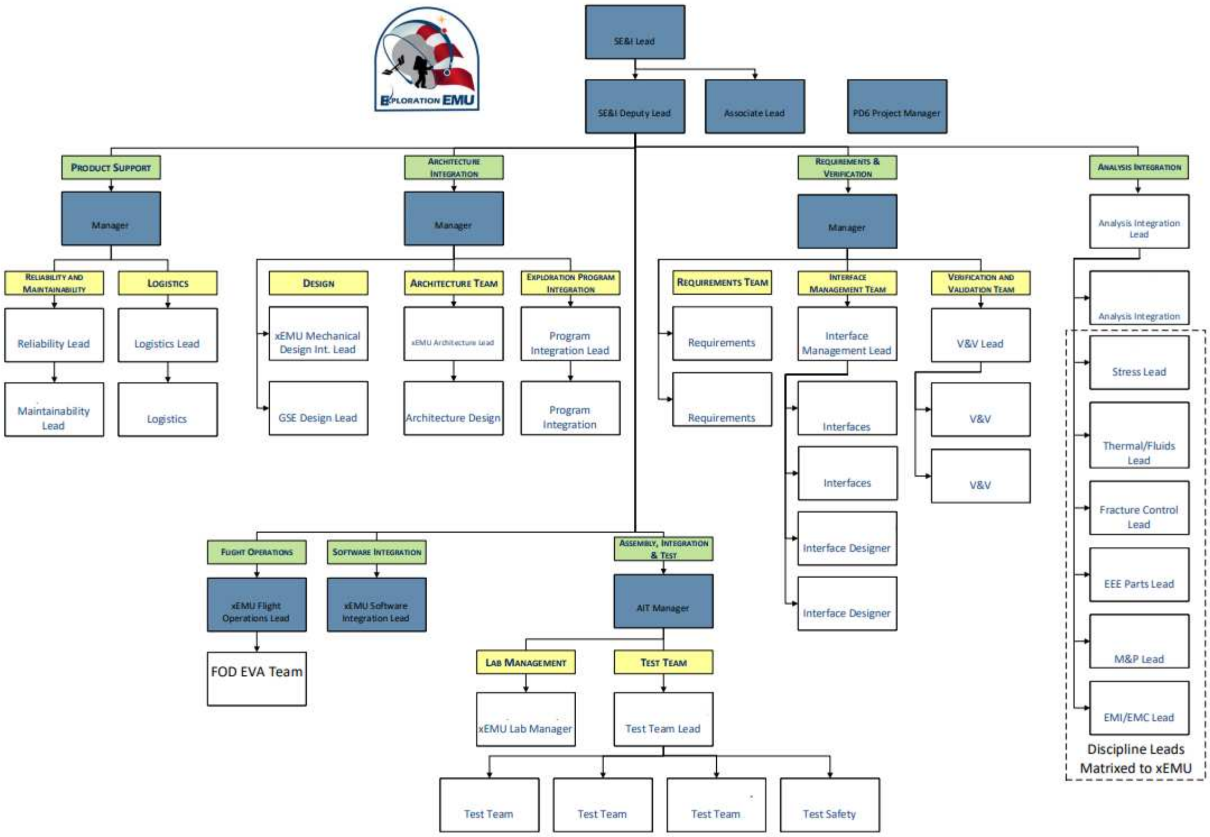


Figure 12 – SE&I xEMU Organizational Chart

3.2. Jacobs Systems Engineering Department

Jacobs Technology retains its own systems engineering department separate to xEMU that supports a variety of projects within the JETS umbrella of scope. While there is no dedicated overall organizational chart, each project will have a supporting systems engineering group in a similar spirit to the of the SE&I group for the xEMU project. JETS systems engineers are aligned with the groups they are dedicated to, which are primarily the product development groups and software services groups.

4. CHAPTER 4: APPLICABLE SYSTEMS ENGINEERING CONCEPTS

Lede: This chapter comprises the background research sections which helped established the current state-of-the-art with the intention of uniting the current state-of-the-project. This chapter will focus on applicable systems engineering concepts that support (i) project life cycle development with regards to traditional, waterfall development and other lean and agile strategies and (ii) requirements engineering strategies.

.

4.1. Systems Engineering Models

The two approaches utilized across xEMU revolved around a more streamlined method of systems engineering documentation with regards to more rudimentary and less streamlined forms, both of which are utilized by virtue of their importance and relevance to the challenge at hand. These are Document Based Systems Engineering (DBSE) and Model Based Systems Engineering (MBSE).

4.2. Managing Requirements with Document Based Systems Engineering

DBSE is the practice of engineers creating specifications for a system using various documents. Many of these are found within the xEMU project and contain but are not limited to Con-Ops, scenarios, system/subsystem/component requirements, architecture description documents and verification and validation matrices. Historically, this practice is done by different teams or individuals with the intent of populating the aforementioned documents. These tasks as they are done separately and autonomously with variation amongst teams and individuals may lead to contradictions, overlaps, inconsistencies, knowledge gaps and the like. With the advent of complex computing systems and emerging technologies leading to advanced product development, the need for a more sophisticated model arose (Delligatti, 2013). An example in practice would follow a system architect making a tenth-iteration design change a single component in a system hierarchy into multiple components to help satisfy stakeholders concerns by creating separate and differentiable atomic requirements. The system architect then makes an autonomous decision to create a new name for the components to illustrate the narrower focus. In order to complete this task, the architect must then do a search on every applicable document. This will require a thorough search into the entire system database and manual changes to all applicable artifacts. This effort presents many challenges, key among them the time required to make the changes, find every applicable document to make the changes, but most importantly what would be the downstream effects which are not clearly discernable using a DBSE approach. On a long enough project timeline, this approach will lead to slippages in schedule due to increased errors found in documentation and the time to fix them appropriately.

4.2.1. Managing Requirements with Model Based Systems Engineering

To properly streamline and capitalize on the efforts of DBSE, MBSE was developed as a method of utilizing a standard system model where teams or individuals may perform the identical activities to generate documents such as Con-Ops, scenarios, system/subsystem/component requirements, architecture description documents and verification and validation matrices with the intent that efforts and errors are minimized. In contrast to the architect using the DBSE in the previous example, the architect is able to rename the newer components to differentiate the transition in requirement definition. As a result of MBSE practices, the modeling tool utilized by the architect automatically enforces changes instantly to all pertinent diagrams where the block of information appears regardless of the size of the architecture. As the model has a change initiated, so do the accompanying diagrams. Artifacts will also exist to document and connect the change for traceability. In this approach, the inconsistencies and the erroneous nature indicative of the DBSE approach are not afforded the opportunity. The MBSE approach thusly promotes quality throughout the system architecture and supports the idea of the cheapest defects are the ones that can be prevented. This artifact is known as the system model (Delligatti, 2013).

4.2.2. Three Pillars of MBSE

The three pillars of MBSE are a modeling tool, a modeling language and a modeling method (Delligatti, 2013). The Unified Modeling Language (UML) defines the genesis of graphical language to help construct, visualize and document information for complex systems, with its genesis founded in the usage of software systems. Where human language may create ambiguity, modeling language, in specific System Modeling Language (SysML), can help support standardization for communication by avoiding ambiguity with rules to modeling elements and frameworks for meanings in relationships (Hause, 2006). This presents the first pillar of MBSE: the modeling language. The second pillar of a modeling method, which establishes a method on how a language would be executed to support the description of a system architecture. These specific methods of modeling are developed to address why the model is to be used and establishes a set of frameworks for a team to address the challenge the model is to solve. The third pillar of a modeling tool, which are specialized tools designed to execute and enforce the rule of the modeling language. These differ from the diagramming tools as their as an intended meaning the data is meant to generate via the modeling tool (Borky, et al, 2018).

4.2.3. Transformation to SysML

SysML finds its roots in Unified Modeling Language (UML), a systems engineering language meant to disambiguate by visualizing information in a graphical context. This language was developed in the 1990s as a method to organize software-intensive development yet finds its place in various product developments (Hilken, et al, 2020). UML addresses four specific challenges with the usage of models to streamline and organize data. Models allow engineers to (1) graphically visualize a system or by visualization of a current vs. desired state, (2) allow for specification of structure and/or behavior of the system, (3) a template in which guidance is given to the architecture of a system and (4) documentation on the decisions made by the engineers regarding the product/project (Jacobson, et al, 2021). To deal with the needs of systems engineers more adequately as the complexity of the product and documentation throughout the project lifecycle and field of study, SysML was developed and found its use in more hardware/software-intensive systems in contrast to UML which found its utility in more software-intensive systems (Höglund, 2017). SysML can more effectively deal with requirements, by linking model elements; system structure, by enhancing scope of the modeled architecture structure by expansion of benchmarked UML with flows and ports; functional behaviors, by providing the activity model with the capability of characterizing the behavior of a system mathematical; parametric modeling, the ability to streamline calculations across different elements and behaviors within the model; allocations, by creating tests and requirement conditions relative to the behavior of the structure of the system; trade studies, by which the architecture model can benchmark against best practices (Borky, et al, 2018). SysML grammar and structural notions are defined by the

Object Management Group, Inc. (OMG) and specified by *OMG System Modeling Language* as a reference guide to the practices of SysML, analogous to INCOSE providing requirements guidance (Soley, 2013).

4.2.4. Types of Model Diagrams

Hause defined nine separate SysML model diagrams with each corresponding to one of the four pillars of SysML, which are structure, parametrics, requirements and behaviors (Hause, 2006). For the purposes of this case study, the focus will follow the development of Behavior Diagrams (BD) however an examination of all diagram types is listed for comparison and will be used at a minimal capacity when compared to their associated model diagram counterparts.

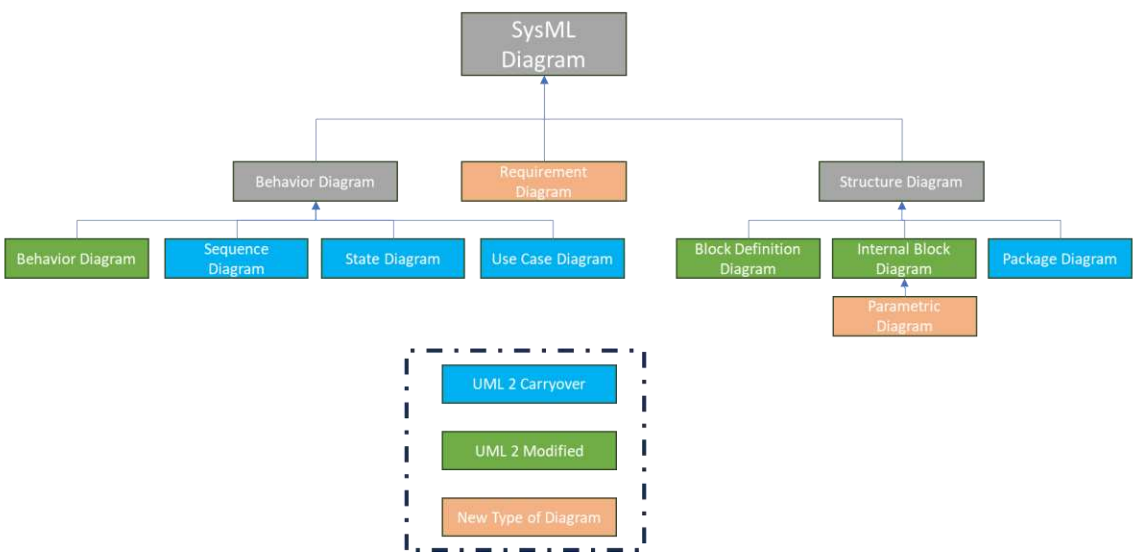


Figure 13 – Model Diagrams Type

4.2.4.1. *Structure Diagrams*

Block Definition Diagrams (BDD) are used to characterize hierarchy and classification trees and are utilized to convey elements such as value types and blocks. These elements define different types of objects that exist in an operational system. Internal Block Diagrams (IBD) illustrate connections between internal parts of the system (i.e., blocks) and their interfaces. Package Diagrams (PKD) shows organization within a form of package containment hierarchy and illustrate how model elements contain dependencies between packages and model elements. (Hause, 2006).

4.2.4.2. *Parametric Diagrams*

Parametric Diagrams (PD) are used primarily to identify constraints including any inequalities that affect the properties of the system. These diagrams are used to but are not limited to supporting reliability, affordability and performance analyses. Trade study support using these diagrams may include candidate physical architecture development (Hause, 2006).

4.2.4.3. Requirement Diagrams

Requirement Diagrams (RD) illustrate requirements in a tabular method. These diagrams follow closely the INCOSE standards of defining attributes by including identities such as name, identification number, source traceability, etc. To support traceability, these documents may link across all various allocated and decomposed requirements from high-level system to low-level component requirements (Hause, 2006).

4.2.4.4. *Behavior Diagrams*

State Machine Diagrams (STM) define how an entity changes its state not only as a function of its current state input but its dependence on particular history inputs. This diagram helps to specify a behavior with a particular focus on an array of states of a block and the potential transition between those states as a function of event occurrences. Activity Diagrams (AD) focus on a behavior with a flow of control of activities. This tool is used to analyze and express the differentiation between the current and the desired behavior. This aids in the modeling of workflow. Sequence Diagrams (SD) detail the flow of activities. The main focus is the interaction of objects over a specific time period and support the visualization of sequences to perform a specific piece of functionality. Use Case Diagrams (UCD) are used to illustrate how actors both external and internal to the system provoke and drive the high-level functionality (Hause, 2002). These diagrams reflect the system from an actor's perspective and interaction between the different actors. To facilitate simplicity, these BDs typically avoid technical jargon and using natural language of the actors. Use cases are used with one or more complex flows. Use cases reflect functional requirements in a method that is visual when compared to written text, represents the satisfaction or lack thereof of a goal, records paths (i.e., scenarios) that traverse an actor in the system that could trigger various scenario events (i.e., main, alternative, exception). In contrast, use cases do not specify user interface or action detail, merely specify the intent. Use cases as a result of their implementation capture system requirements, validate designs and proper implementations for testing and quality assurance and serve as a preliminary framework. Use cases are ideal for system boundary management, exploring scope and managing complexity. Identification of the use case's completion is paramount, by investigating conditions, goals that are to be addressed, identification of procedural or requirement augmentations, subtractions or additions, the existence of unidentified actors, identification if all use cases have been identified and stakeholder elicitation to validate the completion of the cases (Metzger, et al, 2014). Common mistakes when incorporating use cases into a project include underutilization of important requirements, clarify or vagueness in cases, non-functional inclusion, excessive usage of “extends” and “includes” during case study development, fastidiousness on details unnecessary to the proper development of the case studies, improper involvement or lackthereof as it pertains to SMEs, failure to verify/validate use cases, too few use cases, inattention to business rule definition and capture of incorrect use cases (Gottesdiener, 2003).

4.3. Traditional Waterfall vs. Agile Lifecycle Approaches

Beck, Beedle, Bennekum, Cockburn, Cunning-ham and Fowler developed the Agile Manifesto, which serves as the foundation for the transformation for one of the engineering branches at Jacobs Technology to an agile framework for this study. The manifesto contains four values and twelve principles. The four values are: individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation and responding to change over following a plan. Although all items are valued, the former in the pairwise set of ideas are valued more. The twelve principles are summarized as follows: customer satisfaction, welcoming change, delivering frequently, working together, building projects, face-to-face time, measure of progress, sustainable development, continuous attention, keeping it simple, organized teams and reflection of effectiveness (Beck et al, 2001).

Following the understanding of the agile concept, is the selection of the agile methodology to be applied. Cohen, Lindvall and Costa include a condensed illustration of some of the more familiar frameworks included in agile and applications to determine fitness for use. Although many of the methods included in this body have their foundations in software, many of the principles are applicable to any subset of cross-functional industries. The principles reviewed include Extreme Programming, Scrum, Crystal Methods, Feature Driven Development, Lean Development and Dynamic Systems Development. The conclusion of the prevailing study assessed the viability of each principle by its constituents evaluated against those needs best suited for an entity's shift to the framework. In particular, the Scrum framework dictates an overwhelming 56% of all agile frameworks. These findings indicate that the metrics for success of a scrum approach revolve around organizations that are capable of self-developing teams which are within three to nine individuals with quick response to change and short iteration of delivery to satisfy customer deliver and keep stakeholders consistently engaged with incremental contact (Cohen et al, 2004).

4.4. Scrum

As defined by Schwaber and Sutherland, scrum is a process framework that exists within the agile mindset. The framework is a lightweight, simple in nature but difficult to master alternative. As is evident by the bevy of agile alternatives, scrum finds its foundation in software. The Scrum theory is founded upon the empirical process theory of control. This assertation is that knowledge comes from incurred experience and the ability to execute based on what is known. The approach is both incremental and iterative, which allows to optimize risk mitigations and control the predictability. The empirical process is transparent, adaptable and inspectable. Transparency refers to the visibility across all stakeholders, which allows for optimal communication. Scrum, although having its foundation in software, has utility that has been proven in dealing with traditional waterfall impracticalities across industries (Schwaber et al, 2017).

Scrum is an agile methodology which is illustrated by the three pillars of visibility, adaptation and inspection. Scrum is a highly iterative framework, consisting of activities repeated in what are known as sprints. These iterative sprints deliver value to the customer in increments. The Product Owner (or for our purposes, NASA), would provide the performing organization (or for our purposes, Jacobs Technology), with a product backlog. The priority of this backlog is controlled by the Product Owner and is equivalent to the work packages needed to be executed by the performing organization. The backlog also represents the scope of work, which is subject to change, which is where Scrum is valuable. The team, known as the Development Team (or for our purposes, the engineering team of Jacobs Technology), is self-organizing and self-led. Team members are expected to be highly interchangeable as the specialist title is considered obsolete as it is in contrast with the generalist that is self-sufficient, autonomous and can perform a multitude of roles. The hallmark of the system is the ScrumMaster. This servant leader is responsible for maintaining the agile method for the Development Team, Product Owner and any pertinent stakeholder (Schwaber et al, 2017). Currently, Jacobs Technology does not have this position or this framework.

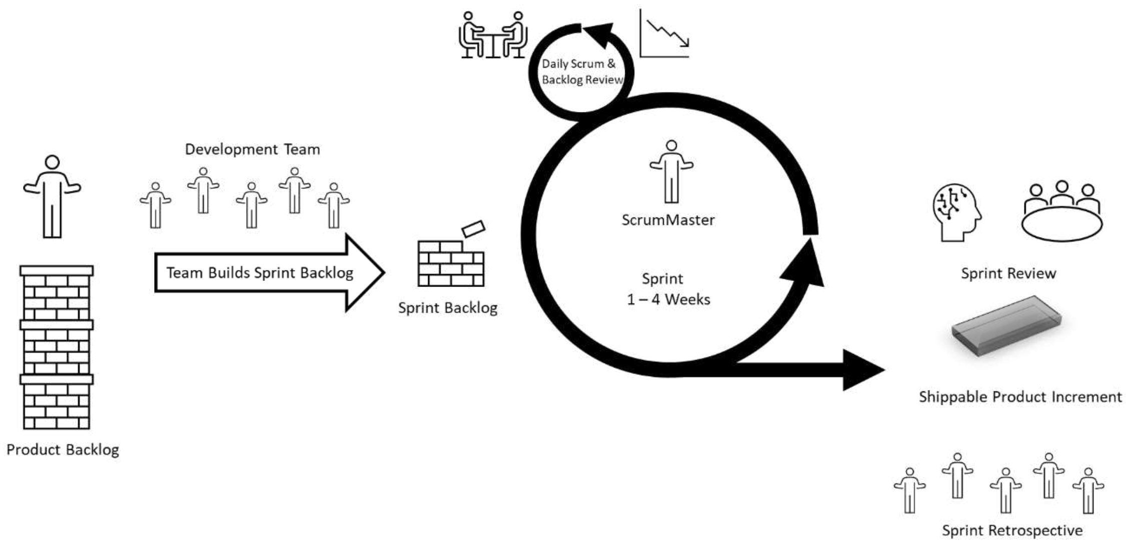


Figure 14 – Scrum Framework

Schwaber instituted the scrum methodology in three separate companies, demonstrating the application is not specific to software industries alone. One of the companies, MetaEco, a software provider, had been operating at a cash deficit. Due to the complexity of the new product, scope definition in the traditional waterfall method was hampering MetaEco's bottom line. After the implementation of scrum, the team's productivity and ability to meet customer requirements was achieved. Missed requirements that had been commonplace at MetaEco were now able to be captured in a product backlog, which was maintained by the Product Owner. The teams that were already conducive to a self-developing mythos, were able to deliver incremental and iterative deliveries in two-week sprints (Schwaber, 2004).

However, Turner and Boehm indicate that scrum has been one of the very agile frameworks to scale up for larger projects, as Jacobs Technology is much larger and more established than MetaEco. These successful scrum scaleups have their place in integrated project teams, where team coaches are part of an echelon of team coaches that can permeate several projects and product lines (Turner et al, 2003). These lead to a strong compliment of Cockburn Level 3 personnel, as dictated by Cockburn and emphasized by Manzo as the following: 3, able to revise a method and breaking its rules to fit an unprecedented new situation; 2, able to tailor a method to fit a precedented situation; 1A, with training, able to perform discretionary methods steps such as sizing stories to fit increment, composing patterns, compound refractory or complex Custom Off The Shelf (COTS) integration; 1B, with training able to perform procedural methods steps such as coding a some method, smile refractory, following coding standards and Configuration Management (CM) procedures or running tests; 1, may have technical skills but unable or unwilling to collaborate or follow shared methods (Cockburn, 2002). These levels of complexity of deliverable work will help classify which people with various sills can be expected to do within a given method of framework. Studies have shown a degree of challenge that rivals the application of agile: the transition of the performing organization to adopt a different mode of operation from the traditional or cemented platform (Manzo, 2003). Jacobs Technology will best be described as either 1A or 1B.

Chen, Ravichandar and Proctor performed an investigation of Cisco Systems and their transition to an agile framework. In accordance with their study, the first step is to identify and help business units and engineering teams adopt the method. The second step is in developing new management practices that are compatible and can sustain agile development. These steps elaborated are to demonstrate to the company how to select organizational units and teams for conversion from non-agile to agile development processes, including and not limited to new management processes associated with the framework shift.

The question prompted follows, “how can companies develop new management environments and practices to enable and support agile development practices?” In addition to background case study on Cisco, research categorized challenge levels with a rubric to pairwise and contrast the former and latter frameworks. In the first challenge level, a benefits assessment readiness assessment and areas where the performing organization could support the transition are grouped. The second group of challenges revolved around the new management practices. These included managing the leading agile engineering teams, planning and forecasting in the agile development process, coordination in the agile development process and recruiting collaborative customers (Chet et al, 2016).

Hekkala, Stein, Rossi and Kari found in a study that organizations which transition to agile can potentially experience failure early in the shift if they are unable to acclimate to the agile mindset from the onset as evident in the discussion that early failure leads to cascading failures. Appointed employees at Omicron, a small software company, were given one-time training to lead agile teams. However, because of working experience with agile methodologies and current organizational process assets, teams were not prepared to readily adapt to a self-guided mentality. Within these teams is the concept of incremental and iterative deliveries that were inept as the Development Team lacked the experience. There was a level of dissention between the Development Team, the Product Owner (or in this case, the customer) and the ScrumMaster (Hekkala et al, 2017).

Ultimately, what was to be a change in paradigm at Omicron led to an agile-named method with a traditional waterfall mentality ever-present. The responsibility was two-fold as it resided with the ScrumMaster to provide the agile leadership as well as the performing organization, which was not coalesced at the upper management levels, reluctant to fully adopt the process. The struggle the ScrumMasters experienced at Omicron was they were no longer servant leaders to the team, but servant leaders to upper management. It is the agile mindset that dictates that serving the team will inevitably serve the performing organization (Sutherland et al, 2011).

Supporting the result of Omicron were the findings Moe, Dingsyr, and Kvangardsnes as detailed in their discussion, as shared leadership indicates a prevailing ideology that team members who are traditionally conditioned to take instruction from management directly examined that the perception of the Development Team was directly correlated to the success of the agile leader, ScrumMaster or equivalent. The opinion of the research indicates that although upper management and the environmental enterprise factors influence project-specific behavior amongst members of the Development Team, that the agile leader within the team ultimately had more significant impact on interaction and success of the agile model within the context of the project. If this were the case, program and portfolio levels were more prevalent in manifesting the agile mindset as opposed to the inverse application of top down, hierarchical management mandate (Moe et al, 2009).

4.4.1. Roles

Scrum teams are comprised of three main roles: the product owner, the Development Team and the ScrumMaster (Griffiths, 2018).

4.4.1.1. Product Owner

The responsibility of the product owner is to maximize the value of the product. This is done by managing the list of work to be done, known as the product backlog. This product backlog is further decomposed as a function of each iterative sprint and the subsequent moving average of the completion of items in each recursive sprint backlog. The product owner maintains the responsibility of communicating the project vision to the Development Team, the project goals and the details of the work needed to be completed for each sprint. Furthermore, the prioritization of the backlog is the key responsibility of the product owner.

4.4.1.2. Development Team

The process of the Development Team performing quality checks and the customer performing product acceptance follows closely the Control Quality and Validate Scope processes where the outputs are Verified Deliverables and Accepted Deliverables, respectively, defined by the Project Management Body of Knowledge (PMBOK), 6th edition. After the sprint review, the sprint retrospective will reflect on the process and identify areas of improvement. Any residual product would be reinstated in the product backlog for the next sprint. The ScrumMaster would preside over the entire operation and exist as a servant leader. The definition of done is determined by the stakeholders so there is no discrepancy when leading into the analogous Verified Deliverables and Accepted Deliverables outputs (Project Management Institute, 2018).

4.4.1.3. ScrumMaster

Sutherland and Ahmad defined the ScrumMaster as the servant leader of the team responsible for the integration efforts, application of the agile mindset across all stakeholders and responsible individual for the removal of impediments for the team.

The ScrumMaster will provide leadership and coaching to assure the success of the agility and structure (Sutherland, 2015).

4.4.2. Artifacts

Ambler describes the scrum artifacts that reside within these scrum events. These are the product increment, product backlog and sprint backlog. These could be analogous to a quarterly deliverable/milestone, work packages remaining and work packages to be delivered in a specified time frame, respectively for Jacobs Technology but limited to the traditional waterfall approach.

4.4.2.1. Product Backlog

The product backlog are the items that are prioritized of all the work that needs to be done or is remaining after several sprints. This prioritization is the ultimate responsibility of the product owner. These items are sorted so that the high-priority work packages are included towards the top of the backlog. The product owner needs to maintain the dynamic nature of this list, as none of the items are static and can change as what is indicatively important to the project throughout the product lifecycle. The Development Team will work on top priority items first and the remaining items are progressively refined.

4.4.2.2. Sprint Backlog

The sprint backlog is the set of items in the product backlog that are selected as the goal of the specific sprint and maintained by the Development Team (Ambler, 2012). In conjunction with the sprint backlog, the Development Team determines the approach to achieve the sprint goal. The sprint backlog provides a highly visible view of all the work that is to be completed during the sprint and much like the product backlog's relationship to the product owner, the exact same applies to the sprint backlog for the Development Team as the team is the ultimate authority.

4.4.2.3. Product Increment

The product increment is a portion of the work that is completed during the sprint. This increment is built by the Development Team. During the sprint review the demo is presented by the Development Team to demonstrate their latest incremented in order to receive feedback from the customer, specifically the product owner. What is important to be defined is the definition of done, which is an agreement between the entire Scrum team as to what must be completed for each product increment (not limited to user stories, epics, etc.) so as to successfully guarantee customer acceptance and a collective vision across the agile development of the product.

4.4.3. Ceremonies

Kniberg differentiates the ceremonies in the context of Scrum. Ceremonies or events that exist in Scrum are commonly referred to as the five activities. These include the sprint planning meeting, daily scrum, product backlog refinement, sprint review and sprint retrospective. These partitions are analogous to the kick-off meeting, weekly updates, scope identification, customer acceptance meeting and lessons learned, respectively for Jacobs Technology but limited in the traditional waterfall approach. All ceremonies are time-boxed; Scrum terminology to indicate a specified time frame (Kniberg, 2015).

4.4.3.1. Sprint Planning Meeting

In this meeting, the entire agile team gathers to agree on what items will be delivered during the upcoming sprint. The product owner will present the latest product backlog and the Development Team discusses with the stakeholders how they will have a shared understanding of the definition of done as it pertains to the sprint. The Development Team will predict what can be delivered in the sprint. This is done as a function of both the burndown charts and the scope of what is to be delivered in the current sprint (Albero, 2014). The Development Team ultimately decides how the functionality will be built and how they will organize and perform the work with the sprint goal being the driving factor.

4.4.3.2. Backlog Refinement

Backlog refinement is the process where the backlog is slowly groomed as a result of the iterative sprints to assist in the delivery of the scope needed by the customer. This effort involves every role player on the scrum team (Griffiths, 2012).

4.4.3.3. Daily Scrum

The daily scrum (scrum defined as a meeting) is a 15-minute, timeboxed activity that is typically held at the same time and place every day to ensure the Development Team is working towards the sprint goal. The ScrumMaster is responsible for running the meeting, making sure it occurs daily, the team stays within scope and follows up on any identified impediments. This daily scrum is typically observed for the Development Team and the ScrumMaster although it is not atypical that other stakeholders pertinent to the focused effort may be included. This can extend towards the following individuals and is not limited to product owners, SMEs and project managers. The scope of the meeting and timeframe is strict in nature. The ScrumMaster will typically lead the meeting and address each of the members of the Development Team. Each of the Development Team members answers the following questions briefly as they relate to their involvement in the sprint goal:

1.) What have I done since the last Scrum?
2.) What do I plan on doing today?
3.) What are any impediments I am facing?

Should there be any impediments, the ScrumMaster takes an action to help resolves these, thus fulfilling the servant leader. As Scrum teams get larger, it is possible to employ the approach called the “Scrum of Scrums” wherein a representative from each of the subset of smaller Scrum teams coordinates a cross-functional meeting which functions in a similar fashion to the aforementioned daily scrum with the additional benefit of asking the question, “what potential conflicts may arise?” This approach satisfies the systems engineering aspect in terms of multi-disciplinary teams working together and reducing the silo affect as a result of compartmentalization (Hunt, 2018).

4.4.3.4. Sprint Review

This review is held at the conclusion of the sprint and includes the members from the Development Team, the ScrumMaster and the product owner, but could also include other pertinent stakeholders, although this is circumstantial and pending on what the sprint provided in terms of functionality from the performing organization to the customer. In the review, the Development Team holds a demonstration on the product increment that was recently developed during the sprint to the pertinent stakeholders, primarily the product owner. The product owner will validate the work using inspection methods against acceptance criteria and will decide if the increment meets the definition of done. This effort also allows for the refinement of the backlog items, both for the elaboration of the product backlog and subsequent sprint backlog, preparing the Development Team for the next sprint (Griffiths, 2012).

4.4.3.5. Sprint Retrospective

Following the sprint review and before the next sprint planning meeting will be the sprint retrospective. This meeting is the responsibility of the Development Team and is part of the inspection and adaptation pillars of Scrum. This meeting is typically for the ScrumMaster and Development Team although other stakeholders may be invited depending on the circumstances. This meeting is primarily to review lessons learned and examine opportunities for improvement. One of the primary driving factors will be the product owner's feedback from the sprint review and a reflection of what items deem attention that could potentially serve as benchmarked experiences that could drive change and implement process improvement for the next sprint (Griffiths, 2012).

4.5. Lean

The Lean philosophy emphasizes cutting waste and inefficiencies. The aim is to minimize partially done work, an impetus on making local decisions as a team vs. by management when applicable, working less process while maintaining value, reductions in task switching and reducing delays where applicable.

Lean development identifies 7 Forms of Waste. These have been established primarily for manufacturing processes but have also found their way in software development. In the context of hardware development, here are the 7 Forms of Waste for software intensive projects that are applicable to GSE development of hardware.

1. Defects

a. Defects are items delivered to the customer that in terms of hardware or documentation. These could also be bugs associated with software builds for the test stations.

2. Hand-offs

   a. This is considered the effort to facilitate motion to communicate information from one group to another. Examples include if teams are not co-located, or work is handed off.

3. Waiting/Delays

   a. These are delays associated with approvals and reviews. These could be signatures on documents or drawings.

4. Task Switching

   a. This is the multi-tasking between several different projects. Lean experts converge on as much as 40% degradation in productivity during task switching. This could include resources working multiple tasks for multiple projects.

5. Extra Processing of Extra Documentation

   a. This includes additional work that does not provide value to the customer. These processes include unused documents or unnecessary approvals for several deliverable types such as drawings, fabrication documents, procedures, etc.

6. Unnecessary Features

a. These are extra pieces of functionality while are nice to have been not entirely necessary. This can include gold-plated items that either are not necessary or were asked for by the customer that were not formally approved.

7. Incomplete/Partial Work

    a. Partial work completed introduces entropy into the systems engineering process and does not deliver value. This includes drawings started that were never finished due to descoping, documents that were created but no longer needed, etc.

4.6. Agile & Lean Concepts In Action

A team from Stanford formulized using an agile tailored approach to build a system-on-a-chip with specialized accelerators. The current approach uses a waterfall-like style to build accelerators using a hardware specification and continues through a number of refinements to fine tune the accelerator design. The waterfall approach suffers from challenges relating to changing application requirements and an incomplete comprehension of the customer need based on a limited understanding of the need at the project onset, resulting in an approach that requires an iterative and incremental approach to the hardware/software designs to generate an accelerator. The main contribution of this work indicates that while using agile to develop software, the hardware itself by virtue of its integration with the software, must also participate in agile development. The team from Stanford identified the importance of separately dealing with different concerns and a method from communicating changing design to all layers of the end-to-end flow, which resulted in shorter design times (Bahr, et al, 2020).

The major lessons learned, and takeaways include a framework to address changing features by implementing seamless communication to all layers in a flow of work and that in certain cases, it is imperative that hardware perform in an agile behavior to pair with its agile software development. In particular, a method was established for dealing with different concerns in a more explicit manner. The team at Stanford recognized that the challenge in integration is about managing the design of the end-to-end flow 's layers to enable the cross-layer constraints to be steadily satisfied and allow the designers to compile and measure the applications on the developed hardware continuously and iteratively. The software, while still at a level of minimal comprehension by the developers, could be understood iteratively and incrementally by providing an end-to-end hardware generation and software compilation flow using programming language.

A team at Clemson and the University of Texas at Dallas explored challenges associated with hardware development with an agile approach, specifically in a Scrum capacity. The comparative analyses were performed at a textual level through logical intersections and a thematic approach. The themes investigated included flexibility, chunkability, scalability, endurability and teamability. The effort established two constraints known as a constraint of physicality, which is a collection of limitations associated with implementing agile principles in hardware development and a second principle guided by 13 principles that should be maintained when applying scrum methods to hardware development. The goal of the study was to determine the gaps and overlaps associated with agile deployment (Peterson, et al, 2021).

Schmidt, et al, identify constraints of physicality (CoP) i.e., the duration needed to construct physical, shippable increments and/or development physical prototypes in the context of hardware development, as the crutch of utilizing scrum or other agile methods in their development by contrast of software-intensive systems with no physicality associated. Programmers are able to compile code and receive a level of functionality nearly immediately, limited only by computational power and programmers' expertise whereas physical products are limited by more tangible, physical laws which govern how quickly tangible prototypes may be made and how economically feasible they present themselves within the context of scope and budget. While challenges associated with this development are highly entangled and manifest in a complex system, it is not always obvious what the root cause or influential factors are at play. The investigation's goal was then to identify the most critical issues by segregating between causes and effects. The study identified 153 challenges and 160 interdependencies with four backbones which indicating the highest influencing in hardware agile development in an effort to reduce constraints of physicality (Schmidt, 2017). The nodal analysis central to these studies identified the 37 nodes which contributed to the challenges associated with agile in a hardware environment with each node identified with associated causes and effects. The central node of the network became the "hard to overcome the constraints of physicality" node as the root of challenges as the node contained six causes and three effects. The remaining nodes were directly or indirectly responsible for these six causes. The first study identified constraints of physicality caused by difficulties (1) separating development tasks, (2) estimation of task duration in terms of resources and timeframe, (3) defining feasible increments per iteration and (4) appropriate flexibility. A second study identified constraints of physicality arising from (1) methodical shortcomings as no agile prescribed method had been established for a physical product development and (2) scalability as hardware development demonstrated via the study that undertaking of hardware projects typically elicited a greater variety of interdisciplinary groups rather than strictly pure software associated endeavors. The third study had indicated that the constraints of physicality resided with (1) feasibly delivering working product increments that are shippable to the customer, (2) responding to changes quickly due to short incremental sprints and (3) which led to the stigma that agile and scrum practices are rarely used in comparison to their software counterpart products in the development of any physical media. The case study associated with the three interdependent studies extracted the main sources of separation (i.e., hard to separate deliverables for each iteration, hard to develop potentially shoppable increments from each iteration), flexibility (i.e., hard to be adequately flexible, managing supplier times and other external dependencies), scaling (i.e., difficulty in scaling due to complexity) and task breakdown (i.e., inability for the development team to adequately project time and resources needed accurately and within a sprint-style delivery envelope).

Relating to Schmidt’s CoP, Ullman realigns his former text on the Mechanical Design Process with a supplement on scrum for hardware designs by identifying 13 principles that could pose challenges and how best to maintain them. The supplemental section identifies the following challenges: (1) hardware modularity as a modular design allows for flexibility (linkage back to one of the four backbones of Schmidt's CoP), (2) difficulty in hardware development over short sprints (another linkage back to one of the four backbones of Schmidt's CoP), (3) difficulty in adding features to finished hardware, (4) hardware is more difficult to simplify, (5) hardware needs more specialization in terms of specialized functions, (6) demonstration function takes longer for hardware, (7) changing hardware costs more money, (8), hardware must work over a range of time and environmental conditions, (9) software testing is more feasible than hardware testing, (10) developing hardware requires ancillary systems not needed for software, (11) differing software and hardware requirements, (12) prototyping, demos and testing often more difficult for hardware, (13) Scrum encourages a Build-Measure-Lean process which is a poor hardware design practice (Ullman, 2009).

Atzberger brings to light the idea of how agile ideas, while has the perception as always being beneficial, has an appropriate level of "hype" associated with its success across all domains in the publication the *Evolution Around Hype of Agile Hardware Development*. The focus of the study pertains to physical products, in specific hardware development projects. Previous studies show that "soft" parameters such as transparency and communication are improved but no tangible data in terms of physical product development. This updated study reinspects this notion of "hype" by performing a follow-up which included a large deviation in terms of expectations and differing perspectives from the standpoints of product development. The results included a Likert scaling from 1 to 5 (1=not experienced, 5 = expected a large degree) with juxtaposition of the 2017 and 2018 years expected vs. actual projections, with ratings of 73 individuals participating in the survey for 2018 and 113 from the previous year of 2017. Overall, anticipated (or perceived) benefits due to agile overall slightly decreased when compared to the previous year, overall numbers with the exception of hard benefits including shortened product development time, improved product quality, improved customer benefit and improved product alignment with corporate strategies were all above the target of 3, which indicates an overall improvement of hardware development for agile. In order to exploit advantages, this study will focus on a method for improving the hard benefits with specific focus to improved product quality, reduced development costs and shortened product development time by tailoring the xEMU-specific Scrum model to appropriately account for these risks (Atzberger, et al, 2019).

Table 1 – Soft vs. Hard Agile-Based Benefits from Atzberger Study

	Facet	2018	2017	Delta
Soft	Communication	4.37	4.1	0.27
	Reduced Risks	3.33	3.26	0.07
	Transparency	3.86	4.08	-0.22
	Improved Corporate Strategies	2.7	2.72	-0.02
	Improved Customer Integration	3.4	3.39	0.01
	Improved Learned Process	3.49	3.54	-0.05
	Increased Commitment	3.79	3.95	-0.16
	Improved Customer Understanding	3.4	3.34	0.06
	Improved Control of Complexity	3.49	3.54	-0.05
	Increased Responds to Change	3.86	4	-0.14
	Higher Chance of Market Acceptance	3.34	3.11	0.23
	Increased Product Development	3.24	3.51	-0.27
	Increased Exploited Emerging Opportunities	3.08	3.07	0.01
	Increased Reaction Speed to Changes	3.83	3.95	-0.12
	Increased Effectiveness of Project Development	3.58	3.97	-0.39
	Increased Customer Satisfaction	3.45	3.72	-0.27
	Improved Development Processes	3.04	3.25	-0.21
Hard	Quality	2.93	3.03	-0.1
	Time-To-Market	2.9	3.28	-0.38
	Early Customer Benefit	2.96	3.25	-0.29
	Improved Delivery Dates	3.04	3.44	-0.4
	Cost	2.15	2.56	-0.41

Augustin's user-centric development paper explores the potential of combining both design thinking and scrum to optimize user integration and exploiting advantages associated with sprints (i.e., short iterations) in hardware-centric projects. Paper investigated a German high-end manufacturer over a three-month development period and demonstrated how deployment of agile methods guided sprint-like cycles within hardware development achievable with the intent of speeding up product development with the assurance that the product matches the intended scope of the user. Design Thinking is the approach that starts with empathy, continues with problem definition, a brainstorming and idea consortium, rapid prototyping following with testing.

The combination of scrum and design thinking brings to light the idea of accelerated iterations with Scrum where early testing with prototypes results in the minimum viable product or MVP that can be used and tested by users and any feedback integrated in future sprints if applicable. The study recognizes the constraints hardware development brings, especially in the context of a physical prototype vs. a virtual prototype commonly associated with software-intensive designs. The most alluring compromise was attempting a very strict 2-to-4-week window and still deliver prototypes but beginning with extremely coarse-grained quality/functionality with successive builds. The study had suggested during its case study that exaggerated rough models i.e., cardboard, paper and office supplies, while iteratively and incrementally building to your final product with an MVP as the centric model. Another aspect this study takes into consideration is the idea of early feedback, typical of scrum and agile-style developments with the exception that the development team observes the user in a practical scenario context vs. obtaining direct questioning. The idea of direct question avoidance would be the bias of the user to the degree of their personal experience vs. unbiased, third-person observation. This also brought to light the idea of letting the user be part of the development team, with the highly implicit opinion that this would indeed be the product owner as well. Synergistic meetings with users brought the emergence of, "is the development team asking the right questions," in the context of early development. One of the major struggles of eliciting feedback from the user is that the prototypes often seldom look like their finished counterparts. One counterpoint was to ask the correct questions and avoid any that might point out flaws in the prototype that may not be indicative of the final product and are simply limitations associated with the abridged version of the MVP. An example of a productive feedback question would be, "what needs to be improved that isn't already working perfectly?" The study entertains the notion of separate pathways associated with iterative development. For example, the typical brainstorming, model making, demo, product testing, product development and user demo of finished product are employed but rather than have a full loop, side loops are presented to create iterative approaches back through the early design cycles if product development isn't in a state where it can be continued through product testing, development and user demo. The proverbial fork in the road would be the presentation to management wherein the design team could continue with final product pathways or if defined approval (a form of PMI's Validated Scope if the performing organization's upper management could be used to exemplify the customer in the early stages of development).

The case study proceeded with 3 sprints each 4 weeks long. Categories included spring planning, research, ideation, visualization, prototyping, model making, sprint review, and sprint retrospective. The planned sprint and true sprint lengths were modeled with results showing true sprints taking less than expected in Sprint 1, true sprints being equal to their planned counterpart in Sprint 2 and Sprint 3's true sprint taking 20% longer than was previously anticipated. One of the notions unrecognized at the onset was the parallel efforts of working separate tasks in tandem (i.e., model making and ideation) that were originally deemed to have their own segregated slots but the efficiency of doing both in tandem paid tribute to the idea of iterative and incremental planning, almost as executing a miniature version of agile development within agile development. The following challenges included: a MVP unobtainable within the first sprint due to product complications with the conclusion that detailed sketches instead of models would be given; intertwined activities not previous planned allowed for completion of the 2nd sprint where a model could be made at the end of the second sprint; the third sprint had a week of delay due to manufacturing, an external dependency, and the benefits of intertwining activities in visualization and prototyping allowed to save time (Augustin, 2019).

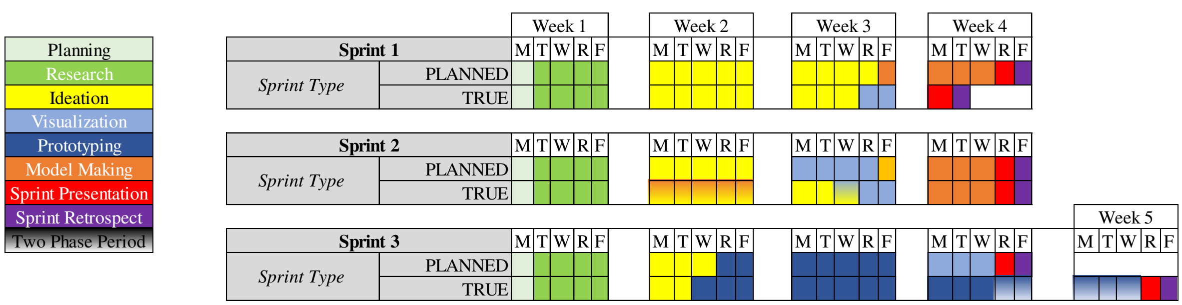


Figure 15 – Augustin Inspired Sprint Schedule

In the *Current Challenges in Agile Hardware Development*, the author Atzberger from Bundeswehr University in Munich revisits a case study performed in 2012 and reevaluates them against the current challenges associated with agile development for hardware projects with the aim of discovering any attempt to explain difficulties of agile development in hardware projects that still persist and the interrelations with their advancements compared to seven years ago was discussed. The team identifies the persistence of CoP in addition to paradigm complexity, designer's dissent, team distribution dilemma and education. The findings of the result then take what were the challenges in 2012 associated with the challenges of 2018 and reidentifies those into CoP, mindset, scaling and team distribution. Regardless of the renaming and reallocation, the initial challenges of the 2012 study in comparison to the 2018 study show initial challenges are still valid. The study showed 6 independent case studies and attempted to align which challenges were recursive. CoP is identified as the complications associated with challenges that occur by virtue of hardware products manifesting into a physical byproduct. CoPs are time-consuming when compared to writing and compiling software code. The biggest challenge with CoP would be the short increment times that are not as feasible with hardware development projects (Atzberger, 2019).

Paradigm complexity refers to the difficulty of transferring a radically new development mind shift into a performing organization that may be perceived as contradictory to the established, traditional development environment by attempting to integrate two seemingly contradictory models with conflicting values. This idea, known as scaling, in some cases with respect to SAFe and LeSS are gaining less popularity (Brenner, et al 2015). Designers dissent refers to the willingness or lack thereof of applying agile methods correctly. Cooperation in this regard is a possible barrier to entry for success in agile. Team distribution dilemma is the inability to work in a co-located environment as outlined by the Agile Manifesto. This case becomes especially interesting with the advent of telework, especially accelerated by the COVID-19 pandemic. The idea of education deals with the introduction and acceptance of agile frameworks into the performing organization and how critical a common understanding and acceptance is for project success (Atzberger, 2019).

Table 2 – Current Hardware Agile Challenges

Category	Category Summary	Specific Challenges	Challenge Description	No. of Case Studies Affected	Challenge on xEMU?
Constraints of Physicality	These are the challenges that are associated with constraints of physicality of building a physical product and the limitations in expediency and ability to build physical product as opposed to the feasibility of building code and the propensity of quick turnaround sprints associated with software-intensive projects.	Realization of shippable increments	Hardware projects typically deliver a final product in a tangible fashion at the end of a project and doing this in incremental deliveries with a physical product may be challenging	5 of 6	Yes
		Feasibility to produce products	Being able to produce a product that meets customers desires with variable scope is challenging when it comes to the effort to change a product that is tangible vs. a non-physical product (i.e., software code)	4 of 6	Yes
		Inability to breakdown product into sprints	Hardware projects that need to be borken down into sprints is challenging given the complexity of mechatronics, interplay between all functionality (i.e., electrical, software)	4 of 6	Possibly
		External dependencies	External dependencies include those that the development team cannot directly control. These include but are not limited to resources, purchasing, delivery of hardware, etc.	4 of 6	Yes
		Production of tools	This challenge would be the availabity of physical resources be it commerical off-the-shelf products, stock material, etc.	4 of 6	Possibly
		Documentation	Testing of components or having high-quality and highly dependable components requires documentation such as material certifications, lot traceability, certificates of conformance, etc.	4 of 6	No
		Specialization of development team	Engineering teams tend be specialized in functional groups (i.e., Software, Electrical, Mechanical) where in agile teams, generalists are preferred	4 of 6	Possibly
		Synchronism of domains	In comparision to software projects, hardware projects take a vast amount of functional expertise to complete (i.e., materials, software, electrical, mechanical, project management, etc.)	4 of 6	Possibly
		Frequent stakeholder feedback	The customer (in Scrum the Product Owner) must be continuously engaged to ensure proper scope is being delivered and value is maximized	4 of 6	No
Mindset	These are the challenges that are associated with the superordinate for understanding of agile by both the individual and the performing organization	Establish agile mindset	The agile mindset, in particular the Agile Manifesto 4 values and 12 principles, must be established especially the notion of frameworks vs. methodologies	6 of 6	No
		Proper training	Training in the agile framework applied, specifically Scrum, must be undertaken for a successful agile deployment	6 of 6	No
		Acceptance of agile in organization	The performing organization and agile team cannot be successful unless the performing organization is able to embrace the change in infrastructure	4 of 6	Yes
		The "Prince Problem"	This issue is a result of hierarchical structures adapted to agile that potentially remove authority of managers and a loss of responsibility which may be deemed threating to the company infrastructure	3 of 6	No
		Commitment of the top management	The performing organization and agile team cannot be successful unless the performing organization is able to embrace the change in infrastructure	4 of 6	No
		Commitment of the middle management	The performing organization and agile team cannot be successful unless the performing organization is able to embrace the change in infrastructure	4 of 6	No
		Multi-project management	One of the issues associated with agile and traditional frameworks is the idea of multi-management i.e., project manager, section manager, functional managers, etc. Here, the development team is its own manager with the ScrumMaster facilitating the Scrum activities and harvesting the mindset	4 of 6	Yes
		Internal process models	The incomptability with certrain organizational process assets	4 of 6	
Scaling	After overcoming the challenge of developing the mindset of both the individual and the company, the next challenge would be scaling which is the rolling out of agile on several projects with potentially different infrastructures, team members and product deliverables.	Transfer of methodical knowledge	In order for agile rollouts to be successful, appropriate transfer of the agile framework must be easily and feasibly transferable	5 of 6	No
		Structure of the company	The performing organization and agile team cannot be successful unless the performing organization is able to successfully adapt the agile framework to the company's structure	6 of 6	Yes
		Silo mentality	Teams within a performing organization especially functionalized may run into issues where they work independetntly from other functional groups. The development team must never be siloed in order to perform effetively	3 of 6	Yes
		Mindset change of organization	The performing organization and agile team cannot be successful unless the performing organization is able to have been influenced by the agile mindset	5 of 6	Yes
		Adaptation of company-specific values	The incomptability with certrain organizational specific values	6 of 6	Yes
Team Distribution	These are the challenges associated with communication and its tools, differences in location and cultures.	Communication of distributed teams	Agile teams are intended to be collocated, meaning not just in the same building but in the same physical space, many times with communication by osmosis and barrierless workspaces	4 of 5	No
		Usage of communication tools	Agile teams focus on push and pull communication in addition to any other modes of communiation that enable for a steady stream of communication, especially when teams cannot be collocated constantly	2 of 6	No
		Ethical and cultural differences	Different teams spread across different countries may have communication issues due to ethical and/or cultural differences	2 of 6	No

Oja performed a case study over a construction project with ABB Grid Integrations in Finland to discover if Agile Project Management (APM) and Scrum, both having found success in software-intensive projects, could be used to improve the project management of projects. The research question was as follows: could APM [and Scrum] be used to improve project management in the case organization during the initial phases of engineering, procurement and construction projects? It is important to note that while this is a construction project, the current team format, characteristics of the performing organization and project environment were very typical of the case study presented in the xEMU GSE Testing Team. The team is led by a project manager, consists of a collection of leads, engineers from project, electrical and mechanical engineering with several engineers working multiple projects in tandem and ancillary personnel that complement the team (purchasers, customers, etc.) with the size of the team complementary to that of the case study. The results of the study suggest that APM was most beneficial in the case study during the initial phases at the project onset and additionally most beneficial as it pertains to improvements in project management. The largest challenges included team members working on multiple projects and the idea of ScrumBut, which is a variation of Scrum, "but" certain aspects are not used. While this goes against one of the main tenants of Scrum that Scrum must be done in its entirety, being able to use a packaged version of scrum at the project onset was proved to be beneficial according to the results of the case study (Oja, 2017).

Table 3 – Concept of Planning Deployment HW vs. SW

Concept of Deployment in Planning Sprints (via an MVP)	
Software	**Hardware**
Negligible amount of time when compiling software	Considerable amount of time procuring items
Negligible amount of time to acquire hardware to support	Frequently custom parts need to be manufactured
Negligible amount of time to assembly hardware to support	Almost certain will parts need to be assembled and tested

The article Briatore, et al published identifies the impracticalities associated with applying analogous task effort estimation to hardware projects as is done in their accompanying software counterparts. The article performs an experiment on engineering students via a pilot validation experiment of a novel form of agile framework with a specific focus on a parametric tool to estimate task effort than the traditional confidence votes when producing a schedule more rigorously. The pilot validation study for the hardware development project will use a benchmarked approach for Scrum on an electrical project used for estimating task efforts. The approach encoded a mathematical model to correlate team-specific data from the electronic board development (hardware facet of the electrical design) with the time needed to complete the work packages. Predictors of time allocation would rely on the identification of key design drivers (i.e., critical path items). The aim would then be to use Scrum in an embedded fashion into the proposed hardware agile approach to reap the benefits of a Scrum on electrical projects while accounting for hardware limitation that would need augmenting not previously accounted for in the software-intensive facet. The team after identifying a workable, analogous electrical agile development began to lay out their current, traditional hardware development and first did a best approximation of tailoring their waterfall style approach to the new agile tailored framework with the major goal of deriving time estimation efforts to realize a measurable before and after implementation against previous projects that had used the traditional waterfall approach for hardware development. The three main considerations when estimating the tasks at the planning phase included: (1) a measurable outcome of each sprint, (2) the sprint length and (3) involvement of the customer. The test was conducted around 28 students split into 7 groups with an 3-day workshop with each day consisting of a sprint with equivalent product owner and ScrumMaster constituting as the study owners with the intention of student iterations through the estimation tool would fine tune their projected schedule estimation at the onset in comparison with their projections after experience with the tasks assigned and re-evaluation with the agile estimation tool by characterizing the ability of a student to predict the time for each task depending on the student's behavior done in previous work compared to the 30-minute projection per work package. At the conclusion of the experiment, 5 groups (which were considered valid data sets based on the assumptions outlined in proper data for evaluation The validation experiment demonstrated an improvement from a minimum of 8% to 18% in schedule savings when employing the tool presented during the planning phases of the development. One of the major findings included increasing the sprint time from 1 week to 2-3 weeks with the caveat that this scalability is specific to the hardware that was developed for the electrical components when juxtaposed to the electrical designs themselves which only took the prescribed week. One of the major limitations of the study would be the inability to provide a readily available metric for scalability from project to project, thus one of the goals of the study for future considerations would be to employ this framework to a larger sample size (Briatore, 2021).

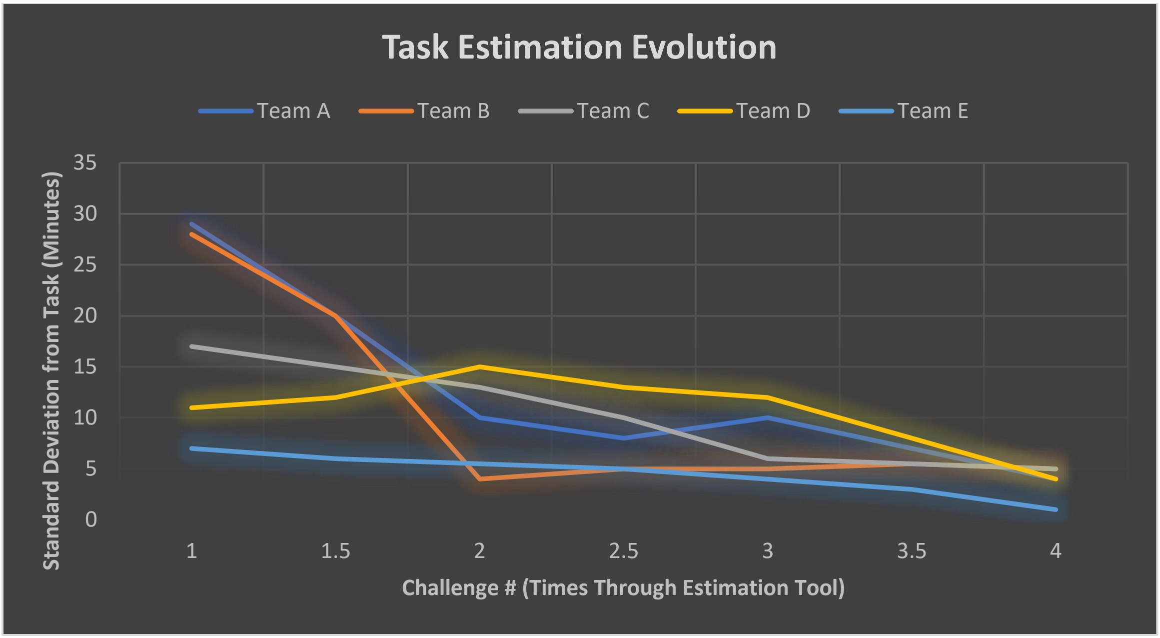


Figure 16 – Task Estimation Evolution by Team

In the pilot study *Effectiveness of Scrum Methodology for Agile Development of Space Hardware* for the payload development project, Garzanti indicated that all sprints with the exception of one had failed to meet the backlog completion. The study determined the failure to be two-fold. First, it was the learning curve for improving workflow handling as engineers underestimated complexity or misunderstood interdependencies between different tasks. This underestimation also included performing tasks that at one point were considered to be further downstream that in reality should have been done in tandem earlier on. Second, procurement and manufacturing lead times lead to complications to allow for sprint completions under tight timeframes. This led to approximately 35% extra time to complete a sprint. On average, the payload teams completed an average of 59% of the work on average with a standard deviation of approximately 12%. While there were fluctuations, the team was still able to deliver the MVP in the final sprint. In terms of customer engagement, Scrum projects benefit from customer engagement where the MVP is able to be demonstrated at the end of each sprint. While this can't be done in its entirety on hardware projects, the team took advantage of rapid prototyping and 3D printing of mechanical parts for demonstrative purposes. Estimating task complexity and time required for implementation is the major challenge for scrum. Fibonacci sequence scoring was used to identify task complexity, however, was proven to be ineffective in sizing. After moving to a simpler time-based scoring system, the team was better able to predict. The main difference between Scrum in hardware vs. software is the time it takes to deliver an MVP as code can be built and tested whereas hardware projects suffer in terms of external dependencies on procurements and manufacturing. The strategy to address the delays in external dependencies would consist in scheduling overall workflow to consider the lead times of suppliers and shifting assembly, integration and testing sprints in accordance with hardware physical arrival. Testing will need to have its own sprint it was discovered for hardware projects, which differs for software as the continuous and iterative testing in each sprint by the customer is a substitute for final testing in a sense. Probably the largest benefit was the psychological factor which was split into tempo, task completion honesty and sprint planning proximations. The tempo of the development process is more cognizant in the minds of the development team with an improvement measured between 3-5% on each successive sprint. Task completion honesty improved as teams learned that their original projections were as much as 32% off target. Task planning in particular affected hardware procurements and manufacturing to include these times at the onset. Originally, the approach would be to spread the tasks among multiple sprints. This turned out to be a disadvantageous strategy. The best practice was determined to extend the time duration of the sprint itself during the planning in order to accommodate more tasks. It is also noted that a highly modular design or adherence as best as possible leads to design and task simplicity when a design change is needed thus approximately as close as possible it's Scrum software-intensive project counterpart

(Garzanti, 2020).

4.7. Requirements Engineering

The importance of requirements engineering is to bring structure to a project in terms of delivering to the customer a product that fulfills the function or operation of what was intended. Without well-defined and organized requirements, scope creep is a most likely a certainty, provoking the potential of an inflated schedule and a bloated budget, thus effectively destroying the project management triangle of cost, scope and schedule. Requirements are crucial as they communicate the stakeholders' intention by following a guideline to help create concise records, complete records, comprehensible records, traceability to source, a way to manage changes adequately and help drive the development of a product from stakeholder/business/product needs, goals/sub-goals, system requirements, subsystem requirements, unit requirements through the process of requirements development, design development and system development with the ultimate goal of obtaining formal validation and client acceptance.

Defects and errors are more expensive the further they go unmanaged. It is important that defects and errors are identified as early as possible and with requirements, that is the first of 3 steps in the Verification and Validation process (2nd and 3rd steps being Design and System, respectively). In requirements verification, we ask the question as engineers, "are the requirements written correctly?" where we rely on organizational requirements writing guidelines. These can be classified as and not limited to the requirements existing atomic, use of the word "shall," are they written in active voice, are complete, concise, comprehensible (not simply desirable) and maintain traceability. These then drive the system design and beg the question of the client, "are we building the right thing?"

Requirements engineering sub-processes in the design of a product can lead to the successful development of exactly what the customer has asked. One of these sub-processes is managing requirements. Requirements are managed throughout the lifecycle and must be elicited by the customer and further monitored and controlled and updated during the product lifecycle. An important part of management is traceability management, to ensure that all requirements are traceable to sources, managing change control wherein requirements can be changed through a formal process to identify areas where the customers' needs are not being met and if approved, changes will affect the baseline and baselines will be updated.

The second sub-process is observing the system context. The context is what drives everything else. If the context changes, so do the requirements. We must understand the context and we can do so through modeling. A helpful approach is a re-evaluation of the system-internal, interaction and context types, using context rich always to help aid or understanding of the systems environment. Parallel to this is examining the system boundaries and understanding the interfaces as these could produce emergent behavior that needs to be managed continuously.

The final sub-process is managing the requirements engineering activities of negotiation (i.e. how requirements are negotiated with the customer), documentation (i.e. how they are documented, organized and changed controlled), verification (i.e. "are the requirements written correctly," "is the design correct", "did we design it right"), elicitation (i.e.

stakeholder elicitation of needs/requirements) and management (how the following processes are adequately planned, executed and monitored/controlled).

The benefits of using goals, scenarios, use cases, solution-based requirements and documentation allow for a method of performing a robust requirements engineering analysis with elicitation, management, negotiation, documentation and validation recursively. To better understand the benefits, an examination of the facets of each will be discussed.

Goals allow for a high-level view of the project, based on the elicited needs of the customer. Goals are then decomposed into sub-goals, which are decomposed into system requirements, decomposed into subsystem requirements which all (goals, sub-goals, system requirements, subsystem requirements) are then traced back to their source, with descriptions and rationales to support each level of decomposition. What is also important is the strict framework in which each of the aforementioned resides. Starting with goals, they should be high-level, presented in an active voice, use "should" and represent the highest view of the customer need (with the system vision being the highest of them all). These goals need to be traceable, have a source, and provide value to the customer.

In a very similar fashion, requirements provide many of the same benefits. They are decomposed goals/sub-goals that deliver value to the customer, use "shall", adhere to the active voice and proper grammar/syntax structure, have rationale, avoid modal verbs and can be given in the system and subsystem level. They can be functional or non-functional (quality or functional requirements that are still lacking in robustness.

Scenarios ultimately benefit the project by allowing for a sequence of interactions that allows to describe the satisfaction of a goal. Goals can be viewed as positive (success in meeting a goal) or negative (failure to meet a goal), can classify misuse scenarios (hostile actors going against the stakeholders wishes), can be viewed in a current (indicative) or desired (optative) capacity with the benefit derived of initiating change if needed. We can classify scenarios as descriptive (meaning they describe and validate the goals and requirements in addition to highlighting the workflow/process), exploratory (meaning they explore alternative solutions/realizations; in a qualitative capacity and support decision making) and explanatory (explaining to parties outside of the effort the sequence of activities; quantitative in capacity). Scenarios can be instance (with concrete actors and activities), type (with abstractions in contrast to instance) and mixed, with type scenarios used for well understood scenarios aspects and instance with scenarios that are not as well understood. Scenarios can help differentiate between a Type A system-internal (with components only acting within the boundaries), a Type B Interactions (within the system and actors exclusively) and the Type C Context (richest, interactions with system and context).

Use cases are a type of scenario that introduce the main scenario (which is the satisfaction of the goal, primary vehicle), alternative scenarios (secondary, but still yield the satisfaction of the goal) and exception (which illustrates the dissatisfaction of the goal). These are employed with context, including and not limited to users, roles, resources, location, post-condition and pre-condition. The benefit here is that the entire spectrum of context is organized, and the satisfaction and dissatisfaction can be modeled using natural and modeling language.

Solution based requirements benefits the project by classifying aspects into data, behavior and functional. Data (which provides details on the entities, data specifics of the context), functional (which illustrates how the entity functions) and behavior (how the entity will behave as a result of an external input, which suggests its function). Use cases have application in both the business requirements and product requirements. By differentiation, use cases can be segregated for business and product requirements to increase robusticity by increasing the breadth and depth of study two-fold. By examining a larger scope of work when juxtaposed to use case product requirements, use case business requirements allow for a larger scope of work to be identified, effectively creating a better product (Roberts, et al 2012).

Documentation benefits the project not by creating paperwork but by the process of collecting, organizing, processing coarse-grained goals into fine-grained requirements, allowing for traceability to source and creating a requirements package that can be viewed by all stakeholders, agreed by pertinent stakeholders and maintained to ensure that the project work is being built the right way (verification) and that the project work resulted in building the right thing (validation).

4.7.1. Needs

Needs are considered the highest level of stakeholder elicitation and is done early in the project lifecycle. Needs, specifically business needs, are transformed into business requirements. The need is a single statement that drives the progressively elaborated goals and requirements. Stakeholder needs in like are transformed into stakeholder requirements. The pattern follows as system needs are transformed into system requirements and similarly for subsystem and unit needs to subsystem and unit requirements. This decomposition from needs to requirements substantiates the claim that building requirements are paramount on traceability back to a source and the source of a need elicited from a stakeholder is key.

A stakeholder is anyone or organization that can positively or negatively influence the success of a project (Dick, 2017). Further described by INCOSE, “Any individual or organization with a vested interest in the System of Interest (SOI), may be affected by the SOI, participate in the development of the SOI, or able to influence the system. Stakeholders are individuals who are considered to be relevant to the development of the SOI and with whom the project team will interact. Stakeholders are the primary source of needs and requirements for an SOI. There are stakeholders both internal and external to the organization including customers and user/operators” (INCOSE, 2015).

Stakeholder needs are determined by requirements discovery via elicitation. Vehicles for obtaining needs may come from and are not limited to customer product preference surveys, customer satisfaction surveys, warranty information, customer complaints/suggestions/concerns and focus group findings.

Potentially the most critical need is the vision statement, which defines the intended change for the current state (indicative) to a desired state (optative). The vision statement also defines a goal and not how the goal is achieved and is a guidance though the entire development process of the product. This vision statement also provides the context for the requirements and if the context changes, so do the requirements. The use of operational scenarios and concept of operations further help progressively elaborate the stakeholder need and determine the operational environment and context, define critical system parameters, expected operational hours, identify interfaces and constraints. Stakeholder needs are:

- A single statement to drive the subsequent goals and requirements.
- Should relate to the problem the system is to solve.
- Should not relate to a solution to the problem the system is to solve.

Examples of needs are demonstrated below:

- “Business wants the rig to maintain angular offsets so that drilling operations offshore are completed successfully.”
- “Business wants the rig to sustain torque capabilities so that drilling operations offshore are completed successfully.”
- “Business wants components to be of American Petroleum Institute (API) and Bureau of Safety and Environmental Enforcement (BSEE) quality in terms of material properties.”

4.7.2. Goals

A goal is an intention regarding the objectives of a system. Goals are refinements of a need, be that from a stakeholder in terms of business needs, engineering needs, product needs, etc. (IEEE, 2018). Documentation of the stakeholders’ intention can be done through a progressive set of goals and sub-goals. Goals are meant to be broad and qualitative and are an elaboration of the need that demonstrates a certain set of expectations for the system. Specifically, not quantitative as that progressive elaboration is a function of further decomposition in which requirements are defined. The syntax of the goal includes the word “should,” uses active voice without modal verbs and are written atomically. Goals justify the requirement and can also nullify any potential requirement.

Goals address the issues that are critical to the project and are further decomposed via scenarios, which illustrate the ability to achieve goal satisfaction or demonstrate the dissatisfaction of a goal. The benefit of coupling goals with scenarios is that goals promote the definition of scenarios and classify them (further development defined in *Section 3.4.3 Scenarios*). Types of goals are defined as and not limited to functional requirements, physical requirements, reliability, resource concerns, manufacturing requirements and human factors. As a result of stakeholder needs are stakeholder goals, which are comprised of the following:

- A goal is refinement of the stakeholder need.
- Address an issue that needs resolution.
- Uses the word “should.”
- Can have sub-goals.
- Allow us to determine if a system can be achieved successfully.
- Documents intention of the stakeholders
- Can help define scenarios to support validation.

Examples of how goals are written are demonstrated below:

- “The system should contain the working pressure of the vessel to maintain pressure integrity.”
- “The system should meet BSEE regulations so as to be operable offshore.”
- “The system should support the entire weight of the payload.”

4.7.3. Scenarios

A scenario is a possible development or sequence of events that describe the satisfaction or dissatisfaction of a goal by defining the concrete steps and relational aspects to the system context. In relation to the goal, scenarios illustrate the value of the system by providing greater detail about the goal and/or subgoal. Scenarios can be used to tell the story of how the system could be used as they convey the flow of events via context information such as but not limited to pre-conditions, conditions before execution of the scenario; post-conditions, conditions after execution of the scenario; information on concurrent activities, activities that happen at the same time that may be difficult to discern without scenario elaboration; actors, users or systems interaction with one another; roles, actors or class of actors engaging externally to the system; locations, actual or fictional setting where scenario is executed; and resources, preconditions relating to persons, information or other material needed to execute a scenario (Pohl 2010).

4.7.3.1. State Scenarios

Changing the current state to a desired state is much of what the transformation of EMU to xEMU. When using state scenarios to classify the current state, known as the indicative state, is the reality of the current capabilities whereas the desired state, known as the optative state, is the potential reality should the product in development come to fruition.

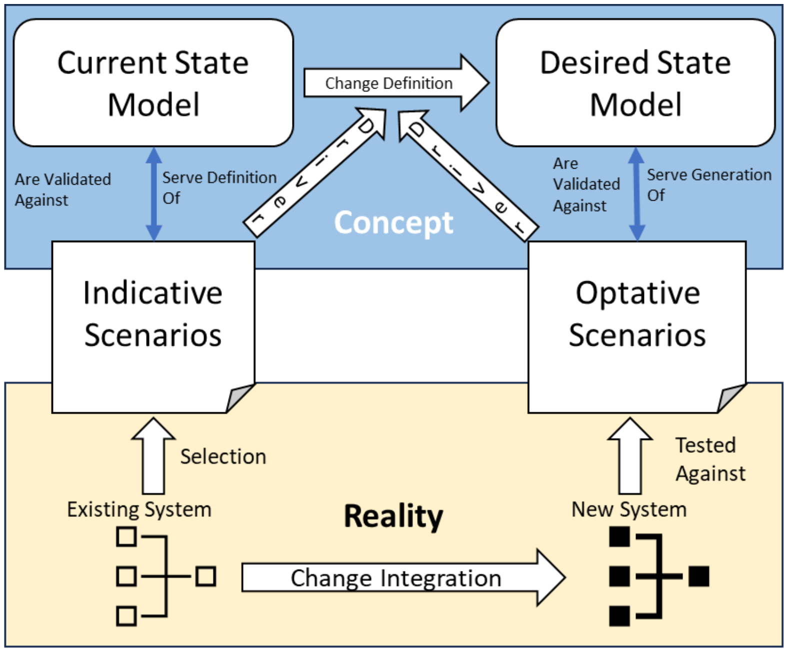


Figure 17 – Current State-to-Desired State Model

4.7.3.2. Positive & Negative Scenarios

Scenarios can be classified as positive or negative. These scenarios are primarily byproducts of their main, exception and alternate scenarios. This distinction derives from the satisfaction or dissatisfaction of a goal. Positive scenarios indicate a sequence of activities that identify the satisfaction of a goal. Negative scenarios indicate a sequence of activities that identify the dissatisfaction of a goal. Both positive and negative scenarios are complimentary. The following is an example of a positive and negative scenario paring:

- "The rupture disk of the casing hanger running tool activates at the burst pressure of 500 PSI."
- "The rupture disk of the casing hanger running tool fails to activate at the burst

pressure of 500 PSI."

4.7.3.3. Misuse Scenarios

When distinguishing between intended usages, misuse scenarios are classified as those that go against the stakeholders wishes and represent the misuse by a hostile actor. An example of a misuse use scenario would be the following:

- "Brian, the little brother of Karen, intentionally inputs the wrong password three times in succession on Karen's phone to lock her out of her phone."

4.7.3.4. Descriptive, Exploratory, Explanatory Scenarios

In certain decompositions of requirement elicitation, it's critical to illustrate the meaning of goals and requirements, justify and explain interactions and explore alternative realization and scenarios. In such cases, descriptive, explanatory and exploratory are instituted to enrich the decomposition. Descriptive scenarios illustrate the intended meaning of requirements and goals and as such can demonstrate innovative ideas. These scenarios describe the workflow or internal processes driving the scenario. A qualitative scenario, exploratory, supports decision making. These scenarios explore and document alternative realizations and solutions. A quantitative scenario, explanatory, explains and justifies different interactions. These scenarios benefit the project as they provide background information, especially delivering value for those who are not part of the effort directly.

4.7.3.5. Instance, Type, Mixed Scenarios

Instance and type scenarios allow for a combination of practices to be employed during the development of a goal-decomposed scenario. Both of these approaches can be combined into what is known as a mixed scenario. Instance scenarios describe definite sequences of interactions between actors and environments, whether the sequence is envisioned or currently in existence (progressively elaborated in *Section 3.4.3.1 State Scenarios*). Type scenarios are not defined in terms of concrete inputs and outputs from specific actors and sequences of interactions. Examples of the following scenarios are:

- Instance: Robert, the astronaut, while completing an extravehicular walk on Node 3 of the International Space Station needs to inject more oxygen into the ventilation loop of his suit. Robert flips the Primary Oxygen Release switch on his Display & Control Unit. On his Graphical User Display, he clicks “Yes” when prompted, “Are you ready to inject oxygen into your space suit?”
- Type: Astronaut while on a spacewalk needs more oxygen. He flips the switch on the control unit and clicks “Yes” on his interface panel.
- Mixed: Robert, the astronaut, while completing an extravehicular walk on Node 3 of the International Space Station and needs to inject more oxygen into the ventilation loop of his suit He flips the switch on the control unit and clicks “Yes” on his interface panel.

4.7.3.6. Type A, B and C Scenarios

When writing scenarios, the context in which the scenarios is written builds the framework for which the subsequent requirements are developed. These scenarios help to document the important data (i.e., essence of the requirements) particularly the context of the environment. Type A, also known as system-internal scenarios, focus entirely on the interactions internal to the system and only those interactions within the system boundaries. These interaction is intra-system exclusive amongst components. Type B, known as interaction scenarios, focus on the sequence of interactions between the actors and the system, also considering any and all Type A interactions. Type C, also known as context scenarios, convey information relating to the system and the context. Type A and Type B scenarios are derived from Type C scenarios. Requirements engineering produces more favorable decomposed requirements when Type C scenarios are used as they are rich in detail.

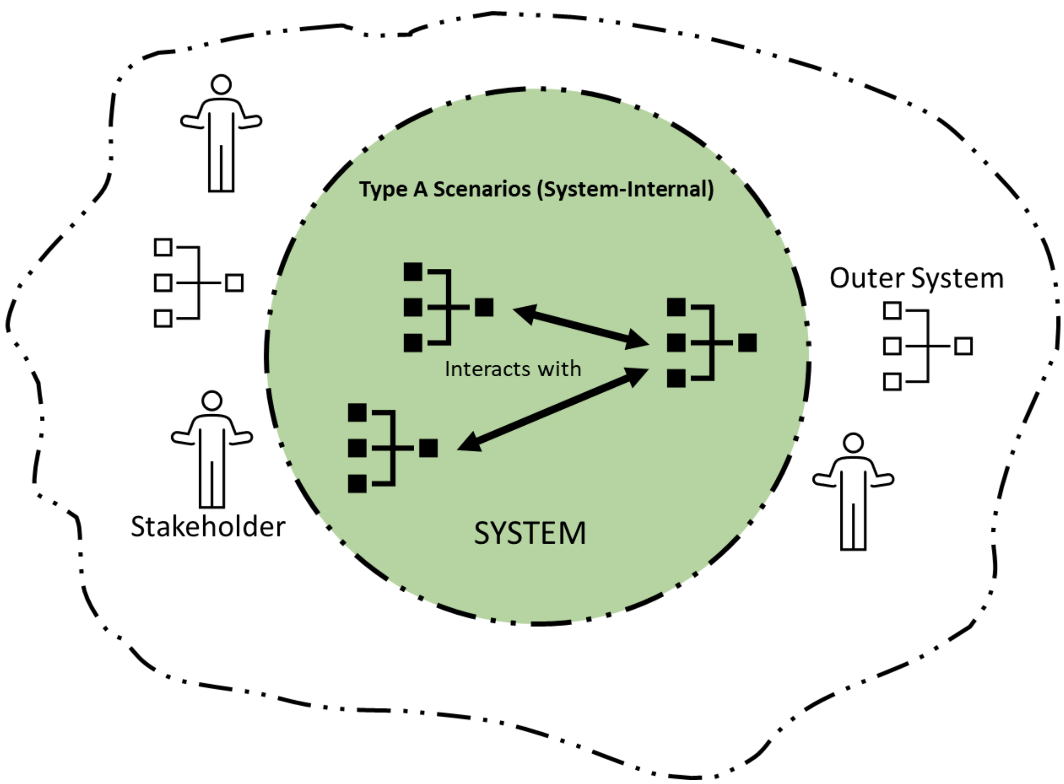


Figure 18 – Type A Scenario Diagram

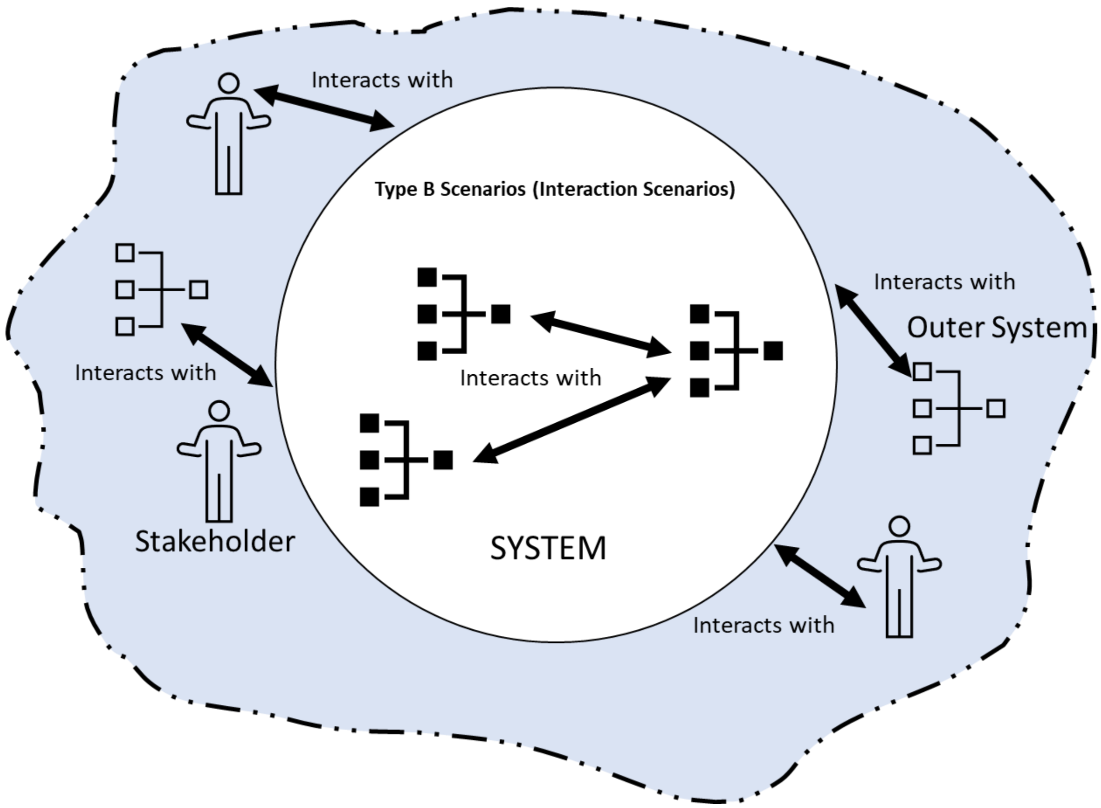


Figure 19 – Type B Scenario Diagram

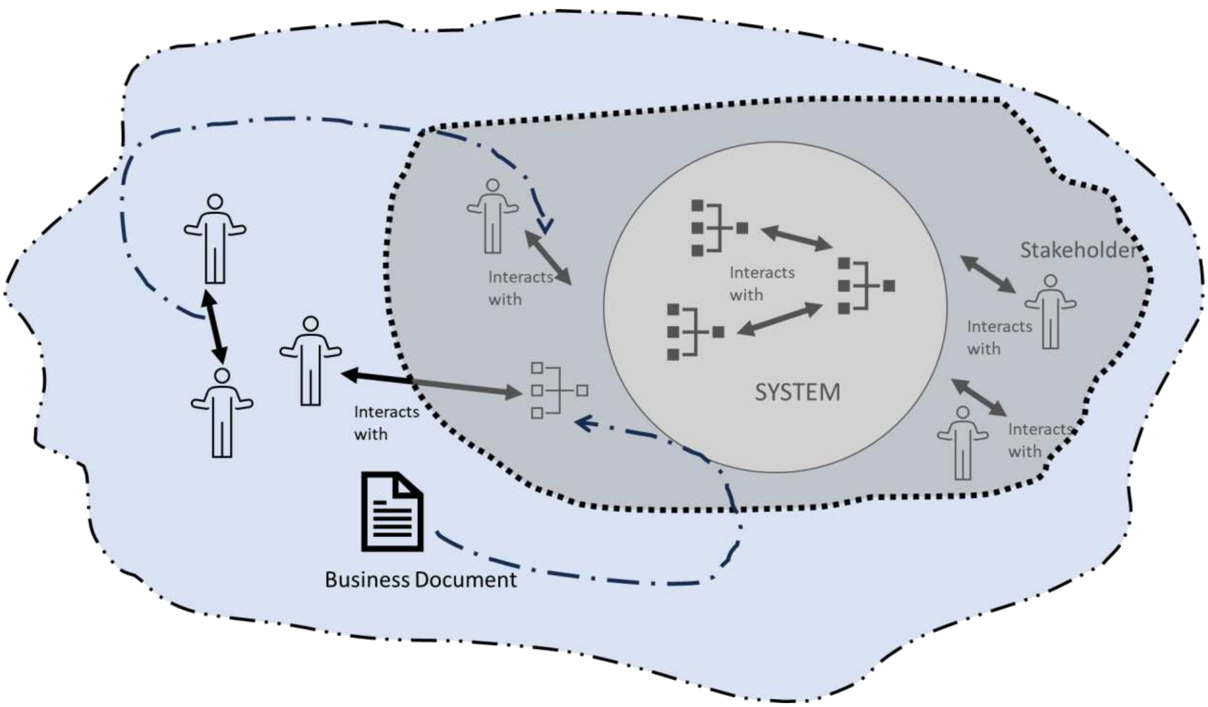


Figure 20 - Type C Scenario Diagram

4.7.3.7. Main, Alternative and Exception Scenarios

The hallmark of the main, alternative and exception scenarios is their inclusion in a use case, which is a specific sequence of actions and is not limited to the main satisfactory method of fulfilling a goal. These uses cases also take in variants, those being the alternative, which still satisfy the goal and exception scenarios, those that dissatisfy the intended goal. The use case contains context information to set the stage, which can include and is not limited to the preconditions, postconditions, roles, actors and location of where the use case is executed. Main scenarios demonstrate the satisfaction of the goal and detail those interaction steps in exactly one scenario. Alternative scenarios also dictate the successful completing of a goal but are secondary and surrogate to the main scenario. Exception scenarios are instances where the goal is dissatisfied and like main and alternative scenarios are atomic in nature.

4.7.4. System and Subsystem Requirements

INCOSE has defined writing requirements as a means to standardize an approach for proper requirements writing. In their *INCOSE Guide to Writing Requirements 2019,"* Since English has many synonyms and words with slightly different shades of meaning based on context, the use of natural language to communicate needs and requirements can make it difficult to be clear, precise, and to avoid ambiguity. However, even though natural language can be an imperfect way of expression, textual forms of communication remain the only universal means of expression that covers the wide variety of concepts that must be communicated throughout a system lifecycle." (Ryan, et al, 2019). During the process of requirement decomposition, natural language as specified by INCOSE has the key benefits of being universal, comprehensible and flexible. Natural language has evolved naturally as a method for human communication. However, the intermingling of perspectives is a potential disadvantage. These disadvantages include ambiguity, semantical and lexical misinterpretations and under-specification wherein the details are not documented accurately. The goal of avoiding these potential, erroneous requirements is the complement of modeling languages, modeling tools and modeling methods, as specified in *Section 3.1: Model Based Systems Engineering.*

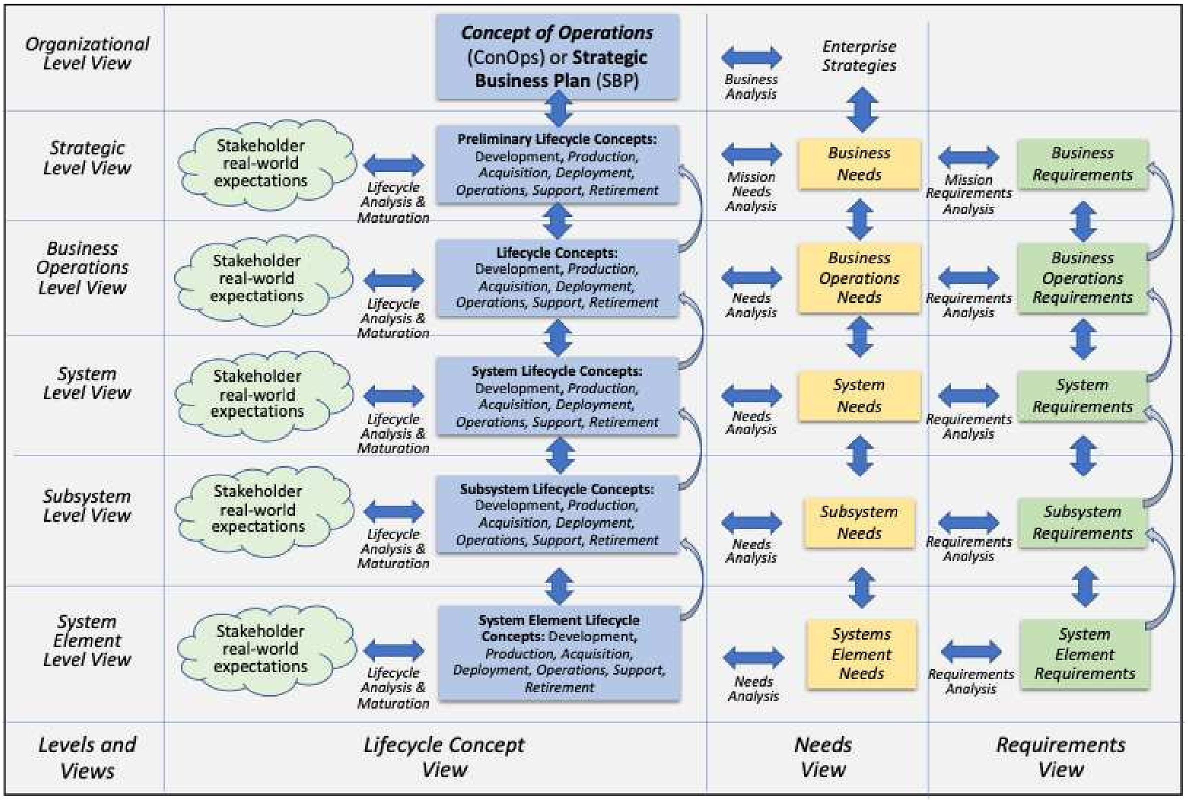


Figure 21 – Concept Driven Needs & Requirements Process

IEEE dictates a requirement as a capability or condition decomposed from a need (business, customer, user) intended to resolve a problem or fulfill an objective or a condition or capability that a system, subsystem and/or set of components must fulfill to satiate a specification, standard or other formally imposed document (IEEE, 2018). Requirements are atomic, which is to say they represent a singular piece of standalone capability. These requirements identify a product or process, such as operational, functional quality requirement or documented constraint(s). Following the S.M.A.R.T acronym for goal requirement writing, goals must be specific, wherein the requirement is unambiguous and written clearly; measurable, wherein the requirement is written with a precise intention with quantifiable parameters; attainable, wherein the requirement represents an optative, incarnate system requirement (one diametrically opposed to an essence system, where technology to fulfill the requirement is not viable); realistic, wherein the requirement maintains feasibility to be fulfilled; and time-tabled, wherein the requirement is dictated to be fulfilled within a specific frame of time (Doran, 1981). Artifacts of the systems requirements build include use cases, design documents, process diagrams, use case diagrams, and one or more of the nine systems engineering modeling diagrams.

Requirements are necessary, and only include capabilities the customer is willing to be contractually obligated to be financially responsible. If the customer indicates the system would be adequate without the requirement, then the requirement in question is no longer a suitable requirement. Understanding interfaces between requirements and system context is critical as emergent behavior. Gonzalez et. al. discovered that emergent behavior was shown in over 37 studies to not be observed in risk mitigation in any System of Systems (SoS) however existed nonetheless after Gonzalez and their team employed metrics to observe behavior between interfacing systems (Gonzalez et al, 2019). An approach should be taken to observe the problem vs. the solution in requirements engineering. The system vision describes the “what” whereas the “how” describes the specified requirements.

Requirements need to be defined at a system level first and decompose or allocate throughout the hierarchy of different levels, starting with Level 1 (system), Level 2 (subsystem) and Level 3 (unit/component). To maintain robusticity, system level requirements are built upon the scenarios, which are the byproduct from the goals and sub-goals which are the byproducts of the needs. At this point, constraints need to be identified and formally justified. During this phase of requirements engineering, realistic and noncompeting requirements with full traceability to a source that are deemed critical to the system and not desired are only acceptable. It is appropriate, however, to define requirements that are not fully defined as non-functional. These non-functional are requirements that are either underdeveloped, quality requirements or requirements that should not be requirements. The *INCOSE Systems Engineering Handbook, Version 4* further dictates that requirements, "establish critical and desired system performance – thresholds and objectives that are (1) critical for system success and (2) desired but not crucial to meet critical parameters" (Wiley, 2015). The following are used to define the system requirements adequately:

- Written in proper grammatical structure.
- Written in proper sentence structure.
- Supported by and not limited to verification, performance, and rationale.
- Supported by context information.
- Supported by traceability to source.
- Supported by priority against other requirements.
- Uses active and not passive voice.
- Uses the word "shall."

4.7.5. Requirements Management Plan

The requirements management cover several facets: the Requirements Management Plan document, requirements traceability approach and change management.

4.7.5.1. *Requirements Management Plan Document*

To maintain the integrity of the requirements database, requirement management must be established to document the necessary information required to effectively manage project requirements from definition, through traceability, to delivery. The Requirements Management Plan is created during the Planning Phase of the project. Its intended audience is the project manager, project team, project sponsor and any senior leaders whose support is needed to carry out the plan. The timing of the workflow and activities are iterative and recursive. Responsibilities for the following activities include: (1) Project Engineer and Systems Engineering for Requirements Management and Development, (2) Project Management and Customer Representative for Validation and Verification Efforts, (3) Full Development Team for Continuous Systems Analysis and (4) Systems Engineering and Integration Team for Architecture Design and Definition. Technical integration is facilitated by application of tools, an environment and infrastructure which are consistent and fully compatible across the portfolio of projects to provide a platform which is predicable, teachable, and repeatable.

The objective of technical requirements definition is to establish a benchmarked approach to requirements management by implementing the system of proper documentation, elicitation, negotiation, validation, and management. This methodology extends to the definition, formatting and implementation of needs, goals, subgoals, system requirements and subsystem requirements with scenario and sub-scenario inclusion with traceability to a source, including verification and validation activities.

The method of requirements traceability is prudent for creating a robust system engineering requirements list to promote a quality product for the customer. System/subsystem relationships are captured in Cradle® and cross referenced against goals, sub-goals and linked to a business need. Ultimately, traceability to the source brings justification of the requirement and completeness to the product during composition and decomposition activities. These requirements need to be associated with an identification number, an associated identification number, technical assumptions and/or customer needs, functional requirements, status, architectural design/documentation, technical specification, system components, software modules (if applicable), test case number (if applicable), test in (if applicable), implemented in (if applicable), verification, validation, and additional comments (if applicable). The objective of the workflow and activities are to consistently, iteratively, and frequently to perform appropriate requirements management engineering in accordance with proper documentation, elicitation, negotiation, validation, and management activities as shown below.

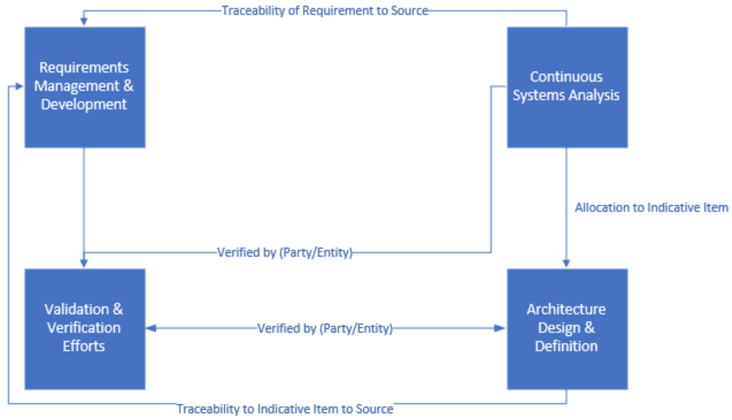


Figure 22 – Requirements Management Process

4.7.5.2. *Requirements Traceability*

The method of requirements traceability is prudent for creating a robust systems engineering requirements list to promote a quality product for the customer. Ultimately, traceability to the source brings justification of the requirement and completeness to the product during composition and decomposition activities. The approach if followed will enable the project team to ensure that the project delivers the requirements exactly as specified (Pohl, 2010). Specifically, these requirements are associated with:

- an identification number.
- an associated identification number.
- technical assumptions and/or customer needs.
- functional requirements.
- a status.
- architectural design/documentation.

- technical specification.
- system component.
- software modules (if applicable).
- test case number (if applicable).
- test in (if applicable).
- implemented in (if applicable).
- verification.
- validation.
- additional comments (if applicable).

4.7.5.3. *Change Management*

Change Management ensures that any changes to a baseline are identified, evaluated, recorded, approved/disapproved, implemented and verified and describe how requirement changes will be assessed and agreed upon. Change Management takes place the Change Control Board (CCB) (Force, 2005). The CCB shall determine when and if a requested change should be made. The CCB will meet bi-monthly or otherwise when directly requested and provide a platform for Project Management to present any baseline changes as they apply to requirements management. The CCB may either approve the requested change, reject the requested change or request an action item to the requestor to provide additional information for conferring at a later date. Before meeting with the CCB, the requestor must:

- Have held a formal meeting with the customer and documented the negotiated proposed change in accordance organizational process assets such as a change control document to initiate a Change Request (CR).

    - All applicable contractual processes, activities and constraints must be identified before meeting with the CCB.

- Each identified action must have traceability to the source document, author and rationale indicating the need for the change, whether it is an additional requirement, addition to requirement or removal of a requirement.

Membership of the CCB consists of and is not limited to an Engineering Director, Change Control Manager, Branch Chief and a representative of the Systems Engineering Department. The procedures for change in requirements are documented as:

- A change is presented in a CR in accordance with organizational process asset such as a proposed action(s) for change control document at a regularly specified CCB meeting.
- The CR is added to the Change Control Log (CCL) and ranked according to priority.
- The CCB will inform the CR requestor of the date of their CCB hearing.
- The CCB will determine if an in-board (formal meeting with the CCB) or an out-of-board (informal meeting in which a presentation per the Proposed Actions for Change Control document detailing the presentation template needed for said meetings) will be held.
- The CCB will then decide on an action to either (1) approve, (2) deny or (3) request additional information to the forum.

4.7.6. INCOSE Manuals

Requirements engineering focuses heavily on INCOSE standards and as such, there are four guides which provide unique insight into the development of needs, requirements, verifications and validations. Those manuals are (i) the Guide to Writing Requirements (GtWR), (ii) the Guide to Needs and Requirements (GtNR), (iii) the Guide to Verifications and Validations (GtVV) and (iv) the Needs and Requirements Manual (NRM).

4.7.6.1. *The Guide to Writing Requirements*

The Guide to Writing Requirements (GtWR) is an INCOSE publication with the purpose and scope of providing practical guidance and cross examination of well-formed needs and requirements statements (Ryan, et al, 2022). This clarity supports the definition of requirement and requirement sets and need and need sets by establishing a set of rules and standards for entities to regulate against. The GtWR further focuses on how to express the needs and requirements statements precisely using natural language in a structure that supports implementation and analysis, independent of any systems engineering tool that historically has been used to manage and capture those needs and requirement.

4.7.6.1.1. *Requirements Attributes*

To create structure, organization and differentiation between various parameters, attributes allow for requirements across different hierarchical positions to be chronicled and categorized by numerous identifiers. Categories include and are not limited to the identification, documentation aspects, content aspects, negation aspects, management aspects, context relationships and management aspects. In this respect, all attributes should possess definition, a range of values, definition of values overall definition and a naming schema (Pohl, 2010). Additionally, attributes may categorize requirements to provide information on status, stability, source and quality. INCOSE defines attribute categories to include rationale, system of interest, verification and validation approaches, trace to parent and/or source, condition of use, states and modes, allocation, approval date, date of last change, stability and person of ownership. Attributes may include requirement categories as follows (Ryan, et al, 2022):

- System: the highest level of requirements in the system or product.
- Functional: defines what the system should provide and likewise what it should not provide.
- Performance: how well the system needs to perform the functions anticipated by the customer or user.
- Quality: properties that the system should contain in order to execute its function; properties pertaining to system, component, a service or function.
- Constraint: any requirement that restricts the method in which the system shall be developed; includes organization, technical and project constraints.
- Operational: properties that define how the system should operate in terms of product and/or service.
- Physical: properties that define or characterize the physical makeup of the product and/or service.
- Design or Construction Standards: these include industry standards that

dictate federal regulations or best practices; may be necessary or optional depending upon the customer or federal regulation imposed.

Attributes are extended to include quality properties, specifically those non-functional requirements as follows (Wiegers, 2003):

- Availability: defines the percentage of time that a system is truly available for usage during nominal and off-nominal operations.
- Flexibility: defines how much effort is required to augment or how define how malleable the current system is to external change.
- Efficiency: a measure and or properties that classify how well the system needs to perform the functions anticipated by the customer or user.
- Integrity: properties that the system should contain to protect against any authorized or malicious access.
- Interoperability: properties that define how the system can exchange data in an intersystem relationship.
- Robustness: properties that classify the degree to which a system or component may function correctly in off-nominal operations.
- Usability: these include industry standards that dictate federal regulations or best practices; may be necessary or optional depending upon the customer or federal regulation imposed.
- Reusability: includes the extent to which a system component can be used in various, peripheral systems.
- Reliability: the extent in which the probability of the system executing

without any failures under a time-phased measurement.

- Maintainability: includes the ease with which the system maybe be able to correct or change a defect in the system.
- Testability: includes the ease in which the system may be able to be tested to find product defects.

4.7.6.1.2. Requirements Characteristics

The GtWR further details characteristics of both needs and requirements statements by providing rationale for the importance of the distinction of characteristics. Whereas requirements attributes are a ways to categorize and organize, requirements characteristics are ways to measure requirements against a standard, in this case INCOSE. Characteristics could be classified as a means of quality measures. INCOSE defines the following characteristics when developing requirements and are extracted verbatim so as to avoid any disambiguation (Ryan, et al, 2022):

- Necessary: the need or requirement statement defines an essential capability, characteristic, constraint, or quality factor needed to satisfy a concept, need or parent requirement. If it is not included in the set of needs and requirements, a deficiency in capability or characteristic will exist which cannot be fulfilled by implementing other needs or requirements in the set.
- Appropriate: The specific intent and amount of detail of the need or requirement statement is appropriate to the level (the level of abstraction) of the entity to which it refers.
- Unambiguous: Need statements must be written such that the stakeholder

intent is clear. Requirement statements must be stated such that the requirement can be interpreted in only one way by all the intended readers.

- Complete: The requirement statement sufficiently describes the necessary capability, characteristic, constraint, or quality factor to meet the need without needing other information to understand the requirement.
- Singular: The stakeholder need, or requirement statement should state a single capability, characteristic, constraint, or quality factor.
- Feasible: The need or requirement can be realized within entity constraints (for example: cost, schedule, technical, legal, ethical, safety) with acceptable risk.
- Verifiable: The requirement is structured and worded such that its realization can be proven (verified) to the customer's satisfaction at the level the requirement exists.
- Correct: The need must be an accurate representation of the concept from which it was transformed. A requirement must be an accurate representation of the need from which it was transformed.
- Conforming: The individual needs and requirements should conform to an approved standard pattern and style guide or standard for writing and managing needs and requirements.

4.7.6.1.3. Requirements Accuracy Criteria

The GtWR further details accuracy of both needs and requirements statements by providing rationale for the importance of the distinction of accuracy when developing needs and requirements. As defined, accuracy are methods to help provide sentence structure with verb and object recognition, using the active voice, using accurate subject verbs, definition of terms (i.e., a glossary), using definite articles, accuracy in units, avoiding the usage of vague terminology, avoiding escape clauses, avoiding open-ended clauses, usage of correct grammar and spelling and avoidance of the word “not” in requirements building. Concision & Ambiguity both indicate that needs and requirement statements avoid superfluous infinitives and utilize a separate clause for each condition or qualification. With specific regards to ambiguity, requirement and need statements shall use correct grammar, spelling, punctuation, correct logical expression convention, avoid where applicable the use of the oblique symbol and "not" so as to promote disambiguation (Ryan, et al, 2022).

4.7.6.1.4. Concision & Ambiguity

Concision & Ambiguity both indicate that needs and requirement statements avoid superfluous infinitives and utilize a separate clause for each condition or qualification. With specific regards to ambiguity, requirement and need statements shall use correct grammar, spelling, punctuation, correct logical expression convention, avoid where applicable the use of the oblique symbol and "not" so as to promote disambiguation (Ryan, et al, 2022).

4.7.6.1.5. Singularity

Singularity with respect to needs and requirements supports the notion of writing a single sentence that contains a singular thought conditioned and qualified by relevant subclauses. In the context of sentence syntax, this idea is further promoted by avoiding combinators to preserve sentence singularity, avoidance of phrases that indicate purpose or reason for need or requirement statement, avoidance of parenthesis or brackets containing subordinate text, the enumeration of sets explicitly using a group noun for naming the set and supporting diagrams or models when the need or requirement statement is complex in its behavior (Ryan, et al, 2022).

4.7.6.1.6. Completeness, Uniformity & Modularity

For a need or requirement to maintain completeness, avoidance of pronouns or indefinite pronouns, headings and support explanation are recommended. For a need or requirement to maintain modularity, adherence to for a need or requirement to maintain uniformity of language, the GtWR indicates that consistent terms be utilized, acronyms be defined, the continued avoidance of abbreviations and adoption of a project-wide guide be utilized. Grouping related needs and requirements together and conforming to a defined structure or template for organization is recommended (Ryan, et al, 2022).

4.7.6.1.7. Other Categories

Other categories include conditions, which state applicability explicitly and specify a single condition per action; realism, which dictates avoidance of unachievable absolutes; uniqueness, which illustrates classification by type or category and expression of need or requirement once and only once; abstraction, including avoidance of stating a solution within the need or requirement; quantification, by providing specific measurable targets; tolerance, by providing defined quantities with a range of values (Ryan, et al, 2022).

4.7.6.1.8. The Guide to Needs and Requirements

The GtNR is an INCOSE publication with the intent of forming the backbone for the systems engineering System of Interest (SOI). This guide provides application guidance for the ideas and activities illustrated in the INCOSE publication Needs and Requirements Manual (NRM). The GtNR additionally allows for those to read, implement and verify that the SOI meets the requirements, validate that the SOI meets the needs in its intended environment of operation and validates that that SOI does not allow for negative impacts via the user against the system. The process follows a waterfall decomposition based on (i) an integrated set of needs that is transformed into (ii) design input requirements which is transformed into (iii) architecture and design which transforms into (iv) the design output specifications which are finally transformed into the (v) SOI (Katz, et al, 2022).

Appendix D of the GtNR provides checklists D1 through D4 which provide the sample need verification checklist, sample need validation checklist, sample requirement verification checklist and sample requirement validation checklist, respectively. Checklist D1 and D2 allow for the full verification and validation of activities of the needs, which later inform the requirement activities, checklists D3 and D4. The D3 checklist unites the GtWR by bringing context to the verification activities with associated characteristics of needs and requirements (i.e., C1 - Necessary, C2 - Appropriate, C3 - Unambiguous, C4 - Complete, C5 - Singular, C6 - Feasible, C7 - Verifiable, C8 - Correct, C9 - Conforming, C10 - Complete, C11 - Consistent, C12 - Feasible, C13 - Comprehensible, C14 - Able to be Validated).

The importance of these checklists is the manner in which contextualization of the GtWR category of characteristics with respect to needs and requirements allows for inspection of a need or requirement against and interrogative process. This approach would later support the establishment of the requirements engineering scorecard for robustness checks, alongside the case studies for xEMU against lunar dust and auxiliary lighting. These statements are derived from the GtNR Appendix D checklists with wording augmented to support the requirements engineering scorecard:

- Are the need expressions well-formed?
- The new requirement does not need to change existing designs.
- Need expression is well formed such that system will be validated to meet need?
- Integrated set of needs is complete?
- Needs are correct?
- Set of needs is complete?
- Set of needs is feasible?
- Integrated set of needs is feasible?
- Integrated set of needs is correct?
- What is necessary for acceptance has been defined and agreed to?
- The needs associated with interfaces are well formed to be validated to meet needs.
- Requirement statement follows template for writing requirements?
- Statement contains basic elements of: entity, what, how well and under what

conditions?

- Are entity names and function names consistent with system architecture model?
- Is the requirement constructed so that compliance can be determined by observing the behavior at the boundary of the entity?
- Is required traceability in place?
- Have the required agreements been completed, articulated by lifecycle state attributes?
- Are a sufficient set of attributes defined for the requirement considering the lifecycle state of the project?
- Are entity names, function names, terms and units used consistently throughout the set?
- Do we have a complete set of system functions?
- Are all requirements traceable to one or more needs, parent requirement, or source?
- Do lifecycle analysis and maturation records exist that justifies the transformation of a need to one or more requirements that will result in a system that will sufficiently satisfy the need?
- Do all functional/performance requirements trace to a function allocated to the entity?
- Do system analysis records exist that confirms technical feasibility of the requirement to an acceptable level of risk while considering project scope and schedule?
- Has the set of requirements been matured through the proper reviews and

agreement processes according to business rules considering the lifecycle state of the project?

- Are all interactions across all interface boundaries represented by an interface requirement? Have all interactions been defined and referenced within the interface requirements?
- Do requirements sufficiently express the intent of the needs from which they were transformed considering the foreseeable set of operating scenarios (including rainy day scenarios, what variation will be seen on interfaces)?
- Are the requirements written in a language understood by the developing organization?
- Does the set of requirements contain any conflicting or inconsistent requirements?

4.7.6.1.9. The Guide to Verification and Validations

The Guide to Verification and Validation (GtVV) provides the user with the ability to help promote practical guidance for successful verification and validation activities across the system lifecycle of activities in addition to supporting the clarifications of misunderstandings of the verification and validation activities. A proper distinction among the planning, defining, execution, reporting and approving of verifications and validations must be adhered for successful activities for verification and validation. When planning for activities, it is recommended to define the success criteria so that a requirement can be tested and quantified during its development. This also aids in supporting the GtWR category of Tolerance, which dictates the relative accuracy against how success criteria can be defined with upper and lower bounds. Success criteria can be defined by avoiding non-verifiable statements (i.e., "The driver interface is user friendly") and should be driven by the need, design inputs requirements, design output specifications, organization design guidelines and acceptance/certification/qualification requirements (Katz, et al, 2022).

The degree of verification and validation criteria can be addressed by understanding the desired or negotiated confidence level, the expected system lifetime performance, tolerance justification and ranges, accuracy and precision. Defining concludes with understanding the verification and validation method. These include but are not limited to inspection, a visual examination to verify or validate the product using measurement as the primary datum; demonstration, the method by which a qualitative determination as opposed to a quantitative measurement helps derive the functional characteristics of the product by observation; test, by which verification and validations can be made my direct measurement of measurable characteristics; analysis, which includes but are not limited to an array of highly quantitative analyses including engineering analysis, simulation, modeling, sampling, etc.

Execution and reporting then implements the aforementioned steps by performing the verification and validation procedures and providing information of those activities to the customer, respectively. These will allow the user to receive formal verification or validation against an activity. In the event that verification or validation is not proven to be successful against the specified success criteria by selected method (i.e., inspection, demonstration, analysis, test), discrepancies and non-compliances will be issued. In the event of a discrepancy and non-compliance, customer and performing organization may request for variance concession or non-compliance disposition for modifying, alternating, scraping or accepting the discrepant product if form, fit, function and safety are not jeopardized.

4.7.6.1.10. The Needs and Requirements Manual

The Needs and Requirements Manual (NRM) provides systems engineering lifecycle concepts with respect to needs, requirements, verification and validation organization across the lifecycle of a system or product (Katz, et al, 2022). The NRM serves as the focal point for all associated INCOSE-related guides. The NRM informs the GtNR, GtVV, GtWR and various domain-specific guides by cross-refining the inputs from the INCOSE Systems Engineering Handbook (SE HB) and System Engineering Body of Knowledge (SEBOK). The NRM supplements and elaborates the INCOSE SE HB by providing a more detailed approach pertaining to the needs, requirements, verifications and validations across the system lifecycle.

The NRM manual provides two inputs important to the development of the requirements engineering scorecard to test for robustness. Section 5.1.2 and Section 7.1.2. include checklists which inform robustness of the need verification or validation, respectively. Section 5.1.2 and Section 7.1.2 ask the implementer questions, and the following questions were inspired and augmented to support the requirements scorecard:

- Were individual needs expressions manually verified and the sets of needs have the characteristics in accordance with the rules defined in the GtWR [19] or similar guide?
- Do the set of needs contain individual needs that are unique, do not conflict with or overlap with other needs in the set, and the units and measurement systems they use are homogeneous?
- Does the language used within the set of needs consistent and all terms used within the requirement statements consistent with the architectural model, project glossary, and data dictionary?
- If included in the project toolset, was an NLP application that provides the capability to automate the verification of the needs statements in terms of how well they adhere to the rules for writing needs and sets of needs utilized?
- Do individual needs expressions have the set of attributes defined and agreed to by the project team?
- Do individual needs expressions have the set of system validation attributes defined and agreed to by the project team?
- Was the project toolset to generate reports to confirm traceability of each need to one or more input artifacts (sources) used?
- Was the project toolset to generate reports to confirm each source shown in GNR Figure 4-12 have at least one derived need that addresses that source used?
- Has confirmation that the project has done risk assessments and for each risk that will be mitigated, the project has established traceability between the risk

and the lifecycle concepts that define a concept for mitigation of that risk and traceability to the need that addresses that mitigation concept been performed?

- Are there needs to address each of the interfaces and interactions across the interface and that each need that addresses an interface trace back to the source that identified that interface?
- Were individual requirements expressions manually verified and the sets of requirements have the characteristics in accordance with the rules defined in the INCOSE GtWR or similar guide?
- Do the set of requirements contain individual requirements that are unique, do not conflict with or overlap with other requirements in the set, and the units and measurement systems they use are homogeneous?
- Is the language used within the set of requirements consistent and all terms used within the requirement statements are consistent with the architectural model and project data dictionary?
- If included in the project toolset, was an NLP application that provides the capability to automate the verification of the requirements statements in terms of how well they adhere to the rules for writing requirements and sets of requirements as well as checking for consistent use of terminology utilized?
- Do individual requirement expressions have the set of attributes agreed to by the project team defined?
- Do individual requirement expressions have the set of system verification attributes agreed to by the project team defined?
- Was the project toolset to generate traceability reports to verify each

requirement traces to the need, an allocated parent requirement, or a source from which it was derived utilized?

- Was the project toolset to generate traceability reports to confirm each SOI need, source, or parent requirement allocated to the SOI has at least one derived requirement that addresses that need, parent, or source utilized?
- Were all interfaces addressed and the associated interface requirements included in the requirement set?
- Is there clarity regarding specific interactions between the SOI and the external system, and that the requirement includes a pointer to where that interaction is defined, recorded, and agreed to?
- Does the external system referred to have a corresponding interface requirement or does the interaction with the SOI being developed in its interface control documentation?
- Does the requirement properly address form, fit, function, quality, and compliance?
- Does each of the SOI requirements allocate to the next level of the architecture?
- Is each allocation correct and complete (i.e., the requirements were allocated to all applicable subsystems and system elements at the next level of the architecture and each of the allocations were to the correct subsystems and system elements)?
- Are the resulting dependent child requirements in response to allocations of performance, quality, or resources properly linked to manage changes to the

allocated/budgeted values?

4.7.7. Requirements Process & The Vee Model

As a result of an ever-changing landscape, the traditional waterfall method provided challenges in product delivery, both in scope definition and iterative and incremental deliveries to the customer. The 1980s offered a new challenge in the way of product development, specifically those that were software-intensive in their builds. The waterfall method had been effective in product development prior to the advent of software-intensive platforms. Waterfall represents a more top-down approach with steps that can generally follow the well-established initiation, requirements gathering, design, testing and acceptance framework. The shift to a more successive approach of incremental and iterative delivery found its roots in the mid-1980s with the spiral model. This model involved the continuous mode for development by iteratively examining strategies, validation methods, objectives and goals. The early 1990s demonstrated a shift to the "Vee" model which reflects both a top-down and bottom-up approach to the development of complex systems (Blanchard, 2004). The IEEE defines a top-down approach as comparing an organization's process against more generally accepted, benchmarked processes where the bottom-up process differentiates from the to-down by taking the assumption that process change in an intrinsic factor driven by the goals, experiences and codified data from an organization. The argument that IEEE proposes is that utilization of both will lead to increased levels of (Thomas, et al, 1994). In software driven development, the perspective that may be approached is the software engineer is analogous to another design engineer who is also responsible for a work package regarding the system's functionality. As the various functions are allocated along the software work packages, the software engineer is appointed to implement and perform those specific functions in the software code. As a result, the software engineer would sit alongside their colleague and develop subsystems and components utilized the computer code as a tool rather than physical components, thus defined as the “Vee’ model (Blanchard, et al, 1990). Test-driven development, as suggested in the “Vee” model presented in a software-intensive product development, defines that in opposition of designing products and writing those tests to check provide acceptance on delivery (i.e., verifications), that products should be driven by those tests proactively at the initiation stages and iteratively revisited for form fit and function. (Crowe, 2018). Beck describes test-driven development as following the succinct series of steps: (1) develop a singular unit test illustrating the aspect of the program, (2) perform the test with the anticipation that the program would fail by virtue of lacking a specific feature or form of functionality, (3) write only the necessary code (i.e., elegance of written code) that could pass the test with the simplest framework, (3) refractor (i.e., optimization without adding any new pieces of functionality) code until it conforms with enough ability to meet the minimum acceptance criteria and (4) iteratively apply unit test as they accumulate throughout the project life (Beck, 2003).

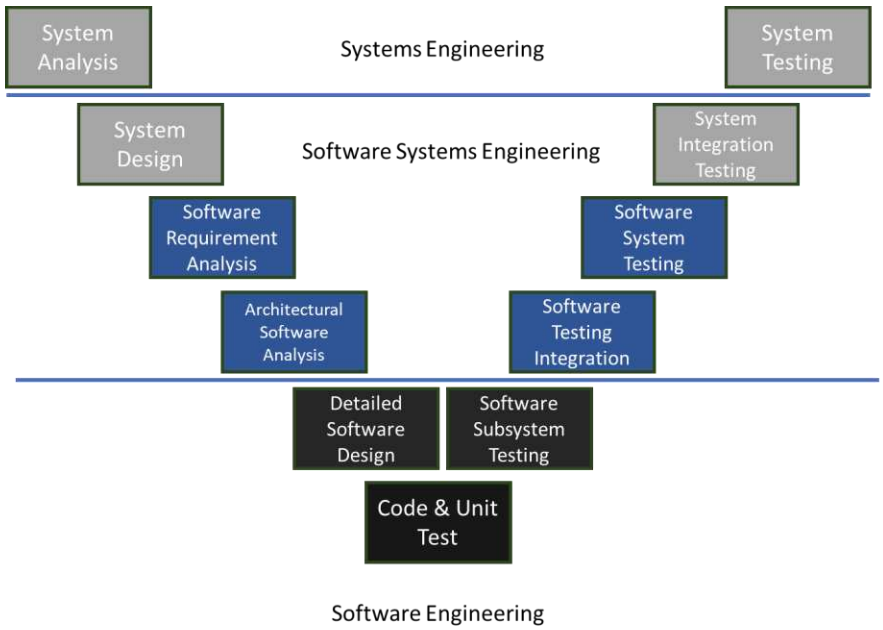
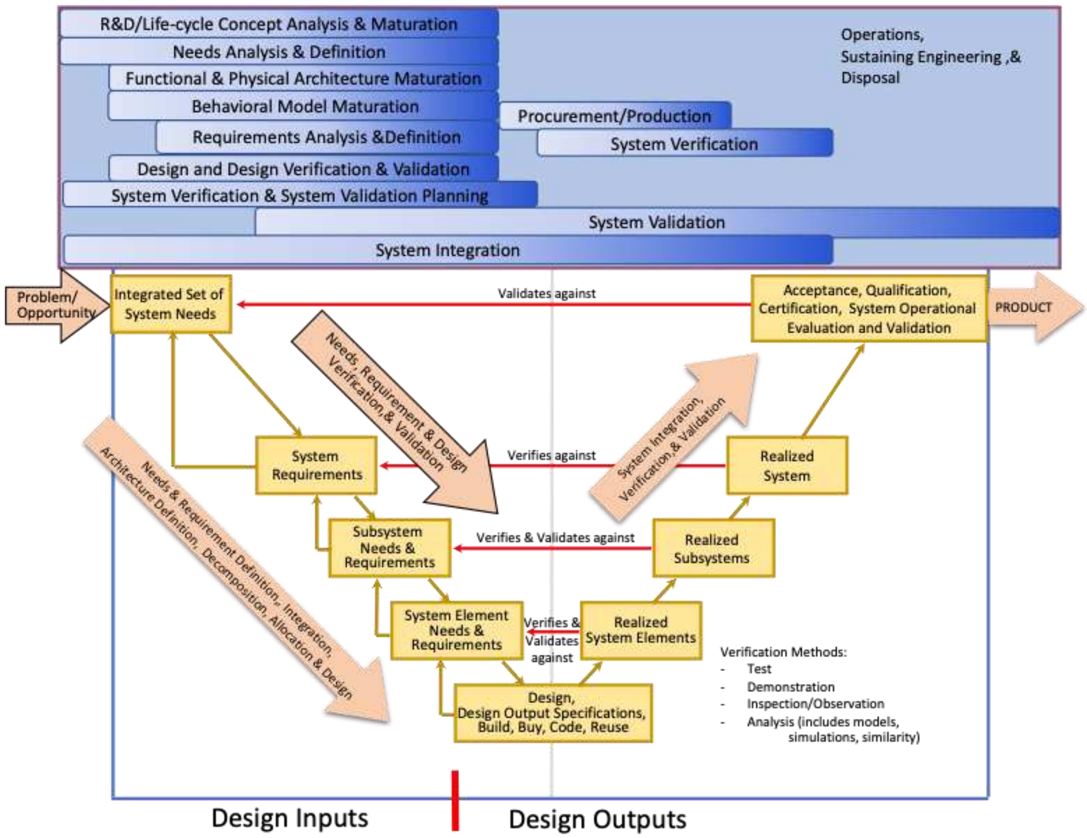


Figure 23 – Software "Vee" Model

These concepts in terms of test-driven development and the "Vee" model while having their origins in the software-intensively driven product development do have their application in the other areas, more specifically as it pertains to development of spacesuit software and hardware development (Kossiakoff, et al, 2020). The "Vee" model encompasses the breadth and depth of the sequential progression of plans, requirements and products with the impetus behind their drive and documentation via configuration management. The "Vee" model closely resembles the principles dictated in *Section 4.4.11 Verification & Validation* with the objectives in place to minimize project risks, improve quality, reduce the total cost over the project lifecycle and optimize communication across all stakeholders (INCOSE, 2012).



Adapted from M. Ryan, L. Wheatcraft, L.S., "On the Use of the Terms Verification and Validation", February 2017 and INCOSE SE HB, Version 4, Figures 4.15 & 4.19

Figure 24 – Engineering "Vee" Model

4.7.8. The Specification Tree

A specification tree allows for the complete scope of the product to be identified in a hierarchy so that requirement packages at a coarse level may be observed, documented and organized in a tiered relationship. These typically flow with the following tiered levels but may be augmented depending upon the product complexity: Level 0: Customer; Level 1: System; Level 2: Subsystem; Level 3: Component. These trees focus on parent/child relationships to help distinguish traceability and maintain structure. These tiers could be founded to include the highest level of customer desire, the need, starting at Level 0, with a progression of decomposed needs allocated to each of the sublevels commencing with Level 1 and proceeding until full decomposition of the tree is completed. The generation of a specification may take shape with regards to functionality or a product approach. The best approach involves the comprehensive management of the organization's boundaries and context in which the performing organization and client both operate. Depending upon the product, the approach defers but most often the preferred framework is to verify each box within the specification tree details a unique and atomic organization or work package that can be implemented and each set of requirements verifiable (Hood, et al, 2007).

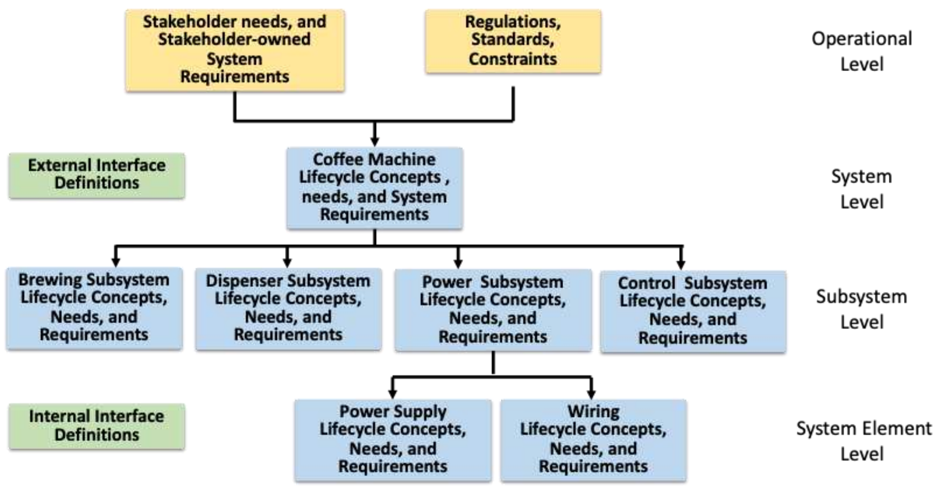


Figure 25 – System Specification Tree

A benefit from requirements organized into a specification tree are the ability to illustrate scope, prevent scope creep and maintain traceability across the lower-level requirements and provide verification data for individual requirements and how they relate to overall system verification and how these changes may be assessed for impact across the different levels of the product structure (Dick, et al, 2017).

4.7.9. Verification & Validation

Nomenclature variation between verification vs. validation has seen its ambiguity and as a result, product development has been hampered without proper identification and differentiation of each term. The Tacoma Narrows Bridge was a parametric scaleup from a previous design. Although the theory and usage were considered consistent, the system context and operating conditions were varied, as the force of wind variation was not considered. As a result, several crosswinds created instability and caused the bridge to fail. The design engineer developing the bridge had lifted the same requirements levied against the previous project. The methodology consisted of verifications based around the previous design. However, the validation method was erroneous as the system context and requirement need had changed (Bahill, et al, 2004). To prevent ambiguity, the terms verification and validation need a modifier preceded to the term to indicate the proper context of the usage. These subjects would be system, subsystem and component/unit. In short and in the context of xEMU, verifications would be the instances the performing organization checks their work to verify the engineering work packages were executed correctly whereas the validations would be the customer validating the product indeed meets the need. Verifications and validations according to INCOSE are well defined in each of the following phases: requirements, design and system delivery (Katz, et al, 2022). Ryan, et al define graphically the interrelationships across the project lifecycle. Validation asks the question at the requirement, design and systems levels, respectively (Ryan, et al, 2017):

Verification

- "Are the requirements written correctly?"
- "Did we design it correctly?
- "Did we build it correctly?

Validation

- "Are we building the right thing?
- "Do we have the right design?"
- "Did we build the right thing?"

A central concept in verifications and validations across the project lifecycle is that the process is continuous. This implies that continuous iterations of verifications and validations maintain stakeholder accountability by requiring standards and checklists, recursive training and enforcement by management. Benefits include prevention of reworks, reduction of reviews and monitoring and controlling of excessive or unneeded documentation.

4.7.9.1. *Verification*

Requirements verifications guides the ability for the performing organization to ensure the process of determining the degree of correctness and characterize those requirements against a standard, in particular INCOSE standards. The focus in requirements verifications will be the performing organization examining the wording and structure using checklists, guidelines and rules both governed by INCOSE and the performing organization (Katz, et al, 2022). The use of models including Earned Value Metrics (EVM) for costing and verifications using DOORS (i.e., in the case of xEMU Cradle®®) support traceability, completeness, check for inconsistency, etc. Studies have shown that of these tools utilized, only 15% of cases applied requirements verifications inspections models correctly, 35% of cases did not apply requirements verifications inspections models correctly and 50% of cases variably applied requirements verifications inspections models correctly (Fanmuy, et al, 2012).

Design verifications guides the ability for the performing organization to ensure the process ensuring the design meets the rules defined by the organization's best practices and/or industry standards. The focus will be the degree of correctness followed during the performing organization's design process. This includes preliminary and final design reviews, analysis throughout design and that the overall design reflects the systems requirements specifications in terms of what the customer needs (Katz, et al, 2022). Studies show that a large percentage of lifecycle costs are due to early design verifications, especially as they pertain to complex hardware and software intensive designs. Early verification of components and their functionality of systems are highly critical, especially at the project onset. Methods to mitigate risks associated with high costs due to reworks or inability to sufficiently capture scope in present avenues of functional mock0ups, early integration efforts and low design data-intense prototypes (Maropoulos, et al, 2010). Drechsler ascertains that, early detection, analogously as the proposed predispositions establish by Fanmuy, et al in requirements verifications, will save time and resources. The result of simulation-based verifications will lead engineers to spend more time creating stimulus and getting involved in overall verification and less on a creative design (Drechsler, 2018). Creative designs can lead to potential scope creep and as a result become derivative of the framer's intent of the previous ideation systems verifications. Drechsler proposes that an agile development movement, specifically in

Systems verifications are done after the requirements and design verifications and are focused on the manufacturing and coding process, but in terms of xEMU would be the functional build and checkout and fitness of use testing. Methods used to drive these verifications are inspections, tests, demonstrations and various analyses at the performing organization level prior to any customer acceptance or validation. Verifications at this phase include tests, inspection, demonstration and analysis (Katz, et al, 2022). Simulation has its role in system verification as scaling can be used to designs of virtually any size. With increased system complexity, the less complete a simulation may be to adequately model the system in terms of maintaining proper system verifications. Formal verification and simulations alone may not be adequate to accomplish success at the system validation level and therefore it is prudent to combine different approaches to serve the purpose of validation of diverse designs (Li, et al, 2010). A combination of functional checkouts of both hardware and software during the GSE building can be done two-fold to address Li's concerns. First, sprint-style reviews of LabVIEW code throughout the project lifecycle will be beneficial during system checkout, especially when paired with an ever-changing hardware build which these codes must support and maintain compatibility against. In summary, system verification will confirm that the designed and built system meets the requirements and thus fulfills the customer's needs.

4.7.9.2. *Validation*

Requirements validation is the customer side of confirmation that requirements will meet the stakeholder need and is expressed in a language understood by the performing organization. The focus then becomes by the customer to deliver a message that is clearer understood and agreed upon by the performing organization whereases the focus in requirements verification is the correctness, grammar, structure and organization of the requirement is in place (Katz, et al, 2022). Cimatti, et al review requirements validation for hybrid systems, indicative of the xEMU project, and propose that failure in requirements may have unacceptable results in development of safety-critical applications. The argument pertains to the cause of safety-related functional errors traced to issues at the requirements specification. This is specifically highlighted in the context of a hybrid system where controlling components interact with the physical environment via actuators and sensors (Cimatti, et al, 2012). This concept of sensor and actuator interaction in a physical environment as safety controls mirrors analogous components found in various locations of the xPLSS.

Design validation is a confirmation from the customer side post to the design verifications on the performing organization side that the design will meet the intended purpose of the operational environment. The focus becomes the assertation that the stakeholder expectations were captured as a result of the design set forth by the performing organization. Validation techniques in this phase include walkthroughs and checklists (Katz, et al, 2022). Performance engineering activities may be augmented to enhance design validation. These may include risk reduction activities such as prompt lists, qualitative and quantitative analyses and risk breakdown structure reviews to ensure a system can meet its nonfunction requirements. These can be done before the system is even in place. These include prototyping, modeling and simulation, and trade-off analysis during both pre-system acquisition and system acquisition (Metzger, et al, 2014).

Systems validation and happens post system verification and assures that the designed built and designed meets the intended purpose against the stakeholders' requirements. The focus will be on the full system and how well it conforms to these requirements. Tools at this phase are test, walkthrough and customer demonstration (Katz, et al, 2022). As the project lifecycle continuous moves through its "Vee" model transition, the system validation phase will validate against the Con-Ops to ensure a recursive and iterative fact check of the system requirements that were driven by the Con-Ops (Metzger, et al, 2014). As a result of system validation, requirements documentation may be updated with the actual results of validation activities of particular interest when the actual results yield better resulted than when a requirement had anticipated to perform in the system or where a requirement was waived (PMI, 2016). In summary, systems validation will confirm that the system designed is built and verified and fulfills the intended purpose in its operational environment.

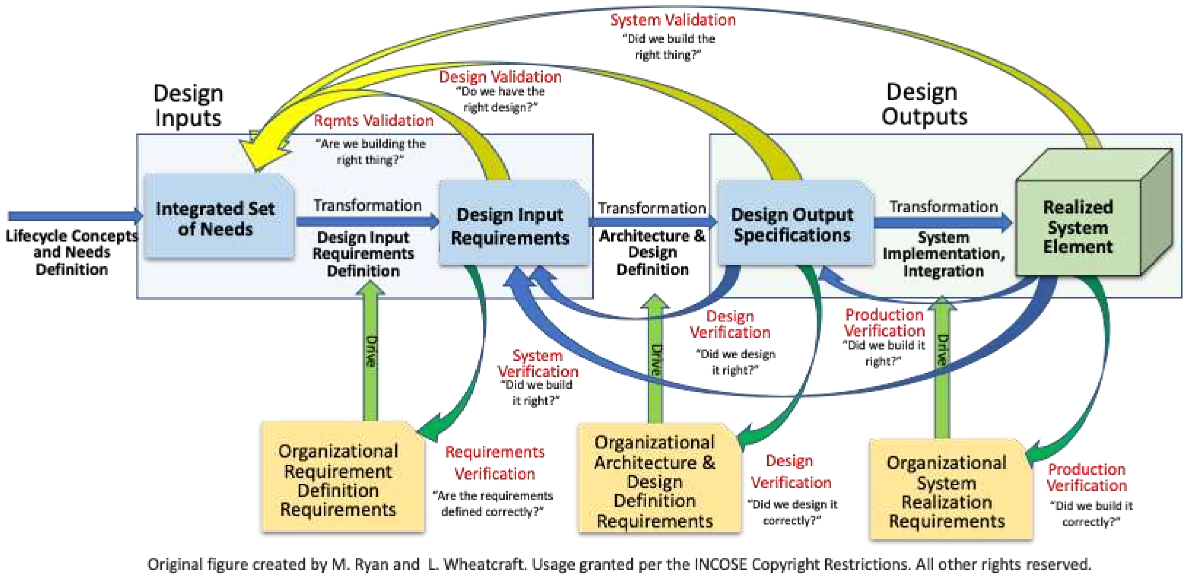


Figure 26 – Verifications & Validations through the Project Lifecycle

4.7.9.3. Requirements Engineering in Action

Carson identifies that structure and content identify how one can obtain correct and valid requirements. In supporting a rubric by which to develop requirements, the author establishes a requirement type (i.e., functional/performance) as it pertains to design/environmental/suitability and elements by which to define (agent, interface-input, shall statement, timing, function, design constraint, interface-output, characteristic, performance, condition, environment exposure, event trigger, duration). These requirement parameters should include context from developments regarding concerns, Con-Ops, missions, functions, inputs, outputs and performance. Validation should derive from analysis and simulation, should be validated by stakeholder approval and completeness is ensured by treating anomalous conditions, which would be parallel to the exception or "else" conditions in a use case scenario (Carson, 2021).

Carson initially patented the model to emphasize the need for requirements to be unambiguous and verifiable. Carson illustrates that a requirement may only be fully defined when a verification is complete and accepted by the requirement initiator. The patented model first has an operator (i.e., acquirer or supplier) input into the computer (i.e., user interface system) which prompts the user interface with a display and user input. The requirement generator allows for an identifier and format module to automate requirement building. Based upon the user inputs, a quality assessor will assign a quality level score, fed by quality criteria graded against an analyzer module. The requirement is then populated and fine-grained according to natural language, type and elements. This requirement or requirement set can feed into a requirements management tool. (Carson, 2015).

Wheatcraft, et al, described that even as far back as 2007, the NASA Inspector General identified that project's inability to fully define project requirements prior to entering a contract places projects at risk of significant budget and schedule overruns. The Constellation Space Suit Element was talked to produce an Element Requirements Document, the project requirements document, within a 3-month period. The Constellation Space Suit Element Project, briefed on the consequences derived from poor requirement set, instituted a continuous requirement validation process that allows for iterative and incremental corrections to allows for consistent and complete requirement sets. The result was an order of magnitude reduction in review comments on the Element Requirements Document when compared to its parent System Requirements Document. The process included training the team on how to develop requirements that are free of defects and that requirement correctness is dependent on continuous requirement validation before the requirement is included into the official Element Requirements

Document. NASA expressed praise of the requirement document as a result of this continuous improvement process (Wheatcraft et al, 2010).

One of the major facets to project success was taking the time to perform upfront planning and allow for initial investments to be made for upfront plans to most indicatively represent the plans that will potentially unfold. The study cites there are two types of requirements validations: continuous and discrete. Continuous requirements are applied continuously throughout the product life cycle, while others happen at discrete points, for example Critical Design Reviews, System Design Reviews, Test Readiness reviews, etc. Continuous requirement validation should be dictated by some form of checklist, particularly the "*NASA System Engineering Handbook, NASA/SP-2007-6105, Rev. 1, Appendix C, "How To Write A Good Requirement.*" While Wheatcraft communicates that gate keepers should manage and facilitate that good requirement processes be correctly applied, it is indeed the responsibility of the entire team.

In the context of how the requirement risk mitigation efforts unfolded, the team's schedule was added an extra week to ensure proper requirement quality. After the extension of one week, it was determined that all products met expectations and were good for System Requirement Review (SRR). The Kick-Off Meeting detailed the following steps to ensure requirement quality at the onset. The "Scrub" team is the first line of triage in the requirements quality checking process. The process follows the somewhat linear process detailed below:

- Subsystem Team: Draft Requirements
- Scrub Team: Edit & Clean Requirements
- Architecture Board: Review Technical Appropriateness
- Draft in Architecture Tool
- Configuration Management Approval
- Requirement Allocated to Architecture Tool Database

At the conclusion of the development, the result reduced an order of magnitude reduction on the review comments and Review Item Discrepancy (RIDs) against the suit Element Requirement Document when compared to the System Requirements Document of the same product. NASA management communicated that the requirements were "… the most comprehensive and of the highest quality they ever remember seeing." and the JSC Engineering Directorate Crew and Thermal Systems Division (CTSD) Chief stated, "I can't say enough about how amazed I am by this set of requirement documents. As far as I know, no other Constellation Program has allocated and decomposed anywhere near to this level of depth. You are the first. I have also never seen anything like these from previous programs." (Wheatcraft et al, 2011).

Hooks reported that studies conducted by NASA showed an average cost and schedule overrun and underperformance for approximately 65% of 29 programs. Furthermore, hooks details that the cost to fix a requirement increases exponentially as the project continues through the project lifecycle development. The takeaway is that finding and fixing defects cost is 10 times more expensive during product testing than fixing the requirement during the requirements phase (Hooks, 2001).

NASA released the *NASA Space Flight Program and Project Management Handbook* in February 2010. The handbook details project and program management best practices strategies from well-established managers with intent of providing emphasis of the importance of prudent requirement development. The handbook illustrates the following (NASA, 2010):

- "All acquisitions should start with a requirement definition that clearly identifies the Agency's desired outcome for a contract."
- "Establishing a good set of program mission/operation concepts that are evolved into a useful set of program requirements is one of the most critical products for program success."
- "The most common negative finding made by independent review teams is that a project did not place sufficient effort and importance on understanding and developing project requirements."
- "One of the greatest risks that a project faces comes from ill-defined requirements."
- "Poorly written requirements, incomplete requirements, and poorly written contracts result in cost overruns and schedule slips."
- "Managers need to be able to identify risks and add the mitigation costs to the program baseline. When risks are identified and the qualitative value assigned to the risk has been verified, the PM needs to act in the timeframe associated with that risk."

5. CHAPTER 5: CHANGE OF PROJECT DIRECTION

Lede: This chapter describes the Inspector General's Report on the development of xEMU, the introduction of a new industry partner and the shift from xEMU on the JETS program ushers in a new potential contractor. The importance for this chapter is to provide background information on why the case study was selected as a lessons learned as the project ending was an opportunity to collect information and capture the entire picture holistically.

5.1. Inspector General Report

In August of 2021, the Office of the Inspector General released an audit Report No. IG-21-025 titled *NASA's Development of Next-Generation Spacesuits*. The report indicated that," *NASA's current schedule is to produce the first two flight-ready xEMUs by November 2024, but the Agency faces significant challenges in meeting this goal. This schedule includes approximately a 20-month delay in delivery for the planned design, verification, and testing suit, two qualification suits, an ISS Demo suit, and two lunar flight suits. These delays—attributable to funding shortfalls, COVID-19 impacts, and technical challenges—have left no schedule margin for delivery of the two flight-ready xEMUs. Given the integration requirements, the suits would not be ready for flight until April 2025 at the earliest. Moreover, by the time two flight-ready xEMUs are available, NASA will have spent over a billion dollars on the development and assembly of its next-generation spacesuits. Given these anticipated delays in spacesuit development, a lunar landing in late 2024 as NASA currently plans are not feasible. That said, NASA's inability to complete development of xEMUs for a 2024 Moon landing is by no means the only factor impacting the viability of the Agency's current return-to-the-Moon timetable. For example, our previous audit work identified significant delays in other major programs essential to a lunar landing, including* *the Space Launch System rocket and Orion capsule. Moreover, delays related to lunar lander development and the recently decided lander contract award bid protests will also preclude a 2024 landing* (Martin, 2021)."

5.2. xEVAS Request for Information

According to the NASA xEVAS JSC Office of Procurement website, NASA stated that, *"NASA's interest is identifying interest from Industry in the provision of commercial EVA services wherein the agency relies upon the contractor to provide the full suite of services and equipment required to perform all activities and operations required to enable EVA capability for NASA's current and future missions. NASA's use of the term "commercial" is not meant to be confused with the term "commercial item" as used in the Federal Acquisition Regulations (FAR). NASA is in the process of gathering information to make a final commercial item determination and the responses to this notice and accompanying RFI are expected to inform that decision* (Gaspard, 2021)."

5.3. Project Closure of xEMU

As a result of the xEVAS new contract and industry partners bidding on said contract, the xEMU contract would be replaced with the xEVAS contract and thus xEMU would cease operations at the end of the 2022 fiscal year. The Inspector General report indicated that, *"In October 2019, NASA issued a Request for Information (RFI) to determine industry capabilities to fulfill future spacesuit needs. At that time, NASA intended to initiate a hybrid contract consisting of a single prime contractor for integration and multiple awards for development and sustainment known as the Exploration Extravehicular Activity Production and Services (xEVAPS) contract. However, after 18 months NASA canceled the xEVAPS RFI and issued a new RFI in April 2021 for the Exploration Extravehicular Activity Services* *(xEVAS), significantly altering its approach for future suit acquisition by purchasing services instead of equipment. As previously discussed, to date NASA has spent more than $420 million on spacesuit design and development, but the new xEVAS RFI gives industry the choice to either leverage NASA's designs or propose their own. Therefore, the extent to which NASA's investments will be utilized is unclear. Additionally, the xEVAS RFI does not stipulate that the suit be compatible with both the ISS and Artemis programs, a distinction that could result in industry developing (and NASA purchasing) two different spacesuits—one for use in low Earth orbit on the ISS and another for use on the lunar surface during Artemis missions. Given the Station's limited expected lifespan, developing a suit solely for the ISS may not prove cost effective* (Martin, 2021).*"*

6. CHAPTER 6: INSTANCES OF SYSTEMS ENGINEERING CHALLENGES

Lede: This chapter illustrates the case studies from an empirical evidence standpoint, outlining both the project lifecycle development and requirements engineering case studies. xEMU as a project and subsequent case study presents a bevy of challenges across all disciples, specifically in the field of systems engineering principles and its best practices. A unique opportunity presents itself during project closure as the breadth and depth of the project lifecycle is fully unfolded. As a result, project management and engineering are able to review lessons learned, examine past events and determine what major reflections could be incorporated to better accommodate future projects and similar work. In many instances, there is a general, subjective conclusion based on qualifiable data while in other instances, there are opportunities to model the objective data to present an analysis that is more robust than simple conjecture. Section 7 will examine and discuss the results and in applicable instances propose alternative solutions, quantify comparable data or find proximate answers for data that is not capable of being properly quantified. The information gathered in Section 6 was a function of interviews and questionaries with SMEs and systems engineers, surveys and focus groups with Development Team members and brainstorming sessions with the aforementioned as to the method of approaching the subjective nature of various challenges across the project. With regards to which work is done by the performing organization vs. novel to this dissertation, each case and sub case study is itemized below:

- All work reflected in *Sections 1-5* provide background information on project and dissertation approach that was collated and was a fully independent effort of the professional work.
- All work reflected in *Section 6* pertains to the collection of data, which was work done

outside the dissertation but its collection to inform the data in the dissertation was done fully independent of the professional work. This includes the project work chronicles on GSE, the lunar dust requirements and the auxiliary lighting requirements. The lunar dust requirements were fed directly into the requirements engineering scorecards in a forward pass approach while the auxiliary lighting requirements improvements that were done were performed by the engineers independent of the tool but reversed engineered with the intent of fine tuning the scorecard.

- All work reflected in *Sections 7-9* pertains to the novel work done as a result of the findings in *Section 6* and is fully independent of the professional work done on contract. This includes both prototypes to project lifecycle development and requirements engineering, all surveys, all tempered model testing, all users' manuals and templates, any publications and presentations done publicly and any outside testing with academic institutions.

6.1. Requirements Engineering Development Overview

The development of the requirements engineering will be studied across the three major subsystems of xEMU relating to xPGS, xINFO and xPLSS with heavier emphasis on the xPLSS subsystem. The plan was to smoothly decompose requirements from customer needs to eventual unit requirements, which in turn traces back to higher level requirements where verification and validation can be properly done against an atomic requirement and interfaces are more easily recognized as a result of decomposition. In addition to atomic requirements easily verified and validated, the project planned to also finely tune requirements so that allocated requirements may be more readily augmentable.

6.1.1. Requirements Organization & Specification Tree

One of the responsibilities of the SE&I group is to maintain full traceability to source and requirements decomposed atomically with full integration bottom to top with lower requirements contingent on higher level requirements with management of the interrelated processes. The xEMU system functions as a result of the three complimentary subsystems, xPLSS, xPGS and xINFO. Requirements are separated into 5 levels; Level 0 consists of agency and program requirements; Level 1 consists of EVA office requirements; Level 2 consists of project (system) requirements; Level 3 consists of subsystem requirements; Level 4 consists of end items (component) level requirements. Effectively, Level 2 requirements are used to derive the project or system requirements; Level 3 requirements are decomposed from the system requirements into subsystem requirements; Level 4 requirements are decomposed from subsystems requirement into unit/component requirements. The system requirements are then derived at Level 2 as a function of the Concept of Operations (Con-Ops), the Project Technical Specification (PTRS) and Architecture Description Document (ADD) which then influence the Level 3 and Level 4 specifications. While the primary customers of NASA are the Gateway, HLS and ISS programs at Level 0, the EVA office at Level 1 develops the standards and documentation to support xEMU project management at Level 2 to derive the Con-Ops, PTRS and ADD which drive the xEMU subsystem leads to the Level 3 which drive the component owners responsible for the End Item Specification (EIS) at Level 4.

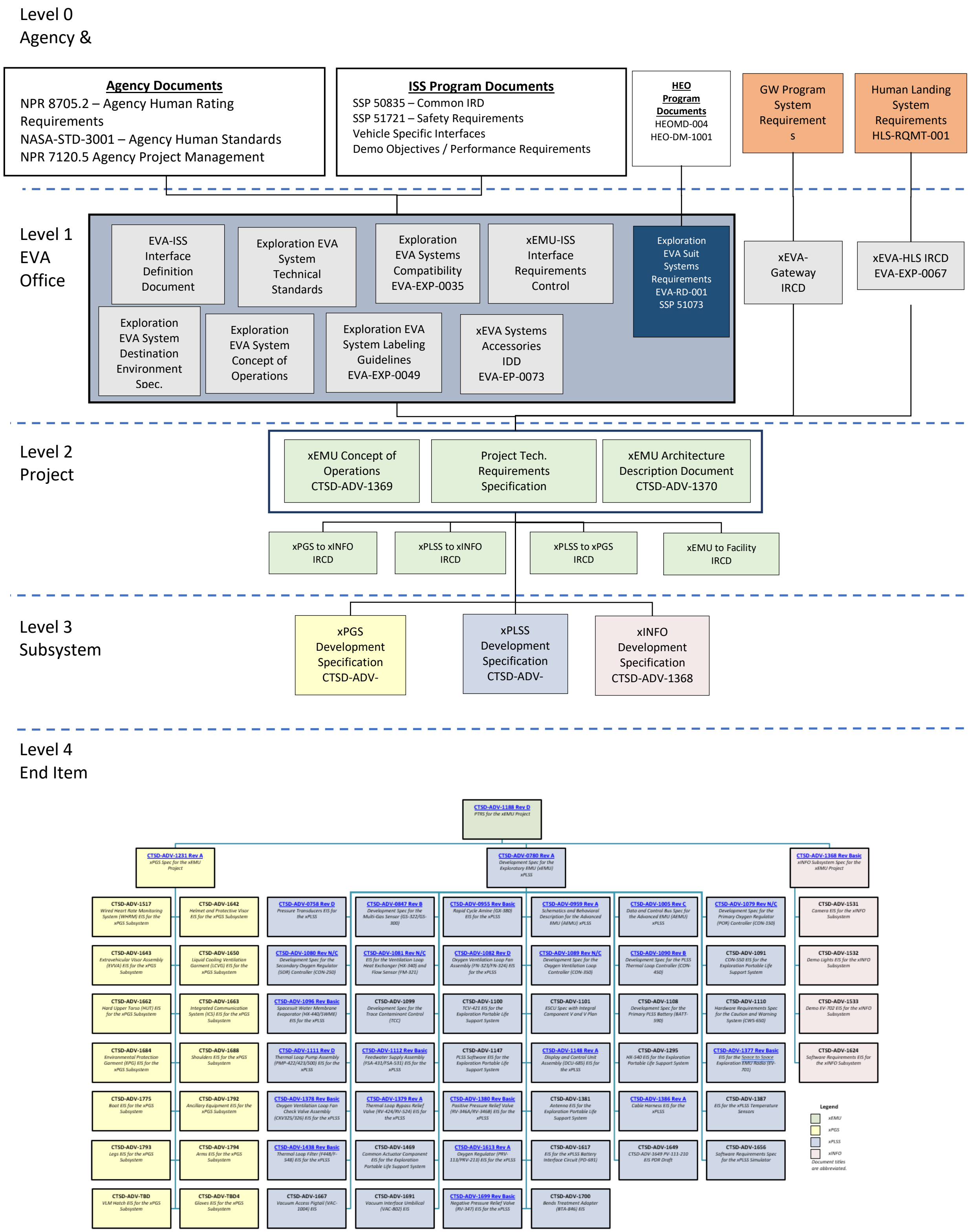


Figure 27 – xEMU Specification Tree: Level 0 through Level 4

6.1.2. Systems Requirements Review Context Shift

As a result of the 2019 “Boots on the Moon” initiative, the original schedule was accelerated nearly two years and the system context from building an ISS demo unit to a lunar suit had changed. This motivation from NASA selecting xEMU as the EVA suit for the Artemis missions (Martin, 2021). This new scope initiated a Delta-SRR (Systems Requirement Review) to account for the change in system context. An executive review was held in late 2019 to summarize the changes to the SSP Level 0 Requirements and how the PTRS containing the 219 Level 1 requirements at the EVA office echelon would be influenced and how the Con-Ops and ADD would be modified to accommodate the change in system context, which reside at the Level 2 system requirement echelon, the primary focus of content during the Delta-SRR. The documents that drive the Level 3 requirements are PTRS, Con-Ops and ADD.

The Delta-SRR plan covered the critical facets of system context shift and was held in December 2019.

- The goal of the Delta-SRR is to establish a complete baseline set of requirements for the initial lunar mission, with the PTRS Rev C from June 2019 to be updated to Rev D to incorporate the lunar 2024 mission requirements.
- The three primary offices would need to verify compatibility issues and multiple configurations per suit, with these offices being EVA system, HLS system and Gateway.
- Interfaces between different groups and/or subsystems would need in addition to their

requirements and specifications would need Internal Requirement Control Documents (IRCDs)

- Identification that different processes from different programs may necessitate multiple products to satisfy the same function across each program (i.e., needing three separate safety review panels for each configuration)
- A compatibility matrix of potential changes as part of the review process for communication and clarity across the three programs
- Only documents to be reviewed would be the PTRS, Con-Ops and ADD
    - IRCDs, SEMP, CM would not be affected until after Level 1 and Level 2 documentation is implemented.
- Delta-SRR entrance criteria to include:
    - Release of all applicable technical documents prior to review
    - Definition of the architecture prior to review
    - Preliminary plans at minimum at a preliminary level state
- System context will be defined in the following categories with compatibility matrices assigned to each for differentiation:
    - xEMU ISS Demo
    - xEMU Microgravity
    - xEMU Lunar Surface
    - xEMU Sustained Lunar Surface
- The only feedback to be received would be technical as it pertains to changes from the xEMU ISS Demo unit to xEMU Lunar units.
    - No detailed design reviews, architecture only

- Requirements may be changed real-time if possible and constitute a proper CM as all pertinent stakeholders are present at the Delta-SRR

The Delta-SRR highlighted the 205 requirements from Rev C of the PTRS. Rev D would then include 212 (as of this date, 219 exist) with 7 requirements deleted, 14 requirements added and approximately 50 requirements changed or augmented, chief among the changes would be the transition from an ISS suit to a lunar suit, where dust (i.e., regolith) mitigation would be a major challenge. At this high level of change, many requirements at the lower levels would be subject to change.

6.1.3. Requirements Engineering Documents

The three following central documents drive ability for the project to provide the verifiable and validated product in the xEMU: (1) the Con-Ops, (2) the ADD and (3) the PTRS. The Con-Ops are influenced by the Level 1 EVA Office Con-Ops and institute it as its parent document. The Con-Ops document specific to xEMU helps define the functions and interactions by operational scenarios in terms of successful EVA criteria. The suit interactions and functional capabilities into divided into three distinct missions: (1) ISS demonstration in low-earth orbit, (2) cislunar orbit and (3) lunar surface operations. The Con-Ops use a tabular method in conjunction with natural language to describe an activity (i.e., physical tasks) and an operation (defined as either ISS, Cislunar or Lunar Surface). The Con-Ops as a result support the detailing of hardware in the ADD, which provides an extensive description of the purpose and functionality of all xEMU hardware. Conversely, the information from the Con-Ops pushes architectural elements for hardware. The ADD details the architecture of the spacesuit and its various configurations and unites the xEMU requirements specified in the PTRS. The Level 2 requirement for all intents and purposes is the primary specification, the PTRS, acts as the systems requirements upper echelon for the subsystem and EIS specifications. The PTRS establishes the design and performance for all the xEMU configurations, supporting 219 system level requirements. By using both the Con-Ops and ADD in tandem, the combination pushes and substantiates requirements in the PTRS, which effectively builds the entire scope of work packages for the xEMU project.

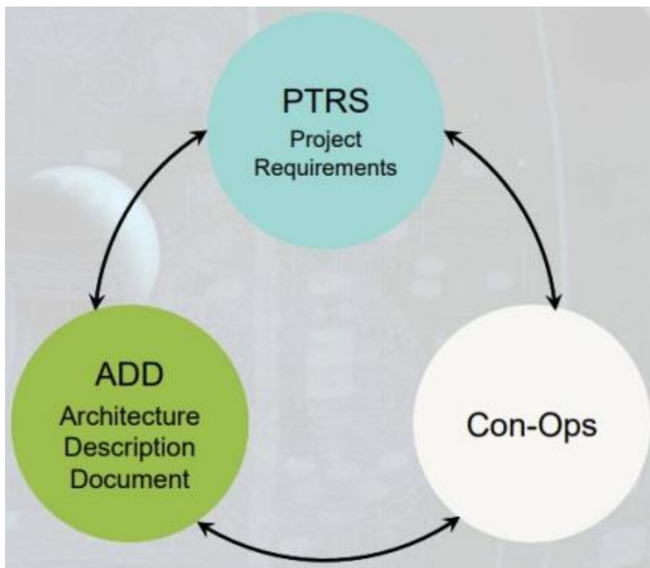


Figure 28 – Requirements Engineering Circle of xEMU

The SEMP's purpose is to describe the technical approach for organizing product, team and process development with the intention of accomplishing the maturation of the project requirements with the focus of design definition and sustainment. The goal of this purpose is to deliver the product within cost, schedule, scope within all necessary constraints. The SEMP governs the integration and implementation of the various levels of the project specification tree that are managed over the product lifecycle.

6.1.4. System of Systems Integration

During the requirements engineering development for the xEMU project, one of the greater challenges was that basic architecture had already existed through prior development activities that dated back approximately a decade and had spanned multiple NASA organizations driven by different presential administrations. A specific example of how this made for an unorthodox system engineering approach is that the xPLSS subsystem already had its decomposition all inclusively in its Level 3 document, CTSD-ADV-780. This presented a unique challenge as the subsystem existed prior to the xEMU needs and requirements build and subsequent decomposition. The goal would then be to reverse decompose the requirements that were in the CTSD-ADV-780 document to maintain their traceability and congruence with higher level requirements and to also allocated the Level 4 type requirements present in the Level 3 document so as to follow suit with the project's approach to requirements decomposition and documentation. A decision was made at the onset of the project that xPLSS, unlike its xINFO and xPGS counterparts:

- Would not use the modeling tool, Cradle®, to organize, control and update requirements documents. This is to state that xINFO and xPGS would be following the MBSE approach while xPLSS would be following a more a DBSE approach to requirement management.
- The decomposition of requirements starting at Level 1 through Level 4 would not be executed but instead the already existing requirements in the Level 3 Subsystem echelon would remain fully decomposed and allocate requirements to the Level 4 EIS echelon. Requirements would still need to be traceable to the higher echelon levels with the ability to be verified and validated, nonetheless.

6.1.5. Cradle® Intended Usage

With regards to a software tool in respect to the three pillars of SysML, the xEMU project requirements engineering database is organized using Cradle®, a MBSE program used for managing requirements and creating a systems engineering design database. The benefits of using Cradle® are the ability to manage data in all forms regardless of size, the static or dynamic nature to work efficiently and effectively across multiple areas of the requirements engineering arena. This allows for traceability to source, tracking every actionable change, the ability to include configuration management, and guarantee the completeness, correctness and quality pedigree needed for the project in terms of documentation and historical record keeping. Cradle® is intended to support the full systems engineering lifecycle of the xEMU project and provides concurrent access by multiple users. It provides unlimited scalability, flexible interfaces, metrics to measure system health, and is customizable without using macros, programing or scripts. Cradle® Requirements Management Software | 3SL Cradle® Software North America (us-3sl.com). The project uses Cradle® Items and Item Types. Item Types include STD COMPLIANCE, PROJ REQ, PROJ SPEC, AND REQ VERIFICATION which stand for standard compliance, project requirement, project specification and requirement verification, respectively. The Item Type supports the definition of common attributes, such as identity, name, kay, group, owner, modified date, etc. and categories such as short fields, single pick lists and multi-pick lists in addition to frames such as long text fields, photos and attachments. Creating new items are done by selecting “New Item” in the Cradle® ribbon with the pertinent information listed below.

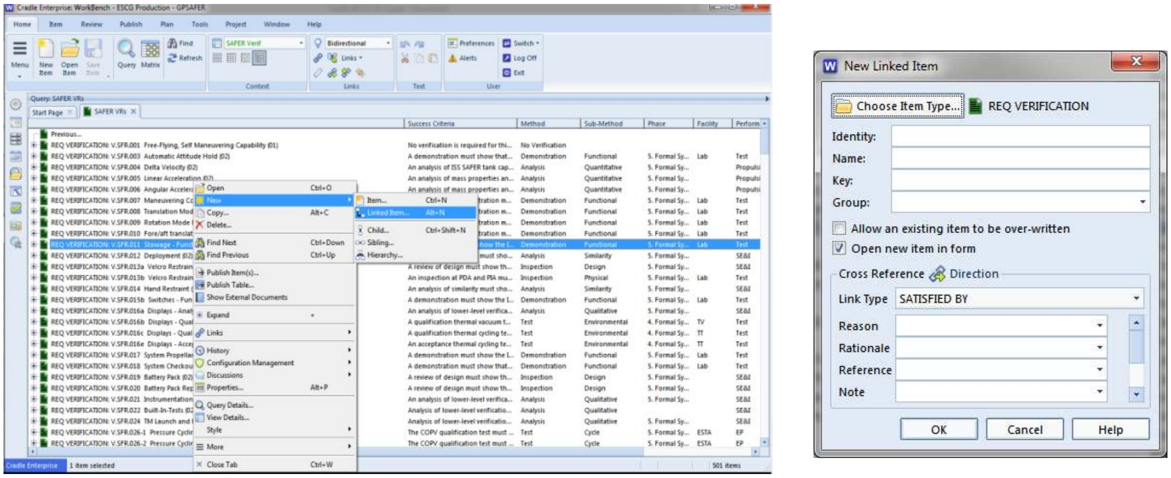


Figure 29 – Creation of New Items and Cradle Database Overview

As a result of inputting requirements directing into Cradle® methodically, specification documents for Level 0 through Level 5 are generated from requirements archive. A Word Document file is published automatically to include text sections, references to applicable documents, explicit parent and child relationships, compliance tables and a listing of TBX (TBD – To Be Determined; TBR – To Be Resolved) tables. Baselining a requirement comes first if there is no existing requirement. When editing a baseline requirement, it is protocol to maintain links and history to preserve traceability as overwriting existing items will overwrite any working drafts, thus effectively resetting the baseline. Each item is owned by the Cradle® user that initiated the creation. Queries may also be solicited for extra data.

Documents are the end product of Cradle®, using the Cradle® Document Publisher. This tool is paramount in the MBSE philosophy as requirements are written in Cradle® in a methodical manner, with attributes such as identify nomenclature, version, draft, name, text, rationale, object, modification date, items status and any linked requirements.

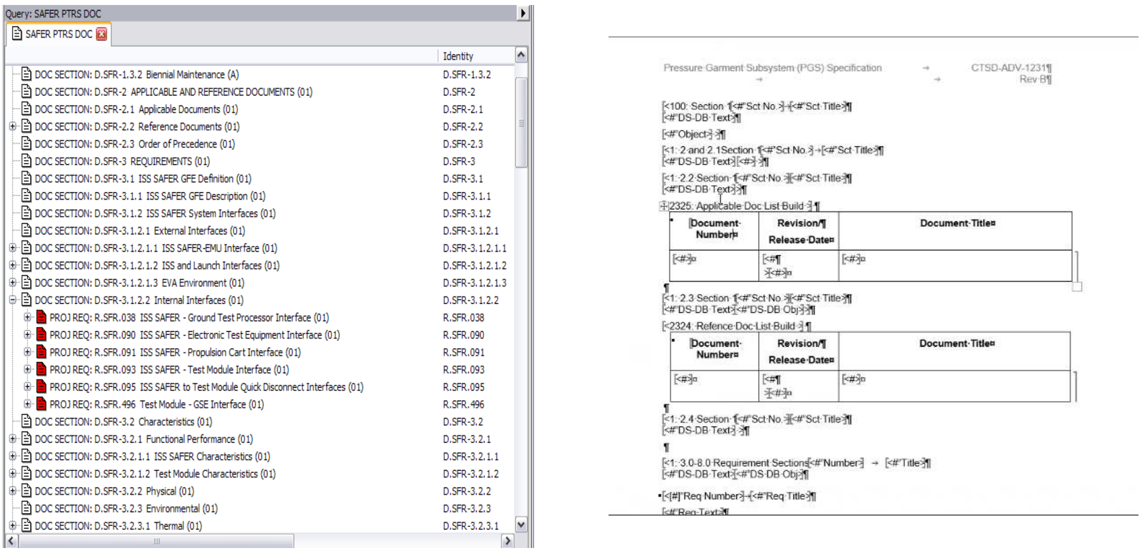


Figure 30 – MBSE Word Document Population via Project Requirement Input in Cradle

6.1.6. Cradle® Actual Usage

While the architecture was **developed** per the intent of the project, there have been instances where execution has not followed the anticipated approach set forward by the SE&I Architecture Team. The xPGS and xINFO groups followed the standard protocol of MBSE whereas xPLSS followed the approach of changing the word document, which was previously agreed upon at the project onset. The challenge comes in updating manually any parent or child as a result of changing one requirement or verification activity. Across xPLSS, there exist 832 subsystem requirements at Level 3 and 40 EIS documents at Level 4 containing several thousand requirements. As there was no SE&I mandate, systems engineers were organized to update the Word document then populate

Cradle®. The xPLSS subsystem also had missing IRCDs and instead used a singular document that would explain the interfaces between xPLSS and the various systems. Cradle® has a built in Configuration Management (CM) system. The CM department however did not use Cradle® to do configuration management and performed it externally to the Cradle® environment.

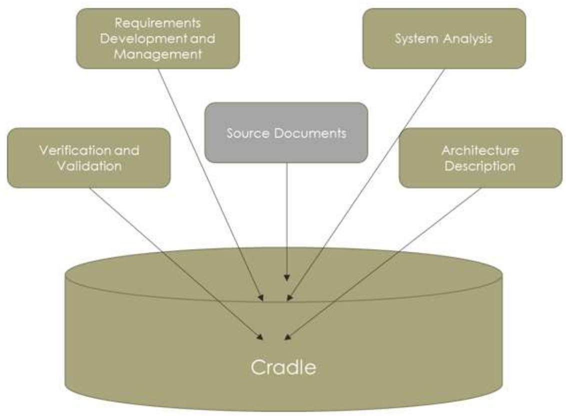


Figure 31 – Cradle® Push Communication Method

6.2. Case Studies in Requirements Building

Two specific case studies are evaluated and characterized against INCOSE standards. These two studies were selected as they represent two indicative cases of challenges on the project as it pertains to requirements engineering but also contained the most quantifiable data for investigation and evaluation of potential alternative method of approach. These two case studies evaluate the importance of robust decomposition of requirements, traceability with clear verifications in place, cross-functional understanding of customer needs across different programs and subsystems, proper adherence to industry (i.e., INCOSE) standards and the usage of behavioral diagramming (i.e., use cases, sequence diagrams, etc.) to help support and ensure the designed, built and verified system meets the intended purpose in its operation environment. These case studies do not and are not meant to necessarily represent the entire suit of requirements engineering decompositions across the project.

6.2.1. Display & Control Unit Auxiliary Lighting

The primary lighting source of the xEMU system is a part of the xINFO subsystem known as the Informatics Lights. These were based on legacy designs from the EMU with added application from evolved industry standard lighting. The overall architecture consists of functions to satisfy worksite lighting, used to provide adequate illumination to the crew member's 2-hand work envelope; translation lighting, used to provide illumination to a larger volume outside the 2-hand work envelope for crew member situational awareness; multiple light sources to provide redundancy and adequate coverage area; 3-point toggle switching for crew member control of the Informatics Lights. The requirements decomposition at the project onset from Level 2 system requirements to Level 3 subsystem requirements as follows:

Table 4 -Worksite and Microgravity Illumination System Requirement

Level 2: System Level Requirements of xEMU			
xEMU System Requirement #	**xEMU System Requirement**	**System Requirement Shall Statement**	**System Requirement Rationale**
[xEMU.FUN.051]	WORKSITE ILLUMINATION	The xEMU shall provide EVA worksite illumination.	Rationale: Work lights provide local lighting at the worksite to enable task completion. There is the possibility of having to work in a darkened environment in all DRMs. Having directional lighting attached directly to the suit is useful to allow both hands free for task completion. The same lighting can be used for translation illumination in micro-gravity. It is anticipated that the same lighting will meet both xEMU.FUN.051 and xEMU.FUN.072.
[xEMU.FUN.072]	MICROGRAVITY TRANSLATION ILLUMINATION	The xEMU shall provide illumination for microgravity EVA translation.	Rationale: There is the possibility of having to translate through a darkened environment in all DRMs. Having directional lighting attached directly to the suit is useful to allow for safer translation. It is anticipated that the same lighting will meet both xEMU.FUN.051 and xEMU.FUN.072

Decomposition of the system requirements to subsystem requirements are as follows:

Table 5 - Worksite Illumination Subsystem Decomposition

Level 3: Subsystem Level Requirements of xINFO				
xINFO Subsystem Requirement #	**xINFO Subsystem Requirement**	**Subsystem Requirement Shall Statement**	**Subsystem Requirement Rationale**	**Traceability to System Req.**
R.INFO.2204	LIGHTING CHROMATICITY	The Informatics subsystem shall provide lights with chromaticity between 2700K and 6500K as defined by ANSI C78-377, Specifications for the Chromaticity of Solid-State Lighting Products	Commission on Illumination (CIE) 1931 chromaticity chart. For variable CCT systems, it is important that humans and cameras within that environment see color correctly and interpret the light as white light anywhere along the color range of white light as defined by ANSI C78-377. To ensure the chromaticity of the multi-lamp system is not compromised all light modules will have the same color temperature values to within +/- 50K. Exceptions to this requirement include conditions that do not require color vision, such as window operations and sleep environment, as determined by a task analysis.	[xEMU.FUN.051], [xEMU.FUN.072]
R.INFO.2205	LIGHTING COLOR ACCURACY	The Informatics subsystem shall provide lights with CRI of 90 +/- 10 by IES TM-30 methodology as written	[Condensed} Accurate representation of the colored environment impacts several areas of concern for human performance and behavior, including critical color matching tasks (e.g., matching litmus strips to cue cards). Accurate representation of the colored environment impacts several areas of concern for human performance and behavior, including critical color matching tasks (e.g., matching litmus strips to cue cards) and the representation of skin tone and biological material (e.g., for health diagnostics). Rapid advancements in modern lighting technology such as solid-state lighting require careful consideration of the proper color fidelity metric selection for the evaluation of color rendition properties of a light source.	[xEMU.FUN.051], [xEMU.FUN.072]
R.INFO.2210	HELMET BUBBLE REFRACTIVE ILLUMINATION	The Informatics lights shall emit illumination cones that introduces no more than 50 lux [TBR] into the suited crew members helmet bubble.	The majority of the flux (primary beam pattern) from the Informatics lamps should avoid significantly penetrating the helmet bubble of the suited crewmember. The justification is that significant penetration into the helmet volume, by a light source, could cause that light to bounce against reflective objects inside the helmet, reflecting back onto the helmet bubble, creating a “veiling reflection” that could decrease visibility, especially when working in predominantly dark operational environments.	[xEMU.FUN.051], [xEMU.FUN.072]

R.INFO.2211	LIGHT RADIATION	The Informatics subsystem shall expose suited crewmembers to light intensities that are less than [TBD] nits.	Non-ionizing radiation can cause permanent damage to suited crew members by direct exposure and reflection of the informatics light on other surfaces including other components of the suit per NASA-STD-3001, Volume 2 NASA Space Flight Human-System Standard.	[xEMU.FUN.051], [xEMU.FUN.072]
R.INFO.2212	VISIBLE ONE-HANDED WORK ENVELOPE ILLUMINATION	The Informatics subsystem shall provide an average minimum illumination of 150 lux [TBR] over the area defined by the suited crewmember's visible one-handed work envelope in Figure 5-1.	150 lux is an enveloping requirement for minimum amount of lighting required for working areas where general tasks are regularly performed, per OSHA 1926.56, Illumination; EN 12464, Light and Lighting: and testing in JSC's Lighting Evaluation and Test Facility (LETF). This light level also meets requirements for microgravity translation. Figure 5.4-1 shows the optimal one-handed work envelope for the EMU and will be used as an approximation until ongoing evaluations to define the optimal one handed work envelope for the xEMU is completed (TBD).	[xEMU.FUN.051], [xEMU.FUN.072]
R.INFO.2215	VISIBLE TWO-HANDED WORK ENVELOPE ILLUMINATION	The Informatics subsystem shall provide an average minimum illumination of 350 lux [TBR] over the area defined by the suited crewmember's visible two-handed work envelope.	350 lux is a goal for the amount of lighting required for detailed task work, per the Orion Human-Systems Integration Requirements (Orion Multi-Purpose Crew Program 70024, Rev C) and testing in JSC's Lighting Evaluation and Test Facility (LETF). Figure 5.4-1 shows the optimal two-handed work envelope for the EMU and will be used as an approximation until ongoing evaluations to define the optimal two handed work envelope for the xEMU is completed (TBD).	[xEMU.FUN.051], [xEMU.FUN.072]
R.INFO.5045	INFORMATICS LIGHTS MEAN TIME BETWEEN FAILURES	The Informatics lights shall have an MTBF of at least 29146 hours for EVA operations for a single nominal EVA	Per xEMU.RMLL.019 only inherent hardware failures, which result in termination of an EVA, are considered for this requirement. Failure of Informatics lights could result in insufficient illumination to complete mission objectives if natural or other artificial illumination is unavailable or insufficient.	[xEMU.FUN.051], [xEMU.FUN.072]

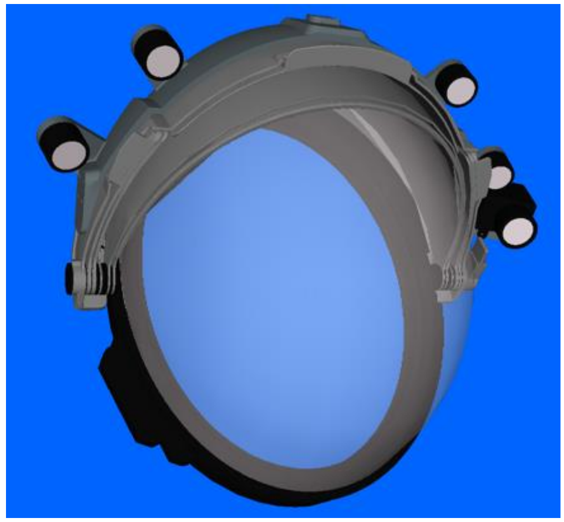


Figure 32 – Rudimentary Primary Illumination for xINFO

As a result of a hazard analysis during a safety review, an issue was raised as to how to address the possible failure of the primary lighting mechanism on xEMU's xINFO subsystem. The need addresses the ability to properly illuminate the crew member's window of visibility during an EVA if primary lighting fails. Requirements were subsequently generated as a result of this safety meeting to satisfy these needs but were allocated to the xPLSS subsystem instead of the xINFO system where the primary lighting needs are addressed. This was done as the DCU had capability to add external, auxiliary lighting (also identified as LT-585), with the assumption that the Auxiliary Thermal Control Loop (ATCL) is active. As a result of the additional functionality added, the Human Health & Performance (HH&P) levied questions against the abilities of the xPLSS subsystem. The first question raised was to define the function and purpose of the auxiliary lights and by extension, what beam distribution and illumination requirements shall be specified and how would those be implemented in an abort or terminate EVA.

Delving into the history of the primary and auxiliary lighting, the documents that define Level 3 requirements are the Project Technical Requirements Specification (PTRS), Concept of Operations (Con-Ops) and Architectural Description Document (ADD). The Con-Ops helps define the functions and interactions for each of the following scenarios: low-earth orbit, cislunar orbit and lunar surface. The operations scenarios are described in the Con-Ops in terms of successful EVA criteria. The Con-Ops as a result support the detailing of hardware in the ADD, which provides an extensive description of the purpose and functionality of all xEMU hardware. By using both the Con-Ops and ADD in tandem, the combination pushes and substantiate requirements in the PTRS. The Con-Ops use a tabular method in conjunction with natural language to describe an activity (i.e., physical tasks) and an operation (defined as either ISS, Cislunar or Lunar Surface). The collection of every activity for every operation is defined as the Operational Scenarios. There are 13 Operational Scenarios, each of which have an activity associated with the three possible operations of ISS, Cislunar or Lunar Surface. These 13 scenarios define operations that could occur in the following phases: pre-flight testing and training, earth launch and logistics, suit assembly, descent and landing, EVA preparations, pre-EVA, EVA, post-EVA, maintenance, ascent and docking, post docking, post-flight, contingency. A sample of one of the operational scenarios is listed below (paraphrased and not verbatim, with export control data protected and not reflected in the sample).

Table 6 – Sample Operational Scenario

Activity	ISS Operation	Cislunar Operation	Lunar Surface Operation
Egress & Suit Don	After suit is powered by the battery supply completely, the crew member disconnects suit from electrically harnessing and water/air umbilical. The degasser/heat-exchanger will	Similar to ISS Operation.	TBD For Lunar Surface Operation.

	activate and begin transmitting thermal cooling via the xPLSS. The RCA will vent built up NH4 to vacuum passively. Oxygen will be provided to the crew member.		

For the auxiliary lighting investigation, there was not a Con-Ops Operational Scenario or activity specified, although primary lighting and the feature of auxiliary lighting are made mention in several instances during the Operational Scenario development. The ADD does make mention of auxiliary lighting as a function of the design description of the DCU but without having context to its emergency usage. As these Con-Ops are used in conjunction with ADD to help generate the PTRS which then helps to decompose those system requirements to the subsystem requirements (or in the case of xEMU, the EIS), the requirements for auxiliary power were not written prior to the aforementioned examination of a hazard analysis during a safety review. As a result, requirements were written to satisfy the hazards of loss of primary lighting with the addition of a virtual working group to write the requirements and discuss scenarios involving the ATCL which would activate the proposed auxiliary lights.

In the ADD, auxiliary lights are defined as illuminating as part of the ATCL and is controlled by the ATCL controller, a device designed around a Field Programmable Gate Array (FPGA) used to control the ATCL and power supply for the auxiliary lights and functions via a DC/DC converter. The ADD further defines the auxiliary lights as a series of LEDs in parallel with resistors to control power. As a result of the configuration, the design supports the usage of redundant power from a battery source (BATT-590) for the auxiliary light system while still allowing functionality over the DCU.

Areas of potential concern as to the functionality of the auxiliary lights were discovered during the virtual working group following the hazard analysis examination during the safety review. Areas were either documented or are addressed in both the ADD and Con-Ops documents but not substantially controlled. The first is the Exploratory Mini Workstation (xMWS) which allows for crew members to access a utility belt with a swing arm and gimbal while using tools during an EVA. Although addressed, one of the areas unexplored was the compatibility with auxiliary lights, whose illumination could be impeded with the xMWS blocking the light source, effectively creating a possible conflict of systems within systems. Second, would both emergency translation operations which would need to specify the color vision, contrast, suit and distance illumination and two-handed work envelope. Third would be the Exploratory Service and Cooling Umbilical (ESCU), which is used to recharge the electrical, water and oxygen systems when attached to the DCU during an egress/regress to or from habitat. In similar fashion to the xMWS, the ESCU would certainly obstruct illumination emitted from the auxiliary lighting source. Much like the auxiliary lights system itself, the xMWS, emergency translation and ESCU scenarios are not listed explicitly with any relation to how they would function with auxiliary lighting.

The next step was to detail the scenarios and tasks with natural language to help build requirements. Three scenarios were defined. First, was the situation where primary lights fail, which induces a loss of visibility and illumination, captured as a result of the hazard analysis. This would require auxiliary lights to be used intermittently for translation illumination or allow crew to wait on standby until day passes for proper illumination.

This first scenario also established that this would not create an end result of EVA termination or abortion. Second, is where the primary thermal control system fails, characterized by inadequate thermal control, one of the three primary life support systems. In this scenario, the primary lights are still illuminated and while the auxiliary lights are additionally illuminated, they are not required to perform any translation. Third, would be the scenario where the total primary system fails. The cause would be full loss of power and as a result, supplies no power to sustain the CWS. Primary lights would be unavailable, and the crew would survive with an open loop purge which is defined as a method of receiving breathing from the oxygen tanks in the oxygen loop but forgoing the functionality of the ventilation loop which scrubs carbon dioxide generated from the crew member. During this time, the ATCL is active and only illumination will be from the auxiliary lights and vehicle lights. The context supporting the scenario would include an environmental situation where the crew finds themselves in the beginning of night pass and a location inside the truss or near a solar array.

Adjacent scenarios presented in natural language also complement the usage of the xMWS when an event triggers the usage of auxiliary lights. Two options exist: jettison the xMWS which will require a design change for an alternate location for the tether attachment or extend the design of the xMWS to not block the auxiliary lights which would retain the tether attachment. Scenarios were not developed in natural language for the emergency translation operations when an auxiliary light operation is initiated but assumptions were built around color vision, close to suit and distance illumination and the need to update the two-handed work envelop was initiated.

The design of the DCU was modified as a result of the findings to include dual lamp modules for redundancy located on the DCU front face which utilize dual LED sources. Once these design changes were quantified, the DCU needed to accommodate this new ability, but the unit has a highly constrained volume, and the unit was redesigned against these contingent requirements. The microgravity translation illumination requirement was rewritten to account for the ability or lack thereof as it pertained to adequate and sufficient auxiliary lighting for lunar translation:

Table 7 – Worksite Illumination System Requirement

Level 2: System Level Requirements of xEMU			
xEMU System Requirement #	**xEMU System Requirement**	**System Requirement Shall Statement**	**System Requirement Rationale**
[xEMU.FUN.051]	WORKSITE ILLUMINATION	The xEMU shall provide EVA worksite illumination.	Rationale: There is the possibility of having to translate through a darkened environment in all DRMs. Having directional lighting attached directly to the suit is useful to allow for safer translation. It is anticipated that the same lighting will meet both xEMU.FUN.051 and xEMU.FUN.072. No requirement for lunar surface translation illumination is levied on the xEMU. If the lighting used for xEMU.FUN.051 and xEMU.FUN.072 are found to be inadequate for lunar surface translation, another xEVA System team will develop supplemental lighting, and xEMU documentation will be updated to ensure that such supplemental lighting can be accommodated.

Here, we find that the system requirement rationale assumes that functionality may be adequate for lunar surface translation but extends the notion for an update by the xEVA system team to develop supplemental. As the requirement is decomposed from the parent, there will now be updates or additions to the requirements to xPLSS and xINFO. The child from the XEMU.FUN.72 parent read as follows:

Table 8 - Auxiliary Illumination Subsystem Requirement

Level 3: Subsystem Level Requirements of xPLSS				
xPLSS Subsystem Requirement #	**xPLSS Subsystem Requirement**	**Subsystem Requirement Shall Statement**	**Subsystem Requirement Rationale**	**Traceability to System Req.**
R.PLSS.600.153	Auxiliary Lighting	The DCU-685 shall provide 350 lumens of white light emitted from at least 4 source locations separated across the anterior surface of the DCU.	Rationale: This enables EVA abort illumination in the event that all primary task lighting has failed. The selection of four discrete source locations is intended to minimize the chance that a tool or other configuration item could block all of the light sourced from the DCU.	[xEMU.FUN.072]

This requirement was updated to remove the specific lumen requirements (and subsequently remove the lumens completely) and the 4 source locations. The rationales were also updated to includes assumptions as to the situational path avoidance for EVA terminate in terms of contracts, obstacle avoidance, contrast and no color fidelity. The rationale was also updated to allow for reconfiguration lighting to be optimized for micro-gravity or lunar translations with minimal light change to the system.

Table 9 - Auxiliary Lighting Update

xPLSS Subsystem Requirement #	**xPLSS Subsystem Requirement**	**Subsystem Requirement Shall Statement**	**Subsystem Requirement Rationale**	**Traceability to System Req.**

R.PLSS.600.153 (Previous)	Auxiliary Lighting	The DCU-685 shall provide 350 lumens of white light emitted from at least 4 source locations separated across the anterior surface of the DCU.	Rationale: This enables EVA abort illumination in the event that all primary task lighting has failed. The selection of four discrete source locations is intended to minimize the chance that a tool or other configuration item could block all of the light sourced from the DCU.	[xEMU.FUN.072]
R.PLSS.600.153 (Updated)	Auxiliary Lighting	The DCU-685 shall provide auxiliary illumination emitted from the anterior surface of the DCU. Micro-gravity translation requires illumination of two-hand work envelope as shown in Figure TBD.	Rationale: This enables EVA abort illumination when all primary task lighting has failed. Assumption that situational path avoidance for EVA terminate requires contrast, obstacle avoidance, and no color fidelity. The intention is for light source(s) that minimize the chance that a tool or other configuration item could block all the light sourced from the DCU. Intend to allow configuration of lighting optimized for either micro-gravity or lunar surface translation with minimal change to light system (i.e., change lenses).	[xEMU.FUN.072]

The requirement that was decomposed originally without having context of a failure to the primary lighting source was found to have a close resemblance to the prior xPLSS requirement:

Table 10 – Auxiliary Lighting for DCU EIS Requirement

Level 4: End Item Specification Level Requirements of DCU				
DCU EIS Component Requirement #	**DCU EIS Component Requirement**	**Component Requirement Shall Statement**	**Component Requirement Rationale**	**Traceability to Subsystem Req.**
R.DCU-685.082	Auxiliary Light Illumination - Micro-Gravity	The DCU emergency lighting shall provide 350 lumens of white light emitted from at least 4 source locations separated across the anterior surface of the DCU.	Rationale: This enables abort lighting in the event that the primary lighting has failed.	[R.PLSS.600.153]

Much like the xPLSS parent requirement, the requirements were then decomposed as a function of viewing higher level requirements (system) and anticipating lower-level requirements (subsystem and unit) but were nearly identical in nature and thus, not an actual decomposition. The lumens requirement is now properly allocated to the Level 4 area as a function of parenting to the Level 3 requirement addressing the functionality of illumination. However, the lumens, which are a volume of light emitted from the light source, were determined to be the inappropriate measure in a requirement and were changed to lux, which is the measure of brightness or intensity of the light at a prescribed distance relative to the source. Lumens describes total energy and as a result is the incorrect unit. In the same direction, the 4 source locations were further defined at this level of decomposition to be dependent upon the two-handed work envelope, which in the moment of developing the requirements was still in the process of being updated. The number of lamps here was determined to be over prescriptive, with the goal of providing light that is not obstructed. The location specification should have its own designated requirement as the current requirement is no longer atomic in agreement with INCOSE requirements writing standards. Due to the uncertain nature and potential limitations in DCU design, the maximum surface area for lighting system with an envelope for maximum lamp definition would be defined. The emergency lighting as a whole is defined as a standalone system with the implementation specifics left malleable for design engineers to define. Remove from the requirement was the color, as the DCU already had chromaticity defined in its requirements previously. In terms of strict definition, the color "white" has no connotative color association without guidance from the aforementioned chromaticity requirement. It is important to note that color fidelity is utilized to define the manner in which lights affect the perspective of the viewer as it pertains to colored materials. The before and after requirement update is as such:

Table 11 – Update to Illumination DCU EIS Requirement

Level 4: End Item Specification Level Requirements of DCU				
DCU EIS Component Requirement #	**DCU EIS Component Requirement**	**Component Requirement Shall Statement**	**Component Requirement Rationale**	**Traceability to Subsystem Req.**

R.DCU-685.082 (Previous)	Auxiliary Light Illumination - Micro-Gravity	The DCU emergency lighting shall provide 350 lumens of white light emitted from at least 4 source locations separated across the anterior surface of the DCU.	Rationale: This enables abort lighting in the event that the primary lighting has failed.	[R.PLSS.600.153]
R.DCU-685.082 (Updated)	Auxiliary Light Illumination - Micro-Gravity	The DCU auxiliary lighting shall provide average minimum illumination of 150 lux [TBD] at the angles and distances defined by the suited crewmember's visible two-handed work envelope for microgravity translation. Crew two-handed work envelope is shown in Figure TBD.	Rationale: This enables abort lighting when the primary lighting has failed. 150 lux is an enveloping requirement for minimum amount of lighting required for working areas where general tasks are regularly performed, per OSHA 1926.56, Illumination; EN 12464, Light and Lighting: and testing in JSC's Lighting Evaluation and Test Facility (LETF). This light level meets requirements for microgravity translation.	[R.PLSS.600.153]

Resultant requirements were decomposed and allocated from the xPLSS Level 3 subsystem to the Level 4 DCU EIS. These were derived for the location of the auxiliary lights and were originally written as such:

Table 12 – Original Auxiliary Light DCU EIS Requirement

Level 4: End Item Specification Level Requirements of DCU				
DCU EIS Component Requirement #	**DCU EIS Component Requirement**	**Component Requirement Shall Statement**	**Component Requirement Rationale**	**Traceability to Subsystem Req.**
R.DCU-685.XXX [TBD]	Auxiliary Light Source Location	The DCU emergency lighting shall provide 350 lumens of white light emitted from at least 4 source locations separated across the anterior surface of the DCU.	Rationale: This enables abort lighting in the event that the primary lighting has failed. The selection of four or more discrete source locations is intended to minimize the chance that a tool or other items could block all the light.	[R.PLSS.600.153]

The lumens/lux was removed entirely as it was addressed in the traceable parent requirement. The specification of location identified the usage of LEDs without any direction as to where the location would reside. This decision was made as aforementioned to allow for ease of design and flexibility for the engineering design team. The previous and updated requirement as juxtaposed as such:

Table 13 - Updated Auxiliary Light DCU EIS Requirement

Level 4: End Item Specification Level Requirements of DCU				
DCU EIS Component Requirement #	**DCU EIS Component Requirement**	**Component Requirement Shall Statement**	**Component Requirement Rationale**	**Traceability to Subsystem Req.**
R.DCU-685.XXX [TBD] (Previous)	Auxiliary Light Source Location	The DCU emergency lighting shall provide 350 lumens of white light emitted from at least 4 source locations separated across the anterior surface of the DCU.	Rationale: This enables abort lighting in the event that the primary lighting has failed. The selection of four or more discrete source locations is intended to minimize the chance that a tool or other items could block all the light.	[R.PLSS.600.153]
R.DCU-685.XXX [TBD] (Updated)	Auxiliary Light Source Location	The DCU auxiliary lighting shall source locations using any number of Light Emitting Diodes (LEDs) and optical modifiers as required to provide the required control functionality and beam distribution intensity while restricted to available areas shown in Figure X.	Rationale: This enables abort lighting when the primary lighting has failed. The selection of source locations is intended to minimize the chance that a tool or other items could block all the light. Intend to allow configuration of lighting optimized for either micro-gravity or lunar surface translation with minimal change to light system (i.e., change lenses).	[R.PLSS.600.153]

To support the above requirements, the following new requirements were established to include a TBD figure to leave flexibility in the design. After the lux versus lumens situation was resolved, the beam distribution and intensity were able to be defined adequately for microgravity operations, lunar gravity operations and the physical footprint for the lamp.

Table 14 - New DCU EIS Requirements

Level 4: End Item Specification Level Requirements of DCU				
DCU EIS Component Requirement #	**DCU EIS Component Requirement**	**Component Requirement Shall Statement**	**Component Requirement Rationale**	**Traceability to Subsystem Req.**

R.DCU-685.0xx [TBD]	Beam Distribution & Intensity – Microgravity Operations	For microgravity operations, the DCU Emergency Lighting lamp system shall provide, at minimum, an illuminance (lux) at the angles and distances identified in TBD figure, with respect to the identified reference coordinate system.	Rationale: This enables EVA abort illumination when all primary task lighting has failed. Assumption that situational path avoidance for EVA terminate requires contrast, obstacle avoidance, and no color fidelity. The intention is for light source(s) that minimize the chance that a tool or other configuration item could block all the light sourced from the DCU. Intend to allow configuration of lighting optimized for either micro-gravity or lunar surface translation with minimal change to light system (i.e., change lenses).	[R.PLSS.600.153]
R.DCU-685.0xx [TBD]	Beam Distribution & Intensity – Lunar Gravity Operations	For lunar gravity operations, the DCU Emergency Lighting lamp system shall provide, at minimum, an illuminance (lux) at the angles and distances identified in TBD figure, with respect to the identified reference coordinate system.	Rationale: This enables EVA abort illumination when all primary task lighting has failed. Assumption that situational path avoidance for EVA terminate requires contrast, obstacle avoidance, and no color fidelity. The intention is for light source(s) that minimize the chance that a tool or other configuration item could block all the light sourced from the DCU. Intend to allow configuration of lighting optimized for either micro-gravity or lunar surface translation with minimal change to light system (i.e., change lenses).	[R.PLSS.600.153]
R.DCU-685.0xx [TBD]	Lamp System Physical Footprint	The DCU Emergency Lighting lamp system shall utilize any number of Light Emitting Diodes (LEDs) and optical modifiers as required to provide the required control functionality and beam distribution intensity while not exceeding maximum physical envelope dimensions as indicated in TBD Figure.	Rationale: This enables EVA abort illumination when all primary task lighting has failed. Assumption that situational path avoidance for EVA terminate requires contrast, obstacle avoidance, and no color fidelity. The intention is for light source(s) that minimize the chance that a tool or other configuration item could block all the light sourced from the DCU. Intend to allow configuration of lighting optimized for either micro-gravity or lunar surface translation with minimal change to light system (i.e., change lenses).	[R.PLSS.600.153]

As a result of the auxiliary lighting, emergency lighting glare requirements were added as follows:

Table 15 - New DCU EIS Glare Requirement

Level 4: End Item Specification Level Requirements of DCU				
DCU EIS Component Requirement #	**DCU EIS Component Requirement**	**Component Requirement Shall Statement**	**Component Requirement Rationale**	**Traceability to Subsystem Req.**
R.DCU-685.133	Emergency Light Glare	The DCU-685 Emergency Abort Lighting shall not create glare for the suited crew.	Rationale: This is intended to prevent excessive light reflections or direct light transmission from impairing the crew visibility during an EVA termination using the LT-585. This requirement complies with R.PLSS.600.167 Emergency Abort Lighting Glare (LT-585/DCU-685) in CTSD-ADV-780.	[R.PLSS.600.153]

One of the challenges presented is the nature of how the light will function, regardless of the anticipation. By virtue, the auxiliary lighting will create glare as they are light sources that emit a light with the potential to create a glare, depending on the interactions the suit and/or crew member will have with the environment. For example, if the crew member is grounded (i.e., not floating in microgravity), it would be feasible to define maximum luminance as a function of angular offset requirement. This would thus drive a hard requirement for project to provide a baffle above the lights to dampen the effect of glare. Due to the system context of microgravity operations, the crew could be in positions that would compromise anti-glare and put a light source within direct view of an opposing crew member while on EVA. The system context is only able to control what is fixated to the crew member and not their behavior while on EVA. The new requirement R.DCU-685.133 will need to have a correction to update the Level 3 requirement subsystem at the xINFO level and propose an inquiry with the radiation team to clarify if a glare or bright light solution accommodates a negative risk for a non-stationary light source such as microgravity operations during an EVA. For instance, a glare solution could define the maximum luminance, however a more potent definition could be to clearly state the controls that would be required to accommodate the operational limitations of the light source. These controls might include mitigating by design to include minimizing suited crewmembers potential view of their own light source, usage of a diffusion film or diffuser materials with optical guides that obscure the direct view by the crew member of the source of the LED, or usages of baffles or hoods to obscure a light source from an observing crew member from the auxiliary light source produced by the crew member. Chromaticity is addressed in a singular requirement at the Level 4 EIS echelon.

Table 16 - Chromaticity DCU EIS Requirement

Level 4: End Item Specification Level Requirements of DCU				
DCU EIS Component Requirement #	**DCU EIS Component Requirement**	**Component Requirement Shall Statement**	**Component Requirement Rationale**	**Traceability to Subsystem Req.**
R.DCU-685.120	Emergency Light Chromaticity	The DCU emergency lights shall have a chromaticity that falls within the chromaticity gamut for white light for the Correlated Color Temperature (CCT) range of 2700K to 6500K as defined by ANSI C78-377, Specifications for the Chromaticity of Solid-State Lighting Products.	Rationale: This is intended to protect for the ability to perform Gold Salt or other decontamination testing using the illumination from the abort lighting should the illumination levels prove sufficient. This is compliant with NASA-STD-3001, [V2 8059]. This requirement complies with R.PLSS.600.169 Abort Lighting Chromaticity (LT-585/DCU-685) in CTSD-ADV-780.	[R.PLSS.600.169]

As written, this requirement is satisfactory. However, the range of color could be improved to narrow a selection and disambiguate and clarify an allowed color for emergency lighting to improve visibility in lower light conditions. This range could be reduced to “cooler” correlated color temperature (CCT) values while still in compliance with NASA standards (i.e., NASA-STD-3001). A recommendation would be to require the DCU emergency light chromaticity to have a CCT with a “color” color temperate so as to increase the amount of green wavelength content in the light source spectrum with the motivation to optimize the visual contrast, especially during surface lighting conditions that have the potential to be too low to retain an adequate color vision. A recommendation would be that before specifying the same LEDs as would be used on the primary lighting source for the xINFO subsystem that beforehand, that Human-In-The-Loop (HITL) testing be performed with a light system with dimmable light capabilities. This system set could vary between an array of CCT values to determine an appropriate or list or appropriate values that provide maximum contrast at lower light levels with dark and light-colored surface materials. Emergency lighting system are typically optimized to include just enough lighting for safety and operations and as such emphasize visual contrast over color vision.

Color fidelity is addressed as a singular requirement at the Level 4 EIS echelon.

Table 17 - Color Fidelity DCU EIS Requirement

Level 4: End Item Specification Level Requirements of DCU				
DCU EIS Component Requirement #	**DCU EIS Component Requirement**	**Component Requirement Shall Statement**	**Component Requirement Rationale**	**Traceability to Subsystem Req.**
R.DCU-685.121	Emergency Light Color Fidelity	The DCU emergency lights shall have a score of 90 ± 10 on a color fidelity metric that is appropriate for the utilized lighting technology as designated by the Color Fidelity Metric (Rf) defined by IES TM-30 methodology.	This requirement complies with CTSD-ADV-780, Subsystem Specification for the Exploration EMU (xEMU) Portable Life Support Subsystem (PLSS) [R.PLSS.600.153] EMERGENCY ABORT LIGHTING (LT-585/DCU-685)	[R.PLSS.600.153]

While this requirement is correctly measurable, a recommendation for the project is the determination of power distribution desired to produce the illumination goals for emergency lighting for the DCU’s auxiliary lights. An inquiry is recommended for the project to meet with stakeholders and determine if crew members need to make critical color discrimination decisions under the emergency lighting operations. It is important to note the inverse relationship between color fidelity and efficacy (i.e., higher color fidelity yields lower efficacy). Once light levels drop below a particular threshold, the accuracy in color vision is diminished. Color vision impacted twofold: by surface reflectance and illuminance. While it is possible that the lamp could have an adequate color fidelity score, the lamp may also inherently not provide illumination sufficient at a task surface to properly distinguish colors of materials and stakeholder elicitation should be sent an inquiry if color is needed and thus provide a requirement in the specification to address if emergency lighting is required for tasks needing color critical evaluations of materials which should elicit a standoff distance from the suit to be quantified. And as a result, may require certain illuminance (i.e., lux) as a function of a potential distance from the DCU emergency light to enable color vision.

The aforementioned requirements in the Subsystem and EIS levels (Level 3 and Level 4, respectively) have produced the following light requirements that have yet to be reviewed by the SE&I forum and the CM group. These include TBDs in both temporal conditions, power during off-nominal conditions, rationale and traceability.

Table 18 - New Lamp DCU EIS Requirement

Level 4: End Item Specification Level Requirements of DCU				
DCU EIS Component Requirement #	**DCU EIS Component Requirement**	**Component Requirement Shall Statement**	**Component Requirement Rationale**	**Traceability to Subsystem Req.**
R.DCU-685.NEW	Lamp System Performance Tolerance Thermal	The DCU Emergency Lighting System shall provide all required lighting performance criteria within performance tolerance for the full range of thermal operating conditions.	TBD	TBD

R.DCU-685.NEW	Lamp System Flicker	To limit flicker and where direct current control of lighting intensity is not used, the DCU Emergency Lighting System lamp shall limit a pulse width modulation (PWM) driver frequency settings to no less than 5000Hz.	TBD	TBD
R.DCU-685.NEW	Lamp System MTBF	The Emergency Lighting System shall have a mean time between failure of TBD hours.	TBD	TBD
R.DCU-685.NEW	Lamp System Off Nominal Performance	TBD requirement to establish reduced function light output due to off nominal power conditions.	TBD	TBD

One of the challenges associated with requirement building was the insufficient glossary, both in terms of one that was known to have needed establishing but also those terms that were a function of new and needed requirements and additional context for the system that arose during decomposition after the safety review. The three ranges that refer to human vision adaptation are photopic, mesopic and scotopic. Photopic references to cone vision and typically covers adaptation levels where illuminance is greater than 3 cd/m^2 (30 lux with 30% reflectance). Mesopic refers to both the active cones and active rods in human vision and typically convers adaptation where illuminance is between 3 and 0.01 cd/m^2. Scotopic refers to the rod in human vision that typically covers adaptation where illuminance is less than 0.01 $cd/m^2$2 (Burkhard, et al, 1981). Peak sensitivity in these rods is at 507 nanometers, found in the blue-green portion of the visible light spectrum. As light levels drop below 0.001 cd/m^2, only the rods are active and maintain the ability to distinguish between colors as finer details are diminished (Burkhard, et al, 1981).

It is recommended that both in the Level 3 and Level 4 specifications to include the following lighting definitions. This will provide the project with adequate natural language distinction to avoid disambiguation and promote a shared understanding of system context and requirement definition.

Illuminance: Photopic measurement of luminous flux incident over a defined surface area. Units are in lux or lumens/meters^2.

Luminance: Photopic measurement of luminous flux emitted per solid angle from a surface. Units are in candela/meters^2 (cd/m^2).

Luminous Flux or Luminous Power: Total photopic quantity of light emitted from a source from all angles. Units are in lumens.

Luminous Intensity: Derived photopic unit that represents the luminous flux or energy emitted per solid angle. Units are in candela (cd).

Luminous Efficacy: For lighting applications, typically efficacy is used to describe the efficiency of a light source to convert energy (power) to light. Units are in lumens/watt.

Spectral Efficacy: For lighting applications, spectral efficacy has the same units as efficacy or luminous efficacy (lumens/watt) with the distinct difference in that spectral efficacy represents the spectral conversion of a light source's radiometric power in watts to photometric units in lumens.

Solid Angle: This is volumetric angular section from a unit sphere and is analogous to the well-known trigonometric concept of the unit circle. Units are in steradians (sr). An entire sphere equals 4π sr.

Beam Distribution: Sometimes called Beam Distribution, Beam Angle, or Beam Characterization. This represents a 90° hemisphere or 180° spherical characterization of the intensity of light at multiple angles from the source. Typically, illuminance measurements are captured at a fixed radius at multiple angles. Beam distribution is usually reported in relative percent intensity per angle with estimated lumen output, candela per angle, or illuminance per radius per angle.

HWHM: Acronym for shorthand numerical description of a lamp beam pattern. HWHM stands for half-width-half-maximum. If HWHM is given for a lamp, the number represents the half-angle (angle drawn from normal vector from center of the lamp) at which the lamp's beam distribution intensity falls to 50%.

Spectral Irradiance: Radiometric unit, analogous to illuminance, representing the radiant flux per surface area per wavelength. Units are in watts/meter^2/nanometer (W/m^2/nm).

Spectral Radiance: Radiometric unit, analogous to luminance, representing the radiant flux emitted by a given surface area per solid angle per wavelength. Units are in watts/steradian/meter^2/nanometer (W/sr/m^3/nm).

Spectral Power Distribution: Waveform representing energy (absolute or relative) emitted per a range of wavelengths. All light sources have a unique spectral power distribution (SPD) that is impacted by its chemistry. The SPD is an essential dataset for estimating metrics dependent on wavelength.

Chromaticity: This is a calculated metric where the format of the units can be different depending on which standard is used. Chromaticity describes the color of an object, whether that be a surface material or light source. Chromaticity can't be estimated without the usage of a spectrophotometer to measure the spectral power distribution of a light emitting source or reflectance spectrum of a material.

Color Fidelity: This is a calculated metric where the format of the units can be different depending on which standard is used. Color Fidelity describes the accuracy of a light source to render the appearance of colored materials accurately where the definition of perfect is how the Sun's renders the color of materials. Color Fidelity can't be estimated without the usage of a spectral radiance or spectral irradiance meter to measure the spectral power distribution of a light emitting source.

Goniophotometer: Specialized test equipment configuration that includes a rotation stage and is used to collect beam distribution data for a light source. The type of goniophotometer is defined by the location of the rotation stage (lamp verses sensor).

Diffusion/Diffusor: A light diffusion material or diffusor is a material designed to scatter or redirect light that passes through it, or it can also represent a rough surface that light impacts and scatters multiple directions from.

Reflectance/Reflector: The property of a material to reflect and scatter light. Reflectance of surface materials is an important lighting system property as it impacts how humans and cameras observe the environment and the efficiency of lighting systems to illuminate surfaces to sufficient levels to create the desired luminous contrast. Reflectance can be considered part of architecture and can be used as a tool in the form of a reflector.

Uniformity: This is a property that is typically applied for surface illumination but can also be applied to the light emitting face of light sources. Uniformity is usually defined in the form of ratios such as maximum/minimum, and average/minimum with a defined sampling grid size. Uniformity is an important safety and usability metric to minimize human error due to uneven illumination. Uniformity is achieved through a combination of beam distribution design, lamp placement, and understanding of reflective surfaces for the operational area.

Glare: This is a property that describes various problems in human perception of light and the interaction of light with surfaces and materials within an operational environment. Distracting glare is an "annoyance" where, because of reflection and refraction, it creates visual artifacts making it harder to see and resolve an object. Discomforting glare is caused by bright direct and reflected light that makes it hard to look at the object because of the brightness level. Disabling glare causes objects to appear to have lower contrast because of scatter inside the eye. Blinding glare is caused by a direct or indirect light source and is so bright that the observer can't see or is visually compromised.

After establishing new requirements, editing current requirements, initiating design changes, defining terms in natural language in a glossary and properly initiation of the decomposition of future requirements, the EVA office (initiators of the Level 1 requirements which help define the Level 2 Project/System Requirement suite) dictated to project that an auxiliary lighting source for emergency operations was not a needed requirement at the xPLSS or xINFO levels and thus a design change and future requirement suite to satisfy auxiliary lighting was no longer needed. Instead, an auxiliary supplemental light source (i.e., flashlight-type mechanism) would be available to crew upon emergency lighting upon any off-nominal occurrences. The final path forward was to utilize the supplemental lighting source and forgo any complexities associated with creating new requirements, forgo scrubbing current requirements to support a fit verification and verifications matrix and ultimately changing the design which may introduce some emergent behavior across subsystems. Interviews and brainstorming sessions with systems engineers involved in the process determine that the root cause of the challenge traces back to the missing requirement to distinguish between the need for an embedded, native on-suit light source. Upon inspection, there was no top-level requirement above the Level 3 or Level 4 requirements to support the auxiliary lighting on-suit. Although some xINFO, xPLSS and DCU level requirements make mention of lighting and supplemental lighting, none could trace back to a higher-level requirement that would have been driven by a customer need. System engineers on the project draw a comparison between the difference between a feature and a function. A feature is an unintended and at times either useful or non-useful artifact that isn't adequately linked to a requirement and as such is an artifact by virtue of implementation. The existing requirements that discussed auxiliary lighting were then added to and revised as a function of the safety review. These original requirements created a feature and, in this case, malicious as it drove potential design changes, added and changed requirements with the end result of never being needed as the top-level requirements were non-existent for contingency illumination. Regardless of the hazard analysis, having a control for the usage of auxiliary lighting.

6.2.2. Lunar Dust Mitigation

One of the higher profile requirement suites for xEMU was the addition of functionalities across xEMU to account for lunar dust, more specifically the mitigation of lunar dust. One of the customer concerns was the ability to relinquish lunar dust from the suit upon reentry to habitat following an EVA. More discretely, the lunar dust referred to in particular is the accumulation of regolith. The lunar surface is comprised of regolith, a sediment of unconsolidated debris, with a thickness of between 5 to 10 centimeters of a density of approximately 1.5 grams per centimeter cubed. This regolith also contains rock fragments and breccia from the local bedrocks on the lunar surface. This regolith appears as a fine gray soil as approximately half the weight of the lunar soil is between 60 to 80 microns in size (McKay, 1991). The lunar regolith also could include trace percentages of pyroclastic material, a type of volcanic glass that was uncovered during the Apollo 15, 16 and 17 missions (Delano, et al., 1981). The lunar dust interface requirements are established at the Level 0 Program requirement echelon and instituted at the Level 1 EVA Office requirement echelon. As this requirement is at the highest hierarchy, each of the three primary stakeholder groups, EVA system, HLS system and Gateway system are involved, with each stakeholder eliciting similar yet differing needs. For example, lunar dust is a need more critical to the HLS and Gateway systems as they both operate in the lunar context whereas the EVA system has historically supported EVAs during ISS operations wherein lunar dust is not in the system context. Although in the strictest definition of the term, EVA is an extravehicular activity outside of a habitat. The current system context of EMUs accounts for EVAs only performed in the ISS context.

These needs were unknown, even during the Apollo program, with no precedent in place to anticipate the need for regolith liberation. These presented a series of challenges on crew operations and hardware performance since regolith has a propensity to be easily adherable to suit surfaces and is abrasive in nature. As dust adhered to the suit surface, it proved to be problematic during Apollo missions and crew cabin and lunar rover battery radiators needed to be cleaned periodically and unplanned (Afshar-Mohajer, et al., 2015). The static dust on the lunar surface is dissimilar to the dust on Earth. Due to the lack of atmosphere and reduced gravitational attraction, dust travels longer distances and has a higher propensity to make contact and create a higher susceptibility to lunar retention on space suit hardware (Wood, 1991).

The strategies for reducing the amount of regolith that is released into the habitable zone are discretely divided into three processes. Firstly, prevention will be identified to help reduce the amount of regolith that adheres to the xEMU. This will include methods in reducing the capture points on the xEMU based on the lessons learned from the Apollo program and reducing the amount that adhere specifically to the surface of the xPGS. The improved mobility of the suit will help to reduce contact points with regolith as the current EMU mobility is restricted. The mobility of the legacy EMU features both fiberglass and soft goods (i.e., fabric) items. Mobility features include jointed pleats and rotational bearings, although the current EMU has restricted severely when compared to its xEMU counterpart due primarily to torque-angle data, limited space suit joints and the excessive weight of the suit (Schmidt, 2001). The new geometry changes for xEMU allow for rotational bearings at the waist, shoulder joint mobility, reduced weight and EPG integration. A dust mitigating environmental protection garment (Ross, 2019). These anticipated design changes allow for reduced retention of regolith. Second, reduction will be identified to remove dust that has been captured by the suit and minimize dust that may re-adhere to the suit. These are both addressed in the new geometry proposal. Thirdly, the mitigation and contamination control are anticipated to prevent regolith from spreading into the habitat and proactively clean areas that have been contaminated in the habitat with unrelinquished suit regolith. The requirements for lunar regolith liberation are defined in both the Con-Ops for EVA and in the ADD for the hardware description. These two documents nearly approximate the collection of use cases but in a limited capacity. The Con-Ops (paraphrased) describe the lunar dust mitigation as such:

Table 19 - Dust Mitigation Operational Case

Activity	Lunar
Dust Mitigation	When on the lunar surface, crew will remove dust on the suit prior to ingress to habitat by limiting contamination. Dust mitigation methods will be limited to 15 minutes per crew member. TBD tools will be left external to the vehicle and/or habitat to effectively reduce the amount of dust liberated from the xEMU. In contingency scenarios, partial or no dust mitigation may be performed prior to ingress to habitat.

The ADD discusses dust mitigation in terms of verification activities based on the Level 1 expectations with lunar dust liberation. The assessments cover the functional survivability of xEMU hardware that comes in contact with lunar dust and the mitigation of dust during EVAs between habitat-to-habitat movement during ingress and egress. The lunar dust, in particular regolith, needs to be limited when returning to habitat. This liberation is regarding the return to cabin environment after an EVA. The regolith liberation requirement is different across the primary stakeholder groups in the HLS, EVA and Gateway systems, as each system has a different fulfillment due to functionality required. Thus, it is expected that the xEMU will need to satisfy at the EIS level the most conservative value to satisfy all HLS, EVA and Gateway system requirements. The EVA are listed as such with HLS/Gateway (as they are elicited in tandem due to cooperative, combined operations) shall meet the expired performance operational requirements of the 50 grams/crewmember after EVA in similar nature to the EVA requirements below:

Table 20 - Level 1 Dust Mitigation Requirement

Level 1: EVA Level Requirements				
EVA Requirement #	EVA Requirement	EVA Requirement Shall Statement	EVA Requirement Rationale	Traceability

R.SS-3033	Lunar Surface Dust Mitigation	The xEVA System shall limit the amount of regolith liberated in the cabin environment to less than 100 grams for each two-crew lunar surface EVA.	Extensive work has been done to establish a Permissible Exposure Limit (PEL) for Acute and Chronic exposures of flight crew to lunar surface regolith. The total value per two-crew EVA in this requirement is established to provide a worst-case bounding condition for nominal scenarios so that surface assets such as Human Landers can size Environmental Control/Life Support System (ECLSS) filters and other mitigation features provided by the vehicle can be designed to achieve the relevant Acute and Chronic PELs. 100 grams is based upon an allocation of no more than 50 grams per suit and the expectation that all lunar surface EVAs are conducted with two crew. It is acknowledged that this requirement is for nominal scenarios only, contingency events which lead to the termination or abort of a lunar surface EVA will likely reduce or eliminate the time and ability to execute dust mitigation activities. See EVA-EXP-0074, xEVA System Overview: Dust Mitigation for an explanation of the planned approach and methodology to provide a practical and verifiable system-level solution.	SSP 51073 EVA-RD-001 Rev B EVA Suit Systems Requirement Document, Section 3.317-18.

Table 21 - Lunar Dust Level 2 Requirement

Level 2: System Level Requirements of xEMU

xEMU System Requirement #	xEMU System Requirement	xEMU System Requirement Shall Statement	xEMU System Requirement Rationale	Traceability to EVA Requirement
xEMU.ENV.030	Lunar Dust Mitigation	The xEMU shall limit the regolith liberated in the cabin environment to less than 100 grams for each two-crew lunar surface EVA.	This requirement is derived from lessons learned from Apollo and Constellation projections and was conservatively reduced assuming better technologies and tools availability. The approach must be based on the ALARA principle, and to approach the 100-gram limit will require both design solutions and operational mitigations (i.e., cleaning) to achieve this goal on every EVA. This only applies to nominal scenarios and operations, as terminated/aborted EVAs will likely not allow for time to remove dust from the suit. Design considerations may include cleanability of the materials and minimization of crevices.	R.SS-3033

After the Level 2 requirements, Level 3 and Level 4 for the subsystem and EIS are defined, respectively. As all of the subsystems participate in lunar activity, subsequently they will all be subject to lunar dust and as a result lunar dust mitigation protocols. From here, the decomposition for the xPLSS, xINFO and xPGS requirements are defined and then requirements decomposed and allocated to the EIS for each subsystem at the component level. Due to the numerous requirements for xPGS, a review has been done across all regolith requirements and one set of requirements that is indicative of the nature of requirements decomposition will be presented, as the spirit of allocation is similar, although requirements across the same Level 3 echelon are different for xPGS. For xPLSS and xINFO, full decomposition across Level 2 is done, whereas for xPGS there are numerous requirements. The full requirements suite pertains to the regolith cleanability, entrapment, sealing, mechanism protection, repulsion, dissipation and temporary covering. For this case study, the cleanability and tooling removal of regolith is examined more closely than the design of dust tolerance as it satisfies the most representative of the Level 2 requirement framers' intent of the regolith liberation. As progression through the system requirement decomposition through the xPLSS, xPGS and xINFO commences, important detail must be examined as to the fulfillment of the Level 2 system requirement level. As there are numerous requirement, component and subsystem owners, the intent of the 10- gram per two-person regolith requirement was meant to satisfy the liberation of lunar dust into the cabin. A distinction, however, was never made as to what would reduce regolith retention. There is a design aspect, a cleaning aspect and a removal aspect. At the current moment, the fully decomposed requirements had different instances during their progression identified progressively the inadequacies of their development, however, due to the nature of convoluted definitions of prevention and liberation, the first concept of identifying a means to remove regolith has not yet been satisfied according to the current requirements to be stated. Secondly, as progression of decomposition continued, the identification of prevention by design was identified and much like the liberation requirement, is still in an inadequate state for requirement satisfaction tracing back to the needs, which were first developed to liberate regolith. At this moment, liberation of regolith begins with dust tolerant fabrics and mechanics, proceeds with cleanability requirements and tooling requirements for liberation during ingress/egress operations. In terms of requirements definition and decomposition, the xINFO Level 3 and Level 4 requirements are listed as follows:

Table 22 - Regolith Level 3 xINFO Subsystem Requirement

Level 3: Subsystem Level Requirements of xPLSS

xINFO Subsystem Requirement #	xINFO Subsystem Requirement	Subsystem Requirement Shall Statement	Subsystem Requirement Rationale	Traceability to System Req.
R.INFO.4025	Regolith Entrapment	The Informatics subsystem should limit design features that can trap environmental regolith to the most extent possible.	Design features such as deep folds, creases, seams, pockets, etc. can entrap or carry regolith back into the habitat. The habitat themselves have requirements on how much regolith their environmental system should be able to handle; thus, xEMU needs to try and minimize the amount of environmental regolith it potentially brings back into the habitat. This requirement goes hand in hand with the cleanability requirement as if these features are required that they can be cleaned as easily as possible.	xEMU.ENV.030

Table 23 - Regolith Level 4 xINFO EIS Requirement

Level 4: End Item Specification Level Requirements of DCU				
Lights EIS Component Requirement #	**Lights EIS Component Requirement**	**Component Requirement Shall Statement**	**Component Requirement Rationale**	**Traceability to Subsystem Req.**
R.LIT.4127	Regolith Entrapment	The Informatics lights should limit design features that can trap environmental regolith to the most extent possible.	Design features such as deep folds, creases, seams, pockets, etc. can entrap or carry regolith back into the habitat. The habitat themselves have requirements on how much regolith their environmental system should be able to handle; thus, xEMU needs to try and minimize the amount of environmental regolith it potentially brings back into the habitat. This requirement goes hand in hand with the cleanability requirement as if these features are required that they can be cleaned as easily as possible.	R.INFO.4025

This decomposition ends with the lights as the xINFO subsystem externally (much like the xPLSS subsystem), does not comprise the majority of the regolith collection needing regolith collection. Upon examination, the Level 4 requirement nearly borrows identically from the Level 3 requirements, leading to a disagreement between atomic and decomposable requirements per INCOSE standards, as the “shall” statement and rationale are not dissimilar enough to merit proper allocation. In terms of naturals language, both "shall" statements at Level 3 and Level 4 use "shall" which is a violation of INCOSE best practices in terms of writing standards. As these pertain to regolith liberation, there is no statement that clearly specifies how liberation occurs in terms of preventable design nor does a tool or otherwise is specified as a means of removal. Additionally, although a simulant could be used to replicate lunar dust and regolith liberation, no verification or controls are in place to satisfy the measurement of liberation on the lunar surface before a return to habitat. The xPLSS subsystem decomposition from Level 2 is as follows:

Table 24 -EPG Coverage xPLSS Subsystem Requirement

Level 3: Subsystem Level Requirements of xPLSS				
xPLSS Subsystem Requirement #	**xPLSS Subsystem Requirement**	**Subsystem Requirement Shall Statement**	**Subsystem Requirement Rationale**	**Traceability to System Req.**
R.PLSS.700.00	EPG Coverage (Z-003 EPG)	The PLSS shall, for each specified major assembly, provide an independent EPG	The VAC-802 and VAC-1004 are not included as they are vacuum umbilicals that flow rarified gases with little concern of condensation or freezing. The EPG for ESCU-801 needs to address thermal but not-necessarily the same constraints that the EPG needs to address for the PGS or PLSS/DCU applications. Hence, in order to reduce system mass, it is advised to address the required thermal performance for IVA application of the ESCU-801 in which the sink temperature of the crewlock walls is seen with static water conditions.	xEMU.ENV.030

Table 25 – EPG Coverage xPGS EIS Requirement

Level 4: End Item Specification Level Requirements				
EIS Component Requirement #	**EIS Component Requirement**	**Component Requirement Shall Statement**	**Component Requirement Rationale**	**Traceability to Subsystem Req.**

R.EPG.144	xPLSS EPG General Coverage	The full EPG layup, shall be installed on all external surfaces of the xPLSS subsystem unless explicitly stated otherwise in this section.	The EPG provides the xPLSS subsystem with a protective layer from the external environments. EPG properties, as defined by the requirements in this document, are a factor in subsystem and system performance. However, some limited deviations are necessary to support specific functionality.	R.PLSS.700.00

The EPG layup, an extension of the xPGS subsystem, is used to satisfy the regolith protection in terms of coverage. Much like the xINFO subsystem, the xPLSS contains no statement that clearly specifies how liberation occurs in terms of preventable design nor does a tool or otherwise is specified as a means of removal. Additionally, although a simulant could be used to replicate lunar dust and regolith liberation, no verification or controls are in place to satisfy the measurement of liberation on the lunar surface before a return to habitat.

Regolith cleanability suggest the ability to liberate regolith after an EVA. The xPGS cleanability is defined in the Regolith Cleanability Level 3 requirement under R.PGS.4222. This subsystem rationale requirement shall statement aligns very closely with the system shall statement at the xEMU.ENV.030 level, “The xEMU shall limit the regolith liberated in the cabin environment to less than 100 grams for each two-crew lunar surface EVA.” The xPGS root requirement at Level 3 before Level 4 decomposition is defined as follows:

Table 26 - Regolith Cleanability Level 3 xPGS Subsystem Requirement

Level 3: Subsystem Level Requirements of xPGS				
xPGS Subsystem Requirement #	**xPGS Subsystem Requirement**	**Subsystem Requirement Shall Statement**	**Subsystem Requirement Rationale**	**Traceability to System Req.**
R.PGS.4222	Regolith Cleanability	The PGS shall allow for the cleaning of environmental regolith from surfaces with external items so that the maximum amount of lunar dust liberated from the suit does not exceed 50 g per crewmember per EVA.	Over time environmental regolith can abrade and/or embed into surfaces. If regolith gets past initial lines of defense, like the EPG, the item should be able to be thoroughly cleaned to remove as much regolith as possible to extend the life of the hardware. This is primarily aimed at areas where creases, folds, seams, etc. may entrap or allow accumulation of regolith.	xEMU.ENV.030, xEMU.ENV.010

The shall statement is similar to the traceable parent requirement xEMU.ENV.030, with the difference of referring to a singular crew member with a 50-gram limit instead of a two-person crew with a 100-gram limit. The xPGS contains the majority of the equipment needing regolith liberation and this includes the suit's arms, boots, EPG, gloves, helmet, HUT, legs, shoulders, hatch and waist/brief/hip assemblies. There is not further decomposition of these items as shown in the Level 4 EIS requirements at the specification level listed below:

Table 27 - Cleanability Level 4 xPGS EIS Requirements

Level 4: End Item Specification Level Requirements				
EIS Requirement #	**EIS Subsystem Requirement**	**EIS Requirement Shall Statement**	**EIS Requirement Rationale**	**Traceability to Subsystem Req.**
R.ARMS.420	Lunar Regolith/Dust - Cleanability	The Arms Assembly shall allow for the cleaning of environmental lunar regolith/dust from surfaces.	Over time environmental lunar regolith/dust can abrade and/or embed into surfaces. If lunar regolith/dust gets past initial lines of defense, like the EPG, the item should be able to be thoroughly cleaned to remove as much lunar regolith/dust as possible to extend the life of the hardware. This is primarily aimed at areas where creases, folds, seams, etc. may entrap or allow accumulation of regolith.	R.PGS.4222

R.BOOT.422	Lunar Regolith/Dust - Cleanability	The Boot Assembly shall allow for the cleaning of environmental lunar dust/regolith from surfaces.	Over time environmental lunar regolith/dust can abrade and/or embed into surfaces. If lunar regolith/dust gets past initial lines of defense, like the EPG, the item should be able to be thoroughly cleaned to remove as much lunar regolith/dust as possible to extend the life of the hardware. This is primarily aimed at areas where creases, folds, seams, etc. may entrap or allow accumulation of regolith. It is expected to have some type of cleaning outside or inside the airlock to remove as much loose lunar dust/regolith as possible from the exterior of the suit before coming into the habitable volume. This requirement is to allow cleaning in, on, and behind the EPG and the inside of the suit, if required, during an IVA cleaning process that is beyond the standard wiping clean interior surfaces of [TBD].	R.PGS.4222
R.BOOT.517	Cleanliness	The Boot Assembly components shall be maintained clean to level Generally Clean (GC) per JPR5322.1.	TBD	R.PGS.4222
R.EPG.156	Regolith Cleaning	The EPG shall allow for the cleaning of environmental regolith from surfaces.	Over time environmental lunar regolith particles can abrade and/or embed into surfaces. The EPG should be able to be thoroughly cleaned to remove as much regolith as possible to extend the life of the hardware. This is primarily aimed at areas where creases, folds, seams, etc. may entrap or allow accumulation. It is expected to have some type of cleaning outside or inside the airlock to remove as many loose particles as possible from the exterior of the suit before coming into the habitat volume. This requirement is to allow cleaning on the external and internal surfaces of the EPG if required, during an IVA cleaning process that is beyond the standard maintenance cleaning.	R.PGS.4222

R.GLVS.418	Lunar Regolith/Dust - Cleanability	The Gloves Assembly shall allow for the cleaning of environmental lunar regolith/dust from surfaces.	Over time environmental regolith can abrade and/or embed into surfaces. If regolith gets past initial lines of defense, like the EPG, the item should be able to be thoroughly cleaned to remove as much regolith as possible to extend the life of the hardware. This is primarily aimed at areas where creases, folds, seams, etc. may entrap or allow accumulation of regolith. It is expected to have some type of cleaning outside or inside the airlock to remove as much loose regolith as possible from the exterior of the suit before coming into the habitable volume. This requirement is to allow cleaning in, on, and behind the EPG and the inside of the suit, if required, during an IVA cleaning process that is beyond the standard wiping clean interior surfaces to Generally Clean per xEMU.RMLL.008.	R.PGS.4222
R.HELM.501	Cleanliness	The Helmet Assembly shall be cleaned to level Visually Clean (VC)-Sensitive and maintained to level Generally Clean (GC) until launch, per JPR 5322.1H, Contamination Control Requirements Manual.	Because the xEMU has external surfaces that are sensitive to contamination and could interface with ISS, xEMU will adhere to SN-C-0005 via SSP 50835 for cleanliness. VC-Sensitive meets the intent of SSP 50835, but because SN-C-0005 was retired in 2011, the current JSC Contamination Control Requirements Manual (JPR 5322.1) is referenced.	R.PGS.4222
R.HUT.503	Cleanliness	The HUT hardware and components shall be cleaned to level Visibly Clean Sensitive (VC-S) + Ultraviolet (UV) and maintained to level Generally Clean (GC), per JPR5322.1H	TBD	R.PGS.4222

R.LEG.420	Cleanliness	The Legs Assembly shall allow for the cleaning of environmental lunar regolith/dust from surfaces.	Over time environmental lunar regolith/dust can abrade and/or embed into surfaces. If lunar regolith/dust gets past initial lines of defense, like the EPG, the item should be able to be thoroughly cleaned to remove as much lunar regolith/dust as possible to extend the life of the hardware. This is primarily aimed at areas where creases, folds, seams, etc. may entrap or allow accumulation of lunar regolith/dust. It is expected to have some type of cleaning outside or inside the airlock to remove as much loose lunar regolith/dust as possible from the exterior of the suit before coming into the habitat volume. This requirement is to allow cleaning in, on, and behind the EPG and the inside of the suit, if required, during an IVA cleaning process that is beyond the standard wiping clean interior surfaces to Generally Clean per xEMU.RMLL.008. extend the life of the hardware. This is primarily aimed at areas where creases, folds, seams, etc. may entrap or allow accumulation of regolith. It is expected to have some type of cleaning outside or inside the airlock to remove as much loose regolith as possible from the exterior of the suit before coming into the habitable volume. This requirement is to allow cleaning in, on, and behind the EPG and the inside of the suit, if required, during an IVA cleaning process that is beyond the standard wiping clean interior surfaces to Generally Clean per xEMU.RMLL.008.	R.PGS.4222
R.SHDR.501	Cleanliness	The Shoulder Assembly shall be capable of being cleaned to the level of Generally Clean on internal and external surfaces per JPR 5322.1, Contamination Control Requirements Manual.	TBD	R.PGS.4222

R.VLMH.117	Regolith - Cleanability	The VLM Hatch shall allow for the cleaning of environmental regolith from surfaces.	Over time environmental regolith can abrade and/or embed into surfaces. If regolith gets past initial lines of defense, like the EPG, the item should be able to be thoroughly cleaned to remove as much regolith as possible to extend the life of the hardware. This is primarily aimed at areas where creases, folds, seams, etc. may entrap or allow accumulation of regolith. It is expected to have some type of cleaning outside or inside the airlock to remove as much loose regolith as possible from the exterior of the suit before coming into the habitable volume. This requirement is to allow cleaning in, on, and behind the EPG and the inside of the suit, if required, during an IVA cleaning process that is beyond the standard wiping clean interior surfaces to Generally Clean per xEMU.RMLL.008.	R.PGS.4222
R.WBH.417	Lunar Regolith/Dust - Cleanability	The WBH Assembly shall allow for the cleaning of environmental lunar dust/regolith from surfaces.	Over time environmental lunar dust/regolith can abrade and/or embed into surfaces. If lunar dust/regolith gets past initial lines of defense, like the EPG, the item should be able to be thoroughly cleaned to remove as much lunar dust/regolith as possible to extend the life of the hardware. This is primarily aimed at areas where creases, folds, seams, etc. may entrap or allow accumulation of lunar dust/regolith. It is expected to have some type of cleaning outside or inside the airlock to remove as much loose lunar dust/regolith as possible from the exterior of the suit before coming into the habitable volume. This requirement is to allow cleaning in, on, and behind the EPG and the inside of the suit, if required, during an Intra-Vehicular Activity (IVA) cleaning process that is beyond the standard wiping clean interior surfaces to [TBD].	R.PGS.4222

R.WBH.519	General Cleanliness	The WBH Assembly shall be capable of being cleaned to the level of Generally Clean on internal and external surfaces per JPR 5322.1, Contamination Control Requirements Manual.	TBD	R.PGS.4222

The motivation behind the R.PGS.4222 Level 3 requirement, driven by the xEMU.ENV.030 Level 2 requirement, was to create a set of requirements that illustrate and mitigate the liberation of regolith. Most of these requirements as stated help to verify a certain level of cleanliness (i.e., "…shall be capable of being cleaned to the level of Generally Clean on internal and external surfaces…") and allow for the ability to be cleaned (i.e., shall allow for the cleaning of environmental lunar dust/regolith from surfaces). In this case, it seems as if there are two things to identify at this level and should be done for each part: (1) the cleanliness level (i.e., Generally Clean, Visibly Clean, Very Clean, Precision Clean, etc.) and (2) the capability of the part being cleaned. This will help verify how clean the suit is and how you can clean the suit. Additionally, and less defined, was how parts on the xEMU, specifically those pertaining to the xPGS subsystem as the majority of the xEMU is comprised of these components, is the design for the suit to be dust tolerant. There is an incoherence across all the EIS requirements. Certain requirements call for a specific level of cleanliness while others dictate that there should be capability of being cleaned. There is no specific process for measuring the amount of regolith retained or liberated nor is a directed method in terms of tooling to remove said regolith and there is no further decomposition of these items as shown in the Level 4 EIS requirements.

The next set of requirements pertain to the entrapment of regolith; manners in which the dust can be limited to entrapment based on design and soft goods selection. These start with the Level 3 requirements as follows:

Table 28 - Entrapment Level 3 xPGS Subsystem Requirement

Level 3: Subsystem Level Requirements of xPGS				
xPGS Subsystem Requirement #	**xPGS Subsystem Requirement**	**Subsystem Requirement Shall Statement**	**Subsystem Requirement Rationale**	**Traceability to System Req.**
R.PGS.4224	Regolith - Entrapment	The PGS should limit design features that can trap environmental regolith to the most extent possible.	Design features such as deep folds, creases, seams, pockets, etc. can entrap or carry regolith back into the habitat. The habitat themselves have requirements on how much regolith their environmental system should be able to handle; thus, xEMU needs to try and minimize the amount of environmental regolith it potentially brings back into the habitat. This requirement goes hand in hand with the cleanability requirement as if these features are required that they can be cleaned as easily as possible.	xEMU.ENV.030

This Level 3 requirement does not currently follow INCOSE writings standards as the "shall" statement uses the word "should." It is also advisable in the "shall" statement which techniques, benchmarked processes or specific approaches could be used. The rationale could potentially elaborate on some of these best practices, particularly citing current NASA guidelines or any information retained from the Apollo missions. These subsystem requirements for regolith entrapment decompose into the following EIS requirements:

Table 29 - Entrapment Level 4 xPGS EIS Requirements

Level 4: End Item Specification Level Requirements				
EIS Component Requirement #	**EIS Component Requirement**	**Component Requirement Shall Statement**	**Component Requirement Rationale**	**Traceability to Subsystem Req.**
R.ARMS.417	Lunar Regolith/Dust - Entrapment	The Arms Assembly should limit design features that can trap environmental lunar regolith/dust to the most extent possible.	Design features such as deep folds, creases, seams, pockets, etc. can entrap or carry lunar regolith/dust back into the habitat. The habitat themselves have requirements on how much lunar regolith/dust their environmental system should be able to handle; thus, xEMU needs to try and minimize the amount of environmental lunar regolith/dust it potentially brings back into the habitat. This requirement goes hand in hand with the cleanability requirement as if these features are required that they can be cleaned as easily as possible.	R.PGS.4224
R.BOOT.421	Lunar Regolith/Dust - Entrapment	The Boots should limit design features that can trap lunar regolith/dust to the most extent possible.	Design features such as deep folds, creases, seams, pockets, etc. can entrap or carry regolith back into the habitat. The habitat themselves have requirements on how much regolith their environmental system should be able to handle; thus, xEMU needs to try and minimize the amount of environmental regolith it potentially brings back into the habitat. This requirement goes hand in hand with the cleanability requirement as if these features are required that they can be cleaned as easily as possible.	R.PGS.4224
R.EPG.014	Closure Verified Sealed	EPG separable components will be verified closed by TBD method.	TBD	R.PGS.4224
R.GLVS.416	Lunar Regolith/Dust - Entrapment	The Gloves Assembly should limit design features that can trap environmental lunar regolith/dust to the most extent possible	Design features such as deep folds, creases, seams, pockets, etc. can entrap or carry regolith back into the habitat. The habitat themselves have requirements on how much regolith their environmental system should be able to handle; thus, xEMU needs to try and minimize the amount of environmental regolith it potentially brings back into the habitat. This requirement goes hand in hand with the cleanability requirement as if these features are required that they can be cleaned as easily as possible.	R.PGS.4224

R.HELM.906	Regolith - Entrapment	The Helmet should limit design features that can trap environmental regolith to the most extent possible.	TBD	R.PGS.4224
R.LEG.417	Lunar Regolith/Dust - Entrapment	The Legs Assembly should limit design features that can trap environmental lunar regolith/dust to the most extent possible.	Design features such as deep folds, creases, seams, pockets, etc. can entrap or carry lunar regolith/dust back into the habitat. The habitat themselves have requirements on how much lunar regolith/dust their environmental system should be able to handle; thus, xEMU needs to try and minimize the amount of environmental lunar regolith/dust it potentially brings back into the habitat. This requirement goes hand in hand with the cleanability requirement as if these features are required that they can be cleaned as easily as possible.	R.PGS.4224
R.SHDR.907	Regolith - Entrapment	The Shoulder should limit design features that can trap environmental regolith to the most extent possible.	Design features such as deep folds, creases, seams, pockets, etc. can entrap or carry regolith back into the habitat. The habitat themselves have requirements on how much regolith their environmental system should be able to handle; thus, xEMU needs to try and minimize the amount of environmental regolith it potentially brings back into the habitat. This requirement goes hand in hand with the cleanability requirement as if these features are required that they can be cleaned as easily as possible.	R.PGS.4224
R.VLMH.122	Regolith - Entrapment	The VLM Hatch should limit design features that can trap environmental regolith to the most extent possible.	Design features such as deep folds, creases, seams, pockets, etc. can entrap or carry regolith back into the habitat. The habitat themselves have requirements on how much regolith their environmental system should be able to handle; thus, xEMU needs to try and minimize the amount of environmental regolith it potentially brings back into the habitat. This requirement goes hand in hand with the cleanability requirement as if these features are required that they can be cleaned as easily as possible.	R.PGS.4224

R.WBH.418	Lunar Regolith/Dust - Entrapment	The WBH Assembly should limit design features that can trap environmental lunar dust/regolith to the most extent possible.	Design features such as deep folds, creases, seams, pockets, etc. can entrap or carry lunar dust/regolith back into the habitat. The habitat themselves have requirements on how much lunar dust/regolith their environmental system should be able to handle; thus, xEMU needs to try and minimize the amount of environmental lunar dust/regolith it potentially brings back into the habitat. This requirement goes hand in hand with the cleanability requirement as if these features are required that they can be cleaned as easily as possible.	R.PGS.4224

In similar fashion to the Level 3 requirements, the Level 4 requirements do not currently follow INCOSE writings standards as the “shall” statement uses the word “should.” In one case, the “shall” statement is listed as, EPG separable components will be verified closed by TBD method,” which does not list the separable EPG components (at the Level 4 or possibly even decomposed further to define all separable components). Any TBX listing is appropriate, however, during this stage of the project where design is finished or nearing completion, TBX requirements should be defined. It is also advisable in the “shall” statement which techniques, benchmarked processes or specific approaches could be used. The rationale could potentially elaborate on some of these best practices, particularly citing current NASA guidelines or any information retained from the Apollo missions. Much of the rationale is copied over from requirement to requirement, with minimal context. In summary, all of the piece parts which comprise the majority of regolith liberation and dust tolerant needs specify the need for said liberation and tolerance however do not adequately and precisely define how, nor do any have verifications or controls in place at the subsystem integrated level for testing, the system level for testing or system level in-field usage.

The next set of requirements pertain to the mechanism protection of regolith; manners in which the dust can be limited due to the mechanical design and development.

Table 30 - Mechanism Protection Level 3 xPGS Subsystem Requirement

Level 3: Subsystem Level Requirements of xPGS				
xPGS Subsystem Requirement #	**xPGS Subsystem Requirement**	**Subsystem Requirement Shall Statement**	**Subsystem Requirement Rationale**	**Traceability to System Req.**
PGS.4228	Regolith - Mechanism Protection	The PGS shall utilize environmental regolith protections that preclude environmental regolith from contaminating mechanisms where applicable.	If regolith is allowed into mechanisms, it can cause premature wear and/or failure of a mechanism. It is beneficial to have features that could trap and/or prevent regolith from contaminating the essential features of the mechanisms. Means of preclusion can include but not limited to, passive preventions design features like seals, filter, brushes and traps, and/or active features like electro-static dust repulsion. If using passive means, making them replaceable, cleanable, or both, can further extend the life of the mechanism.	xEMU.ENV.030

Per INCOSE writing standards, this Level 3 requirement is satisfactory. It is recommended however that the mention of, "…where applicable," be substantiated in a more quantifiable sense at this level of decomposition. These subsystem requirements for regolith mechanism protection decompose into the following EIS requirements at Level 4:

Table 31 - Mechanism Level 4 xPGS EIS Requirement

Level 4: End Item Specification Level Requirements

EIS Component Requirement #	EIS Component Requirement	Component Requirement Shall Statement	Component Requirement Rationale	Traceability to Subsystem Req.
R.ARMS.419	Lunar Regolith/Dust - Mechanism Protection	The Arms Assembly shall utilize environmental lunar regolith/dust protections that preclude environmental lunar regolith/dust from contaminating mechanisms where applicable.	If lunar regolith/dust is allowed into mechanisms it can cause premature wear and/or failure of a mechanism. It is beneficial to have features that could trap and/or prevent lunar regolith/dust from contaminating the essential features of the mechanisms. Means of preclusion can include but not limited to, passive preventions design features like seals, filter, brushes and traps, and/or active features like electro-static dust repulsion. If using passive means, making them replaceable, cleanable, or both, can further extend the life of the mechanism.	PGS.4228
R.BOOT.420	Lunar Regolith/Dust - Mechanism Protection	The Boot Assembly shall utilize environmental lunar regolith/dust protections that preclude environmental lunar regolith/dust from contaminating mechanisms where applicable.	If regolith is allowed into mechanisms, it can cause premature wear and/or failure of a mechanism. It is beneficial to have features that could trap and/or prevent regolith from contaminating the essential features of the mechanisms. Means of preclusion can include but not limited to, passive preventions design features like seals, filter, brushes and traps, and/or active features like electro-static dust repulsion. If using passive means, making them replaceable, cleanable, or both, can further extend the life of the mechanism.	PGS.4228
R.EPG.014	Closure Verified Sealed	EPG separable components will be verified closed by TBD method.	TBD	PGS.4228
R.GLVS.417	Lunar Regolith/Dust - Mechanism Protection	The Gloves Assembly shall utilize environmental lunar regolith/dust protections that preclude environmental lunar regolith/dust from contaminating mechanisms where applicable.	If regolith is allowed into mechanisms, it can cause premature wear and/or failure of a mechanism. It is beneficial to have features that could trap and/or prevent regolith from contaminating the essential features of the mechanisms. Means of preclusion can include but not limited to, passive preventions design features like seals, filter, brushes and traps, and/or active features like electro-static dust repulsion. If using passive means, making them replaceable, cleanable, or both, can further extend the life of the mechanism.	PGS.4228
R.LEG.419	Lunar Regolith/Dust - Mechanism Protection	The Legs Assembly shall utilize environmental lunar regolith/dust protections that preclude environmental regolith from contaminating mechanisms where applicable.	If lunar regolith/dust is allowed into mechanisms it can cause premature wear and/or failure of a mechanism. It is beneficial to have features that could trap and/or prevent lunar regolith/dust from contaminating the essential features of the mechanisms. Means of preclusion can include but not limited to, passive preventions design features like seals, filter, brushes and traps, and/or active features like electro-static dust repulsion. If using passive means, making them replaceable, cleanable, or both, can further extend the life of the mechanism. Currently, ISS Demo and Initial Lunar will	PGS.4228

			use the EMU lower legs and there is no current plan to incorporate lunar regolith/dust seals for the initial lunar mission. A new xEMU leg design will incorporate lunar requirements as applicable and may be incorporated in time for the initial lunar mission.	
R.SHDR.909	Regolith - Mechanism Protection	The Shoulder shall utilize environmental regolith protections that preclude environmental regolith from contaminating mechanisms where applicable.	If regolith is allowed into mechanisms, it can cause premature wear and/or failure of a mechanism. It is beneficial to have features that could trap and/or prevent regolith from contaminating the essential features of the mechanisms. Means of preclusion can include but not limited to, passive preventions design features like seals, filter, brushes and traps, and/or active features like electro-static dust repulsion. If using passive means, making them replaceable, cleanable, or both, can further extend the life of the mechanism.	PGS.4228
R.VLMH.123	Regolith - Mechanism Protection	The VLM Hatch shall utilize environmental regolith protections that preclude environmental regolith from contaminating mechanisms where possible.	If regolith is allowed into mechanisms, it can cause premature wear and/or failure of a mechanism. It is beneficial to have features that could trap and/or prevent regolith from contaminating the essential features of the mechanisms. Means of preclusion can include but not limited to, passive preventions design features like seals, filter, brushes, and traps, and/or active features like electro-static dust repulsion. If using passive means, making them replaceable, cleanable, or both, can further extend the life of the mechanism.	PGS.4228
R.WBH.421	Lunar Regolith/Dust - Mechanism Protection	The WBH shall utilize environmental lunar regolith/dust protections that preclude environmental lunar regolith/dust from contaminating mechanisms where applicable	If lunar regolith/dust is allowed into mechanisms it can cause premature wear and/or failure of a mechanism. It is beneficial to have features that could trap and/or prevent lunar regolith/dust from contaminating the essential features of the mechanisms. Means of preclusion can include but not limited to, passive preventions design features like seals, filter, brushes and traps, and/or active features like electro-static dust repulsion. If using passive means, making them replaceable, cleanable, or both, can further extend the life of the mechanism.	PGS.4228

Much like the predecessor requirements, the “shall” statement is identical detailing that, “…assembly shall utilize environmental lunar regolith/dust protections that preclude environmental lunar regolith/dust from contaminating mechanisms where applicable.”

Component rationale too is an identical from each Level 4 requirement, which does not necessarily constitute and underdeveloped requirement but begs the notion that differentiation across different parts of the xPGS aren't defined in a granular context at this level of decomposition.

The next set of requirements pertain to the repulsion and temporary covering protection of regolith; manners in which the dust can be limited due to repulsion via electrostatic or electrodynamic means and disposable covering. Note that these two Level 3 requirements are comparative and thus categorized in unison as their decomposed counterparts at Level 4 are stated in similar fashion.

Table 32 – Regolith Repulsion & Covering Level 3 xPGS Requirements

Level 3: Subsystem Level Requirements of xPGS				
xPGS Subsystem Requirement #	**xPGS Subsystem Requirement**	**Subsystem Requirement Shall Statement**	**Subsystem Requirement Rationale**	**Traceability to System Req.**
R.PGS.4232	Regolith - Repulsion	The PGS should utilize electro-static or electrodynamic means of regolith repulsion from surfaces.	Lunar regolith can carry charge and by using a similar charge an electromagnetic repulsion effect can be utilized to repel dust and fine soil from critical areas of the suit. It is unclear at the moment if dissipation of static electric charge in R.PGS.4234 can be harnessed to achieve both goals.	xEMU.ENV.030
R.PGS.4236	Regolith - Temporary Covers	The PGS should allow the use of disposable environmental covers externally on the suit.	Disposable covers may be used to protect the EPG from regolith. Covers could range from chap like pant covers to full bunny suits that would take the brunt of regolith damage from falls, kneeling, or other activities involving contact with the surface.	xEMU.ENV.030

Much like other Level 3 requirements at the xPGS and xINFO levels, the R.PGS.4232 and R.PGS.4236 requirements do not currently follow INCOSE writings standards as the "shall" statement uses the word "should." Otherwise at this particular level of decomposition, the writing and structure is adequate with exception to the "should" verb. These subsystem requirements for repulsion and temporary covering of regolith decompose into the following EIS requirements at the Level 4:

Table 33 - Regolith Covering Level 4 xPGS EIS Requirements

Level 4: End Item Specification Level Requirements				
EIS Component Requirement #	**EIS Component Requirement**	**Component Requirement Shall Statement**	**Component Requirement Rationale**	**Traceability to Subsystem Req.**
R.EPG.159	Regolith - Repulsion	The EPG should utilize electro-static or electrodynamic means of regolith repulsion from surfaces (TBD).	Lunar regolith can carry charge and by using a similar charge an electromagnetic repulsion effect can be utilized to repel dust and fine soil from critical areas of the suit.	R.PGS.4232
R.EPG.161	Regolith - Temporary Covers	The EPG should allow the use of disposable environmental covers externally on the suit.	Disposable covers may be used to protect the EPG from regolith. Covers could range from chap like pant covers to full bunny suits that would take the brunt of regolith damage from falls, kneeling, or other activities involving contact with the surface.	R.PGS.4236

These requirements for Level 4 have been allocated only to the EPG items. The "shall" statements include "should" verbiage, another violation of INCOSE best writing practices. These statements are also identical to the Level 3 requirements with the exception of the mention of the EPG. Verifications across the full decomposition of Level 3 and Level 4 requirements as they pertain to dust tolerance design, regolith liberation and cleanability/cleanliness levels mirror comparably with the verifications written for the boots (Note: for simplicity, only the boots EIS Level 4 is demonstrated with the notion that this methodology was applied successively to all other EIS Level 4 requirements for other xPGS portions of the suit including the EPG, helmet, waist/brief/hip, arms, legs, shoulders, hatch and gloves.

Table 34 - Verifications for Boots Design

EIS Requirement	Shall Statement	Verification Success Criteria	Verification Method	Subsystem Trace

R.BOOT.422	The Boot Assembly shall allow for the cleaning of environmental lunar dust/regolith from surfaces	Demonstration shall show that the Boot Assembly allows for the cleanability of lunar regolith/dust from surfaces. This is primarily aimed at areas where creases, folds, seams, etc. may entrap or allow accumulation of regolith.	Demonstration, Functional	R.PGS.4222
R.BOOT.517	The Boot Assembly components shall be maintained clean to level Generally Clean (GC) per JPR5322.1.	An inspection of the Boot Assembly components shall show that it is maintained clean to level GC on internal surfaces per JPR 5322.1.	Demonstration, Functional	R.PGS.4222
R.BOOT.421	The Boots should limit design features that can trap lunar regolith/dust to the most extent possible.	The Boot Assembly should limit design features that can trap environmental lunar regolith/dust to the most extent possible.	No Verification	R.PGS.4224
R.BOOT.420	The Boot Assembly shall utilize environmental lunar regolith/dust protections that preclude environmental lunar regolith/dust from contaminating mechanisms where applicable.	A design analysis of the Boot shall show that it utilizes environmental protections that preclude lunar regolith/dust from contaminating mechanisms where applicable.	Analysis, Qualitative	R.PGS.4228

The boots verifications are indicative of the representation across the different Level 4 parts of the xPGS and as such, provide a sample to qualify requirement evaluation of effectiveness in terms of robusticity. Per *INCOSE Guide for Writing Requirements,* system and verification guidelines, Section 1.2 dictate that development of these verifications and validations must happen prior to system acceptance and use. As of the writing of this dissertation, designs have been finalized and requirements verifications and validations are either partially finished, non-existent or by in large underdeveloped (Katz, et al, 2022). *INCOSE Guide for Writing Requirements* continues to elaborate on the textual needs fulfilling effective communication as the intended message needs to be clear, free from ambiguity and atomic (Katz, et al, 2022).

As the lunar dust requirements have been thoroughly reviewed, the current path forward for the performing organization and NASA is to perform a revamp to the current requirements structure. This will include a more robust decomposition of the requirements suite leading to a broader statement at the higher levels and more finely grained requirements allocation through the lower levels of the requirements. The goal is to satisfy the need to differentiate clearly between the ability to design a dust tolerant suit, a more concise and specific means to clean the suit and measure regolith liberation/retention on the suit and to categorize cleanliness levels per current NASA desired standards.

6.3. GSE Testing Team Development

The GSE Testing Team creates testing stations for various components across the xPLSS to perform both developmental and qualification testing and represent a case study in team development, traditional waterfall to Agile development, and an investigation into systems engineering verifications and validations across an individual subproject on xEMU.

6.3.1. Traditional Waterfall for Projects

Jacobs Technology follows a traditional waterfall framework for the majority of their projects, including xEMU-related teams. Historically, a waterfall framework was adequate as it pertained to general research and development as the customer would be unable to provide an inexplicit scope and was generally receptive to Jacobs Technology management when billed and given a target date. There was flexibility for both parties and progress was unhurried and methodical. The “Boots on the Moon” initiative instigated a paradigm shift in expectation for all stakeholders. Both the customer and performing organization found themselves at an impasse as neither organization was prepared to alter their traditional methodologies.

The entire JETS program in association with NASA sponsorship unveiled an aggressive approach to meeting the president’s expectation by launching the following campaign: Moon 2024. Jacobs Technology organized the GSE Testing Team in a more formal manner, required them to build new contracts with cost, scope and schedule detailed with formal acceptance from the customer and instituted a milestone deliverable in the form of Customer Review Points to periodically gain acceptance at each delivery gate of the project.

For the remainder of 2019 through 2021, Jacobs Technology's GSE Testing Team has unsuccessfully made the projected schedule, missing every milestone delivery date on the contract. In addition, engineers and managers have had difficulty securing scope and keeping customers engaged over the past year. Without progress on building and developing test stations, the customer's perception is that the budget spent is not reflecting the forward progress, thus suggesting that the Inspector General's push to commercialize the next generation space suit is merited.

The GSE Testing Team is also struggling with the size of its group. In an effort provide additional resource support to make the 2024 delivery date and crash the schedule, the GSE Testing Team increased personnel from 12 team members to over 40. Many of these team members do not have the ideal experience level in terms of career-related experience, aerospace experience or both. The team is not collocated and there is insufficient project management support with no relevant organization process assets to support the risk of the project failing to meet the objectives. An alternative framework following an agile methodology which is suitable for teams working in unknown environments with a flux in scope with a defined cost and schedule is hypothesized to improve team velocity and cohesion, aid in meeting schedule and cost by improving earned value metrics, engage all stakeholders properly and deliver a superior product than the current framework which Jacobs Technology operates.

The GSE Testing Team (the project team) is a collection of project managers (management); mechanical engineers, electrical engineers, project engineers, software engineers (the engineering team) and electro-mechanical and mechanical technicians (lab technicians) that perform specialized tasks in order to meet the project objectives. Jacobs Technology works in strong matrix environment according to the PMBOK, 6th edition. In this matrix within the Jacobs Technology interpretation, the project manager's authority is moderate, resource availability is high, control of the project budget is within the project manager's control, the position is full-time as well as that of the administrative staff. The goal of the GSE Testing Team is to create test stations for each individual component on the xPLSS and eventually to support a top-level assembly test. From this approach, the team follows the "Vee" model of successive verifications and validations at the Con-Ops and system level requirements drive the designs at a system and component level and implementation of testing verifies and validates component, subsystem and system aspects. Currently, there is a necessity for nearly 20 test stations to complete all xPLSS verifications and evaluations with the GSE Testing Team responsible to some capacity of the majority.

The project managers are responsible for building the project budget, helping define scope, maintaining the schedule, guiding the GSE Testing Team members both personally and professionally to project completion, identifying and maintaining stakeholder engagement, controlling communications, directly report to the customer (NASA) on project status and fulfills the role as risk manager. The project manager furthermore maintains control of delivery of the product to the customer, develops all contracts of the associated work and is part of the project closure team. Each branch in the organizational chart contains leads. These leads provide SME judgement to its respective branch, perform the necessary engineering/technician work when available and contribute expert judgement to project management regarding the velocity of the branch they lead.

Mechanical engineers (and to a supporting degree, drafters) are responsible for the mechanical design of the testing equipment. This may include direct or indirect support of the finite element analysis, stress calculations, computer-aided drafting and design and release of the drawings into the product data management library with the approval of their lead. The majority of the mechanical engineers on the GSE Testing Team have 1-3 years of experience and are resourced to the project team.

Electrical engineers are responsible for the electrical design of the testing equipment. This includes the power equipment calculations, providing electrical schematics to the mechanical team, guidance for electrical procurements to the project/test engineers and release of the electrical schematics into the product data management library with the approval of their functional manager. The majority of the electrical engineers on the GSE Testing Team have 1-3 years of experience and are resourced to the project team.

Software engineers are responsible for the coding architecture (i.e., LabVIEW) for the testing equipment, as each testing station relies on a graphical user interface (GUI) to operate. This includes developing a standard architecture, writing station-specific code, providing information for procurements of software and supporting hardware needed to the project engineer, and delivering a stand-alone user interface for the station in hardware/software package. The majority of software engineers on the GSE Testing Team members carry 3-5 years' experience and are resourced to the project team. These individuals represent the smallest group on the team with the highest amount of specialized skillset as LabVIEW certification and expertise is a commodity on the project.

Project and test engineers share similar roles and are responsible for guiding the completion of the test station and function to a limited capacity as a project manager onto their test station. Their responsibilities include participating in scope, cost, schedule and with the customer, relevant stakeholders and project managers of GSE, procure articles for the test station, guide the design engineers (mechanical, electrical, software) on the team to coalesce to support successful test station development and provide cost and schedule projections to the project manager for their compression and approval. The majority of project engineers on the GSE Testing Team are 5-10 years experienced and are exclusive members, dedicated to the project until its completion without any functional or resource manager to report to other than GSE Testing Team project management.

Electro-mechanical and mechanical technicians are responsible for building the test stations, calibrating equipment, reviewing completed drawings and procedures for fitness of use, collaborate with the engineering team to operate the station and work across many test stations at once. The majority of technicians on the GSE Testing Team are 10-15 years experienced and are resourced to the project team.

For the past several years, the GSE Testing Team has performed under a Level of Effort (LOE) contract. This LOE contract allows for no finite schedule but assigns a cost for the operating period (i.e., a fiscal year) with a scope that highlights milestones but is relatively variable in nature dependent upon the needs of the customer. The operating period is a fiscal year mirrors that of NASA's period of performance that begins October 1st and ends September 30th. The benefit of this type of contract is to allow for complex work that the customer is unsure or unaware will be needed and relies on the performing organization, Jacobs Technology, to operate as a conduit of knowledge for the customer. The GSE Testing Team will meet periodically with the customer and provide target finish dates with no specific deliverable other than information regarding possible testing, proposed designs of hardware and sustained work associated with xEMU as a greater whole.

In early 2019, President Trump issued a "Boots on the Moon" initiative to provide funding of approximately $4 billion dollars a year to facilitate the complete design, manufacturing, testing and delivery of a next generation spacesuit for lunar exploration for the year 2024 (Wall, 2019). As a result of this effort, NASA has transitioned some of the GSE Testing Team's test station development to progress from a less formal LOE contract to a more robust and formal Completion Form (CF) contract. This CF contract specifies a definite scope, cost and schedule. This radical change left the GSE Testing Team and Jacobs Technology with an aggressive goal with no platform adequate to confront the challenge. Historically, the team performs testing and design capabilities to developmental test and quality components on the xPLSS with Class III level of rigor, with certain testing stations to be built with some GSE or a hybrid of Class III/GSE. For this specific case study, one of the CF projects will focus on a 100% GSE test station. The test station is intended to perform qualification testing on the SWME which as stated in the xPLSS overview a heat-exchanger and de-gasser in the thermal loop. The process involves water in the liquid cooling ventilation garment (LCVG), donned by crew with a series of tubing passing the fluid throughout the suit, absorbing heat from the body while circulating. The warm water is pumped through the SWME as it evaporates water vapor while maintaining the flow of liquid water. The cooled water is recycled through the LCVG (Bue, et al, 2014). The SWME uses hollow fiber technology to perform the de-gassing and heat-exchanging as water passes through the hollow fibers, release heat through an exothermic reaction and degasses as fluid travels passively through the hollow fibers (Makinen, 2014). The GSE Testing Team would then need to develop a testing station to perform qualification on the aforementioned test article, the SWME.

The GSE Testing Team followed the traditional waterfall method typically instituted within the company. The bullets, descriptions and Table 35 to follow discuss the Customer Review Point phases which are detailed in the contract. Phases 1 through 6 indicate all of the instances where the customer will engage with the performing organization. Each point defines a description of what is to be reviewed and a percentage listed. For context, these four Phases generally take 1 to 2 years to complete depending on the complexity of the testing station, given there are no scheduling constraints. In essence, this means a customer will interact 4 times within those 1 to 2 years with the project team as these would cover more of the validation process while intermittent requirements reviewing, design, fabrication and testing by the performing organization constitute the verification process. During these review points, the customer will see a presentation from a project engineer that explain how the test station's progression status. The complete engineering team is there to offer support to the project engineer and the project manager will speak on the entire schedule, discuss budget and identify areas of clarification from the customer. Outside of this interaction, test readiness reviews and beyond are typically not under control of the test station team. The test station team will provide feedback and therefore anything after Phase 6 is considered on-going support from another team, with the project team considered matrix at that point.

- GSE Kickoff Meeting – JETS shall hold a kickoff meeting with the NASA customer to outline scope, conceptual design, schedule, and cost. Detailed test station development will initiate upon closure of this kickoff meeting. Meeting closure is indicated by final approval of this CF contract.

- Initial Design Review – JETS shall hold a meeting with the NASA customer to review the initial test station design. JETS shall have mechanical and electrical schematics complete for review with all instrumentation fully specified. JETS shall review how the functional schematics meet the test station requirements.

- Pre-Fabrication Design Review – JETS shall hold a review with the NASA customer once the design is ready for fabrication. JETS shall show completion of action items and incorporation of design changes assigned at the initial design review. JETS shall present all critical interfaces (i.e., interfaces with flight hardware) and all hazard controls in intimate detail.

- Pre-Test Readiness Review (Pre-TRR) – JETS shall hold a review with the NASA customer once the test fixture assembly is completed and ready for a branch TRR. The TRR board may assign actions; any such actions must be closed by the JETS test fixture Development Team. Closure of all TRR action items and approval by the board will indicate final delivery of the test fixture from the Development Team to the PLSS testing team.

- Test Readiness Review (TRR) – JETS shall present test readiness of the to the NASA customer and NASA Branch Chief or designee. The TRR board may assign actions; any such actions must be closed by the JETS test fixture Development Team. Closure of all TRR action items and approval by the board will indicate final delivery of the test fixture from the Development Team to the PLSS testing team.

- Test Station Delivery (TSD) - JETS shall present the test rig following a successful functional checkout and acceptance data package to the customer. Unless otherwise stated within the contract, JETS will consider the rig delivered and move on to project closure.

Table 35 - GSE Test Team Project Lifecycle Verification and Validation Points

Phase	Description	% Project Life
1	GSE Kick-Off Meeting	5%
2	Preliminary Design Review	10%
3	Prefabrication Design Review	50%
4	Pre-Test Readiness Review	80%
5	Test Readiness Review	90%
6	Test Station Product Delivery	100%

For purposes of the study, the performing organization's testing team's project will be referred to in two tenses: (1) the Waterfall model as defined by the traditional and on-going project approach and (2) the Scrum model as newly defined as our agile framework after an analysis on the success of the traditional waterfall approach. As of this dissertation, the testing station is currently approaching Phase 3, Pre-Test Readiness Review. Figure 33 below displays the organization hierarchy under the Waterfall model. In this approach, team members respond to change based on their availability and workload. No particular work package is specific to any individual on the GSE Testing Team to the extent of their specialization. The primary project manager (designated as Project Manager 1) will receive requirements from the customer (testing component owner or equivalent on the NASA side) and communicate those to a Lead. The secondary project manager (designated as Project Manager 2) performs many of the same activities as Project Manager 1 but in a more supplemental capacity. A Project Engineer will be assigned one of several of the components or supporting equipment by the Lead Engineer. The specific leads from each group will distribute work packages reactively to members in their respective legs from the workload and schedule determined by project management and coordination with all leads. Due to the breadth and depth of the GSE Testing Team's expectations from the customer, the project team could be considered doing many, smaller projects primarily because of the complexity and bandwidth needed to plan, execute, monitor and complete testing work. This shift from LOE to CF has the benefit of discerning these tasks more discretely. Of the organizational chart for the GSE Testing Team below, a fraction (roughly 1/5) of the team will be allocated to the case study testing station efforts and will include a project manager, electrical/mechanical/software engineers, drafters and technicians. This organizational chart is augmented slightly to account for fluctuation in personnel over the past years and also to display adequate resources to account for the shift to a more Scrum-style team.

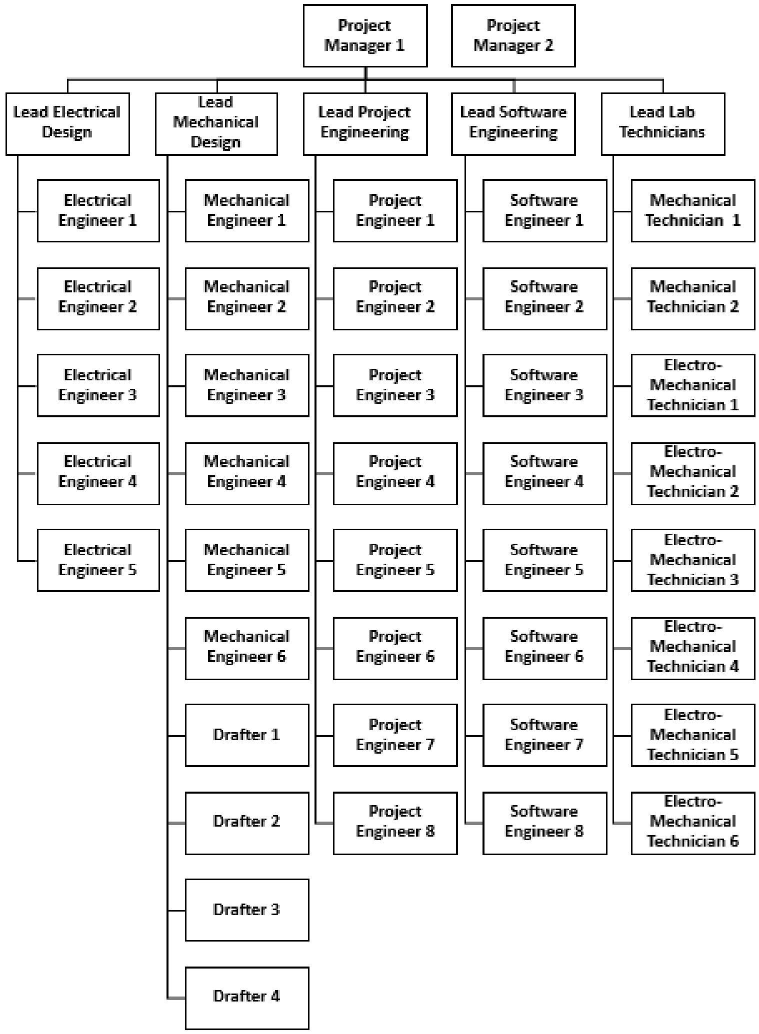


Figure 33 - Greater GSE Testing Team Organizational Chart

The customer, NASA, provides a level of priority for each of their components or supporting equipment which need a dedicated test station. It is important to note that no test station is contractually agreed upon until the GSE Kick-Off Meeting is held, where scope, cost and schedule are defined and signed and approved. The current method by which project management and engineering leads receive requirements from the customer are detailed before project conception and have no formal process. After the following testing stations are determined and given to project management, the test stations are delegated to the project engineers to distribute work packages and discuss schedule with project management in preparation of the first phase of the Customer Review Points. Project engineers will meet leads where they will assign electrical, software, lab technicians and mechanical engineers to provide initial designs and projections of team capacity. Once an initial schedule is built with costing based on a premature understanding of the scope, a project engineer will host a kick-off meeting on behalf of project management.

Commencing with the new project for SWME qualification, project engineers and project management met with the component owner of the SWME and NASA management to elicit feedback on the type of system needed. This would be a brainstorming session as defined by PMBOK 6th edition as a technique to identify a collection of ideas within a specified, and short period of time (Hunt, 2018). It is important to note that while this was a stakeholder meeting, it was not one of the four Customer Review points. It is also important to note that when developing requirements for the test station, there are no direct requirements dictating what the test station can and can't do. The test station in essence is an extension of its test article, the SWME, which has a complete set of requirements. Contractually, the new test station for the SWME would have its own requirements but will not be in Cradle® no decomposed. At the conclusion of the stakeholder meeting, the project engineer identified the rig requirements as a function of needs of the stakeholders and EIS of the test article. The test station would be known as the SWME Integrated Performance and Pressure Evaluator (SIPPE). The SIPPE on a high-level will perform the following:

- provide a GSE class test stand for testing of qualification & flight SWME/Mini-ME (miniature membrane evaporator) cartridges, assemblies, and valves. Test articles will include the following: cartridge(s) with/out inter-bundle temperature sensors, HX-440/SWME (spacesuit water membrane evaporator), HX-540/Mini-ME (miniature membrane evaporator, the backup unit in the event the primary SWME fails), and BPV (back pressure valve).
- provide a sub-ambient capable single loop mass flow & temperature regulated circulating fluid loop with high vacuum access and ambient air injection functionality.
- meet or exceed material requirements, measure heat rejection and degassing performance at SOCs (standard operating conditions), support freeze testing, and BPV characterization testing.

Before the first Customer Review point, the GSE Kick-Off Meeting, the schedule, scope of the test station and budget were given to NASA for review. The first iteration of budget and schedule lined out the following (Note: in previous sections of the dissertation, system/subsystem/component requirements were listed as they were deemed non-sensitive materials as they did not properly specify proprietary information. As this design is closer to more sensitive information, only the budget, schedule and scope at a high-level as previously illustrated will be disclosed). The baseline budget and schedule were defined as follows:

- Authorized Funding: (dollar value undisclosed but CPI available, includes all labor and materials)

- Start Date: 1/21/2020.
- End Date: 2/23/2021.

The contract in addition to the baselined budget and schedule includes the entire scope and only scope to be completed. The contract is signed by the project manager, component owner, design leads and representatives of the NASA customer. Other components of the contract are the project background, task description, drawing tree, documentation to be delivered supporting the testing station, requirements the test station will fulfill pertaining to the EIS Level 4 requirements and their verifications and validations, the conceptual or final mechanical/electrical designs depending on the progression or revision of the contract, exclusions to the contract, risks identified with mitigations, delivery method and a milestones/deliverables table. The milestones and deliverables table for the SIPPE contract is listed as follows:

Table 36 - Original Baseline for SIPPE Milestones & Deliverables Chart

Milestone/Deliverable	**Due Date**
SIPPE Test Station Development	
1. Milestone: Host SIPPE GSE Kickoff Meeting	01/21/2020
2. Milestone: Host SIPPE Initial Design Review	03/04/2020
3. Deliverable: Fabrication Release of all SIPPE Drawings	06/29/2020
4. Milestone: Host SIPPE Pre-Fabrication Design Review	07/08/2020
5. Milestone: Host SIPPE Pre-Test Readiness Review (Pre-TRR)	11/24/2020
6. Milestone: Host Vacuum System Pre-Test Readiness Review (Pre-TRR)	12/01/2020
7. Milestone: Host SIPPE Test Readiness Review (TRR)	12/08/2020

8. Deliverable: SIPPE Hazard Analysis Document	12/14/2020
9. Milestone: Host Vacuum System Test Readiness Review (TRR)	12/15/2020
10. Deliverable: Vacuum System Hazard Analysis Document	12/21/2020
11. Deliverable: Final Release of all SIPPE Drawings (redlines to Fabrication Drawings)	12/30/2020
12. Deliverable: SIPPE Test Station Procedure Document	01/15/2021
13. Deliverable: SWME Test Article Procedure Document	01/22/2021
14. Deliverable: Vacuum System Procedure Document	01/22/2021
15. Deliverable: SIPPE Operator Certification Letters	01/29/2021
16. Deliverable: SIPPE Test Fixture Serial Number 1, Test Station Delivery (TSD)	01/29/2021
17. Deliverable: Vacuum System Delivery	01/29/2021
18. Deliverable: SIPPE Test Fixture Serial Number 2, Test Station Delivery (TSD)	02/23/2021

The most current revision of budget and schedule were defined as follows:

- Authorized Funding: (dollar value undisclosed but CPI available, includes all labor and materials).
- Start Date: 1/21/2020.
- End Date: 2/28/2022 (projected to surpass this date).

Table 37 – Revision for SIPPE Milestones & Deliverables Chart

Milestone/Deliverable	**Due Date**
SIPPE Test Station Development	
1. Milestone: Host SIPPE GSE Kickoff Meeting	01/21/2020 (completed)
2. Milestone: Host SIPPE Initial Design Review	03/04/2020 (completed)
3. Deliverable: Fabrication Release of all SIPPE Drawings	09/21/2021 (completed)

4. Milestone: Host SIPPE Pre-Fabrication Design Review	01/14/2021 (late)
5. Milestone: Host SIPPE Pre-Test Readiness Review (Pre-TRR)	11/19/2021 (late)
6. Milestone: Host Vacuum System Pre-Test Readiness Review (Pre-TRR)	12/23/2021 (late)
7. Milestone: Host SIPPE Test Readiness Review (TRR)	12/10/2021 (late)
8. Deliverable: SIPPE Hazard Analysis Document	12/16/2021 (late)
9. Milestone: Host Vacuum System Test Readiness Review (TRR)	12/10/2021 (late)
10. Deliverable: Vacuum System Hazard Analysis Document	12/21/2021 (late)
11. Deliverable: Final Release of all SIPPE Drawings (redlines to Fabrication Drawings)	02/01/2022 (late)
12. Deliverable: SIPPE Test Station Procedure Document	11/29/2021 (late)
13. Deliverable: SWME Test Article Procedure Document	12/23/2021 (late)
14. Deliverable: Vacuum System Procedure Document	01/03/2022 (late)
15. Deliverable: SIPPE Operator Certification Letters	12/23/2021 (late)
16. Deliverable: SIPPE Test Fixture Serial Number 1, Test Station Delivery (TSD)	02/07/2022 (late)
17. Deliverable: Vacuum System Delivery	02/07/2022 (late)
18. Deliverable: SIPPE Test Fixture Serial Number 2, Test Station Delivery (TSD)	02/28/2022 (late)

EVMs were developed to characterize the deviation from the baseline of the project's cost and schedule. Cost Performance Index (CPI) and Schedule Performance Index (SPI) were derived from the Actual Cost (AC), Earned Value, (EV) and Planned Value (PV) metrics based on the Budget at Completion (BAC). Equations are found in the Appendix

(Morris, et al, 2010).

Table 38 – Earned Value Metric for SIPPE Project Lifecycle to Date

EVM Metric	Mar-21	Apr-21	May-21	Jun-21	Jul-21	Aug-21	Sep-21	Oct-21	Nov-21	Dec-21	Jan-22	Feb-22
CPI	0.3445	0.3784	0.4086	0.4329	0.4539	0.4778	0.4588	0.4853	0.5123	0.5356	0.5567	0.4995
SPI	0.3476	0.3643	0.3763	0.3935	0.4063	0.4156	0.4283	0.4310	0.4408	0.4613	0.4811	0.5010

EVM Metric	Jan-20	Feb-20	Mar-20	Apr-20	May-20	Jun-20	Jul-20	Aug-20	Sep-20	Oct-20	Nov-20	Dec-20	Jan-21	Feb-21
CPI	4.2452	4.2826	1.4140	0.8870	0.6285	0.4567	0.3746	0.3094	0.2986	0.3063	0.3155	0.3032	0.3106	0.3135
SPI	0.7432	0.8175	0.5632	0.4600	0.4042	0.3692	0.3452	0.3387	0.3336	0.3293	0.3257	0.3227	0.3265	0.3299

6.3.2. JETS Processes Influencing GSE Testing Team Development

Outside of the xEMU SE&I umbrella are other challenges facing the GSE Testing Team and their development of test stations. Due to the rigor of GSE pedigree, processes are in place that are meant to serve as quality measures however take substantial time and effort. These processes are the Jacobs design review processes and the fabrication and procurement of GSE hardware.

6.3.2.1. Design Process

JETS provides engineering support to NASA but follows a series of internal processes that are both contractually agreed or expected processes set forth by the performing organization that collectively determine how SIPPE should be designed. These documents are: (1) JPR 8500.4 Engineering Drawing System Manual, that establishes the JSC Engineering Drawing System requirements as the official medium to provide a medium for procedures to follow when planning, releasing, monitoring and controlling drawings and support documentation; (2) JSC-08080-2B Johnson Space Center Design & Procedural Standards, which contains design and procedural requirements for human spaceflight equipment based on best practices and lessons learned; (3) NASA-STD-5005 Standard for GSE Equipment, that establishes guidance and requirements for the fabrication and design of GSE with the intentions of providing a safe, reliable and robust, cost-effective design. With the new schedule that had been delivered by the administration with the "Boots on the Moon" initiative, the project management for the GSE Testing Team determined that several processes in place that had hampered the development of other projects lateral to SIPPE under the GSE Testing Team umbrella presented case studies onto themselves in order to mitigate the facilitation of process augmentation. There are two types of drawing release: (1) fabrication and (2) final release. Fabrication release allows for procurement of hardware without strict rigor while final release is the more formalized release of the drawing, which incorporates any redlines or as-built changes made during the fabrication process.

The first of these design processes that posed a challenge was the JETS Design Review Process. NASA expects to have an Engineering Drawing Control Center (EDCC) release of all GSE drawings. The EDCC is a Product Data Management (PDM) library that stores and controls NASA-related drawings. This system already allows for a robust review with quality and engineering signatures to verify the release. The JETS Design Review Process is a process where drawings must go through an extensive review by several of the resource/functional groups for quality and robustness of design. To support this process is the establishment of an internal review system document, the J391. The purpose of the J391 document is to provide a detailed design review of parts, drawings and assemblies before procurement by project engineering to ensure form, fit, function and quality adherence. The most important aspect that JETS hopes to manage is the risk with the intention that by having a robust system of reviews with all peer groups for all fabrication released drawings, project and technical risk are both mitigated. The following resource/functional groups are required to review all fabrication release drawings: Stress, Materials, Quality, Project Management, Mechanical/Electrical Lead, Engineering Group Manager, Engineer Director. Below is a flow chart from drawing inception to drawing release.

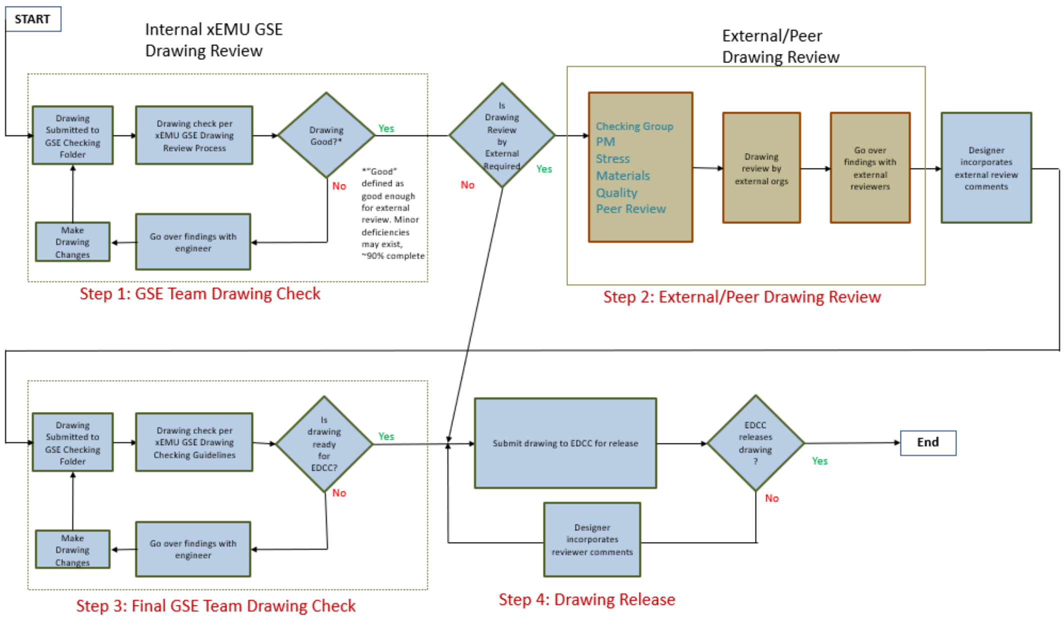


Figure 34 - GSE Drawing Review Process

Project management alongside the Mechanical Lead determined how to effectively plan for the release stage of drawings throughout the project lifecycle and determined to breakdown the following steps and completions based on expert judgement. As not all drawings would be the same, designs would vary from simple to standard to complex. Next, would be to organize their sequential steps and assign a percentage done with durations in weeks. Finally, buy-in from the team was considered during the development of the GSE Drawing Steps and Timeline chart, as per Agile and Sprint-organized teams (Griffiths, 2012). The conclusion is the entire process from conceptualization to release should take between 5 ½ weeks to 13 ½ weeks with the reviewer’s process taking between 2 ½ weeks to 4 ¼ weeks.

Table 39 – GSE Drawing Steps and Timeline Chart

Step	Fabrication Release Stage	Drawing Type			
		Simple	**Standard**	**Complex**	**Standard**
		Duration (weeks)			**% Complete**
0	CAD Conceptualization (FAB)	2	3	4	
1	Initial Drawing (FAB)	1	2	3	32%
2	Design Lead Review (FAB)	0.5	1	2	54%
3	Redlines Incorporated (FAB)	0.5	1	2	65%
4	J391 Reviewers (FAB)	0.5	1	1	76%
5	Design Lead Backcheck (FAB)	0.25	0.5	0.5	86%
6	Project Signatures (FAB)	0.25	0.25	0.25	92%
7	EDCC Release (FAB)	0.5	0.5	0.5	100%

The project engineers and project managers of the GSE Testing Team retained several metrics in order to understand the how previous projects operated with the J391 form and developed the following metrics with durations for review checks and discovered the following: simple drawings had a duration of 34 days; standard drawings had a duration of 58 days; complex drawings had a duration of 57 days. This far exceed the anticipated projection based on the GSE Testing Team for SIPPE's detailed design development. It is from the viewpoint of the customer that these JETS Design Review processes in addition to the NASA established EDCC review was hampering the ability for JETS to deliver products more quickly.

Table 40 - Drawing Reviewer Metrics

Total Duration	Drawing Difficulty
12	Standard
12	Standard
108	Simple
108	Simple
108	Simple
107	Complex

107	Complex
107	Complex
44	Complex
44	Complex
44	Complex
28	Simple
28	Simple
28	Simple
28	Simple
121	Complex
71	Complex
35	Standard
5	Complex
13	Complex
13	Standard
18	Standard
15	Complex
17	Complex
20	Standard
13	Simple
13	Simple
120	Standard
120	Standard
185	Standard
185	Standard
185	Complex
10	Complex
10	Simple
22	Simple
22	Simple
26	Complex
26	Standard
26	Simple
29	Standard
29	Standard
29	Simple
29	Standard
35	Complex
35	Standard
20	Simple
20	Simple

20	Complex
20	Simple
20	Simple
2	Simple

The second of these design processes that posed a challenge were the drawing methods. There are two drawing types identified by the design groups: Type I (Mono-Detailed Drawings) and Type II (Simple-Detailed Drawings). Type I drawings are intended to be used when fully detailed drawings are needed when configuration-based hazard controls are involved, primarily when utilizing high pressure systems. These detailed drawings include piping diagrams with fully denoted parts, electrical diagrams with annotations and separable parts lists. Type II drawings are those that do not require full three-dimensional models and are not as fully detailed when compared to their Type I counterparts. According to contractual agreements, the performing organization, Jacobs, recognizes the usage of releasing Type I drawings for GSE work whereas the customer, NASA, prefers the Type II drawings where applicable and in cases where risk is lower in order to expedite and facilitate a faster product delivery. It is from the viewpoint of the customer that these JETS Design Review processes in addition to the NASA established EDCC review was hampering the ability for JETS to deliver products more quickly. The argument from JETS is that the NASA proposal of utilizing Type II single, simplified drawings introduce higher risks than multiple, mono-detail drawing Type I drawings that JETS supports.

6.3.2.2. Procurement and Fabrication of GSE Hardware

GSE testing stations either directly or indirectly interface with flight hardware (i.e., Class I hardware). As a result of this interface, GSE hardware must be controlled similar in nature to flight hardware with the intention that its control will guarantee the form, fit, function, quality and safety criteria as the interface is deemed critical with respect to the flight hardware. GSE hardware juxtaposed to its Class III counterpart, requires additional process and quality control. GSE hardware is significantly more expensive and as a result increases budget and schedule to the GSE Testing Team's SIPPE project. This effort manifests itself during the procurement phase of hardware until the completion of fabrication. The timely delivery of hardware from vendors impacts schedule by impacting how quickly the delivered hardware may be fabricated. The timely delivery of the GSE Test Station in the SIPPE is impacted by the rigor of quality control and processes by building controlled hardware onsite at NASA Johnson Space Center, as GSE is allocated to this category.

First, the area of concern are the long lead times and expense applied to the procurement of GSE hardware. According to the performing organization contractual agreements (i.e., JPR 8500.4, JSC-08080-2B, NASA-STD-5005), GSE hardware must be purchased against stricter quality codes. As a result of these stricter codes, many vendors may be unable to provide the appropriate documentation thus overruling some of the available vendors. Providing additional paperwork in the form of material certifications, Certificates of Conformance (CofC) or lot traceability, these criterion present additional lead times and funding in addition to the minimal number of vendors available to provide the equipment. Class III hardware in comparison does not require this level of quality. Second, another area of concern is the extended period of fabrication at NASA Johnson Space Center. Historically, GSE-style testing stations have taken significant time to fabricate as a function of the excessive paperwork in terms of Technical Process Specifications (TPSs), Discrepancy Reports (DRs) and the Mandatory Inspection Points (MIPs) where the Quality department must send a presentative to review critical interfaces, fabrication with technicians and any pertinent paperwork attached. TPSs and DRs add considerable schedule time in terms of documentation generation, execution of work and closing of documentation. As a result of the aforementioned processes, two current GSE testing stations have been in development and fabrication for multiple years. While the reasons for their delays are not limited strictly to their fabrications, a considerable period of their development is due to the extended period of building GSE hardware onsite at NASA Johnson Space Center. It is from the viewpoint of the customer that these processes in addition to the NASA established fabrication of equipment onsite hampers JETS to deliver products more quickly. The argument from JETS is that the rigor of quality applied to GSE hardware during the procurement phase will adequately satisfy the Cost of Quality (COQ) and reduce risks associated with potential hardware discrepancies and furthermore argues that the necessary paperwork in terms of TPSs, DRs, MIPs and Quality department checks during fabrication also reduce risks and provide a rigor of quality for these GSE testing stations.

A third area of concern outside of the GSE designation is the extended periods parts are sent to the Calibration & Cleaning Departments which historically has taken other testing stations (GSE or Class III) considerable time to receive parts back from the department. Parts must be cleaned so as to guarantee the performance and calibrated so as to guarantee instruments are providing the proper readings. The root cause for extended periods in cleaning is due to the fact that the departments clean and calibrate parts for multiple projects, thus convoluting the need and insistence on schedule expectations. Currently, groups send their parts with paperwork on expected delivery dates from Calibration & Cleaning with various results with respect to actual deliveries. Due to the number of parts and projects involved, it is not possible to provide metrics for each group but for the GSE Testing Team, parts have been at Calibration & Cleaning for as little as a few days and for as long as approximately a year.

7. CHAPTER 7: DISCUSSION ON SYSTEMS ENGINEERING IMPROVEMENTS

Lede: This chapter illustrates the case studies from a discussion standpoint, outlining both the project lifecycle development and requirements engineering case studies. The results of the challenges across the systems engineering disciplines will follow a two-fold discussion where data was characterized as qualifiable and quantifiable as not all data could be analyzed comparably and equally. *Section 6.1* focused on and collected data on the requirements engineering development discussion. The collection process provided elements that while beneficial to research, were not sufficiently quantifiable to present a resolution with analytical results. Instead, data was evaluated and qualified empirically. In summary, *Section 6.1* will be evaluated with a proximate discussion of evaluation in *Section 7.1*. *Section 6.2* focused on and collected data regarding specific requirements for various hardware on the xEMU. Data collected was quantified and developed to propose and compare how requirements could be characterized against INCOSE standards and propose alternative methods that may have improved the requirements engineering process. *Section 6.3* focused on and collected data regarding the GSE Testing Team's development against a traditional waterfall project lifecycle. Data collected was quantified and developed to propose and compare how an Agile-based method could be frame-worked in specific instances to improve EVMs and how verifications and validations could be influenced with characterized against INCOSE standards and how these methods may have optimized budget, schedule and customer engagement. In summary, *Section 6.2* and *Section 6.3* will be evaluated with a more definite discussion of evaluation in *Section 7.2* and *Section 7.3*.

7.1. Requirements Engineering Development Discussion

The development of the requirements engineering was studied across the three major subsystems of xEMU relating to xPGS, xINFO and xPLSS with heavier emphasis on the xPLSS subsystem. In summary, the requirements engineering development in terms of system context change with nearly one-third of added or changed requirements, schedule compression by nearly two years, allocating nearly double the resources with many individuals unfamiliar with the system or previous untrained and complex engineering tasks as it applies to next generation space suit develop was given a proximate solution of being satisfactory given the challenges. The requirements were organized in a specification tree with the intention that the full scope of requirements is represented and the system architecture, Cradle®, was adequate in documenting and organizing the data. When a system context shift was indicated, a Delta-SRR was developed to gain full comprehension across all pertinent stakeholder groups as to the addition, subtraction and augmentation of requirements, especially when the context across all subsystems groups was as transformational as a shift from an ISS-based suit to a lunar suit. The three documents which drove the ability to create verifiable and validated requirements were the Con-Ops, the ADD and the PTRS. The PTRS was organized by Cradle® using current MBSE principles across two of the three subsystems while the ADD and Con-Ops in tandem provided information on functions and interactions of the equipment along with their descriptions. Proximately, the Con-Ops and ADD were adequate in providing the necessary data to build a requirements database. As it pertained to the case studies in *Section 7.2*, the addition of BDs, in specific SDs and UCDs could have contributed to the clarification for the need of an auxiliary lighting source and the clarification as to the appropriate needs and more accurate decomposition of lunar dust in terms of cleanability and prevention (a more robust analysis provided in *Section 7.2*). Proximately, the SEMP adequately described the technical approach for organizing product, team and process development with the intention of accomplishing the maturation of the project requirements with the focus of design definition and sustainment. Several of the recommendations given by the sample space of stakeholders across the project would have been the more representative organization of stakeholder groups in the SE&I organization chart located in *Section 3.1* and the release of the SEMP that had been in an everchanging release cycle for much of the period of the project lifecycle. One of the more controversial challenges with difficulty in codifying is the integration of the SoS of the three primary subsystems. The xPLSS subsystem did not follow the same approach in terms of MBSE standards that Cradle® was designed. Based on the questionnaire results and no other data that could be adequately collected or refutations to the claim, while the xPLSS subsystem developed their Level 3 requirements in a monolithic fashion (i.e., decomposed all of their requirements in one document) and reverse populated the requirements into Cradle® instead of using Cradle® to input requirements first then generate the specifications in Word document format, while there were in excess of 800 requirements that were at the xPLSS Level 3, there is no evidence or employee indication that this process caused any significant issues in requirements or product development. While it seems intuitive that all three subsystems should have been developed similarly and with the same rules and guidelines, the progressively developed xPLSS subsystem would have spent a generous amount of time to fully change their approach to accommodate for the inclusion of Cradle® and CM, which were not accounted for in their development years prior. As a result, project management allowed for xPLSS to operate outside of the Cradle® environment in terms of CM and MBSE standards and while this may have plausibly been an inconvenience, plausibly added time to schedule to manually input requirements into Cradle® and forgo the intended CM controls of Cradle®, there is no sufficient evidence that could be codified or quantified to refute the notion that the exclusion of CM controls, Cradle® inputs and MBSE standards were detrimental to xPLSS development. CM met with SE&I at a forum where requirements were discussed, updated and managed in a controlled environment with a CRs written for the CCB and recorded in a CCL. Furthermore, surveys to team members and managers indicate that one of the more critical facets to the challenge was the assumption that resource loading and schedule fast tracking were assumed to appropriately account for the large changes in system context while also bringing in the schedule approximately two years, which is more related to project management and project planning.

7.2. Case Studies in Requirements Discussion

With respect to the qualitative results found in *Section 7.1* with regards to requirements engineering decomposition, the great majority of the questionnaires from employees across the project found the approach of as adequate. The two case studies below highlight systems engineering challenges and while indicative of some of the greater challenges on the project, they do not necessarily define the entire approach or systematic method of requirements decomposition. The case studies are used to characterize the specific approaches against INCOSE standards and how certain aspects of the project's development may have been augmented to capture the framer's intent of decomposition more optimally for these two case studies.

7.2.1. Display & Control Unit Auxiliary Lighting Discussion

While the xPLSS and DCU subsystem and component were decomposed either by project lifecycle progression via the development team or recommendation efforts as a result of the study, the primary focus is on prevention and in this case the prevention of additional work and resources against design efforts. There was no intention from the primary stakeholder office to design an auxiliary (or also identified as secondary) lighting source to be designed into the set but exists outside of the suit as a peripheral. Just like many facets of project development, identification of risks, issues and challenges is best satiated at the beginning of the project lifecycle. While the efforts during Delta-SRR were adequate, the complexity for each individual subsystem and component at an elevated echelon may lose discernment upon immediate inspection. While the efforts of the ADD and Con-Ops were adequate, the usage of the suit across various, cross-functional stakeholder groups may cause the requirements to suffer deprivation of true customer needs. It is recommended that early in the project lifecycle that component owners and subsystem managers develop use cases within the Con-Ops, both in natural language and conceptual models to enrich needs, goals and requirement robusticity. These use cases are provided for the lighting sources, with a particular focus on the differentiation between a primary lighting source and auxiliary (secondary) lighting source. Further decomposition of use case tabular steps may be introduced but for simplicity, only the highest level main, alternative and exception scenario steps are utilized as they effectively resolve the differentiation between a designed auxiliary (secondary) lighting source and an already provided and designed in-house solution.

While a collection of needs, goals, scenarios, requirements with use case diagrams and natural/tabular language organizations are non-sequential as each elaborates and smooths decomposition, the first item of inspection is a cursory view of using a needs-to-requirements matrix in the context of worksite illumination. New requirements are shaded in orange and juxtaposed along with their existing, primary lighting requirements for worksite illumination. Any verbiage outside the nominal, xEMU requirements engineering nomenclature is merely notional for the example provided. The needs decompose into the desire for worksite illumination with designation for the xINFO to retain the primary lighting while the xPGS (suit side) will retain space on the xMWS for peripheral lighting (i.e., vacuum-ready flashlight source).

7.2.2. Lunar Dust Mitigation Discussion

While there are several areas of focus across creating a dust tolerant suit and mitigating the potential for regolith to carry into the habitat, the main focus of the analysis and subsequent recommendation is for the dust mitigation efforts in the context of liberation of regolith prior to ingress into lunar habitat. This will be a two-fold effort approach: (1) identification of the ability to successfully verify and validate this requirement and (2) characterize the methods via use case diagramming, natural language and tabular format. The main challenge associated with retention of less than 50 grams per crew member (100 gram for two member EVA) is the ability to measure the amount of regolith retained vs. regolith removed. As previously stated, the 100 gram per two member EVA was derived from previous Apollo missions in nearly 50 years. The challenge is both in the validity of the regolith via Apollo evidence assertion and the ability to measure how much regolith is not removed. Referring to the *INCOSE Guide to Writing Standards*, there will be no full-proof process by the performing organization to verify that the requirements have been written correctly and in turn the customer could struggle to validate if the performing organization is building the right product. This methodology will proliferate through the design and systems verifications and validations. There will be several variables to now consider as the xEMU suit differentiates from the Apollo suits. These will be and are not limited to:

- design differences between the Apollo suits and the xEMU suits with regards to regolith liberation as differentiation in design allows for a more modular suit but the question remains if this introduces cavities for regolith retention regardless of the suit's soft goods' ability to be dust tolerant.
- the area of EVA and amount of time spent on the lunar surface; the xEMU is designed for extended EVAs and by virtue could be exposed to the lunar environment for extended periods in comparison to the counterpart.
- the conditions of the lunar surface; while these may be static, collection of regolith may vary in different areas of the lunar surface.

In the context of understanding the operation associated with removal of lunar dust, Table 21 provided a paraphrased excerpt from the Con-Ops document and while this is sufficient for an understanding of the liberation effort prior in ingress, it is recommended to provide more context. To revisit, the table suggest: "When on the lunar surface, crew will remove dust on the suit prior to ingress to habitat by limiting contamination. Dust mitigation methods will be limited to 15 minutes per crew member. TBD tools will be left external to the vehicle and/or habitat to effectively reduce the amount of dust liberated from the xEMU. In contingency scenarios, partial or no dust mitigation may be performed prior to ingress to habitat." There are a few recommendations to make against this singular Con-Op. First, would be the identification of tools to be used. These tools will have to be present, available, easy to use and not induce a longer than 15-minute cleaning after any EVA. Second, there will need to be success criteria. There appears to be a goal of satisfaction and dissatisfaction inherent in the statement but a recommendation as to the measure of success must be more explicit. Understanding the tools to be used would be a recommended first step before engaging in a discussion of regolith liberation and retention verifications and validations. While the design of the suit should be made to be dust tolerant, only an investigation into the removal of regolith is to be examined. The following are a list of tools that have either been used on the lunar surface (directly or analogously) or on the ISS.

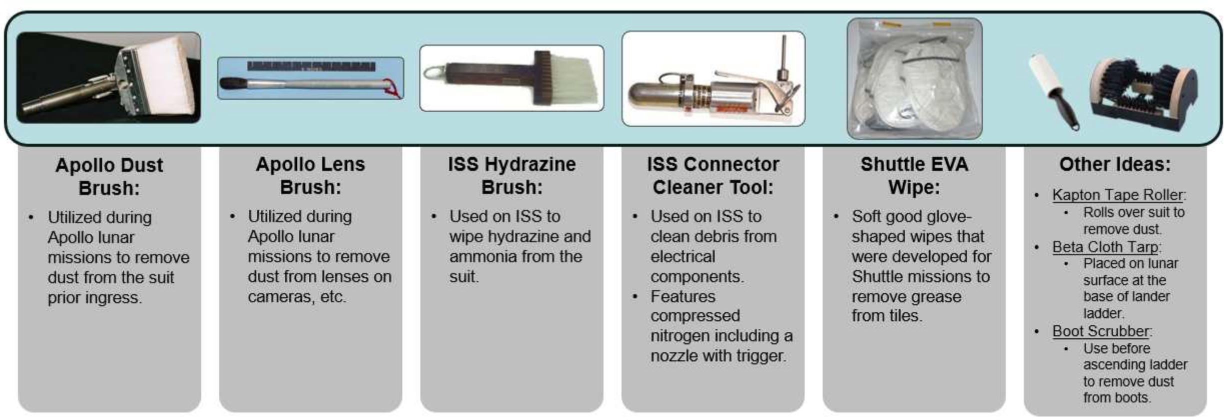


Figure 35 – Potential Lunar Dust Liberation Tools

In this case, especially given with the system context shift from an ISS suit to a lunar suit, there is a potential to start with the verifications and validations that are possible and trace back to system level needs and requirements. The verifications and validations, examining at the end-product system level, could perform a test with a considerable amount of regolith simulant applied generously to the suit in a vacuum chamber, measured beforehand and then removed with a crew member as they would be tested on the lunar surface. Several iterative tests could measure the ability for regolith removal and possibly the amount of lunar dust that could be liberated may be modeled. The suit after regolith removal testing may then be precision cleaned and the regolith that was not liberated could be removed and thus a measurement vs. the pre-condition suit vs. the post-condition suit under testing will allow engineers to understand what the true regolith retention vs. liberation might be in this controlled vacuum environment using the lunar dust simulant. While being able to measure the amount of regolith liberated or retained on the suit will not readily be possible on the lunar surface, this testing will satisfy the notion that a particular amount of regolith may be removed under a certain amount of time, leaving the time under regolith removal to be a quantifiable approach that crew members could use, that can be equitable to dust removal. For example, if indeed the two member crew performing a validation test in a vacuum with regolith simulant under a 15-minute cleaning provided evidence that sufficient regolith had indeed been removed, this 15-minute time could be established as the verification and validation method instead of the 100 gram number that while could be the value of regolith that is minimized upon ingress would not be able to be validated during actual lunar operations while the 15-minute window could be validated. Furthermore, the derivation from this 100-gram basis came from studies showing that during Apollo Program missions, there was an estimate of 227 grams of surface dust per crew member during an EVA which may have entered the habitable environment post-EVA (Agui, et al, 2009). Of the 227-gram load, the fraction $< 10\mu m$, which accounts for 7% of the total dust intrusion load of 15.9 grams per crew member during an EVA, has the potential to be suspended in the cabin environment (Agui, et a, 2011). The requirement was built around the assumption that based on preliminary testing of orthofabric and simulant with dust mitigation technologies and techniques at a 90% effectiveness level, that 78% reduction of dust could be achieved when compared to the Apollo program.

7.3. GSE Testing Team Development Discussion

The GSE Testing Team creates testing stations for various components across the xPLSS to perform both developmental and qualification testing and a case study in SIPPE was described in *Section* 6.3 in terms of its team development, traditional waterfall to Agile development, and an investigation into systems engineering verifications and validations across an individual subproject on xEMU. A focus group was established across 3 iterations to temper the hypothesized model for project lifecycle development.

7.3.1. JETS Processes Influencing GSE Testing Team Development

The two facets studied during the GSE Testing Team development were the JETS processes such as procurements, fabrication and design of GSE equipment and the Agile development and implementation of Agile frameworks to influence verifications and validations of the GSE product.

7.3.1.1. Design Review Process Discussion

The first of these design processes that posed a challenge was the JETS Design Review Process. After the J391 metrics were reviewed and when compared to the large disparities in terms of timely turnarounds by signatories, the GSE Testing Team recommended to JETS management to implement a reduction in process wherein the GSE Testing Team dictates when a SME signature (materials, stress, etc.) is needed. This effort effectively eliminates signatories that could add schedule and, in many cases, may not add value in their SME areas. Instead of the (6) signatories needed, only (2) would be needed per drawing, which would be the design lead and project manager. The form would then be recommended to include all functional groups but use a checkbox method with the design lead the authority in selecting which SMEs are required to sign.

The design lead is then empowered to dictate several aspects that are related to Agile team development. First, the design lead is practicing the concept of Agile ownership. By removing process and accepting accountability, Agile teams independently organize their work and are able to achieve greater results by taking on the risk and accepting ownership (Koning, 2019). Second, the design lead is practicing servant leadership by removing impediments from the team that may hamper productivity and considering when the SME is needed instead of pushing for signatories in the extent of following an established process. This empowerment and stewardship provide the groundwork for mediating processes with the intent of implementing a stronger organizational focus on sustainability and corporate responsibility (Russell, et al, 2002). Third, the entire Development Team is practicing the concept of favoring generalists over specialists. Agile teams prefer to focus on grooming generalists, individuals that can perform a wide variety of tasks, instead of performing one specialized task (i.e., electrical engineering over mechanical engineering). The goal is then to collaborate with the Development Team to create a multidisciplinary team to ensure a high level of expertise, which will encourage cross-training and once again restore empowerment across the Development Team and leads alike (Sohaib, et al, 2010). Fourth, by removing the process of signatories attached to the J391, by allowing the engineering group to quickly acquire, build and test equipment beforehand allows for rapid prototyping, which offers a method to improve communication and stakeholder engagement. Käpyaho, et al support through their agile development case study on user interface prototyping that this approach supports the minimization of documentation (Käpyaho, et al, 2015). This minimized documentation would include the reduced signatories needed to release drawings through the JETS model.

The second of these design processes that posed a challenge were the drawing methods. The two drawing types identified by the design groups: Type I (Mono-Detailed Drawings) and Type II (Simple-Detailed Drawings), with the Type I drawings being a contractual agreement that the customer and performing organization are bound. The argument from JETS is that the NASA proposal of utilizing Type II single, simplified drawings introduce higher risks than multiple, mono-detail drawing Type I drawings that JETS supports. The GSE Testing Team performed a risk assessment in an effort to, if possible, challenge and offer a contractual amendment due to the customer's wish to explore opportunities to expedite schedule based on shifting from Type I to Type II drawings on certain GSE products, including the SIPPE testing station. The first step in the risk assessment was to focus on areas in the Risk Management Plan for xEMU and utilized its approach to analysis analogous to PMI's Identify Risks and Perform Qualitative Analysis with an initial brainstorming session (Hunt, 2018). The brainstorming discussion yielded the following results to support using Type I drawings:

- Single simplified drawing method release complexity and difference in JETS/NASA perception.
- A lack of 3D model and configuration control concerns
- Compatibility risks associated with FAB, Final, and revision approval requirements.
- Compatibility risks associated with procurement/vendors.
- Compatibility risks associated with build phases.

7.3.1.2. Procurement and Fabrication of GSE Hardware Discussion

The areas of interest for process improvement were the procurement and fabrication of GSE hardware with an additional interest to the calibration and cleaning for procured parts. GSE project management and engineering determined that contractual constraints were in place that (JPR 8500.4, JSC-08080-2B, NASA-STD-5005) that dictated the designations and requirements needed to procure GSE parts. The customer with the support from the GSE Testing Team management developed a proposal for the performing organization's upper management to provide an effective approach to both satisfy the risks associated with GSE hardware procurement and fabrication while proving that the additional, current process is unnecessary. The performing organization's approach was that the additional rigor and process guarantees quality and reduces risk. In an analogous manner to the JETS Design Process approach, the GSE Testing Team and the customer collaborated to help distinguish that the process could be expedited while maintaining the same level of quality.

For procurement process improvement, the distinction of two separate articles were established: GSE Critical and GSE Non-Critical. GSE Critical Components would be defined as those that:

- Control a hazard.
- Interface directly with flight hardware.
- Affect the structural integrity significantly enough to compromise flight hardware.

GSE Non-Critical components would be those that do not meet the above specification. This assumption, if agreed upon by the performing organization's upper management and Quality Department would allow for ease in procurements with minimal quality codes and pertinent documentation including CofCs, material certifications, lot traceability, etc. The recommendation by the GSE Testing Team would be for the acceptance of procurements that are critical to be purchased under the GSE-Critical designation with full rigor and pedigree while the non-critical components would be procured as GSE but with the Non-GSE Critical designation that would have less rigor and pedigree but allow for contractual obligations to be maintained as both procured articles would still have the GSE designation. This recommendation was approved by JETS Quality Department. The GSE Testing Team recognized that fabrication of hardware at NASA Johnson Space Center involved more process when compared to their competitor, commercial partners, as those competitors would be building hardware outside of NASA thus allowing for a reduction in documentation and quality process while preserving the intended rigor and pedigree of the form, fit, function, quality and safety measures and with the intention of delivering in a more expedited nature and within a smaller operating budget. Jacobs maintains an offsite facility. To further improve upon the procurement by taking advantage of the reduction in process offsite while still retaining the quality and proper control over risks, the GSE Testing Team developed an additional facet to the GSE procurement process for fabrication offsite.

First, it was recommended that GSE Critical parts be purchased as Class III with limited quality codes to reduce costs and lead times and further reduce GSE Non-Critical parts to be purchased as Class III without any quality codes attached thus removing the obligation for extensive paperwork with regardless to CofCs, traceability, material specifications, etc., with the caveat that when dictated by the design lead, quality codes may be applied. The notion of purchasing as Class III is that these parts will also forgo the Receiving Inspection process, which is intended for parts that mandate higher inspection and quality control.

Second, the fabrication of hardware offsite will be done against released drawings on a form similar to a TPS. The benefit of the Class III designation also allows for the entire test station to be built from these procured parts offsite without the extensive documentation, quality control and with the ability to have the entire assembly inspected when delivered onsite to NASA Johnson Space Center and then upgraded to GSE. Third, to facilitate faster calibration and cleaning turnarounds, it is recommended that an owner collates a spreadsheet for all hardware with equipment-specific calibration cycles and need dates for cleaning. This person would act in a servant leader capacity to work exclusively with the Calibration & Cleaning Department while including a push and pull communication tool on Microsoft OneNote for project engineers to update and notify the servant leader of equipment needed calibration and/or cleaning. It is also recommended that in addition to facilitating expedition of parts to be cleaned and calibrated by the servant leader liaison with the Calibration & Cleaning department, that the liaison confer with the department to waive instruments that may not need to be calibrated as frequently and question when the calibration clock begins (i.e., instrument calibration begins once the hardware is delivered regardless of its use date) with support from the hardware's Original Equipment Manufacturer (OEM). Finally, metrics were compared based on historical data collated from several GSE rigs to compare the new approach to procurement, fabrication, cleaning and calibration proposed metrics. Calibration, cleaning, fabrication and procurement data was derived from the performing organization's procurement, calibration & cleaning databases and fabrication historical data against similar GSE testing stations with an applied three-point Beta estimation method using optimistic, pessimistic and most likely projections, recommended by PMBOK, 6th edition (PMI, 2018). A form of instrumentation that was sent to cleaning and calibration on other projects and is to be utilized by the SIPPE project was characterized against the current and desired project schedule times. A reduced and abridged schedule is shown for comparison. Calendar dates are simply notional and are not indicative of any particular project). The data shows that with these proposed changes, procurement SPI improves to 1.68 and calibration, cleaning, calibration and fabrication improves SPI to 1.28 using the EVM metrics specified by PMBOK, 6th Edition (PMI, 2018).

Task Name	Duration	Start	Finish
Fabrication/Assembly - Onsite Build	**212 days**	**Mon 11/1/21**	**Tue 8/23/22**
Top Level Mechanical Assembly	**165 days**	**Mon 11/1/21**	**Fri 6/17/22**
Mechanical Piping Assembly	**124 days**	**Thu 3/3/22**	**Tue 8/23/22**
TPS #XXXXX (Rough Assembly)	**124 days**	**Thu 3/3/22**	**Tue 8/23/22**
TPS #XXXXX (Cleaning)	**64 days**	**Thu 3/3/22**	**Tue 5/31/22**
TPS #XXXXX (Calibration)	**91 days**	**Thu 3/3/22**	**Thu 7/7/22**
TPS #XXXXX (Final Assembly)	**23 days**	**Fri 7/8/22**	**Tue 8/9/22**
Electrical Wiring Assembly	**193 days**	**Mon 11/1/21**	**Wed 7/27/22**
Electrical Power Distribution Box Assembly	**52 days**	**Thu 3/3/22**	**Fri 5/13/22**
Electrical Data Acquisition Box Assembly	**62 days**	**Thu 3/3/22**	**Fri 5/27/22**
Electrical Safety Box Assembly	**67 days**	**Thu 3/3/22**	**Fri 6/3/22**
Altered Item Assembly	**32 days**	**Thu 3/3/22**	**Fri 4/15/22**
Electrical Harness Assembly	**20 days**	**Thu 3/3/22**	**Wed 3/30/22**

Figure 36 – Fabrication Onsite Build Schedule

Task Name	Duration	Start	Finish
Fabrication/Assembly - Offsite Build w/ C&C Improvements	**165 days**	**Mon 11/1/21**	**Fri 6/17/22**
Top Level Mechanical Assembly	**165 days**	**Mon 11/1/21**	**Fri 6/17/22**
Mechanical Piping Assembly	**64 days**	**Thu 3/3/22**	**Tue 5/31/22**
TPS #XXXXX (Rough Assembly)	**64 days**	**Thu 3/3/22**	**Tue 5/31/22**
TPS #XXXXX (Cleaning)	**24 days**	**Thu 3/3/22**	**Tue 4/5/22**
TPS #XXXXX (Calibration)	**11 days**	**Thu 3/3/22**	**Thu 3/17/22**
TPS #XXXXX (Final Assembly)	**23 days**	**Wed 4/6/22**	**Fri 5/6/22**
Electrical Wiring Assembly	**105 days**	**Mon 11/1/21**	**Mon 3/28/22**
Electrical Power Distribution Box Assembly	**42 days**	**Thu 3/3/22**	**Mon 5/2/22**
Electrical Data Acquisition Box Assembly	**52 days**	**Thu 3/3/22**	**Mon 5/16/22**
Electrical Safety Box Assembly	**67 days**	**Thu 3/3/22**	**Fri 6/3/22**
Altered Item Assembly	**32 days**	**Thu 3/3/22**	**Fri 4/15/22**

Electrical Harness Assembly	**20 days**	**Thu 3/3/22**	**Wed 3/30/22**

Figure 37 - Fabrication Offsite Build Schedule

Task Name	Duration	Start	Finish
Purchase Order - GSE Part	**79 days**	**Tue 3/30/21**	**Fri 7/16/21**
Submit Purchase Request (PR)	5 days	Tue 3/30/21	Mon 4/5/21
Obtain Electrical Parts Approval	5 days	Tue 4/6/21	Mon 4/12/21
Obtain Product Quality Assurance	5 days	Tue 4/13/21	Mon 4/19/21
Create PO/Award (bid process and Award)	20 days	Tue 4/20/21	Mon 5/17/21
Delivery from Vendor	12 weeks	Mon 4/26/21	Fri 7/16/21
Receive Part	3 days	Mon 5/31/21	Wed 6/2/21

Figure 38 – GSE Part Procurement Schedule

Task Name	Duration	Start	Finish
Purchase Order - Class III Equivalent Part	**47 days**	**Tue 3/30/21**	**Wed 6/2/21**
Submit Purchase Request (PR)	5 days	Tue 3/30/21	Mon 4/5/21
Obtain Electrical Parts Approval	5 days	Tue 4/6/21	Mon 4/12/21
Obtain Product Quality Assurance	5 days	Tue 4/13/21	Mon 4/19/21
Create PO/Award (bid process and Award)	4 days	Tue 4/20/21	Fri 4/23/21
Delivery from Vendor	5 weeks	Mon 4/26/21	Fri 5/28/21
Receive Part	3 days	Mon 5/31/21	Wed 6/2/21

Figure 39 – Class III Part Procurement Schedule

7.3.2. Agile Development Proposal and Verifications & Validations Discussion

Examining the SIPPE team more closely during the mechanical and electrical design phases, of the greater challenges was defining the Definition of Done (DoD), a widely held Agile mindset paradigm. A case study was performed with the DoD concept utilized to serve as a vehicle for implementing standards, introducing compliance measures, using training and templates to drive quality improvements and reduce costs. The conclusion, although more proximate, empirically indicates a reduction of defects and technical deb on the project (Davis, 2013). For the GSE Testing Team, SPI and CPI steadily decreased and held at approximately 0.33 for the majority of the intermediary and final design phases (August 2020 through February 2021) due to the fact that design efforts did not complete as anticipated

(July 2020). One of the Agile concepts also applicable is the joint accountability and self-empowered team frameworks by having collective buy-in on project deadlines and milestones (Crowe, 2018) which was not discussed with the entire Development Team prior. To gain team buy-in and continuous engagement, the second project manager of the GSE Testing Team acted in the capacity of a ScrumMaster for the SIPPE team. Three major developments occurred that increased both CPI and SPI as a result of this new implementation. First, electrical engineers had daily standup meetings in a Scrum-style capacity. These 15-minute meetings daily allowed the ScrumMaster to quickly evaluate progress, communicate with pertinent stakeholders and remove any impediments encountered by the team. Second, mechanical engineers were given a drawing tracker that included the entire breadth and depth of engineering drawings on the drawing tree which encompassed the entire scope of the SIPPE work. This method also encouraged team members to better understand schedule considerations and own their work. Kanban sheets were used to approximate a backlog with expectations on what volume of work was needed to be accomplished. Third, an experienced project/mechanical engineer who was dedicated to the GSE Testing Team was promoted to design lead to offset the workload of the current design lead, who had been multi-tasking across several projects on xEMU. Studies indicate that up to 40% of productivity is lost to multi-tasking (Cherry, 2012).

Validations value the techniques of inspections, reviews, walkthroughs and prototyping, with the latter having the finest measure of uncovering the true requirements of a product through trial and demonstration. The best method to engage stakeholders and receive their acceptance and agreement on requirements is iterative progression and a tangible prototype from which to baseline. Midcourse adjustments are the norm as Agile planning is doe throughout the project and more of a less upfront effort (Griffiths, 2012). Prototyping is specific mostly to iterative lifecycles, where improvement of the product is a result through successive iterations involving prototyping technology. When the engineering team has secured an alternative and built a prototype, testing is performed and can satisfy validation efforts on the proposed solution. If the design does not meet the required customer performance, the engineering design process is repeated until a satisfactory solution is provided (Kamrani, et al, 2010). The application of software-intensive, Scrum frameworks may find their place in more hardware-centric project environments.

The question then becomes how can the GSE Testing Team utilize a Scrum and Agile tailored framework in their hardware builds that gains the benefits of those found in software-intensive systems but also avoids the limitations that Scrum or Agile may inherently embody in the hardware-intensive system? The mythos of Agile-based developments is that they are frameworks and only as useful as they may be tailored to suit the performing organization's needs. The development team may need to learn, be trained and tailor the Agile practices to deliver value on a regular basis. PMI recommends tailoring options based on a project factor vs. tailoring option method. Table 41 displays areas that the GSE Testing Team can tailor their approaches, with a specific focus on rate of process improvements required by the level of team experience, the quality of the product increment, flow of work and impediments, multiple teams needing to build a product and the project team being inexperienced in the way of Agile approaches (Alliance, 2017).

Table 41 – Tailoring Options

Project Factor	Tailoring Option
Steady or sporadic demand pattern	Using a cadence helps with the demonstration, retrospective and uncovering new work
Process improvement vs. team experience level	Retrospect more frequently to uncover improvements
Workflow interrupted by delays or impediments	Making work visible by using Kanban boards, experimenting with work in process limits for workflow improvement
Quality of product increment is considered poor	Considering test-driven development practices
Multiple teams needed to build product increment	Attempt to scale one to many agile teams to understand which approach is optimal
Project team members are not familiar with agile	Training team members in fundamentals of agile or related methodologies

In a study conducted from 2006 to 2015, 21 case studies were analyzed regarding the implementation of Agile based methods on non-software-intensive systems. While these studies varied between manufacturing, design and electronics, the reported conclusion indicated that increases in transparency, flexibility, quality, collaboration, motivation, speed, increased knowledge sharing, improved focus, impediment removal, clear sense of progress (i.e., Definition of Done) and improved resource allocation. The quantifiable methods included high-touch, low-tech tools such as Kanban sheets, Scrum principles including Daily Scrums and sprints and more general iterative and incremental deliveries. The reported challenges included a change in mindset in the organization, buy-in from mangers, long-term planning and scope creep (Gustavsson, 2016). With the compiled, external case study results, the anticipated work packages and internal case study of the GSE Testing Team SIPPE Team utilizing Agile principles to improve EVMs, the following areas of Waterfall vs. Agile are juxtaposed:

Table 42 – Waterfall vs. Agile Areas of SIPPE Testing Team

Waterfall Areas Performed	Agile Areas to Exploit
Initiating and Planning (PMI Process Group)	Sprint Planning
Executing (PMI Process Group)	Sprint
Monitoring & Controlling (PMI Process Group)	Daily Scrums, Removing Impediments
Closing (PMI Process Group)	Final Product Increment & System Test
Project Schedule & Budget	Burndown Chart (with EVMs)
Project Scope	Product Backlog, Sprint Backlog
Short Daily Meetings	Daily Standup
Customer Review Points	Sprint Demo and Review & Product Increment
Greater GSE Testing Team Meetings	Sprint Retrospective & Scrum of Scrums
Deliver Milestones/Deliverables	Deliver Incremental Value
Team Members Largely I-Shaped	Cross-Training for M-Shaped Individuals
Managers and Leads	ScrumMaster & Product Owner

8. CHAPTER 8: ITERATION TESTING

Lede: This chapter delineates the iterative testing that went into building both prototypes for both project lifecycle development and requirements engineering. The approach included a candidate case selection process, case selection process for iterative testing, and the iterative development of both prototypes. The first 3 tempering of the model were done by engineering comparative analysis with a before and after section and the conclusion of testing to support an approximation of a hypothesized, optimized model. Any testing after the 3rd iteration was taken above and beyond the scope of the project to external entities for further model vetting. Further vetting includes elicitation from INCOSE members, including those who authored the work from which the research is cited, Likert scale polling, and short questionnaires. Acceptance of the model past the 3rd iteration included two-factor success: a comparative analysis of post FMEA risk numbers and Likert scale customer satisfaction.

8.1. Iteration Background

The goal before iteration testing is to determine candidate case studies and select specific case studies that would undergo model tempering to solve a specific challenge or set of challenges on the xEMU project. To select potential candidate studies, preliminary fact-finding efforts, criterial selection of potential candidate studies, and a list of candidate studies were developed.

8.2. Candidate Case Study Selection Criteria

Future case studies for development would be further iterated and tempered only if they met all of the following criteria:

i. The candidate case study presents a challenge on the project and is under the systems engineering field of study.
ii. The candidate case study has qualifiable data that can be assessed to determine potential root causes.
iii. The candidate case study has quantifiable data that can be assessed to determine potential root causes.
iv. The candidate case study has qualifiable data that can be assessed against a hypothesis.
v. The candidate case study has quantifiable data that can be assessed against a hypothesis.
vi. The candidate case study has qualifiable data that can be utilized to be tempered in an iterative model to satisfy an approach to solving a challenge on the project.
vii. The candidate case study has quantifiable data that can be utilized to be tempered in an iterative model to satisfy an approach to solving a challenge on the project.
viii. The candidate case study, within itself and juxtaposed to lateral candidate case studies, has a scope that can be illustrated, investigated and results analyzed within a dissertation boundary.

8.3. Candidate Case Study Selection

This iteration consisted of an investigation lead by questionnaires, interviews and focus groups. Stakeholders from both the performing organization (i.e., JETS, NASA contractor) and the customer (i.e., NASA civil servants) had inputs elicited. The following candidate case studies were evaluated.

8.3.1. Verification and Validation

Across the project, several xEMU members advised that verification and validations while by in large were adequate, at times were not well defined. The notion verification and validation in some cases were used interchangeable led to the impression that perhaps the differentiation and application may be optimized.

8.3.2. Configuration Management

During the initial brainstorming and interview sessions, several systems engineers indicated that configuration management of requirements could be an avenue of investigation. The typical process allows for a streamlined and organized practice for establishing and maintaining a consistent and updated requirements engineering database. One of the central concerns focused on challenging the current in place methods as well as a potential investigation into possible misuse of the system.

8.3.3. Purchasing of Hardware

The method by which the GSE Testing Team purchases hardware is focused on a strict control process to ensure that what is purchased meets a certain standard of quality. While ensuring quality is key to delivering a safe and high-pedigree product, many of the development team members indicated that the process may be optimized.

8.3.4. Requirements Building Tools and Usage

The systems engineering team uses Cradle® as their requirements building tool. Several of the systems engineers indicated challenges when using and implementing the architecture. This tool allows for requirements building both in terms of organization of high level to low level requirements and a systematic method to creating requirements documents.

8.3.5. JETS Design Review Process

The systems engineering team uses Cradle® as their requirements building tool. Several of the systems engineers indicated challenges when using and implementing the architecture. This tool allows for requirements building both in terms of organization of high level to low level requirements and a systematic method to creating requirements documents.

8.3.6. Mass of xPLSS Backplate Development

The backplate houses the majority of xPLSS related components, specifically those avionics, electrical and mechanical components that allow for successful operation of the three primary life support systems in the ventilation, oxygen and thermal loops. One of the systems engineers indicated that the backplate did not have an End Item Specification and traced to a higher-level specification document which is to illustrate that the backplate was not part of the architecture and in a sense not fully decomposed. One of the challenges was the mass tracking.

8.3.7. Electrical Controller Development

A failure in an electrical controller during testing that maintained the functions of the ventilation loop, in specific the CO2 scrubber, brought to light potential inconsistencies with best practices, specifically some of those that related to systems engineering, be it configuration management and proper drawing release processes.

8.3.8. Project Life Cycle Development

The xEMU project uses a traditional waterfall method to organize project lifecycle development. While this method has a proven record, especially for predictable work, the manner in which work is performed has changed and the technological landscape continues to advance. In particular, the GSE Testing Team schedule and budget performance faced challenges with

8.3.9. Lunar Dust Mitigation Requirements

One of the hallmarks of the new space suit is the ability to sustain long during EVAs on the lunar surface. While a space suit suitable for the moon would not be a first, challenges exist in the systems engineering development of requirements building as many of the well-established practices were not established when the first EMUs operated on the lunar surface. In addition to augmenting and pairing former and current technologies to existing practices were the discernments and challenges regarding the definition of lunar dust mitigation and the intended requirements derived from the needs versus the anticipated needs.

8.3.10. Auxiliary Lighting Requirements

In the even the primary suit light fails, an auxiliary lighting system is expected to deliver adequate light to illuminate the crew members work environment and visible line of sight. While the primary lighting requirements existed, a challenge was present with regards to auxiliary lighting and its development.

8.4. Case Study Selection & Model Tempering

Following a compatibility matrix, these are the following criteria that allow for a robust case study analysis. It is important to note that while there are proximate causes in some instances, there was either not enough evidence, time or capability to assess the candidate case study robusticity and as such, either proximate causes or candidate case studies that could not stand alone but were found to be complimentary to stand alone case studies were integrated to allow for analysis.

Table 43 – Candidate Case Study and Key

Candidate Case Study	Criteria							
	i	ii	iii	iv	v	vi	vii	viii
Verifications & Validations	X	X	X	X	X	X	X	X
Configuration Management	X	X						
Purchasing of Hardware	X	X	X	X	X	X	X	X
Requirements Building Tools & Usage	X	X						
JETS Design Review Process	X	X	X	X	X	X	X	X
Mass of xPLSS Backplate	X	X	X	X				
Electrical Controller Development	X	X		X				
Project Lifecycle Development	X	X	X	X	X	X	X	X
Lunar Dust Mitigation Requirements	X	X	X	X	X	X	X	X
Auxiliary Lighting Requirements	X	X	X	X	X	X	X	X

Key
Cannot Support or Stand Alone as a Case Study
In Support of a Stand-Alone Case Study
Stand-Alone Case Study

During the candidate selection process, it was evident that while many of these candidate case studies were atomic, their stretch exceeded into other domains (i.e., JETS Design Review Process, Purchasing of Hardware and Verification and Validation were symptoms of the challenge of Project Life Cycle Development). The following or the selected case studies.

- Project Lifecycle Development
    - Implement additional areas of Purchasing of Hardware and JETS Design Review Process
- Lunar Dust Mitigation
    - Implement additional areas of Verifications and Validations and areas of Requirements Building
- Auxiliar Lighting Requirements
    - Implement additional areas of Verifications and Validations and areas of Requirements Building

Each of the aforementioned case studies will constitute the central case studies investigated in this dissertation. Not apparent at the onset but two of the three are in essence investigating corresponding challenges in requirements building. As such, each would constitute a refinement to the requirements building approach from differing directions. For the project lifecycle engineering and for maximization of time and resources, the SIPPE GSE Testing Team will have a more formal decomposition of cost and schedule performance and team development whereas complimentary GSE Testing Teams will only have their schedule and cost performance evaluated to vet the assumption that the SIPPE GSE Testing Team cost and schedule may have been optimized per the current hypothesis. The manner in which the study will commence will include a methodical approach wherein a model is created, tempered and finalized either by guaranteeing an approach based on full screening or by exhaustion of time and resources with limitations explicitly documented. Each iteration will present a series of improvements and limitations in a tabular format. The goal with each iteration is to begin with a coarse-grained direction with fine graining throughout the tempering with major direction listed, time period of study, data gathering techniques and the aforementioned improvement and limitations. While each iteration is unique, a series of methodical steps is listed yet altered to suit the needs of the iteration. The following template is used to reflect both new and cascading information into tempered model iterations:

- Iteration #: List (Cascading) Major Implementations/Amendments to Model
- Time Period: List Duration of Iteration
- Data Gathering: List Data Gathering Techniques
- Improvements: List (Cascading) Improvements During Iteration
- Limitations: List (Cascading) Limitations (and Eliminated Limitations)

8.5. ITERATION 1A: Project Lifecycle Development

8.5.1. Iteration Development

The period of performance of Iteration 1 focused between February 2021 and February 2022. The data gathering techniques included:

- Initial Literature Review: Where Scrum and other agile concepts from academic and peer reviewed journals are viewed.
- Case Study Review: Where the specific case study of SIPPE is evaluated in terms of cost and schedule performance against current waterfall breakdown.
- Earned Value Analysis: The CPI and SPI are numerically represented as performance measures against product delivery.
- Brainstorming: Various, semi-formal sessions were elicited with stakeholders from across the project to assess which areas of systems engineering presented challenges and for project lifecycle development, those specific challenges documented.
- Interviews: Various, semi-formal sessions were elicited with stakeholders from across the project to assess which areas of systems engineering presented challenges and for project lifecycle development, those specific challenges documented.
- Focus Groups: Various, semi-formal sessions were elicited with stakeholders from across the project to assess which areas of systems engineering presented challenges and for project lifecycle development, those specific challenges documented.
- Questionnaires: Various, semi-formal sessions were elicited with stakeholders

from across the project to assess which areas of systems engineering presented challenges and for project lifecycle development, those specific challenges documented.

- 1st Survey: A Likert scale survey to assess the current vs. desired state of agile offerings on the project with responses from the Development Team and customer.

The goal of the iteration is to develop Scrum and eliminate the waterfall development. This will be done by building the product backlog, developing the sprints, developing the Scrum schedule and re-identifying the Development Team from a waterfall to a Scrum-style team. Full Scrum and sprints were added to the entire project lifecycle.

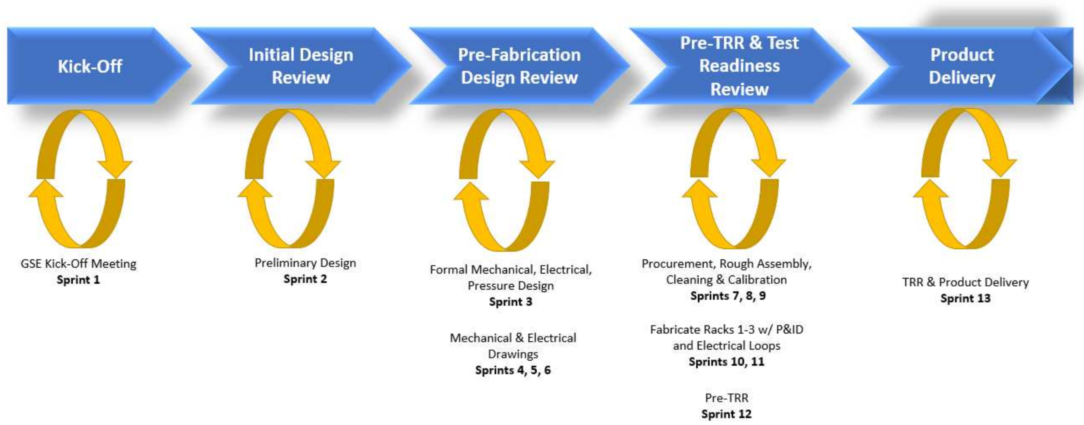


Figure 40 – Iteration 1A Lifecycle Approach

A survey with 24 individuals across the project (customer and performing organization) were polled on the Likert scale to elicit information regarding waterfall sentiment on the project; 1 – Strongly Disagree, 5 – Strongly Agree; any score above a 3.5 (median between "Neither

Agree or Disagree and Somewhat Agree) is deemed a positive indication on position; any score below a 3 is considered a negative indication on position. The survey below was used to compare the current vs. potential new project lifecycle development to examine how malleable the team would be to a change in lifecycle development. Results indicate that the current waterfall development has not been successful on the project.

Table 44 - Current vs. Proposed Lifecycle General Likert Questions

#	Current Project Lifecycle Development General Questions	Mean
1	The traditional life cycle approach on the xEMU project is the preferred approach to team development	3.43
2	The traditional life cycle approach on the xEMU project is the preferred approach to product delivery	3.26
3	My team is not adaptable to change from one project lifecycle approach to another	2.42
4	I would not consider changing the traditional lifecycle approach to xEMU product development	2.33
5	The current lifecycle approach with associated xEMU projects have positively impacted schedule development	2.45
6	The current lifecycle approach with associated xEMU projects have positively impacted scope development	2.64
7	The current lifecycle approach with associated xEMU projects have positively impacted budget development	2
8	I would prefer to operate strictly in the traditional waterfall project lifecycle method by which xEMU operates	2.13
9	As a customer, I feel the contractor kept me well engaged using current traditional waterfall lifecycle methods	3.69
10	As a contractor, I feel we kept the customer well engaged with using traditional waterfall lifecycle methods	4.13
11	A shared vision across the project was developed with current traditional waterfall lifecycle methods	3.86
12	A good definition of done was harnessed using the traditional waterfall lifecycle method	3.59
13	Processes are easily tailorable on the project	2.29
14	Products from lessons learned are successfully implemented	3.5
15	Continuous improvement methods are implemented	3.17
16	Team assessments are done periodically	3.79

A survey with 24 individuals across the project (customer and performing organization) were polled on the Likert scale to elicit information regarding a potential shift to agile, with a combination of the Agile Manifesto (Beck et al, 2001) and the 12 Principles of Agile (Crowe, 2018). 1 – Strongly Disagree, 5 – Strongly Agree; any score above a 3.5 (median between "Neither Agree or Disagree and Somewhat Agree) is deemed a positive indication on position; any score below a 3 is considered a negative indication on position. Results indicate that the team is malleable and requesting a shift to agile.

Table 45 – Agile Development General Likert Questions

#	Agile Development General Questions	Mean
1	I prefer responding to changes over following a plan	3.5
2	I prefer customer collaboration over contract negotiation	4.17
3	I prefer working hardware/software over comprehensive documentation	4.42
4	I prefer individuals & interactions over processes & tools	3.83
5	Delivering products early and continuously to the customer is important	4.13
6	I welcome changes to requirements even late in product development	2.79
7	Products should be built around motivated individuals	3.96
8	The most effective way to communicate is face-to-face	3.46
9	Working hardware/software is the primary measure of progress	3.5
10	It is important to maintain a sustainable working pace	4.7
11	Simplicity is the preferred approach to product development	4
12	Self organizing teams generate the most value	3.74
13	Agile method utilization is the preferred approach to project lifecycle development	3.53

A survey with 24 individuals across the project (customer and performing organization) were polled on the Likert scale to elicit information regarding Scrum and the desired vs. current delivery method that project had offered Scrum-related artifacts; 1 – Strongly Disagree, 5 – Strongly Agree; any score above a 3.5 (median between "Neither Agree or Disagree and Somewhat Agree) is deemed a positive indication on position; any score below a 3 is considered a negative indication on position. The survey below was used to compare the desired vs. current new project lifecycle development to examine how malleable the team would be to a change in lifecycle development. Results indicate that the current waterfall development has not offered a similar Scrum artifact and that the team would prefer to operate in Scrum in some capacity. Certain areas will either be exploited or avoided depending on the delta between desired and current project delivery methods.

Table 46 – Scrum Current vs. Proposed Likert Questions

#	Field	Mean	Count
	3 - Scrum General Questions		
1	I prefer working in teams over working alone	4	24
2	I feel re-estimates on schedule are frequently needed to course corrected once the execution process commences	4.26	23
3	I feel complex products will have requirements that have a higher likelihood to change	4.42	24
4	I prefer to work closely with the customer several times a week	3.46	24
5	I believe the customer should be well engaged with project and product development	4.25	24
6	I feel that once lessons learned are presented, they are used adequately to improve the project	3.54	24
7	I believe that priorities must be re-evaluated continuously	4.33	24
8	I prefer to develop a generalized skillset over developing a specialized skillset	3.92	24
9	I like finding different ways to help optimize task efficiency	4.54	24
10	I prefer teams that are self directing vs. directed by a traditional manager	4	24
11	I prefer working in an open team space (i.e., no offices, no barriers)	3.08	24
12	I am motivated by trust over fear	4.77	22
13	I believe conflict is not to be avoided	3.92	24
14	I believe in team success over individual success	4.46	24
15	I am not afraid to make mistakes at work	4	24
16	I believe the leadership style should not remain constant during team development	3.41	22
17	Iterative demonstrations with the customer reveal actual requirements	4.09	22
18	Timeboxing (i.e., strict time constraints) allows for results-based plans	3.38	24
19	I would like to contribute to project planning	4.38	24
20	I believe planning should include evaluations against a prototype	4.42	24
21	I prefer 15 minute daily tag up meetings over weekly status meetings	3.04	24
22	I prefer the idea of releasing functional sub-products periodically over releasing the sum of all sub-products at the end of the project	4.54	24
23	I believe the actual work needed on the project is discovered by performing the work over pre-determined planning	4.17	23
	4 - Scrum Current Project Position		
#	**Field**	**Mean**	**Count**
1	We work in teams more than working alone	3.54	24
2	Our schedule does not need to be frequently course correct once the execution process commences	1.74	23
3	Our products do not have requirements that have a higher likelihood to change	2.67	24
4	We work closely with the customer several times a week	3.35	23
5	The customer is well engaged with project and product development	3.41	22
6	Lessons learned that are presented make an impact to improve the project	3.7	23
7	Priorities on the project are re-evaluated continuously	3.63	24
8	Functional managers develop a generalized skillset over developing a specialized skillset	3.7	23
9	We find different ways to help optimize task efficiency	3.83	24
10	Our teams are self directing vs. directed by a traditional manager	3.58	24
11	We work in an open team space (i.e., no offices, no barriers)	3.38	24
12	The project motivates by trust over fear	4.08	24
13	The project believes conflict is not to be avoided	3.57	23
14	The project believes in team success over individual success	4.25	24
15	The project fosters an environment where we should not be afraid to make mistakes at work	3.42	24
16	The project expresses that the leadership style should not remain constant during team development	3	21
17	We exercise iterative demonstrations with the customer to reveal actual requirements	3.24	21
18	We practice timeboxing (i.e., strict time constraints) for results-based plans	3.21	24
19	The development team currently contributes to project planning	3.68	22
20	In terms of project planning, we include evaluations against a prototype	3.09	23
21	On the project, we utilize 15 minute daily tag up meetings instead of weekly status meetings	1.96	24
22	On the project, we release functional sub-products periodically instead of releasing the sum of all sub-products at the end of the project	2.96	23
23	On the project, the actual work reflects the planned work	2.96	24

#	Desired Delivery Method	Current Delivery Method	Delta	Exploit Approach	Avoid Approach
1	4	3.54	0.46		
2	4.26	1.74	2.52	X	
3	4.42	2.67	1.75	X	
4	3.46	3.35	0.11		X
5	4.25	3.41	0.84		
6	3.54	3.7	-0.16		X
7	4.33	3.63	0.7	X	
8	3.92	3.7	0.22		
9	4.54	3.83	0.71		
10	4	3.58	0.42		
11	3.08	3.38	-0.3		X
12	4.77	4.08	0.69	X	
13	3.92	3.57	0.35		
14	4.46	4.25	0.21		
15	4	3.42	0.58		X
16	3.41	3	0.41		
17	4.09	3.24	0.85		
18	3.38	3.21	0.17		
19	4.38	3.68	0.7		
20	4.42	3.09	1.33	X	
21	3.04	1.96	1.08		
22	4.54	2.96	1.58	X	X
23	4.17	2.96	1.21	X	

A questionnaire with 24 individuals across the project (customer and performing organization) was given to allow for opened-ended feedback. The question and abridged answers suggested agile and, in some cases, Lean approaches. Responses are as follows:

- *Q1: How would you change any aspects of the current lifecycle development structure of projects on xEMU?*
- *A1:*
    - *"Less reliance on central authority for decisions*
    - *"...lean towards processes that work best..."*
    - *"...development continuously slip on the schedule due to scope creep, underestimating the amount of project tasks, and underestimating the amount of time to complete project tasks...a shift to a more agile project management approach would be substantially more beneficial."*
    - *"...tailor existing processes."*
    - *"...to know their tasks and never be sitting idle. Creating a backlog of tasks and milestones..."*
    - *"I would set smaller iterative deliverables to share with the customer on a more frequent basis..."*
    - *"Make the schedules more high-level deliverables and then use visual chunks."*
    - *"I would change the lifecycle according to progress. As the project matures, the lifecycle can change to be most beneficial."*
    - *"Adapt a prototype phase to test and evaluate work. "*

- *“Less delays of work packages like needing 9 signatures…”*
- *“The current processes & skills training are a problem…employee retention…”*
- *“I would like to combine Scrum and Extreme Programming efforts into our framework while retaining waterfall methods that are applicable.”*
- *“We should probably try using the Agile approach.”*
- *“I would probably choose Scrum because it prioritizes getting work done more quickly. If there is one thing that the customer would want from us, it would be to see deliverables faster and even ahead of schedule.”*
- *“Hybrid scrum”*

The majority of Scrum concepts were lifted directly from best practices and slightly tailored. Following an identification of the project in terms of team, cost, scope and schedule challenges, a reappraisal details the modifications needed to bring the team to Scrum-capable. Alteration of the current, waterfall framework demands first a reclassification of roles and responsibilities of all members. Scrum promotes a ScrumMaster to promote the agile method, a Product Owner to develop scope in the object of a product backlog, a Development Team that will manage the sprint backlog that is comprised of generalists and not so many specialists.

Step 1: Identify Current Team

First, the team is shown in its current state with the Project Managers at the apex, leads secondary and development team tertiary.

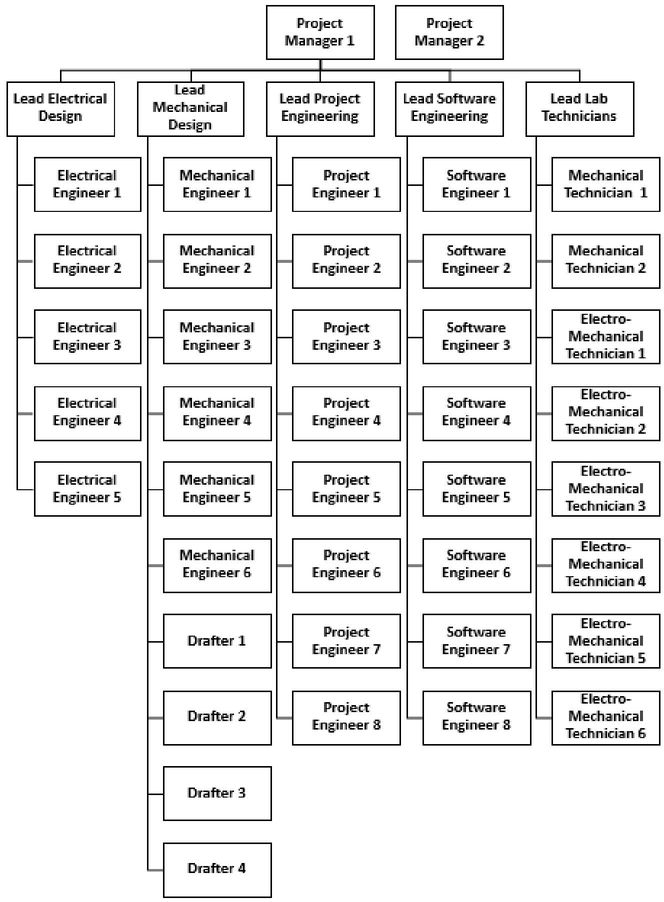


Figure 41 – Current GSE Testing Team

Step 2: Transform Current Team to Scrum Team

A slight modification to the intended Scrum model is made by preserving the current leads. They will act in a similar fashion by providing domain level expertise with the intention of cross training the entire team regardless of principle. The team will be divided into five development teams with the intention of providing each team to each of the particular testing stations needed by the NASA customer.

Table 47 – Analogous Scrum Positions in GSE Testing Team

Traditional Waterfall Framework	**Scrum Framework**

Owner of Specific xPLSS Component	Product Owner
Project Manager 1	N/A (Remains Project Manager)
Project Manager 2	ScrumMaster
All Engineers	Development Team
Electrical Engineers	Electro/Software Generalists
Software Engineers	Electro/Software Generalists
Project Engineers	Mechanical Generalist
Mechanical Engineers	Mechanical Generalist
Electro-Mechanical Engineers	Test Generalist
Mechanical Technician	Test Generalist
Drafter	Drafter Generalist
All Leads	Leads (Support Development Teams)

Although not entirely possible to classify the entire Development Team of specialists into generalists due to the variety of the multi-disciplinary team needed perform testing, there is sufficient, relatable disciplines to collate certain team members in generalist groups. The groupings are pragmatic as the roles and responsibilities for grouped roles have sufficient crossover and knowledge of the mirrored group is a transition that can be conducted with on-the-job training and vendor related training to bring team members to the desired skillset. This proposal is based on the fact that 75% of the Development Team have 5 or less years of experience and are assumed to be more malleable than their senior counterparts. The following table will allow for a quick reference of roles and responsibilities and their analogous counterpart in both frameworks.

The team hierarchy structure is then re-identified as a result of both the reclassification effort from Waterfall to Scrum but keeping as faithful to the current team structure and doing a best fit for all team members. Additional team members will be needed also to account for the backlog of work. It is important to note that not all project team members will encounter as easy a transition due to the level of complexity; this will be a ScrumMaster's responsibility to remove impediments to this process. Separate Development Teams are assigned 1 through 5 and will be applied to each level of priority, 1 through 5. One of these proposed teams could be the current SIPPE team presented in the case study. As all promised date items are now missed, NASA considers all items top priority. The Product Owner will then have a two-fold product backlog: the backlog of test stations and order needed to be performed by the Development Teams but also the product backlog within each of those test stations. Each Development Team will work on one test station at a time, only focusing on product backlog items in that station. After a reappraisal of the remaining work left for all test stations and the number of test stations needed with the number of Development Teams available, meeting the Moon 2024 initiative will still be possible. The notable changes other than the nomenclature for the project organizational hierarchy chart below is that all leads will serve as support during the Scrum ceremonies and are available on a SME expert level. Leads no longer serve as schedule directors as the Development Team will be self-led and prioritize the sprint backlog as a function of the product backlog. The ScrumMaster also exists on the same hierarchy as the project management and leads but acts as a servant leader, removing impediments and ensuring the proper application of Scrum for all stakeholders.

As per the contractual agreement, project management will still remain and exist as the conduit to the commercial side in terms of schedule and budget. Development Teams and the Product Owner will develop scope continuously. The Product Owner will also be horizontal to the ScrumMaster, Project Manager and Leads. Although this individual will be outside of the performing organization as a NASA employee, the Scrum team will treat this individual as part of the Scrum team to manage expectations and keep the customer involved. Although there is only one Product Owner, it is permissible for multiple Product Owners to exist for each one of the Development Teams but only one team to avoid conflicts of interest.

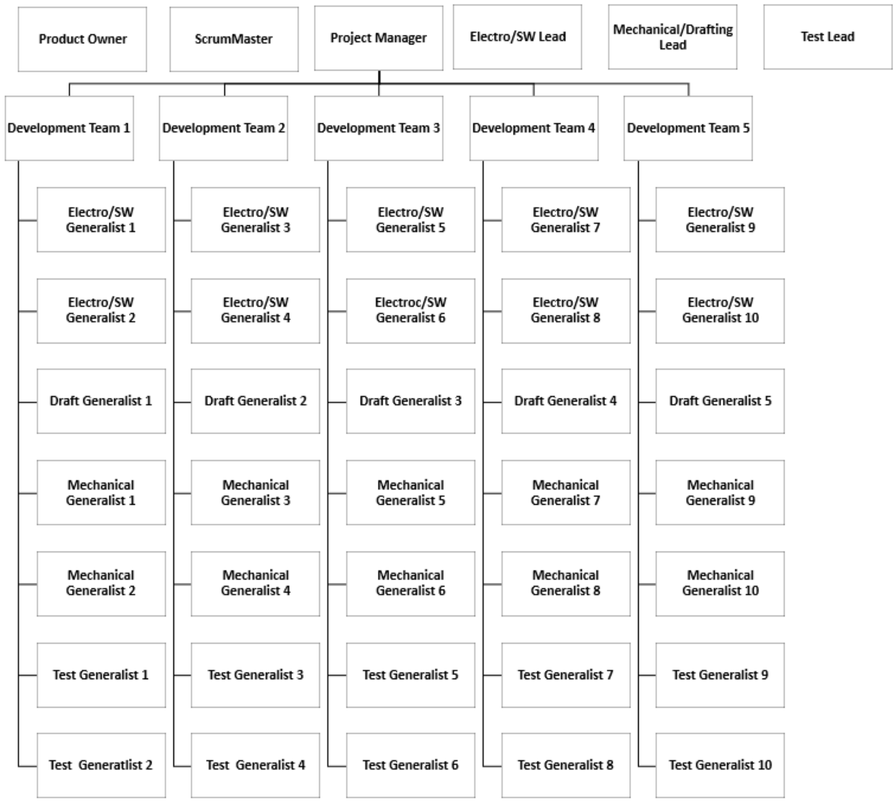


Figure 42 – Proposed Scrum-Style GSE Team

Step 3: Identify Traditional Lifecycle Approach

The GSE Testing Team followed the traditional waterfall method typically instituted within the company. The bullets, descriptions and table to follow discuss the Customer Review Point phases which are detailed in the contract. Phases 1 through 6 indicate all of the instances where the customer will engage with the performing organization. Each point defines a description of what is to be reviewed and a percentage listed. For context, these four Phases generally take 1 to 2 years to complete depending on the complexity of the testing station, given there are no scheduling constraints. In essence, this means a customer will interact 4 times within those 1 to 2 years with the project team as these would cover more of the validation process while intermittent requirements reviewing, design, fabrication and testing by the performing organization constitute the verification process. During these review points, the customer will see a presentation from a project engineer that explain how the test station's progression status. The complete engineering team is there to offer support to the project engineer and the project manager will speak on the entire schedule, discuss budget and identify areas of clarification from the customer. Outside of this interaction, test readiness reviews and beyond are typically not under control of the test station team. The test station team will provide feedback and therefore anything after Phase 6 is considered on-going support from another team, with the project team considered matrix at that point.

- GSE Kickoff Meeting – JETS shall hold a kickoff meeting with the NASA customer to outline scope, conceptual design, schedule, and cost. Detailed test station development will initiate upon closure of this kickoff meeting. Meeting closure is indicated by final approval of this CF contract.

- Initial Design Review – JETS shall hold a meeting with the NASA customer to review the initial test station design. JETS shall have mechanical and electrical schematics complete for review with all instrumentation fully specified. JETS shall review how the functional schematics meet the test station requirements.

- Pre-Fabrication Design Review – JETS shall hold a review with the NASA customer once the design is ready for fabrication. JETS shall show completion of action items and incorporation of design changes assigned at the initial design review. JETS shall present all critical interfaces (i.e., interfaces with flight hardware) and all hazard controls in intimate detail.

- Pre-Test Readiness Review (Pre-TRR) – JETS shall hold a review with the NASA customer once the test fixture assembly is completed and ready for a branch TRR. The TRR board may assign actions; any such actions must be closed by the JETS test fixture Development Team. Closure of all TRR action items and approval by the board will indicate final delivery of the test fixture from the Development Team to the PLSS testing team.

- Test Readiness Review (TRR) – JETS shall present test readiness of the to the NASA customer and NASA Branch Chief or designee. The TRR board may assign actions; any such actions must be closed by the JETS test fixture Development Team. Closure of all TRR action items and approval by the board will indicate final delivery of the test fixture from the Development Team to the PLSS testing team.

- Test Station Delivery (TSD) - JETS shall present the test rig following a successful functional checkout and acceptance data package to the customer. Unless otherwise stated within the contract, JETS will consider the rig delivered and move on to project closure.

Table 48 – Waterfall Phased GSE Test Station Delivery Milestones

Phase	Description	% Project Life
1	GSE Kick-Off Meeting	5%
2	Preliminary Design Review	10%
3	Prefabrication Design Review	50%
4	Pre-Test Readiness Review	80%
5	Test Readiness Review	90%
6	Test Station Product Delivery	100%

Step 4: Transform Schedule from Traditional into Scrum Lifecycle

The current schedule was decomposed in a way that it had not been previously during the GSE test station's initial concept. This was an effort made by all Development Team members, the Project Manager and any relevant stakeholders. The first step was to build the product backlog. The backlog was built in traditional Scrum fashion, with a stakeholder elicitation meeting. The full product backlog was a collection in excess of 500 unique work packages.

Table 49 – Full GSE Test Station Product Backlog

Full Product Backlog	
Task Name	
Initial Planning Items	Discipline
Meet with Component Owner and Review Rig Requirements	EE/ME/SW
Meet with Resource Managers & Receive ROMs	PM
Create Rig Cost Estimate	PM
Create Completion Form PD	PM
Create Rig Schedule	PM
Define Rig Scope	PM
Create Mechanical P&ID	ME
Create Electrical Block Diagram	EE
Create Rough CAD Model	ME
Create PowerPoint presentation	PE
Define Initial Requirements	SW
Re-Configure Existing Legacy Modules from Repository	SW
Buy Short Lead COTS Components	PE
Buy Long Lead COTS Components	PE
Buy Computer Equipment	PE
Buy IT Equipment	PE
Design Calculations	Discipline
Identify Analysis Tasks	PE
Complete Instrument Uncertainty Analysis	PE
Perform Preliminary RV Calculations	PE
Perform Thermal Shroud Calculations	PE
Perform Sub-Ambient Calculations	PE
Estimate CG and Floor Loading	PE
Perform Flow Calcs (Initial)	PE
Perform Vacuum Quality Calcs (Initial)	PE
Perform Pressure Calcs (Initial)	PE
Matrix Groups Assessments	Discipline
Define Structural Stress Tasks	ME

Define Thermal Tasks	ME
Assess Approved Materials	ME
Product Owner Needs	Discipline
Identify Operational Modes	EE
Review Verification Requirements	EE
Define Anticipated Prototypes	EE
CAD Model	Discipline
Account for Plumbing Needs	ME
Verify Number of Racks	ME
Model Orientation Jig & Thermal Shroud	ME
Define Box Dimensions and Items for EE Preliminary Design	ME
Define IT Needs & Meet with Software Team	ME
Complete Preliminary CAD Model	ME
Hold Preliminary Design Review with GSE Lead	ME
Gain Approval from GSE Lead	ME
Mechanical Engineering Documents	Discipline
Complete Preliminary Mechanical P&ID	ME
Complete Preliminary BOM	PE
Prepare Preliminary Drawing Tree	ME
Power Overview	Discipline
Meet with PE to determine Power components	EE
Identify Power Box Lights/Switches	EE
Create Powered Equipment List	EE
Finalize Peer Review Powered Equipment List	EE
Finalize Block Diagram	EE
Rack 1	Discipline
Complete Rack Structure Drawing 1	ME/EE
Complete Rack Assembly Drawing 2	ME/EE
Complete Rack Altered Item Drawing 3	ME/EE
Electrical Harnesses	Discipline
Complete Electrical Interconnect Diagram (Complex) 4B	EE
Complete Harness Instruments to Data	EE
Complete Harness Vacuum to Instruments	EE
Complete Harness Safety to Data	EE
Complete Harness Vacuum to Data	EE
Complete Printed Circuit Board for 450 Interface 4D	EE
Software User Interface	Discipline
Create Basic UI and Present to PE	SW
Implement UI Functionality	SW
Review UI Functionality with PE	SW
Integrate All Modules and Test with Simulated/Benchtop HW	SW
Software Logic	Discipline
Logic Flow Charts	SW
UI Screenshots	SW
Startup, Shutdown, Maintenance	SW
Generate SW Configuration Document	SW

Rack 2	Discipline
Complete Rack Structure Drawing 1 (Standard)	ME/EE
Complete Rack Structure Drawing 2 (Standard)	ME/EE
Complete Rack Structure Drawing 3 (Simple)	ME/EE
Complete Altered Item Drawing 4 (Simple)	ME/EE
Complete Rack Sub Assembly Drawing 5 (Simple)	ME/EE
Complete Rack Structure Assembly Drawing 6 (Complex)	ME/EE
Rack 3	Discipline
Complete Box Top Assy Drawing 1 (Complex)	ME/EE
Complete Box Enclosure Assy (Standard)	ME/EE
Complete Box Enclosure Panels (Standard)	ME/EE
Rack 3 Box 1	Discipline
Complete Box Top Assy Drawing 1 (Complex)	ME/EE
Complete Box Enclosure Assy (Standard)	ME/EE
Complete Box Enclosure Panels (Standard)	ME/EE
Rack 3 Box 2	Discipline
Complete Box Top Assy Drawing 1 (Complex)	ME/EE
Complete Box Enclosure Assy (Standard)	ME/EE
Complete Box Enclosure Panels (Standard)	ME/EE
Rack 3 All Boxes	Discipline
Complete Box Top Assy Drawing 1 (Complex)	ME/EE
Complete Box Enclosure Assy (Standard)	ME/EE
Complete Box Enclosure Panels (Standard)	ME/EE
Rack Top Levels	Discipline
Complete Top-Level Assembly (Complex)	ME/EE
Complete P&ID (Complex)	ME/EE
Complete Installation Drawing (Complex)	ME/EE
Complete Enclosure Panels (Simple)	ME/EE
Complete Rack Sub Assembly Drawing 5 (Simple)	ME/EE
Complete Rack Structure Assembly Drawing 6 (Complex)	ME/EE
Order Parts	Discipline
Order COTS Instruments	PE
Order Custom Parts	PE
Pre-TRR Preparation	Discipline
Prepare Documents	PE
Review Documents (Jacobs internal)	PE
Hold TRR Presentation	PE
Documentation	Discipline
Operation Manual (Deliverable)	PE/ME/Tech
Hazard Analysis (Deliverable)	PE/ME/Tech
Test Procedure (Deliverable)	PE/ME/Tech
Task Name	Discipline
Generate Code Review Presentation	SW
Host Code Review	SW
Respond to Comments from Code Review	SW
Modify SW Based on Feedback	SW

Functional Checkout of	Discipline
Open TPS	PE
Class I certification with PSMO Representative	PE/Tech
Pressurize Rig	PE/Tech
Perform Rig Acceptance Test and Functional Checkout	PE/Tech
Complete Quality Verifications	PE/Tech
Document Any Design Changes/Redlines	PE/Tech
DR# XXXXX (Placeholder)	PE/Tech
Close TPS	PE
Update Drawings to as built for Final Release (ALL DRAWINGS)	ME/EE
TRR Preparation	Discipline
Distribute TRR Documents for Review Period (>= 5 days)	PE
Hold TRR (Milestone)	PE
Closeout Immediate TRR Actions	PE
Closeout Remaining TRR Actions	PE
Test Station Delivery	Discipline
File All Documentation	PE
Send Formal Delivery Notification to Customer	PM
Data Overview	Discipline
Meet with PE to determine DAQ components	EE
Identify National Instruments Hardware	EE
Identify DAQ Box Lights/Switches	EE
Finalize Peer Review of DAQ BD	EE
Finalize Block Diagram	EE
Safety Overview	Discipline
Meet with PE to determine Safety components	EE
Identify Safety Box Lights/Switches	EE
Finalize Peer Review of Safety Box BD	EE
Finalize Block Diagram	EE
Derating	Discipline
Meet with PE to determine harness connections	EE
Determine facility and rig connections	EE
Determine preliminary connections between boxes	EE
Create Harness Exploration	EE
Peer Review Harness Exploration	EE
Software Design	Discipline
Gather Requirements from Project Engineer	SW
Gather EE inputs from Electrical Engineer	SW
Determine Level of Automation Required from Project Engineer	SW
Construct Preliminary LabVIEW Architecture	SW
Verify with PE & EE Preliminary Architecture of Software	SW
Hazard Identification	Discipline
Identify Hazards (Rig, Facility, Test Article)	PE
Identify Hazard Controls	PE
Identify Critical/Non-Critical Components	PE

Software Models	Discipline
Create new module	SW
Test New Module on HW	SW
Integrate New Module into Architecture	SW
Intermediate ME Items	Discipline
Complete Model Intermediate Iteration	ME
Hold Internal Design Review with GSE Design Lead	ME
Gain Approval from GSE Design Lead	ME
Create Stress Ticket	ME
Create Thermal Ticket(s)	ME
Meet with Stress Analysis Team and Deliver Ticket and Model	ME
Meet with Thermal Analysis Team and Deliver Tickets and Models	ME
Top Level EE Block Diagram	Discipline
Collate all box and harness items for Top Level BD	EE
Complete Altium Block Diagram	EE
Deliver Block Diagram to Drafting - Feeds Interconnect drawing	EE
Complete Data Box	Discipline
Finalize Back Panel Connector Selection	EE
Complete Circuit/Schematic Diagram	EE
Complete BOM of Electronics	EE
Complete Wiring Table for Techs	EE
Deliver DAQ Box Package to Mechanical Design	EE
Complete Safety Box	Discipline
Finalize Back Panel Connector Selection	EE
Complete Circuit/Schematic Diagram	EE
Complete BOM of Electronics	EE
Complete Wiring Table for Techs	EE
Deliver Safety Box Package to Mechanical Design	EE
Complete Derating	Discipline
Verify Power Box Derating	EE
Verify DAQ Box Derating	EE
Verify Safety Box Derating	EE
Verify Harness Derating	EE
Formal Pressure Design	Discipline
Technical Review of Initial Pressure Systems Calcs	PE
Create OCCP for any custom pressurized parts	PE
Prepare PSMO package	PE
PSMO Review and signoff of Design	PE
Complete Theory of Operations	Discipline
Provide full electrical design justification	EE
Provide evidence of standards compliance	EE
Configurate Automation	Discipline
Present Detailed Automation Flowchart/Logic Plan (If needed)	SW
Generate Automation Functionality in LabVIEW	SW
Procurements	Discipline

Buy COTS Boxes	PE
Fabricate P&ID	Discipline
Open TPS	PE
Layout/Fabricate Plumbing	PE/Tech/ME
Complete Full Rough Assembly	PE/Tech/ME
Document Any Design Changes/Redlines	PE/Tech/ME
Disassemble and Prep for Cleaning/Calibration	PE/Tech/ME
Close TPS	PE
Update Drawings to as built for Final Release	PE/Tech/ME
Cleaning Items	Discipline
Open TPS	PE
Complete Cleaning Form #XXX	PE
Deliver to B9	PE
Estimated Cleaning Lead Time	PE
Receive from B9	PE
Close TPS	PE
Calibrating Items	Discipline
Open TPS	PE
Complete Cleaning Form #XXX	PE
Deliver to B9	PE
Estimated Cleaning Lead Time	PE
Receive from B9	PE
Close TPS	PE
Fabricate Data Box Assembly	Discipline
Open TPS	PE/Tech/ME
Assemble	PE/Tech/ME
Document Any Design Changes/Redlines	PE/Tech/ME
Obtain Quality Verifications	PE/Tech/ME
Complete DR (if applicable)	PE/Tech/ME
Close TPS	PE/Tech/ME
Update Drawings to as built for Final Release	PE/Tech/ME
Fabricate Safety Box Assembly	Discipline
Open TPS	PE/Tech/ME
Assemble	PE/Tech/ME
Document Any Design Changes/Redlines	PE/Tech/ME
Obtain Quality Verifications	PE/Tech/ME
Complete DR (if applicable)	PE/Tech/ME
Close TPS	PE/Tech/ME
Update Drawings to as built for Final Release	PE/Tech/ME
Calibrating Items	Discipline
Open TPS	PE
Assemble Rack 1 Mechanical Structure	PE/Tech/ME
Close TPS	PE
Update Drawings to as built for Final Release	PE
Open TPS	PE/Tech/ME
Assemble Rack 2 Mechanical Structure	PE

Close TPS	PE
Update Drawings to as built for Final Release	PE/Tech/ME
Open TPS	PE
Assemble Rack 3 Mechanical Structure	PE
Close TPS	PE/Tech/ME
Update Drawings to as built for Final Release	PE
Complete DR (if applicable)	PE
Calibrating Items	Discipline
Open TPS	PE
Assemble Rack 1 Component Structure	PE/Tech/ME
Close TPS	PE
Update Drawings to as built for Final Release	PE
Open TPS	PE/Tech/ME
Assemble Rack 2 Component Structure	PE
Close TPS	PE
Update Drawings to as built for Final Release	PE/Tech/ME
Open TPS	PE
Assemble Rack 3 Component Structure	PE
Close TPS	PE/Tech/ME
Update Drawings to as built for Final Release	PE
Complete DR (if applicable)	PE
Open TPS	PE
Assemble Final Piping Assembly	PE/Tech/ME
Close TPS	PE
Update Drawings to as built for Final Release	ME
Complete DR (if applicable)	PE
Open TPS	PE
Assemble Final Electrical Assembly	PE/Tech/EE
Close TPS	PE
Update Drawings to as built for Final Release	EE
Complete DR (if applicable)	PE

The schedule with incorporated sprints included in excess of 2,400-line items, 13 sprints and 451 days of schedule. It is important to note that this schedule is nearly 6 months longer than the original schedule before the case study began had anticipated, suggesting that the original project schedule was erroneous. This schedule also considered historical data on actual GSE schedule time that was previously unavailable.

Table 50 – Iteration 1A Full Schedule

Task Name	Duration	Start	Finish
Detail Development Schedule	511 days	Wed 10/30/19	Wed 10/13/21
GSE Kick-Off Meeting (Milestone) (Sprint 1)	39 days	Mon 12/2/19	Thu 1/23/20
Sprint Planning Meeting 1	1 day	Mon 12/2/19	Mon 12/2/19
Prepare Documents Sprint 1A	16 days	Tue 12/3/19	Tue 12/24/19
Prepare Documents Sprint 1B	16 days	Wed 12/25/19	Wed 1/15/20
Sprint Review	5 days	Thu 1/16/20	Wed 1/22/20
Sprint Retrospective	1 day	Thu 1/23/20	Thu 1/23/20
Initial Design Review Meeting (Milestone) (Sprint 2)	22 days	Fri 1/24/20	Mon 2/24/20
Complete Preliminary Designs (Sprint 2)	22 days	Fri 1/24/20	Mon 2/24/20
Sprint Planning Meeting 2 (All Sprints Concurrent)	1 day	Fri 1/24/20	Fri 1/24/20
Preliminary Mechanical Design 2A	15 days	Fri 1/24/20	Thu 2/13/20
Preliminary Electrical Design 2B	10 days	Fri 1/24/20	Thu 2/6/20
Preliminary Software Design 2C	18 days	Fri 1/24/20	Tue 2/18/20
Hazard Identification 2D	15 days	Fri 1/24/20	Thu 2/13/20
Sprint Review	3 days	Wed 2/19/20	Fri 2/21/20
Sprint Retrospective	1 day	Mon 2/24/20	Mon 2/24/20
Detail Design and Drafting	203 days	Tue 2/25/20	Thu 12/3/20
Create Formal Mechanical Design	25 days	Tue 2/25/20	Mon 3/30/20
Complete Intermediate & Formal Mechanical/Electrical/Pressure Designs (Sprint 3)	25 days	Tue 2/25/20	Mon 3/30/20
Sprint Planning Meeting 3 (All Sprints Concurrent)	1 day	Tue 2/25/20	Tue 2/25/20
Complete ME Intermediate Sprint 3A	12 days	Wed 2/26/20	Thu 3/12/20
Sprint Review	3 days	Wed 3/25/20	Fri 3/27/20
Sprint Retrospective	1 day	Mon 3/30/20	Mon 3/30/20
Create Drawings and Fab Release	178 days	Tue 3/31/20	Thu 12/3/20
First Round ME/EE Drawings Sprint 4	61 days	Tue 3/31/20	Tue 6/23/20
Sprint Planning Meeting 4 (All Sprints Concurrent)	1 day	Tue 3/31/20	Tue 3/31/20
SIPPE Test Rig Rack 1 (Sprint 4A)	56 days	Wed 4/1/20	Wed 6/17/20
Sprint Review	3 days	Thu 6/18/20	Mon 6/22/20
Sprint Retrospective	1 day	Tue 6/23/20	Tue 6/23/20
Second Round ME/EE Drawings Sprint 5	59 days	Wed 6/24/20	Mon 9/14/20
Sprint Planning Meeting 5 (All Sprints Concurrent)	1 day	Wed 6/24/20	Wed 6/24/20
SIPPE Test Rig Rack 2 Sprint 5A	54 days	Thu 6/25/20	Tue 9/8/20
Sprint Review	3 days	Wed 9/9/20	Fri 9/11/20
Sprint Retrospective	1 day	Mon 9/14/20	Mon 9/14/20
Third Round ME/EE Drawings Sprint 6	58 days	Tue 9/15/20	Thu 12/3/20
Sprint Planning Meeting 6 (All Sprints Concurrent)	1 day	Tue 9/15/20	Tue 9/15/20
SIPPE Test Rig Rack 3 (Sprint 6)	53 days	Wed 9/16/20	Fri 11/27/20
SIPPE Non-Exclusive Rack Items (Sprint 6)	41 days	Wed 9/16/20	Wed 11/11/20
Sprint Review	3 days	Mon 11/30/20	Wed 12/2/20
Sprint Retrospective	1 day	Thu 12/3/20	Thu 12/3/20
Create Formal Electrical Design	81 days	Wed 2/26/20	Wed 6/17/20
All Rig Drawings Fab Released (Deliverable)	0 days	Thu 12/3/20	Thu 12/3/20
Create Formal Software Design	126 days	Wed 10/30/19	Wed 4/22/20
Initial SW Considerations Sprint 1C	6 days	Wed 10/30/19	Wed 11/6/19
Configure New Modules (if non-existent in database 9 Modules) Sprint 2E	15 days	Mon 1/27/20	Fri 2/14/20
Configure Automation Sprint 3H	15 days	Wed 2/26/20	Tue 3/17/20
Generate User Interface Sprint 4E	20 days	Wed 3/18/20	Tue 4/14/20
Documentation Sprint 4F	11 days	Wed 4/8/20	Wed 4/22/20
Pre-Fab Design Review (Milestone) (Sprints 3 - 6)	7 days	Fri 12/4/20	Mon 12/14/20
Procurement Sprint 7 (with Sprit 1 & 3 Portions)	403 days	Wed 10/30/19	Fri 5/14/21
Sprint Planning Meeting 7	1 day	Fri 12/4/20	Fri 12/4/20
COTS Instruments Sprint 7A (Sprint Completion = Ordering Only, Not Delivery)	116 days	Thu 12/3/20	Fri 5/14/21
COTS Components Sprint 1D (Sprint Completion = Ordering Only, Not Delivery)	167 days	Wed 10/30/19	Thu 6/18/20
COTS Boxes Sprint 3I (Sprint Completion = Ordering Only, Not Delivery)	248 days	Tue 3/10/20	Fri 2/19/21

COTS Computer & IT Sprint 1E (Sprint Completion = Ordering Only, Not Delivery)	106 days	Wed 10/30/19	Wed 3/25/20
Custom Parts Sprint 7B (Sprint Completion = Ordering Only, Not Delivery)	88 days	Fri 12/4/20	Tue 4/6/21
Sprint Review	1 day	Fri 12/25/20	Fri 12/25/20
Sprint Retrospective	1 day	Mon 12/28/20	Mon 12/28/20
Fabrication/Assembly	286 days	Thu 6/18/20	Thu 7/22/21
Fabricate Top Level Mechanical Assembly	**71 days**	**Wed 4/7/21**	**Wed 7/14/21**
Fabricate Rack #1	**71 days**	**Wed 4/7/21**	**Wed 7/14/21**
Sprint Planning Meeting 10 (All Sprints Concurrent)	1 day	Wed 4/7/21	Wed 4/7/21
Structure Assembly Sprint 10A	**49 days**	**Wed 4/7/21**	**Mon 6/14/21**
Sprint Review	3 days	Tue 6/15/21	Thu 6/17/21
Sprint Retrospective	1 day	Fri 6/18/21	Fri 6/18/21
Sprint Planning Meeting 11 (All Sprints Concurrent)	1 day	Mon 6/21/21	Mon 6/21/21
Component Assembly Sprint 11A	**65 days**	**Thu 4/15/21**	**Wed 7/14/21**
Sprint Review	3 days	Fri 6/18/21	Tue 6/22/21
Sprint Retrospective	1 day	Wed 6/23/21	Wed 6/23/21
Fabricate Rack #2	**45 days**	**Wed 4/7/21**	**Tue 6/8/21**
Structure Assembly Sprint 10B	**31 days**	**Wed 4/7/21**	**Wed 5/19/21**
Component Sprint 11B	**39 days**	**Thu 4/15/21**	**Tue 6/8/21**
Fabricate Rack #3	**55 days**	**Tue 4/13/21**	**Mon 6/28/21**
Structure Assembly Sprint 10C	**52 days**	**Tue 4/13/21**	**Wed 6/23/21**
Component Assembly Sprint 11C	**48 days**	**Thu 4/22/21**	**Mon 6/28/21**
Fabricate P&ID Assembly (Flow Loop)	**126 days**	**Mon 12/7/20**	**Mon 5/31/21**
TPS #XXXXX (Rough Assembly) Sprint 7C	**35 days**	**Mon 12/7/20**	**Fri 1/22/21**
TPS #XXXXX (Cleaning)	**34 days**	**Mon 1/25/21**	**Thu 3/11/21**
Sprint Planning Meeting 8	1 day	Mon 1/25/21	Mon 1/25/21
Cleaning Sprint	**34 days**	**Mon 1/25/21**	**Thu 3/11/21**
Sprint Review	0 days	Thu 2/25/21	Thu 2/25/21
Sprint Retrospective	1 day	Fri 2/26/21	Fri 2/26/21
TPS #XXXXX (Calibration)	**34 days**	**Fri 3/12/21**	**Wed 4/28/21**
TPS #XXXXX (Final Assembly) Sprint 11D	**23 days**	**Thu 4/29/21**	**Mon 5/31/21**
Fabricate EID (Electrical Loop) Sprint 11E	**21 days**	**Thu 6/24/21**	**Thu 7/22/21**
Fabricate DAQ Box Assembly Sprint 9B	**36 days**	**Mon 2/22/21**	**Mon 4/12/21**
Fabricate Safety Box Assembly Sprint 9C	**36 days**	**Mon 2/22/21**	**Mon 4/12/21**
Fabricate Harness Assembly Sprint 5B	**34 days**	**Thu 6/18/20**	**Tue 8/4/20**
All Rig Drawings Final Released (Deliverable)	25 days	Fri 7/23/21	Thu 8/26/21
Documentation (Deliverable) Sprint 12A (Finish Documents Only, Not Release)	207 days	Tue 12/29/20	Wed 10/13/21
Operation Manual (Deliverable)	168 days	Tue 12/29/20	Thu 8/19/21
Hazard Analysis (Deliverable)	54 days	Fri 7/23/21	Wed 10/6/21
Test Procedure (Deliverable)	59 days	Fri 7/23/21	Wed 10/13/21
Pre-TRR Review (Milestone)	20 days	Fri 7/23/21	Thu 8/19/21
Sprint Planning Meeting 12 (All Sprints Concurrent)	1 day	Fri 7/23/21	Fri 7/23/21
Pre-TRR Sprint 12B	**10 days**	**Fri 7/23/21**	**Thu 8/5/21**
Sprint Review	**0 days**	**Thu 8/5/21**	**Thu 8/5/21**
Sprint Retrospective	10 days	Thu 8/5/21	Thu 8/19/21
Software Integration	7 days	Mon 7/26/21	Tue 8/3/21
Finish Code Review Sprint 12C	7 days	Mon 7/26/21	Tue 8/3/21
TRR and Test Station Delivery (Milestone & Deliverable)	37 days	Thu 8/19/21	Fri 10/8/21
Sprint Planning Meeting 13 (All Sprints Concurrent)	1 day	Thu 8/19/21	Thu 8/19/21
Functional Checkout of Test Station Sprint 13A	**17 days**	**Fri 8/20/21**	**Mon 9/13/21**
TRR Sprint 13B	**11 days**	**Tue 9/14/21**	**Tue 9/28/21**
Sprint Review	**5 days**	**Wed 9/29/21**	**Tue 10/5/21**
Sprint Retrospective	**3 days**	**Wed 10/6/21**	**Fri 10/8/21**

A typical sprint (Sprint #1) is shown for clarity. Note that the project schedule is shown notionally with the dates from the original GSE SIPPE project start date.

Task Name	Duration	Start	Finish
GSE Kick-Off Meeting (Milestone) (Sprint 1)	4 wks	Mon 12/2/19	Fri 12/27/19
Sprint Planning Meeting 1	0.2 wks	Mon 12/2/19	Mon 12/2/19
Backlog Review	1 day	Mon 12/2/19	Mon 12/2/19
Team Discussions and Schedule Estimation	0.2 wks	Mon 12/2/19	Mon 12/2/19
Define Sprint Goal	0.2 wks	Mon 12/2/19	Mon 12/2/19
Establish the Definition of Done	0.2 wks	Mon 12/2/19	Mon 12/2/19
Plan To Deliver Sprint Goal	0.2 wks	Mon 12/2/19	Mon 12/2/19
Prepare Documents Sprint 1A	3.2 wks	Tue 12/3/19	Tue 12/24/19
Meet with Component Owner and Review Rig Requirements	16 days	Tue 12/3/19	Tue 12/24/19
Meet with Resource Managers & Receive ROMs	16 days	Tue 12/3/19	Tue 12/24/19
Create Rig Cost Estimate	16 days	Tue 12/3/19	Tue 12/24/19
Create Completion Form PD	16 days	Tue 12/3/19	Tue 12/24/19
Create Rig Schedule	16 days	Tue 12/3/19	Tue 12/24/19
Define Rig Scope	16 days	Tue 12/3/19	Tue 12/24/19
Prepare Documents Sprint 1B	3.2 wks	Tue 12/3/19	Tue 12/24/19
Create Mechanical P&ID	16 days	Tue 12/3/19	Tue 12/24/19
Create Electrical Block Diagram	3.2 wks	Tue 12/3/19	Tue 12/24/19
Create Rough CAD Model	3.2 wks	Tue 12/3/19	Tue 12/24/19
Create PowerPoint presentation	3.2 wks	Tue 12/3/19	Tue 12/24/19
Sprint Review	0.4 wks	Wed 12/25/19	Thu 12/26/19
PM approval	1 day	Wed 12/25/19	Wed 12/25/19
Technical Reviewer approval	1 day	Wed 12/25/19	Wed 12/25/19
Hold GSE Kick-Off Meeting (Milestone)	1 day	Wed 12/25/19	Wed 12/25/19
Deliver Product Increment: Contract & Initial Parts Ordered, SW Defined	0 days	Wed 12/25/19	Wed 12/25/19
Create Minutes and Action Items List	1 day	Thu 12/26/19	Thu 12/26/19
Send Formal Completion Email with Notes	0 days	Wed 12/25/19	Wed 12/25/19
Sprint Retrospective	0.2 wks	Fri 12/27/19	Fri 12/27/19
Reflect on the Process	1 day	Fri 12/27/19	Fri 12/27/19
Identify Potential Improvements	1 day	Fri 12/27/19	Fri 12/27/19

Figure 43 – Sprint 1 Detailed Schedule Approach

Step 5: Improvements & Limitations

Each iteration will present a series of improvements and limitations. For Iteration 1, improvements to model allows for incremental and iterative deliveries to promote value to the customer quickly, allows for priorities to be organized periodically, utilizes and burndown chart for schedule forecast, promotes generalists over specialists via cross training, appoints a servant leader in a ScrumMaster, allows for consistent customer engagement and promotes learning by osmosis. Limitations include many of the improvements instigated yet derives a means by which to achieve them. These include:

- How to develop hardware with scrum: This is a software-intensive process as software can change and iterate much more quickly and cheaply as opposed to hardware. These challenges include a series of constraints of physicality in purchasing, modification and changing of hardware vs. software.
    - Goal: successfully provide scrum framework tailoring to optimize hardware development teams.
- The approach that must be taken to organize work priorities, workflow and characterization of true schedule: while a template is given to organize work, this varies greatly by how hardware vs. software is built. While priorities may seem apparent at the onset, certain aspects of hardware development cannot quickly and easily be incremented and implemented and as a result, backlog organization, workflow and characterization of true schedule remain challenges.
    - Goal: characterize workflow, priority organization and schedule prediction by developing a method to illustrate, organize and improve schedule performance.
- Challenges associated with cross functional development: While it is attractive to have a team of generalists, how to train and cross-develop these individuals is a challenge in itself before the team can be considered M-Shaped vs. I-Shaped.
    - Goal: find a method to cross-train all various disciplines of the development team.
- How to develop to identify and optimize impediment improvement: A ScrumMaster will allow for the first step of address impediments, especially those seen terms of process, but how to implement them methodically and identify and address is still unaddressed.

    - Goal: how to methodically identify, address and remove impediments by means of a ScrumMaster.
- How to integrate if required with waterfall: At the conclusion of the iteration and before the onset of Iteration 2, it is evident that a standalone Scrum model may not be sufficient to address not only the needs of the performing organization but also its compatibility.
    - Goal: (if required) find a means to combine best practices from both methods while improving or eliminating the shortcomings (i.e., inability for waterfall to quickly address change, challenge for Scrum to be implemented with hardware teams).
- How to work with a remote working team: As complications from COVID intermingled with the challenges already described, Scrum promotes itself by working with a centralized team. How to address this with the advent of COVID cannot be understated.
    - Goal: how to fully integrate a remotely working team unable to learn by osmosis in an office environment.
- How to obtain acceptance by customers, development team and performing organization: Maybe the most significant challenge would be acceptance from customer and performing organization. But first, acceptance by the develop team needs to be harnessed.
    - Goal: creating a process by which the development team and subsequently customer and performing organization can vet via survey to gain acceptance of tempered model.

Step 6: Overall Summary of Iteration

An abridged table is given to reflect the aforementioned iteration number, time duration, data gathering techniques, improvements and limitations.

Table 51 – Iteration 1A Summary

Iteration 1	Time Period	Data Gathering	Improvements	Limitations
Scrum Team Developed	February 2021 - February 2022	Initial Literature Review	Allows for Incremental and Deliveries	How To Develop Hardware with Scrum
		Case Study Review	Allows for Iterative Deliveries	How To Organize Work Priorities
		Earned Value Analysis	Allow for Priorities to be Organized Periodically	How To Demonstrate Workflow
		Brainstorming	Utilizes a Burndown Chart for Schedule Forecast	How To Characterize True Schedule
		Interviews	Promotes Generalists over Specialists	How To Create Cross-Functional Generalists
Waterfall Removed		Focus Groups	Appoints a Servant Leader in a ScrumMaster	How To Develop Impediment Improvements
		Questionnaires	Allows for Consistent Customer Engagement	How To Integrate with Traditional Waterfall
		1st Survey	Promotes Learning by Osmosis	How To Utilize with a Remotely Working Team
				How To Address Processes That Don't Add Value
				Acceptance by Development Team

8.6. ITERATION 2A: Project Lifecycle Development

8.6.1. Iteration Development

In the second iteration, Lean is introduced to bring impediments to the via a modification to the traditional FMEA method. The period of performance of Iteration 2 focused between February 2022 and May 2022. The data gathering techniques included:

- Intermediate Literature Review: Where Lean concepts from academic and peer reviewed journals are viewed.
- Process Investigation: A deep dive into the areas of waste across GSE projects which may be augmented or eliminated to provide the expected value with improvements to cost and schedule while preserving quality and safety

expectations.

- Brainstorming: Various, semi-formal sessions were elicited with stakeholders from across the project to assess which areas of systems engineering presented challenges and for project lifecycle development, those specific challenges documented.
- Interviews: Various, semi-formal sessions were elicited with stakeholders from across the project to assess which areas of systems engineering presented challenges and for project lifecycle development, those specific challenges documented.
- Focus Groups: Various, semi-formal sessions were elicited with stakeholders from across the project to assess which areas of systems engineering presented challenges and for project lifecycle development, those specific challenges documented.
- Questionnaires: Various, semi-formal sessions were elicited with stakeholders from across the project to assess which areas of systems engineering presented challenges and for project lifecycle development, those specific challenges documented.
- 2nd Survey: A Likert scale survey to assess the current vs. desired state of Lean offerings on the project with responses from the Development Team and customer.

The goal of the iteration is to develop Lean into Scrum development. This will be done by identifying the 7 Forms of Lean Waste, creating a prompt list with metric collection, using a modified FMEA to quantify process improvement with results reporting.

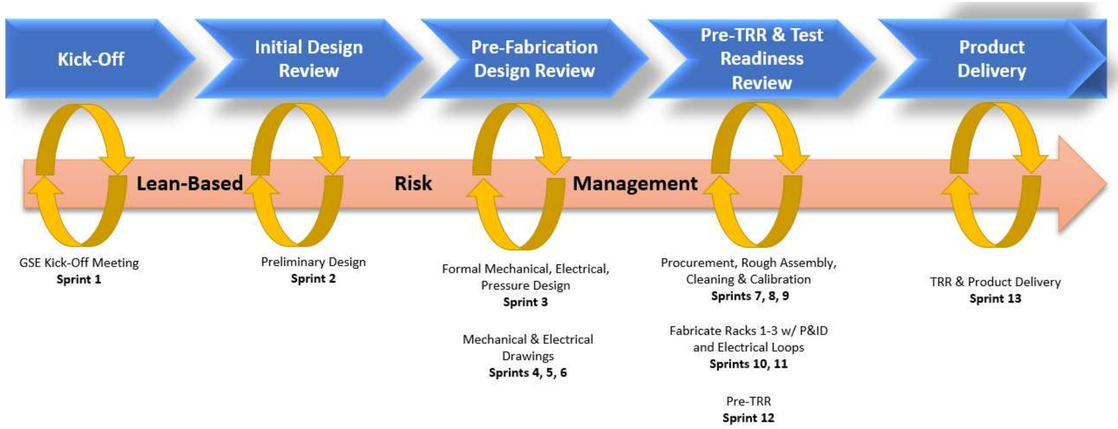


Figure 44 - Iteration 2A Lifecycle Development

A survey with 24 individuals across the project (customer and performing organization) were polled on the Likert scale to elicit information regarding Lean and the desired vs. current delivery method that project had offered Scrum-related artifacts; 1 – Strongly Disagree, 5 – Strongly Agree; any score above a 3.5 (median between "Neither Agree or Disagree and Somewhat Agree) is deemed a positive indication on position; any score below a 3 is considered a negative indication on position. The survey below was used to compare the desired vs. current Lean offering on project to examine how malleable the team would be to a change in lifecycle development. Results indicate that the current waterfall development has not offered a similar Lean artifact and that the team would prefer to operate in Lean in some capacity. Certain areas will either be exploited or avoided depending on the delta between desired and current project delivery methods.

Table 52 - Lean Current vs. Proposed Likert Questions

#	Field	Mean	Count
1	I prefer to minimize partially done work	4.26	23
2	I prefer producing finished work quickly via iterative delivery	4.09	23
3	I prefer for the development team to make local decisions rather than the manager	4.13	23
4	I prefer working with less process	4.04	24
5	I prefer working only one project at a time	2.79	24
6	I prefer working with little delays in terms of work packages (i.e., approvals, signatures, etc.)	4.58	24

#	Field	Mean	Count
1	We minimize partially done work	2.96	23
2	We produce finished work quickly via iterative delivery	2.48	23
3	The development team usually makes local decisions rather than the manager	3	23
4	We work with minimal process	1.7	23
5	We work only one project at a time	2.29	24
6	We work with little delays in terms of work packages (i.e., approvals, signatures, etc.)	1.65	23

#	Desired Delivery Method	Current Delivery Method	Delta	Exploit Approach	Avoid Approach
1	4.26	2.96	1.3	X	
2	4.09	2.48	1.61	X	
3	4.13	3	1.13	X	
4	4.04	1.7	2.34	X	
5	2.79	2.29	0.5		X
6	4.58	1.65	2.93	X	

A questionnaire with 24 individuals across the project (customer and performing organization) was given to allow for opened-ended feedback. The question and abridged answers suggested agile and, in some cases, Lean approaches. Responses are as follows:

- *Q2: What standards, practices, actions or any other comments would help optimize project lifecycle development on the xEMU project?*
- *A2:*
  - *"A pre-assessment prior to contractual work beginning would be of value to remove or substitute such items in the design process from causing resistance*

to flow."

- *"Requiring less signatures on paperwork for the developmental phase or having a plan in place for other signature authorities when one takes more than 24 hours to sign something."*
- *"Toll gate processes…"*
- *"Minimum purchase thresholds to expedite product development and immediate problem evaluation with all needed parties. Remove bottleneck of 1 decision maker. Let design team do what they do without micromanaging."*
- *"Just to remove unnecessary time delays…"*
- *"Team wide agreement on processes and procedures, very frequently ran into issues where we have to get clarity on which rules apply to which rooms and projects and guiding documents not being accurate."*
- *"Mistakes are repeated, more lessons learned. Actual processes are highly tailored & usually poorly documented, which results in a steep learning curve for personnel, additional training based on CMAS is recommended. Requirements frequently change, this is going to happen, work needs to be accomplished in smaller chunks so changes can be incorporated without too much rework."*

Step 1: Identify the 7 Forms of Lean Waste

The GSE Testing Team understood but had not formally appraised the team's posture on Lean Waste mitigation efforts. As identified in the literature section, the forms of Lean Waste include defects, hand-offs, waiting/delays, task switching, extra processing/documentation, unnecessary features, incomplete work. A focus group of 8 team members with a facilitator leading the effort began the process of Lean

Waste identification with subsequent prompt list, metric collection and modified FMEA. The team identified the following areas of Lean Waste: TPS/DRs, drawing signatures, drafting review cycle, procurements, calibration cycles, cleaning cycles, document cycle time, JETS drawing review process, customer suggested drawing process.

Step 2: Create a Prompt List for Metric Collection

The focus group of 8 team members provided the facilitator with a list of Lean Waste areas with project data points on average wait times.

- <u>TPS and DRs: 41 data points.</u>
    - most projects have 3 figures worth of TPSs and DRs, remove process *most* of the time.
- <u>Drawing signatures: 51 data points.</u>
    - Projects can have up to 40 drawings, remove process *most* of the time.
- <u>Drafting review cycle & JETS Drawing review process: 50 data points.</u>
    - Projects can have up to 40 drawings, remove process *most* of the time.
- <u>Procurements: 525 data points.</u>
    - Projects can have up to 250 purchases, remove process *most* of the time.
- <u>Calibration cycle: 118.</u>
    - Projects can have ~20 items to calibrate, reduce process.
- <u>Cleaning cycle: 235 data points.</u>
    - Projects can have 3 figures worth of items to clean, reduce process.
- <u>Document cycle time: 20 data points.</u>
    - Projects can have 5-10 items with one document taking as much as 8 months to sign (in critical path).

- Reduce process.

- <u>Customer Suggested Drawing Process: no data points.</u>
  - This was a suggestion by the customer on drawing process approaches and as such, no metrics exist.
  - The goal is to identify if the approach is taken, how risky it may be.

The team then provided by Delphi Method on potential wait time if the area of Lean Waste was augmented or removed. A sample of the Delphi Method for selected Lean Waste categories is given below. In the event that a possible form of Lean Waste may be totally eliminated, no Delphi Method assessment was performed.

Table 53 – Delphi Method Approach for GSE Testing Team

Assign Number/Letter To Engineer Polled	List Activity and Poll Three Estimate Numbers			Beta Distibution	Triangular Distribution	Variances
Engineer #	Optimistic Time	Most Likely Time	Pessimistic Time	(Best+4*Likely+Worst)/6	(Best+Likely+Worst)/3	((Worst-Likely)/6)^2
Internal Drafting Review						
1	1	2.5	5	3	3	0.44
2	2	3	4	3	3	0.11
3	1	3	5	3	3	0.44
4	0.5	3	5	3	3	0.56
Average	1.125	2.875	4.75	2.895833333	2.916666667	0.390625
Cleaning Times (If ScrumMaster Maintains Priority)						
1	1	2	3	2	3	0.11
2	2	4	5	4	5	0.25
3	1	2	4	2	3	0.25
4	1	3	5	3	4	0.44
5	2	3	5	3	4	0.25
6	0.5	3	5	3	3	0.56
7	1	2	3	2	2	0.11
8	0.5	2	3	2		0.17
Average	1.125	2.625	4.125	2.625	3.456349206	0.269097222
Cleaning Times (If ScrumMaster Maintains Priority)						
1	1	2	3	2	2	0.11
2	1	2	4	2	2	0.25
3	0.5	1	3	1	2	0.17
4	2	3	4	3	3	0.11
5	1	5	6	3	4	0.69
6	1	5	6	5	4	0.69
7	1	5	6	5	4	0.69
8	2	3	5	3	3	0.25
Average	1.1875	3.25	4.625	2.947916667	3.020833333	0.372395833
Simple Fab Signatures						
1	1	2	3	2	2	0.11
2	1	2	3	2	2	0.11
3	1	2	3	2	2	0.11
4	1	2	3	2	2	0.11
Average	1	2	3	2	2	0.111111111

Finally, of all the 7 Forms of Lean Waste identified, the average wait times were collected and juxtaposed against GSE Testing Team focus group potential wait times.

Table 54 – Lean Prompt List

#	Area of Concern	Internal or External	Project Management Area(s) Affected	Lean Waste Category	Additional Information	Average Wait Time	Potential Wait Time	Potential Corrective Action?
1	JETS Drawing Review Process	Internal	Schedule, Cost	Waiting, Hand-off, Task Switching, Extra Processing	The GSE team has had challenges with the time it takes to go through the entire performing organization-imposed drawing review process.	30	3	Abridged signature process only to include necessary signatures based on GSE lead recommendation
2	Cleaning Cycle Times	External	Schedule	Hand-off, Waiting, Task Switching	The GSE team has had challenges with delivery on certain items sent to the cleaning department.	16	3	Cleaning times have been improved over the years with the queue through NASA being more manageable. One of the ways the team can improve cleaning times is by maintaining priority with the institution of a ScrumMaster to coordinate with the cleaning department.
3	Calibration Cycle Times	External	Schedule	Hand-off, Waiting, Task Switching	The GSE team has had challenges with delivery on certain items sent to the calibration department.	31	3	Reduced calibration times on certain items including those that have not been used or do not require yearly calibration, a dedicated ScrumMaster to coordinate ahead of time with the calibration lab, a dedicated facility strictly for GSE calibrations through JETS, an additional, external calibration vendor to reduce the 1600 item queue that at one point been in calibration.
4	Purchasing & Quality Assurance	Internal	Schedule, Cost	Extra Processing, Unnecessary Features	The GSE team has had challenges with delivery on certain items sent to the purchasing department with quality-imposed quality codes to GSE hardware.	72	47	By reducing the quality codes on certain GSE procurements, lead times can be diminished. These include the lead times to find a vendor that can provide certain documents to satisfy quality codes and the lead time dedicated to the vendor providing said codes.
5	Document Signature Cycle Times	Internal/External	Schedule	Hand-off, Waiting, Task Switching, Extra Processing	The GSE team has had challenges with delivery on certain items sent to various external and internal stakeholders for signature.	20	5	Document signatures for procedures and hazard analysis will affect how quickly the development team can continue tasks to deliver a product. One of the ways in which the GSE team has determined cycle times can be reduced is by diminishing the review window from 2 weeks to 1 week and minimizing the number of signatures needed. This will maintain quality of product and also shorten the needed quality added from those signing.
6	Technical Process Specification Signatures Time	External	Schedule	Hand-off, Waiting, Task Switching, Extra Processing	The GSE team has had challenges with commencement of hardware delivery with TPS signature cycle times.	6	0	GSE products require TPS signatures primarily from the quality department. In certain cases, and if hardware is built outside of NASA facilities, the GSE team believes quality can be maintained and products can be built against a less stringent process wherein the process very similarly follows the Class III process without "paper."
7	Discrepancy Report Signatures Time	External	Schedule	Hand-off, Waiting, Task Switching, Extra Processing	The GSE team has had challenges with commencement of hardware delivery with DR signature cycle times.	7	0	GSE products require DR signatures primarily from the quality department. In certain cases, and if hardware is built outside of NASA facilities, the GSE team believes quality can be maintained and products can be built against a less stringent process wherein the process very similarly follows the Class III process without "paper."
8	Drafting Review Time	Internal	Schedule	Waiting, Hand-off, Task Switching, Extra Processing	The GSE team has had challenges with delivery on certain items sent to the drafting department.	15	3	The drafting department is an external department where many different project utilize their drafters. While there is value added to having a drafting review, either having a dedicated drafter or utilizing pair programming to eliminate waste times could reduce time in drafting significantly.
9	Customer Suggested Drawing Process	External	Schedule, Cost, Quality	Special: Intended to reduce extra processing	The customer has included a different drawing review process to reduce times however the performing organization has concerns regarding the risk this potentially simpler process may elicit.	Various, See Sample FMEA	Various, See Sample FMEA	In the supporting FMEA, the GSE team has distinguished the benefits of maintaining the current drawing method (Mono vs. Multi Detail) to illustrate while there appears to be a benefit to the expedited process, it may incur more risk and deliver less quality, pushing out cost and schedule.

Step 3: Create Modified FMEA to Quantify Process Improvement

The template provided allows for a FMEA to either demonstrate the potential improvements by removing a form of waste or challenging a proposed solution to illustrate the potential positive or negative effects to each of the categories of safety, cost, schedule, risk, scope and quality. As the focus group was using the FMEA, all rankings and failure modes were fully understood and agreed upon by all team members. While a strict ranking and associated numerical or qualitative value is open to flexibility, saliency is increased when the number of stakeholders included increases.

Once the Lean Waste and metrics are populated into prompt list, this will help inform the FMEA tool. The tool works twofold both as a risk management tool and FMEA. The tool contains the following categories for the first step of the process:

- Risk Number
    - What is the risk number associated with the process?
- Name
    - What is the name of the risk/opportunity associated with the process?
- Identification Number
    - What is the associated identification number of the risk/opportunity?
- Description
    - What is the risk associated with the process?
- Itemized From Description
    - How would these risks/opportunities decompose from the parent risk listed in the description?

        - Instead of a traditional failure mode, next level effects and end effects to the cascading effects are consolidated into one category for simplicity while still preserving and effectively illustrating the process.

- Impact Areas
    - Safety? Schedule? Quality? Cost? Scope?
- Risk or Opportunity Distinction
    - One of the hallmarks of this augmented FMEA is that it is modified to work inversely when compared to a typical FMEA. For example, one of the central purposes of a traditional FMEA is to reduce risk, which this FMEA functions as by identifying a risk with an associated likelihood and consequence at the onset and an updated likelihood and consequence evaluation after actions to correct the current project posture are proposed. In addition, the tool also functions as a means to understand if an opportunity that can improve schedule, budget or scope is sensitive to fluctuations in reduced quality or higher risks of safety. If the post likelihood and consequence are within an acceptable limit (i.e., in the green zone of the Likelihood and Consequence matrix), it could be deemed and a viable option to exploit the opportunity while safety and quality are still at acceptable levels.

Once population of the preliminary information is collected, an assessment on consequence and likelihood was performed.

- Consequence: How severe is the impact should the risk manifest?
- Likelihood: What is the probability of this risk manifesting?

Scales for each of the consequence and likelihood categories were utilized. Categories were presented with a general, non-numerical value as agreed upon by the focus group.

Table 55 – Lean Consequence Ranking

Consequence Ranking					
Category	**1**	**2**	**3**	**4**	**5**
Quality	Remote loss of quality	Minimal loss of quality	1 standard deviation away from quality standard	2 standard deviations away from quality standard	3 standard deviations away from quality standard
Safety	Remote risk of injury	Minimal risk of injury	Minor injury	Severe injury	Loss of life
Cost	< $50K impact	$50k to $100K impact	$100K to $250K impact	$250K to $500k impact	> $500K impact
Scope	Remote impact to scope objectives	Minimal impact to scope objectives	Considerable impact to scope objectives	Major impact to scope objectives	Severe impact to scope objectives
Schedule	Major disruption of service not involving client interaction and resulting in either associate re-work or inconvenience to clients	1-to-2-month impact	3-to-4-month impact	5-to-6-month impact	> 7-month impact to schedule

Table 56 – Lean Likelihood Ranking

Likelihood Ranking		
Score	**Description**	**Probability Range**
1	Very Unlikely	< 10 %
2	Unlikely	10% to 30%
3	Possible	> 30% to 60%
4	Likely	> 60% to 90%
5	Very Likely	> 90 %

At the conclusion of the consequence and likelihood assignment was the population of the Risk Priority Number (RPN). This RPN is given twice: once before analysis of alternatives and recommendations and once after analysis of alternatives or recommendations. The range is a number between 1 and 25. The template automatically populated a risk color associated with the degree of risk and requirement posture if left unmitigated. The RPN is a product of the two risk categories:

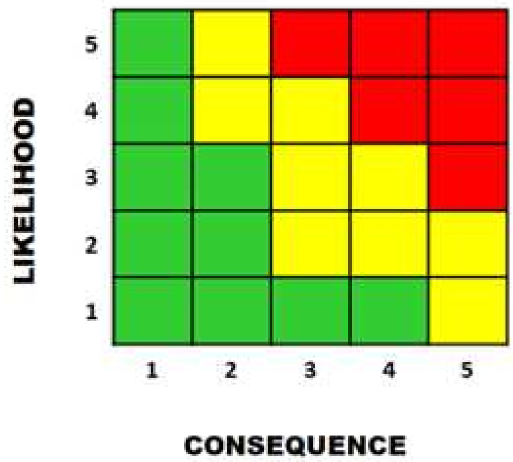


Figure 45 – Lean Likelihood x Consequence Matrix

After identification of the primary categories of potential issues and effects and assignment of a RPN, the next steps were to identify what risk mitigation efforts, if any, should be implemented.

- Action Recommended: What are the possible actions to remedy the requirement?
- Responsible Party: Who is responsible for making sure the actions are completed?
- Actions Taken: Will the Action Recommended be taken with respect to RPN?

The GSE Testing Team implemented corrective actions in the form of alternatives or recommendations from the previous step and updated the value of the RPN with the intention of reducing the risk posture of the requirement. As indicated previously, RPN is given twice: once before analysis of alternatives and recommendations and once after analysis of alternatives or recommendations. The range is a number between 1 and 25. The template automatically populated a risk color associated with the degree of risk and requirement posture if left unmitigated.

The sample template of the FMEA is shown below for Item #9 from the prompt list.

Table 57 - Lean FMEA Sample: Drawing Approach

PRELIMINARY INFORMATION							RPN INPUTS			POST RPN EVALUATION			POST RPN		
Risk #	Name	Description	ID#	Risk/Opportunity	Itemized From Description	Impact Areas?	CONSEQUENCE (1-5)	LIKELIHOOD (1 - 5)	RPN (1 - 25) (BEFORE ANALYSIS)	Action Recommended	Responsible Party	Actions Taken	CONSEQUENCE (1-5)	LIKELIHOOD (1 - 5)	RPN (1 - 25) (AFTER ANALYSIS)
What is the risk/opportunity number associated with the process?	What is the name of the risk/opportunity associated with the process?	What the risk associated with the process?	What's the identification?	Is this a risk or opportunity for the project?	How would these risks/opportunities decompose from the parent risk listed in the description?	Safety? Schedule? Quality? Cost? Scope?				What are the possible actions to remedy the potential risk or exploit opportunity?	Who is responsible for making sure the actions are completed?	Will the Action Recommended by taken with respect to RPN?			
9.1	Single Simplified Drawing Complexity (also referred to as Multi-Detail Drawing Method)	Single simplified drawing method will lead to a single drawing that is very long. This will lead to additional time in terms of drawing preparation, navigation for engineering/technicians, extended BOM that is difficult to navigate, and multiple revisions to change one.	91A	RISK	91A) RISK: Single drawing method will take longer to release due to multiple components, pages, etc.	Schedule / Cost	5	3	15	Switch to Mono-Detail Drawing Method	Project Manager	Yes	2	3	6
			91B	RISK	91B) RISK: Difficulty in navigating a long drawing for stakeholders (engineers, technicians, etc.) due to extended BOM, multiple pages, and all components on one page.	Schedule / Cost / Quality	3	2	6	Switch to Mono-Detail Drawing Method	Project Manager	Yes	1	2	2
			91C	RISK	91C) RISK: Larger drawings will subsequently have more revisions and entire document will have to be revised instead of having individual drawings that can be redlined.	Schedule / Cost / Quality	5	3	15	Switch to Mono-Detail Drawing Method	Project Manager	Yes	2	3	6
			91D	RISK	91D) RISK: Single simplified drawing could have less detail per component.	Scope / Cost / Quality	4	3	12	Switch to Mono-Detail Drawing Method	Project Manager	Yes	2	3	6
9.2	No 3D Model Exists	Lack of 3D model leads to lack of repeatability in future even if photo is taken, less robust of a design with possible safety ramifications with a likelihood of missing critical interfaces and no 3D model for the stress group and drafter will not be able to utilize annotation feature.	92A	RISK	92A) RISK: Lack of repeatability for future rigs lost, increasing costs and schedule for future rigs projects.	Schedule / Scope / Cost	4	3	12	Switch to Mono-Detail Drawing Method	Project Manager	Yes	2	3	6
			92B	RISK	92B) RISK: Missing interfaces lead to redesigns and loss of quality.	Schedule / Quality	4	3	12	Switch to Mono-Detail Drawing Method	Project Manager	Yes	2	3	6
			92C	RISK	92C) RISK: No model for Stress Analysis Group to analyze.	Schedule / Quality	5	3	15	Switch to Mono-Detail Drawing Method	Project Manager	Yes	1	3	3
			92D	RISK	92D) RISK: Annotation feature lost from 3D modeling package.	Schedule / Quality / Cost	5	3	15	Switch to Mono-Detail Drawing Method	Project Manager	Yes	1	3	3
			92E	RISK	92E) RISK: Safety concerns with a less robust design due to 2D CAD.	Quality	2	2	4	None	Project Manager	No	2	2	4
			92F	RISK	92F) RISK: Loss of Configuration Control for GSE	Schedule / Cost / Quality	4	4	16	Switch to Mono-Detail Drawing Method	Project Manager	Yes	1	4	4
9.3	Complexity for Final Release Signatories	Signatories may require detailed drawings to understand and analyze the hardware effectively. In a single drawing, there is a likelihood that finer details will be missed.	93A	RISK	93A) RISK: Missing details due to single simplified drawing method could have less dimensions, less details, etc.	Scope / Schedule / Quality	4	3	12	Switch to Mono-Detail Drawing Method	Project Manager	Yes	2	3	6
9.4	Complexity for Vendor	A single simplified drawing will have all components included and this could be difficult when needing to send out for procurements and could possibly introduce additional NDAs or Export Control issues.	94A	RISK	94A) RISK: Complexity of drawing could lead to communication issues with vendor/procurement.	Schedule / Cost	4	2	8	Switch to Mono-Detail Drawing Method	Project Manager	Yes	2	2	4
			94B	RISK	94B) RISK: NDA or Export Control issues associated with complex BOM, not limited to Jacobs, cross contamination along multiple vendors, loss of intellectual property and "Blue-Line Process" application.	Schedule / Cost	3	4	12	Switch to Mono-Detail Drawing Method	Project Manager	Yes	1	4	4
9.5	Compatibility for rig builds	One drawing may lead to complications with paperwork (DRs, TPs) and drawing interpretation for sub-assemblies.	95A	RISK	95A) RISK: Complications for DRs and TPs.	Schedule / Quality / Cost	4	2	8	Switch to Mono-Detail Drawing Method	Project Manager	Yes	2	2	4
			95B	RISK	95B) RISK: Drawing interpretations for sub-assemblies.	Schedule / Cost	4	2	8	Switch to Mono-Detail Drawing	Project Manager	Yes	2	2	4

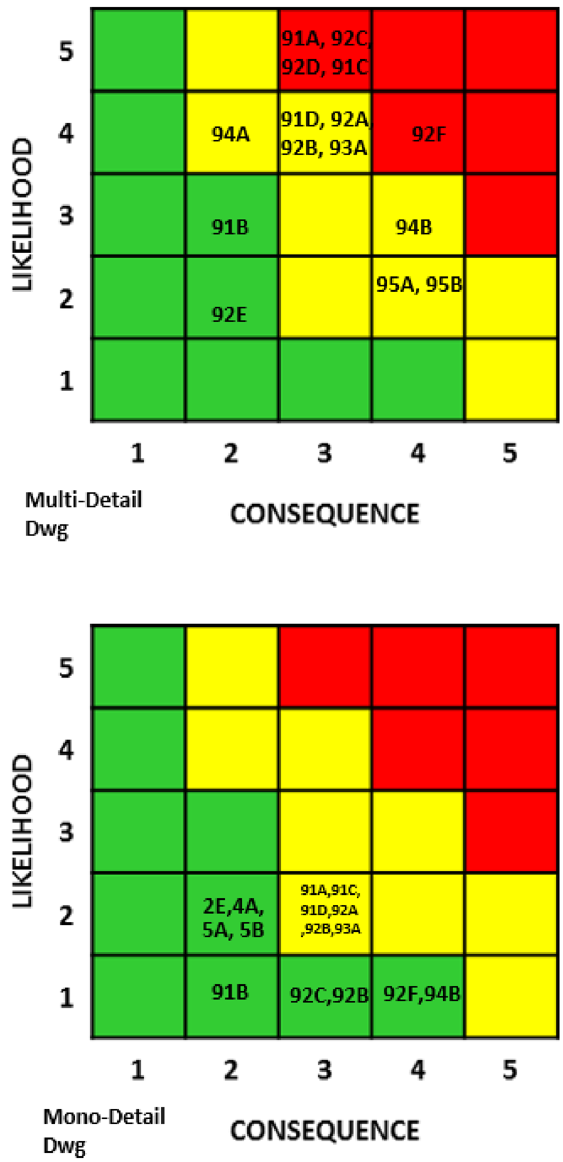


Figure 46 - Before (Top) and After (Bottom) FMEA RPN Modifications for Drawing Approach

The sample template of the FMEA is shown below for Item #4 and Item #8 from the prompt list with graphs included to show how a threat can increase cost and schedule while an opportunity may marginally increase safety and quality within an acceptable margin while saving tremendously on cost and schedule.

Table 58 - Lean FMEA Sample: Procurement & Drafting Time Approach

PRELIMINARY INFORMATION							RPN INPUTS			POST RPN EVALUATION			POST RPN		
Risk #	Name	Description	ID#	Risk/Opportunity	Itemized From Description	Impact Areas?	CONSEQUENCE (1-5)	LIKELIHOOD (1 - 5)	RPN (1 - 25)(BEFORE ANALYSIS)	Action Recommended	Responsible Party	Actions Taken	CONSEQUENCE (1-5)	LIKELIHOOD (1 - 5)	RPN (1 - 25)(AFTER ANALYSIS)
What is the risk/opportunity number associated with the process?	What is the name of the risk/opportunity associated with the process?	What the risk associated with the process?	What's the identification?	Is this a risk or opportunity for the project?	How would these risks/opportunities decompose from the parent risk listed in the description?	Safety? Schedule? Quality? Cost? Scope?				What are the possible actions to remedy the potential risk or exploit opportunity?	Who is responsible for making sure the actions are completed?	Will the Action Recommended by taken with respect to RPN?			
4	GSE Non-Critical Hardware Procurement Purchased as Strict GSE	Procurements for GSE hardware are prolonged in terms of schedule and more expensive when compared to their Class III or Class I-E counterparts if they are bought strictly as GSE. Column D assumes if purchased as strict GSE, Column I assumes bought as Class III, Non-Critical GSE or Upgraded.	4A	RISK	4A) RISK: Extended time in finding vendors: With GSE, purchasing will spend more time securing a vendor that can provide the necessary paperwork (i.e., CofCs, Q-Codes, traceability, etc.)	Schedule	5	3	15	Procure as GSE Non-Critical or Class III and Upgrade	Project Manager, Design Lead.	Yes	2	3	6
			4B	RISK	4B) RISK: Difficulty obtaining Q-codes: Even if the vendor is secured, the Q-Codes (quality codes) historically are not guaranteed to be met. This delays schedule.	Schedule	5	4	20	Procure as GSE Non-Critical or Class III and Upgrade	Project Manager, Design Lead.	Yes	1	2	2
			4C	RISK	4C) RISK: Higher costs in procuring as GSE: This is twofold (1/2). Even if the vendor is secured, the Q-Codes (quality codes) historically are not guaranteed to be met. This increases costs to find new vendors.	Cost	3	5	15	Procure as GSE Non-Critical or Class III and Upgrade	Project Manager, Design Lead.	Yes	2	3	6
			4D	RISK	4D) RISK: Higher costs in procuring as GSE: This is twofold (2/2). Producing Q-codes increases costs from the vendor.	Cost	3	5	15	Procure as GSE Non-Critical or Class III and Upgrade	Project Manager, Design Lead.	Yes	2	3	6
			4E	OPPORTUNITY	4E) OPPORTUNITY: While not all items need to be procured with the same pedigree (i.e., Q-codes, CofCs, traceability), many do not. As such, quality can still be maintained and still delivery a sound product.	Quality	1	1	1	Procure as GSE Non-Critical or Class III and Upgrade	Project Manager, Design Lead.	Yes	2	2	4
			4F	OPPORTUNITY	4F) OPPORTUNITY: While not all items need to be procured with the same pedigree (i.e., Q-codes, CofCs, traceability), many do not. As such, safety can still be maintained and still delivery a sound product.	Safety	1	1	1	Procure as GSE Non-Critical or Class III and Upgrade	Project Manager, Design Lead.	Yes	2	2	4
8	Drafting Time	As a facet of drawing review processes, JETS maintains a separate drafting department that will dedicate drafters to GSE work to preserve quality and safety of the end product delivered the customer in the form of robustness in drawing reviews. The GSE team believes that by running a Scrum Model, teams are self empowered and can check the work locally via pair programming. GSE team thinks that the 15 day normal period can be reduced to 1-3 days.	8A	RISK	8A) RISK: With drafting work sent to external functional groups, schedule will be extended due to signatories working in external groups that may not be dedicated to the project's schedule.	Schedule	4	5	20	Allow for project to review its own drawings in most cases.	Project Manager, Test Lead, Design Lead.	Yes	2	2	4
			8B	RISK	8B) RISK: With drafting work sent to external functional groups, schedule will be extended due to drafters working in external groups that may not be dedicated to the project's schedule - this will affect the project's budget.	Budget	4	5	20	Allow for project to review its own drawings in most cases.	Project Manager, Test Lead, Design Lead.	Yes	2	2	4
			8C	OPPORTUNITY	8C) OPPORTUNITY: There is an opportunity to save on schedule and budget, while preserving scope by allowing engineers to act in a drafting capacity by focusing on pair programming to check and double check work.	Scope	2	1	2	Allow for project to review its own drawings in most cases.	Project Manager, Test Lead, Design Lead.	Yes	2	1	2
			8D	OPPORTUNITY	8D) OPPORTUNITY: There is an opportunity to save on schedule and budget, while preserving safety. There is an opportunity to save on schedule and budget, while preserving safety by allowing engineers to act in a drafting capacity by focusing on pair programming to check and double check work.	Safety	2	1	2	Allow for project to review its own drawings in most cases.	Project Manager, Test Lead, Design Lead.	Yes	2	1	2
			8E	OPPORTUNITY	8E) OPPORTUNITY: There is an opportunity to save on schedule and budget, while preserving quality. There is an opportunity to save on schedule and budget, while preserving quality by allowing engineers to act in a drafting capacity by focusing on pair programming to check and double check work.	Quality	2	1	2	Allow for project to review its own drawings in most cases.	Project Manager	Yes	2	1	2

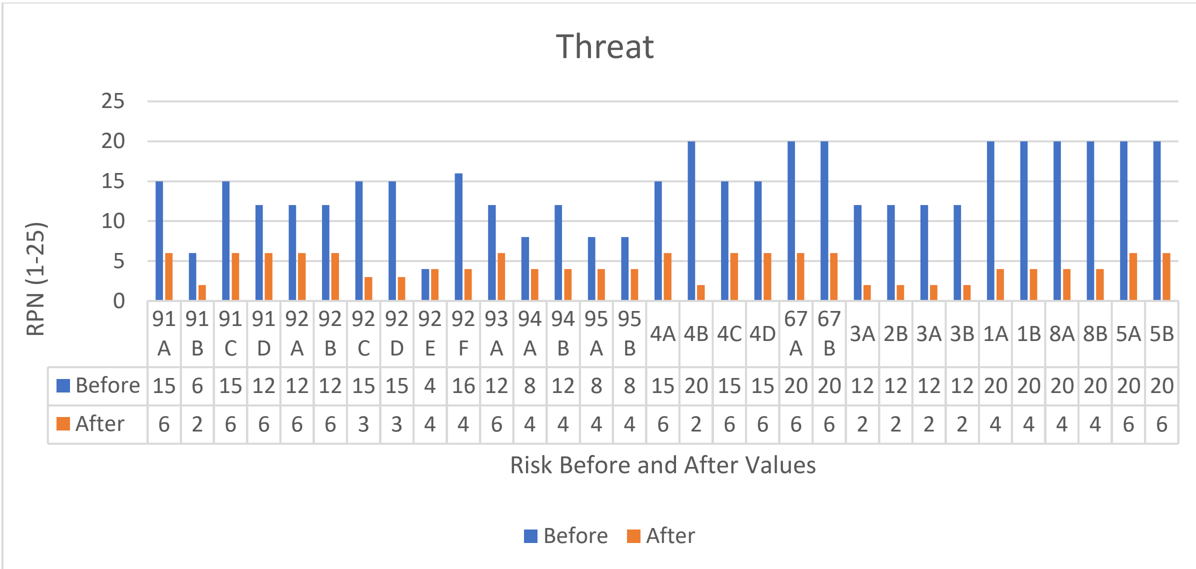


Figure 47 – Lean FMEA Before and After for Drawing Approach

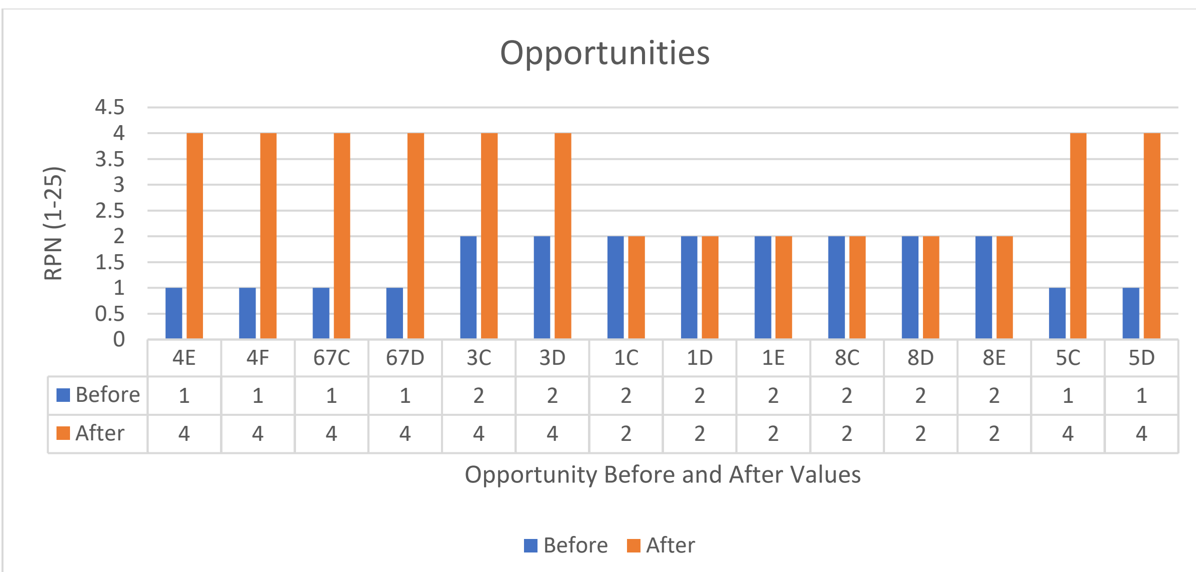


Figure 48 - Lean FMEA Before and After for Procurement and Drafting Approach

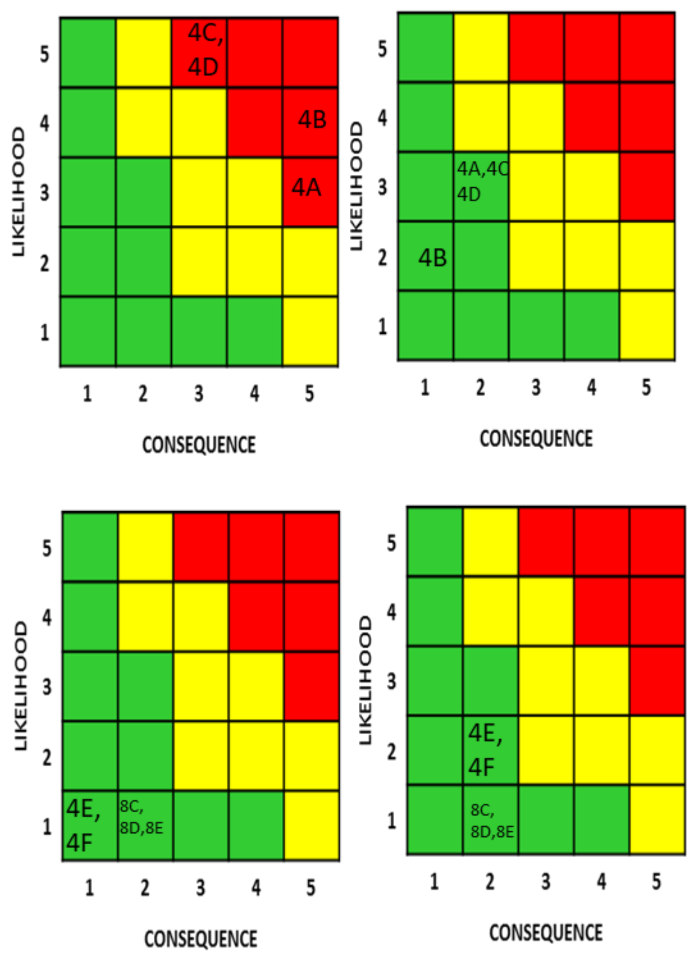


Figure 49 - Before & After FMEA RPN Modifications for Procurement & Drafting Approach

Step 4: Incorporate Updated Schedule Times into Project Schedule

Assuming all modifications could be made to the schedule based on the aforementioned 7 Forms of Lean Waste identifications and mitigations, the project schedule was reduced from 451 days to 335 days with no contingency and 375 days with risk contingency. The final schedule will be presented in Iteration #3 after full agile methods have been applied.

Step 5: Improvements & Limitations

Each iteration will present a series of improvements and limitations. For Iteration 2, improvements to model allows for the 7 Forms of Lean Waste mitigation to improve schedule velocity, budget control and guarantee of quality and safety risks. Limitations include many of the improvements instigated yet derives a means by which to achieve them. These include:

- How to develop hardware with Scrum: This is a software-intensive process as software can change and iterate much more quickly and cheaply as opposed to hardware. These challenges include a series of constraints of physicality in purchasing, modification and changing of hardware vs. software.
  - Goal: successfully provide Scrum framework tailoring to optimize hardware development teams.
- The approach that must be taken to organize work priorities, workflow and characterization of true schedule: while a template is given to organize work, this varies greatly by how hardware vs. software is built. While priorities may seem apparent at the onset, certain aspects of hardware development cannot quickly and easily be incremented and implemented and as a result, backlog organization, workflow and characterization of true schedule remain challenges.
  - Goal: characterize workflow, priority organization and schedule prediction by developing a method to illustrate, organize and improve schedule performance.
- Challenges associated with cross functional development: While it is attractive to have a team of generalists, how to train and cross-develop these individuals is a challenge in itself before the team can be considered M-Shaped vs. I-Shaped.
  - Goal: find a method to cross-train all various disciplines of the development team.
- How to develop to identify and optimize impediment improvement: A ScrumMaster will

allow for the first step of address impediments, especially those seen terms of process, but how to implement them methodically and identify and address is still unaddressed.

  - Goal: how to methodically identify, address and remove impediments by means of a ScrumMaster.
  - Mitigation: Utilization of the 7 Forms of Lean Waste FMEA tool to help quantify risks of Lean Waste and bring to the attention of upper management in an attempt to remove impediments.

- How to integrate if required with waterfall: At the conclusion of the iteration and before the onset of Iteration 3, it is evident that a standalone Scrum model may not be sufficient to address not only the needs of the performing organization but also its compatibility.
  - Goal: (if required) find a means to combine best practices from both methods while improving or eliminating the shortcomings (i.e., inability for waterfall to quickly address change, challenge for Scrum to be implemented with hardware teams).
- How to work with a remote working team: As complications from COVID intermingled with the challenges already described, Scrum promotes itself by working with a centralized team. How to address this with the advent of COVID cannot be understated.
  - Goal: how to fully integrate a remotely working team unable to learn by osmosis in an office environment.
- How to obtain acceptance by customers, development team and performing organization: Maybe the most significant challenge would be acceptance from customer and performing organization. But first, acceptance by the develop team needs to be harnessed.

- Goal: creating a process by which the development team and subsequently customer and performing organization can vet via survey to gain acceptance of tempered model.

Step 6: Overall Summary of Iteration

An abridged table is given to reflect the aforementioned iteration number, time duration, data gathering techniques, improvements and limitations.

Table 59 – Iteration 2A Summary

Iteration 2	Time Period	Data Gathering	Improvements	Limitations	Limitations Addressed
Scrum Team Developed	February 2022 - May 2022	Intermediate Literature Review	Allows for Incremental and Deliveries	How To Develop Hardware with Scrum	
		Process Investigation	Allows for Iterative Deliveries	~~How To Organize Work Priorities~~	Approach Functional Managers About Decreasing Multi-Tasking
		Brainstorming	Allow for Priorities to be Organized Periodically	How To Demonstrate Workflow	
		Interviews	Utilizes a Burndown Chart for Schedule Forecast	How To Characterize True Schedule	
		Focus Groups	Promotes Generalists over Specialists	How To Create Cross-Functional Generalists	
		Questionnaires	Appoints a Servant Leader in a ScrumMaster	~~How To Develop Impediment Improvements~~	Lean Addresses Ability to Help Remove Impediments
Lean Introduced		2nd Survey	Allows for Consistent Customer Engagement	How To Integrate with Traditional Waterfall	
			Promotes Learning by Osmosis	How To Utilize with a Remotely Working Team	
			Identification of Overprocess (Lean)	~~How To Address Processes That Don't Add Value~~	Lean can be used to address possible processes that may not add value
			Identification of Hand-Off (Lean)		Lean can be used to address dependent areas where time can be minimized
			Identification of Waiting (Lean)		Lean can be addressed to address processes that can have wait times reduced
			Identification of Tasking Switching (Lean)		Lean can be used to address Functional Managers About Decreasing Multi-Tasking

8.7. ITERATION 3A: Project Lifecycle Development

8.7.1. Iteration Development

In the third iteration, the Scrum model is modified to be partially followed up until the sprint method is no longer viable. This is a function of internally controlled processes that can be guided by the ScrumMaster to operate as Scrum and later processes (mostly externally imposed like operations) to follow the traditional, waterfall methods. XP and FDD implements are included to allow for pair programming for cross training and developing user stories and stories pointing, respectively. Kanban is introduced to visualize workflow and set work in process limits. In addition, schedule velocity methods are harnessed as a function of wave rolling planning and a visual template on Scrum and user's manual provided to the team. The period of performance of Iteration 3 focused between May 2022 and October 2022. The data gathering techniques included:

- Final Literature Review: Where Kanban concepts from academic and peer reviewed journals are viewed.
- Interviews: Various, semi-formal sessions were elicited with stakeholders from across the project to assess which areas of systems engineering presented challenges and for project lifecycle development, those specific challenges documented.
- Focus Groups: Various, semi-formal sessions were elicited with stakeholders from across the project to assess which areas of systems engineering presented challenges and for project lifecycle development, those specific challenges documented.
- Questionnaires: Various, semi-formal sessions were elicited with stakeholders from across the project to assess which areas of systems engineering presented challenges and for project lifecycle development, those specific challenges documented.
- 3rd Survey: A Likert scale survey to assess the current vs. desired state of Kanban, XP and FDD offerings on the project with responses from the Development Team and customer.

The final framework includes all previous iteration inclusions with the exception of addition of Kanban to help visualize workflow and control WIP, especially in the drawing review/release process. The final framework also preserves sprints to areas they are most applicable, during the Kick-Off, Preliminary and Detailed Design Phases (i.e., Pre-Fabrication); a conversion to waterfall is preserved for more established operations. Sprints will conclude with burndown chart analysis and future sprints wave roll planned to hypothetically optimize schedule velocity. Implements from XP and FDD allow for pair programming to help cross train engineers and the decomposition of large portions of work into more manageable work packages, respectively. This approach is the hypothesized optimized GSE Test Station Team development known as the MAC.

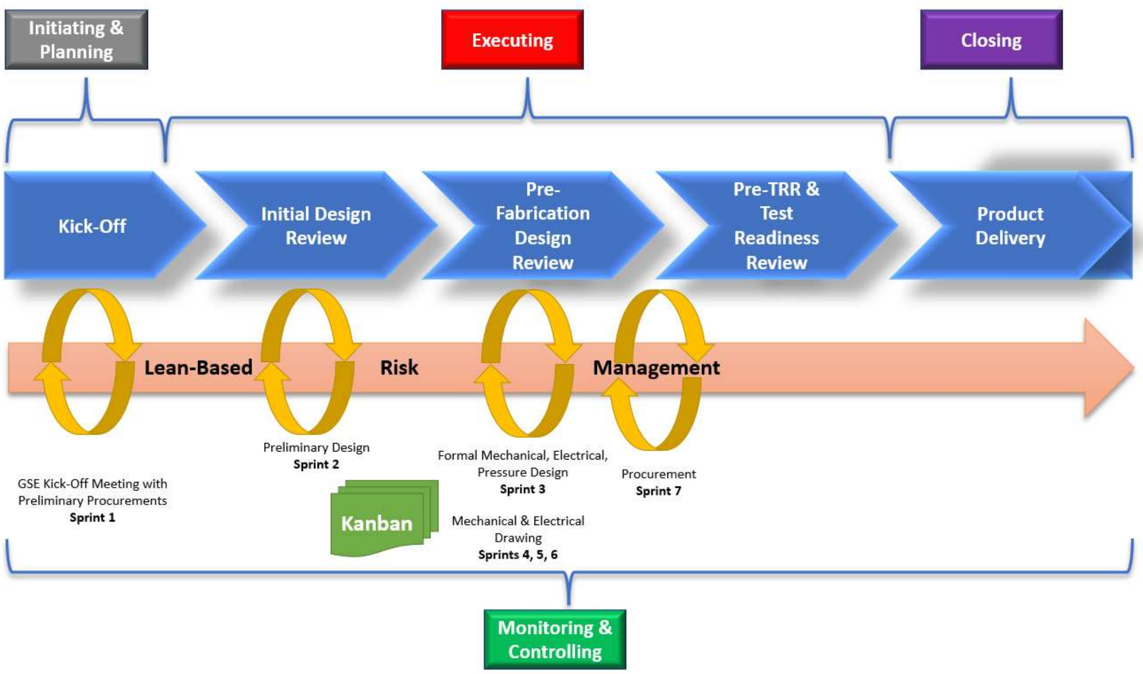


Figure 50 – Iteration 3A Lifecycle Approach

A survey with 24 individuals across the project (customer and performing organization) were polled on the Likert scale to elicit information regarding XP and FDD and the desired vs. current delivery method that project had offered Scrum-related artifacts; 1 – Strongly

Disagree, 5 – Strongly Agree; any score above a 3.5 (median between “Neither Agree or Disagree and Somewhat Agree) is deemed a positive indication on position; any score below a 3 is considered a negative indication on position. The survey below was used to compare the desired vs. current XP and FDD offering on project to examine how malleable the team would be to a change in lifecycle development. Results indicate that the current waterfall development has not offered a similar XP or FDD artifact and that the team would prefer to operate in one of these agile approaches in some capacity. Certain areas will either be exploited or avoided depending on the delta between desired and current project delivery methods.

Table 60 – XP and FDD Current vs. Proposed Likert Questions

5 - Extreme Programming General Questions			
#	Field	Mean	Count
1	I would like to work in pairs (i.e., pair programming)	3.3	23
2	I prefer writing tests before designing equipment	2.58	24
3	I prefer to keep designs simple	4.46	24
4	I like the idea of different functional groups convening to uncover design issues	4.54	24
5	I like the idea of smaller portions of work delivered more frequently over larger portions of work delivered less frequently	4.46	24
6	Aside from a traditional manager, I like the idea of a mentor to facilitate current processes	4.29	24
7	Aside from a traditional manager, I like the idea of a mentor to facilitate process improvement	4.29	24
8	Aside from a traditional manager, I like the idea of a mentor to facilitate impediment removal	4.17	24
9	I like the idea of the customer being an integral part of the development team weekly	4	23
10	I prefer to work at sustainable pace	4.63	24
11	I like the idea of learning more engineering skills outside of my current skillset	4.58	24

#	Desired Delivery Method	Current Delivery Method	Delta	Exploit Approach	Avoid Approach
1	3.3	2.63	0.67		
2	2.58	2.55	0.03		X
3	4.46	3.13	1.33	X	
4	4.54	3.83	0.71	X	
5	4.46	2.96	1.5	X	
6	4.29	3.29	1	X	
7	4.29	3.04	1.25	X	
8	4.17	2.63	1.54		
9	4	3.05	0.95		
10	4.63	3.57	1.06	X	
11	4.58	4	0.58	X	

#	6 - Extreme Programming Current Project Position	Mean	Count
1	We work in pairs (i.e., pair programming)	2.63	24
2	We write tests before designing equipment	2.55	22
3	We keep designs simple	3.13	23
4	On the project, different functional groups convene to uncover design issues	3.83	24
5	On the project, we deliver smaller portions of work more frequently instead of larger portions of work delivered less frequently	2.96	23
6	Aside from a traditional manager, we have a mentor to facilitate current processes	3.29	24
7	Aside from a traditional manager, we have a mentor to facilitate impediment removal	3.04	24
8	Aside from a traditional manager, we have a mentor to facilitate process improvement	2.63	24
9	The customer works as part of the development team weekly	3.05	19
10	We work at sustainable pace	3.57	23
11	Functional managers encourage us to learn engineering skills outside of our current skillset	4	23

11 - Feature Driven Development General Questions			
#	Field	Mean	Count
1	I prefer breaking down work into chunks over managing large work packages	4.17	24
2	I prefer to do inspections as frequently as possible	3.67	24
3	I prefer to have a visible backlog of chucks of work over a traditional schedule	3.54	24
4	Writing tests before designing hardware promotes defect detection	3.5	22

#	Desired Delivery Method	Current Delivery Method	Delta	Exploit Approach	Avoid Approach
1	4.17	3.26	0.91	X	
2	3.67	2.86	0.81		
3	3.54	3.58	-0.04		
4	3.5	2.38	1.12		

12 - Feature Driven Development Current Project Position			
#	Field	Mean	Count
1	We break down work into chunks instead of managing large packages of work	3.26	23
2	We do inspections as frequently as possible	2.86	22
3	We have a visualization to track progress on completed work over a traditional schedule	3.58	24
4	We adequately detect defects by writing tests before designing equipment	2.38	21

A survey with 24 individuals across the project (customer and performing organization) were polled on the Likert scale to elicit information regarding Kanban and the desired vs. current delivery method that project had offered Scrum-related artifacts; 1 – Strongly Disagree, 5 – Strongly Agree; any score above a 3.5 (median between “Neither Agree or Disagree and Somewhat Agree) is deemed a positive indication on position; any score below a 3 is considered a negative indication on position. The survey below was used to compare the desired vs. current Kanban offering on project to examine how malleable the team would be to a change in lifecycle development. Results indicate that the current waterfall development has not offered a similar Kanban artifact and that the team would prefer to operate in this approach in some capacity. Certain areas will either be exploited or avoided depending on the delta between desired and current project delivery methods.

Table 61 – Kanban Current vs. Proposed Likert Questions

#	Field	Mean	Count
1	I prefer a visualization of workflow in order to track tasks (i.e., value map streaming)	4.17	24
2	I prefer in restricting the amount of work in progress	3.7	23
3	I prefer low-tech tools (Kanban/Burndown Charts) vs. high-tech (i.e., MS Project)tools for tracking project development	3.38	24
#	**Field**	**Mean**	**Count**
1	We use a visualization of workflow in order to track tasks (i.e., value map streaming)	3.58	24
2	We practice restricting the amount of work in progress	2.88	24
3	We use low-tech tools (Kanban/Burndown Charts) vs. high-tech (i.e., MS Project) tools for tracking project development	2.25	24

#	Desired Delivery Method	Current Delivery Method	Delta	Exploit Approach	Avoid Approach
1	4.17	3.58	0.59	**X**	
2	3.7	2.88	0.82		
3	3.38	2.25	1.13	**X**	

Step 1: Modify Schedule

The final schedule in iteration 3 was reduced from 13 sprint and 451 days of schedule from the first iteration to 7 sprints and 335 days of schedule. The most critical reduction in schedule was the Lean mitigation in the form of the modified FMEA. The full backlog from the first iteration is still present. The Kick-Off, Initial Design Review, and Detailed Design Phases (i.e., Pre-Fabrication) Reviews while Pre-TRR, TRR and Delivery Phases are followed with Waterfall.

Table 62 – Iteration 3A Schedule

Task Name	Duration	Start	Finish
Test Station SIPPE GSE Schedule	67.4 wks	Wed 10/30/19	Thu 2/11/21
Detail Development Schedule	67.4 wks	Wed 10/30/19	Thu 2/11/21
GSE Kick-Off Meeting (Milestone) (Sprint 1)	7.8 wks	Mon 12/2/19	Thu 1/23/20

Initial Design Review Meeting (Milestone) (Sprint 2)	4.4 wks	Fri 1/24/20	Mon 2/24/20
Complete Preliminary Designs (Sprint 2)	4.4 wks	Fri 1/24/20	Mon 2/24/20
Sprint Planning Meeting 2 (All Sprints Concurrent)	0.2 wks	Fri 1/24/20	Fri 1/24/20
Preliminary Mechanical Design 2A	3 wks	Fri 1/24/20	Thu 2/13/20
Preliminary Electrical Design 2B	2 wks	Fri 1/24/20	Thu 2/6/20
Preliminary Software Design 2C	3.6 wks	Fri 1/24/20	Tue 2/18/20
Hazard Identification 2D	3 wks	Fri 1/24/20	Thu 2/13/20
Sprint Review	0.6 wks	Wed 2/19/20	Fri 2/21/20
Sprint Retrospective	0.2 wks	Mon 2/24/20	Mon 2/24/20
Detail Design and Drafting	21.2 wks	Tue 2/25/20	Tue 7/21/20
Create Formal Mechanical Design	5 wks	Tue 2/25/20	Mon 3/30/20
Complete Intermediate & Formal Mechanical/Electrical/Pressure Designs (Sprint 3)	5 wks	Tue 2/25/20	Mon 3/30/20
Sprint Planning Meeting 3 (All Sprints Concurrent)	0.2 wks	Tue 2/25/20	Tue 2/25/20
Complete ME Intermediate Sprint 3A	2.4 wks	Wed 2/26/20	Thu 3/12/20
Sprint Review	0.6 wks	Wed 3/25/20	Fri 3/27/20
Sprint Retrospective	0.2 wks	Mon 3/30/20	Mon 3/30/20
Create Drawings and Fab Release	16.2 wks	Tue 3/31/20	Tue 7/21/20
First Round ME/EE Drawings Sprint 4	5.4 wks	Tue 3/31/20	Wed 5/6/20
Sprint Planning Meeting 4 (All Sprints Concurrent)	0.2 wks	Tue 3/31/20	Tue 3/31/20
SIPPE Test Rig Rack 1 (Sprint 4A)	4.4 wks	Wed 4/1/20	Thu 4/30/20
Sprint Review	0.6 wks	Fri 5/1/20	Tue 5/5/20
Sprint Retrospective	0.2 wks	Wed 5/6/20	Wed 5/6/20
Second Round ME/EE Drawings Sprint 5	5.4 wks	Thu 5/7/20	Fri 6/12/20
Sprint Planning Meeting 5 (All Sprints Concurrent)	0.2 wks	Thu 5/7/20	Thu 5/7/20
SIPPE Test Rig Rack 2 Sprint 5A	4.4 wks	Fri 5/8/20	Mon 6/8/20
Sprint Review	0.6 wks	Tue 6/9/20	Thu 6/11/20
Sprint Retrospective	0.2 wks	Fri 6/12/20	Fri 6/12/20
Third Round ME/EE Drawings Sprint 6	5.4 wks	Mon 6/15/20	Tue 7/21/20
Sprint Planning Meeting 6 (All Sprints Concurrent)	0.2 wks	Mon 6/15/20	Mon 6/15/20
SIPPE Test Rig Rack 3 (Sprint 6)	4.4 wks	Tue 6/16/20	Wed 7/15/20
SIPPE Non-Exclusive Rack Items (Sprint 6)	4.4 wks	Tue 6/16/20	Wed 7/15/20
Sprint Review	0.6 wks	Thu 7/16/20	Mon 7/20/20
Sprint Retrospective	0.2 wks	Tue 7/21/20	Tue 7/21/20
Create Formal Electrical Design	9.4 wks	Wed 2/26/20	Thu 4/30/20
Complete Top Level System Block Design Sprint 3B	**2 wks**	**Wed 2/26/20**	**Tue 3/10/20**
Complete DAQ Block Design Sprint 3C	**2 wks**	**Wed 2/26/20**	**Tue 3/10/20**
Complete Safety Box Block Design Sprint 3D	**2 wks**	**Wed 2/26/20**	**Tue 3/10/20**
Complete Derating Analysis Sprint 3E	**2 wks**	**Wed 2/26/20**	**Tue 3/10/20**
Create Formal Pressure Systems Design Sprint 3F	**3 wks**	**Wed 2/26/20**	**Tue 3/17/20**
Complete Theory of Operations Sprint 3G	**4 wks**	**Wed 2/26/20**	**Tue 3/24/20**
Complete Electrical Interconnect Diagram (Complex) 4B	**4.4 wks**	**Wed 4/1/20**	**Thu 4/30/20**
Complete Harness Design (All Standard) 4C	**4.4 wks**	**Wed 4/1/20**	**Thu 4/30/20**
Complete Printed Circuit Board for 450 Interface 4D	**4.4 wks**	**Wed 4/1/20**	**Thu 4/30/20**
All Rig Drawings Fab Released (Deliverable)	0 days	Tue 7/21/20	Tue 7/21/20
Create Formal Software Design	25.2 wks	Wed 10/30/19	Wed 4/22/20
Initial SW Considerations Sprint 1C	1.2 wks	Wed 10/30/19	Wed 11/6/19
Configure New Modules (if non-existent in database 9 Modules) Sprint 2E	3 wks	Mon 1/27/20	Fri 2/14/20
Configure Automation Sprint 3H	3 wks	Wed 2/26/20	Tue 3/17/20
Generate User Interface Sprint 4E	4 wks	Wed 3/18/20	Tue 4/14/20
Documentation Sprint 4F	2.2 wks	Wed 4/8/20	Wed 4/22/20
Pre-Fab Design Review (Milestone) (Sprints 3 - 6)	1.4 wks	Wed 7/22/20	Thu 7/30/20
Procurement Sprint 7 (with Sprit 1 & 3 Portions)	61.2 wks	Wed 10/30/19	Wed 12/30/20
Sprint Planning Meeting 7	0.2 wks	Wed 7/22/20	Wed 7/22/20
COTS Instruments Sprint 7A (Sprint Completion = Ordering Only, Not Delivery)	23.2 wks	Tue 7/21/20	Wed 12/30/20
COTS Components Sprint 1D (Sprint Completion = Ordering Only, Not Delivery)	60 wks	Wed 10/30/19	Tue 12/22/20

COTS Boxes Sprint 3I (Sprint Completion = Ordering Only, Not Delivery)	27.8 wks	Tue 3/10/20	Mon 9/21/20
COTS Computer & IT Sprint 1E (Sprint Completion = Ordering Only, Not Delivery)	21.2 wks	Wed 10/30/19	Wed 3/25/20
Custom Parts Sprint 7B (Sprint Completion = Ordering Only, Not Delivery)	13.6 wks	Wed 7/22/20	Fri 10/23/20
Sprint Review	0.2 wks	Wed 8/12/20	Wed 8/12/20
Sprint Retrospective	0.2 wks	Thu 8/13/20	Thu 8/13/20
Fabrication/Assembly (Waterfall)	33.2 wks	Fri 5/1/20	Fri 12/18/20
Fabricate Top Level Mechanical Assembly	**8 wks**	**Mon 10/26/20**	**Fri 12/18/20**
Fabricate Rack #1	**8 wks**	**Mon 10/26/20**	**Fri 12/18/20**
Fabricate Rack #2	**8 wks**	**Mon 10/26/20**	**Fri 12/18/20**
Fabricate Rack #3	**8 wks**	**Mon 10/26/20**	**Fri 12/18/20**
Fabricate P&ID Assembly (Flow Loop)	**19.6 wks**	**Thu 7/23/20**	**Mon 12/7/20**
Fabricate EID (Electrical Loop)	**22.8 wks**	**Thu 6/11/20**	**Tue 11/17/20**
Fabricate DAQ Box Assembly	**4.4 wks**	**Tue 9/22/20**	**Wed 10/21/20**
Fabricate Safety Box Assembly	**4.4 wks**	**Tue 9/22/20**	**Wed 10/21/20**
Fabricate Harness Assembly Sprint 5B	**5.8 wks**	**Fri 5/1/20**	**Wed 6/10/20**
All Rig Drawings Final Released (Deliverable)	25 days	Mon 12/21/20	Fri 1/22/21
Documentation (Deliverable) (Waterfall)	39 wks	Wed 10/30/19	Tue 7/28/20
Operation Manual (Deliverable)	38 wks	Wed 10/30/19	Tue 7/21/20
Hazard Analysis (Deliverable)	38 wks	Wed 10/30/19	Tue 7/21/20
Test Procedure (Deliverable)	39 wks	Wed 10/30/19	Tue 7/28/20
Pre-TRR Review (Milestone) (Waterfall)	4 wks	Mon 12/21/20	Fri 1/15/21
Prepare Documents	1 wk	Mon 12/21/20	Fri 12/25/20
Review Documents (Jacobs internal)	1 wk	Mon 12/28/20	Fri 1/1/21
Hold Pre-TRR Review (Milestone)	0 days	Fri 1/1/21	Fri 1/1/21
Complete Pre-TRR	2 wks	Fri 1/1/21	Fri 1/15/21
Software Integration	1.4 wks	Thu 12/31/20	Fri 1/8/21
Finish Code Review	1.4 wks	Thu 12/31/20	Fri 1/8/21
Finalize Rig Assembly and Function	6.2 wks	Fri 12/18/20	Mon 2/1/21
TRR (Milestone) (Waterfall)	2.8 wks	Mon 1/18/21	Thu 2/4/21
Test Station (Deliverable)	28.4 wks	Wed 7/29/20	Thu 2/11/21

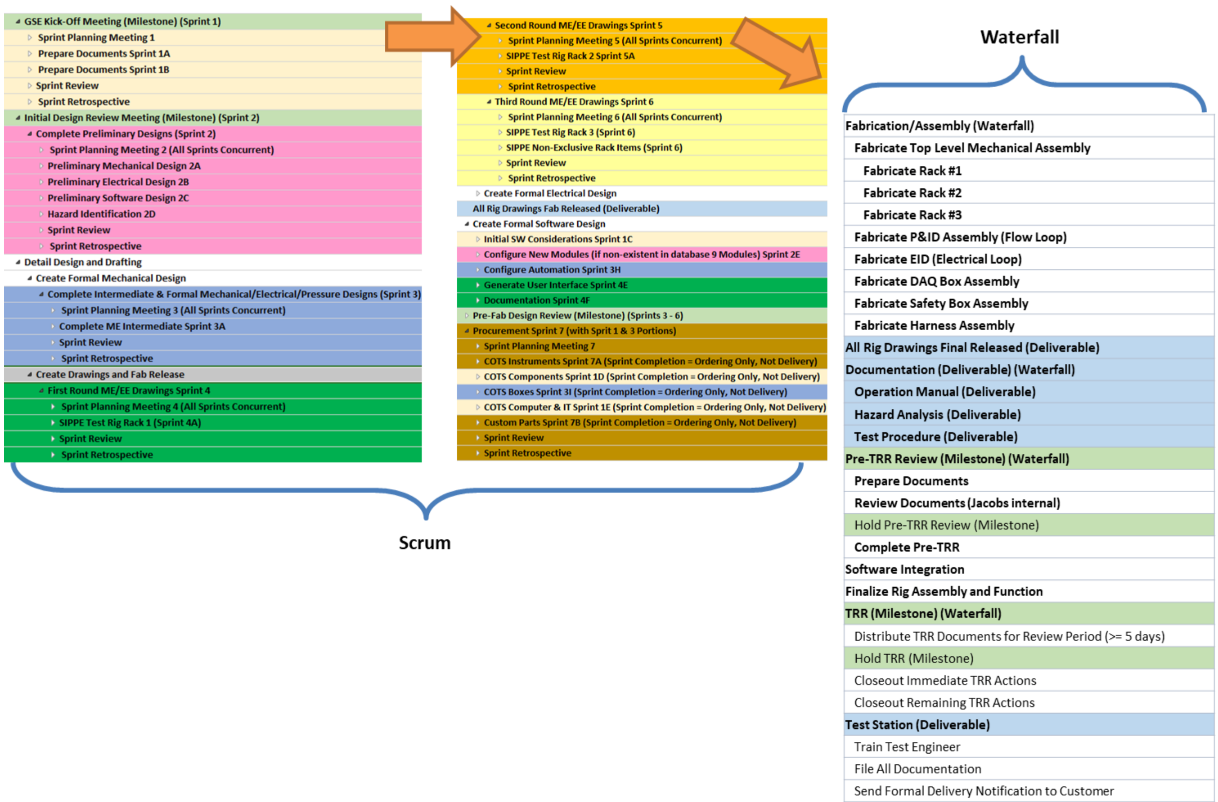


Figure 51 – Scrum vs. Waterfall in Iteration 3A Schedule

Step 2: Illustrate Scrum

The team requested that in addition to the Scrum-style schedule with explicit line items for each sprint (i.e., Sprint, Sprint Retrospective, Sprint Demo, etc.). This also demonstrates how FDD supports the creation of user stories, epics and story pointing while supporting the inclusion of pair programming for cross-disciplinary training (i.e., mechanical engineers reviewing electrical schematics, electrical engineers providing inputs to mechanical engineers for drawings). The diagram shown illustrates the first sprint with the traditional Scrum ceremonies but with the removal of the daily scrum, which by Likert scale polling was removed from schedule. A method by which Scrum benefits hardware aside from wave rolling planning via schedule velocity calculations and readjustments after each sprint is the working prototype which parallels the actual test station. While this is done in a minimalist capacity due to the complex nature of the hardware, early piping and instrumentation proof of concepts can be established using existing project hardware to gain customer acceptance early in the design process before the final design and procurement of actual, delivered hardware is completed.

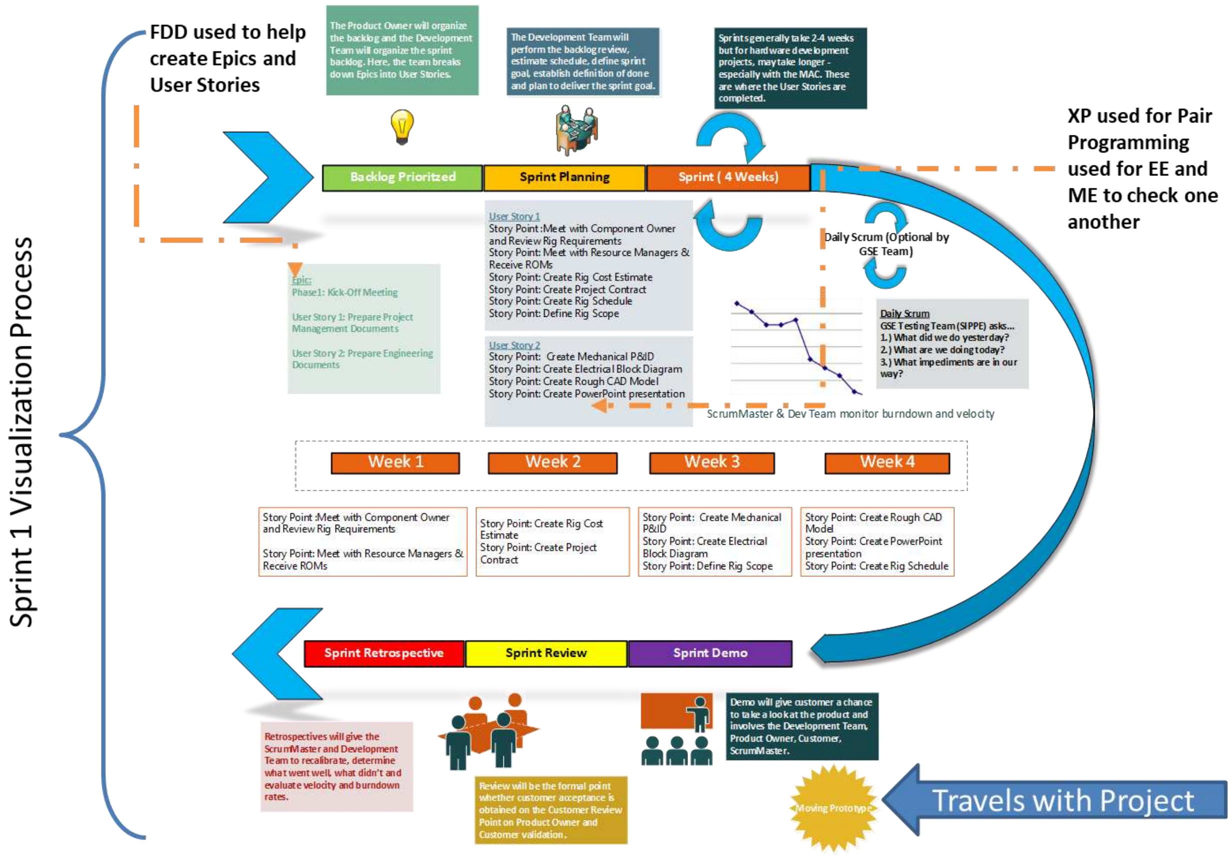


Figure 52 - Sprint 1 Conceptualized

For context and summary, the following is a list of all intended sprints:

- Sprint 1 Product Increment: Complete Project Contract, Initial Parts Ordered.
- Sprint 2 Product Increment: Working Fluid and Electrical Loop Prototypes, Host Initial Design Review.

- Sprint 3 Product Increment: EE Block Diagram, Pressure System Design, Theory of Constraints, Intermediate ME/EE CAD Design, LabVIEW Interface Ready & Intermediate Parts Ordered.
- Sprint 4: Product Increment: Round 1 ME Drawings, Interconnect Diagram, PCB Design, Harness, User Interface Done.
- Sprint 5 Product Increment: Round 2 ME Drawings.
- Sprint 6 Product Increment: Round 3 ME Drawings Done, Host Pre-Fabrication Design Review.
- Sprint 7 Product Increment: Final & Custom Parts Ordered.

Step 3: Establish Wave Roll Planning

The focus group helped established story pointing for projected task effort and once the work is completed (based on actual metrics from previous projects and estimation methods via Delphi Method on actual task effort as a project, partial or full, could not be supported by current funding) recorded the actual effort. The data between the projected vs. actual effort is used to inform future sprints. This effort is to be performed during the sprint retrospectives as a function of the ", what could we have done differently?" question prompted at the conclusion of a sprint. The first sprint it utilized as an example and assumes an 8-hour effort during each day to help complete the work packages. It is important to note that this is a unique example as each work package is similar in constitution (i.e., effort).

The first step was to organize the task effort by assigning story points in a manner in which a burn rate (i.e., a periodic measurement of task velocity to complete story points) could be established. In theory, this burn rate should be constant and follow, when possible, a linear progression so as to promote a methodical and efficient use of resource effort. The second step was to record the actual effort to demonstrate the reality of the effort that is performed, which in practice would not be perfectly linear. The third step was to reconcile the differences in an effort to re-estimate task efforts for future scheduling. This was done by establishing a burndown chart (i.e., a graphical depiction of projected vs. actual schedule velocity against story point completion over time) to characterize projected effort vs. actual effort and a recalculation of equivalent story points moving forward by dividing the original story point (i.e., the planned work EVM for task effort estimation) by the Schedule Performance Index (SPI).

Table 63 – Projected vs. Actual Effort in Story Pointing

				Week 1					Week 2					Week 3					
	Task Projected Effort (Burn Down Rate)	**Points**	Day 0	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Day 15	Day 16
Work Remaining	Meet with Component Owner and Review Rig Requirements	128	128	120	112	104	96	88	80	72	64	56	48	40	32	24	16	8	0
	Meet with Resource Managers & Receive ROMs	128	128	120	112	104	96	88	80	72	64	56	48	40	32	24	16	8	0
	Create Rig Cost Estimate	128	128	120	112	104	96	88	80	72	64	56	48	40	32	24	16	8	0
	Create Completion Form PD	128	128	120	112	104	96	88	80	72	64	56	48	40	32	24	16	8	0
	Create Rig Schedule	128	128	120	112	104	96	88	80	72	64	56	48	40	32	24	16	8	0
	Define Rig Scope	128	128	120	112	104	96	88	80	72	64	56	48	40	32	24	16	8	0
	Create Mechanical P&ID	128	128	120	112	104	96	88	80	72	64	56	48	40	32	24	16	8	0
	Create Electrical Block Diagram	128	128	120	112	104	96	88	80	72	64	56	48	40	32	24	16	8	0
	Create Rough CAD Model	128	128	120	112	104	96	88	80	72	64	56	48	40	32	24	16	8	0
	Create Powerpoint presentation	128	128	120	112	104	96	88	80	72	64	56	48	40	32	24	16	8	0

				Week 1					Week 2					Week 3					
	Task Actual Effort	**Points**	Day 0	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Day 15	Day 16
Work Remaining	Meet with Component Owner and Review Rig Requirements	128	128	120	112	104	96	88	80	72	64	56	48	40	32	24	16	8	0
	Meet with Resource Managers & Receive ROMs	128	128	121	111	101	99	92	87	81	75	65	55	45	42	34	24	9	0
	Create Rig Cost Estimate	128	128	120	112	104	79	74	69	59	49	39	29	19	9	0	0	0	0
	Create Completion Form PD	128	128	108	106	104	99	89	79	69	64	54	44	34	29	19	9	0	0
	Create Rig Schedule	128	128	120	112	104	96	88	80	72	64	56	48	40	32	24	16	8	0
	Define Rig Scope	128	128	127	117	107	99	91	83	75	74	64	54	44	34	24	19	9	0
	Create Mechanical P&ID	128	128	126	116	106	96	86	76	71	66	61	56	46	36	26	16	6	0
	Create Electrical Block Diagram	128	128	123	118	113	108	98	88	85	82	67	52	37	22	7	0	0	0
	Create Rough CAD Model	128	128	126	116	106	96	91	81	76	71	68	63	48	38	28	18	8	0
	Create Powerpoint presentation	128	128	127	126	125	124	123	122	121	120	119	84	34	0	0	0	0	0

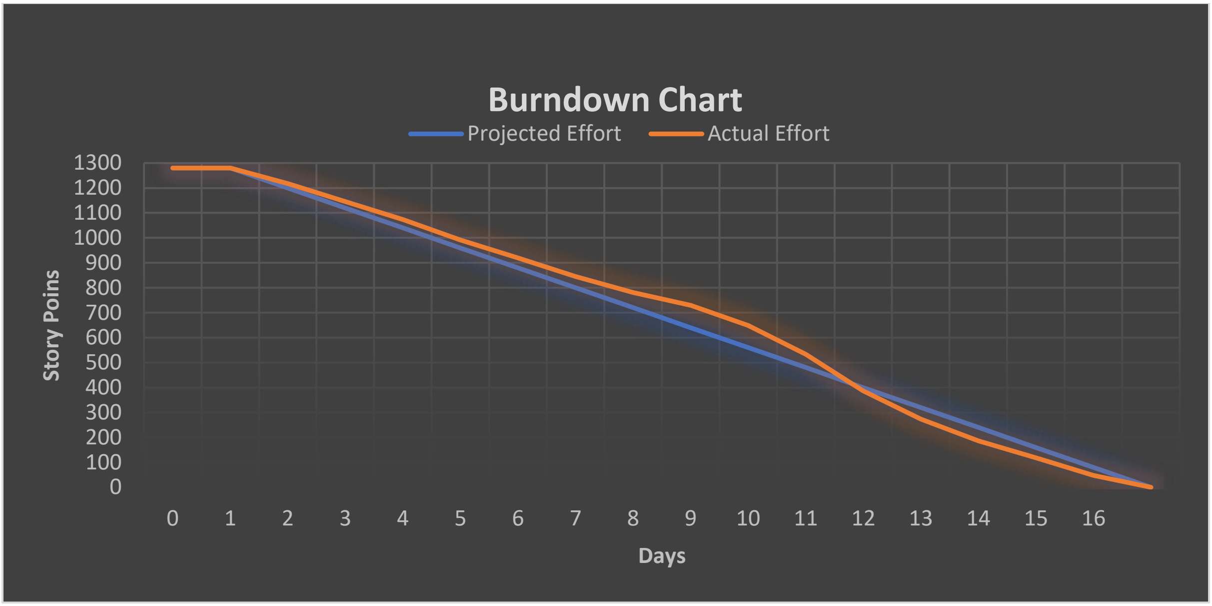


Figure 53 – Burndown Chart for Sprint 1

Table 64 – EVM-Inspired Wave Rolling Planner

Initial Parameters to Determine New Point Values				**Running Average New Point Value**
SPI	**Point Value**	**Itemized SPI Value**	**Group**	*Point Value / SPI*
1.07	128	Meet with Component Owner and Review Rig Requirements	Project Management	128
1.07	128	Meet with Resource Managers & Receive ROMs	Project Management	128
1.33	128	Create Rig Cost Estimate	Project Management	104
1.14	128	Create Contract	Project Management	120
1.07	128	Create Rig Schedule	Project Management	128
1.07	128	Define Rig Scope	Project Management	120
1.07	128	Create Mechanical P&ID	Mechanical	120
1.23	128	Create Electrical Block Diagram	Electrical	112
1.07	128	Create Rough CAD Model	Mechanical	128
1.45	128	Create PowerPoint presentation	Mechanical, Electrical	96

Step 4: Introduce Kanban

The Kanban philosophy was proposed by the GSE Testing Team with the idea that it could provide a flexible project management approach that promotes continuous collaboration and emphasizes visualization of work, typically through a collection of boards and cards. These tools that organize work are high-touch and low-tech and differ from traditional scheduling methods (i.e., Microsoft© Project).

The first step was to identify the areas across the project lifecycle development where Kanban could be implemented. The reasons may be as follows and were particularly effective for the GSE team:

- visualization of workflow tool for the Development Team to promote visibility of deliverables to help meet schedule and complete scope.
- a reporting tool for the Product Owner and ScrumMaster to evaluate schedule velocity and scope completion.
- a tool to reveal areas in which certain processes may suffer from one of the forms of Lean Waste.

A Kanban Board can be used in GSE development especially in:

- Detailed Design and Drawing phase by monitoring drawings through the drawing workflow process.
- fabrication of GSE hardware if and when TPSs and/or DRs are used.
- documentation such as procedures or hazard analyses that may need multiple signatory steps.

For reference, the GSE group found that using Kanban or a form of tracking tool worked especially effectively in the drawing and TPS/DR. The second step was the development of a template to allow for tracking of project artifacts with metrics available for tracking, visualization of workflow and schedule reporting capabilities. The template comes in the form of a tracker which allows for all distinct phases of the process in question to be illustrated, with percentages of work completed and remaining hours displayed. During GSE Detail Design and Development, drawing trackers were used for the mechanical and electrical engineers responsible for the development and release of drawings to visualize workflow and report metrics back to management, with the added benefit that the lead would know at all times where each drawing was in the drawing cycle. This also allowed for the development team to maintain ownership throughout the process, a facet of Scrum which encourages team ownership.

The GSE team agreed to use a drawing tracker and determined times adequate for one of the GSE projects via the Delphi Method with members of the Development Team. The Delphi Scored Duration template is not meant to be a standalone tool or indicative of any particular GSE build but a method by which each design phase (fabrication or final release for GSE builds in this example) has times and gates assigned with metrics in terms of hours, sequential order and responsible parties assigned. Another method would be a Kanban board, similar to the drawing tracker but with much less resolution. The Kanban board added benefit is that a WIP is added to each category which may also inform the tracker as to how many project artifacts can be in any particular category. Delphi scoring by project leads can be set, which also can influence how much throughput (in this case, number of drawings) can be completed in each sprint. If a Scrum of Scrum is employed and as each Development Team has a finite set of leads, this tool may also benefit several Scrum teams working in parallel that only have one lead which would present as the bottleneck.

Table 65 – Drawing Tracker

Drawing #	Drawing Description	Scheduled Completion	Initial Drawing Complete	Release	Current Stage
1	Schematic, Mechanical	6/26/2020	1/29/2021	2/5/2021	Initial Drawing (FAB)
2	Gas Piping Diagram - Schematic and Creo Piping	6/26/2020	1/29/2021	2/5/2021	Initial Drawing (FAB)
3	Water Piping Diagram - Schematic and Creo Piping	6/26/2020	1/29/2021	2/5/2021	Initial Drawing (FAB)
4	Fluid Panel Structure	N/A	1/13/2021	1/13/2021	EDCC Release (FAB)
5	Fluid Panel	6/26/2020	1/14/2021	1/14/2021	EDCC Release (FAB)
6	Fluid Panel Assembly	N/A	1/13/2021	1/15/2021	EDCC Release (FAB)
7	Rack, Modified, Instrument Altered Item Drawing Rev	N/A	1/22/2021	2/16/2021	Design Lead Review (FINAL)
8	Rack, Modified, Storage Altered Item Drawing Rev	N/A	1/22/2021	2/16/2021	Design Lead Review (FINAL)
9	Regulator and Solenoid panel	6/26/2020	1/22/2021	2/10/2021	EDCC Release (FAB)
10	Regulator and Solenoid Assembly	6/26/2020	1/22/2021	2/10/2021	EDCC Release (FAB)
11	Solenoid Bracket	6/27/2020	1/12/2021	2/9/2021	Redlines Incorporated (FAB)
12	Inlet Regulator Bracket	6/28/2020	1/21/2021	2/9/2021	CAD Conceptualization (FAB)
13	Plate, Platform, Top Rev	N/A	1/22/2021	2/12/2021	Design Lead Review (FINAL)
14	Plate, Platform, Top Assy Rev	N/A	1/22/2021	2/17/2021	Design Lead Review (FINAL)
15	Mechanical Assembly Rev B	N/A	1/28/2021	2/24/2021	Incorporate Redlines/Revisions (FINAL)
16	Master Assembly Rev B	N/A	2/11/2021	2/18/2021	Incorporate Redlines/Revisions (FINAL)
17	Pressure & Temperature Arrangement Drawing	TBD	2/1/2021	2/26/2021	CAD Conceptualization (FAB)
18	Drawing Tree	12/20/2020	2/1/2021	2/26/2021	Incorporate Redlines/Revisions (FINAL)

Drawing #	Stage Progress	Total Progress	Total Hours	Hours Left	Responsible Party	Comments
1	100%	54%	40	18	Resource 1	Need to be checked.
2	95%	53%	40	19	Resource 1	Date move to right. Target update before 1/29/21.
3	95%	53%	40	19	Resource 1	Date move to right. Target update before 1/29/21.
4	100%	100%	40	0	Resource 2	Fab Released 1/22/2021.
5	100%	100%	40	0	Resource 2	Fab Released 1/22/2021.
6	100%	100%	40	0	Resource 2	Fab Released 1/22/2021.
7	0%	17%	40	33	Resource 3	Sent to Lead.
8	0%	17%	40	33	Resource 3	Send to Lead to be checked.
9	100%	100%	40	0	Resource 3	Fab Released 1/21/2021.
10	100%	100%	40	0	Resource 3	Fab Released 1/21/2021.
11	0%	65%	40	14	Resource 4	New bracket (sheet metal) for solenoid.
12	80%	26%	40	30	Resource 4	New bracket (sheet metal) for gas regulator.
13	100%	25%	40	30	Resource 4	Update drawing an send to Lead for checking
14	100%	25%	40	30	Resource 4	Update drawing an send to Lead for checking
15	50%	13%	40	35	Resource 1	Lead to provide realines, if any.
16	20%	10%	40	36	Resource 2	Lead to provide realines, if any.
17	0%	0%	40	40	Resource 3	Need to determine the next level of assembly.
18	95%	16%	40	34	Resource 4	Revise and up to Rev A.

Table 66 - Drawing Tracker Delphi Method Planner

			Drawing Type					
			Simple	Standard	Complex	Standard		
Step	**FAB Release Stage**	**% Done**	*Duration (weeks)*			**% Complete**	**%Δ**	**Group**
0	CAD Conceptualization (FAB)	% Needed	2	3	4	0%	32%	*Project Team Independent*
1	Initial Drawing (FAB)	% Needed	1	2	3	32%	22%	*Project Team Independent*
2	Design Lead Review (FAB)	% Needed	0.5	1	2	54%	11%	*Project Team Independent*
3	Redlines Incorporated (FAB)	% Needed	0.5	1	2	65%	11%	*Project Team Independent*
4	Additional Reviewers (FAB)	% Needed	0.5	1	1	76%	11%	*Matrix Dependency*
5	Design Lead Backcheck (FAB)	% Needed	0.25	0.5	0.5	86%	5%	*Matrix Dependency*
6	PSRP (FAB)	% Needed	0.25	0.25	0.25	92%	3%	*Matrix Dependency*
7	EDCC Release (FAB)	% Needed	0.5	0.5	0.5	95%	5%	*Matrix Dependency*
						100%	-100%	
Step	**FINAL Release Stage**	**% Done**	*Duration (weeks)*			**% Complete**	**%Δ**	**Group**
0	CAD Revisions (FINAL)	% Needed	0.5	1	2	0%	8%	*Project Team Independent*
1	Incorporate Redlines/Revisions (FINAL)	% Needed	0.5	1	1.5	8%	8%	*Matrix Dependency*
2	Design Lead Review (FINAL)	% Needed	0.5	1	1	17%	8%	*Matrix Dependency*
3	JETS Checking Review (FINAL)	% Needed	2	3	4	25%	25%	*Matrix Dependency*
4	TTDR (FINAL)	% Needed	2	3	4	50%	25%	*Matrix Dependency*
5	Drawing in EDRS (FINAL)	% Needed	0.5	0.5	0.5	75%	4%	*Matrix Dependency*
6	Complete EDRS Sig Redlines (FINAL)	% Needed	1	2	3	79%	17%	*Matrix Dependency*
7	EDCC Release (FINAL)	% Needed	0.5	0.5	0.5	96%	4%	*Matrix Dependency*
						100%	-100%	

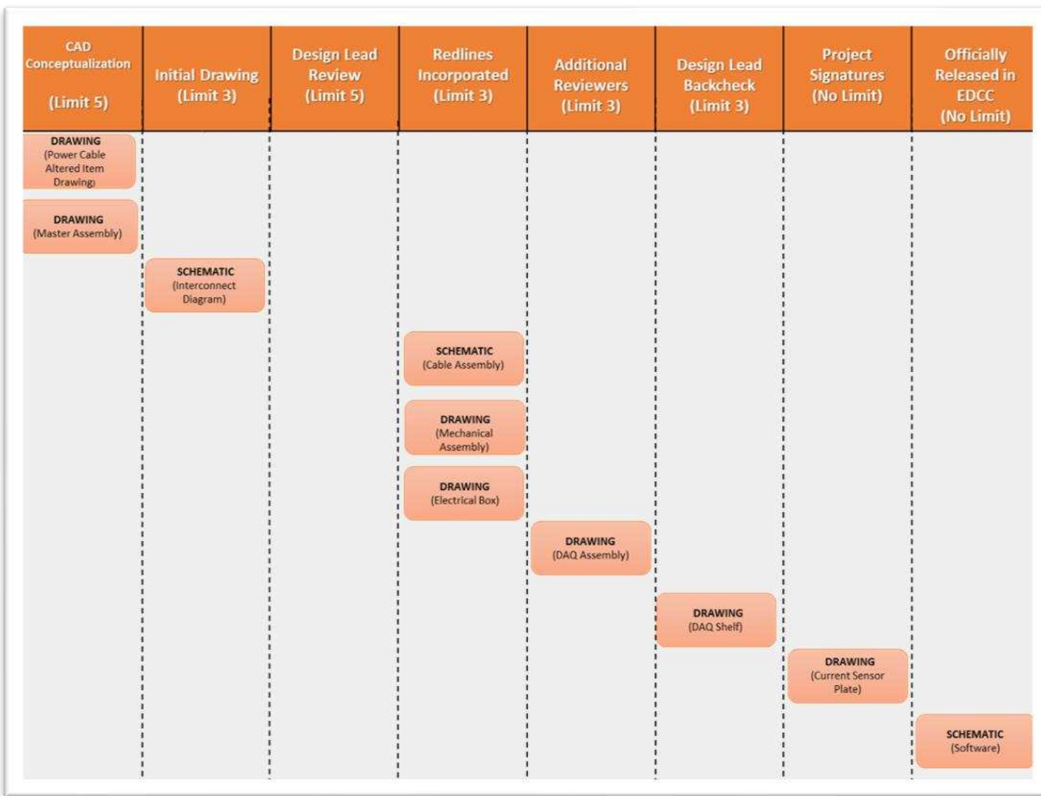


Figure 54 – Kanban Drawing Tracker

The importance of obtaining metrics allows for a method for the ScrumMaster and Project Manager to continuing wave rolling planning for the areas of Scrum in which schedule can be continually forecasted. This can also serve as a method to use metrics in a capacity to monitor and control project performance in evaluation of the forms of Lean Waste. These could be classified as velocity moving averages and SPI modifications to schedule to wave roll plan. The preferred method was to keep a weekly tracker up to date with copy over from each previous week so that a visualization of progress and workflow can be used to compare and obtain deltas throughout the project progression.

Step 5: Improvements & Limitations

Each iteration will present a series of improvements and limitations. For Iteration 3, improvements to model allows for the wave rolling planning to improve schedule velocity and Kanban to improve workflow visualization. Limitations from previous iterations have all been addressed and concerns closed for the focus group. These include:

- How to develop hardware with Scrum: This is a software-intensive process as software can change and iterate much more quickly and cheaply as opposed to hardware. These challenges include a series of constraints of physicality in purchasing, modification and changing of hardware vs. software.
  - Goal: successfully provide Scrum framework tailoring to optimize hardware development teams.
  - Mitigation: use a working, moving prototype in a minimal capacity to address proof of concept.
- The approach that must be taken to organize work priorities, workflow and characterization of true schedule: while a template is given to organize work, this varies

greatly by how hardware vs. software is built. While priorities may seem apparent at the onset, certain aspects of hardware development cannot quickly and easily be incremented and implemented and as a result, backlog organization, workflow and characterization of true schedule remain challenges.

  - Goal: characterize workflow, priority organization and schedule prediction by developing a method to illustrate, organize and improve schedule performance.
  - Mitigation: use Scrum to facility product backlog burndowns and schedule velocity calculations to improve schedule forecasting and

- Challenges associated with cross functional development: While it is attractive to have a team of generalists, how to train and cross-develop these individuals is a challenge in itself before the team can be considered M-Shaped vs. I-Shaped.
  - Goal: find a method to cross-train all various disciplines of the development team.
  - Mitigation: facilitate XP in terms of pair programming to cross train interdisciplinary engineering groups.
- How to integrate if required with waterfall: At the conclusion of the iteration and before the onset of Iteration 3, it is evident that a standalone Scrum model may not be sufficient to address not only the needs of the performing organization but also its compatibility.
  - Goal: (if required) find a means to combine best practices from both methods while improving or eliminating the shortcomings (i.e., inability for waterfall to quickly address change, challenge for Scrum to be implemented with hardware teams).
  - Mitigation: reduce Scrum to areas that are more malleable (early phases) of

project where early value can be given to the customer while keeping them engaged, wave rolling planning can be beneficial, establish a working prototype and implement Waterfall into more traditional operations.

- How to work with a remote working team: As complications from COVID intermingled with the challenges already described, Scrum promotes itself by working with a centralized team. How to address this with the advent of COVID cannot be understated.
    - Goal: how to fully integrate a remotely working team unable to learn by osmosis in an office environment.
    - Mitigation: Kanban in the form of drawing trackers allows for visibility and engagement of teammates from a pull-communication location.
- How to obtain acceptance by customers, development team and performing organization: Maybe the most significant challenge would be acceptance from customer and performing organization. But first, acceptance by the develop team needs to be harnessed.
    - Goal: creating a process by which the development team and subsequently customer and performing organization can vet via survey to gain acceptance of tempered model.
    - Mitigation: After Iteration 3, focus groups have accepted the current model, with the limitation that it has not been fully vetted in a full project capacity.

Step 6: Overall Summary of Iteration

An abridged table is given to reflect the aforementioned iteration number, time duration, data gathering techniques, improvements and limitations.

Table 67 – Iteration 3A Summary

Iteration 3	Time Period	Data Gathering	Improvements	Limitations	Limitations Addressed
Modified Scrum Developed	May 2022 - October 2022	Final Literature Review	Allows for Incremental and Deliveries	How To Develop Hardware with Scrum	Via Rapid Prototyping, Scrum Used for Highly In Planning
		3rd Survey	Allows for Iterative Deliveries	How To Organize Work Priorities	Approach Functional Managers About Decreasing Multi-Tasking
Lean Introduced		Interviews	Allow for Priorities to be Organized Periodically	How To Demonstrate Workflow	Kanban with Burndown Chart to Demonstrate Workflow and Improve Schedule Forecasts
		Focus Groups	Utilizes a Burndown Chart for Schedule Forecast	How To Characterize True Schedule	Utilize Burndown Charts to Course Correct Schedule with Modified EVM Calculation
Modified Kanban Introduced			Promotes Generalists over Specialists	How To Create Cross-Functional Generalists	Approach Functional Managers About Increasing Multi-Tasking
			Appoints a Servant Leader in a ScrumMaster	How To Develop Impediment Improvements	Lean Addresses Ability to Help Remove Impediments
XP Introduced			Allows for Consistent Customer Engagement	How To Integrate with Traditional Waterfall	Utilize Traditional Waterfall as Skeleton Framework
			Promotes Learning by Osmosis	How To Utilize with a Remotely Working Team	Usage of Microsoft Teams and OneNote
FDD Introduced			Identification of Overprocess (Lean)	How to Quantify Lean Waste	FMEA used to quantify risk points while providing quality and safety are not changed
			Identification of Hand-Off (Lean)	How to Quantify Lean Waste	
Modified Waterfall Re-Introduced			Identification of Waiting (Lean)	How to Quantify Lean Waste	
			Identification of Tasking Switching (Lean)	How to Quantify Lean Waste	

8.8. ITERATION 1B: Requirements Engineering Development

8.8.1. Iteration Development

The period of performance of Iteration 1 focused between February 2021 and February 2022. The data gathering techniques included:

- Initial Literature Review: Where requirements engineering concepts from academic, and peer reviewed journals are viewed.
- Case Study Review #1: Where the specific case studies of lunar dust mitigation and auxiliary lighting help support the notion of the development of a tool for requirement scoring. This iteration focused more on the information gathered for the lunar dust mitigation study.
- Brainstorming: Various, semi-formal sessions were elicited with stakeholders from across the project to assess which areas of systems engineering presented challenges and for project lifecycle development, those specific challenges documented.

- Interviews: Various, semi-formal sessions were elicited with stakeholders from across the project to assess which areas of systems engineering presented challenges and for requirement engineering development, those specific challenges documented.
- Focus Groups: Various, semi-formal sessions were elicited with stakeholders from across the project to assess which areas of systems engineering presented challenges and for requirement engineering, those specific challenges documented.
- Questionnaires: Various, semi-formal sessions were elicited with stakeholders from across the project to assess which areas of systems engineering presented challenges and for requirement engineering development, those specific challenges documented.
- 1st Survey: A Likert scale survey to assess the current vs. desired state of Scrum offerings on the project with responses from the Development Team and customer.

The goal of the iteration is to create a scorecard for requirements to test robustness. For context, robustness is a measure of how fit a requirement is for use against the most appropriate, current state-of-the-art. Literature review and expert elicitation on project indicate that INCOSE documents, including but not limited to the NRM, GtWR, GtNR and the GtVV would be leveraged to create the scorecard.

A survey with 22 individuals across the project (customer and performing organization) were polled on the Likert scale to elicit information regarding waterfall sentiment on the project; 1 – Strongly Disagree, 5 – Strongly Agree; any score above a 3.5 (median between "Neither Agree or Disagree and Somewhat Agree) is deemed a positive indication on position; any score below a 3 is considered a negative indication on position. The survey below was used to compare the current vs. potential new requirements engineering developments to examine how malleable the team would be to a change in requirements engineering but also how successful the current approach remains. Results indicate that overall, the project found requirements engineering to be adequate. However, after filtering results against the lunar dust mitigation efforts and auxiliary lighting requirements, there was a demand by project to improve use case tooling, matrix decomposition tooling, requirements scorecard inclusion and glossary tooling. The "possible tool" column is separate from the survey and indicates which tool may be most helpful for the team (paired with color coding on question(s) posed).

Table 68 – Requirements Engineering General Likert Questions

#	**Requirements Engineering General Questions**
1	Systems engineering in terms of team hierarchy is appropriately organized on the project
2	Systems engineers are appropriately able to influence processes on the project
3	There is a need for systems engineer to influence processes on the project
4	Systems engineers are appropriately familiar with the xEMU system
5	In place architectures have been built appropriately to properly satisfy requirements building
6	I prefer to work with an MBSE (Model Based Systems Engineering) tool (i.e., Cradle) for requirements management over standard DBSE (Document-Based Systems Engineering) tools
7	In place MBSE tools (i.e., Cradle) have been used appropriately as it pertains to requirements building
8	Configuration management is appropriately followed in terms of Cradle requirements management
9	The Architecture Design Document has been effective in requirements building
10	The Concept of Operations have been effective in requirements building
11	The Systems Engineering Management Plan was appropriate for the project
12	A specification tree across the entire system (including subsystems, units) was appropriately developed
13	Requirements building in terms of stakeholder elicitation has been appropriate on the project
14	Requirements building in terms of project management was appropriately managed on the project
15	Decomposition of requirements via top level customer needs was appropriately used to derive requirements
16	Decomposition via goals was appropriately used to derive requirements

17	Decomposition via sub-goals was appropriately used to derive requirements
18	System requirements are indicative of the stakeholder needs
19	Subsystem requirements are decomposed accurately from their system requirements
20	End item requirements are decomposed accurately from their subsystem requirements
21	Use cases were appropriately used to derive requirements
22	Scenarios were illustrated when developing requirements
23	Requirements were appropriately decomposed
24	Verifications were appropriately written
25	Validations were appropriately written
26	Unit tests appropriately represent the verifications and validations they are written against
27	Subsystem tests appropriately represent the verifications and validations they are written against
28	System tests appropriately represent the verifications and validations they are written against
29	There is a clear distinction of difference between verifications vs. validations
30	Successful unit tests correlate to robustly written and developed verifications and validations
31	Successful subsystem tests correlate to robustly written and developed verifications and validations
32	Successful systems tests correlate to robustly written and developed verifications and validations
33	The Engineering V&V model implementation is not critical to the project's success
34	The Engineering V&V model was appropriately utilized by the project
35	I understand when a requirements package completely meets the customer's needs
36	There was a tool to illustrate when a requirements package completely met the customer's needs on the project
37	There was an appropriate rubric to build requirements on the project (scoring, grading, etc.)
38	Having a score card would be beneficial when developing requirements
39	I am appropriately versed in INCOSE requirements standards
40	We used INCOSE requirements standards on the project
41	Having a requirements matrix decomposition tool starting from high level customer needs all the way through to end items specifications would be helpful
42	We used a requirements matrix decomposition tool to organize high level customer needs all the way through to end items specifications
43	I prefer having a glossary for terms during requirements building
44	We had a glossary for terms during requirements building
45	Requirements are appropriately given owners during their development
46	Rationales were documented for the requirements appropriately
47	Requirement characteristics were appropriately captured
48	Requirement attributes were appropriately captured
49	Requirements are appropriately traceable to parents
50	Requirements are appropriately traceable to children
51	The xEMU system context was appropriately represented
52	Current state to desired state requirements transformation in terms of a lunar suit was appropriately captured

	Overall Project		**Lunar Dust**		**Auxiliary Lighting**		**Possible Tool**
#	**Mean**	**Count**	**Mean**	**Count**	**Mean**	**Count**	**(If Applicable)**
1	4.5	22	-	-	-	-	

2	4.1	21	-	-	-	-	
3	4.45	22	-	-	-	-	
4	3.95	21	3	2	5	1	
5	3.82	17	-	-	4	1	
6	3.94	18	-	-	-	-	
7	3.5	18	-	-	-	-	
8	3.86	14	-	-	-	-	
9	3.83	18	4	2	3	1	
10	3.61	18	2.5	2	3	1	
11	3.8	15	2.5	2	3	1	
12	3.86	14	2	1	4	1	
13	3.85	20	2	2	1	1	
14	4.06	17	2.5	2	3	1	
15	3.78	18	1.5	2	2	1	
16	3.62	13	2	2	1	1	Use Case
17	3.75	12	1	1	1	1	
18	4.47	19	3.5	2	1	1	
19	4.29	17	3	2	1	1	
20	4.11	18	1.5	2	5	1	
21	3.5	18	3	2	2	1	Matrix Decomposition, Use Case, Scorecard
22	3.31	16	2.5	2	1	1	
23	4	17	3	2	2	1	
24	3.63	19	1.5	2	2	1	
25	3.38	16	1	2	1	1	
26	4.18	17	1	1	3	1	
27	4.53	15	1	1	2	1	
28	4.27	15	1	1	0	0	
29	3.5	20	1.5	2	2	1	
30	4.18	11	1	1	3	1	
31	4.08	12	1	1	2	1	
32	4.2	10	1	1	0	0	
33	2.21	19	2	1	3	1	
34	3.82	17	4	1	3	1	
35	4	19	3.5	2	5	1	
36	2.69	16	1.5	2	1	1	Matrix Decomp., Use Case, Scorecard, Glossary
37	2.5	14	1	2	1	1	
38	3.81	16	4.5	2	5	1	
39	3.37	19	2.5	2	4	1	

40	3.85	13	3	1	1	1	Scorecard, Glossary
41	4.4	20	4	2	5	1	
42	3.36	14	0	0	1	1	
43	4.35	20	4	2	5	1	
44	3.91	11	2	1	4	1	
45	3.88	17	3.5	2	5	1	
46	4.45	20	4	2	4	1	
47	3.89	18	2.5	2	2	1	Matrix Decomposition
48	3.73	15	2.5	2	1	1	
49	4.7	20	4	2	2	1	
50	4.37	19	4	2	5	1	
51	4.2	15	1.5	2	3	1	Matrix Decomposition
52	3.6	15	1	1	1	1	

A questionnaire with 22 individuals across the project (customer and performing organization) was given to allow for opened-ended feedback. The question and abridged answers suggested a modification to the project's approach to requirements engineering. Responses are as follows:

- *Q1: What could be improved with the requirements engineering model or any general comments/concerns?*

- *A1:*
    - *"Challenges: Dust/Regolith, comm, lighting. There was initially moderate resistance to using Cradle by some teams, but I think that requirements management as a whole improved as teams used Cradle more."*
    - *"The ADD shouldn't influence requirements, it's the other way around*

(requirements influence architecture). However, xEMU went through a decade of tech dev research beforehand and therefore the SE&I cycle was a bit backwards. An agile approach with might have worked better for xEMU SE&I. Clear customer requirements/objectives would have also been a huge help."

- *"I didn't see Cradle used as a tool to anything more than storing data. It could have been used a lot more for traceability work. It is unfortunate that, as usual, not enough resources were committed to training folks to use the tool and getting the most out of it."*
- *"...emphasis on requirements traceability is very important early in the development phase, so that anything that may be missed (at the component, subsystem, system levels) could be captured as development progresses (and not wait until the end to find disconnects). Another thing I struggled with was better definition of integrated testing requirements, to better understand implication on component level requirements. This may have occurred earlier in the project, but there was visibility lacking on test plans/requirements until testing was upon us. "*
- *"There were delays in preparing for testing because the system requirements were not decomposed at the beginning to figure out what test equipment/instrumentation would be needed to gather that data. "*
- *"Efforts were made to load the Con-Ops into Cradle to better address scenarios - This suggestion was rejected. The Architecture group was more concerned with producing an ops useable description product that captured*

design versus influence design."

- *"Component spec requirements were not properly decomposed from their parents."*
- *"More knowledge and empowerment to systems engineers for technical discussions/decisions to work with the component owner."*
- *"Requirements that were more subjective or difficult to define (e.g., mobility tasks, lunar dust, usability) made for a lot of discussion and frustration among engineers."*

Step 1: Develop a Scorecard

The most cursory attempt was to quantify which aspects of requirements engineering were most key in developing a sound requirement. Experts on lunar dust mitigation and auxiliary lighting selected several requirements that may benefit from a robustness test. Scores were given a simple a score of Yes/No/Maybe with assigned scores of 1/0.5/0, respectively and a percent of fitness against robustness completed across the following categories: traceability, verifiability, validatability, correctness, terminology, rationale.

Table 69 – Coarse Grading of Current Lunar Dust and Auxiliary Lights Requirements

	Lunar Dust Requirements																	
INCOSE METRIC	R.SS-3033	xEMU.ENV.030	R.INFO.4025	R.INFO.4030	R.PGS.4222	R.LIT.4127	R.LIT.4126	R.ARMS.420	R.BOOT.517	R.EPG.156	R.GLVS.418	R.HELM.501	R.HUT.503	R.LEG.420	R.SHDR.501	R.VLMH.117	R.WBH.417	R.WBH.519
Traceability?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Verifiable?	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
Validatable?	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
Correctly Written?	Yes	Yes	No	Yes	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No
Adequate Terms?	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Adequate Rationale?	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
GRADE	33%	33%	33%	67%	67%	50%	50%	50%	50%	50%	50%	50%	50%	50%	50%	50%	50%	50%

	Auxiliary Lighting											
INCOSE METRIC	R.PLSS.600.153	R-DCU-685.082	R.DCU-685.133	R.DCU-685-.120	R.DCU-685.121	R.DCU-685.NEW1	R.DCU-685.NEW2	R.DCU-685.NEW3	R.DCU-685.NEW4	R.DCU-685.0xx [TBD]	R.DCU-685.0xx [TBD]	R.DCU-685.0xx [TBD]
Traceability?	No	No	No	No	No	No	No	No	No	No	No	No
Verifiable?	No	No	No	Maybe	Maybe	No	No	No	No	No	No	No
Validatable?	No	No	No	Maybe	Maybe	No	No	No	No	No	No	No
Correctly Written?	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Adequate Terms?	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
Adequate Rationale?	Yes	Yes	Yes	Yes	Yes	No	No	No	No	Yes	Yes	Yes
GRADE	17%	17%	67%	67%	67%	33%	33%	33%	33%	33%	17%	17%

The majority of INCOSE concepts were lifted directly from best practices and slightly tailored. The first iteration attempted to quantify and measure requirement robustness against the GtWR characteristics. Only a sampling of the various GtWR metrics were selected per the areas of concern illustrated by the team and an analytical hierarchy process applied. A cursory, first iteration review indicated that these best practice requirement characteristics would be indicative of the most important comparative measures for requirement robustness, which the expert panel of 6 focus group members did not agree.

Table 70 – Iteration 2A Scoring Rubric

ID No.	Best Practices	Grade?
1	Is the requirement traceable?	75%
2	Is the requirement verifiable?	65%
3	Is the requirement able to be validated?	90%
4	Is the requirement correctly written?	90%
5	Does the requirement use adequate terms?	40%

6	Does the requirement have adequate rationale?	55%

Step 2: Improvements & Limitations

Each iteration will present a series of improvements and limitations. For Iteration 1, there was no major improvement to the model aside from the notion of including a viable method for requirement scoring, which during first iteration elicited more question for future iteration tempered modeling. Some improvements including a foray into checking requirement traceability, verifiability, validatability, writing correctness, usage of adequate terms and rationale. Limitations include many of the improvements instigated yet derives a means by which to achieve them. These include:

- How to develop a quantitative method to evaluate robustness of requirement.
    - Goal: successfully provide a scorecard based on all valuable parameters that dictate what a robust requirement needs to fulfill.
- The approach that must be taken to effectively allow team members to operate outside of MBSE tools in a capacity that gives clarity to requirement decomposition in a minimalist manner.
    - Goal: illustrate requirement decomposition in a lightweight matrix decomposition tool across needs to requirements to subsystem requirements to component requirements.
- Challenges associated with developing against the Con-Ops and ADD as some of the operations and usages were perceived as underdeveloped in the requirements building phase against current project documents.
    - Goal: develop a use case scenario tool.
- Issues persist with nomenclature (i.e., regolith definition, auxiliary lighting definition

and intended usage).

- Goal: develop a glossary tool.

Step 6: Overall Summary of Iteration

An abridged table is given to reflect the aforementioned iteration number, time duration, data gathering techniques, improvements and limitations.

Table 71 – Iteration 2A Summary

Iteration 1	Time Period	Data Gathering	Improvements	Limitations
INCOSE Best Writing Practice Checklist	October 2021 - February 2022	Initial Literature Review	Allows for checks on requirement traceability	No Quantitative Method to Evaluate Robustness of Requirement
		Case Study Review #1 Lunar Dust	Allows for checks on requirement verifiability	No Method for Decomposition Checks
		Brainstorming	Allows for checks on requirement validatability	No Method for Addressing Use Case Scenarios
		Interviews	Allows for checks on requirement writing correctness	No Method for Glossary of Terms
		Focus Groups	Allows for checks on requirement adequate terms	
		Survey	Allows for checks on requirement adequate rationale	

8.9. ITERATION 2B: Requirements Engineering Development

8.9.1. Iteration Development

In the second iteration, an upgrade was made to the scorecard using the GtWR, NRM, GtVV and GtNR and the analytical hierarchy process was removed. A screening is done, leveraging a simpler method for fitness of robustness, following a Yes/No/Maybe against well-established INCOSE metrics. Additional tools were developed as requested and include the glossary, use case, and requirements matrix decomposition tools. The period of performance of Iteration 2 focused between February 2022 and May 2022. The data gathering techniques included:

- Intermediate Literature Review: Where INCOSE requirements engineering concepts from academic, and peer reviewed journals are viewed.

- Case Study Review #2: Where the specific case studies of lunar dust mitigation and auxiliary lighting help support the notion of the development of a tool for requirement scoring. This iteration focused more on the information gathered for the auxiliary lighting study.
- Brainstorming: Various, semi-formal sessions were elicited with stakeholders from across the project to assess which areas of systems engineering presented challenges and for project lifecycle development, those specific challenges documented.
- Interviews: Various, semi-formal sessions were elicited with stakeholders from across the project to assess which areas of systems engineering presented challenges and for INCOSE requirements engineering, those specific challenges documented.
- Focus Groups: Various, semi-formal sessions were elicited with stakeholders from across the project to assess which areas of systems engineering presented challenges and for INCOSE requirements engineering development, those specific challenges documented.
- Survey: A Likert scale survey to assess requirements engineering offerings on the project with responses from the Development Team and customer.

The goal of the iteration is to further develop the scorecard and present the use case, requirements decomposition matrix and glossary tools to the panel of experts for approval.

A survey with 22 individuals across the project (customer and performing organization) were polled on the Likert scale to elicit information regarding waterfall sentiment on the project; 1 – Strongly Disagree, 5 – Strongly Agree; any score above a 3.5 (median between "Neither Agree or Disagree and Somewhat Agree) is deemed a positive indication on position; any score below a 3 is considered a negative indication on position. The excerpts from the original survey were used to compare the current vs. potential new requirements engineering developments with regards to a matrix decomposition tool, a glossary tool and a use case tool. Results indicate that for the lunar dust mitigation efforts and auxiliary lighting requirements, there was a demand by project to improve use case tooling, matrix decomposition tooling, requirements scorecard inclusion and glossary tooling.

Table 72 – Use Case, Decomposition Matrix, Glossary and Scorecard Specific Likert Questions

QUESTION (Like Scale 1 to 5) 5 – Strongly Agree	Overall Project		Lunar Dust		Auxiliary Lighting	
Decomposition via goals was appropriately used to derive requirements	3.62	13	2	2	1	1
Decomposition via sub-goals was appropriately used to derive requirements	3.75	12	1	1	1	1
System requirements are indicative of the stakeholder needs	4.47	19	3.5	2	1	1
Use cases were appropriately used to derive requirements	3.5	18	3	2	2	1
Scenarios were illustrated when developing requirements	3.31	16	2.5	2	1	1
Requirements were appropriately decomposed	4	17	3	2	2	1
Having a requirements matrix decomposition tool starting from high level customer needs all the way through to end items specifications would be helpful	4.4	20	4	2	5	1
We used a requirements matrix decomposition tool to organize high level customer needs all the way through to end items specifications	3.36	14	1	1	1	1
I prefer having a glossary for terms during requirements building	4.35	20	4	2	5	1
We had a glossary for terms during requirements building	3.91	11	2	1	4	1
Requirements are appropriately traceable to parents	4.7	20	4	2	2	1

Step 1: Develop Use Case Scenario Tool

The use case scenario template was developed against the literature reviews and feedback with the focus group panel of experts. A template was established, both in graphical and tabular form and refined using the lunar dust mitigation and auxiliary lighting use cases for specific scenarios. These diagrams are used to illustrate how actors both external and internal to the system provoke and drive the high-level functionality. On the xEMU project where certain requirements were built on a Concept of Operations (Con-Ops) or detailed in an Architecture Design Document, instances revelated that a cursory application of a use case diagram may have yielded proper direction on requirements decomposition.

Table 73 – Use Case Template

Template	
Context Information	
Pre-Condition	*List the conditions that exist before the scenario*
Post-Condition	*List the conditions that exist after the scenario*
Actors	*List the actors interacting with the system*
System	*List the system*
Resources	*List all necessary resources in order for scenario to reach satisfaction/dissatisfaction*
Location	*List the location, actual or fictional*
Goal Satisfaction	*Indicate the success criteria for goal satisfaction*
Goal Dissatisfaction	*Indicated the failures to induce a goal dissatisfaction*

Use Case Scenarios	
Main Scenario	***List Scenario Name***
1	*List the steps sequentially as they would be performed to achieve main scenario completion*
2	*List the steps sequentially as they would be performed to achieve main scenario completion*
3	*List the steps sequentially as they would be performed to achieve main scenario completion*
4	*List the steps sequentially as they would be performed to achieve main scenario completion*
5	*List the steps sequentially as they would be performed to achieve main scenario completion*
Alternative Scenario	***List Scenario Name***
1	*List the steps sequentially as they would be performed to achieve alternative scenario completion*
2	*List the steps sequentially as they would be performed to achieve alternative scenario completion*

3	*List the steps sequentially as they would be performed to achieve alternative scenario completion*
Exception Scenario	***List Scenario Name***
1	*List the steps sequentially as they would be performed to achieve exception scenario completion*

The above templates were utilized by both the lunar dust mitigation and auxiliary lighting teams.

Table 74 – Use Case Auxiliary Lighting Example

Example	
Context Information	
Pre-Condition	Crew Member Prepares for EVA.
Post-Condition	Crew Member Completes EVA.
Actors	Astronaut (A) in xEMU on EVA; Astronaut (B) not in xEMU. assisting (A) from habitat.
System	xEMU Space Suit.
Resources	All necessary power, cooling, oxygen systems aboard habitat to supply xEMU; xMWS donned on crew member's hip.
Location	Microgravity in a vacuum.
Goal Satisfaction	Worksite is properly illuminated to allow for EVA success
Goal Dissatisfaction	Worksite is inadequately illuminated preventing successful EVA.

Use Case Scenarios	
Main Scenario	**Primary Lighting Source**
1	Astronaut doffs suit.
2	Astronaut does a system check on lighting sources.
3	Astronaut egresses from habitat and activates primary lighting.
4	Astronaut utilizes primary lighting to successfully complete EVA.
5	Astronaut ingresses to habitat and deactivates primary lighting.
Alternative Scenario	**Secondary Lighting Source**
1	If primary lighting fails after doffing but before egress, astronaut will follow off-nominal procedures to bring primary lighting to nominal conditions.

2	If primary lighting fails after egress but before EVA, astronaut will follow off-nominal procedures to bring primary lighting to nominal conditions.
3	If the primary lighting fails during EVA, a secondary lighting option will be an auxiliary tool located in the xMWS that the crew member will utilize in a similar fashion of a flashlight to illuminate the worksite.
Exception Scenario	**Contingency Return to Habitat**
1	If the primary lighting and secondary lighting fail during EVA, the crew member must discontinue EVA tasks and use environmental lighting to successfully return to habitat.

\

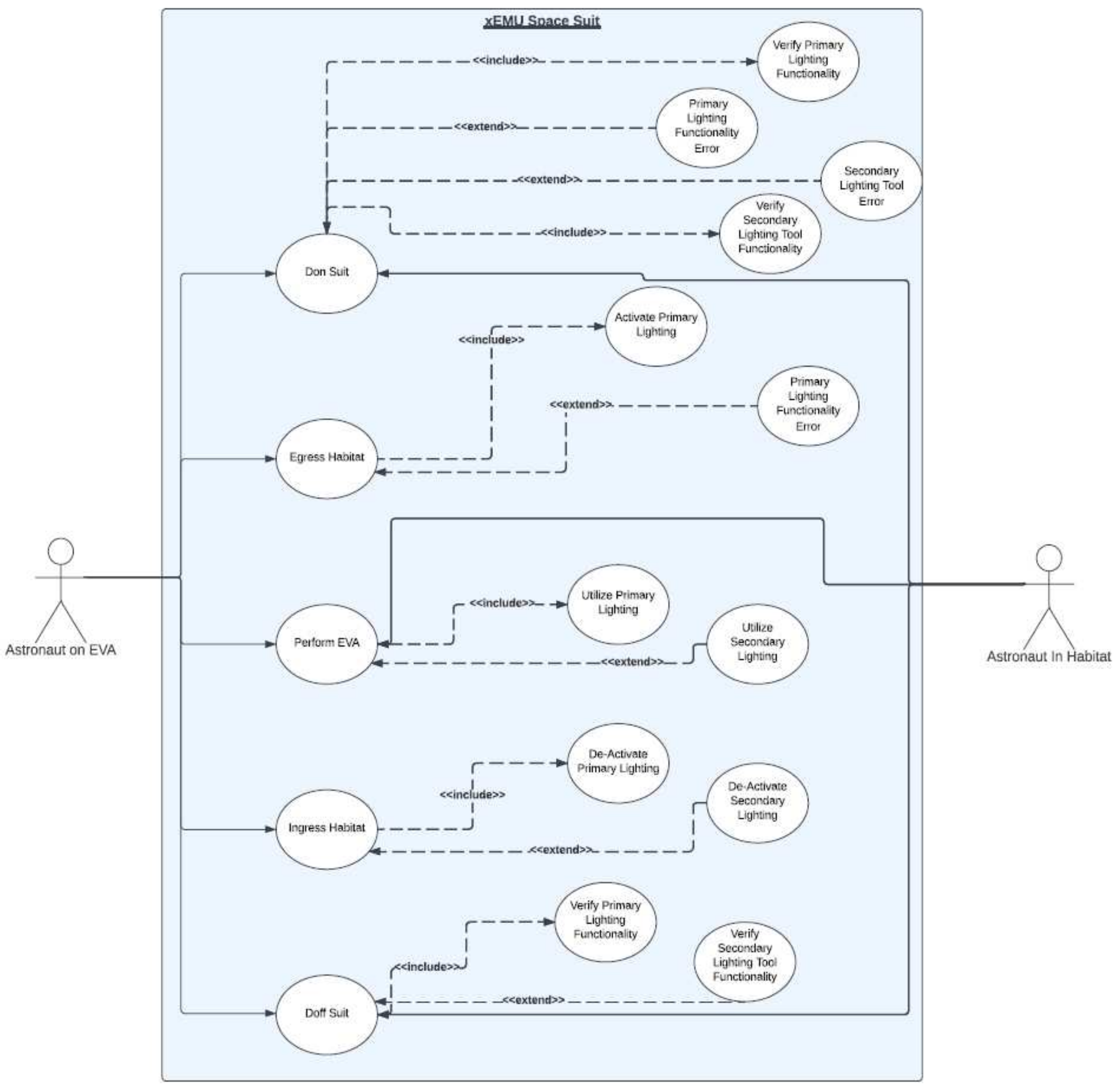


Figure 55 – Use Case Diagram Auxiliary Lighting Example

Table 75 - Use Case Lunar Dust Example

Example	
Context Information	
Pre-Condition	Crew Member Collects Regolith during EVA.
Post-Condition	Crew Member Removes Regolith after EVA.
Actors	Astronaut (A) in xEMU on EVA having suit cleaned; Astronaut (B) cleaning the suit of Astronaut (A). Note: Both crew will need to be cleaned but only one is characterized in use case.
System	xEMU Space Suit.
Resources	Lunar dust cleaning tools outside of the habitat
Location	Microgravity in a vacuum.
Goal Satisfaction	Crew successfully cleans suit and removes appropriate regolith.
Goal Dissatisfaction	Crew unsuccessfully cleans suit and crew returns with uncleaned suit into habitat.

Use Case Scenarios	
Main Scenario	**Primary Tools**
1	Astronaut (A) and (B) doff suits.
2	Astronaut (A) and (B) performs EVA
3	Astronaut (A) and (B) return to habitat but does not enter
4	Astronaut (A) cleans Astronaut (B) with Primary Tools for 15-minutes to remove regolith
5	Astronaut (A) is ready to ingress to habitat.
Alternative Scenario	**Secondary Tools**
1	If the primary tools are not available, Astronaut (B) will access the Secondary Tools to clean Astronaut (A).
2	Astronaut (A) cleans Astronaut (B) with Secondary Tools for 15-minutes to remove regolith
3	Astronaut (A) is ready to ingress to habitat.
Exception Scenario #1	**Contingency Return to Habitat**
1	If there is an off-nominal event (i.e., Emergency) where time is not available to clean the crew, Astronaut (A) and Astronaut (B) will return to habitat uncleaned.

Exception Scenario #2	**Contingency Return to Habitat**
1	In the event the Primary and Secondar Tools are unable or unavailable to clean crew within the 15-minute window, Astronaut (A) and Astronaut (B) will return to habitat uncleaned.

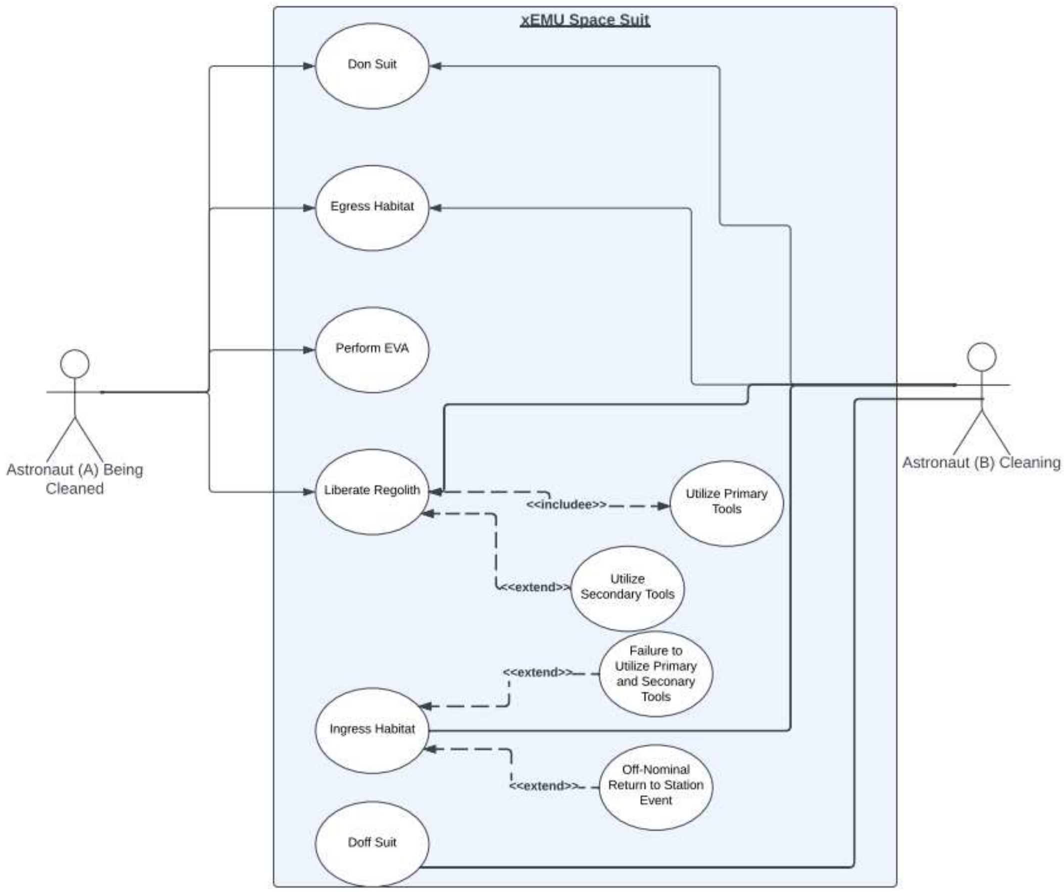


Figure 56 - Use Case Diagram Lunar Dust Example

Step 2: Develop Glossary Tool

The glossary tool is a simple but effective method to ensure all terms, formulas and units are correctly defined, promote proper requirement decomposition, subsequent design and are accepted by both the customer and performing organization. First, the units and terms currently defined by the project were addressed, especially those that are applicable to the requirement or requirement set in question.

Table 76 – Proposed Glossary Terms

Accepted Project Engineering Terms	Project Definition	Units
Time	seconds	s
Length	foot	ft
Mass	pound mass	lbm
Force	pound -force	lbf
Temperature	degree Fahrenheit	F
Voltage	volts	V
Current	amps	A
Resistance	ohms	Ω
Power	watts	W
Pressure	pounds per square inch	psi
Angle	degrees	°

Second, the units and terms currently proposed by the project were addressed so that the new requirement or requirement set in question can either be correctly developed or glossary units and terms denied as they may negatively impact the development of the requirement or requirement set. It was also advisable to list the source of the addendum request. The auxiliary lighting team utilized such a development as illustrated in the tool below.

Table 77 – Proposed Glossary

Proposed Engineering Term	Proposed Definition	Abbreviation or Unit	Approved?
Beam Distribution	Sometimes referred to as Beam Distribution, Beam Angle, or Beam Characterization. This represents a 90° hemisphere or 180° spherical characterization of the intensity of light at multiple angles from the source. Typically, illuminance measurements are captured at a fixed radius at multiple angles. Beam distribution is usually reported in relative percent intensity per angle with estimated lumen output, candela per angle, or illuminance per radius per angle.	N/A	Pending

Half-Width-Half-Maximum	Acronym for shorthand numerical description of a lamp beam pattern. HWHM stands for half-width-half-maximum. If HWHM is given for a lamp, the number represents the half-angle (angle drawn from normal vector from center of the lamp) at which the lamp's beam distribution intensity falls to 50%.	HWHM	Pending
Spectral Irradiance	Radiometric unit, analogous to illuminance, representing the radiant flux per surface area per wavelength. Units are in watts/meter^2/nanometer (W/m^2/nm).	W/m^2/nm	Pending
Spectral Radiance	Radiometric unit, analogous to luminance, representing the radiant flux emitted by a given surface area per solid angle per wavelength. Units are in watts/steradian/meter^2/nanometer (W/sr/m^3/nm).	W/sr/m^3/nm	Pending
Spectral Power Distribution	Waveform representing energy (absolute or relative) emitted per a range of wavelengths. All light sources have a unique spectral power distribution (SPD) that is impacted by its chemistry. The SPD is an essential dataset for estimating metrics dependent on wavelength.	SPD	Pending
Chromaticity	This is a calculated metric where the format of the units can be different depending on which standard is used. Chromaticity describes the color of an object, whether that be a surface material or light source. Chromaticity can't be estimated without the usage of a spectrophotometer to measure the spectral power distribution of a light emitting source or reflectance spectrum of a material.	N/A	Pending
Color Fidelity	This is a calculated metric where the format of the units can be different depending on which standard is used. Color Fidelity describes the accuracy of a light source to render the appearance of colored materials accurately where the definition of perfect is how the	N/A	Pending

	Sun's renders the color of materials. Color Fidelity can't be estimated without the usage of a spectral radiance or spectral irradiance meter to measure the spectral power distribution of a light emitting source.		
Goniophotometer	Specialized test equipment configuration that includes a rotation stage and is used to collect beam distribution data for a light source. The type of goniophotometer is defined by the location of the rotation stage (lamp verses sensor).	N/A	Pending
Diffusion/Diffusor	A light diffusion material or diffusor is a material designed to scatter or redirect light that passes through it, or it can also represent a rough surface that light impacts and scatters multiple directions from.	N/A	Pending
Reflectance/Reflector	The property of a material to reflect and scatter light. Reflectance of surface materials is an important lighting system property as it impacts how humans and cameras observe the environment and the efficiency of lighting systems to illuminate surfaces to sufficient levels to create the desired luminous contrast. Reflectance can be considered part of the architecture and can be used as a tool in the form of a reflector.	N/A	Pending
Uniformity	This is a property that is typically applied for surface illumination but can also be applied to the light emitting face of light sources. Uniformity is usually defined in the form of ratios such as maximum/minimum, and average/minimum with a defined sampling grid size. Uniformity is an important safety and usability metric to minimize human error due to uneven illumination. Uniformity is achieved through a combination of beam distribution design, lamp placement, and understanding of reflective surfaces for the operational area.	N/A	Pending

Glare	This is a property that describes various problems in human perception of light and the interaction of light with surfaces and materials within an operational environment. Distracting glare is an “annoyance” where, because of reflection and refraction, it creates visual artifacts making it harder to see and resolve an object. Discomforting glare is caused by bright direct and reflected light that makes it hard to look at the object because of the brightness level. Disabling glare causes objects to appear to have lower contrast because of scatter inside the eye. Blinding glare is caused by a direct or indirect light source and is so bright that the observer can’t see or is visually compromised.	N/A	Pending

Step 3: Develop a Requirement Matrix Decomposition Tool

One of the areas of interest during the case study development was the traceability to source. While a higher-level customer need or immediate distinction among the requirement suite was made, there were instances where either a feature was introduced as a requirement or a system context shift created a questionable connection to an overall need and decomposition that may have followed a non-linear approach.

The requirement matrix decomposition tool could be used for the following:

- to allow for an inspection of a newly added requirement.
- to allow for an inspection of how a system context shift may affect downstream requirements.
- a re-evaluation of the decomposition more generally in a high-touch, low-tech capacity.

The requirement matrix decomposition tool is not to be used for the following:

- as a replacement tool for an existing requirement decomposition architecture.

- a final decomposition of any requirement or set of requirements.

The first step was to illustrate the customer need, which should already be established early on if the requirement package has been decomposed and connect this to the system requirement of interest. The following categories allow for a streamlined method in organizing the requirement decomposition investigation. The figure below outlines the overall business need and the first requirement level with goals, subgoals, rationale and scenarios detailed with priority and requirement type.

Table 78 – Need and System Level Requirement Decomposition Matrix Description

Business Need/Description	Goals	Subgoals	Source of Requirement	System Requirement No. & Statement	Requirement Type	Prioritization	Rationale	Scenario(s)
Business needs define the highest level of stakeholder elicitation. With these, requirements are decomposed. • A single statement to drive the subsequent goals and requirements • Should relate to the problem the system is to solve • Should not relate to a solution to the problem the system is to solve	Goals illustrarte the intention of the objective of the system and exist as refinements on the need. • A goal is refinement of the stakeholder need • Address an issue that needs resolution • Uses the word "should" • Can have sub-goals • Allow us to determine if a system can be achieved successfully • Documents intention of the stakeholders • Can help define scenarios to support validation	Subgoals illustrate peripheral or derivative intentions of the system that exist exclusively as an extension of the goal.	Where does this requirement origninate? Examples include stakeholders, working groups, contracts, project documents	What is the corresponding system requirement number and statement? Note: Ensure proper project ID and INCOSE Best Writing Practices utilized.	What type of category would this requirement be? Examples include system, functional, non-functional, quality.	With regards to neighboring requirements, how would this requirement rank in terms of priority?	What is the justification for having this requirement? This section may be as robust as needed to include all information supporting the requirement and the subsequent decomposition.	A scenario is a possible development or sequence of events that describe the satisfaction or dissatisfaction of a goal by defining the concrete steps and relational aspects to the system context. • Parameters to define include but are not limited to actors, roles, pre-conditions, post-conditions, users, system interactions, locations, etc. • Scenarios may include but are not limited to state, misuse, descriptive, exploratory, explanatory, instance, mixed, system-internal, interaction, context, main, alternative, exception, etc.

The second step was to decompose the subsystem requirements and connect them to the system requirement of interest. These will be an extension and decomposition of the system level requirement to the subsystem level requirement. The figure below outlines the subsystem requirement level with goals, subgoals, rationale and scenarios detailed with priority and requirement type.

Table 79 - Subsystem Requirement Decomposition Matrix Description

Subsystem Requirement No. & Statement	Rationale	Requirement Type	Prioritization	Rationale	Scenario(s)
What is the corresponding subsystem requirement number and statement? Note: Ensure proper project ID and INCOSE Best Writing Practices utilized.	What is the justification for having this requirement? This section may be as robust as needed to include all information supporting the requirement and the subsequent decomposition.	What type of category would this requirement be? Examples include system, functional, non-functional, quality.	With regards to neighboring requirements, how would this requirement rank in terms of priority?	What is the rationale for having this requirement? This section may be as robust as needed to include all information supporting the	A scenario is a possible development or sequence of events that describe the satisfaction or dissatisfaction of a goal by defining the concrete steps and relational aspects to the system context.

The third step was to decompose the unit or component requirements and connect them to the subsystem requirement of interest. These will be an extension and decomposition of the subsystem level requirement to the unit or component level requirement. The figure below outlines the unit or component requirement level with rationale and scenarios detailed with priority and requirement type.

Table 80 – End Item Requirement Decomposition Matrix Description

End Item Requirement No. & Statement	Requirement Type	Prioritization	Rationale	Scenario(s)
What is the corresponding subsystem requirement number and statement? Note: Ensure proper project ID and INCOSE Best Writing Practices utilized.	What type of category would this requirement be? Examples include system, functional, non-functional, quality.	With regards to neighboring requirements, how would this requirement rank in terms of priority?	What is the rationale for having this requirement? This section may be as robust as needed to include all information supporting the	A scenario is a possible development or sequence of events that describe the satisfaction or dissatisfaction of a goal by defining the concrete steps and relational aspects to the system context. • Parameters to define include but are not limited to actors, roles, pre-conditions, post-conditions, users, system interactions, locations, etc. • Scenarios may include but are not limited to state, misuse, descriptive, exploratory, explanatory, instance, mixed, system-internal, interaction, context, main, alternative, exception, etc.

A decomposition of the auxiliary lighting from xEMU is given as a notional example for context, illustrating the tool's usage from customer need to illustration of goals/subgoals with system and subsystem requirement listed with priority, requirement type and rationale listed.

Table 81 - Sample Requirement Matrix Decomposition for Auxiliary Lighting

Business Need/Description	Goals	Subgoals	Source of Requirement	System Requirement No. & Statement	Requirement Type	Prioritization	Rationale	Scenario(s)
N1: The customer wants the space suit to have its own lighting source for crew member visibility during EVA tasks.	G1: The xEMU should provide a lighting source for the space suit to provide crew with visibility during EVA tasks.	G1.1 The xEMU shall provide a primary lighting source to provide crew with visibility during EVAs.	J. Williams	R.PTRS.1: The xEMU shall provide a primary worksite lighting source to provide the crew with visibility during EVAs.	Functional	High	When an astronaut leaves the habitat, having the ability to see clearly during an EVA is critical. A reliable source that is a permanent fixture to the suit is vital to the success of the EVA and furthermore critical to the survival of the crew member.	The astronaut uses the primary lighting source located on the helmet to successfully complete routine tasking during a nominal EVA.

Business Need/Description	Subsystem Requirement No. & Statement	Rationale	Requirement Type	Prioritization	Rationale	Scenario(s)
N1: The customer wants the space suit to have its own lighting source for crew member visibility during EVA tasks.	R.INFO.2210: The Informatics lights shall emit illumination cones that introduces no more than 50 lux [TBR] into the suited crew members helmet bubble.	The majority of the flux (primary beam pattern) from the Informatics lamps should avoid significantly penetrating the helmet bubble of the suited crewmember.	Functional	High	When an astronaut leaves the habitat, having the ability to see clearly during an EVA is critical. A reliable source that is a permanent fixture to the suit is vital to the success of the EVA and furthermore critical to the survival of the crew member.	The astronaut uses the primary lighting source located on the helmet to successfully complete routine tasking during a nominal EVA.
	R.INFO.2211: The Informatics subsystem shall expose suited crewmembers to light intensities that are less than [TBD] nits.	Non-ionizing radiation can cause permanent damage to suited crew members by direct exposure and reflection of the informatics light on other surfaces including other components of the suit.				

Step 4: Scorecard Updates

The second iteration of the requirements scorecard was an incorporation of all aspects of the GtWR. This included all categories, such as requirement and requirement set characteristics, accuracy, concision, non-ambiguity, completeness, realism, conditions, uniqueness, abstraction, quantifiers, tolerance, quantification, uniformity of language, modularity, attributes (requirement definition and intent, verification or validation, maintaining organization). The lunar dust mitigation high level requirements were used as an example with this modified tool based on the best practices of the GtWR.

- *The xEVA System shall limit the amount of regolith liberated in the cabin environment to less than 100 grams for each two-crew lunar surface EVA.*

Table 82 – Sample Lunar Dust Best Practices Graded

Category	ID No.	Best Practices	Score (1-3)
Requirement Characteristics	C1	Is the requirement necessary?	3
	C2	Is the requirement appropriate?	3
	C3	Is the requirement unambiguous?	2
	C4	Is the requirement complete?	1
	C5	Is the requirement singular?	1
	C6	Is the requirement feasible?	1
	C7	Is the requirement verifiable?	1
	C8	Is the requirement correct?	2
	C9	Is the requirement conforming?	2
Requirement Set Characteristics	C10	Does the requirement set represent a complete set?	1
	C11	Does the requirement set demonstrate consistency?	2
	C12	Is the requirement set feasible?	1
	C13	Is the requirement set comprehensible?	3
	C14	Is the requirement set able to be validated?	1
Accuracy	R1	Is the sentence structured correctly?	3
	R2	Is the active voice used?	3
	R3	Are the subject and verb appropriate?	3
	R4	Are specific terms defined?	1
	R5	Is the definite article "the" used?	3
	R6	Are appropriate units used?	3
	R7	Are vague terms avoided?	2
	R8	Are escape clauses avoided?	3
	R9	Are open ended clauses avoided?	3
Concision	R10	Are superfluous infinitives avoided?	3
	R11	Are separate clauses used for each condition/quality?	3
Non-Ambiguity	R12	Is correct grammar used?	3
	R13	Is correct spelling used?	3
	R14	Is correct punctuation used?	3
	R15	Are "or" and "and" logical expressions used correctly?	3
	R16	Is the word "not" avoided	3
	R17	Is the oblique symbol avoided?	3
Singularity	R18	Do relevant subclauses quantify a singular thought?	0
	R19	Are combinators avoided?	2
	R20	Are extra phrases avoided to indicate purpose?	3
	R21	Are parenthesis and brackets containing subordinate text avoided?	3
	R22	Are sets enumerated explicably vs. using a group noun to set name?	0

	R23	Do complex requirements refer to supporting diagrams or models?	0
Completeness	R24	Are pronouns avoided?	3
	R25	Are headings to support explanation avoided?	3
Realism	R26	Are unachievable absolutes avoided	1
Conditions	R27	Are applicability conditions stated explicitly?	2
	R28	Is propositional nature of the condition expressed explicitly?	2
Uniqueness	R29	Is the requirement classified in accordance with problem to be addressed	3
	R30	Is this requirement not duplicated elsewhere?	3
Abstraction	R31	Does this requirement avoid stating a solution?	3
Quantifiers	R32	Is "each" used instead of "all/any/both"?	3
Tolerance	R33A	Are quantities appropriately defined?	1
	R33B	Are ranges appropriately defined?	1
Quantification	R34	Are measurable performance targets available and appropriate?	1
	R35	Are temporal dependencies explicitly defined	2
Uniformity of Language	R36	Are terms and units of measure used consistently throughout requirement sets?	0
	R37	Are consistent acronym sets used?	0
	R38	Are abbreviations avoided?	3
	R39	Is the project style guide used for the requirement?	0
Modularity	R40	Is the requirement grouped with like requirements?	0
	R41	Is the requirement conformed to a defined structure or template?	0
Attributes: Requirement Definition & Intent	A1	Does the requirement have an appropriate rationale?	2
	A2A	Is there a primary verification method?	1
	A2B	Is there a primary validation method?	1
	A3A	Is there a primary verification approach?	1
	A3B	Is there a primary validation approach?	1
	A4	Does the requirement trace to a parent?	3
	A5	Does the requirement trace to a source?	3
	A6	Does the requirement state conditions of use?	0
	A7	Does the requirement specify states and modes?	0
	A8	Is the requirement allocated properly?	3
Attributes: Verification or Validation	A9A	Does the requirement have a specified verification level?	2
	A9B	Does the requirement have a specified validation level?	2
	A10A	Does the requirement have a specified verification phase?	2
	A10B	Does the requirement have a specified validation phase?	2
	A11A	Does the requirement have a specified verification results?	2
	A11B	Does the requirement have a specified validation results?	2
	A12A	Does the requirement have a specified verification status?	2
	A12B	Does the requirement have a specified validation status?	2
Attributes: Maintaining Organization	A13	Does the requirement have a unique identifier?	3
	A14	Does the requirement have a unique name?	3
	A15	Does the requirement have an originator or author?	3
	A16	Does the requirement have a date of origin?	3
	A17	Does the requirement have an owner?	3
	A18	Does the requirement have a list of key stakeholders?	3
	A19	Does the requirement have a specified change board?	0
	A20	Does the requirement have a specified change status?	3
	A21	Does the requirement have a specified version number?	3
	A22	Does the requirement have a specified approval date?	3

	A23	Does the requirement have a specified date of last change?	3
	A24	Does the requirement have a high likelihood of stability?	1
	A25	Does the requirement have a specified a responsible person?	3
	A26	Does the requirement have verification status	3
	A27	Does the requirement have validation status	3
	A28	Does the requirement have a maturity assessment?	1
	A29	Does the requirement have a status?	0
	A30	Does the requirement have trace to interfaces?	2
	A31	Does the requirement trace to peer requirements?	2
	A32	Does the requirement have a priority?	3
	A33	Does the requirement have a criticality associated?	1
	A34	Does the requirement have a risk of implementation value assigned?	1
	A35	Does the requirement exist as part of a risk mitigation?	1
	A36	Does the requirement have a key driving need or requirement?	3
	A37	Does the requirement have an available comment section?	2
	A38	Does the requirement fall into a category?	3

2.28
TOTAL

The total score, 2.28 is an aggregate of all scored items with the exception of non-scored items. The panel of experts found that while this scoring method is simple, does not fully imply a direction nor an action for the systems engineers to take. While the requirement met, possibly met, didn't meet or the metric did not apply, the question the focus group asked was, "what is the benefit."

Step 5: Improvements & Limitations

Each iteration will present a series of improvements and limitations. For Iteration 2, there was no major improvement to the scoring model aside from the 0 to 3 scoring. Some improvements including a foray into checking requirement traceability, verifiability, validatability, writing correctness, usage of adequate terms and rationale. Limitations include many of the improvements instigated yet derives a means by which to achieve them. These include:

- How to develop a quantitative method to evaluate robustness of requirement.

    - Goal: successfully provide a scorecard based on all valuable parameters that dictate what a robust requirement needs to fulfill.
- The approach that must be taken to effectively allow team members to operate outside of MBSE tools in a capacity that gives clarity to requirement decomposition in a minimalist manner.
    - Goal: illustrate requirement decomposition in a lightweight matrix decomposition tool across needs to requirements to subsystem requirements to component requirements.
    - Mitigation: tool developed.
- Challenges associated with developing against the Con-Ops and ADD as some of the operations and usages were perceived as underdeveloped in the requirements building phase against current project documents.
    - Goal: develop a use case scenario tool.
    - Mitigation: tool developed.

- Issues persist with nomenclature (i.e., regolith definition, auxiliary lighting definition and intended usage).
    - Goal: develop a glossary tool.
    - Mitigation: tool developed.

Step 6: Overall Summary of Iteration

An abridged table is given to reflect the aforementioned iteration number, time duration, data gathering techniques, improvements and limitations.

Table 83 – Iteration 2B Summary

Iteration 2	Time Period	Data Gathering	Improvements	Limitations	Limitations Addressed
INCOSE Best Writing Practices Scorecard	February 2022 - May 2022	Intermediate Literature Review	Allows for checks on requirement traceability	No Quantitative Method to Evaluate Robustness of Requirement	
Glossary Tool		Case Study Review #2 Auxiliary Lighting	Allows for checks on requirement verifiability		
Use Case Tool		Brainstorming	Allows for checks on requirement validatability		
Decomposition Tool		Interviews	Allows for checks on requirement writing correctness		
		Focus Groups	Allows for checks on requirement adequate terms		
		Questionnaires	Allows for checks on requirement adequate rationale		
			Allows for Quick Requirement Decomposition	~~No Method for Decomposition Checks~~	Decomposition Tool Added
			Allows for a Method to Write Out Terms in a Glossary	~~No Method for Glossary of Terms~~	Glossary Tool Added
			Allows for a Method to Think Out Scenarios	~~No Method for Addressing Use Case Scenarios~~	Use Case Tool Added
				No Method of Template for User to Follow	

8.10. ITERATION 3B: Requirements Engineering Development

8.10.1. Iteration Development

In the third iteration, the requirements scorecard is modified to include the previous scorecard portion to exist as a screening with a more robust FMEA to finish the tool. This iteration was exclusively on the requirements scorecard. The final framework includes all previous iteration inclusions. The period of performance of Iteration 3 focused between May 2022 and October 2022. The data gathering techniques included:

- Final Literature Review: Where INCOSE requirements engineering concepts from academic, and peer reviewed journals are viewed.
- Case Study Review #2: Where the specific case studies of lunar dust mitigation and auxiliary lighting help support the notion of the development of a tool for requirement scoring. This iteration focused more on the information gathered for the lunar dust mitigation study.
- Brainstorming: Various, semi-formal sessions were elicited with stakeholders from across the project to assess which areas of systems engineering presented challenges and for project lifecycle development, those specific challenges documented.
- Interviews: Various, semi-formal sessions were elicited with stakeholders from across the project to assess which areas of systems engineering presented challenges and for INCOSE requirements engineering, those specific challenges documented.
- Focus Groups: Various, semi-formal sessions were elicited with stakeholders from across the project to assess which areas of systems engineering presented challenges and for INCOSE requirements engineering development, those specific challenges documented.
- Survey: A Likert scale survey to assess requirements engineering offerings on the project with responses from the Development Team and customer.

The goal of the iteration is to create a scorecard for requirements to test robustness. For context, robustness is a measure of how fit a requirement is for use against the most appropriate, current state-of-the-art. Literature review and expert elicitation on project indicate that INCOSE documents, including but not limited to the NRM, GtWR, GtNR and the GtVV would be leveraged to create the scorecard. The final product as a result of this iteration is the tool known as the RES.

A survey with 22 individuals across the project (customer and performing organization) were polled on the Likert scale to elicit information regarding waterfall sentiment on the project; 1 – Strongly Disagree, 5 – Strongly Agree; any score above a 3.5 (median between "Neither Agree or Disagree and Somewhat Agree) is deemed a positive indication on position; any score below a 3 is considered a negative indication on position. The survey below was used to compare the current vs. potential new requirements engineering developments to examine how malleable the team would be to a change in requirements engineering but also how successful the current approach remains. Results indicate that overall, the project found requirements engineering with regards to a scorecard to be plausibly in a position to be improved. After filtering results against the lunar dust mitigation efforts and auxiliary lighting requirements, there was a demand by project to improve requirements engineering scoring.

Table 84 -Requirement Scoring Likert Questions

QUESTION (Like Scale 1 to 5) 5 – Strongly Agree	Overall Project		Lunar Dust		Auxiliary Lighting	
There was a tool to illustrate when a requirements package completely met the customer's needs on the project	2.69	16	1.5	2	1	1
There was an appropriate rubric to build requirements on the project (scoring, grading, etc.)	2.5	14	1	2	1	1
Having a score card would be beneficial when developing requirements	3.81	16	4.5	2	5	1
We used INCOSE requirements standards on the project	3.85	13	3	1	1	1

Step 1: Establish Preliminary Screen for Robustness

After the requirement, requirement set or itemized areas of interested are investigated, the item(s) were graded against a preliminary scoring rubric. The scoring rubric is comprised of two items: (i) the scoring rubric grader and (ii) the best practices scorecard. The steps were: (i) select the requirement, (ii) review the best practices scorecard and (iii) grade selected best practice scorecard items against the scoring rubric grader with a score between 0-3. While the preliminary scoring rubric is tailorable to smooth coarseness of ranking, the xEMU case studies did not reconcile a specific ranking associated with one trait carrying more weight or a smoother ranking system. For the case studies examined, the requirement either a meets, does not meet, might meet or is not applicable to meeting a level of robustness. This allowed for a cursory examination which was adequate to identify coarse-grained robustness.

Table 85 – RES Scoring Rubric Grader

Scoring Rubric Grader		
Ranking	**Response**	**Criteria**
0	**N/A**	Not Applicable or Not Graded
1	**No/False**	Needs Corrective Action
2	**Maybe**	Consideration Given for Possible Corrective Action but Acceptable
3	**Yes/True**	Acceptable

The best practice scorecard allowed for grading of the requirement against any of the itemized categories. These categories associated identification number and best practice are all derived from GtWR. The user may utilize the template such that:

- the user may give an overall grade on specifically graded practices to derive a score with the average against only the items in question graded.
- the user may find areas that are given a Ranking of 1 or 2 and execute corrective action regardless of overall score.

- at the conclusion of grading, if the requirement or set of requirements does not meet the success criteria, FMEA grading will follow.

Table 86 – RES Best Practices Scorecard

Category	ID No.	Best Practices	Score (1-3)
Requirement Characteristics	C1	Is the requirement necessary?	
	C2	Is the requirement appropriate?	
	C3	Is the requirement unambiguous?	
	C4	Is the requirement complete?	
	C5	Is the requirement singular?	
	C6	Is the requirement feasible?	
	C7	Is the requirement verifiable?	
	C8	Is the requirement correct?	
	C9	Is the requirement conforming?	
Requirement Set Characteristics	C10	Does the requirement set represent a complete set?	
	C11	Does the requirement set demonstrate consistency?	
	C12	Is the requirement set feasible?	
	C13	Is the requirement set comprehensible?	
	C14	Is the requirement set able to be validated?	
Accuracy	R1	Is the sentence structured correctly?	
	R2	Is the active voice used?	
	R3	Are the subject and verb appropriate?	
	R4	Are specific terms defined?	
	R5	Is the definite article "the" used?	
	R6	Are appropriate units used?	
	R7	Are vague terms avoided?	
	R8	Are escape clauses avoided?	
	R9	Are open ended clauses avoided?	
Concision	R10	Are superfluous infinitives avoided?	
	R11	Are separate clauses used for each condition/quality?	
Non-Ambiguity	R12	Is correct grammar used?	
	R13	Is correct spelling used?	
	R14	Is correct punctuation used?	
	R15	Are "or" and "and" logical expressions used correctly?	

	R16	Is the word "not" avoided	
	R17	Is the oblique symbol avoided?	
Singularity	R18	Do relevant subclauses quantify a singular thought?	
	R19	Are combinators avoided?	
	R20	Are extra phrases avoided to indicate purpose?	
	R21	Are parenthesis and brackets containing subordinate text avoided?	
	R22	Are sets enumerated explicably vs. using a group noun to set name?	
	R23	Do complex requirements refer to supporting diagrams or models?	
Completeness	R24	Are pronouns avoided?	
	R25	Are headings to support explanation avoided?	
Realism	R26	Are unachievable absolutes avoided	
Conditions	R27	Are applicability conditions stated explicitly?	
	R28	Is propositional nature of the condition expressed explicitly?	
Uniqueness	R29	Is the requirement classified in accordance with problem to be addressed	
	R30	Is this requirement duplicated elsewhere?	
Abstraction	R31	Does this requirement avoid stating a solution?	
Quantifiers	R32	Is "each" used instead of "all/any/both"?	
Tolerance	R33A	Are quantities appropriately defined?	
	R33B	Are ranges appropriately defined?	
Quantification	R34	Are measurable performance targets available and appropriate?	
	R35	Are temporal dependencies explicitly defined?	
Uniformity of Language	R36	Are terms and units of measure used consistently throughout requirement sets?	
	R37	Are consistent acronym sets used?	
	R38	Are abbreviations avoided?	
	R39	Is the project style guide used for the requirement?	
Modularity	R40	Is the requirement grouped with like requirements?	
	R41	Is the requirement conformed to a defined structure or template?	
Attributes: Requirement Definition & Intent	A1	Does the requirement have an appropriate rationale?	
	A2A	Is there a primary verification method?	
	A2B	Is there a primary validation method?	
	A3A	Is there a primary verification approach?	
	A3B	Is there a primary validation approach?	
	A4	Does the requirement trace to a parent?	
	A5	Does the requirement trace to a source?	

	A6	Does the requirement state conditions of use?	
	A7	Does the requirement specify states and modes?	
	A8	Is the requirement allocated properly?	
Attributes: Verification or Validation	A9A	Does the requirement have a specified verification level?	
	A9B	Does the requirement have a specified validation level?	
	A10A	Does the requirement have a specified verification phase?	
	A10B	Does the requirement have a specified validation phase?	
	A11A	Does the requirement have a specified verification results?	
	A11B	Does the requirement have a specified validation results?	
	A12A	Does the requirement have a specified verification status?	
	A12B	Does the requirement have a specified validation status?	
Attributes: Maintaining Organization	A13	Does the requirement have a unique identifier?	
	A14	Does the requirement have a unique name?	
	A15	Does the requirement have an originator or author?	
	A16	Does the requirement have a date of origin?	
	A17	Does the requirement have an owner?	
	A18	Does the requirement have a list of key stakeholders?	
	A19	Does the requirement have a specified change board?	
	A20	Does the requirement have a specified change status?	
	A21	Does the requirement have a specified version number?	
	A22	Does the requirement have a specified approval date?	
	A23	Does the requirement have a specified date of last change?	
	A24	Does the requirement have a high likelihood of stability?	
	A25	Does the requirement have a specified a responsible person?	
	A26	Does the requirement have verification status?	
	A27	Does the requirement have validation status?	
	A28	Does the requirement have a maturity assessment?	
	A29	Does the requirement have a status?	
	A30	Does the requirement have trace to interfaces?	
	A31	Does the requirement trace to peer requirements?	
	A32	Does the requirement have a priority?	
	A33	Does the requirement have a criticality associated?	
	A34	Does the requirement have a risk of implementation value assigned?	
	A35	Does the requirement exist as part of a risk mitigation?	
	A36	Does the requirement have a key driving need or requirement?	

	A37	Does the requirement have an available comment section?	
	A38	Does the requirement fall into a category?	

When presented with this best practices scorecard, members of both the lunar dust mitigation and auxiliary lighting provided their own modified rubric to be graded.

Table 87 – RES Additional Practices Scorecard

Category	#	Additional Practices	Score (1-3)
Challenge Requirement	CR1	The requirement is not based around an existing feature.	
	CR2	The new requirement does not need to change existing designs?	
	CR3	Customer need cannot be met by simpler means?	
	CR4	There is not an alternative method available.	
	CR5	Requirement should not be challenged.	
	CR6	Requirement not implicit of a design solution.	
	CR7	Does the requirement limit the design potential?	
	CR8	Is the requirement constrained based on existing hardware?	
	CR9	Requirement does not require existing hardware to significantly change?	
	CR10	No alternative requirement possible to satisfy the need?	
Knowledge Availability	KA1	Requirements have a value that has an adequate basis?	
	KA2	Environment is understood in a way that is measurable for requirement feasibility?	
	KA3	Is project knowledge on the topic adequate enough to support this requirement?	
	KA4	Do vendor supplied parts meet the intent of the requirement?	
Operational Settings	OS1	Temporal setting doesn't affect the requirement?	
	OS2	Requirement does not initiate any emergent behavior in the system?	
	OS3	Environmental factors impact the requirement validation success?	
	OS4	An environment change during usage will not invalidate requirement?	
	OS5	Does the requirement consider cycling of usage?	
	OS6	Does the auxiliary lighting indicate a for use design?	
	OS7	Are the failure mode conditions fully comprehensible?	
Requirement Organization	RO1	Does the added requirement conflict with existing requirements?	
	RO2	Has the new change been approved by all stakeholders formally?	
	RO3	Are all glossary terms comprehensible to adequately build the requirement?	

	RO4	Requirement does not require a change in system context?	
	RO5	Requirement does not require a change in concept of operations?	
	RO6	Requirement does not require a change in the architecture design document?	
	RO7	Is upstream parent need or requirement not driving the downstream requirement?	
	RO8	This requirement does not need to be added in other subsystems?	
	RO9	This requirement does not have a TBX?	
Project Management	PM1	Changes to existing designs will not affect cost?	
	PM2	Changes to existing design will not affect schedule?	
	PM3	Changes to existing design will not impact quality standards?	
	PM4	Changes to existing design will not impact safety concerns?	
	PM5	The requirement does not hinge on more than one business need?	
	PM6	The requirement does not hinge on more than one customer?	
Verifications & Validations Concerns	VV1	Is the requirement range achievable?	
	VV2	During operation at the user level, can the requirement be validated?	
	VV3	Can the requirement environment be simulated?	
	VV4	Verifications do not need to be developed before writing the requirement?	
	VV5	Validations do not need to be developed before writing the requirement?	
	VV6	Can the cleanliness be verified before cleaning?	
	VV7	Can the cleanliness be validated after cleaning?	
	VV8	Can a simulant be used?	
	VV9	Can a simulant be made?	
	VV10	Additional testing does not need to be performed before the requirement is written?	
	VV11	Has any testing been performed prior to the requirement being written?	
	VV12	Has a testing method (feasible or not feasible) been defined?	

For example, the lunar dust and auxiliary lighting case study requirements were evaluated against the additional practices only for implementation in the intermediate screening. It is important to note that while these categories were selected for further risk management and scoring, they are not an exhaustive list and are strictly given for notional context for user. These scores should be collected while performing a focus group or brainstorming session with identified, relevant stakeholders.

Table 88 – Sample Requirement and Violations

Requirement	Additional Practice Categories Identified (Below Score of 3)
The xEVA System shall limit the amount of regolith liberated in the cabin environment to less than 100 grams for each two-crew lunar surface EVA.	RO3, VV1, VV2, VV3, VV4, VV5, VV6, VV7, VV8, VV9, VV10, VV11, VV12, PM1, PM2, PM4, PM6
The DCU emergency lighting shall provide 350 lumens of white light emitted across 4 source locations separated across the anterior surface of the DCU.	CR6, CR7, KA1, KA3, RO3

Step 2: Perform Intermediate Screen for Robustness

Once the scoring rubrics for both or either the best or additional practices have been reviewed, those selected requirements and categories were populated to the Risk and FMEA Tool (RFT). The tool works twofold both as a risk management tool and FMEA. The tool contains the following categories for the first step of the process:

- Requirement number
    - What is the requirement number?
- Requirement name
    - What is the name of the requirement?

- Requirement shall statement
    - What level does this requirement exist or would exist in?
- Challenge Description
    - What is the challenge or summary of challenges associated with the requirement?
- Best Practice or Additional Practice Description
    - What is the INCOSE Best Practice or Additional Best Practice Description?
- Potential Requirement Issue
    - What is the potential issue associated with moving forward with this requirement if unchanged?
- Potential Undesirable Effects
    - What are the immediate effects after decomposition if the requirement is not changed?
- Potential Next Level Effects
    - What are the next level effects if the requirement is not changed?
- Potential End Effects
    - What is the potential end effect if the requirement is not changed?

The potential requirement issue, undesirable effect, next level effects and end effects are meant to illustrate either or both a cascading failure representation and failures that may occur disjointed as a result of the current requirement state. The user may augment these categories, extend or abridge according to the user needs, as the focus group members had.

Step 3: Perform Final Screen for Robustness in FMEA

After population of the preliminary information was completed, an assessment of the following categories allowed for a risk posture to be established. An additional feature to this FMEA is the Reach category, which allows for an additional facet in understanding the criticality associated with the requirement's current robusticity position.

- Consequence: How severe is the impact should the risk manifest?
- Likelihood: What is the probability of this risk manifesting?
- Reach: What is the breadth and depth of this requirement impacting peripheral requirements?

Templates for each of the consequence, likelihood and reach categories are given and are tailorable for the user. In many cases, they categories are presented with a general, non-numerical value so that the user may modify them to suit their needs.

Table 89 - RES Best Practices Consequence Ranking

	Consequence Ranking for Best Practices				
Category	**1**	**2**	**3**	**4**	**5**
Requirement Characteristics	Remote risk of uncertainty	Minimal risk of characteristic uncertainty	Considerable risk of characteristic uncertainty	Major risk of characteristic uncertainty	Severe risk of characteristic uncertainty
Requirement Set Characteristics	Remote risk of set characteristic uncertainty	Minimal risk of set characteristic uncertainty	Considerable risk of set characteristic uncertainty	Major risk of set characteristic uncertainty	Severe risk of set characteristic uncertainty
Accuracy	Remote risk of requirement inaccuracy	Minimal risk of requirement inaccuracy	Considerable risk of requirement inaccuracy	Major risk of requirement inaccuracy	Severe risk of requirement inaccuracy
Concision	Superfluous infinities and separate clauses are clearly used	Superfluous infinities or separate clauses could cause minimal confusion	Superfluous infinities or separate clauses could cause considerable confusion	Superfluous infinities or separate clauses could cause major confusion	Superfluous infinities or separate clauses not used

Non-Ambiguity	Remote risk of ambiguity	Minimal risk of ambiguity	Considerable risk of ambiguity	Major risk of ambiguity	Severe risk of ambiguity
Singularity	Minor or no risk of singularity misuse	Minimal singularity misuse	Some singularity misuse	Major singularity misuse	Definite singularity misuse
Completeness	Remote or minor requirement incompleteness	Minimal requirement incompleteness	Considerable requirement incompleteness	Major requirement incompleteness	Severe lack of requirement completeness
Realism	Achievable absolutes avoided	Achievable absolutes mostly avoided	Achievable absolutes somewhat avoided	Achievable absolutes mostly avoided	Achievable absolutes not avoided
Conditions	Applicable conditions stated explicitly	Applicable conditions stated explicitly	Applicable conditions stated explicitly	Applicable conditions stated explicitly	Applicable conditions not stated explicitly
Uniqueness	No requirement duplicity and expresses problem to be addressed	Minor chance of requirement duplicity or unclear expression of problem to be addressed	Considerable chance of requirement duplicity or unclear expression of problem to be addressed	Major chance of requirement duplicity or unclear expression of problem to be addressed	Requirement duplicity and does not express problem to be addressed
Tolerance	Quantities and ranges properly defined	Quantities and/or ranges mostly well defined	Quantities and/or ranges considerably underdefined	Quantities and/or ranges highly underdefined	Quantities and ranges not properly defined
Quantification	Applicable and appropriate targets and temporal dependencies	Mostly applicable and appropriate targets and/or temporal dependencies	Considerable inapplicable and inappropriate targets and/or temporal dependencies	Mostly inapplicable and inappropriate targets and/or temporal dependencies	Inapplicable and inappropriate targets and temporal dependencies
Uniformity of Language	Language completely uniform	Language mostly uniform	Language somewhat non - uniform	Language highly non-uniform	Language completely non-uniform
Modularity	Requirement is grouped and conformed to defined structure.	Requirement is mostly grouped and conformed to defined structure.	Requirement is somewhat ungrouped and not conformed to defined structure.	Requirement is mostly ungrouped and not conformed to defined structure.	Requirement is not grouped or conformed to defined structure.
Attributes: Requirement Definition & Intent	Requirement definition and intent fully	Requirement definition and intent mostly comprehensible	Requirement definition and intent considerably lacking	Requirement definition and intent majorly lacking	Requirement definition and intent

	comprehensibl e				completely lacking
Attributes: Verification & Validation	Requirement verification and validation fully comprehensibl e	Requirement verification and validation mostly comprehensible	Requirement verification and validation considerably lacking	Requirement verification and validation majorly lacking	Requirement verification and validation completely lacking
Attributes: Maintaining Organization	Requirement organization fully comprehensibl e	Requirement organization mostly comprehensible	Requirement organization considerably lacking	Requirement organization majorly lacking	Requirement organization completely lacking

Table 90 – RES Additional Practices Consequence Ranking

	Consequence Ranking for Additional Practices				
Category	**1**	**2**	**3**	**4**	**5**
Project Management: Quality	Remote loss of quality	Minimal loss of quality	1 standard deviation away from quality standard	2 standard deviations away from quality standard	3 standard deviations away from quality standard
Project Management: Safety	Remote risk of injury	Minimal risk of injury	Minor injury	Severe injury	Loss of life
Project Management: Cost	< $50K impact	$50k to $100K impact	$100K to $250K impact	$250K to $500k impact	> $500K impact
Project Management: Scope	Remote impact to scope objectives	Minimal impact to scope objectives	Considerable impact to scope objectives	Major impact to scope objectives	Severe impact to scope objectives
Project Management: Schedule	Minor or no schedule impact	1-to-2-month impact	3-to-4-month impact	5-to-6-month impact	> 7-month impact to schedule
Requirement Organization	Minor or no risk of disorganizatio n	Minimal risk of disorganization	Some risk of disorganization	Major risk of disorganizatio n	Severe risk of disorganizatio n
Verifications & Validations	Minor or no risk of ambiguation	Minimal or possible inability to verify or validate requirement	Some inability to verify or validate requirement	Major inability to verify or validate requirement	Unable to verify or validate requirement in any capacity

Operational Setting	Operational setting fully understood	Operational setting possibly or minimally misunderstood	Operational setting somewhat understood	Operationally setting majorly misunderstood	Operational setting severely misunderstoo d
Knowledge Availability	Minor or no lack of project knowledge to substantiate	Minimal lack of project knowledge to substantiate	Considerable lack of project knowledge to substantiate	Major lack of project knowledge to substantiate	Severe lack of project knowledge to substantiate
Challenge Requirement	No need to challenge requirement	Requirement could be challenged but not strongly recommended	Requirement could be challenged	Requirement should most likely be challenged	Requirement should without doubt be challenged

Table 91 - RES Likelihood Ranking

Likelihood Ranking		
Score	**Description**	**Probability Range**
1	Very Unlikely	< 10 %
2	Unlikely	10% to 30%
3	Possible	> 30% to 60%
4	Likely	> 60% to 90%
5	Very Likely	> 90 %

Table 92 – RES Reach Ranking

Reach Ranking		
Score	**Description**	**Requirement Range**
1	Negligent Reach	Impacts no other requirements
2	Minor Reach	Impacts 1 requirement
3	Considerable Reach	Impacts 2-4 requirements
4	Major Reach	Impacts 5-10 requirements
5	Extensive Reach	Impacts 10+ requirements

At the conclusion of the consequence, likelihood and reach assignment was the population of the

Risk Priority Number (RPN). This RPN is given twice: once before analysis of alternatives and recommendations and once after analysis of alternatives or recommendations. The range is a number between 1 and 125. The template automatically populates a risk color associated with the degree of risk and requirement posture if left unmitigated. The RPN is a product of the three risk categories:

Risk Priority Number = Consequence x Likelihood x Reach

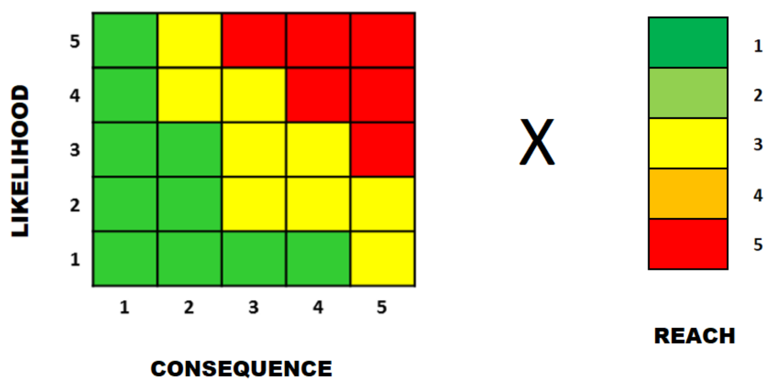


Figure 57 – Likelihood x Consequence x Reach Matrix

Step 4: Perform Final Analysis

After identification of the primary categories of requirement potential issues and effects and assignment of a RPN, the next steps were used to identify what risk mitigation efforts, if any, should be implemented.

- Action Recommended: What are the possible actions to remedy the requirement?
- Responsible Party: Who is responsible for making sure the actions are completed?
- Actions Taken: Will the Action Recommended be taken with respect to RPN?

If the user implements corrective actions in the form of alternatives or recommendations from the previous step, the user will update the RPN with the intention of reducing the risk posture of the requirement. As indicated previously, RPN is given twice: once before analysis of alternatives and recommendations and once after analysis of alternatives or recommendations. The range is a number between 1 and 125. The template will automatically populate a risk color associated with the degree of risk and requirement posture if left unmitigated.

The sample template of the RFT is shown below with lunar dust and auxiliary lighting requirements shown for context. Note that these are not exhaustive and only show values from the additional best practices list with specific items tailored from the lunar dust and auxiliary lighting requirements.

Table 93 – RES FMEA for Lunar Dust and Aux Lights Sample

Challenge Description	ID No.	Best Practice or Additional Practice Description	Potential Requirement Issue	Potential Undesirable Effects	Potential Next Level Effects	Potential End Effects	CONSEQUENCE (1-5)	LIKELIHOOD (1 - 5)	REACH (1 - 5)	RPN (1 - 125) (BEFORE ANALYSIS)	Action Recommended	Responsible Party	Actions Taken	CONSEQUENCE (1-5)	LIKELIHOOD (1 - 5)	REACH (1 - 5)	RPN (1 - 125) (AFTER ANALYSIS)
What is the challenge with the requirement?	What is the violation ID No.?	What is the violation description?	Potential issues with requirement if unchanged?	Immediate effect if requirement is not changed?	What are the next level effects if the requirement is not changed?	What is the potential end effect if the requirement is not changed?					What are the possible actions to remedy the requirement?	Who is responsible for making sure the actions are completed?	Will the Action Recommended be taken with respect to RPN?				
Definition of regolith	RO3	Are all glossary terms comprehensible to adequately build the requirement?	Will not be able to decompose requirements correctly to account for understanding of lunar dust/regolith	Space suit is not correctly designed to account for lunar dust/regolith	Space suit is not correctly tested (verified) and delivered (validated) to account for lunar dust/regolith	Space suit not fit for usage on the lunar surface.	4	5	5	100	Need to further define what regolith is, how it differs in various areas on the lunar surface, updating or creating a glossary definition.	A. Person	Yes (Projection)	2	2	5	20

Challenge Description	ID No.	Best Practice or Additional Practice Description	Potential Requirement Issue	Potential Undesirable Effects	Potential Next Level Effects	Potential End Effects	CONSEQUENCE (1-5)	LIKELIHOOD (1 - 5)	REACH (1 - 5)	RPN (1 - 125) (BEFORE ANALYSIS)	Action Recommended	Responsible Party	Actions Taken	CONSEQUENCE (1-5)	LIKELIHOOD (1 - 5)	REACH (1 - 5)	RPN (1 - 125) (AFTER ANALYSIS)
What is the challenge with the requirement?	What is the violation ID No.?	What is the violation description?	Potential issues with requirement if unchanged?	Immediate effect if requirement is not changed?	What are the next level effects if requirement is not changed?	What is the potential end effect if the requirement is not changed?					What are the possible actions to remedy the requirement?	Who is responsible for making sure the actions are completed?	Will the Action Recommended be taken with respect to RPN?				
4 locations is over prescriptive.	CR6	The requirement is not implicit of a design solution.	The 4 source locations and anterior surface designation could hamper design process, number of lamps is overly prescriptive.	Potential lower-level requirements will levy location restrictions and further constrain unit level components.	Design process could become more expensive and produce a product of lesser quality or modularity.	Design process could become more expensive and produce a product of lesser quality or modularity.	4	5	3	60	The DCU ~~emergency~~ auxiliary lighting shall provide average minimum illumination of ~~350 lumens of white light~~ 150 lux [TBD] at the angles and distances defined by the suited crewmember's visible two-handed work envelope for microgravity translation ~~emitted from at least 4 source locations separated across the anterior surface of the DCU~~. Crew two-handed work envelope is shown in Figure X.	SE&I Auxiliary Lighting Team	Yes	1	1	3	3

Once a posture on requirement robusticity is quantified, the data can be used to inform the following but not limited to:

- upper management as lessons learned as a project postmortem on existing requirements.
- upper management as a tool to be used during initial project requirement development for a new project.
- proposing alterations to current requirements or as a means to implement alternative directions with regards to design.
- as a means to inform project on possible cost, scope, schedule, quality and risk postures associated with the current requirement or requirement set.

It is recommended that the RFT be used to illustrate, when applicable, a root cause of an issue associated with a requirement or requirement suite or a trend that can be used to inform project on potential course correction(s).

The lunar dust mitigation group included a total of 30 unique requirements (above the single requirement used as an example in this iteration) with 169 total violations across those 30 requirements. A violation trend with regards to number of violations across all 30 unique lunar dust requirements were recorded to characterize a trend to inform project on a commonality to help address root causes. Once a root cause can be determined, corrective measures can be taken, which were suggested in the RFT. It is important to note that as expected, the highest-level requirements (Level 1) present the highest risk as they impact the most child requirements. Highest Offenders on the RPN included.

- If requirements can verified/validated (i.e., the ability to test against).
- If ranges are well defined (i.e., 100 g requirement).

- If requirement glossary and terms well defined (i.e., regolith definition, cabin environment).

The lunar dust group concluded that it may be possible that engineers do not have adequate information about lunar dust to effectively write requirements.

High level root causes included (paraphrased from focus group lunar dust experts):

- 100 g requirement in Level 2 (which affects the most requirements) too strict, hard to verify/validate and origin of calculation has been criticized.
- Cabin environment drove requirement which has challenged if the correct design was established and why cabin drove requirement.
- Regolith is a very broad terms and most of our lunar dust and simulants are not indicative of actual lunar dust.
- Current testing capabilities are extremely limited, and any testing so far has not yielded results that support a requirement that can be written for this domain.

Table 94 – Violations for Lunar Dust Requirements

Violation Description	Violation #	# of Occurrences
Is the requirement complete?	C4	1
Is the requirement conforming?	C9	3
Requirements have a value that has an adequate basis?	KA1	14
Changes to existing designs will not affect cost?	PM1	1

Changes to existing design will not affect schedule?	PM2	1
Changes to existing design will not impact safety concerns?	PM4	1
The requirement does not hinge on more than one customer?	PM6	1
Is the sentence structured correctly?	R1	10
Are "or" and "and" logical expressions used correctly?	R15	2
Are ranges appropriately defined?	R33B	16
Are measurable performance targets available and appropriate?	R34	8
Are specific terms defined?	R4	17
Are escape clauses avoided?	R8	1
Are all glossary terms comprehensible to adequately build the requirement?	RO3	21
Is upstream parent need or requirement not driving the downstream requirement?	RO7	9
This requirement does not have a TBX?	RO9	1
Is the requirement range achievable?	VV1	1
Additional testing does not need to be performed before the requirement is written?	VV10	12
Has any testing been performed prior to the requirement being written?	VV11	1
Has a testing method (feasible or not feasible) been defined?	VV12	12
During operation at the user level, can the requirement be validated?	VV2	1

Can the requirement environment be simulated?	VV3	1
Verifications do not need to be developed before writing the requirement?	VV4	1
Validations do not need to be developed before writing the requirement?	VV5	1
Can the cleanliness be verified before cleaning?	VV6	6
Can the cleanliness be validated after cleaning?	VV7	6
Can a simulant be used?	VV8	10
Can a simulant be made?	VV9	10

By observing the before and after FMEA results from the RFT, the lunar dust mitigation focus group concluded that the RFT RPN pre-mitigation could improve risk posture and effectively lower the RPN post-mitigation. While no actual corrective action was taken by the project due to resource constraints, RPN values before and after FMEA are shown below to simulate potential risk posture improvements if corrective actions were enacted.

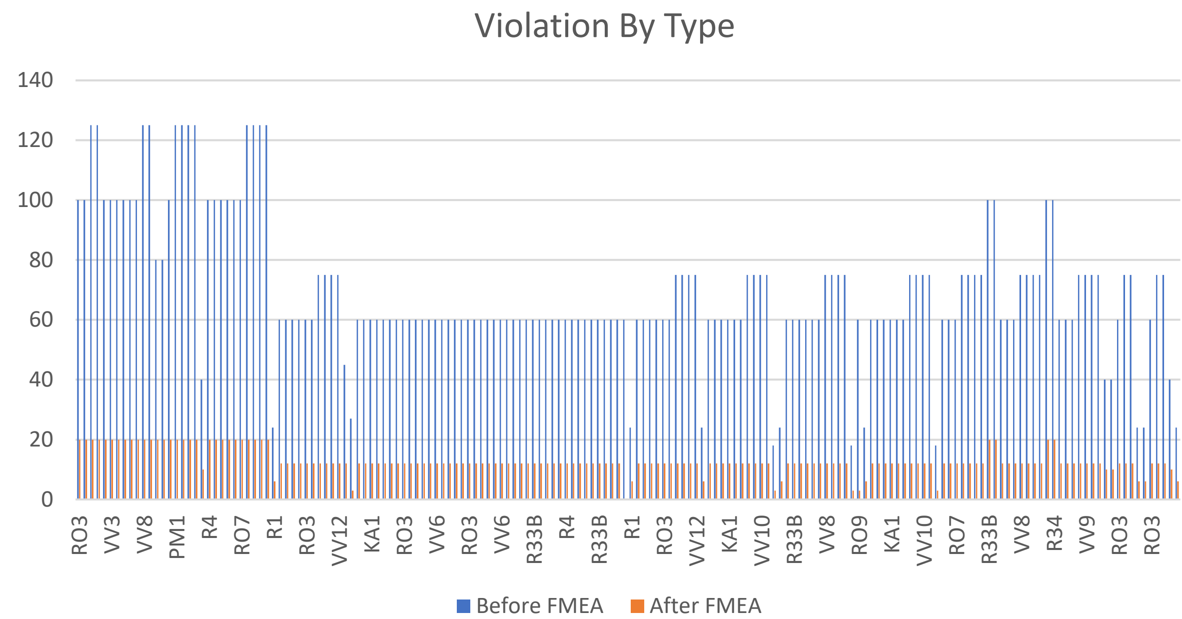


Figure 58 – Before and After RPN for Lunar Dust

The auxiliary lighting focus included a total of 5 unique requirements (above the single requirement used as an example in this iteration) with 21 total violations across those 5 requirements. A violation trend with regards to number of violations across all 5 unique auxiliary lighting requirements were recorded to characterize a trend to inform project on a commonality to help address root causes. Once a root cause can be determined, corrective measures can be taken, which were suggested in the RFT. It is important to note that as expected, the highest-level requirements (Level 1) present the highest risk as they impact the most child requirements. Highest Offenders on the RPN included.

- Does the requirement have a parent? (None of them do).
- If requirement glossary and terms well defined (i.e., lumens vs. lux, white light, chromaticity).

- An opinion from an engineer supporting: “In conclusion, all you need is a flashlight, we didn’t need to design an entire auxiliary lights system.”

The auxiliary lighting focus group concluded that it may be possible that engineers do not have adequate information about lunar dust to effectively write requirements.

High level root causes included (paraphrased from focus group lunar dust experts):

- No parent requirement and subsystem modified their equipment to satisfying additional lighting.
- Stakeholders eventually dictated that a flashlight was sufficient.
- Now, the current auxiliary lighting has introduced more challenges (looping into wiring of another system, direct lighting shining on adjacent crew members)

Table 95 - Violations for Aux Lights Requirements

Violation Description	Violation #	# Of Occurrences
Are all glossary terms comprehensible to adequately build the requirement?	RO3	2
Does the requirement trace to a parent?	A4	3
Requirement has a value that has an adequate basis?	KA1	3
Environment understood in a way that is measurable for requirement feasibility?	KA2	4

Requirement not implicit of a design solution.	CR6	4
Does the requirement limit the design potential?	CR7	5

By observing the before and after FMEA results from the RFT, the auxiliary lighting focus group concluded that the RFT RPN pre-mitigation could improve risk posture and effectively lower the RPN post-mitigation. While no actual corrective action was taken by the project due to resource constraints, RPN values before and after FMEA are shown below to simulate potential risk posture improvements if corrective actions were enacted.

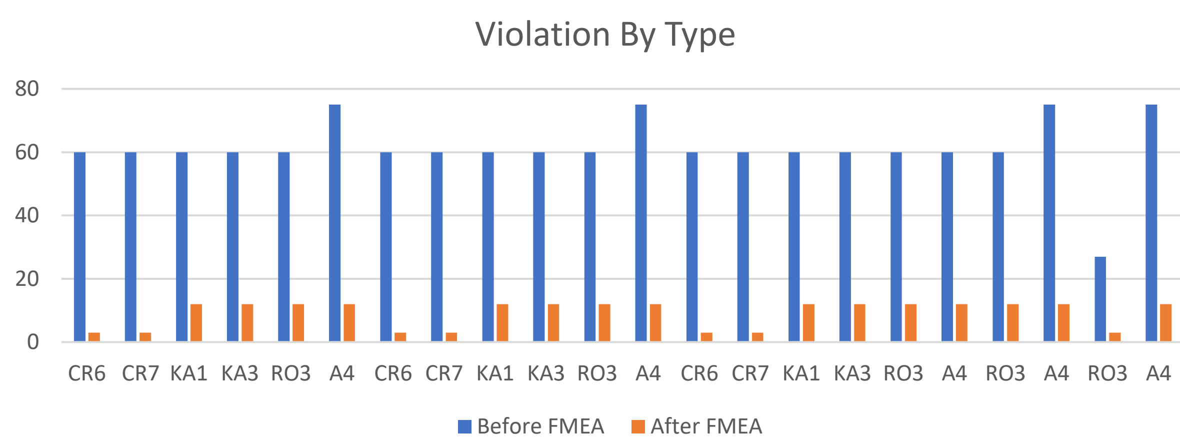


Figure 59 – Before and After RPN Value for Aux Lights

Step 5: Improvements & Limitations

Each iteration will present a series of improvements and limitations. For Iteration 3, there was a major improvement to the scoring model in addition to the 0 to 3 scoring as a preliminary screening, the FMEA tool allowed for a quantitative method to inform project management and systems engineers the robustness of their requirement sets. Limitations were fully closed in this iteration. These included:

- How to develop a quantitative method to evaluate robustness of requirement.
    - Goal: successfully provide a scorecard based on all valuable parameters that dictate what a robust requirement needs to fulfill.
    - Mitigation: requirements engineering scorecard created.
- The approach that must be taken to effectively allow team members to operate outside of MBSE tools in a capacity that gives clarity to requirement decomposition in a minimalist manner.
    - Goal: illustrate requirement decomposition in a lightweight matrix decomposition tool across needs to requirements to subsystem requirements to component requirements.
    - Mitigation: tool developed.
- Challenges associated with developing against the Con-Ops and ADD as some of the operations and usages were perceived as underdeveloped in the requirements building phase against current project documents.
    - Goal: develop a use case scenario tool.
    - Mitigation: tool developed.

- Issues persist with nomenclature (i.e., regolith definition, auxiliary lighting definition and intended usage).
    - Goal: develop a glossary tool.
    - Mitigation: tool developed.

Step 6: Overall Summary of Iteration

An abridged table is given to reflect the aforementioned iteration number, time duration, data gathering techniques, improvements and limitations.

Table 96 - Iteration 3B Summary

Iteration 2	Time Period	Data Gathering	Improvements	Limitations	Limitations Addressed
INCOSE Best Writing Practices Scorecard	May 2022 - October 2022	Final Literature Review	Allows for checks on requirement traceability	No Quantitative Method to Evaluate Robustness of Requirement	
Glossary Tool		Case Study Review #2 Auxiliary Lighting	Allows for checks on requirement verifiability		
Use Case Tool		Brainstorming	Allows for checks on requirement validatability		
Decomposition Tool		Interviews	Allows for checks on requirement writing correctness		
		Focus Groups	Allows for checks on requirement adequate terms		
		Questionnaires	Allows for checks on requirement adequate rationale		
			Allows for Quick Requirement Decomposition	No Method for Decomposition Checks	Decomposition Tool Added
			Allows for a Method to Write Out Terms in a Glossary	No Method for Glossary of Terms	Glossary Tool Added
			Allows for a Method to Think Out Scenarios	No Method for Addressing Use Case Scenarios	Use Case Tool Added
				No Method of Template for User to Follow	User Manual Added

8.11. ITERATION 4B: Requirements Engineering Development

8.11.1. Iteration Development

This iteration started with an additional model tempering. One of the principal authors of the INCOSE GtWR indicated that the following would be helpful:

- A similar scorecard in the style of the Best Practices and Additional Practices for needs activities in the spirit of the INCOSE GtNR and NRM manuals.
- Utilizing the scoring of requirements and less focus during trials on the glossary, decomposition and use case tools as the scoring of requirements was deemed by the author as novel and a place for dedicated focus.

Samples of those tables which are done before the requirement scoring is provided below. These are to be done prior (and are optional based on the need of the user(s)). Scoring is on the same 1-3 scale. The Requirements Checklist also has a filtering system to see which parts of the subsequent scorecards are most applicable:

Table 97 – Needs Preliminary Sweep

Document	Category	ID No.	Needs Preliminary Sweep	Score (1-3)
INCOSE Guide to Needs and Requirements May 2022, Appendix D1 & D2	**Requirement (Need)Characteristics**	D1.1	Are the need expressions well-formed?	
	Attributes: Requirement Definition & Intent	D1.2	Does each need expression contain a complete set of attributes?	
	Attributes: Verification or Validation	D1.3	Need expression is well formed such that system will be validated to meet need?	
	Requirement (Need) Characteristics	D2.1	Needs are correct?	
		D2.2	Set of needs is complete?	
		D2.3	Set of needs is feasible?	
		D2.4	Integrated set of needs is feasible?	
		D2.5	Integrated set of needs is correct?	

	Attributes: Maintaining Organization	D2.6	What is necessary for acceptance has been defined and agreed to?	
		D2.7	The needs associated with interfaces are well formed to be validated to meet needs?	

Table 98 – Requirements Checklist

INCOSE Document	Section	#	Needs & Requirements, Guide to Needs & Requirements Checklists	Score (1-3)	If Scored 1 or 2, Consider Reviewing:
INCOSE Needs and Requirements Manual May 2022	Section 7.1.2 Checklist	7.1.2.1	Were individual requirements expressions manually verified and the sets of requirements have the characteristics in accordance with the rules defined in the INCOSE GtWR [19] or similar guide.		Best Practices: C1 Thru C9
		7.1.2.2	Do the set of requirements contain individual requirements that are unique, do not conflict with or overlap with other requirements in the set, and the units and measurement systems they use are homogeneous?		Best Practices: C10 Thru C14
		7.1.2.3	Is the language used within the set of requirements consistent and all terms used within the requirement statements are consistent with the architectural model and project data dictionary?		Best Practices: R1 Thru R40, Additional Practices: Requirement Organization
		7.1.2.5	Do individual requirement expressions have the set of attributes agreed to by the project team defined?		Best Practices: All Attribute Categories
		7.1.2.6	Do individual requirement expressions have the set of system verification attributes agreed to by the project team defined?		Best Practices: Attributes: Verification or Validation, Additional Practices: All Categories (Minus Project Management)
		7.1.2.9	Were all interfaces addressed and the associated interface requirements included in the requirement set?		Best Practices: Attributes: Verification or Validation, Additional Practices: All Categories (Minus Project Management)
		7.1.2.10	Is there clarity regarding specific interactions between the SOI and the external system, and that the requirement includes a pointer to where that interaction is defined, recorded, and agreed to?		Best Practices: All Attribute Categories, Additional Practices: All Categories (Minus Project Management)
		7.1.2.11	Does the external system referred to have a corresponding interface requirement or includes the interaction with the SOI being developed in its interface control documentation?		Best Practices: All Attribute Categories, Additional Practices: All Categories (Minus Project Management)

		7.1.2.12	Does the requirement properly address form, fit, function, quality, and compliance?		Best Practices: All Attribute Categories, Additional Practices: All Categories (Minus Project Management)
		7.1.2.14	Is each allocation correct and complete (i.e., the requirements were allocated to all applicable subsystems and system elements at the next level of the architecture and each of the allocations were to the correct subsystems and system elements)?		Best Practices: All Attribute Categories, Additional Practices: All Categories (Minus Project Management)
		7.1.2.15	Are the resulting dependent child requirements in response to allocations of performance, quality, or resources properly linked to manage changes to the allocated/budgeted values?		Additional Practices: Project Management

After the tempering based on the author of the GtWR comments were made, testing began outside the NASA teams to help vet the model. Several meetings were conducted to elicit feedback, both within Jacobs, NASA and INCOSE groups. Due to constraints on Jacobs and NASA personnel previously used as test subjects and limited feedback from INCOSE individuals to be test subjects (mostly due to non-disclosures on requirements, availability), two sections of a senior design mechanical engineering class at Texas A&M were available to participate. The following tests were run with the RES in its entirety:

- 7 unique groups brough their requirements to be graded for robustness using the RES.
- A total of 25 student participants.
- A 3-to-4-hour block was available for students to participate.
- Results include a customer satisfaction survey and pre/post FMEA RPN to both facilitate the vetting of the tool.
- The typical test ran as follows:
    - General meet and greet of students and RES guide as the facilitator (10 mins).
    - Walkthrough of the agenda (5 mins).
    - Student description of project and presentation of requirement set (30 mins).

- RES tool commencement (2-3 hours):
  - Needs Sweep scoring.
  - Requirements Checklist scoring.
  - Requirements selected to be checked for robustness.
  - Best Practices Scorecard scoring.
  - Additional Practices Scorecard Scoring (if applicable).
  - Violations tallied and requirements fed into FMEA spreadsheet.
  - FMEA performed for each requirement and/or violation detected.
- Survey distribution (10 mins) and meeting adjournment.

Testing was held over a series of two months and the RES was not augmented nor the process listed in the "RES tool commencement" to control variation amongst test subjects and groups. While non-disclosures were not obtained to illustrate requirements and their travel through the tool, FMEA RPN scores and customer satisfaction surveys were obtained to gain metrics on vetting of the RES. Likert scales are as follows: 1: Strongly Disagree. 5: Strongly Agree. Deltas were checked between the scores and areas to avoid or exploit were determined.

Table 99 – Texas A&M Student Likert Scale Scores

#	Question	Min	Max	Mean	Std Deviation	Var	Count
1	This tool helped identified short comings in the requirements database	5	5	5	0	0	25
2	There were short comings in our requirement database	3	5	4.76	0.51	0.26	25
3	I found the Needs Preliminary Sweep to be helpful	2	5	4.56	0.85	0.73	25
4	I found the Requirements Preliminary Sweep to	3	5	4.6	0.63	0.4	25

	be helpful						
5	I found the Best Practices Scorecard to be helpful	3	5	4.4	0.85	0.72	25
6	I found the Additional Practices Scorecard to be helpful	3	5	4.1	0.7	0.49	10
7	I found the Reach Matrix to be helpful	2	5	4.67	0.8	0.64	24
8	I found the Consequence Matrix to be helpful	4	5	4.79	0.41	0.16	24
9	I found the Likelihood Matrix to be helpful	3	5	4.79	0.5	0.25	24
10	I found the FMEA Spreadsheet to be helpful	4	5	4.88	0.33	0.11	24
11	This tool will be helpful prior to requirement building activities	2	5	4.56	0.75	0.57	25
12	This tool will be helpful during requirement building activities	3	5	4.72	0.53	0.28	25
13	This tool will be helpful as a postmortem to requirements building	3	5	4.67	0.55	0.31	24
14	This tool improved my approach on requirements building	4	5	4.8	0.4	0.16	25
15	I found this tool easy to use	2	5	4	0.98	0.96	25
16	I would recommend this tool to my performing organization	4	5	4.6	0.49	0.24	25
17	This tool will improve requirements building activities	4	5	4.84	0.37	0.13	25
18	This tool will improve project management activities	2	5	4.54	0.76	0.58	24
19	This tool will improve risk management activities	4	5	4.8	0.4	0.16	25

20	Recommended Actions we discovered will help mitigate requirement risk(s)	3	5	4.75	0.6	0.35	24
21	This tool is properly built in the intention of INCOSE best practices	3	5	4.5	0.72	0.52	22

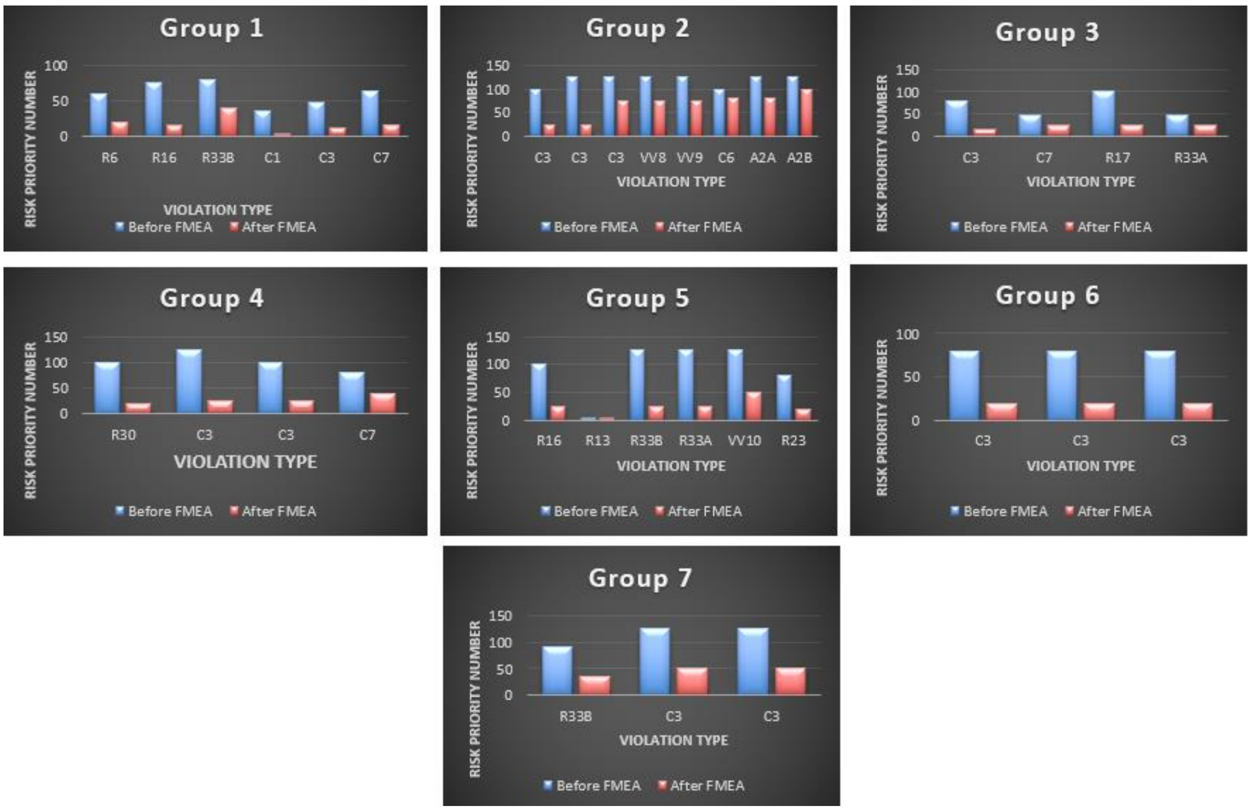


Figure 60 – Texas A&M Before/After FMEA on RPNs

A total of 15 requirements discovering 34 violations were yielded and in all cases, the RPN was lowered after potential mitigations applied. While the tool was favorably accepted by the students in the survey and the RPNs did indeed reduce with the RES tool, a filtering of the less satisfied students revealed the following items to be less than desirable.

- Wording confusing on several of the scoring questions.
- Length of the process, specifically the FMEA.
- No automation.

- The learning curve, especially if there is no facilitator to guide the process.

The tool was then augmented to simplify wording. The length, automation and learning curve have not been optimized as FMEAs are inherently lengthy, the learning curve is a function of how well individuals are versed in FMEAs and INCOSE products and per the GtWR author's comments, while certain aspects of the process could be and have been automated, understanding if requirements are correct, unambiguous, verifiable, etc. cannot be automated without a discussion in a panel with all the pertinent stakeholder teams.

The following items on the RES were reworded or full tables shown if items were deleted:

Table 100 – Needs Preliminary Sweep Rewording & Deletes Updates

ID No.	Needs Preliminary Sweep
D1.1	Are need expressions well-formed?
D1.2	Does each need expression contain a complete set of attributes?
D1.3	Are need expressions adequate to validate the need?
D2.1	Needs are correct?
D2.2	Set of needs is complete?
D2.3	Set of needs is feasible?
D2.6	What is necessary for customer acceptance has been defined and agreed to?
D2.7	The needs associated with interfaces are well formed to validate the needs?

Table 101 – Requirements Checklist Rewording & Deletes Updates

#	Needs & Requirements, Guide to Needs & Requirements Checklists
7.1.2.1	Were requirements written with regards to applicable INCOSE documents? (See RES Manual for applicable documents).
7.1.2.2	Are requirements unique, do not conflict with or overlap with other requirements in the set, and the units and measurement systems consistent?
7.1.2.3	Is the language used within the set of requirements consistent and all terms used within the requirement statements are consistent with project glossary?

7.1.2.5	Do requirements have the set of attributes defined or agreed to by the project team?
7.1.2.6	Do requirements have the set of system verification attributes agreed to by the project team defined?
7.1.2.9	Were all interfaces addressed in the requirement set?
7.1.2.12	Does the requirement properly address form, fit, function, quality, and compliance?

Table 102 – Best Practices Rewording Updates

R30	Is this requirement not duplicated elsewhere?

Table 103 – Additional Practices Rewording Updates

CR7	The requirement does not limit the design potential?
KA1	Requirement has a value that has an adequate basis.
OS1	Temporal settings won't affect the requirement.
OS3	Environmental factors do not impact the requirement validation success?
RO1	The added requirement does not conflict with existing requirements.
RO7	Is upstream parent need or requirement driving the downstream requirement?

As a result of testing and post preliminary exam work, *Project Lifecycle Development for a Next Generation Space Suit Project* article and *Requirements Engineering Scorecard and the Next Generation Space Suit* (Cabrera, et al, 2023) articles were published.

9. CHAPTER 9: CONCLUSION

In order to fully illustrate the conclusion of the case study's findings, a re-examination of the research questions are provided.

Research Question #1:

Can the lunar dust and auxiliary lighting suit requirements' challenges be resolved by applying a characterization against INCOSE writing practices to guarantee robustness via current state-of-the-art requirements development methods in a scorecard?

Research Question #2:

Will a modified, agile-hybrid project lifecycle development model applied to waterfall teams develop a superior product within time and schedule constraints in a hardware-intensive environment?

With regards to Research Question #1, hardware-intensive environments which typically follow waterfall may be suitable to additions to agile and Lean-based strategies. Further testing of the MAC prototype to establish feasibility is recommended. With regards to Research Question #2, there is a need for scoring requirements to promote superior products and help guide engineers with key driving requirements. While other scoring tools exist, those perform a more cursory view of sentence structure, proper ranges/tolerances/quantities. The RES could be a compliment to these other tools by providing context to the true nature of the requirement's intent and help guarantee robustness. While the RES lacks automation and requires conversations to elicit the intent of the requirement, it has shown most effective as an introductory tool to individuals for understanding INCOSE standards and assist projects' key driving requirements.

NASA is the institution that develop systems engineering to address complex issues arising from an everchanging landscape where cohesion between interdisciplinary groups could not go unmitigated. A central theme across the primary case studies is communication. As it applies to requirements engineering, the ability to effectively organize, document and communicate information is where many of the challenges resided, whether it's organizing an entire project scope in a series of interconnected documents or creating a model or using natural language to communicate customer needs. As was the case with the Agile study, constant communication allows for ideas to be shared, impediments to be removed, employees to be effectively groomed and a higher chance of success as it pertains to the product in question. As the project lifecycle is comprised of many interdisciplinary groups, xEMU is no different and aside from the branch of systems engineering, other contributing factors led to challenges across the project landscape, in particular project management. Another challenge came to the abundance of process; a hallmark of systems engineering drawing its roots from the NASA program itself, and while it important, should never abstain a team from accountability. Two consequences identified by Slegers, et were the potential to remove accountability by strictly relying on process and misplaced effort, which is to express the idea of determining the best way to complete work and merely not an over emphasis and higher reliance on how the work should be completed in terms of process (Slegers, 2011), If we examine the shift in system context as dramatic as it were, the movement of schedule two years early and the resource allocation of many untrained and inexperienced engineers, a proximate and unquantified assumption based on the qualitative and subjective nature is that while many challenges and instances for optimization could have been executed, the project as a whole performed as satisfactory as one could expect, especially given the unique and overcomplex nature of constructing a lunar suit for the moon and possibly beyond.

REFERENCES

Albero Pomar, Filipe et al. (2014). "Understanding Sprint Velocity Fluctuations for Improved Project Plans with Scrum: a Case Study." Journal of Software : Evolution and Process (pp. 776–783).

Alliance, A. G. I. L. E. (2017). Agile Practice Guide, Project Management Institute, 2017: Agile Practice Guide (Vol. 1). Bukupedia.

Ambler, S. W., & Lines, M. (2012). Disciplined agile delivery: a practitioner's guide to agile software delivery in the enterprise. Upper Saddle River, NJ: IBM Press.

Afshar-Mohajer, N., Wu, C. Y., Curtis, J. S., & Gaier, J. R. (2015). Review of dust transport and mitigation technologies in lunar and Martian atmospheres. Advances in Space Research, 56(6), 1222-1241.

Agui, J. H., Erickson, C. Y., & Perry, J. L. (2011). Analytical and Computational Particulate Load Models in Planetary Surface Spacecraft Cabins. https://doi.org/10.2514/6.2011-5046.

Agui, J.H., & Stocker, D.P. (2009). NASA Lunar Dust Filtration and Separations Workshop Report. ID: 127511578.

Atzberger, A., & Paetzold, K. (2019, July). Current challenges of agile hardware development: What are still the pain points nowadays?. In Proceedings of the Design Society: international conference on Engineering Design (Vol. 1, No. 1, pp. 2209-2218). Cambridge University Press.

Atzberger, A., Gerling, C., Schrof, J., Schmidt, T. S., Weiss, S., & Paetzold, K. (2019, June). Evolution of the hype around agile hardware development. 2019 IEEE International Conference.

Augustin, L., & Schabacker, M. (2019, July). Combining Scrum and Design Thinking for a highly iterative and user-centric hardware development project. In Proceedings of the Design Society: International Conference on Engineering Design (Vol. 1, No. 1, pp. 2189-2198). Cambridge University Press.

Bahill, A. Terry, & Henderson, S. J. (2005). Requirements development, verification, and validation exhibited in famous failures. Systems engineering, 8(1), (pp.1-14).

Bahr, R., Barrett, C., Bhagdikar, N., Carsello, A., Daly, R., Donovick, C. & Zhang, K. (2020, July). Creating an agile hardware design flow. In 2020 57th ACM/IEEE Design Automation Conference (DAC) (pp. 1-6). IEEE.

Bue, G. C., Makinen, J. V., Miller, S., & Campbell, C. (2014). Spacesuit Water Membrane Evaporator; An Enhanced Evaporative Cooling System for the Advanced Extravehicular Mobility Unit Portable Life Support System. TFAWS Active Thermal Paper Session, Thermal & Fluids Analysis Workshop (GRC 2014).

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunning-ham, W., Fowler, M., et al. (2001). Manifesto for agile soft-ware development.

Beck, K. (2003). Test-driven development: by example. Addison-Wesley Professional.

Blanchard, B. S. (2004). System engineering management. John Wiley & Sons.

Blanchard, Benjamin S., Wolter J. Fabrycky, and Walter J. Fabrycky. (1990) Systems engineering and analysis. Vol. 4. Englewood Cliffs, NJ: Prentice Hall.

Borky, J. M., & Bradley, T. H. (2018). Effective model-based systems engineering. Springer.

Brenner, R. & Wunder, S. (2015). Scaled agile framework: Presentation and real world example. 2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW), IEEE, pp. 1–2.

Briatore, S., & Golkar, A. (2021). Estimating Task Efforts in Hardware Development Projects in a Scrum Context. IEEE Systems Journal, 15(4), 5119-5125.

Burkhard, D. G., & Shealy, D. L. (1981). Simplified formula for the illuminance in an optical system. Applied Optics, 20(5), 897-909.

Cabrera, M., Simske, S. (2023, June 28). Project Lifecycle Development for a Next Generation Space Suit Project. INCOSE Insight, Volume 26, Issue 2, pages 57
65. https://doi.org/10.1002/inst.12447.

Cabrera, M., Simske, S., Worrell, J. (2023, July 16). Requirements Engineering Scorecard and the Next Generation Space Suit. 2023 52nd International Conference of Environmental Systems, Session 401. ICES-2023-25. https://ttu-ir.tdl.org/handle/2346/94482.

Campbell, C. (2017). Schematics and Behavioral Description for the Advanced EMU (AEMU) Portable Life Support Subsystem (PLSS) (No. JSC-CN-40493).

Carson, R. S. (2015, October). Implementing structured requirements to improve requirements quality. In INCOSE International Symposium (Vol. 25, No. 1, pp. 54-67).

Carson, Ronald (2021, June). Developing Complete and Validated Requirements. INCOSE Seattle Metropolitan Chapter Conference, Seattle Pacific University. 10.13140/RG.2.28526.74561.

Chen, R., Ravichandar, R., Proctor, D. (2016). Managing the transition to the new agile business and product development model: Lessons from Cisco Systems. Kelly School of Business, Business Horizons. Bushor-1315, 1-10.

Cherry, K. (2012). Multitasking: The cognitive costs of multitasking. About Cognitive Psychology. http://psychology. about. com/od/cognitivepsychology/a/costs-of-multitasking. htm.

Cimatti, Roveri, M., Susi, A., & Tonetta, S. (2012). Validation of requirements for hybrid systems: A formal approach. ACM Transactions on Software Engineering and Methodology, 21(4), 1–34. https://doi.org/10.1145/2377656.2377659.

Cockburn, A. Agile Software Development, Addison-Wesley, 2002.

Cohen, D., Lindvall, M., & Costa, P. (2004). An introduction to agile methods. In M. V. Zelkowitz (Ed.), Advances in computers vol. 62: Advances in software engineering. Cambridge, UK: Academic Press.

Dalton, Jeff. "Definition of Done."Apress, Berkeley, CA, 2018, pp. 159–161.

Crowe, A. (2018). The PMI-ACP exam: How to pass on your first try. Velociteach.

Davis, K., & Meginnis, I. (2019, July). Testing of the NASA Exploration Extravehicular Mobility Unit Demonstration (xEMU Demo) Architecture at the Neutral Buoyancy Laboratory. 49th International Conference on Environmental Systems.

Davis, N. (2013, August). Driving quality improvement and reducing technical debt with the definition of done. In 2013 Agile Conference (pp. 164-168). IEEE.

Delano, J. W., & Livi, K. (1981). Lunar volcanic glasses and their constraints on mare petrogenesis. Geochimica et Cosmochimica Acta, 45(11), 2137-2149.

Delligatti, L. (2013). SysML distilled: A brief guide to the systems modeling language. Addison-Wesley.

Dick, J., Hull, E., & Jackson, K. (2017). Requirements Engineering (4th ed.). Springer International Publishing.

Doran, G. T. (1981). There's a S.M.A.R.T. way to write management's goals and objectives. Management Review, 70, 35.

Drechsler, R. (2018). Formal system verification. Cham: Springer.

Fanmuy, G., Fraga, A., & Llorens, J. (2012). Requirements verification in the industry. In Complex systems design & management (pp. 145-160). Springer, Berlin, Heidelberg.

Force, U. A. (2005). SMC Systems Engineering Primer and Handbook. Space & Missile Systems Center.

Gaspard, C. (2021, April 14). Exploration Extravehicular Activity Services (xEVAS) Request for Information (RFI). https://www.nasa.gov/jsc/procurement/xevas.

Garzaniti, N., & Golkar, A. (2020). Performance Assessment of Agile Hardware Co-development Process. In 2020 IEEE International Symposium on Systems Engineering (ISSE) (pp. 1-6). IEEE.

Gonzalez, Gustavo, Horita, Flavia. “Proceedings of the XXXIII Brazilian Symposium on Software Engineering.” Emergent Behavior in System-of-Systems: A Systematic Mapping Study, 23 Sept. 2019, https://doi.org/10.1145/3350768.

Goodman, G. (1991). Information requirements for space station freedom EVA (No. 911526). SAE Technical Paper.

Gottesdiener, E. (2003). Use Cases: best practices. Rational Software white paper.

Griffiths, Mike. (2012). PMI-ACP Exam Prep. RMC Publications.

Gustavsson, T. (2016). Benefits of agile project management in a non-software development context: A literature review. In Fifth International Scientific Conference on Project Management in the Baltic Countries, April 14-15, 2016, Riga, University of Latvia (pp. 114-124). Latvijas Universitate.

Hause, M., & Thom, F. (2002). Using Use Cases and Scenarios as Part of a Safety Case. In Proceedings of the “Triple SE”(Safe and Secure Systems) Engineering Symposium, INCOSE UK Chapter, Spring Symposium.

Hause, M. (2006, September). The SysML modelling language. In Fifteenth European Systems Engineering Conference (Vol. 9, pp. 1-12).

Hekkala, R., Stein, M-K., Rossi, M., & Smolander, K. (2017). Challenges in Transitioning to an Agile Way of Working. In Proceedings of the Hawaii International Conference on System Sciences (pp. 5869-5878). IEEE.

Herzberg, Frederick; Mausner, Bernard; Snyderman, Barbara B. (1959). The Motivation to Work (2nd ed.). New York: John Wiley.

Hilken, F., & Hamann, L. (2020). History of the USE Tool 20 Years of UML/OCL Modeling Made in Germany. J. Object Technol., 19(3), 3-1.

Höglund, C. (2017). Guidelines for UML or SysML modelling within an enterprise architecture.

Hood, C., Wiedemann, S., Fichtinger, S., & Pautz, U. (2007). Requirements management: The interface between requirements development and all other systems engineering processes. Springer Science & Business Media.

Hooks, I. F. and Farry, K. A. (2001). Customer-Centered Products: creating successful products through smart requirements management; AMACOM Books, NY, NY.

Hunt, J. A. (2018). PMI-ACP project management institute agile certified practitioner exam study guide. Sybex, Wiley.

INCOSE. (2012). Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, version 3.2.2. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.2.

INCOSE & Wiley. (2015). INCOSE systems engineering handbook: A guide for system life cycle processes and activities (4th ed.). Wiley-Blackwell.

Jacobson, L., & Booch, J. R. G. (2021). The unified modeling language reference manual. Pearson Higher Education.

Jordan, N. C., Saleh, J. H., & Newman, D. J. (2006). The extravehicular mobility unit: A review of environment, requirements, and design changes in the US spacesuit. Acta Astronautica, 59(12), 1135-1145.

Kamrani, A. K., & Nasr, E. A. (2010). Engineering design and rapid prototyping. Springer Science & Business Media.

Käpyaho, M., & Kauppinen, M. (2015, August). Agile requirements engineering with prototyping: A case study. In 2015 IEEE 23rd International requirements engineering conference (RE) (pp. 334-343). IEEE.

Katz, T., Orr, K., Wheatcraft, L. (2022, May). Guide to Needs and Requirements. INCOSE-TP-2021-003-01, v1.0.

Katz, T., Wolfgang, R., Wheatcraft, L. (2022, May). Guide to Verification and Validation. INCOSE-TP-2021-004-01, v1.0.

Katz, T., Ryan, M., Wheatcraft, L., Wolfgang, R (2022, May). Needs and Requirements Manual. INCOSE-TP-2021-002-01, v1.1

Koning, P. (2019, September 5). How Can Agile Leaders Create the Right Context for Ownership. Scrum.Org. https://www.scrum.org/resources/blog/how-can-agile-leaders-create-right-context-ownership.

Kossiakoff, A., Biemer, S. M., Seymour, S. J., & Flanigan, D. A. (2020). Systems engineering principles and practice (3rd ed.). Wiley-Blackwell.

Kniberg, H. (2015). Scrum and XP from the trenches: How we do Scrum. C4Media.

Lee, R.C. (2012) Success Factors of Running Scrum: A Qualitative Perspective," Journal of Software Engineering and Applications, Vol. 5 No. 6, 2012, pp. 367-374.

Makinen, J. V., Bue, G. C., Colunga, A., & Vogel, M. (2014). Mini-Membrane Evaporator for Contingency Spacesuit Cooling. Therm. Fluids Anal. Work. TFWAS 2014.

Manzo, J. (2003). Agile Development Methods, the Myths, and the Reality: A User Perspective," Proc. USC-CSE Agile Methods Workshop, USC Center for Software Engineering.

Maropoulos, P. G., & Ceglarek, D. (2010). Design verification and validation in product lifecycle. CIRP annals, 59(2), 740-759.

Martin, P. K. (2021, September 10). NASA's Development of Next Generation Space Suits. Report No. IG-21-025.Pdf. https://oig.nasa.gov/docs/IG-21-025.pdf

Martin, P. K. (2021, November 15). NASA's Management of Artemis Missions. Report No. IG-22-003.Pdf. https://oig.nasa.gov/docs/IG-22-003.pdf

Metzger, L. S., Rebovich Jr, G., Cormier, R. A., Norman, S. J. T., Schuh, D. L., Smyton, P. A. & Wendt, F. C. (2014). Systems Engineering Guide: Collected Wisdom from MITREs Systems Engineering Experts. MITRE CORP BEDFORD MA BEDFORD United States.

McClelland, D. C., & Burnham, D. H. (1977). Power is the great motivator. Boston, MA: Harvard Business Press.

McKay, D. S., Heiken, G., Basu, A., Blanford, G., Simon, S., Reedy, R. & Papike, J. (1991). The lunar regolith. Lunar sourcebook, 567, 285-356.

Moe, N.B., Dingsyr, T., and Kvangardsnes, O. (2009). Understanding shared leadership in agile development: A case study in: 'Book Understanding shared leadership in agile development: A case study, IEEE.

Morris, P. W., & Pinto, J. K. (Eds.). (2010). The Wiley guide to project organization and project management competencies. John Wiley & Sons.

Li, & Thornton, M. (2010). Digital System Verification: A Combined Formal Methods and Simulation Framework (Vol. 27). Morgan & Claypool Publishers. https://doi.org/10.2200/S00257ED1V01Y201002DCS027.

NASA. (2010). Space Flight Program and Project Management Handbook, NPR 7120.5, February 2010.

Oja, J. (2017). Agile project management in engineering, procurement and construction projects: A case study of ABB Grid Integration Finland.

Peacock, P. R., Wilde, R. C., Lutz, G. C., & Melgares, M. A. (1991, January). EMU processing-A myth dispelled. In International Conference on Environmental Systems (No. SAE PAPER 911348).

Peterson, M., & Summers, J. (2021). When worlds collide–a comparative analysis of issues impeding adoption of agile for hardware. Proceedings of the Design Society, 1, 3451-3460.

Pohl, K. (2010). Requirements engineering: Fundamentals, principles, and techniques. Springer. https://doi.org/10.1007/978-3-642-12578-2.

Project Management Institute. (2018). A Guide to the project management body of knowledge, 6th Edition.

Prouty, B. R., Wilde, R. C., & Gan, M. (1991). Evolution of the extravehicular mobility unit for future missions (No. 911349). SAE Technical Paper.

Robertson, S., & Robertson, J. (2012). Mastering the requirements process: Getting requirements right. Addison-Wesley.

Ross, A., Rhodes, R., & McFarland, S. (2019, July). NASA Advanced Space Suit Pressure Garment System Status and Development Priorities 2019. 49th International Conference on Environmental Systems.

Russell, R. F., & Stone, A. G. (2002). A review of servant leadership attributes: Developing a practical model. Leadership & organization development journal.

Ryan, Michael., Wheatcraft, Lou (2019). "INCOSE Guide for Writing Requirements." INCOSE-TP-2010-006-03, July 2019.

Ryan, M. J., & Wheatcraft, L. S. (2017, July). On the use of the terms verification and validation. In INCOSE International Symposium (Vol. 27, No. 1, pp. 1277-1290).

Ryan, M., Wheatcraft, L., Zinni, R., Dick, J., Baska, K. (2022, May). Guide to Writing Requirements. INCOSE-TP-2010-006-03, v3.1.

Schmidt, T. S., Chahin, A., Kößler, J., & Paetzold, K. (2017). Agile development and the constraints of physicality: a network theory-based cause-and-effect analysis. In DS 87-4 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 4: Design Methods and Tools, Vancouver, Canada, 21-25.08. 2017 (pp. 199-208).

Schwaber, Ken. (2004). Agile Project Management in Scrum. Redmond: Microsoft Press.

Schwaber, K. (2014). The enterprise and Scrum: Redmond, WA: Microsoft Press.

Schwaber, Ken., Sutherland, Jeff. (2017). The Definitive Scrum Guide. Attrition of Share-Alike license of Creative Commons. https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf.

Slegers, N. J., Kadish, R. T., Payton, G. E., Thomas, J., Griffin, M. D., & Dumbacher, D. (2012). Learning from failure in systems engineering: A panel discussion. Systems Engineering, 15(1), 74-82.

Soley, C. R. (2013). Object Management Group, Inc. Computer Science 2: Research and Applications, 215.

Software & Systems Engineering Standards Committee of the IEEE Computer Society. (2018). ISO IEC IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering. IEEE.

Sohaib, O., & Khan, K. (2010, June). Integrating usability engineering and agile software development: A literature review. In 2010 international conference on Computer design and applications (Vol. 2, pp. V2-32). IEEE.

Sutherland, J., Ahmad, N. (2011). How a Traditional Project Manager Transforms to Scrum: PMBOK vs. Scrum, presented at Agile 2011, Salt Lake City.

Sutherland, J., & Coplien, J. O. (2019). A Scrum book: the spirit of the game. Raleigh, NC: The Pragmatic Bookshelf.

Thomas, M., & McGarry, F. (1994). Top-down vs. bottom-up process improvement. IEEE Software, 11(4), 12-13.

Todd, K., Hostetler, J., Espinosa, N., & Chullen, C. (2020, July). Exploration Portable Life Support System Hatch Component Design Challenges and Progress. 2020 International Conference on Environmental Systems.

Tripp, John & Riemenschneider, Cynthia & Thatcher, Jason. (2016). Job Satisfaction in Agile Development Teams: Agile Development as Work Redesign. Journal of the Association for Information Systems. 17. 267-307.

Turner, R., & Boehm, B. (2003). Balancing agility and discipline: A guide for the perplexed. Boston: Addison-Wesley/Pearson Education.

Ullman, D. (2009). EBOOK: The Mechanical Design Process. McGraw Hill.

Wall, M. (2019, September 21). Putting Astronauts on the Moon in 2024 Is a Tall Order, NASA Says. Space.Com. https://www.space.com/nasa-2024-moon-mission-difficult.html.

Wells, H., Dalcher, D., and Smyth, H. (2015) 'The adoption of agile management practices in a traditional project environment: An IT/IS Case Study', in: 'Book The adoption of agile management practices in a traditional project environment: An IT/IS Case Study' (IEEE, 2015, edn.).

Wheatcraft, L. S., & Hill, T. R. (2010, July). 11.2. 1 Getting Started on the Right Foot: Developing Requirements for Constellation's Next Generation Space Suit. In INCOSE International Symposium (Vol. 20, No. 1, pp. 1326-1342).

Wheatcraft, L. S. (2011, June). 5.1. 2 Triple Your Chances of Project Success Risk and Requirements. In INCOSE International Symposium (Vol. 21, No. 1, pp. 543-558).

Wiegers, K.E. (2003). Software requirements, 2nd Ed., Microsoft Press, Washington.

Wood, K. (1991). Design of equipment for lunar dust removal. NASA Contractor Report, 190014.

APPENDIX

Table 104 – Component Identifiers

Component Type Identifiers "CCC"	Component
CKV	Check Valve
CON	Controller
DN	Drain Port
DP	Differential Pressure Transducer
FM	Flow Meter
FN	Fan
FSA	Feedwater Supply Assembly
GS	Gas Sensor
GX	Gas Exchange Scrubber
HV	Hand Valve
HX	Heat Exchanger
OR	Orifice
P	Port
PG	Pressure Gauge
PMP	Pump
PRV	Pressure Regulating Valve
PP	Pitot Probe
PT	Pressure Transducer
PV	Pressure Vessel
QD	Quick Disconnect
RV	Relief Valve
S	Feed-through
SOV	Solenoid Operated Valve
TCC	Trace Contaminant Control
TCV	Thermal Control Valve
TP	Test Port
TS	Temperature Sensor
VP	Vacuum Access Port

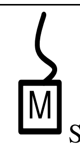
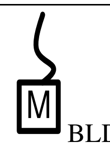
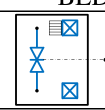
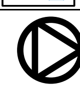
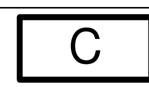
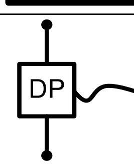
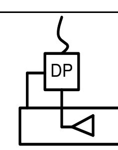
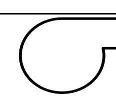
Table 105 – Loop Identifiers

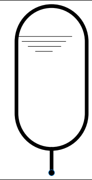
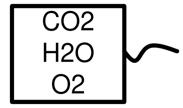
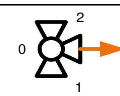
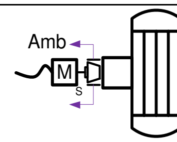
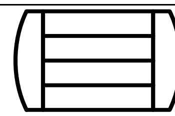
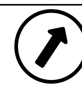
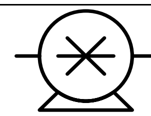
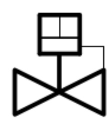
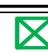
Loop Identifiers	"LL"
Primary Oxygen Loop	1
Secondary Oxygen Loop	2
Oxygen Ventilation Loop	3
Thermal Control Loop	4
Auxiliary Thermal Control Loop	5
Vacuum Manifold	10

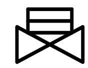
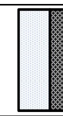
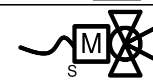
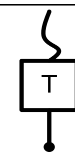
Table 106 – Filter Identifiers

Filter Identifier	Rating
F1	2µ
F2	15µ
F3	20µ
F4	25µ
F5	40µ
F6	100µ
F7	140µ
F8	200µ
F9	440µ
F10	550µ

Table 107 - Component and Symbol Identifier

Component	Identifier	Symbol
Amine CO_2/H_2O Scrubber Bed	---	
Motor Actuator (Stepper motor-based)	---	
Motor Actuator (Brushless DC motor)	---	
Multiple QD Connector with closed when mated valve	---	
Check Valve	CKV	
Controller	CON	
Differential Pressure Transducer	DP	
Filter	F	
Flow meter	FM	
Fan	FN	

Component	Identifier	Symbol
Feedwater Supply Assembly (water accumulator bladder)	FSA	
Gas Sensor	GS	
Hand Valve (3-way)	HV	
Evaporator with stepper actuator poppet style back-pressure valve	HX	
Heat Exchanger (air to water)	HX	
Orifice	OR	
Pressure Gauge	PG	
Pump	PMP	
Pitot Probe	PP	
Pressure Regulating Valve (with piston-based outlet pressure sense)	PRV	
Pressure Regulating Valve (with bellows/diaphragm-based outlet pressure sense and ambient pressure reference)	PRV	
Pressure Transducer	PT	
Pressure Vessel	PV	
Quick Disconnect	QD	

Component	**Identifier**	**Symbol**
Relief Valve (Proportional in-line)	RV	
Trace Contaminant Control	TCC	
Thermal Control Valve (stepper motor actuated diverting valve)	TCV	
FwetTemperature Sensor	TS	
Vacuum Access Port	VP	

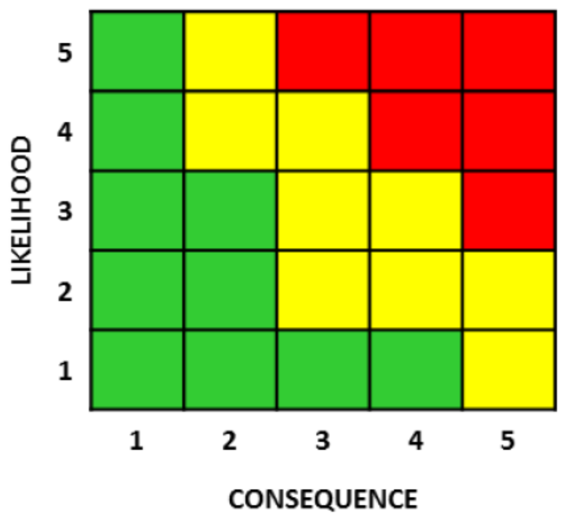


Figure 61 – Generic Likelihood vs. Consequence Matrix

Table 108 –Generic Likelihood vs. Score Matrix

	Score	**Description**	**Probability Range**
Likelihood	1	Very Unlikely	< 10 %
	2	Unlikely	10% to 30%
	3	Possible	> 30% to 60%
	4	Likely	> 60% to 90%
	5	Very Likely	> 90 %

‘

Table 109 -Generic Consequence vs. Range of Impacts Matrix

		Range of Impact to Project Objectives				
	Category	**1**	**2**	3	**4**	**5**
Consequence	Quality	Remote loss of quality	Minimal loss of quality	1 standard deviations away from quality standard	2 standard deviations away from quality standard	3 standard deviations away from quality standard
	Schedule	< 2 week impact to schedule	1 to 2 month impact	3 to 4 month impact to schedule	5 to 6 month impact to schedule	> 7 month impact to schedule
	Cost	< $50K impact	$50K to $250K impact	$250K to $500K impact	$500K to $1MM impact	> $1 MM impact
	Scope	Remote impact to scope objectives	Minimal impact to scope objectives	Considerable impact to scope objectives	Major impact to scope objectives	Severe impact to scope objectives
	Safety	Remote risk of injury	Minimal risk of injury	Minor injury	Severe injury	Loss of life

Table 110 - Earned Value Metrics

Abbreviation	**Item**	**Equation**	**Definition**
PV	Planned Value	N/A	Authorized budget granted for work scheduled.
EV	Earned Value	N/A	Measure of work performed as a function of authorized funding.
AC	Actual Cost	N/A	Actual cost of the work performed during a specified period.
SV	Schedule Variance	EV - PV	Measure of the difference between the earned value and the planned value.
CV	Cost Variance	EV - AC	Measure of the difference between the earned value and the actual cost.
SPI	Schedule Performance Index	EV/PV	Rate of schedule cost health over a period of time, expressed as a fraction.
CPI	Cost Performance Index	EV/AC	Rate of project cost health over a period of time, expressed as a fraction.
BAC	Budget At Completion	N/A	Measure of the total budget granted for a project.
EAC	Estimate At Completion	ETC + AC	Measure of the estimate of the total budget granted for a project.

Table 111 – Abbreviations List

Abbreviations List	
AC	Actual Cost
AC/DC	Alternating Current / Direct Current
AD	Activity Diagram
ADD	Architecture Description Document
AEMU	Advanced Extravehicular Mobility Unit
API	American Petroleum Institute
ATCL	Auxiliary Thermal Control Loop
BAC	Budget at Completion
BATT	Battery
BD	Behavior Diagram
BDD	Block Definition Diagram
BSEE	Bureau of Safety and Environmental Enforcement
CCB	Change Control Board
CCL	Change Control Log
CCT	Correlated Color Temperature
CD	Candela (units)
CD/M^2	Candela Per Meters Squared (units)
CF	Completion Form
CM	Configuration Management
CofC	Certificate of Conformance
Con-Ops	Concept of Operations
COQ	Cost Of Quality
COTS	Custom Off The Shelf
CPAS	Capsule Parachute Assembly System
CPI	Cost Performance Index
CR	Change Request
CTSD-ADV	Crew & Thermal Systems Division Document Type
CWS	Caution and Warning System
DBSE	Document Based Systems Engineering
DCU	Display and Control Unit
DoD	Definition of Done
DR	Discrepancy Report
DVT	Design Verification Testing
EDCC	Engineering Drawing Control Center
EIS	End Item Specification

EMU	Extravehicular Mobility Unit
ENV	Environment (abbreviation, requirement attribute)
EPG	Environment Protection Garment
ESCU	the Exploratory Service and Cooling Umbilical
EVA	Extravehicular Activity
EVM	Earned Value Metrics
FAB	Fabrication
FAR	Federal Acquisition Regulations
FMEA	Failure Modes and Effects Analysis
FPGA	Field Programmable Gate Array
FSA	Feedwater Supply Assembly
FUN	Functional (abbreviation, requirement attribute)
FY	Fiscal Year
GRC	Glenn Research Center
GSE	Ground Support Equipment
HH&P	Human Health & Performance
HITL	Human-In-The-Loop
HLS	Human Launch Support
HUT	Hard Upper Torso
HWHM	Half-Width-Half-Medium
HX	Heat Exchanger
IBD	Internal Block Diagram
IEEE	Institute of Electrical and Electronics Engineers
INCOSE	International Council of Systems Engineering
IRCD	Internal Requirement Control Documents
ISS	International Space Station
JETS	JSC Engineering and Technology Services
JPR	JSC Procedural Requirements Document
JSC	Johnson Space Center
LCD	Liquid Crystal Display
LCVG	Liquid Cooling and Ventilation Garment
LED	Light Emitting Diode
LGVS	Liquid Ventilation Garment System
LOE	Level of Effort
MAC	Modified Agile Concept
MAG	Maximum Absorbency Garment
MBSE	Model Based Systems Engineering
MINI-ME	Miniature Membrane Evaporator
MIP	Mandatory Inspection Point
NASA	National Aeronautics and Space Administration

NASA-STD	NASA Standard Document
OEM	Original Equipment Manufacturer
OMG	Object Management Group
PD	Parametric Diagram
PDM	Product Data Management
PKD	Package Diagram
PMBOK	Project Management Body of Knowledge
PMI	Project Management Institute
PO	Purchase Order
POR	Secondary Oxygen Regulator
POV	Primary Oxygen Vessel
PP&C	Project Planning & Controls
PR	Purchase Request
PSI	Pounds Per Square Inch (units)
PT	Pressure Transducer
PTRS	Project Technical Specification
PV	Planned Value
R	Requirement (abbreviation, requirement attribute)
RCA	Rapid Cycle Amine
RD	Requirement Diagram
RES	Requirement Engineering Scorecard
RFI	Request for Information
RFT	Risk & FMEA Tool
RTD	Resistance Temperature Device
S&MA	Safety & Mission Assurance
SD	Sequence Diagram
SE&I	Systems Engineering & Integration
SEMP	Systems Engineering Management Plan
SIPPE	SWME Integrated Performance and Pressure Evaluator
SME	Subject Matter Expert
SOI	System of Interest
SOR	Secondary Oxygen Regulator
SoS	System of Systems
SOV	Secondary Oxygen Vessel
SPD	Spectral Power Distribution
SPI	Schedule Performance Index
SRR	Systems Requirement Review
STE/D	**Special Test Equipment/Devices**
STM	State Machine Diagram
SWME	Spacesuit Water Membrane Evaporator

SysML	System Modeling Language
TBD	To Be Determined
TBR	To Be Resolved
TBX	To Be Resolved/Determined
TPS	Technical Process Specification
TRR	Test Readiness Review
TSD	Test Station Delivery
UCD	Use Case Diagram
UML	Unified Modeling Language
W/M^2/NM	Watters Per Meter Squared Per Nanometer (units)
WBS	Work Breakdown Structure
xEMU	Exploratory Extravehicular Mobility Unit
xEVAS	Exploration Extravehicular Activity Services
xINFO	Exploratory Informatics System
xMWS	Exploratory Mini Workstation
xPGS	Exploratory Pressure Garment System
xPLSS	Exploratory Portable Life Support System