Developing a Component Reuse Strategy for Space Launch Vehicles
by
Marissa Ann Good
B.S. Aerospace Engineering, Massachusetts Institute of Technology, 2012
Submitted to the MIT Sloan School of Management and the Department of Aeronautics and Astronautics in Partial Fulfillment of the Requirements for the Degrees of
Master of Business Administration
and
Master of Science in Aeronautics and Astronautics
In conjunction with the Leaders for Global Operations Program at the
Massachusetts Institute of Technology
June 2017
©2017 Marissa Ann Good. All rights reserved.
The author hereby grants MIT permission to reproduce and to distribute publicly copies of this thesis document in whole or in part in any medium now known or hereafter created.

Signature redacted
MIT Sloan School of Management
Department of Aerospace Engineering
May 12, 2017

Signature redacted
Olivier L. de Weck, Thesis Supervisor
Professor of Aeronautics and Astronautics and Engineering Systems

Signature redacted
Roy Welsch, Thesis Supervisor
Professor of Statistics and Engineering Systems

Signature redacted
Dr. Youssef M. Marzouk, Chair of the Graduate Program Committee
Associate Professor of Aeronautics and Astronautics

Signature redacted
Maura Ilerson
Director of MBA Program, MIT Sloan School of Management
DISCLAIMER NOTICE

Due to the condition of the original material, there are unavoidable flaws in this reproduction. We have made every effort possible to provide you with the best copy available.

Thank you.

The images contained in this document are of the best quality available.
Developing a Component Reuse Strategy for Space Launch Vehicles

by

Marissa Ann Good

Submitted to the MIT Sloan School of Management and the Department of
Aeronautics and Astronautics on May 12, 2017 in Partial Fulfillment of the
Requirements for the Degrees of
Master of Business Administration
and
Master of Science in Aeronautics and Astronautics

Abstract
Launch vehicle hardware is traditionally very expensive to design, develop, produce and certify, because it must operate in extreme environments with high reliability. The result is that most hardware for NASA-funded launch vehicles is custom built to execute a specific mission on a single platform. In contrast to other industries (e.g. automotive), very few components are used across product platforms, a strategy known as reuse that has the potential to decrease the cost, schedule and risk of new product introduction. Budget constraints on NASA’s next launch vehicle, the Space Launch System (SLS), brought about a desire to realize some of the benefits associated with reuse. However, the reuse strategy as employed has met limited success. This brings about the fundamental question: is there something inherently unique about launch vehicle design that prevents or limits reuse? If not are there strategies that can be implemented to realize the benefits of proactive reuse during launch vehicle design?

The Boeing Company, the prime contractor of the SLS cryogenic stages, would like to develop a reuse approach as they begin work on the next phase of the SLS, the Exploration Upper Stage (EUS), to improve project affordability. To develop this approach, a case study of the Core Stage (CS) was performed to identify lessons learned, resulting in the following insights:

1. Capturing the benefits of reuse is enabled by modularity and platforms within single-vehicle architectures rather than across vehicles. The time offset between any two launch vehicles is too great (20-30 year product lifecycles) for reuse across vehicles. Furthermore, manned and unmanned vehicles carry different requirements which must be considered when evaluating the potential for shared assets.
2. Race should be defined as the baseline, rather than as an opportunity. This requires aligning incentives and architecting the organization to enforce reuse from the outset.
3. Plan for forward reuse. Consider future requirements when designing the current vehicle. Reuse will not happen by coincidence; it must be designed into the system.

These insights form the basis of a reuse approach for the Exploration Upper Stage (EUS). In combination with some organization and process-based suggestions, a strategy to realize the benefits of reuse has been developed for the EUS and other future launch vehicles.

Dr. Olivier L. de Weck,
Thesis Supervisor, Professor of Aeronautics and Astronautics and Engineering Systems

Dr. Roy Welsch,
Thesis Supervisor, Professor of Statistics and Engineering Systems
This page has been intentionally left blank.
Acknowledgements

Over the past year I have received insight and help from a number of people and organizations, each of which have made the completion of this thesis possible.

First, I want to thank the Boeing Space Launch System team for hosting me during my research. Being able to join such a tremendous organization for six months to study and learn was truly an honor. Specifically, I want to thank my manager, David Rigby, for providing guidance and encouragement throughout my project. I also want to offer my gratitude to the larger propulsion team, from which a number of individuals took time to help me develop the thesis that follows. I also want to thank Craig Abler and Karen Schuster for making the project possible.

I want to thank my advisors, Roy Welsch and Olivier de Weck, for their support throughout this journey. Their insight allowed me to successfully navigate both the managerial and technical challenges presented in this project.

The Leaders for Global Operations community was also instrumental to the completion of this thesis. I also want to thank the MIT Office of the Dean for Graduate Education for their support of my studies over these past two years through the Zakhartchenko Fellowship.

Finally, I want to thank my family and friends for their never-ending love and support.
This page has been intentionally left blank.
Table of Contents

ABSTRACT ................................................................................................................................. 3
ACKNOWLEDGEMENTS ............................................................................................................... 5
TABLE OF CONTENTS .............................................................................................................. 7
LIST OF FIGURES .................................................................................................................... 9
LIST OF TABLES ....................................................................................................................... 10

1 INTRODUCTION ..................................................................................................................... 11
   1.1 THESIS OBJECTIVE ........................................................................................................ 11
   1.2 PROJECT MOTIVATION .................................................................................................. 13
   1.3 APPROACH AND METHODOLOGY .............................................................................. 13
   1.4 MAJOR FINDINGS ......................................................................................................... 14
   1.5 THESIS ROADMAP ..................................................................................................... 16

2 BACKGROUND ....................................................................................................................... 17
   2.1 INDUSTRY OVERVIEW ............................................................................................... 17
   2.2 THE BOEING COMPANY ............................................................................................. 17
   2.3 NASA’S SPACE LAUNCH SYSTEM ............................................................................. 18
   2.4 THE BOEING COMPANY AND THE SLS PROGRAM .................................................... 19
   2.5 REUSE EFFORTS ON SLS ........................................................................................... 20

3 LITERATURE REVIEW ......................................................................................................... 23
   3.1 DEFINING REUSE AND COMMONALITY .................................................................. 23
   3.2 BENEFITS AND PENALTIES OF SHARED ASSETS .................................................... 25
       3.2.1 Benefits ............................................................................................................... 26
       3.2.2 Penalties ............................................................................................................. 27
   3.3 LIFECYCLE OFFSETS .................................................................................................. 29
   3.4 DIVERGENCE THEORY ............................................................................................... 30
   3.5 MANAGEMENT TECHNIQUES TO COMMONALITY AND REUSE............................ 31

4 PROPOSED FRAMEWORK – ANALYSIS APPROACH ....................................................... 33
   4.1 DIVERGENCE DEMONSTRATED VIA LIFECYCLE OFFSETS ................................... 33
   4.2 DIVERGENCE QUANTIFIED THROUGH A BILL OF MATERIALS ANALYSIS ........ 33
   4.3 SYSTEM ARCHITECTURE MAPPING THROUGH DESIGN STRUCTURE MATRICES ... 35

5 CASE STUDY 1 (RETROSPECTIVE): CORE STAGE REUSE ................................................ 37
   5.1 SETTING THE STAGE: NASA MANDATED REUSE .................................................. 37
   5.2 INTERNAL ACTIONS TO SUPPORT REUSE .............................................................. 39
       5.2.1 Mid-V Systems Engineering ............................................................................... 39
       5.2.2 REUSE PLAN IN THE SYSTEMS ENGINEERING MANAGEMENT PLAN ........ 41
       5.2.3 Defining the Reuse Baseline: Risk versus Opportunity ..................................... 43
   5.3 INTERVIEW-BASED ASSESSMENT OF REUSE .......................................................... 44
   5.4 LIFECYCLE OFFSETS IN THE SPACE LAUNCH SYSTEM CORE STAGE .............. 48
   5.5 BOM ANALYSIS: REUSE LEVELS THROUGHOUT THE CORE STAGE PROGRAM .... 51
   5.6 CORE STAGE ARCHITECTURE SHOWN THROUGH DESIGN STRUCTURE MATRICES ... 56
   5.7 ORGANIZATIONAL BARRIERS TO REUSE IN THE CORE STAGE ......................... 60
   5.8 CORE STAGE SUMMARY OF FINDINGS ................................................................. 62
# List of Figures

**Figure 1:** The Evolution of the Space Launch System [2] ............................................................... 19  
**Figure 2:** Expanded View of the Space Launch System [2] .............................................................. 19  
**Figure 3:** Reuse Classification System as Defined by Boas [1] ................................................................. 25  
**Figure 4:** Three Types of Offsets as Studied by Boas ........................................................................... 29  
**Figure 5:** Representation of Anticipated Commonality Levels over Program Lifecycle [1] ........................................................... 34  
**Figure 6:** The V Model of Systems Engineering [8] .............................................................................. 39  
**Figure 7:** The Mid-V Model of Systems Engineering ............................................................................ 40  
**Figure 8:** CS Reuse Process Defined in the SEMP ................................................................................. 42  
**Figure 9:** Representative Org Chart for SLS CS ................................................................................... 46  
**Figure 10:** Timeline of NASA Manned Spaceflight Systems ................................................................. 49  
**Figure 11:** BOM Study of the Level of Reuse on the Core Stage over the Program Lifecycle .................. 51  
**Figure 12:** Changes in the CS MPS over the Program Lifecycle ........................................................... 54  
**Figure 13:** Total Program Changes in the CS over the Program Lifecycle ............................................... 56  
**Figure 14:** CS Flow DSM ....................................................................................................................... 58  
**Figure 15:** CS Structural DSM ............................................................................................................. 59  
**Figure 16:** Lifecycle Offsets between the EUS, CS and Delta IV ............................................................ 66  
**Figure 17:** EUS Flow-Based DSM ....................................................................................................... 68  
**Figure 18:** EUS Structural DSM ......................................................................................................... 69  
**Figure 19:** Proposed Reuse Process to Improve Likelihood of Reuse Success ..................................... 80
List of Tables

TABLE 1: THE BENEFITS OF REUSE AND COMMONALITY ON COST, SCHEDULE AND RISK VARY FOR THE LEADING AND LAGGING VARIANTS IN A PRODUCT FAMILY. ............................................................. 27
TABLE 2: THE PENALTIES OF REUSE AND COMMONALITY ON COST, CAPABILITY AND PERFORMANCE VARY FOR THE LEADING AND LAGGING VARIANTS IN A PRODUCT FAMILY. ..................... 28
TABLE 3: COMPARING THE BENEFITS OF THE CS REUSE STRATEGY TO A STANDARD REUSE STRATEGY ................................................................................................................................. 38
TABLE 4: NUMBER OF INTERVIEWEES ORGANIZED BY LEVEL OF AUTHORITY .................................................. 45
TABLE 5: COMPARING THE BENEFITS OF COMMONALITIES STRATEGIES, THEORETICAL AND IN PRACTICE ON THE SLS CS AND EUS .................................................................................. 67
1 Introduction

Developing a complex product requires many decisions to be made to satisfy competing goals: cost, schedule, performance, and risk position, to name a few. Decades of development of complex systems have shown that it is difficult to satisfy all of these goals simultaneously. In turn the goals must be prioritized. In order to aid decision makers in this prioritization, various strategies have been developed, one of which is the reuse of assets from existing products in the creation of new products. Reuse promises the benefits of decreased costs and a shortened schedule, as there is no need to invest the capital or the development time to create a new asset. Additionally, the product risk is lower, as known assets present a higher likelihood of success due to operational experience. This strategy has been applied in many industries, from power tools to cars to airplanes [1], in the form of commonality, or the sharing of components amongst members of a product family. However little research exists related to the reuse of assets in the creation of space launch vehicles.

Launch vehicle hardware is traditionally very expensive because it is required to operate in harsh environments (e.g. high g-loads, extreme temperatures, ambient and vacuum pressure) with high reliability. Given this challenge, most launch vehicle components are built to execute a specific mission on a single platform, and are designed and operated accordingly. In turn very few components are used across product platforms. Each new launch vehicle development program tends to incur extremely high non-recurring engineering costs in order to develop an entirely new set of components. These high costs are often accompanied by long development cycles, with significant work that must be done to buy down risk. It seems that the reuse of hardware across products could combat these challenges: lowering non-recurring costs, shortening the schedule and decreasing risk. The question stands: is there something inherently unique about launch vehicle design that prevents or limits reuse? And if not are there strategies that can be implemented to enforce component reuse during launch vehicle design? These questions form the topic of this thesis.

1.1 Thesis Objective

Most recently an attempt was made to implement reuse during the development of a new launch vehicle, the Space Launch System (SLS), owned by the National Aeronautics and Space
Administration (NASA)\(^1\). With limited funds set aside for NASA in the coming years, the budget for SLS is quite constrained and is also limited year-over-year. This limited budget drove NASA to select a vehicle architecture built around reusing designs from past launch vehicles. This reuse strategy was adopted by the Boeing Company, the prime contractor for the Core Stage (CS) of SLS, who planned to reuse many of the component designs and technologies developed for previous launch vehicles in their CS design.

However, as the vehicle design matured, much of this planned reuse fell away, commonly referred to as divergence in the literature, leading to increased development costs and a lengthened schedule. There are many thoughts as to what caused reuse to diminish on the SLS CS. This thesis aims to move from speculation to data-driven reasoning to explain this divergence as well as provide insight into possible solutions to prevent it in future launch systems. The goals of this thesis are three-fold:

1) Perform a retrospective case study of the CS to identify, document and understand the problems that arose that caused divergence in the CS.

2) Assess the planned reuse strategy for the Exploration Upper Stage (EUS). Develop alternative strategies to better capture the desired benefits of reuse in this stage.

3) Generalize these strategies such that they can be applied to future launch vehicles in order to bring about the desired benefits of reuse, namely lower cost and a shortened development schedule.

Each of these objectives taken independently would provide much value to the aerospace industry, however developing them sequentially will allow insights to first be gleaned from the CS, then applied to the EUS and ultimately generalized for future launch vehicle development programs. In turn this thesis should provide value both to the company at which this research was performed as well as the aerospace industry.

\(^1\) Reuse will be defined at length in section 3.1. As a brief clarification though, reuse here refers to the repurposing of existing designs in a new application – for example, reusing a valve design.
1.2 Project Motivation

This topic arose due to recent work by the Boeing Company on the SLS vehicle, where the Boeing Company built their plan for the CS around reusing component designs from the Space Shuttle and other launch vehicles. This strategy was in line with the vehicle architecture strategy selected by NASA, reusing a number of expensive, complex components from the Space Shuttle on the SLS vehicle. Reusing components permitted the Boeing Company to deliver a plan that minimized cost and shortened the development schedule. However, the final design incorporates only a small number of reused component designs. The objective of this thesis is to understand what happened from the original project plan to present day that caused this divergence, and then to develop strategies to combat this divergence from occurring so as to capture the desired benefits of reuse in the future.

Ultimately this project comes out of a need for prime contractors of space exploration programs to find ways to provide a higher performance product at a lower cost. The aerospace industry is becoming increasingly competitive year after year, with new market players and a more cost-conscious consumer. To remain competitive in this ever-changing market, costs must be lowered with no impact on performance or reliability. Specifically, in the case of human space exploration, there is no option to reduce cost if this leads to a decrease in the system reliability. When flying human crews, safety is of utmost importance. In turn most hardware for human space exploration missions is designed to be highly fault tolerant, resulting in expensive and slow development cycles. Reuse and commonality have been utilized in other industries to achieve lower costs while also providing shorter product development cycles and more robust products. The question remains as to whether these strategies, when applied to launch vehicles, will result in the gains needed to continue to advance the extent and pursuit of human space exploration.

1.3 Approach and Methodology

To develop a strategy for component reuse on space launch vehicles, the current state must first be understood. For this project, the current state is represented by a case study of the CS of SLS. The study begins with a series of informal interviews conducted to develop an understanding of the situation at hand. A wide array of individuals with varying levels of responsibility and
functional backgrounds are interviewed. From these interviews valuable insight is gained into the challenges encountered when implementing reuse, and an appropriate scope at which to perform this research is defined. The scope is narrowed from reuse throughout the CS to specifically focusing on reuse within the Main Propulsion System (MPS). Next a literature review is conducted to gain insight into academic work and best practices done to date on this topic. This is accompanied by an internal document review 1) to gain familiarity with the MPS and 2) to establish the program’s reuse strategy.

With this background knowledge in place, the quest for a data-driven assessment of reuse begins. First a detailed time study of industry products, specifically launch vehicles, is performed. Next a bill of materials (BOM) analysis of the CS demonstrates divergence, while also quantifying reuse levels throughout the program. This BOM analysis coupled with important program milestones identifies potential sources of divergence. The sources of divergence are further considered by a study of change propagation in the program. Next the CS architecture is mapped using Design Structure Matrices (DSM), which allows for identification of significant system interactions. These interactions are used to understand how changes in one subsystem may drive divergence in another.

These analyses taken together offer an understanding of the sources of divergence on the CS and lessons learned are developed. Next a study of the EUS reuse strategy occurs, using similar methods to those applied to the CS where possible. The lessons learned from the CS are applied to the reuse plan of the EUS to improve the likelihood of achieving the benefits of reuse on the EUS. Finally, generalization of these lessons learned allows for development of a plan to improve the chances of realizing the desired benefits of reuse in future launch vehicles.

### 1.4 Major Findings

Through a retrospective case study of the CS and a prospective case study of the EUS, the answer to the first question posed by this thesis is answered: yes, there are unique aspects of launch vehicles that prevent direct reuse of designs from previous launch vehicles. A range of arguments will support this finding. The first is that lifecycle offsets in launch vehicles are too significant to support reuse. Launch vehicles are offset by decades, which poses a huge challenge
to reuse due to obsolescence of technology, processes, and supply chains. Furthermore, architecture choices made from one vehicle to the next are likely to shift due to differing mission needs from reliability to performance to cost to schedule. These choices are even further impacted when considering manned versus unmanned systems. All of these factors come together to bring about architecture changes, which result in changing requirements that inhibit reuse.

Although the successful implementation of reuse in launch vehicles is challenging, there are opportunities to employ similar strategies, such as commonality, to launch vehicles to support the goals afforded by reuse. Rather than focusing on the direct reuse of old hardware designs in new launch vehicles, this thesis argues that programs should focus on developing commonality within a launch vehicle itself. By viewing a launch vehicle as a family of products, with each stage being a single product in the family, the concept of commonality can be applied at the vehicle level. Furthermore, launch vehicles often include plans to expand capability over time. Advanced iterations of the vehicle fill the role of additional products in the family. By applying a commonality strategy at the vehicle level, the desired benefits of reuse can be achieved without requiring the use of heritage component designs. However, it should be noted that even at the vehicle level, implementing commonality might prove challenging if the stages of the vehicle are being designed and built by different contractors, as is often the case in government-funded programs. In turn, it cannot be said that in all cases commonality applied at the launch vehicle level will be successful. Rather, the approach to commonality should be customized for each program, taking into account the various factors that play into the success of said strategy.

Employing a commonality strategy offers the potential benefits promised by a reuse strategy. However, there will still be challenges in implementation. Specifically, attention must be given to planning for future products. Commonality will not occur naturally. If the first product is developed without thought as to the needs of the second product, it is extremely unlikely that components will be shareable between the two. Concerted efforts must be placed on planning for commonality from the program outset. In addition, commonality must be identified as a critical success metric. The program must build its baseline around the commonality strategy in order to incentivize individual decisions to support reuse. If commonality is not fully committed to from
the program outset, it is unlikely that it will be successfully employed in later portions of a program.

This thesis aims to support an argument for pursuing (proactive) commonality as opposed to (reactive) reuse when it comes to new launch vehicle development. It builds upon prior work performed in the areas of reuse and commonality in complex systems, and offers value to future designers of complex systems with large offsets as they aim to develop new products at lower costs with shortened schedules and higher reliability.

1.5 Thesis Roadmap

This thesis mirrors the research approach taken over the course of this project. Chapter 2 provides a brief background on the industry, the project sponsor, the launch vehicle, and the reuse approach to date. A literature review in Chapter 3 will present past work relevant to the topics of reuse, commonality, and complex system design. From these academic sources a solution framework is proposed in Chapter 4. This framework leads directly into the first case study, detailed in Chapter 5, in which the SLS CS reuse strategy is analyzed. The SLS CS offers a rich study of the challenges faced when employing reuse in a space launch vehicle, with findings built around interviews, a time study, a bill of materials analysis, and architecture mapping. These findings are then coupled with organizational context and challenges to develop lessons learned. Next a case study of the SLS EUS, currently in earlier phases of development than the CS, is presented in Chapter 6. The EUS team is working to employ a three-pronged reuse strategy, and the challenges and lessons learned from this strategy are discussed. Finally results, recommendations, and future research opportunities round out this thesis in Chapter 7.
2 Background

This chapter will set the stage for the research topic at hand, beginning with a discussion of the space industry, and more specifically The Boeing Company’s position in the industry. This will be followed by a discussion of the SLS vehicle, architecture and missions, as well as The Boeing Company’s involvement in the SLS program. Finally, a discussion of the role of reuse in the SLS vehicle will wrap up the chapter.

2.1 Industry Overview

The human space launch industry began during the Cold War when the Soviet Union and the United States engaged in a space race to put the first human into orbit. NASA built a series of launch vehicles during the 1950’s and 1960’s, ultimately culminating with the first manned landing on the Moon in 1969, carried by the mighty Saturn V. From 1981 to 2011 NASA focused its efforts on missions to low earth orbit (LEO) by operating the Space Shuttle, first for purely exploration purposes, and then starting in 1998 to support the transport of American astronauts and cargo to and from the International Space Station.

After nearly 70 years of American leadership in human space exploration, significant changes hit NASA in 2010: the Space Shuttle was retired and NASA’s Constellation Program was canceled. This left the United States fully dependent upon Russia to shuttle astronauts to and from the International Space Station. At this inflection point, NASA took a step to diversify its investment in human space travel. Through the Commercial Orbital Transportation Services and Commercial Crew Development contracts, NASA provided funding to commercial companies to aid in developing the private spaceflight sector, ultimately hoping to commercialize transport of both humans and cargo to and from LEO. NASA also began development of a deep space vehicle, the SLS, and a corresponding crew capsule, the Multi Purpose Crew Vehicle, to support human exploration beyond-earth-orbit.

2.2 The Boeing Company

The Boeing Company is the world’s largest aerospace company, manufacturing commercial airplanes as well as defense, space and security systems. The Boeing Company is made up of
five business units: Commercial Airplanes; Defense, Space & Security; Capital Corporation; Shared Services Group; and Engineering, Operations and Technology.

Boeing Defense, Space & Security is a global organization headquartered in St. Louis, Missouri that provides solutions in the fields of military fixed-wing aircraft, rotorcraft, satellite systems, launch vehicles, and weapons. Boeing Defense, Space & Security is divided into five business units: Military Aircraft, Global Services and Support, Network and Space Systems, Phantom Works, and Development. Boeing Defense, Space & Security is also involved in United Launch Alliance (ULA), a joint venture with Lockheed Martin to provide reliable, cost-effective launch services to the United States government.

The Boeing Company has been involved in human space exploration since the dawn of the space race, when McDonnell Aircraft Corporation was selected to design, test and build the Mercury capsule, NASA’s first manned spacecraft. With over 50 years of space industry expertise, The Boeing Company was selected as one of the primes for the development of SLS. This work is being spearheaded at the Boeing Defense, Space & Security facility in Huntsville, AL, where the research for this thesis was performed.

2.3 NASA’s Space Launch System

SLS will be the world’s most powerful launch vehicle when completed, providing capability for the launch of astronauts and payloads into deep space. The SLS vehicle is an evolvable launch vehicle, meaning its capabilities can be expanded over time with additional engineering resources. This evolvable nature was chosen to support NASA’s future goals for deep-space missions. These versions of SLS are referred to as blocks, with the current planned blocks shown in Figure 1. The first block (known as Block 1) is currently being produced and is targeted to first launch in 2018.
2.4 The Boeing Company and the SLS Program

SLS is composed of various systems procured from multiple suppliers. Figure 2 displays the Block 1 vehicle divided into its various systems.
The Boeing Company is the prime contractor for the design, development, test and production of the vehicle’s avionics suite and the cryogenic stages: the CS, the Interim Cryogenic Propulsion Stage, and the EUS.

The Critical Design Review for the CS is complete and the team is moving towards vehicle build and integration. A significant percentage of the hardware needed for the first build has been procured and is in the process of being assembled and tested at the component level.

The EUS, which also forms a critical part of this research project, has recently passed through Preliminary Design Review (PDR), and is moving to detailed design and analysis.

2.5 Reuse Efforts on SLS

The architecture of SLS was built on reusing critical pieces of hardware and technology from previous launch vehicles, including the Space Shuttle, to reduce development time and cost. Although the practice of reuse and commonality is common in other industries, it is rarely employed in aerospace. Yet for the SLS vehicle, reuse was viewed as an absolute necessity due to a severely restricted budget. Compared to the approximate $141.5 billion spent to develop the Space Shuttle [3], NASA had a budget of just $7.1 billion for SLS [4]. To best utilize this budget, NASA focused reuse on the core technologies that are most costly to develop: the propulsion systems. The SLS will reuse:

1) The Space Shuttle Main Engines (RS-25D) as the main engines of the SLS Core Stage
2) Space Shuttle-derived solid launch vehicle boosters as the SLS solid launch vehicle boosters (growth from 4 segments to 5)
3) ULA’s Delta IV Cryogenic Second Stage as the SLS Block 1 upper stage
4) ULA’s Delta IV Second Stage Engines (RL10) as the EUS main engines

These reuse choices determined much of the architecture for the vehicle, setting the vehicle size and power level.

---

2 All dollar figures adjust to 2014 dollars.
Given NASA’s choice to prioritize reuse, it was natural for the Boeing Company to match NASA’s architecture by proposing to build the CS around significant design reuse. The CS is composed of many different subsystems (e.g. propulsion, avionics, and structures), each of which is required to perform specific functions. Two subsystems in particular, the Main Propulsion System (MPS) and the Thrust Vector Control System (TVC) are strongly tied to the main engines. The primary function of the MPS is to support the flow of fluids from the propellant tanks to the engines, resulting in component-level requirements that are strongly tied to the needs of the engines they support. The TVC system is used to control the positioning of the engines relative to the vehicle centerline, permitting the thrust to be vectored to control the flight of the launch vehicle. In turn, the TVC system design is closely coupled to the main engines so many components of the TVC subsystem were identified as reuse candidates. These two subsystems represent a significant percentage of the planned reuse proposed by the Boeing Company when developing the CS development plan.
This page has been intentionally left blank.
3 Literature Review

This literature review section will provide a brief discussion of prior academic work performed related to reuse and commonality. First reuse and commonality will be defined followed by a discussion of the benefits and penalties of both strategies. Two critical topics found in previous research which influence reuse, divergence and lifecycle offsets, will be considered and analytical tools to allow for their identification will be developed. A discussion of management techniques that have successfully supported reuse and commonality will round out this chapter.

3.1 Defining Reuse and Commonality

The term “reuse” can mean many different things, and thus it seems worthwhile to spend a bit of time defining this term. Below are definitions for the three types of reuse that will be discussed in this thesis:

1) Hardware Reuse: physical reuse of existing hardware (e.g. taking a piece of hardware that is already built and repurposing it for a new application)
2) Design Reuse: reuse of an existing hardware design (e.g. making a new part with a new serial number from an existing drawing for use in a new application)
3) Derived Reuse: modified reuse of an existing hardware design (e.g. modifying a drawing and using the modified design in a new application)

The SLS CS includes all three of these reuse types, however the most common type discussed in this thesis is design reuse, which will be referred to henceforth as “reuse.” Any deviation from design reuse will be referred to by name.

Another term used throughout this thesis is commonality, or the sharing of assets across a product family. To explain the difference between commonality and reuse, consider the lifecycles of two products, product A and product B. In the case where product A and product B are being developed concurrently, commonality is a strategy employed to utilize common assets between these two products to realize an economic benefit. Commonality requires considering the needs of both products simultaneously, and developing assets to satisfy these needs. In contrast consider the case where product A is developed before product B, and assets from product A are identified for use in product B retroactively. Reuse represents the repurposing of these assets from product A to realize an economic benefit for product B. In reuse no initial
thought is given to designing assets to benefit both products. Instead product B retroactively identifies beneficial assets from product A. Product A gains to benefit from reuse. In contrast commonality benefits both products.

Boas developed a classification system, shown in Figure 3, to define reuse classes for a program in which two products are developed sequentially (product A followed by product B). During the development of product A, some parts were identified as unique to product A while others were intended to be common with product B. During the development of product B, five unique component classes were found. Class 1 represents parts that were intended unique to product A and remained unique. Class 2 represents parts that were intended to be common but were only used in product A. Class 3 represents parts that were intended to be unique to product A, but were actually used in both products. In turn class 3 best aligns with the definition of reuse as given in the first chapter of this section: reuse which was reactive. Class 4 represents parts that were intended to be common and were common. This class represents commonality as discussed in the previous paragraph: proactive reuse planned from the outset of development. Class 5 represents the unique parts designed solely for product B. Understanding these classifications aids in understanding the differences between reuse and commonality.
Another definition of reuse that is currently quite common in the launch vehicle industry is the reuse of an entire launch vehicle or stage. For example the Space Shuttle was built as a reusable vehicle: it would deliver its payload to LEO and then return to earth for refurbishment prior to its next launch. Several private companies are currently pursuing stage reuse, where the first stage of the launch vehicle is returning to Earth after delivering its payload to be used again for another launch. Although stage and vehicle reuse is an extremely interesting topic, it is not the focus of this thesis. For the purpose of this thesis “reuse” refers not to vehicle reuse, but to hardware, design, or derived reuse as defined above.

Now that these critical terms have been defined, the benefits and penalties of commonality and reuse will be discussed.

### 3.2 Benefits and Penalties of Shared Assets

Prior research identifies both benefits and penalties of sharing assets between products, whether through reuse or commonality. Consider two products developed in series: the first product is...
defined as product A and latter products as product B. A detailed discussion of the benefits and penalties of shared assets on the product lifecycles of products A and B follows, and is critical to understanding the rationale behind employing reuse and commonality strategies in complex system development.

3.2.1 Benefits

The benefits of reuse and commonality improve the cost, schedule and risk positions of a program, discussed below. A summary of these benefits is given in Table 1, which highlights that product B potentially realizes the vast majority of the benefits from reuse and commonality with product A benefiting only from commonality.

1) *Reduction in development time*

When developing product B, sharing assets from product A will shorten product B’s *development timeline*. Utilizing an asset that has already been developed and tested drastically reduces the time required to prepare that asset for integration with its parent system. In turn sharing assets decreases time to market, an important factor for successful product introduction [5].

2) *Reduction in development, production, and operation costs*

Product B also benefits from reduced *development* cost, when utilizing assets developed for product A. For any existing asset used by product B, there is a reduction in the labor hours spent on development, which reduces development cost [5]. Costs are further reduced during *production*. Assuming that product A and product B are produced simultaneously, utilizing common assets between the two systems brings about economies of scale [6]. This reduces production costs for both product A and product B on a per unit basis by spreading fixed costs across a higher volume of components. There can also be improved learning curve effects. *Operation* costs are also reduced for both product A and product B. By lowering the number of different types of assets required to build any given set of products, lower costs in inventory, materials management, logistics, and purchasing are achieved [5].

3) *Decreased risk*
Pre-existing assets also lower risk compared to developing all new assets. Assets that have been successfully demonstrated in product A provide greater confidence when utilized in product B. Critical to this thesis specifically, hardware demonstrated in space carries a higher technology readiness level (TRL), an important metric in evaluating risk for NASA systems. Utilizing assets that have high TRL’s increases confidence in the system’s likelihood of success.

Table 1: The benefits of reuse and commonality on cost, schedule and risk vary for the leading and lagging variants in a product family.

<table>
<thead>
<tr>
<th>Benefits</th>
<th>Reuse</th>
<th>Commonality</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Product A</td>
<td>Product B</td>
</tr>
<tr>
<td>Reduction in development time</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Reduction in development cost</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Reduction in production cost</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Reduction in operation cost</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Decreased risk</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

3.2.2 Penalties

The penalties of reuse and commonality impact the cost, capability and performance of a program, discussed below. Table 2 provides a summary of these penalties, highlighting that product B is negatively impacted by both strategies with product A only penalized by commonality.

1) *Excess development and integration cost*

When utilizing an asset from product A into product B, whether through commonality or reuse, there will be increased costs in either development or integration. When a common part is design, product A will incur greater development costs than if it developed an asset to only meet its own needs. A common asset will generally have a larger number of requirements to satisfy both products. This will increase complexity leading to increased development costs, adding to the total program cost of Product A. Product B will in turn be less costly, resulting in a lower product *family* cost. However, product A must carry the excess cost initially. In the case of reuse, gaps will exist between the capability of the reused asset and the needs of product B. To close these gaps
and allow for successful integration into product B, additional analysis or testing will be required, resulting in increased integration costs, likely to impact both products.

2) *Excess capability*

Developing a common asset or reusing an asset will result in excess capability in the products [5], depending on the difference in levels of performance of the two (or more) products. In the case of a common asset, the asset must meet the requirements of both products simultaneously (the union), resulting in some amount of excess capability for both products. In contrast when reusing an asset, product B may carry the excess capability penalty as there may be functionality in the asset required for product A that is not needed for product B.

3) *Decreased performance*

Because common or reused assets carry excess capability, product performance will be decreased. This penalty is especially pertinent in the case of launch vehicles, where vehicle mass is one of the primary figures of merit. Any excess capability in performance that results in increased component mass will negatively impact vehicle mass and therefore vehicle performance.

<table>
<thead>
<tr>
<th>Penalties</th>
<th>Reuse</th>
<th>Commonality</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Product A</td>
<td>Product B</td>
</tr>
<tr>
<td>Excess development cost</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Excess integration cost</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Excess capability</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Decreased performance</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

Table 2: The penalties of reuse and commonality on cost, capability and performance vary for the leading and lagging variants in a product family.

This discussion of the benefits and penalties of reuse and commonality provide some insight into why a program may choose to pursue one strategy versus the other. A challenge remains in quantifying whether the benefits of each of these strategies outweigh the penalties for any given product. In the case of the SLS CS it was determined early on that reuse offered sufficient benefit to outweigh the penalties, resulting in the selection of that strategy by the program. To determine whether or not that assessment was correct, further academic methods for assessing reuse are discussed below.
3.3 Lifecycle Offsets

Lifecycle offsets refer to the temporal relationships that exist between the lifecycle phases of different products [1]. Research by Boas on the relationship between lifecycle offsets and reuse found the degree of offset to be critical to the successful implementation of reuse. The three offset classifications utilized by Boas were parallel, sequential overlapping, and sequential non-overlapping. Figure 4 graphically presents these offsets. Product A is known as the lead variant and product B as the latter variant.

<table>
<thead>
<tr>
<th>Parallel</th>
<th>Sequential Overlapping</th>
<th>Sequential Non-Overlapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product A</td>
<td>Product A</td>
<td>Product A</td>
</tr>
<tr>
<td>Product B</td>
<td>Product B</td>
<td>Product B</td>
</tr>
</tbody>
</table>

Offset | Overlap | Offset

Figure 4: Three Types of Offsets as Studied by Boas

Boas found that parallel development offers the highest likelihood for successful implementation of commonality. Developing products simultaneously leads to the creation of decision-making processes that capture the most benefit for the product family. By realizing commonality at the product family level, the optimal solution for the family can be identified bringing about the maximum benefit to the program.

As product families move from parallel development towards sequential, lifecycle offsets present increasing challenges due in part to a lack of knowledge of the needs of latter variants during the development of the lead product. This results in choices being made to maximize the economics of the lead variant, rather than those of the product family. Lifecycle offsets further weaken the product family, as spans in time are accompanied by changes in technology and product obsolescence. As technology changes or products become obsolete, there may be a benefit for the latter variant to elect a new process or component in order to realize economic benefits. This may improve the economics of the latter variant but can hurt that of the lead and the product family, which benefits from commonality. The larger the lifecycle offset between products, the
more challenging it is to capture the economic benefits of commonality within the product family.

As we approach the topic of reuse in a space launch vehicle context, understanding lifecycle offsets will be critical to assessing reuse as a strategy to lower costs, shorten schedules and decrease risk.

3.4 Divergence Theory

Divergence is a concept that states that commonality in a platform tends to decrease throughout the product lifecycle, for both beneficial and non-beneficial reasons [1]. Boas performed case studies on seven different product families, ranging from military aircraft to printing presses to semiconductor manufacturing equipment. Through these studies, four types of changes were identified as common sources of divergence: changing requirements; learning; availability of new technologies; and component obsolescence [1]. It was also found that divergence is triggered by an enabler, either an economic incentive or a lack of coordination [1].

When developing a new product, it would be ideal if requirements could be fully defined at the project’s outset and held constant throughout. This would allow for a linear product development process, foregoing any churn due to changing requirements. However, requirements do change over the lifecycle of a program, resulting in an iterative process. These iterations can drive divergence, as any requirement change should be accompanied by an assessment of alternative solutions, bringing about an opportunity to select an alternative and drive a decrease in commonality.

Learning is another common source of divergence, as there is a tendency for additional knowledge gained during production and operations to drive design modifications for later instances (or block upgrades) of the product. As a design moves through its lifecycle, challenges will be faced and counter-measures developed. These solutions can result in changes to the design and divergence in the system. For example consider a system in which the same component is used in two locations. During the operations phase of the product, it is found that one of these locations (A) tends to require more frequent maintenance than does the other (B). It
would be beneficial for A’s design to be altered to support removal for maintenance. However B’s design fits its needs and should remain unchanged. A new part may then replace A, with B unchanged, driving divergence in the system.

The final two change sources identified by Boas, availability of new technologies and obsolescence, are related in many ways to lifecycle offsets, discussed above, so no further discussion will take place here.

Each of these sources of change may result in divergence; however an enabler must exist to trigger divergence, often either an economic incentive or a lack of coordination [1]. In certain cases elimination of a common component leads to an economic incentive, thereby resulting in a beneficial form of divergence. In these cases, the change from common to unique results in a more profitable product family and in turn is an acceptable form of divergence. However there are also cases where the main enabler of divergence is a lack of coordination. In these cases divergence is non-beneficial and should be avoided. Greater care to the incentive structure and the decision making process should be given to prevent non-beneficial divergence.

As we move through the case study of the SLS CS and on to the course of action for EUS, it is critical to recall that divergence can be both beneficial and non-beneficial. All divergence should not be viewed as a “negative.” Rather the sources and enablers of divergence must be understood in order to classify divergence and to develop appropriate solutions to counter non-beneficial divergence.

### 3.5 Management Techniques to Commonality and Reuse

Much research has been done to identify management techniques that result in the successful implementation of commonality and reuse. A study related to the development of the Sony Walkman highlights the importance of management’s commitment to commonality. Boas also developed a large number of management recommendations during his multi-industry commonality study.
When considering the development of the Sony Walkman product family, Sanderson and Uzumeri found several aspects of management that led to Sony’s command of the personal portable stereo market [7]. The first was the commitment of management to minimize design cost. In an industry with a multitude of customer demands, Sony kept cost at the core of its product strategy. This led to the development of product modules and platforms. By requiring engineers to design new products around existing platforms, Sony was able to produce a wide variety of models with high cost and quality. This management strategy maintained commonality and discouraged divergence.

Boas also identified a number of management approaches to capture the benefits of commonality:

1) Commonality will not occur accidentally. The identification of opportunities for commonality must be managed proactively. Formal ownership of commonality should be established to ensure that opportunities for commonality are identified.

2) Commonality can be beneficial or detrimental and should not be blindly implemented. Instead a detailed evaluation of each opportunity for commonality should determine the proper course of action.

3) Once commonality is identified as beneficial, ensure that it is implemented. This may require metrics to support reuse, but at a minimum requires the creation of processes that ensure changes are evaluated in terms of their impact on commonality.

These proven management techniques will help guide later discussions related to successful implementation of reuse and commonality in launch vehicles.
4 Proposed Framework – Analysis Approach

The analytical approach of this thesis is built upon three unique assessments. The first is a study of the lifecycle offsets relevant to the SLS stages. A bill of materials analysis will measure reuse levels throughout the program lifecycle. The expectation is that reuse levels will peak during the early planning phases of the program and will decrease thereafter, showing divergence as defined by the literature. Finally, design structure matrices (DSM) will map the subsystem interactions within each of these stages. These matrices provide insight into dependencies within each stage, and when coupled with changes may lead to divergence in the system.

4.1 Divergence Demonstrated via Lifecycle Offsets

The first step in developing a reuse strategy is to understand the temporal offsets between the products being considered. In the case of launch vehicles, it is useful to consider offsets both between different NASA-developed launch vehicles as well as between the stages of a single vehicle. Launch vehicles are developed infrequently, with lifecycles on the order of decades. In turn we are likely to see very large offsets between the development timelines of two independent vehicles (e.g. the Space Shuttle and SLS). However, most launch vehicles are composed of multiple stages in order to complete the mission, and in turn the stages tend to be developed sequentially with some overlap. Considering the lifecycle offsets at the vehicle and stage level may provide insight into similarities and differences in the challenges faced when attempting to reuse designs across these platforms.

4.2 Divergence Quantified through a Bill of Materials Analysis

The literature has demonstrated that as the development of a complex system progresses it is typical to observe a decrease in commonality levels, referred to as divergence. Through initial interviews it is quite clear that the SLS CS has undergone significant divergence, however no formal analysis has identified the root cause. In turn a BOM analysis of the CS was performed to identify trends in reuse over time. By mapping these trends to key program dates and considering trends in change management data, insight into the sources of divergence was developed.

Prior to performing a BOM analysis on the CS, it is important to understand what a typical BOM analysis of commonality looks like. A BOM analysis results in a graphical representation of the
success (or failure) at incorporating reuse into a design. Boas proposed Figure 5 as a representative graph of commonality over program lifecycle. The x and y axes represent time and percent common parts, respectively. This model proposes maximum reuse levels occurring near the start of the program, during the planning phases when optimism about the successful implementation of reuse is highest. At the outset of the program commonality levels are expected to increase rapidly, due to limited knowledge as to the true feasibility of reuse. Point A represents the peak of this planning phase. From this point, divergence occurs resulting in ever decreasing levels of reuse as the program matures. The sources of divergence discussed in section 3.4 drive this decrease. Point B, although not commonly observed in Boas' studies, represents a potential mid-project attempt to increase reuse levels after significant divergence has occurred. These attempts aim to reincorporate reuse into a program late in the game with a hope of realizing some of the planned (and lost) economic benefits. Unfortunately, Boas found that late attempts to reestablish reuse in a program were rarely successful.

Figure 5: Representation of Anticipated Commonality Levels over Program Lifecycle [1]

The data for Boas' BOM analysis comes directly from a bill of materials. By comparing part numbers across a family of products, Boas can easily identify shared components in a product. However, in the case of SLS, a standard BOM analysis could not be performed due to the program's position on the use of part numbers versus specifications.
The program chose to create a new specification for every part on the launch vehicle, including the case where a part design was to be reused. Rather than utilizing an existing part number, an entirely new specification and part number were assigned. To indicate the intention for a vendor to reuse a design if possible, the original part number was defined in the text of the specification. This part number would typically be called out in the item definition/description at the start of the specification. However, it was ultimately up to the vendor to decide how to satisfy that specification. The vendor was not required to deliver a reused design. The vendor was paid to deliver a part that fully met the specification, which may or may not have been possible with a reused part. In turn tracing heritage part numbers in the CS BOM was impossible.

Instead the BOM analysis data for the CS was pulled from a mix of sources and was used to map out divergence over the life of the program. Proposal data was used to gather an understanding of the initial baseline. This data was then supplemented with information from a risk and opportunity management tool, which identified any changes to hardware baselines, including reuse over the course of the program. In turn the analysis shows sharper trends than would likely be observed in a true BOM-based analysis due to limited availability of temporal information. Nonetheless this BOM analysis provides interesting insights into the overall trends of reuse in the program, which will be discussed in Section 5.5.

4.3 System Architecture Mapping through Design Structure Matrices

Commonality is ultimately tied to product architecture. There are a variety of ways to map the architecture of a system. One possible tool is a DSM, which provides a clear and compact representation of the interfaces and interactions that exist within a system. Given the complexities in the design of the CS, developing a DSM to map the existing connections was the next step in the system analysis.

DSM’s can be constructed at different levels of abstraction, from the individual component level to subsystems to systems. For this thesis DSM’s were developed at the subsystem level. The reasons for this choice are as follows:

1) The DSM is meant to serve as a readily digestible map of the system, one that can be used as a communication tool. Performing a component level DSM would result in an
extremely large and complex DSM, making it challenging to use the DSM for communication.

2) Most launch vehicle stages share common subsystems, however the specific interfaces of these subsystems may vary. By modeling at the subsystem level it becomes possible to quickly identify architecture choices that are unique to a certain stage.

3) A sufficient level of granularity is necessary for the DSM to provide meaningful insights. A system level DSM would have ignored the complexity that exists within each of the major systems (i.e. propulsion, structures, avionics).

By analyzing at the subsystem level, it is possible to understand how choices made within a subsystem have impacted the overall system level performance. After developing a DSM of the system, the dependencies between subsystems are identified, bringing about further insight into sources of divergence.
5 Case Study 1 (Retrospective): Core Stage Reuse

A case study of the CS will provide insight into the incorporation of reuse in the SLS vehicle thus far. In order to allow for a thorough analysis to be performed, the decision was made to focus in on the MPS rather than attempting to understand all intended reuse scenarios across the stage. A discussion of NASA’s role in the mandating of reuse begins this chapter, followed by the processes developed by the Boeing Company to support reuse in the CS. A summary of information given through interviews will present some of the organizational context surrounding reuse on the CS. Next an analysis of lifecycle offsets will detail the temporal landscape of the CS development with respect to other NASA-funded launch vehicles. A BOM analysis and change management data will highlight periods of divergence in the program. These periods of divergence will lead to the development of architectural mapping data to identify subsystem interactions that drove reuse out of the system. Finally, a discussion of the challenges found and lessons learned will round out the CS case study.

5.1 Setting the Stage: NASA Mandated Reuse

When the Boeing Company developed its development plan for the CS, cost and schedule reigned supreme. NASA needed a cost-effective launch vehicle quickly in order to return the capability of human space exploration to the United States. In turn NASA mandated that the SLS be architected to incorporate significant reuse, both design and hardware reuse, from prior space vehicles, primarily the Space Shuttle and Constellation.

One candidate identified by NASA for hardware reuse was the RS-25D. Launch vehicle engines are extremely costly to develop, requiring years to design, test, and verify. This development schedule is further lengthened when the engine must support human space exploration, with the levying of more stringent requirements. In turn the SLS architecture needed to incorporate an existing launch vehicle engine if the program was to meet its cost and schedule targets. This need combined with the fact that NASA had remaining RS-25D’s in storage led to the decision to reuse the RS-25D as the engines for the CS.

This engine selection afforded the Boeing Company the opportunity to further support NASA’s cost and schedule goals, creating a low cost stage with a shortened development schedule, by
reusing various pieces of support hardware developed for use with the RS-25D. One example of a component that seemed a likely candidate for reuse was the pre-valve, a valve that isolates the fluid flow path between the propellant tank and the RS-25D. The pre-valve was designed to accommodate specific parameters set by the engine, for example pressure, flow rate and temperature. Given that the engine was to remain unchanged, it seemed that this valve could be used as-is in the new CS design. This same logic was applied throughout the CS, resulting in a target reuse level of 80% for the MPS. These reused designs offered the two benefits NASA was looking for, decreased cost and a shortened development schedule, as well as the added benefit of lowering risk as flight-tested hardware carries a higher TRL than a new design. The expected benefits of reuse are shown in Table 3. Here you can see that the CS reuse strategy realizes only some of the benefits of a standard reuse strategy, in which both products A and B are still in operation. Two areas in which expected benefits are not realized are production and operations. Because Space Shuttle hardware is no longer being actively produced or operated, the CS cannot fully capture the benefits of reuse. Reduction in production costs come about due to economies of scale, as multiple products utilizing the same asset lowers per unit production costs. However in the case of the CS, the Space Shuttle is no longer producing hardware, meaning that economies of scale do not exist. Operational costs, which are reduced in reuse scenarios due to sharing of spares and support equipment between two products, are also left unrealized in the case of the CS. Because the Space Shuttle is no longer in operation, the CS must fully support all operational costs rather than sharing these with the Space Shuttle program.

Table 3: Comparing the benefits of the CS reuse strategy to a standard reuse strategy

<table>
<thead>
<tr>
<th>Benefits</th>
<th>Reuse</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Product A</td>
</tr>
<tr>
<td>Reduction in development time</td>
<td></td>
</tr>
<tr>
<td>Reduction in development cost</td>
<td></td>
</tr>
<tr>
<td>Reduction in production cost</td>
<td></td>
</tr>
<tr>
<td>Reduction in operation cost</td>
<td></td>
</tr>
<tr>
<td>Decreased risk</td>
<td></td>
</tr>
</tbody>
</table>
5.2 Internal Actions to Support Reuse

In order to achieve the 80% reuse target set for the program, a series of choices were made at the outset of the program to support the incorporation of reuse into the stage design. The creation of a new form of systems engineering, titled Mid-V Systems Engineering, was intended to support subsystem level reuse. A reuse process was incorporated into the Systems Engineering Management Plan (SEMP), defining the standard approach to be taken when proposing reuse. However, the program also chose to baseline new designs throughout the stage, offering reuse as an opportunity to change the baseline. Each of these choices and their implications will be discussed below.

5.2.1 Mid-V Systems Engineering

The Mid-V Systems Engineering model was proposed to promote subsystem level reuse. In order to discuss Mid-V, we must first gain an understanding of the traditional V model of systems engineering. The V model has become the standard approach to systems engineering in NASA projects since being developed in the 1980’s. A graphical representation of the V model is shown in Figure 6.

![Figure 6: The V Model of Systems Engineering [8]](image)

The V model is composed of two legs. The left leg represents the efforts to: define the project scope, develop and decompose the requirements, and identify concepts to fulfill the project goals. The right side represents verification, validation, integration and operation of the system. When following the standard V Model, a project begins at the top, left side of the V by defining...
the highest program objectives and then progresses systematically down the left leg until detailed design is achieved. Once detailed design is complete the right side of the V is traversed, ensuring that at each step the project goals are satisfied. In order to build a successful product, work must be done to tie activities on the left side of the V directly to planned activities on the right side.

In contrast to this standard V model, the Boeing Company proposed a truncated version of the V for the CS, the so-called Mid-V model (see Figure 7). The Mid-V model proposes bypassing the lower half of the V, via the use of heritage designs at the subsystem level. This allows for less effort to be spent in detailed design and unit testing, instead moving straight from subsystem level design to subsystem verification.

![Figure 7: The Mid-V Model of Systems Engineering](image)

The Mid-V model is based on a premise that requirements at the component level will remain unchanged from those of the heritage hardware. Unfortunately, this premise did not hold true for the CS, and gaps between the CS and heritage requirements were uncovered as the program matured. Some of these gaps were identified early on in the program, for example a number of government specifications had changed (or been updated) since the Space Shuttle was built, which was shown at the outset of the SLS program. However the extent that these specification changes would have on the methods of hardware design and test were not well understood until much later in the program, when subject matter experts began to evaluate methods to meet such specifications. Once such example is the development of a new specification for fracture mechanics, which is discussed in detail in Section 5.4. Environmental gaps were also uncovered as the vehicle design matured. Analytically determining the acoustic environment of a launch vehicle requires significant modeling effort. At the outset of the program, an assumption was
made that the acoustic environment of the CS would be very similar to the engine bay of the Space Shuttle, as the boosters and main engines were common. However as the actual design of the SLS vehicle matured, significant changes to the acoustic environment were discovered during analytical modeling. These environments directly impacted the vibrational requirements of individual components, resulting in gaps that were too large to permit reuse (see discussion of valve test failure in Section 5.5). These are just a few examples of ways in which gaps were uncovered at the component level later in the program, which presented a challenge to the Mid-V reuse plan.

5.2.2 Reuse Plan in the Systems Engineering Management Plan

A reuse implementation plan was also developed and incorporated into the SEMP at the outset of the program. A number of subject matter experts in the area of reuse were called upon to develop this plan. The plan was written such that a small team of engineers could employ it to gain sufficient insight into the potential for reuse. These insights were then presented to technical program leaders, who would determine whether or not to pursue reuse. The plan is composed of several steps:

1) Develop the component requirements and gather the pertinent information for potential reuse designs
2) Conduct a performance-based gap analysis to identify areas where requirements are not fully encompassed by prior designs
3) Answer a list of questions to understand if any non-performance based gaps exist
4) Present findings and receive approval (or rejection) from the technical board to pursue reuse

This four step process is summarized in Figure 8.
The gap analysis sits at the core of this plan, and is a strong tool for any engineer to use to assess the technical potential for reuse. However this four-step process faces several challenges. The first is that the process does not define a way to identify reuse opportunities in the first place. The process relies on an existing set of components that have been identified as reuse candidates. This may result in parts that could be reused being overlooked, if the engineers working on the program are unaware of an existing design that could meet their needs. It seems this process could be improved by incorporating guidelines to help identify reuse candidates at the start of any design process.

Another challenge presented by this process exists at step 4, where significant variation is observed in the way gap analysis results are used to determine whether or not to pursue reuse. Depending on the makeup of the technical board, a reuse plan may be approved in a situation when a new design may be best and vice versa. The decision to pursue reuse, although based on a strong technical analysis, does not produce consistent results. Ultimately the decision-making process should be made more rigorous in order to yield results that will best support the program goals and objectives.
5.2.3 Defining the Reuse Baseline: Risk versus Opportunity

Another interesting aspect of reuse implementation on the SLS program relates to how reuse was classified and tracked throughout the lifecycle of the program. When developing a new system, a large number of components are required in order to make the system function. One can pursue reuse, derived reuse or a new design when identifying candidates for a component. Depending on which choice is made, the program baseline is established. In a program built around reuse to support the needed program cost and schedule, choosing to baseline direct reuse seems most appropriate, as this would establish a baseline with the lowest cost and shortest schedule. However, in the case of the CS the choice was made to baseline a new set of requirements for each component, with a hope (and request to vendors) that an existing design be utilized to meet each of the new requirements. The choice to pursue a “new design” as the baseline, with an opportunity to incorporate a reused design at a later date only if all requirements were fully satisfied by the old design, promotes developing new hardware. Reuse is viewed as an opportunity, with a new design being the baseline, which presents several challenges to incorporating reuse.

First, if the program is allowed to define a new set of requirements for every needed component, it is highly unlikely that a prior design will show full compliance. Instead a new component is baselined as this is the only way to demonstrate compliance, tying cost and schedule initially to the creation of a new part. Once the cost and schedule of developing a new component is allocated, there is little to no incentive to alter course and pursue reuse. Even if reuse is expected to satisfy all requirements, there will likely be some additional work needed, whether test or analysis, to prove that the existing design is compliant. This additional work coupled with the progress made towards designing a new component results in a strong aversion to swapping a new design for a reused design, as it appears to be more costly and time consuming to change the baseline than to simply move forward with the new design. The problem with this approach is that consideration of the cost and schedule benefits of reuse occurs too late to support a change in strategy. By defining a new part as the baseline, any change from the baseline is discouraged. Even if the decision is made to switch from a new to a reused design, the likelihood of divergence is quite high due to an existing backup, the work performed in support of a new design.
For reuse to be implemented successfully, it must be viewed as the baseline from the outset of the program. Rather than defining an entirely new set of requirements for each component, an attempt should be made to assess whether the old requirements are sufficient to meet the needs of the new product. If significant changes to the requirements are needed, this is a clear indicator that reuse is unlikely to provide a better alternative than a new design. If few gaps between old and new requirements are found, reuse should be established as the baseline with a known risk carried that is tied to the potential for divergence from reuse to a new design if the reused design is found unable to meet all requirements at a later date.

5.3 Interview-Based Assessment of Reuse

To begin to get a sense for the feelings towards reuse, a series of unstructured interviews were performed, first within the SLS program and later across the Boeing Company. The interview scope and data collection process is discussed first, followed by the results of the CS interviews. Finally a discussion of interesting findings shared throughout the Boeing Company when implementing reuse wraps up this section.

To begin understanding reuse challenges on SLS, a range of individuals were interviewed, from executives to managers to individual contributors. A table summarizing the interviewees is given in Table 4. Effort was made to interview individuals from a range of positions within the organization, in order to understand how reuse opinions vary throughout the organization. The interview approach taken was based on Grounded Theory [9]. A preliminary set of high-level, open-ended questions was developed prior to the first interview. During each interview, notes were taken to capture the primary points of the interviewee. These notes were then flushed out after completion of the interview, to ensure proper recording of the interviewee’s views. The next step was analyzing the results of the interview, and using these results to improve the questions prior to the next interview. Most individuals were interviewed just once, with a few individuals requiring follow-up discussions once a greater number of interviews had been conducted and initial hypotheses had been formed. The goal of conducting unstructured interviews was to allow each individual to well represent their understanding of the role of reuse in the program. The progressive nature of the questions over time aimed to dig deeper in each
interview to get at the root of what led each individual to their point of view. This summarizes
the approach of the interviews. What follows are the results of said interviews.

Table 4: Number of interviewees organized by level of authority

<table>
<thead>
<tr>
<th>Function</th>
<th>Number of Interviewees</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executive Leadership</td>
<td>9</td>
</tr>
<tr>
<td>Mid-Level Managers</td>
<td>7</td>
</tr>
<tr>
<td>First Line Managers</td>
<td>11</td>
</tr>
<tr>
<td>Individual Contributors</td>
<td>15</td>
</tr>
</tbody>
</table>

The interviewees were consistently split into three camps when it came to reuse:

1) Reuse is critically important to the success of the program.
2) Reuse only exists during early phases of a program.
3) Reuse is not integral to successfully completing my job.

To understand this variation in thought, the structure of the organization will be analyzed and
thought will be given as to where in the organization each of these views was observed.

A representative organizational hierarchy of the CS program is shown in Figure 9. The executive
leadership team is composed of a program manager and the chief engineer. Under this executive
leadership sits a range of Integrated Product Teams (IPTs), with unique specialties ranging from
propulsion to avionics to systems engineering. Each of these IPT’s has a lead (mid-level
manager) responsible for managing the IPT and delivering on program metrics. Under this lead
sits a number of first-line managers who provide functional leadership to engineers and other
individual contributors. Most of the individuals involved with creating the CS development plan
sat in either executive leadership or IPT lead roles, with a few managers and individual
contributors sprinkled in. The remainder of the team was established once the Boeing Company
was selected as the prime contractor for the CS. This cursory understanding of the organizational
structure will permit a discussion of how the viewpoints regarding reuse vary throughout the
organization.
Individuals who led the CS development plan, those with extensive experience developing new products, most commonly referenced the first viewpoint. These individuals developed a strong opinion during development plan creation that reuse would result in both economic and schedule benefits to the program, as they understood the benefits offered by the strategy of reuse. Even after seeing the many challenges faced by the team in implementing reuse on the CS, these leaders continued to believe in reuse as a promising strategy to produce a low cost and efficient launch vehicle.

However not all individuals shared this viewpoint. There were a number of managers who mentioned reuse as a strategy that only exists early in programs. These managers commonly had experience on other programs, some of which also planned for high levels of reuse at the program outset. However, they had yet to see reuse implemented successfully. Time and time again reuse was touted as a program critical objective, yet still each program would see reuse fall away as priorities shifted from cost and schedule to performance. After repeated experiences observing the implementation of reuse resulting only in challenges, these individuals no longer view reuse as a legitimate strategy, instead sharing a belief that reuse is simply a strategy to
improve metrics during early design stages of the program, a strategy that will never realize its proposed benefits.

The third point of view, that reuse is not particularly pertinent to successfully completing their job, was found most often at the individual contributor level, specifically throughout the engineering work force. With few exceptions, engineers felt as though reuse was not a priority in their day-to-day work. Some recalled experiences in which they had been asked to consider reuse, however more often than not it was found to be more efficient to start with fresh designs. There were no metrics in place to incentivize engineers to reuse hardware. Furthermore, many of the engineers are young, with limited experience working on other launch vehicle programs. In turn little knowledge exists in this part of the work force related to potential reuse opportunities. Instead new designs were built to satisfy needs, as this approach seemed to be simpler than trying to search through old specifications and part catalogs to try and find something to repurpose. These individuals did not oppose reuse – they simply did not consider reuse when performing their daily work. Most were open to the prospect of reusing designs; however, it was clear that reuse would only be feasible if the necessary information regarding existing components was more readily available.

After conducting over 40 interviews in the CS program, the scope of interviews was expanded to include individuals from other Boeing programs, namely managers from Boeing Commercial Airplanes and the Reuse, Commonality and Modularity working group. In these interviews it was clear that reuse was a technique commonly utilized throughout the company to lower costs in the development of new technologies. However, it was striking to find that the challenges faced throughout the Boeing Company that caused reuse to fail consistently fell into one of three buckets, both in the initial interviews of SLS employees and during the interviews of individuals from other Boeing programs:

1) Changing requirements
2) Changing environments or
3) Supply chain obsolescence
Whether airplanes or spacecraft, these challenges seemed unavoidable. Interestingly enough, many of these challenges had been encountered in the CS program, and are detailed in sections 5.4 and 5.5 as potential sources of divergence.

The variety of opinions with respect to the importance of reuse observed in the CS program is far from ideal. A strategy cannot be implemented unless it is valued at all levels of the organization, especially in the case of reuse, where the lowest level engineers must behave differently in their daily work for successful implementation. Upper level management cannot simply state that hardware should be reused; an understanding of the benefits of reuse must be spread throughout the organization and incentivized. Engineers must be trained on the benefits of reuse to provide an intrinsic motivation to seek out situations in which reuse is beneficial. Furthermore, processes must be put in place to streamline the identification of reuse opportunities. Program level metrics tied to the benefits of reuse should be measured at the IPT level to incentivize all managers to pursue beneficial reuse. The managerial insights gained through these interviews provide the first glimpse into changes that must be made to realize the desired benefits of reuse in the SLS program.

5.4 Lifecycle Offsets in the Space Launch System Core Stage

Boas found lifecycle offsets to be critical to the successful implementation of commonality strategies. In order to understand the lifecycle offsets relevant to the CS, the temporal relationship of SLS to other NASA vehicles was determined using a lifecycle analysis. In the past 50 plus years, NASA has led the development of three manned spaceflight launch systems: the Space Shuttle, the Constellation Program (Ares I and Ares V), and the SLS. The Space Shuttle Program was initiated in 1972 and began operations just ten years later. In 2005 after fourteen years of sustained operations, NASA began working on the Constellation Program as a way to guarantee continued manned space exploration by the U.S. following the planned retirement of the Space Shuttle in 2010. However, the short-lived Constellation Program (which also included building a lunar base as an objective) was cancelled in 2010 due to budget and schedule challenges. When the Space Shuttle was retired in 2011, NASA began development of the SLS, transitioning NASA’s exploration goals from LEO missions to those beyond earth orbit. A timeline of these programs can be seen in Figure 10.
As is evident by this timeline, significant lifecycle offsets exist between these three programs, which poses a challenge to reuse. A 23-year offset between the final round of production for the Space Shuttle and the start of production for CS components presents the opportunity for many changes to occur which impact reuse, namely advancements in technology, often seen in the form of changing requirements and specifications, as well as supply chain differences.

One example of a technical advancement that was encountered time and time again during this case study was the expansion of knowledge in the field of fracture mechanics that has occurred in the last three decades. Many of the original Space Shuttle specifications did not include specific requirements related to fracture mechanics, because little was known about the implications of fracture mechanics on the life of metallic components in the late 1960’s and early 1970’s. In contrast the CS hardware is required to meet a set of stringent fracture mechanics-based requirements released by NASA (MSFC-5019) in 2008. This specification came about due to the improved understanding of fracture mechanics and its inspection methods that have been developed over the past 30 years. Although this specification adds rigor to the design of components, few, if any, of the existing Space Shuttle designs comply with the specification. Even though most components from the Space Shuttle proved flight worthy over the 29-year operational life of the Space Shuttle, their designs do not strictly meet MSFC-5019. This brings

---

3 “Fracture Control Requirements for Spaceflight Hardware”
about the question as to whether or not the new specification is unnecessarily stringent. This thesis will not attempt to answer that question, although it is an interesting topic for potential research. Rather, this example is given simply to highlight the fact that large lifecycle offsets between programs makes reuse challenging due to technological advances. This confirms Boas’ conclusion and the importance of lifecycle offsets as a driver of divergence.

Challenges were also faced in the supply chain. While most of the companies that were suppliers to the Space Shuttle still existed at the outset of the CS development program, much of the relevant know-how in the company did not. This challenge is easily observed in suppliers of castings. Developing castings is very time intensive, but once complete the process should be highly repeatable yielding consistent results. However, when attempting to reuse Space Shuttle hardware designs for the CS, it was found that this qualified process no longer produced cast parts that met the specifications. The speculated reason for the process no longer providing adequate results is that the process was not sufficiently robust, likely depending upon specific undocumented techniques (intrinsic recipes) developed 30 years ago by the operators who built the Space Shuttle castings. Thus replicating the process with different operators 23 years after Space Shuttle production had ceased produced different, and unsatisfactory, results. Certainly there were challenges expected in the supply chain, as many companies that existed 23 years ago no longer exist today. However, for those companies that do still exist, the expectation that the parts could be produced in a consistent manner to those produced 23 years ago turned out to be a poor assumption.

The challenges faced by the SLS program due to lifecycle offsets are consistent with those found by Boas, as well as those identified by the Boeing Company as obstructions to reuse. Technological advancements, such as improvements in the understanding of fracture mechanics, were found to be detrimental to reuse. While product obsolescence was not discussed explicitly, process obsolescence did exist and ultimately led to product obsolescence. It seems that lifecycle offsets of the magnitude faced by space launch vehicles make reuse an exceedingly difficult strategy to implement.
5.5 BOM Analysis: Reuse Levels throughout the Core Stage Program

It was clear from the project outset that as the program progressed reuse levels fluctuated. In order to gain a better understanding of the trends in reuse with time, a BOM analysis was performed. The results of the BOM analysis are shown in Figure 11. This BOM analysis tracks the number of parts identified as either design or hardware reuse candidates over the course of the program lifecycle. Due to the structure of the available data, it was not possible to track derived reuse. However research by Boas has shown that derived reuse is unlikely to result in a significant benefit, so tracking just design and hardware reuse will still provide interesting insights.

The BOM analysis shows a maximum reuse level of approximately 50%, 30% lower than the 80% target. A variety of factors caused divergence from this "high" point, which will be discussed below. First though, let us begin with a discussion of the phases early in the program where increases in reuse levels are observed. The thrust vector control (TVC) system was the first subsystem in which reuse was officially incorporated into the BOM, represented by the reused parts through June 2012. NASA mandated that the TVC system be based on hardware reuse, as they wanted to utilize a stock of TVC parts remaining after the decommissioning of the Space Shuttle. In turn the Boeing Company incorporated these components into their BOM early in the CS design. These parts make up the 10% reuse observed through June of 2012.
As the program continued to mature, two significant increases in intended reuse level were observed. The first was in August of 2012 when the Boeing Company’s first internal push to incorporate reuse in the MPS system began. By incorporating reuse from the Space Shuttle, primarily components that supported the RS-25D, the reuse level grew from 10% to over 30%. The reused components ranged from valves to filters to regulators, many of which had been identified as having a potential for reuse in the 80% proposal target. Another significant increase in intended reuse level occurred just after PDR. At this time, the program incorporated a range of ancillary hardware for the TVC system after analysis indicated that this hardware met the system requirements.

After reaching a maximum reuse level of just under 50% in March of 2013, divergence began, punctuated by two large decreases in reuse levels in May and October of 2013. The first significant divergence event occurred after a slew of environmental testing led to a structural failure in a valve, bringing about concerns as to whether reused designs were sufficiently robust to withstand the new environment of the CS. This environmental testing was needed due to a prediction of increased acoustic levels for the CS, which came about during system modeling. To ensure that the reused designs could survive this harsher acoustic environment, an existing valve was put through additional vibration testing at higher power spectral densities. Although finite element analysis of the valve predicted that it had sufficient margins to survive the harsher environment, the structural failure during testing indicated that the margins were insufficient for the increased vibrational load. In turn, a number of planned reuse parts were removed from the BOM, creating divergence.

The second significant divergence event occurred when a program-wide decision was made to abandon the remaining “in question” reuse efforts, and instead move forward with derived or new hardware designs. This decision was made due to challenges encountered in achieving hardware reuse, namely the in-test valve failure and a slew of unsatisfactory results coming out of computer-generated analyses related to hardware operation in the new CS environments. A prevailing sense had come about that reuse of MPS components was unlikely due to changing requirements in the form of environments and specifications. The team had encountered many
roadblocks trying to incorporate reuse into the CS MPS design, ultimately leading to the
program-wide decision to abandon reuse and instead focus on developing new components that
would meet the vehicle requirements.

This analysis brings about some interesting findings:

1. When comparing the BOM analysis for the CS (Figure 11) to the theoretical model proposed
by Boas (Figure 5), we see some stark differences. In the case of the CS MPS, the peak in
reuse levels occurs later in the program than expected. PDR typically marks the point in a
program where a design is quite defined. By PDR significant work has been completed to
prove the feasibility of chosen technologies to meet all requirements. In turn PDR is not
generally considered part of the “planning” phase, when Boas predicted reuse levels to be
highest. This brought about an interesting question: what caused reuse levels to fluctuate so
late in the CS program? Was reuse sufficiently assessed during early planning phases? If not,
did the peak timing shift due to later incorporation of reuse?

2. Second the target reuse level of 80% defined in the development plan was never achieved.
Were there components targeted for reuse at the planning stage that were overlooked when
the design effort began in true? If so, this finding supports the hypothesis that the existing
reuse process lacked critical information related to identification of reuse designs. In turn the
process should be improved to ensure that all proposed reuse components are considered for
reuse at the program outset.

3. Finally, the bulk of divergence occurs in a 6-month window, quite a short time for so many
parts to transition from reused to new. If reuse was an important strategy to the program, it is
surprising that such high levels of divergence were permitted to occur so rapidly. Digging
further into this topic, it was found that a significant change to program requirements
occurred at PDR, when NASA requested a reduction in vehicle weight (a performance-driven
parameter). This represented a shift in program goals, from primarily cost and schedule
focused to performance, which promoted a move from reused to new components.
Furthermore, to meet the new target weight, structural mass was removed from the vehicle,
which changed the dynamic environment of the vehicle. These dynamic changes resulted in
harsher operating environments for individual components. When the valve test failure occurred during April 2013, doubt began to creep in with regards to whether or not any reused components could withstand these new environments. These two factors combined to start the downward trend in reuse.

To supplement the findings above, further study of program changes was performed in hopes that greater insight could be gained as to the drivers of the divergence in reuse observed in the MPS system. The program maintains a database of all changes made since the program start. To understand the sources of divergence, it seemed useful to investigate the cadence of changes in the program. A graph of the number of changes in the MPS system over the program lifecycle is shown in Figure 12. These changes are separated into two categories: those that are internal, changes made by the team at the Boeing Company to either offer an improved solution or to rework a mistake, and decisions that were customer driven, often the result of a changed customer need resulting in a program change flowed to the stage and ultimately to the MPS system.

![SLS CS MPS Changes by Type Over the Program Lifecycle](image)

Figure 12: Changes in the CS MPS over the Program Lifecycle

It is interesting to note that significant changes are observed in the MPS system after PDR, including a large number of changes introduced by the customer. This data seems to indicate that the subsystem likely underwent significant change between PDR and CDR. Reuse demands requirement stability from one application to the next. This points back to the discussion in section 5.2.3 in which the recommendation was made to establish reuse as the baseline at the outset of the program in order to realize the benefits. Establishing reuse as the baseline will only
be effective at supporting reuse if the requirements used to establish the baseline of reused
designs remain consistent throughout the program lifecycle. As large changes in vehicle
requirements occur (e.g. reducing weight and changing vehicle dynamics), the likelihood of
reused designs satisfying these new conditions decreases. In order to establish reuse as the
baseline at the outset of a program, the program must have enough stability in its requirements
for reused designs.

This same set of program change data was then reformatted to show cumulative program
changes over the duration of the program (Figure 13). By representing the cumulative program
changes over time, the desire was to look for trends indicating that program changes would take
on a typical S-curve, as expected over the course of the program. This S-curve was expected to
come about due to the fact that volume of changes should be small at the outset of program
development, when the baseline is being established. Then with time, the system design will
mature, increasing the volume of changes. Finally as development comes to an end, the system
design is nearly complete so change volume should decrease. It is interesting to observe that
change volume is much more linear than expected given the above description. Although a slight
S-curve is observed (clearly the highest rate of change occurs mid-program), the S-curve is quite
gentle indicating that a large number of changes have occurred throughout the program.
5.6 Core Stage Architecture Shown through Design Structure Matrices

To map the complexities that exist in the CS MPS, design structure matrices (DSM) are constructed. DSM’s can be used in a variety of ways to detail the architecture of a system, displaying physical connections to process steps to information flows [10]. To fully capture the complexity of the CS MPS, two DSM’s were constructed, one based on fluid flow and the second on physical connections.

The path that fluids take through the stage is essential to the design of the MPS, yet there is limited flexibility in the design of the fluid systems once an engine is selected. As flow propagates through the MPS, subsystems are intimately linked to one another. Creating a flow based DSM identifies the subsystem interactions that are set by the flow of fluids through the CS, shown in Figure 14. Fluids include LO2, LH2, Helium, etc. Note that connections in this DSM denote flow direction, resulting in an asymmetric matrix. The matrix is organized such that any dependency indicates a flow from the object in the row to the object in the column. For example, the upper left most dependency (colored red) indicates flow from the LO2 (Liquid Oxygen) tank to the LO2 feed subsystem, however there is no reverse flow (from the LO2 feed subsystem to the LO2 tank).
The second DSM (Figure 15) developed is based on the physical connections that exist in the system. This DSM highlights the structural dependencies that each subsystem has with respect to one another as well as its physical location in the launch vehicle. As one can see the structural DSM is symmetric, compared to the asymmetric DSM, because physical connections are always bi-directional. There exist connections in the structural DSM that are a direct result of physical flows, as one can’t have a fluid flow without a structural connection. However physical connections can exist independent of flow paths. One such physical connection exists due to the architecture of using two independent tanks separated by a physical structure, denoted as the intertank. Some launch vehicles instead choose to create a physical connection between the two tank structures, using a common dome, eliminating the intertank. These two architectures create different physical connections between the tanks and the components that exist in this space. This is just one example of an architecture decision that appears in the structural DSM but not in the flow DSM.

Also interesting to note is the fact that there are rows and columns in the flow-based DSM that are empty. This is due to the fact that the DSM’s were constructed first to represent the structural connections, and then the flow-based connections were mapped onto the same DSM. This decision was made because some structural connections must exist even without the need for fluid flow through that connection. To ensure that the analysis fully captured all connections in the system, the structural DSM was developed first, as you cannot have flow without a structural connection. Rather than then adapting the structural DSM to “fit” the flow-based DSM, thereby eliminating empty rows and columns, the DSM’s were left identical to highlight the differences between the two connection types. Fluid-based connections are critical to the proper operation of the CS, and thus are more rigidly defined once the rocket engine system is selected. However the structural connections, which do not support fluid flow, are more flexible. In turn, it was determined that keeping the DSM consistent for both flow-based and structural modeling would best highlight the differences between the two DSM’s, potentially allowing for greater insight to be made as to potential areas of flexibility.
<table>
<thead>
<tr>
<th>SL02 Tank</th>
<th>SL02 Feed</th>
<th>SL02 Al/Drain</th>
<th>L02 Vent</th>
<th>[3L02 Bleed</th>
<th>L02 Pressurization</th>
<th>LH2 Tank</th>
<th>LH2 Feed</th>
<th>LH2 Al/Drain</th>
<th>LH2 Vent</th>
<th>LH2 Bleed</th>
<th>LH2 Pressurization</th>
<th>Right Helium</th>
<th>Ground Helium</th>
<th>Ground Nitrogen</th>
<th>TVC</th>
<th>Engines</th>
<th>Right Termination System</th>
<th>Forward Skirt</th>
<th>Systems Tunnel</th>
<th>InterTank</th>
<th>Engine Section</th>
</tr>
</thead>
</table>

**Figure 14: CS Flow DSM**
Many of these physical connections are a result of different subsystems existing in different areas of the launch vehicle. As is evident in the structural DSM, a majority of components sit in the engine section, an environment whose dynamics are highly coupled to the launch vehicle engines as well as the launch pad. Components in the engine section face different environmental conditions than those in the intertank. When attempting to incorporate reuse or similarity in a launch vehicle stage, consideration must be given to the local environments sustained by the component. If possible common components should be housed in the same region of the launch vehicle so as to limit the variation in environmental requirements. The structural DSM highlights which subsystems may be best aligned to share components due to facing similar environments.

It is also interesting to note the differences in the connections, both flow-based and physical, that occur in the LO2 and LH2 (Liquid Hydrogen) subsystems, the two propellant systems used to feed the RS-25D. In the LO2 subsystem, the fill and drain subsystem is connected with the feed subsystem. In contrast, for the LH2 subsystem the fill and drain line connects directly with the propellant tank. Although an architectural difference like this may seem trivial, it can be a source
of divergence. Consider the fill/drain valve that sits in the fill and drain subsystem. Perhaps there is a desire to use a common fill/drain valve for the LO2 and LH2 subsystems. If a change is made to the LH2 tank this may result in a change in requirements for the LH2 fill/drain valve. However, the LO2 fill/drain valve will be driven by changes to the LO2 feed subsystem. The design of LH2 tank and the LO2 feed system are independent, so a change in one will not result in a change in the other. The result may be a divergence in requirements for the LO2 and LH2 fill/drain valves, and in turn a divergence in shared components. Considering the architectural choices made and how these choices can limit or support reuse is critical to capitalizing on the benefits of reuse.

5.7 Organizational Barriers to Reuse in the Core Stage

The organizational structure of any program will influence how work is accomplished. Understanding this structure and the way in which work is performed can help glean insight into potential sources of non-beneficial divergence. The basics of the organizational structure were given in section 5.3. What follows is a discussion of what it means to develop truly integrated IPTs, and some of the shortfalls observed in the CS team structure.

The idea of an Integrated Product Team is just as one would expect: a team of integrated individuals working on a specific product. In order for an IPT to be truly “integrated,” the team should have individuals from a variety of technical backgrounds all working together to build the best product. The CS program was made up of a number of IPTs, such as MPS, Avionics, Systems, Stages, and others. At the core of each IPT sits a majority of individuals who share a common area of expertise, for example the avionics IPT has a majority of electrical engineers and computer scientists. Creating an integrated team requires embedding representatives from other technical areas into an IPT that is outside of their area of expertise. This builds cross-functional teams best suited to design a given subsystem, while also considering the various needs of the other subsystems. The team structure defined above is the “target” state, however as is true with any organization the CS team ran into challenges that resulted in a disparity between the target and actual states.
In order for an IPT to be truly “integrated” the team must have individuals from a variety of technical backgrounds working in unison to build a subsystem. This intent is challenging to achieve, as it requires individuals to knowingly leave their field of expertise, a place of comfort, and instead join a team in which they have limited domain-specific knowledge. This challenge often results in IPTs being composed of individuals with a variety of titles, indicating various areas of expertise, even though in reality all the team members share similar backgrounds and knowledge bases. This is certainly the case seen within the IPTs of the CS. For example, the propulsion IPT has a number of “systems engineers” whose backgrounds are all in propulsion. These systems engineers tend to perform tasks that are more focused on integration and systems-level problems, however they lack the knowledge and training needed to actually fill the role of a true systems engineer in the propulsion IPT. The result is that Integrated Product Teams are not quite as integrated as one would desire.

In turn each IPT is quite siloed. This problem is further exacerbated by the fact that each IPT has its own metrics. Rather than being evaluated based on the performance of the program as a whole, each IPT lead manages an independent schedule and budget, which often results in actions that may be optimal for the IPT but suboptimal for the program. These are classic challenges in any organization, but are further compounded when trying to incorporate reuse into a subsystem.

To incorporate existing designs, multiple teams must work together to ensure that hardware is integrated successfully. Consider the example of a reused MPS component. In this example the propulsion, avionics, structures, analysis and systems IPTs all need to work together: the propulsion team owns the component, the avionics system must interact with it to provide power and commands, the structures team must place it in an environment in which it can survive, the analysis team must ensure that the chosen mounting location has well defined environments, and the systems team must verify that program-level requirement changes will not prevent reuse. Clearly a significant amount of cohesion between teams must exist for reuse to “work.” Unfortunately, due to a lack of true IPTs on the CS program, cohesion between IPTs was quite limited, creating a barrier to reuse. Another aspect that will influence the ability of IPTs to
seamlessly work together is the ability to develop a “digital twin” of the product, using a Model-Based Systems Engineering (MBSE) approach.

To successfully incorporate reused designs into a program, the organization must be designed to support reuse. Integration of a reused design requires commitment from all technical functions in the organization. This requires the creation and support of strong multi-disciplinary teams incentivized to incorporate reuse.

5.8 Core Stage Summary of Findings

The CS case study provides a wealth of insight into the challenges that arise when trying to incorporate reuse into launch vehicles. For each of the challenges presented, potential alternatives were suggested to better support the goal of realizing the benefits of reuse in space launch vehicles.

1) Reuse across launch vehicles offers a number of benefits: decreased development time and cost and lower risk. However other benefits, namely decreased production and operations costs, are not realized if the launch vehicles are not operated simultaneously. **Launch vehicles being produced simultaneously provide more opportunity to maximize the benefits of reuse.**

2) Reuse demands little to no change to component level requirements. However, requirements between programs are inherently unstable. **The potential for reuse will be greater within a program, as requirements changes can be better controlled.**

3) A strategy cannot be implemented unless it is valued at all levels of the organization. Proper training, metrics and incentives should support core strategies, reuse or otherwise.

4) Lifecycle offsets between launch vehicles are extremely large, in the range of decades, which results in technological changes and obsolescence creating a barrier to reuse. **These offsets are likely too large for successful implementation of reuse.**

5) Reuse must be assessed at the outset of the program and beneficial opportunities must be baselined. Divergence will result in a reduction in reuse levels over time, so planning for the highest level from the start will support realizing at least some of the desired benefits of reuse. Once a new component has been chosen and its cost and schedule is allocated, there is little to no incentive to incorporate reuse.
6) Shared assets depend upon common requirements and environments. **Developing architectural maps highlight areas where commonality is most likely to succeed due to shared environments.** Common assets tied to different subsystems are subject to higher levels of divergence, so care must be given to ensure non-beneficial divergence does not occur.

7) **The organization must be designed to support reuse.** Reused designs require an organizational structure and model-based methods and tools that foster multi-disciplinary approaches to problem solving. Teams must be truly cross functional and incentivized to make decisions that benefit the program as a whole rather than their own subsystem.
This page has been intentionally left blank.
6 Case Study 2 (Prospective): Exploration Upper Stage Reuse

The EUS MPS reuse strategy was composed of three methods: reuse of CS components, reuse of components from another launch vehicle stage, the Delta IV Cryogenic Second Stage (DCSS), and limiting the unique number of part numbers on the EUS. These strategies promised to decrease the overall program cost, due to economies of scale from sharing components between the stages, as well as reduced development schedule and risk for the EUS. Although the EUS reuse strategy was fundamentally different to that employed by the CS, challenges were still encountered in the implementation of reuse, leading to lessons learned and opportunities for improvement.

6.1 Reuse Strategy on the Exploration Upper Stage

The EUS team selected a different reuse strategy than that utilized for the CS. As discussed above, the CS reuse strategy focused heavily on reuse from the Space Shuttle, a vehicle that is no longer being built and operated. In contrast the EUS strategy consisted of three different types of reuse/commonality:

1) Reusing components from the CS
2) Reusing components from the DCSS
3) Reducing the number of unique part numbers within the stage

Compared to the CS strategy, the EUS looked to reuse hardware from two stages, the CS and the DCSS, both currently in production. This strategy decreases the lifecycle offsets between the products (see Figure 16). At the outset it looked as though this strategy would allow for an 84% reuse level in MPS components, 18% coming from shared CS components and 66% from DCSS. The EUS also planned to incorporate as much commonality within the stage as possible, by developing components that would meet the needs of a variety of applications. No target was defined for the number of unique parts on the stage, however component designers were asked to maximize the utility of their components during the design process.
Sharing components between the CS and the EUS offers benefits to both the EUS and the program. Initially the plan was to develop the CS and the EUS in parallel, to allow for the greatest benefits to be gained by developing common assets that would fit the needs of both stages. However due to funding constraints, the team was forced to develop the EUS after the CS was part-way through development. The result is that the EUS team reaps greater benefits from shared components, while the CS bears more of the cost of developing those shared assets. The EUS team benefits from knowledge gained during CS development, as building, testing, and flying the CS will result in invaluable experience integrating and operating the components. The EUS team further benefits from the fact that as the EUS design effort began, a number of individuals who worked on the CS transitioned to the EUS. This further aided in the transfer of knowledge, permitting the EUS team to bypass some of the early development pains associated with new design efforts. Furthermore, the EUS will carry lower risk during initial testing and operation, as confidence in the components will have grown during CS operations prior to the first EUS test. The program will also benefit, as greater economies of scale will be realized. By purchasing a larger number of identical parts from a supplier, unit costs will decrease reducing total program cost.

The second part of the strategy, utilizing assets developed for the Delta IV, is quite interesting. This strategy is similar to the repurposing of Space Shuttle designs on the CS. In the case of the EUS, DCSS designs were identified as candidates for reuse due to the fact that the DCSS engine, the RL10, was selected as the main engine for the EUS. The same thinking that applied in the CS...
reuse strategy occurs here: the RL10 engine sets many of the operational requirements of the stage and in turn it should be possible to repurpose its components on a new stage built around the same engine. One advantage to the DCSS reuse strategy is that, unlike the Space Shuttle which is no longer in production or operation, DCSS is still being produced and flown. This means that the hardware proposed for reuse is actively being built, reducing concerns of supply chain or product obsolescence. Although the Delta IV is currently built by ULA, it was originally developed and owned by the Boeing Company. In turn there was confidence that the team at the Boeing Company had sufficient understanding of the hardware to promote successful incorporation into the EUS.

The final prong of the strategy, developing hardware that can meet a variety of needs, offers many benefits. By utilizing common parts throughout the stage, production, logistics and operations costs are all decreased.

A summary of the benefits realized by the EUS strategy relative to standard reuse and commonality strategies and the CS strategy is shown in Table 5.

Table 5: Comparing the benefits of commonalities strategies, theoretical and in practice on the SLS CS and EUS

<table>
<thead>
<tr>
<th>Benefits</th>
<th>Reuse</th>
<th>Commonality</th>
<th>CS Reuse Strategy</th>
<th>EUS Reuse Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>B</td>
<td>Space Shuttle</td>
<td>SLS CS</td>
</tr>
<tr>
<td>Reduction in development time</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Reduction in development cost</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Reduction in production cost</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Reduction in operation cost</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Decreased risk</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

ULA was created in 2006 as a joint venture between the Boeing Company and Lockheed Martin to provide launch services to the US government.
6.2 Exploration Upper Stage Architecture shown through Design Structure Matrices

A flow based and a structural DSM, shown in Figure 17 and Figure 18 respectively, are constructed for the EUS to aid in evaluation of the architecture of this stage.

![Figure 17: EUS Flow-Based DSM](image-url)
Compared to the CS flow-based DSM, the EUS demonstrates much more similarity in design between the LO2 and LH2 subsystems. This should make sharing of assets between subsystems more likely. One such example is that the fill and drain subsystems are both connected directly to the tank, compared to one connected to the feed system and the other to the tank as was observed on the CS. It is also interesting to compare the structural DSMs from the CS and the EUS. In the EUS the intertank has the greatest number of physical connections, compared to the engine section on the CS. This may provide benefits to the stage, as the intertank area is more isolated from the vibrational environments of the engine, providing less harsh environments for the components located there.

### 6.3 Challenges to Reuse on the Exploration Upper Stage

The commonality strategy developed by the EUS team seemed to be one that offered many benefits at the program level as well as the stage level. However, in order to realize these benefits the team needed to implement it effectively. This is where the team hit many roadblocks.
6.3.1 Core Stage Common Asset Implementation

Although the offset between the CS and the EUS is significantly less than that between the SLS and the Space Shuttle, there are still a number of challenges presented by this offset that reduce the amount of shared assets significantly, from the target 18% to just under 3% at the time this study was concluded.

Poor planning for forward reuse, a common pitfall in platform and commonality strategies, was present in the EUS case study. Components identified by the EUS as potential shared assets were initially designed to satisfy the exact needs of the CS rather than expanding CS component requirements envelope to encompass the needs of the EUS as well. Because the EUS is required to operate in different conditions than the CS (namely for extended periods of time in the vacuum of space with multiple engine firings), the EUS carries some requirements that are more rigorous than the CS. As a result performance gaps were found between the capabilities of the CS components and the needs of EUS, requiring either a new component design capable of encompassing the full needs of both stages or an independent component design purpose built just for the EUS. The latter strategy was chosen as it seemed easier than trying to retroactively incorporate a new design into the CS whose design process was further along. In turn divergence occurred. This was in part due to limited funding during the development of the CS. Designing for shared components is more expensive upfront than designing for a single application. The CS team was only able to consider the needs of the CS when designing CS hardware, due to funding limits. If funding and a specifically stated and strong mandate had existed to develop both the CS and the EUS concurrently, at least up to the point of initial requirement allocation, the team would have identified the differences between the CS and the EUS at the program outset and in turn would have been able to specify hardware requirements to encompass the needs of both stages.

Another source of divergence found on the EUS came about due to learning that occurred during the development of the CS prior to the start of work on the EUS. Because the development of these two stages occurred sequentially, learning during the design and test of CS components flowed directly into the design of the EUS. These lessons were most evident when examining
some of the core requirements enforced at the component level. Due to challenges encountered when designing and procuring CS components, the decision was made to update many of the requirements that are common to components such as transportation and vibration, among others. Because of these requirement changes, the components developed for the CS were no longer compliant with EUS requirements. In order to use a CS component in an EUS application, a minimum level of component testing was required, and perhaps even redesign, to show compliance to the updated requirements. A question as to whether or not these updated requirements were indeed “better” was encountered many times. Unfortunately, it is extremely difficult to compare two similar requirement sets and determine subjectively which is “better.” The decision was made by the program to stick with the updated set of requirements, resulting in additional divergence between the stages. Boas pointed out that learning is a source of divergence, in some cases beneficial. It is difficult to know, however, at this time whether the lessons of the CS applied to the EUS will be beneficial, as neither stage is yet operational. However, it is clear that sequential development, even with shorter offsets, does increase the opportunity for divergence.

6.3.2 Delta Cryogenic Second Stage Reuse Roadblocks
The DCSS reuse implementation was met with very little success, ultimately reporting no reused assets from the DCSS at the time this study concluded. The failure of reuse in this case can be tied to two primary sources: differences in requirements for manned and unmanned systems and mission dependencies in vehicle design.

When considering reuse of hardware from the DCSS on the EUS, it seems that many of the challenges that prevented reuse on the CS would no longer be relevant: the DCSS was still in production and the Boeing Company had knowledge of the stage as its original designers. However, there was one significant difference: the DCSS was not man-rated and the SLS needs to be. The DCSS was designed to carry cargo to orbit, with no requirement to support human space flight. In contrast both stages of SLS must be designed to support manned missions, as this vehicle is being built to return human space exploration capability to the United States. In turn EUS carried a number of requirements that the DCSS was not subject to. A higher level of risk is acceptable when launching satellites into orbit. However, when the payload is comprised of astronauts, the tolerance for risk is reduced. This results in higher safety factors, different
standards for inspection, added redundancy, and other more stringent requirements not levied on the DCSS.

The vehicles were also separated by the fact that they were intended for different missions. The DCSS is primarily used for short duration missions, whereas the EUS is designed to support deep space missions. The result is that these stages have different environmental requirements, further increasing the challenge of utilizing DCSS components on the EUS.

Although many of the performance based requirements for EUS components matched those of the DCSS, due to the shared RL10 engine, there were a number of requirements that were not met by the existing design. As the EUS design matured these gaps were found to be too great to permit reuse of DCSS hardware, resulting in a need to design new parts to support the EUS.

6.3.3 Minimizing Unique Part Numbers
The goal of minimizing unique part numbers on the EUS, increasing commonality within the stage, had shown potential for success at the time this study was concluded with an average of 13 parts applications (qpa = quantity per application) per part number. The CS had around 9 parts per part number, highlighting a notable increase in common parts on the EUS. These values should be evaluated further at a later date, once the EUS is closer to production, to ensure that divergence has not occurred. At this point all signs indicate that this strategy will reduce cost and schedule for the EUS, but this cannot be verified until the stage is actually produced.

6.4 Exploration Upper Stage Summary of Findings
The EUS reuse strategy was intentionally developed to be different than that utilized by the CS, due to struggles encountered with reuse on the CS. In certain areas the EUS was better able to implement strategies, while in others the same challenges found on the CS to limit reuse were also encountered by the EUS, but for different reasons. The EUS case study provides a second set of lessons learned which, when combined with those of the CS, will help to develop a strategy for actualizing the benefits of reuse on launch vehicles.

1) Shorter lifecycle offsets do present greater opportunities for reuse, however realizing the benefits of reuse is far from guaranteed. As discussed by Boas, commonality does not occur without significant effort. In the case of sharing assets across a product family,
incorporating the needs of both the lead and lag variants in the common design is vital to success.

2) **Learning may result in either beneficial or non-beneficial divergence and it is laborious to distinguish between the two.** This was demonstrated by the changing common requirements between the CS and the EUS due to learning that occurred during the design of the CS.

3) **Launch vehicle requirements are dependent upon their mission, making it challenging to share assets between vehicles with different missions.** Specifically, launch vehicles built for human exploration carry different requirements than those used for traditional cargo. Furthermore, vehicles designed primarily to deliver payloads to LEO will differ from those designed for deep space missions with and without restart capability.

4) **Minimizing unique parts in a stage may reduce development time and cost, as greater effort can be placed on designing fewer unique parts.** However further work must be done later in the vehicle lifecycle to: ensure that this is true once the vehicle is in operation and to evaluate the operational benefits of the reduced part count.

As was discussed in the literature review chapter, Boas has shown that commonality does not occur without significant effort. In order for a program to be successful in implementing commonality, the foresight to incorporate any significant design needs into the component from the start must exist. This often requires over-designing components for the lead variant, to successfully use them on future variants. If these sorts of considerations are not given when designing the lead variant, there will most certainly be gaps that will be extremely difficult to design around when trying to integrate the component into future variants. Although it will require more effort and resources up front to consider the needs of both variants, initial planning phases must aim to optimize both stages together in order to bring about the benefits of reuse to the vehicle in the long run.
This page has been intentionally left blank.
7 Conclusions, Recommendations and Forward Work

Commonality and reuse strategies offer the promise of benefits at the program level such that there is no doubt that these methods will continue to be considered and at least partially employed in future space launch programs. To begin the discussion of conclusions and recommendations, consider the two questions that were posed at the start of this research project and thesis:

1) Is there something inherently unique about launch vehicle design that prevents or limits reuse?

2) If not, are there strategies that can be implemented to enforce component reuse during launch vehicle design?

The results of this thesis seem to indicate that launch vehicle design is inherently “unique” and that this uniqueness brings about a significant barrier to reuse. Compared to more traditional products (e.g. automobiles, airplanes, electronics), the cadence at which launch vehicles are designed, built and operated is extremely slow. This leads to significant offsets in product lifecycles which results in product and process obsolescence, challenging the implementation of reuse between families of launch vehicles. Launch vehicles also operate in harsh environments, many of which are uniquely determined by the vehicle design itself, its launch conditions, and the mission. These unique environments create variation in the performance requirements of components, specifically those requirements related to vibrational and thermal performance. Because launch vehicle components are designed with very little margin, any change that causes a harsher environment may render an existing design obsolete, further diminishing the likelihood of successfully reusing designs between launch vehicle families.

Even though this thesis does seem to indicate that launch vehicles are unique in several ways when it comes to reuse and commonality, it still seems that the potential to successfully bring about the benefits of commonality and reuse strategies does exist if specific steps are taken. This conclusion hopes to provide insight into some of these steps, while also offering opportunities for future work. Finally, this thesis provides a foundation from which individual organizations can develop platform strategies to meet their unique needs.
7.1 Conclusions

The benefits of reuse and commonality strategies are extremely attractive when developing launch vehicles, and will likely continue to be pursued to reduce the cost, schedule and risk of new launch vehicle development. However as was observed on both the CS and the EUS of the SLS program, reuse can be an extremely challenging strategy to implement successfully. In turn, it is critical to develop a strategy that supports the complexities that come with launch vehicle development, such as long lifecycle offsets and high system complexity, in order to realize the benefits of reuse. The lessons learned in this thesis promote a strategy built upon the following three findings:

1) Commonality should be pursued through sharing of assets between stages on a single launch vehicle as well as between launch vehicles in an evolvable family. In contrast reuse across launch vehicles is unlikely to be successful due to challenges associated with significant lifecycle offsets and mission-specific requirements.

2) Commonality will not occur by coincidence. It must be planned for and defined from the start of a program and incentivized throughout. For example the requirements envelope must be explicitly quantified and cascaded down rapidly to the subsystems and individual component level.

3) Divergence will occur, requiring the program to incorporate the maximum level of commonality practical during the planning phases of the program.

Each of these conclusions will be expanded upon below.

7.1.1 Commonality Provides a Better Strategy Than Heritage Reuse

Commonality within a launch vehicle is a much more realistic strategy to realize the benefits of reuse than heritage reuse across launch vehicle platforms. Boas found that lifecycle offsets make reuse increasingly difficult, and this thesis further supports that hypothesis by examining the lifecycle offsets of launch vehicles, which tend to be decades. It was shown that offsets of this magnitude are too significant to allow for successful reuse across launch vehicles. Instead of attempting to incorporate heritage designs into a new vehicle, the focus should be shifted to increasing commonality within the vehicle itself.
Commonality across launch vehicles is much more likely to be successful when the vehicles share a mission. For example man-rated launch vehicles carry different requirements than do unmanned vehicles. It was shown that reuse of unmanned assets in a manned vehicle is unlikely, due to lower tolerance for risk in manned systems. Specific differences are due to safety margins and expected levels of redundancy in the case of partial or total component failures during launch. In turn it is critical to consider differences between vehicle missions when proposing reuse. This challenge is lessened when the missions are tied together, which occurs when a family of launch vehicles is developed. In that case the main difference between different variants of the launch vehicle in the same family may be simply the launch capacity (IMLEO – injected mass into low Earth orbit) and target parking orbit. In the case of SLS, which is planned as an evolvable vehicle, there exist opportunities to design assets that can be utilized across launch vehicles that are all part of the same family and share a common mission.

A launch vehicle, whether considered at the stage level or as a member of a family of evolvable launchers, will still be subjected to lifecycle offsets, as resources often require the stages/vehicles to be developed sequentially rather than in parallel. However, the offsets will be much shorter than between independent launch vehicles, increasing the opportunity for successful sharing of assets.

7.1.2 Plan for Commonality at All Program Phases

As discussed by many scholars who have researched this topic in the past, commonality and/or reuse will not occur by coincidence. If it has been determined that commonality provides the best avenue to program success, then an effort must be made to ensure that the commonality strategy is implemented successfully. This requires planning for and incentivizing commonality in all phases of the program. Specifically, in the case of launch vehicles a significant amount of effort must be devoted to defining requirements that will encompass the needs of both the current stage and future stages if commonality is to be successful. When a lifecycle offset exists, the requirements of the lead variant often take precedence because they are being more actively considered at the outset of the program. However, this can lead to divergence when development of the latter variants begins.
Instead a holistic approach to requirement definition and allocation at the product level must occur at the outset of the program. This will allow for the stricter of the requirements to be identified, which may occur on either stage. Components should then be designed to meet the more strict requirements in order to allow for forward reuse. Although it is certain that this will result in non-optimized components at the stage level, the benefits of commonality at the vehicle or program level should outweigh the penalty.

7.1.3 Divergence Will Occur and Should Be Measured

Any reuse or commonality strategy will be impacted by divergence. To counter this fact reuse levels should be maximized during the planning phases of a program. Care should be given to ensure that divergence is properly measured throughout the program, as there is no way to identify potential sources of non-beneficial divergence if divergence itself is not tracked. Once divergence is found, it should be evaluated and categorized as either beneficial or non-beneficial. Beneficial divergence should be permitted whereas non-beneficial divergence should be prevented. Processes should be created to incentivize maintaining commonality when beneficial. This includes the careful use and calibration of lifecycle cost models at the component level.

7.2 Recommendations

The case studies performed in this thesis brought about a number of recommendations to improve the likelihood of successfully capturing the benefits of reuse. These recommendations have been generalized to be applicable to other launch vehicle projects beyond SLS.

1) **The utilization of shared assets is highly dependent on cross-functional teams capable of recognizing opportunities for commonality.** The organizational structure should be designed to support true multi-disciplinary decision-making.

2) **Develop systems to support engineers in implementing reuse and commonality.** One such system is a component repository tool that identifies critical features of existing hardware on the launch vehicle. There is currently no database to query to find an existing design that may meet a designer's needs. A simple tool that identifies all hardware on the vehicle and its critical features will permit engineers to choose an existing design rather than creating something new. This will result in fewer unique parts
on the launch vehicle, cheaper production costs and simpler logistics. Beyond such databases the establishment of “digital twins” in a MBSE environment should be considered.

3) **Incentivize commonality at all levels.** It has been show, both in this study and those by Boas, that commonality will not occur without proper incentives. These incentives must exist from the highest level of management to the most junior engineer. The greatest challenge exists at the engineering level, as most engineers are trained to solve a problem by designing a unique solution. In turn there is no intrinsic motivation to utilize an existing design. In turn engineers should be trained on the benefits of shared assets and incentivized to utilize these assets when beneficial to the program.

4) **Define a commonality champion who is responsible for measuring commonality throughout the program lifecycle.** It is critical to have an individual, or a team, who is responsible for promoting practices that encourage asset sharing. This individual should also ensure that adequate consideration is given to the benefits/penalties of commonality in each situation, and that beneficial commonality is captured while non-beneficial reuse is avoided.

5) **Perform trade studies to ensure changes to new variant requirements will provide a tangible benefit that is greater than the cost of developing new assets for that variant.** It is extremely challenging to determine if one set of requirements is better than another. However, any change to requirements between the lead and lag variants increases the likelihood of divergence. In turn greater effort should be given to quantifying the value of improved requirements compared to the costs of developing new assets. This will help guide a decision as to whether changing requirements will actually result in a better product at the vehicle level.

6) **Develop architectural maps that highlight subsystem interactions.** Utilize these maps in making architectural level decisions to bring about greater levels of system commonality. These maps can also serve to identify opportunities to decrease unique part
numbers, which will bring about development and operational benefits. Ideally such maps, such as the well known DSM, should be auto-generated by the CAD/CAE/PLM systems.

7) **Develop a reuse process that requires baselining reuse at the outset of the process, and limits the creation of new part numbers to the end of the process.** In order to incentivize engineers to reuse hardware, the reuse process must be aligned to the goal of reuse. This requires pushing the baseline of reuse earlier into the process and creating a barrier to entry to getting a new part number approved. One potential process flow is shown in Figure 19. This figure contrasts with the process defined in the CS SEMP (Figure 8). Certainly this is just one example of a potential process, and it should be iterated on by organizations to fit within standard procedures.

![Figure 19: Proposed Reuse Process to Improve Likelihood of Reuse Success](image)

7.3 **Forward Work**

The work performed in this thesis offers many opportunities for future work, from developing accurate cost estimates of hardware development to company-wide opportunities for shared assets.
One of the biggest challenges encountered during this study of reuse related to developing a model to evaluate the actual economic benefit of shared assets versus new development. This could be an interesting area for future work. The hardware developed for launch vehicles is quite costly, however distributing these costs between design, development, production and operations is not done easily. In turn, there remains confusion as to the actual cost of developing new hardware when compared to the cost of performing some additional development analysis or testing to prove an existing asset complies with requirements. It would be interesting to develop a parametric model that estimates the true development, production and operation costs of developing a new asset, for example a large propellant valve, based on size, complexity, weight, and other critical parameters in launch vehicle design. This model would allow for a more objective trade of the value of utilizing shared assets versus developing a unique asset. Certainly there are cases where the benefits of having a unique asset that is perfectly suited to its application trades better than trying to force an existing component to fit some unique set of needs. However, as it stands this trade is subjective. Developing a quantitative model that ties to program cost could help reduce the subjectivity of this trade.

Another opportunity exists to consider opportunities for reuse or commonality across programs in a single company. As was shown earlier launch vehicles are built too infrequently to offer an opportunity for reuse across product lines. However, if the whole of Boeing Defense, Space and Security is considered, there are a variety of missile and defense products being built simultaneously. Currently these programs are all independent, but there are likely opportunities to utilize assets across these programs to lower the total development cost of the programs. This would be a very challenging project, but perhaps beginning with a small enough scope would allow for identification of opportunities for sharing assets. Consider, for example, instrumentation that is used in these types of products, such as accelerometers. Currently all of these programs source instrumentation independently. Perhaps there exists an opportunity to pool the needs of the program and source an array of accelerometers that meet all the needs. A study could be done to compare the coordination cost versus the benefit gained by developing a suite of accelerometers one time to meet a variety of needs. Depending on the result of this study, it could be determined if sharing of assets is beneficial and if not what area of the trade would need to change to bring about sufficient benefits.
These areas of future work offer potential avenues to expand upon the lessons learned and recommendations developed in this thesis. The ability to continue to further explore the universe in which we live depends upon the development of reliable launch vehicles. These vehicles can be made more efficiently, both with respect to time and cost, if a commonality and reuse strategy is employed. Many industries (e.g. automotive) have successfully utilized commonality and reuse strategies to support the development of new products. This thesis aimed to identify some of the marked differences between cars and launch vehicles that complicate the implementation of reuse. These differences helped to develop a set of recommendations from which a reuse strategy for future launch vehicles can be built.
## Appendix A: List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOM</td>
<td>Bill of Materials</td>
</tr>
<tr>
<td>CAD</td>
<td>Computer-Aided Drafting</td>
</tr>
<tr>
<td>CAE</td>
<td>Computer-Aided Engineering</td>
</tr>
<tr>
<td>CS</td>
<td>Core Stage</td>
</tr>
<tr>
<td>DCSS</td>
<td>Delta Cryogenic Second Stage</td>
</tr>
<tr>
<td>DSM</td>
<td>Design Structure Matrix</td>
</tr>
<tr>
<td>EUS</td>
<td>Exploration Upper Stage</td>
</tr>
<tr>
<td>IMLEO</td>
<td>Injected Mass into Low Earth Orbit</td>
</tr>
<tr>
<td>IPT</td>
<td>Integrated Product Team</td>
</tr>
<tr>
<td>LEO</td>
<td>Low Earth Orbit</td>
</tr>
<tr>
<td>LH2</td>
<td>Liquid Hydrogen</td>
</tr>
<tr>
<td>LO2</td>
<td>Liquid Oxygen</td>
</tr>
<tr>
<td>MBSE</td>
<td>Model Based Systems Engineering</td>
</tr>
<tr>
<td>MPS</td>
<td>Main Propulsion System</td>
</tr>
<tr>
<td>NASA</td>
<td>National Aeronautics and Space Administration</td>
</tr>
<tr>
<td>PDR</td>
<td>Preliminary Design Review</td>
</tr>
<tr>
<td>PLM</td>
<td>Product Lifecycle Management</td>
</tr>
<tr>
<td>QPA</td>
<td>Quantity per Application</td>
</tr>
<tr>
<td>RL10</td>
<td>Delta IV Second Stage Engine</td>
</tr>
<tr>
<td>SEMP</td>
<td>Systems Engineering Management Plan</td>
</tr>
<tr>
<td>SLS</td>
<td>Space Launch System</td>
</tr>
<tr>
<td>RS-25D</td>
<td>Space Shuttle Main Engine</td>
</tr>
<tr>
<td>TRL</td>
<td>Technology Readiness Level</td>
</tr>
<tr>
<td>TVC</td>
<td>Thrust Vector Control</td>
</tr>
<tr>
<td>ULA</td>
<td>United Launch Alliance</td>
</tr>
</tbody>
</table>
Bibliography


