A METHODOLOGY FOR MODELING THE VERIFICATION, VALIDATION, AND TESTING PROCESS FOR LAUNCH VEHICLES

A Dissertation
Presented to
The Academic Faculty

by

Alicia Sudol

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering Engineering

Georgia Institute of Technology
December 2015

A METHODOLOGY FOR MODELING THE VERIFICATION, VALIDATION, AND TESTING PROCESS FOR LAUNCH VEHICLES

Approved by:

Professor Dimitri Mavris, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Professor Daniel Schrage
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Olivia Pinon
School of Aerospace Engineering
Georgia Institute of Technology

Professor Graeme Kennedy
School of Aerospace Engineering
Georgia Institute of Technology

Mr. Reginald Alexander
Marshall Space Flight Center
National Aeronautics and Space Administration

Date Approved: September 18th, 2015

To my husband,

Tomek Sudol

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank the people who have helped me get here today through their support and encouragement. First I would like to thank my committee members, Dr. Dimitri Mavris, Dr. Olivia Pinon, Dr. Daniel Schrage, Mr. Reginald Alexander, and Dr. Graeme Kennedy. I appreciate all of the time you spent and feedback you gave to help shape this work. I would especially like to thank Dr. Mavris, my advisor. Joining ASDL has been a wonderful experience for me, and I'm grateful for the opportunity. The environment you have created in this lab provides a great foundation for students to be innovative and explore research areas that really inspire them.

I would also like to thank my research engineer, Stephen Edwards. Your positive attitude and thoughtful feedback has helped me get through some of the more stressful times at ASDL. I appreciate you always having an open door to answer questions and give advice about this often times uncertain process. Thank you to my fellow basement dwellers: Nate Knisely, Eric Zellers, Mock Abdelaal, Huy Tran, and Tyler Milner. I have appreciated your sense of humor and thoughtful conversations, and I will never drink another cup of coffee without thinking of the Georgia Tech Starbucks.

I would like to thank my wonderful husband, Tomek, for all of his support. I could not have done this without you. Also, thank you to my mom, dad, and sisters for all of your encouragement, and to Amanda for always having an ear ready to listen.

You have all influenced me in one way or another throughout this PhD process, and I am here because of your support. Thank you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ALS	Advanced Launch System
AMSAA	Army Material Systems Analysis Activity
CBO	Congressional Budget Office
CDF	Cumulative Distribution Function
CER	Cost Estimating Relationships
CDR	Critical Design Review
CPM	Critical Path Method
CRW	Cost of Rework
DDT&E	Design, Development, Test, & Evaluation
DeMAID	Design Manager's Aide for Intelligent Decomposition
DES	Discrete Event Simulation
DoD	Department of Defense
DSM	Design Structure Matrix
ECSS	European Cooperation on Space Standardization
ESA	European Space Agency
ESAS	Exploration System Architecture Study
FEF	Fix Effectiveness Factor
FMEA	Fault Mode and Effects Analysis
FMECA	Fault Mode, Effects, and Criticality Analysis
FTA	Fault Tree Analysis
GAO	Government Accountability Office
GERT	Graphical Evaluation and Review Technique
INCOSE	International Council on Systems Engineering
IMS	Integrated Master Schedule
ISS	International Space Station

LH2	Liquid Hydrogen
LOC	Loss of Crew
LOM	Loss of Mission
LOX	Liquid Oxygen
LRR	Launch Readiness Review
LV	Launch Vehicle
MCS	Monte Carlo Simulation
MPCV	Multi-Purpose Crew Vehicle
M&S	Modeling and Simulation
MTBF	Mean Time Between Failure
NAFCOM	NASA Air Force Cost Model
NASA	National Aeronautic and Space Agency
NDSM	Numerical Design Structure Matrix
OSP	Orbital Space Plane
PCM	Parts Count Method
PDF	Probability Density Function
PDR	Preliminary Design Review
PERT	Program Evaluation and Review Technique
PNET	Probabilistic Network Evaluation Technique
PRA	Probabilistic Risk Analysis
Prodecol	Production Development Control Lever
RBD	Reliability Block Diagram
RLV	Reusable Launch Vehicle
RM	Risk Matrix
RMSE	Root Mean Square Error
RPL	Rated Power Level
RPN	Risk Priority Number

SLI	Space Launch Initiative
SLS	Space Launch System
SME	Subject Matter Expert
SSME	Space Shuttle Main Engine
TAAF	Test-Analyze-and-Fix
TPM	Technical Performance Measure
TRL	Technology Readiness Level
TUF	Technical Uncertainty Factor
TURC	Technical Uncertainty Rework Cycle
VRM	Verification Requirements Matrix
VV&A	Verification, Validation, and Accreditation
VVT	Verification, Validation, and Test
WBS	Work Breakdown Structure

SUMMARY

Completing the development process and getting to first-flight has become a difficult hurdle for launch vehicles. Program cancellations in the last 30 years were largely due to cost overruns, schedule slips, and a lack of congressional support. Since the Space Shuttle retirement in 2011, cancellations have plagued launch vehicle development programs. A recent example is the cancellation of the Constellation program in 2010, in favor of the new Space Launch System (SLS). SLS is scheduled for its first manned flight in 2021, and the United States now faces a 10 year gap in the ability to independently launch people into space. Despite the demonstrated consequences of cost overruns, the current NASA budgetary environment is imposing even more constraints on SLS than previous launch vehicle programs. A shifting focus to reducing the cost and schedule during the design, development, testing and evaluation (DDT&E) process is necessary to give these programs the opportunity to succeed.

Comparing the life cycle progression of NASA programs to their cost and schedule history illustrates that the largest percentage of growth occurs during NASA Phases C and D. During these phases the system is undergoing verification, validation, and testing (VVT) to eliminate defects and gain knowledge about the system. Unplanned rework cycles that take place during these phases can account for up to 75% of total development cost. Current industry standard VVT planning is largely subjective with no method for evaluating the impact of rework. Although a few academic studies have looked into VVT planning, weaknesses exist in their evaluation of rework cycles and overall assessment of individual VVT activities. This research aims to address the gap of estimating the impact of rework during VVT planning to help improve the

launch vehicle Design, Development, Testing & Evaluation process.

The Rework Impact on Verification, Validation, and Test Strategies (RIVVTS) methodology was developed to meet this goal by quantitatively capturing the effects of unplanned rework during VVT for launch vehicle systems and subsystems. Using the strengths of existing reliability growth techniques, rework cycles are first estimated using FMEA failure distributions. A dependency structure matrix is used to model the impact of rework cycles using the relationships between VVT activities. To provide a complete assessment, four metrics were chosen to evaluate a VVT strategy: reliability, cost, schedule, risk. This approach allows for a quantitative cost, schedule, and reliability projection to be generated. The resulting output distributions are then used to calculate the risk level of a given VVT strategy using a quadratic impact function.

The method developed is first tested on a case study, comparing the method output to actual data from the RS-68 engine. The purpose of this example is to illustrate that this methodology accurately captures the occurrence and impact of rework cycles seen during VVT. The secondary purpose of the example is to validate the development of the cost, schedule, and reliability assumptions for future applications of this method.

Finally, this method is applied to evaluate alternative VVT strategies for a relevant launch vehicle subsystem. First, the method outputs are compared against the overall research objectives to confirm that they are met. Then the alternative VVT strategies are evaluated to determine how the impact of rework can be mitigated. The results give interesting observations regarding the benefit of comprehensive component testing versus early integrated testing. Ultimately, this final application problem demonstrates the merits of the RIVVTS methodology in evaluating VVT strategies. RIVVTS provides a risk-informed decision making tool to reduce the impact of rework cycles on the verification, validation, and testing process of launch vehicle systems and subsystems.

CHAPTER I

INTRODUCTION

1.1 Motivation

The National Aeronautics and Space Administration (NASA) Authorization Act of 2010 established the development of a heavy lift launch vehicle (LV) to be a part of the new Space Launch System (SLS) [7]. As a follow on to the Space Shuttle Program, the human rated SLS will be the first exploration class LV since the Saturn V over 40 years ago [133]. The design of such a large and complex system requires the integration of multiple disciplines, many of which have conflicting objectives. To ensure the optimal design, compromises are necessary to attain a balance among the performance, reliability, cost, and schedule requirements. In addition to the innate difficulties of designing such a system, the SLS is placing particular emphasis on improving the affordability and sustainability of the program [133, 39]. Affordability, in this context, is defined as the ability to develop and operate the SLS within the national means to sustain funding for the program [133]. Therefore, it will be critical to understand the cost and schedule risks as well as the technical risks during the design, development, test and evaluation (DDT&E) period.

Launch vehicles are inherently expensive to develop and produce, and often do not reach first flight [27, 132]. The space industry has become a 'start-stop-restart' process in the last few decades, filled with programs that never make it out of the DDT&E phase. As far back as the late 1980s, the length of the required development time for these complex systems is too extensive to keep up with changes in requirements and national priorities. An example of this is the Advanced Launch System (ALS) which was a joint NASA and Department of Defense (DoD) program with the aim of reducing the cost of putting large payloads in orbit [86]. Two years after the start of the ALS development, reductions in funding and a shift of focus to lightweight weapons lead to its cancellation in 1989. Following that, the National Launch System (NLS) program was created to support the goals described by the Bush administration's National Space Policy Directive 4, National Space Launch Strategy, to develop a new Expendable Launch Vehicle with a focus on improving reliability, cost, and responsiveness [86]. The NLS program was also canceled after two years due to poor communication and differing priorities between NASA and the Air Force. From 1995-2000, NASA focused on the development of Reusable Launch Vehicles (RLV). Two RLV flight test programs were initiated, X-33 and X-34. The X-33 was a joint program with Lockheed Martin that was pushing the boundaries in composite fuel tanks [67]. After suffering technical set backs and failing to meet its goals, the program was deemed too costly and canceled [86]. The X-34, a joint program with Orbital Sciences Corporation, was similarly canceled after initial flight tests determined that additional costs did not justify the potential benefits [86]. After funding for the X-33 and X-34 was cut, focus shifted to the new Space Launch Initiative (SLI). The SLI was a joint industry-government effort from 2000-2004 to determine the best approach to developing a Space Shuttle replacement [43]. It was yet another example of a program that did not make it out of the DDT&E phase as it was canceled in favor of the Orbital Space Plane [43]. OSP was also canceled when NASAs Exploration System Architecture Study (ESAS), which later evolved into the Constellation Program, was formed [43]. Figure 1 highlights these short lived space transportation programs over the last three decades.

This 'start-stop-restart' cycle that space transportation programs have endured does little to improve confidence in NASA's ability to develop the necessary technologies for advanced space flight. Program efforts to achieve significant advances in technology are being stifled in support of a more risk adverse environment, just

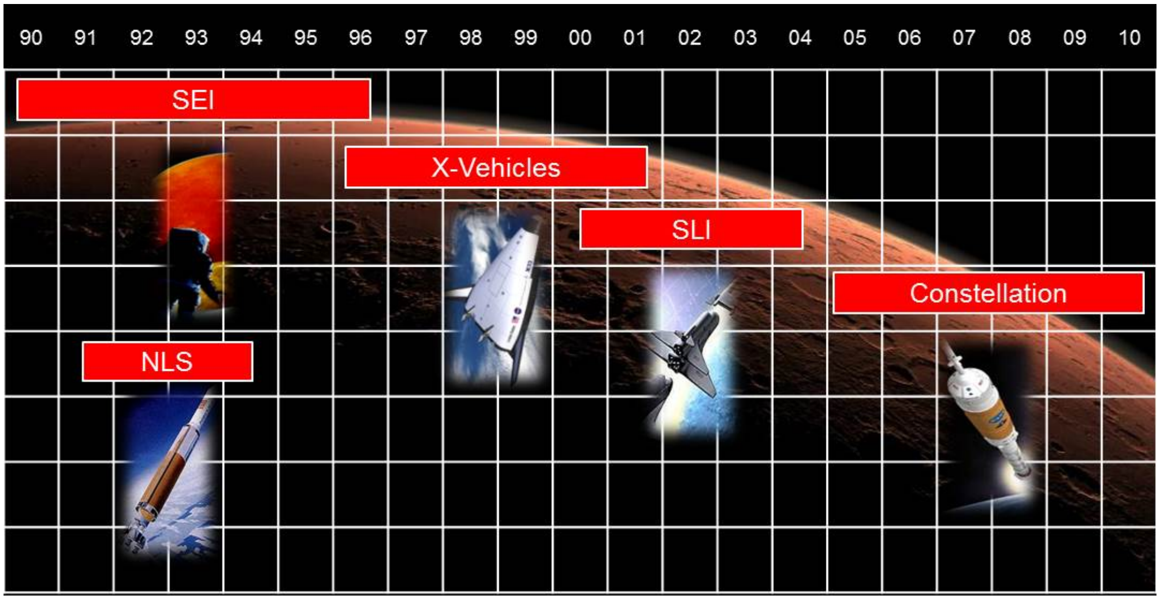


Figure 1: Canceled space transportation programs.

to ensure a program successfully completes the DDT&E phase and makes it through to a first flight. Looking further into the effects of canceling the Constellation program shows that it is more than simply a monetary loss. The Constellation program was formed in 2005 and consisted of the Ares-I and Ares-V launch vehicles, and the Orion Multi-Purpose Crew Vehicle (MPCV) [138]. The original program schedule allowed for a two year gap between the retirement of the Space Shuttle in 2010 and the availability of the Ares-1 and Orion to support the International Space Station (ISS) in 2012 [109]. In 2009, the schedule slipped to 2015, resulting in a five year gap. When the program was canceled in 2010 due to poorly phased funding and additional schedule slips, the U.S. was left without the ability to independently launch people into space. The current Space Launch System (SLS) is not expected to launch a manned mission until 2021, resulting in a 10 year gap. This will be the longest gap in America's manned space flight presence since the Apollo-Shuttle gap in the late 70s [109]. Until the SLS or a commercial alternative becomes available, NASA will pay Russia between $50-$70 million per seat to train and transport crew members to the ISS [109]. NASA has not had the political inertia to sustain the funding necessary to back their complex programs [86], and their current budget environment imposes even more constraints on the new SLS program than previous launch vehicle programs [39]. Creating a sustainable development plan is critical to the long term success of a program. These additional demands highlight a gap in the ability to create a sustainable DDT&E plan for these complex systems, and that is the primary motivation for this research. The following section goes into more detail about the risks of a DDT&E plan to further scope the goals of this research.

After determining the need for the development of a sustainable DDT&E plan, it is necessary to identify where improvements to DDT&E planning can be made. The lifecycle of a project is divided up into phases to help plan and manage development of the system [131]. Figure 2 illustrates how NASA defines the life cycle phases for its projects, lettered Pre-Phase A, Phase A, Phase B, Phase C, Phase D, Phase E, and Phase F. The phase boundaries are defined at natural points for progress assessment and design reviews to determine if the project should continue to the next phase. Examples of these key decision points include Preliminary Design Reviews (PDR), Critical Design Reviews (CDR), and Launch Readiness Reviews (LRR).

Formulation			**Approval**	**Implementation**				**End of Mission**
Pre-Phase A: Concept Studies	Phase A: Concept and Technology Development	Phase B: Preliminary Design & Technology Completion	Proceed to Implementation Phase	Phase C: Final Design and Fabrication	Phase D: System Assembly, Integration, Test, and Launch	Phase E: Operations and Sustainment	Phase F: Close-out	Mission Concluded

Figure 2: NASA lifecycle phases [131].

Many studies have been conducted to improve the quality of early conceptual design, particularly on expanding the ability to explore more architecture space during Pre-Phase A and Phase A [22, 26, 91]. These methods focus on the design and development portion of the DDT&E phase, while testing and evaluation of large systems has been shown to represent 40% of the total life cycle cost [21]. In particular, technical failures uncovered during testing account for a significant portion of the development schedule, cost, and effort across many industries. A Rocketdyne review of the F-1 and J-2 advanced engine programs found that 73% of total development cost was absorbed in eliminating failure modes [64]. Of that 73%, 75% of the effort to eliminate the failure modes was spent on rework, i.e. re-design, re-manufacturing, and re-testing. A semiconductor manufacturing company found similar trends, estimating that unplanned design and manufacturing iterations account for between 33%-66% of total development time [110]. Software projects also state that 40%-50% of their efforts are spent avoiding rework [21]. Unplanned rework dominates much of the development cost and schedule because it is seldom explicitly considered. NASA's Discover and New Frontiers Programs found that no analysis is performed during Phase A and B to determine the amount of redesign or additional testing that might be necessary during the course of Phase C and D design implementation [67].

A previous study investigated the mass, power, cost, and schedule growth of 20 NASA missions [56]. Figure 3 shows that the majority of cost and schedule overrun for these projects are not seen until Phase C and Phase D, the implementation phases or test and evaluation portion of DDT&E [131]. NASA's Phase C is Final Design and Fabrication where hardware fabrication is initiated, and engineering test units that closely resemble actual hardware are tested to establish confidence in the design. A series of CDRs are conducted during Phase C, at the system-level and subsystem levels, prior to fabrication [131]. Phase D is the System Assembly, Integration and Test, Launch, and Check Out. During this phase the complete system goes through a verification, validation, and testing (VVT) process. VVT is a systems engineering tool that is meant to increase knowledge about the system and ensure it is high quality, functionally sound, meet the user's needs [50].

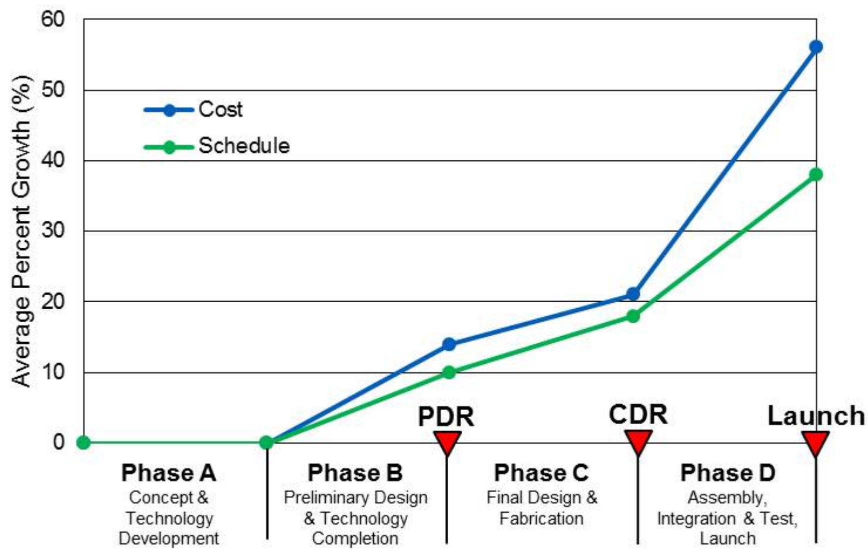


Figure 3: Comparison of average system growth over time [56].

Recognized by the major launch vehicle developers and the International Council on Systems Engineering (INCOSE) as a vital part of the development process, VVT planning currently relies heavily on subject matter experts and does not consistently include statements of risk and uncertainty [108, 131, 153]. Generally, a fully comprehensive VVT procedure is not realistic due to cost and schedule constraints [136]. The lack of structure in developing an efficient VVT plan has been addressed in the literature, which is presented in Chapter 2. Many authors have concluded that unplanned rework efforts can have a significant effect on the success of a development program, but few methods exist to quantify these effects during VVT. While some of these methods recognize the importance of rework, few include the impacts of rework cycles, and no VVT planning methods provide a direct approach to obtain the probability that rework will occur.

1.2 Research Focus and Organization

The primary motivation for this research is the difficulty launch vehicle programs have seen historically in successfully completing DDT&E and making it to first flight. The previous section identified the importance of creating a sustainable VVT plan and considering the impact that rework cycles can have on that process. Therefore the focus of this thesis is on the development of a methodology for modeling the verification, validation, and testing plan for launch vehicle programs that will quantitatively capture the effects of rework cycles. VVT is an iterative process that takes place to different degrees during each design phase. This research is strictly focused on the formal VVT that occurs during Phase C/D, and will simply be referred to as VVT for the remainder of the document [131, 2].

Chapter 2 provides a review of the different approaches for VVT planning. Each technique is introduced, and the strengths and weaknesses of how they assess the value of VVT activities, and how they address the impact of rework cycles is discussed. Following this section, observations are drawn from these existing techniques and the objective for this research is defined. Research questions are posed in the remaining sections of Chapter 2 to introduce the components necessary to build the Rework Impact on Verification, Validation, and Test Strategies (RIVVTS) methodology and meet the objective. Using a combination of research questions, literature review, and experimentation, the components of RIVVTS are formulated in Chapter 3. The experiments are designed to test the hypotheses formulated in Chapter 2. The setup of each experiment is introduced first, followed by the results and observations of that experiment, and conclusions are drawn to either support or refute the hypothesis.

Chapter 4 presents a detailed description of each step in this methodology. At the end of Chapter 4 a case study is presented to validate the output of this method. Following the case study, Chapter 5 presents an application of this methodology to alternative VVT strategies for one system. A detailed discussion of the method outputs are discussed, and the completion of the research objective is verified. The final chapter concludes the thesis by presenting a summary of the findings from the literature review, experiments, and application problem. Finally, the contributions of this work are discussed, and future research or extensions of this methodology are suggested.

CHAPTER II

BACKGROUND RESEARCH

This chapter contains a literature review of launch vehicle VVT process planning and is intended to identify the gaps by establishing what the current state-of-the-art is and how it can be improved. The first section reviews the VVT requirements and current industry standards in planning a VVT strategy. The next section is an in-depth review of the academic research that has been done to advance VVT planning methodologies and addresses some of their shortcomings. Section 2.4 covers the specific elements of VVT that are necessary for decision making, and discusses how they have been treated individually in the literature. Sections 2.5-2.8 discuss those specific elements individually. The last section reviews the current methods that exist in estimating the rework probabilities and impacts.

2.1 Definitions and Purpose of VVT

As mentioned previously, verification, validation, and testing is a set of activities and processes that are chosen to increase knowledge of the system and demonstrate that the system or product meets all requirements. The definitions of verification and validation differ slightly between organizations, but have a common goal. These definitions from the major launch vehicle developers and systems engineering organizations are stated below:

Verification

NASA [131]	"..process of confirming that deliverable ground and flight hardware and software are in compliance with design and performance requirements."

DoD [108]	"...process used to demonstrate that the system design meets applicable requirements and is capable of sustaining its operational role during the life cycle."
ECSS [4]	"...process which demonstrates through the provision of objective evidence that the product is designed and produced according to its specifications and is free of defects."
INCOSE [153]	"...addresses whether the system, its elements, its interfaces, and incremental work products satisfy their requirements."

Validation

NASA [131]	"...process of determining the effectiveness and suitability of the product for use in mission operations by typical users."
DoD [108]	"...process used to ensure the project has confidence that the as-built product will perform its intended functionality in its intended operational environment.
ECSS [4]	"...process which demonstrates that the product is able to accomplish its intended use in the intended operational environment."
INCOSE [153]	"...confirms that the system, as built, will satisfy the user's needs."

The trend in these definitions is that verification deals with satisfying the specified requirements and ensuring the system was built correctly; and validation deals with ensuring that the right system was built regardless of what the exact specifications were in the beginning. Although these two processes address different issues, they use a common set of activities. The verification and validation methods include [102, 53, 7]:

1. Inspection - Visual examination or other non-destructive evaluation to determine product conformance with characteristics best determined by observation (e.g. weight, dimensions, color, or other physical characteristics).

2. Analysis - Evaluation of data by generally accepted analytical techniques to determine the system will meet specified requirements (i.e. modeling and simulation, systems engineering analysis, or probabilistic calculations). Typically used when physical testing in the actual environment cannot be achieved or is cost-prohibitive.

3. Demonstration - Determination of product conformance through the operation of a test article, relying on observation and no or minimal special test equipment and instrumentation.

4. Test - Operation of the system under controlled conditions to quantitatively determine if design or performance requirements are met in applicable environments. Testing is the preferred requirement verification method when the system contains critical failure modes that could result in loss of life or loss of mission.

Testing can serve multiple purposes and be a subset of both verification and validation. Primarily, testing is an activity where the system is employed to verify the design, validate the operational unit, or identify as many defects as possible [61]. Compiling these definitions, the general purpose of a VVT strategy is to gain knowledge about the system while eliminating as many manufacturing and design defects as possible within the allotted time and budget.

VVT planning is the process of determining which activities to use (e.g. analysis, test, inspection, or demonstration) based on the required system performance, risks, and cost and schedule impacts. A review of how VVT planning is currently done by these organizations is provided in the next section, followed by an in depth review of the advances made in process planning in the literature.

2.2 Verification, Validation, and Test Planning

To understand how VVT planning is currently done for launch vehicle programs, a review of NASA, the Department of Defense (DoD), and the European Cooperation for Space Standardization (ECSS) documentation was conducted. In addition, because VVT is fundamentally a System's Engineering component, a review of documentation from the International Council On Systems Engineering (INCOSE) was also completed. These organizations have published multiple requirements documents for verification and testing standards. The VVT planning process is largely based on expert opinion, and therefore no standard planning methodology was found. Another indication of a lack of uniformity is the publication of VVT standards by individual NASA centers. What can be assembled from these requirements are common VVT tools and terminology.

2.2.1 Industry Standards and Requirements Definition

NASA's System's Engineering Handbook contains the primary guidelines for program verification and validation [131]. The design process at NASA consists of seven phases, Pre-Phase A and Phases A-F. The verification planning is done in more detail progressively throughout the design life cycle [131]. Phase D includes the implementation of verification and validation of the system, including testing the system in its operational environment. It states that each program's verification and validation plan should be tailored to the specific project it supports. The methods used may depend on payload classification, cost, schedule, risk implications, or many other aspects. Due to the individualization of each VVT process, this handbook and other documents only contain guidelines for developing an effective VVT plan. Another handbook with guidelines and requirements is the Marshall Space Flight Center Verification Handbook, Volumes I and II [2, 1]. Even within the one NASA center, the verification documentation and processes are not uniform. Each project is allocated the responsibility of developing a VVT program that considers the cost, schedule, and risk impacts of their specific project. While the actual verification process activities do not begin until Phase C, preliminary methods are planned during Phase B.

The DoD documentation on launch vehicle VVT are military standards MIL-STD-1540, Product Verification Requirements for Launch, Upper Stage, and Space Vehicles, and MIL-HDBK-340, Test Requirements for Space Vehicles [108, 106]. MIL-STD-1540 directly addresses launch vehicle verification plan development to demonstrate that the system design meets its requirements. It contains more specifics than the NASA documents because it is directly written for space vehicles, but similarly specifies that the final VVT plan should be tailored to each project [108] according to various factors. MIL-HDBK-340 details the government requirements for each test level, e.g. qualification and acceptance tests, at the unit, subsystem, and system-levels [106].

INCOSE System's Engineering Handbook also describes the progression of the VVT process during a project's development phases [153], as illustrated in Figure 4. As the system transitions from early development phases into final system design and operations, the V&V requirements also change. It mentions a Verification Cross Reference Matrix, also referred to as a Verification Requirements Matrix (VRM) by NASA and the DoD documents [131, 108]. The VRM is a matrix that lists the system requirements and any required tests for verifying that particular requirement. INCOSE goes further by giving each requirement a unique identifier to be used to improve traceability while developing test plans and procedures [153]. Although these organizations provide different levels of detail regarding the VVT planning process, and define their life cycle phases differently, the fundamental purpose of VVT is the same.

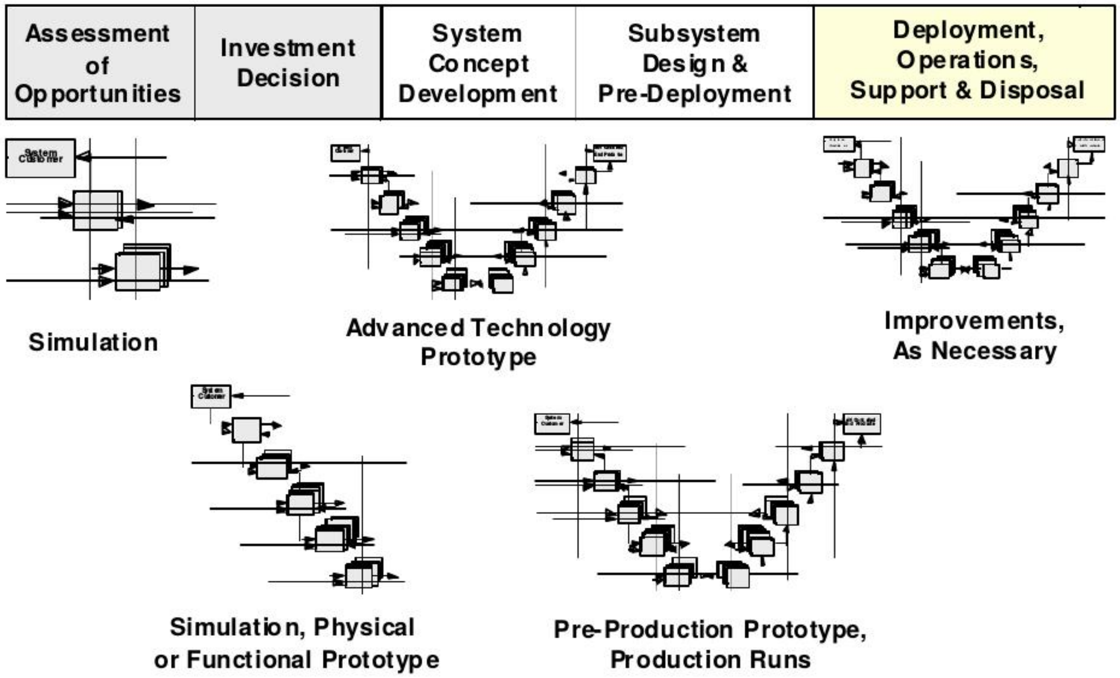


Figure 4: Progression of system verification and validation during design lifecycle [153].

The only consistency in this documentation is the lack of a developed methodology for developing a VVT plan. The current best practice is to use expert judgment in conjunction with the VRM to create a VVT plan while 'considering' cost, schedule, and performance risk impacts. Rework and retesting are not explicitly addressed in this documentation. Academic research has assumed the task of researching methodologies that will enable a more robust and improved VVT planning process in terms of cost, schedule, and performance. These studies are discussed in the next section.

2.2.1.1 Observations

A review of the literature search on VVT planning and evaluation techniques, a few observations about the current state-of-the art can be made. The successive cancellation of launch vehicle programs suggests that a more sustainable DDT&E plan is required, with a focus on reducing cost and schedule overruns during phases C and D during development. The primary government launch vehicle developers, i.e. NASA, DoD, INCOSE, and ESA, each have their own individual set of guidelines and definitions that address verification and validation, and VVT planning. While there are similarities and trends, no industry standard VVT development procedure is currently in use. Uncertainty and risk are recognized as important factors by each individual organization, but are not consistently evaluated during VVT planning process. Each organization addresses uncertainty with different methods and to different degrees during the development phases. Rework cycles and their impact on the program are not explicitly addressed in these documents. The main observation to be made from the review of industry standard approaches, is the lack of continuity of the VVT planning and evaluation techniques.

- Launch vehicles require the development of a more sustainable DDT&E plan
- No industry standard VVT development procedure is currently in use
- While uncertainty and risk are recognized as important factors, they are not consistently evaluated during the VVT planning process
- Rework cycles are not explicitly addressed

2.3 Research Objective

The primary motivation for this research is the gap identified by these observations. It has been identified that current VVT planning and evaluation techniques do not adequately account for rework cycles, and the resulting cost and schedule overruns often lead to program cancellation. Therefore, it is important to evaluate the impact unplanned rework can have on VVT activities, and determine how VVT planning can mitigate these effects. These observations lead to the following research objective:

Research Objective

Reduce cost and schedule overruns by modeling the effects of unplanned rework on the verification, validation, and testing of launch vehicle systems, and determining how VVT strategies can mitigate those effects.

The research objective can be achieved through the formulation and implementation of a structured process, or methodology that meets requirements derived from the identified weakness of industry standard VVT planning and evaluation. First, the qualitative nature, and lack of structure, in evaluating VVT activities does not provide a complete assessment. This leads to the first requirement, which states that a quantitative means for comparing alternative VVT strategies is desired. The second requirement stems from the gap in current approaches to consider the impact of rework cycles. The cost and schedule overruns that result from the additional design and testing activities have been recognized as significant hurdles for launch vehicle programs, but are still excluded from the VVT planning. This leads to the second requirement, that a quantitative estimate for the explicit impact of rework cycles on cost and schedule be produced. The final requirements is derived from the complex nature of launch vehicles. Individual program requirements and testing activities can often become bogged down in details due to the complexity of the system. Therefore, this method must scalable and flexible enough to enable its use at the subsystem and system levels.

Derived Requirements:

1. The method shall produce quantitative means for comparing alternative VVT strategies.

2. The method shall produce quantitative estimates for the impact of rework cycles on cost and schedule during VVT.

3. The method shall be scalable and flexible enough to enable use for large complex

systems.

The methodology that is formulated to meet the research objective and derived requirements provides the foundation for a risk-informed VVT evaluation framework that will aide decision makers in selecting the best set of VVT activities according to specific program goals. This decision support tool for VVT planning is therefore developed to follow a generic set of decision-making steps to enable its seamless integration in the overall decision-making process. For this research, th Georgia Institute of Technology Integrated Product and Process Development (IPPD) approach will be used [129]. A graphical overview of the IPPD methodology is presented in Figure 5. The center column of this umbrella chart represents a generic top-down decision-support process that can be applied to any type of problem.

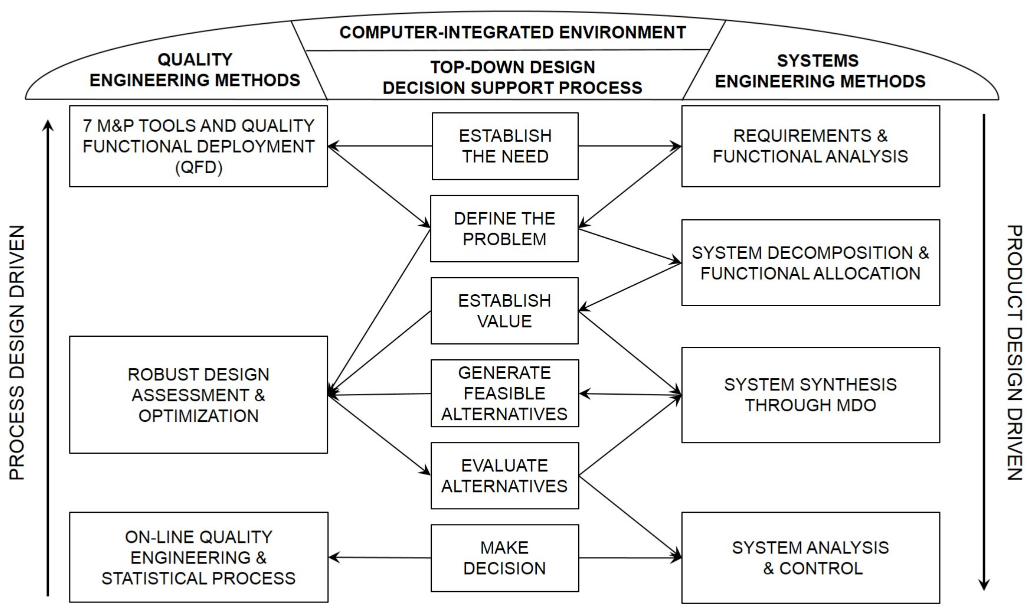


Figure 5: Georgia Tech integrated product and process development approach [129].

There are six steps in the IPPD methodology: establish need, define the problem, establish value, generate feasible alternatives, evaluate alternatives, and make decision. Chapter 1 established the need for this research by identifying the gaps in current methods to reduce cost and schedule overruns caused by unplanned rework.

In defining the problem, the primary objective and requirements for this research were derived through observations and identified weaknesses in the industry standard VVT planning and evaluation techniques discussed in Sections 2.1 and 2.2. The first two steps of this methodology are necessary to understand the expectations of the RIVVTS methodology and how it can be used to mitigate the effects of unplanned rework. To establish value, performance measures or metrics of interest need to be selected. A standard set of metrics allows for a traceable and fair comparison between alternative VVT strategies. The next two steps are two generate feasible alternative VVT strategies and evaluate the alternatives using the previously defined value criteria. The final step is to make a decision, or a risk-informed alternative selection based on the previous evaluation. The first two steps of the IPPD approach were completed through the development of the research objective and derived requirements. Establishing value for this problem leads to the first research question for the RIVVTS method, which is stated below.

> **Research Question: 1**
> What metrics should be used to evaluate the impact of rework on a VVT plan?

The metrics that are considered to some degree in industry VVT planning are cost, schedule, and risk, but there is no consensus on how these metrics should be evaluated or how they are impacted by rework cycles. A literature review of the VVT planning and evaluation techniques is presented in the next section to determine if there are other methods that have been developed to account for rework, and further assess which metrics are needed to evaluate a VVT strategy.

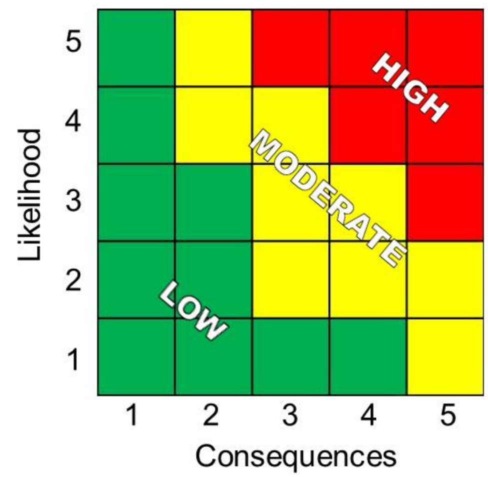
2.4 Literature Review of VVT Planning Techniques

The research efforts on improving VVT planning can be divided into qualitative or quantitative techniques. The qualitative methods focus more on standardizing the process by providing a common format to evaluate VVT activities. The quantitative techniques, alternatively, use various methods to evaluate the actual cost, schedule, and performance value of an individual activity. The following section contains a discussion on the existing methods to help answer research question 1.

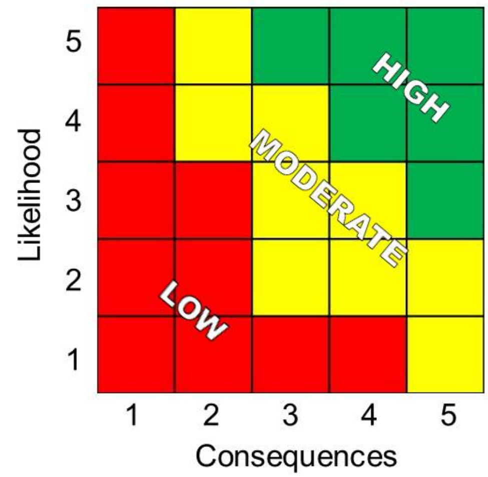
2.4.1 Qualitative VVT Planning Techniques

Meussig and Laack present a formal process of what is currently the industry standard to establish the value of verification, validation, and accreditation (VV&A) of modeling and simulation (M&S) environments [98]. The first step is to systematically identify the risk scenarios and quantitatively determine the probability of occurrence and the severity of impact each scenario will have on the system. Meussig uses an established impact and probability discretization described in MIL-STD-882C [107]. The severity of the impact is divided into four categories catastrophic, critical, marginal, and negligible. The probability of occurrence is similarly divided into five categories frequent, probable, occasional, remote, and improbable. When combined, these categories create a Risk Matrix (RM), which is a commonly used risk assessment tool [107, 131] shown in Figure 6a. This method differs by assigning similar categories to the benefits of VV&A activities, shown in Figure 6b. Using the risk and benefit matrices, the best VV&A activities can be selected by SMEs from a well-defined list of all the available VV&A activities.

Haimes, Kaplan, and Lambert suggest a more structured decision making process for filtering and ranking the identified risk scenarios [61]. A hierarchical holographic modeling approach is used to subdivide the system into more manageable sections for risk identification. Each subtopic then becomes a category of risk scenarios. After



(a) Risk Matrix.



(b) Benefit Matrix.

Figure 6: MIL-STD 882C defined risk matrix and associated benefit matrix [98, 107]

an exhaustive list of risk scenarios is created, Haimes uses multiple steps to rank and filter the scenarios to a more manageable number. The first phase of filtering is based on the expert opinion of the risk manager. Then the remaining risks are categorized according to MIL-STD-882 and arranged in a RM, where the low-priority risks are filtered out. From here the scenarios are filtered based on military system priorities and risk management is introduced to determine the most cost-effective ways to counter the risk scenarios that constitute the most risk to the system.

The last qualitative technique discussed in this section is a methodical approach to the tracking and documenting of verification activities [99]. To improve the verification process of space systems, a modular management process was recommended to be distributed to any and all teams working on the system, from contractors to the system's engineering team. This standardization of the process is meant to ensure that all requirements are sufficiently verified, and nothing is overlooked or undervalued. While not necessarily a study intended to find the optimal VVT strategy, it does elude to the need for a more formal and structured VVT program planning approach.

2.4.2 Quantitative VVT Planning Techniques

The next few methodologies quantitatively determine the value of different VVT strategies. One of these methods was an investigation into the optimal mix of parallel and sequential testing [85]. Uncertainty analysis is used as a measure of value, while cost and schedule are minimized. The iterative nature of the design process is represented by the use of sequential testing, where each test cycle can account for any learning achieved in the previous cycle. Parallel test cycles, on the other hand, do not account for any learning and are conducted as planned. If two cycles are unrelated, then parallel testing is preferred. Each test has a fixed cost and a target uncertainty reduction value. Full tests yield full uncertainty reduction, while any partial tests (e.g. scaled prototypes or wind tunnel testing) leave residual uncertainty relative to the level of testing performed. Loch et al. found that the financial cost and cost of time available to perform tests had a major impact on the testing strategy, where expensive tests are best used sequentially and slower tests are more optimal in parallel. The use of parallel testing does not take full advantage of the possible learning between tests, making it less optimal when using partial tests, which are already limiting the maximum uncertainty reduction available. Rework cycles are not addressed as a form of uncertainty here, but it does highlight the effect that iteration can have on the VVT strategy.

Thomke and Bell addressed the most effective way to incorporate high fidelity tests into the product development process [144]. Test costs were modeled as a function of fidelity, and the cost of rework to correct problems discovered during testing was modeled as a function of time. The authors developed a metric referred to as the Economic Test Frequency, which is a function of the test cost, number of cumulated faults discovered, and rework cost, to determine the optimal test frequency and occurrence for a development program. It was determined that the optimal strategy was highly dependent on the correlation of sequential tests, where mostly uncorrelated test portfolios benefited from more tests at mixed fidelities, and correlated test portfolios only needed a small number of higher fidelity tests. Because only sequential testing was considered, actual test durations were not a factor in determining testing strategies. While the trends discovered during this study provide useful knowledge for moving forward, the general application of this method does not lend itself to more complex systems. Launch vehicles, for instance, will utilize both sequential and parallel testing, and rework costs can vary from minor to significant based on more than just the time it is conducted.

Engel and Barad used a probabilistic approach to determine the cost of risks during VVT [51]. The canonical VVT baseline test strategy is considered an ideal test scenario with no resource constraints. Varying test strategies with different levels of partially performed tests are then compared to the canonical example. Using only sequential testing, any partially performed test is considered increased risk, and a Monte Carlo simulation is used for a stochastic risk assessment. The value is determined by the amount of risk reduction achieved. This method is based on expert opinion of expected cost and duration for each activity, and does not include rework probabilities.

The concept of risk reduction as a measure of value has also been applied to the overall product development process [25, 24]. Browning et al. used a risk value method to track reduction in product performance uncertainty [26]. Multiple technical performance measures are tracked and treated as random variables to measure their uncertainty. In addition to uncertainty, the impact of failure to meet the target performance goals is modeled as a quadratic impact function. Overall product performance risk is measured as a weighted sum of the technical performance measure impact failures. In another study, Browning further expands his ideas on adding value to the product development process [24]. In a risk reduction exercise, each product development activity is assigned a value based on the measure of risk reduction achieved. These values are used to create a predicted value trajectory for the planned activities. Browning suggests that eliminating activities where the value trajectory is flat, indicating significant expenditures with limited risk reduction, is a better approach to reducing costs during product development. Although this strategy is not directly applied to VVT processes, the idea to base VVT strategies on performance improvements could yield insightful results.

The SysTest project, sponsored by INCOSE, developed a generic VVT methodology with regard to its impacts on cost, schedule, and performance risk. A series of papers were published to review the initial outcomes of the project [51, 68, 83]. Hoppe et al., reviewed the results of applying the SysTest methodology across five design phases for six industrial projects [68]. A questionnaire given to the six projects evaluated the effectiveness of SysTest based on test and rework costs. Given the small sample size and inconsistent results, it was difficult to conclude that there was a substantial improvement. Although rework was a measure of success for this methodology, rework probabilities were not considered during the VVT strategy development. There was also no measure of product quality increase shown in the six sample projects. The cost avoidance strategy used in SysTest and other methodologies does not consider the value added by different testing activities [51, 83, 85, 144]. One important distinction between SysTest and the other methodologies described previously is in the output. While cost and/or schedule are factors in all of the studies, many only evaluate trends in the cost and schedule impacts based on VVT strategies. SysTest produces a quantified cost and schedule estimates.

2.4.3 Observations

- Few studies exist that consider both cost and benefit risk for VVT strategies

- A consistent method for estimating rework probabilities does not exist
- No studies quantitatively address the impact of rework on cost and schedule

The academic literature provides a variety of ways to evaluate VVT or testing plans. The approaches discussed in Section 2.4, qualitative and quantitative, can be categorized as either cost or cost vs. benefit. Cost categories found in the literature include the actual cost of testing, the cost of rework cycles, and the time spent testing. These techniques only address half of the VVT definition given in Section 2.2.1, regarding the program's resource constraints. The other half of the definition is to gain knowledge and eliminate defects from the system. The cost vs. benefit approaches account for the full VVT definition by including some measure of quality improvement to the system. The benefit categories for VVT activities are performance demonstration, reducing rework cycles, and risk reduction. Only a few methods were identified that fall into this category, and even fewer included the probability of rework. While there are suggested methods to generate rework probabilities, none of these studies quantitatively addressed the impact of rework on cost and schedule.

2.4.4 Conjecture for Research Question 1

Research question 1 asks the fundamental question for this research objective. In trying to select a launch vehicle VVT strategy, the first step is determining where the value lies in the VVT process and how others have modeled it in the past. A review of the industry standard practices in VVT process planning found a lack of a structured methodology, instead depending on expert opinion to individually tailor the VVT plan. Documentation from the two main U.S. launch vehicle developers and an international systems engineering organization all emphasized the need to consider the cost, schedule and risk impacts when planning is being conducted. These three parameters are then recognized as significant in the process, and need to be considered when assessing VVT value.

The academic literature search similarly focused on those parameters to different degrees, and also suggested that performance value should be included since cost and schedule should not be the only drivers for VVT strategy selection. The studies that address process planning fall into a cost vs. benefit proposition. What is considered a cost to the system and what is considered a benefit is what varies primarily. The methods are divided into two categories, qualitative and quantitative, to determine a 'best' process planning strategy.

The qualitative methods can be dismissed because they do not provide enough information to differentiate between distinct VVT strategies. However, the cost and benefit considerations of these approaches are still of interest. These methods focus on risk identification and SME input for VVT activity selection. The goal is to identify the biggest risks to the system by estimating the probability of risk occurrence and the severity of its impact should it occur. A RM is one of the tools used to assess the risks and down select from an exhaustive list of risks to focus on the high priority ones. The value, then, is in risk reduction through VVT activities. Cost, schedule, and rework are not explicitly addressed, and are only considered qualitatively in terms of risk occurrence impact.

The quantitative methods are more diverse. Again risk or uncertainty reduction is the most common goal in process planning methodologies. Research into sequential and parallel testing determined the value of a test based on the number of design problems uncovered as a function of test fidelity. The later in the testing process defects were uncovered, the most costly the rework effort, but no information was given on how to determine the probabilities of defects being uncovered or on the impacts of rework efforts on activity duration. Correlation between tests was shown to be an important factor in overall process duration when both parallel and sequential testing was included. Browning's work was the first to offer an estimation of actual cost and schedule duration when considering design iterations during product development processes. The goal of the product development process planning was to reduce the risk in technical performance using a risk value method. Rework was considered, but only the probability of one task causing rework in another task. The probability of repeating a task, referred to as internal rework, was implicitly included in the cost/schedule distribution for that task. For design activities internal rework implies that a certain portion of the activity is repeated, but testing activities require a repeat of the full activity. Explicitly addressing internal rework would provide a more accurate model for VVT activities. SysTest also produced cost and schedule predictions, but did not consider rework as an uncertainty input.

There have been a wide variety of methods introduced in evaluating VVT strategies. Many have a purely cost driven motivation, while others use an uncertainty or risk reduction technique. Only two methods produce an actual cost and schedule prediction, but neither fully address rework cycles and how to determine their probabilities. From this review, it can be seen that the main VVT parameters are a quality improvement measure (i.e. reduction in rework cycles or improvements in performance), cost, schedule, and risk. Although few of the methods address all four of these parameters, they collectively summarize the value of a VVT strategy. This review of current methods leads to the development of a conjecture to answer research question 1. The following sections and research questions will investigate these parameters further and determine what current methods exist for evaluating them.

Conjecture: 1
If quality, cost, schedule, and risk are used as metrics to evaluate the impact of rework during VVT, it will provide the most complete assessment of VVT activities, and will enable a quantitative comparison of alternative VVT strategies.

The selection of quality, cost, schedule, and risk creates a foundation for the overall methodology. These metrics can be defined or evaluated in different ways depending on the system, and the current state of development. For example, the most applicable cost estimation technique is heavily dependent on the development phase and the amount of information available about the system at that time. A measure of quality, on the other hand, is dependent on the type of system being developed. The following sections will review existing methods for evaluating each of these metrics, and determine their applicability to launch vehicle systems during Phase C and D VVT.

2.5 Quality

The cost vs. benefit methods discussed in Section 2.4 each use a different measure of quality for the evaluation of VVT activities. Reduction in rework cycles or reduction in uncertainty can be applied broadly to a variety of systems, but certain performance measures are more applicable than others for launch vehicle systems. This broad definition of quality in the literature requires further research to determine the most appropriate definition for the RIVVTS methodology, and is addressed by the following research question:

Research Question: 2
What is the most appropriate measure of quality for assessing the impact of rework on launch vehicle VVT strategies?

The previous VVT strategies touched on two different approaches to measuring the quality of a system defect elimination and performance. SysTest used the number of rework cycles as a measure of success for their project, suggesting that the implementation of the SysTest model improved the quality of the product by reducing the unnecessary rework. Although this was not shown explicitly in the results, the use of rework cycles that were required to eliminate defects were identified as a representation of quality in their VVT strategy [68]. Thomke and Bell also used number of defects as a measure of quality in their investigation of the optimum testing strategy. Their use of fidelity to determine the number of uncovered defects allowed them to directly assess the impact of individual activities [144].

Browning's product development process model assigns a value of overall performance to each activity by determining whether that activity improves or worsens the expected performance value and whether that activity reduces or increases the uncertainty of the expected performance value [26]. These two measures are combined similar to the risk matrix defined in Section 2.4 and is shown graphically in Figure 7. The combined effect on performance and uncertainty measures is used to rank the impact an activity will have on performance, which is then tracked throughout the product development model. An advantage of this method is that it could theoretically be used for any performance measure that is most relevant to the system. Another advantage is that it is designed to explicitly show the effects of a particular activity on performance, enabling a better comparison between alternative development process selections. A disadvantage of this technique is the qualitative nature of the assessment. A general indication of improvement or decline does not provide a very detailed level of analysis.

Bjorkman uses a model based systems engineering approach to improve the value of test and evaluation through uncertainty reduction by tracking relevant technical performance measures (TPM) [19]. This approach is similar to Browning's in that it supports the modeling of individual activity effects on system performance throughout the process. While a TPM trajectory can be represented as a smooth curve, performance improvements are more realistically modeled as step functions, where each VVT activity provides a shift in the curve. In this framework, the added value is the amount of uncertainty reduction, as opposed to performance improvements, but

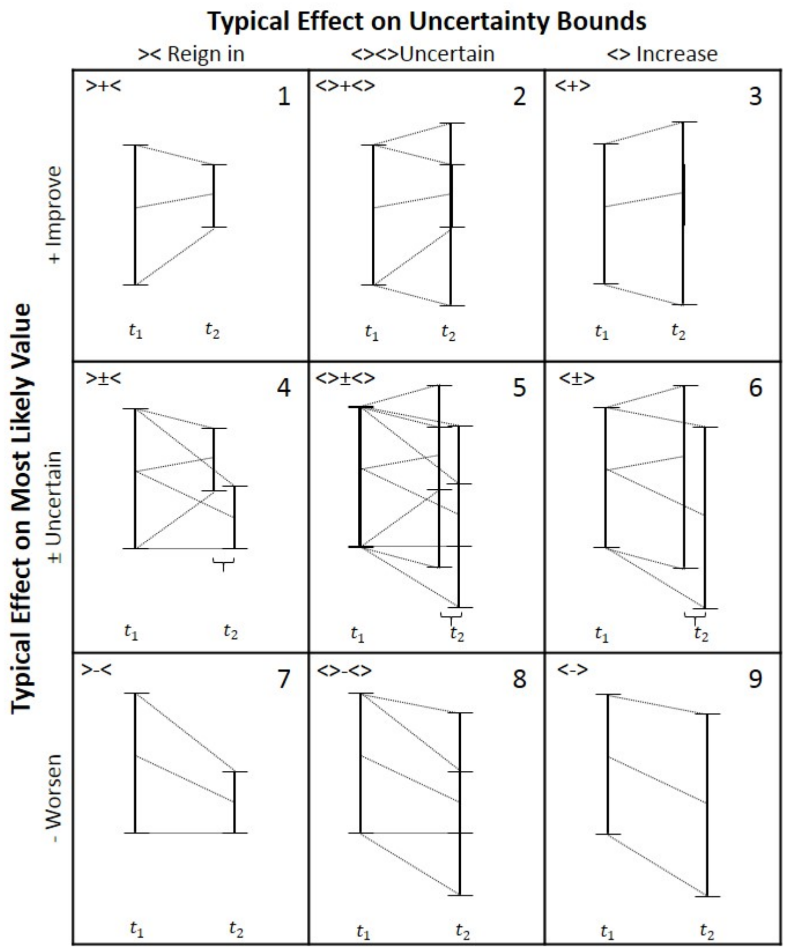


Figure 7: Qualitative activity-based performance effects [26].

the significance of relating the value to the activities is determined to be beneficial.

TPMs are identified as an important part of the systems engineering process by NASA and INCOSE [153, 131]. They are used to evaluate progress during the design and development process and provide a way for project managers to track the success of a project without relying solely on cost and schedule [153]. The selection of relevant TPMs is determined by the particular mission as they relate to key mission success parameters. The following section will give a brief overview of some of the priorities that NASA has identified for the SLS program.

2.5.1 NASA's Launch Vehicle Priorities

As seen in section 1, programs have been canceled due to combinations of cost, schedule, and performance failures. Referring back to Figure 3, it was shown that the average cost growth, even above the programmatic reserves, was 57% and the average schedule slip was 38%. This growth is typically not seen until after the Critical Design Review (CDR). Bitten suggests that this lag in programmatic cost and schedule realization could be due to overly optimistic estimates produced in the early design stages [56]. Other studies of historical NASA missions have been conducted by the United States Government Accountability Office (GAO) and the Congressional Budget Office (CBO) [59]. Programmatic failures noted by these organizations include inadequate definition of technical and management aspects of a program, funding instability, lack of emphasis on technological readiness, program redesign, and budget constraints. The effects of these programmatic failures can have a devastating effect on the life of a program. To overcome these common pitfalls during development the SLS program is prioritizing affordability and scheduling [100]. Reliability is also becoming increasingly important as significant reliability constraints are being implemented. The Space Shuttle had a demonstrated reliability of over 1 in 100 flights for loss of crew, and the SLS is now requiring 1 in 1000 flights [55, 29]. Typically U.S launch vehicles only see a 85-90% reliability over their lifetime [30, 79].

A full magnitude improvement over historical launch vehicle reliabilities is a considerable task. Other performance measures mentioned in the literature are more suitable for other systems or during earlier design phases. For example, detectability is more suitable for a military system, and thrust levels are verified earlier in the design process. All launch vehicle subsystems are subject to reliability requirements during both assembly and testing. Based on this information, reliability can be considered a system-level metric that is a key performance parameter for launch vehicles. A conjecture can be made here to answer research question 2, which leads to a follow-up research question.

> **Conjecture: 2**
> If reliability is used as a quality metric for launch vehicle systems, it will provide a quantitative representation and accurate measure of quality for VVT activities.

> **Research Question: 2a**
> What is the most appropriate method to track and assess reliability during VVT?

The following section provides a discussion of existing reliability analysis methods, and addresses their applicability for this methodology.

2.5.2 Reliability Techniques

Reliability analysis provides tools for assessing the probability that components, parts, or systems will perform as expected in a given environment and for a given time without failure. This is particularly important for crewed launch vehicles, where the most critical failure level is Loss of Crew. The combination of high complexity and high reliability requirements for launch vehicle development programs, is typically managed by dedicated reliability teams whose focus is to identify and eliminate as many defects as possible. The most common techniques for reliability assessment are discussed in this section. A thorough review of these and other methods is available in Dodson's *Reliability Engineering Handbook* [44].

2.5.2.1 Fault Tree Analysis

Fault Tree Analysis (FTA) is a system level reliability assessment tool. It is used to determine top-level failure events, i.e. loss of mission or loss of crew, and then identify all of the smaller contributing events in a trickle-down fashion. Figure 8 provides a notional example of the visual representation of a FTA diagram. The numbered circles represent the lower-level events that contribute to higher-level failures, connected by logical 'gates' [149]. The gate with a rounded top, like the one joining events 3, 4, and 5, represents an "AND" gate, where all three contributing events must occur to lead to the higher-level failure. The pointed gate, like the one joining events 1 and 2, is an "OR" gate, where in the higher-level failure will occur as a result of either lower-level event. This is essentially a root-cause analysis of the top-level failure that the FTA is designed around. It can provide qualitative evaluation of functional relationships and identify weaknesses in the design. It can also provide quantitative results in the form of probability of occurrence for the top-level failures [81]. Similar to the graphical activity network diagrams, FTA diagrams can become intractable for large complex systems.

2.5.2.2 Reliability Block Diagram

Another graphical reliability method is the Reliability Block Diagram (RBD) [16]. Where FTA is an event-oriented layout, RBD is a physical-oriented layout of the system. Figure 9 shows a simple RBD example. The blocks represent physical system components that are strung together according to their physical interaction. In this

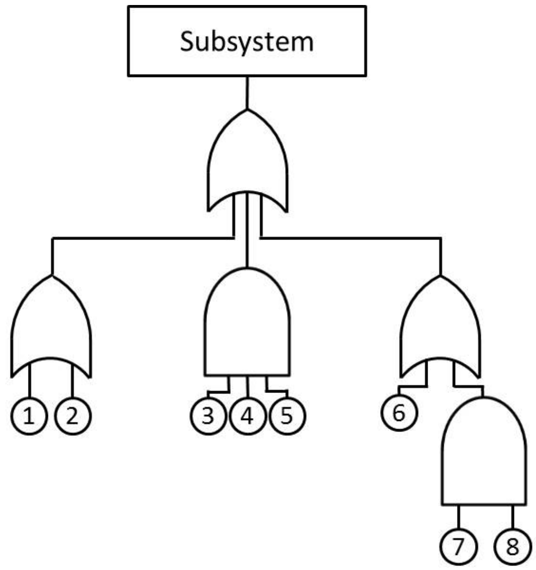


Figure 8: Notional FTA diagram.

example, components A and B work in parallel where if one component fails the system can continue through the other component in parallel. The probability of a parallel system is calculated as follows, where R_i is the reliability of an individual component:

$$R = 1 - \prod_{i=1}^{n} R_i \tag{1}$$

The component group A/B, and components C and D work together in series. If a component in series fails, the process is blocked and cannot proceed. The reliability of a system in series is calculated as follows:

$$R = \prod_{i=1}^{n} R_i \tag{2}$$

While FTA computes the probability of failure, RBD computes the probability of successfully completing the process. However, RBD presents the same weakness as other graphical representations, i.e. it can be difficult to create and follow when dealing with complex systems.

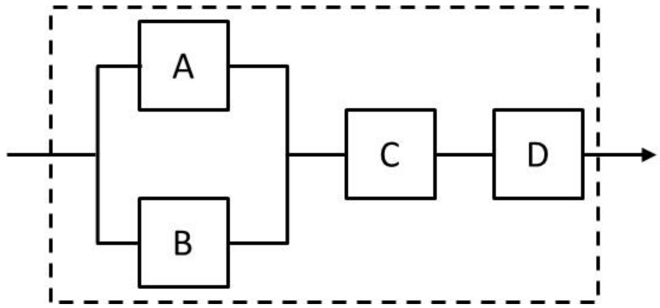


Figure 9: Notional RBD diagram.

2.5.2.3 Failure Mode and Effect Analysis

Failure Mode and Effects Analysis (FMEA), or Failure Mode, Effects and Criticality Analysis (FMECA), is a qualitative reliability assessment technique. The purpose of FMEA is to identify all of the potential failure modes in order to improve the reliability and safety of the system. This technique is common practice during launch vehicle development, and is referred to in both NASA and DoD reliability requirements documentation [3, 105]. FMEA begins with the identification of any known or potential failure modes by a team of subject matter experts. Then the cause and effect of each failure mode is determined through criticality analysis. Three risk factors are isolated:

- Occurrence (O) the probability a failure mode will occur
- Severity (S) the impact a failure mode will have on the system if it occurs
- Detection (D) the probability that a failure mode will be detected during inspection or test

These three factors are used to prioritize the failure modes identified by calculating a Risk Priority Number (RPN) [44]:

$$RPN = O * S * D \tag{3}$$

The failure modes with higher RPN pose the most risk to the system and need to be reexamined. Corrective actions are then recommended for the failure modes identified. DoD MIL-STD-1692, Procedures for Performing a Failure Mode, Effects and Criticality Analysis, provides guidelines to systematically perform FMEA. A sample worksheet, recreated from [105], can be seen in Figure 10 . FMEA has historically

Failure Mode and Effects Analysis

System ______________ Date ______________
Indenture Level __________ Sheet ______ of ______
Reference Drawing ________ Compiled by __________
Mission ______________ Approved by __________

Identification Number	Item/Functional Identification	Function	Failure Modes and Causes	Failure Effects			Failure Detection Method	Compensating Provisions	Remarks
				Local Effects	Next Higher Level	End Effects			

Figure 10: DoD FMEA worksheet [105].

been an important step in preventing failures from occurring and increasing the reliability of the system [84]. However, there are two common criticisms of FMEA. The first is the significant amount of time that goes into performing FMEA. Complex systems can have a vast amount of components that require major effort to analyze. The other criticism is in the value of the RPN metric. The three risk factors that are used to calculate the RPN are considered equally, with no relative weightings. This means different combinations of O, S, and D can yield the same RPN, but have varying level of risk implications [32, 152].

2.5.2.4 Probabilistic Risk Assessment

Probability Risk Assessment (PRA) is a quantitative reliability analysis that uses other reliability techniques in its formulation [137]. The fundamental steps of PRA are listed below:

1. Identify a list of initiating failure events

2. Create an event sequencing diagram from the initiating events

3. Convert event sequencing diagram into an event tree

4. Calculate probability of initiating event

Each of these steps can be completed with another reliability technique. For example, identifying the initiating failure events can be done by leveraging FMEA if it has been completed. Probability distributions are assigned to the initiating events, typically exponential distributions, which can be used in a Monte Carlo Simluation to find the probability density function (PDF) of failure probabilities [137]. While NASA has traditionally preferred qualitative reliability assessments, like FMEA, PRA has become more common since the Challenger accident [114]. Despite the benefits of having a quantitative probabilistic reliability assessment, PRA also has drawbacks stemming from the limitations of the other reliability techniques utilized.

2.5.2.5 Parts Count Method

The Parts Count Method (PCM) is a reliability estimation technique used during early design when detailed information about the system is limited. The part count is defined as the number of physically separate parts [57, 116]. The functional relationships between parts, locations, and attributes are not relevant in this reliability calculation. The advantage of PCM is the rapid reliability prediction enabled by its simplicity. For this reason, it is useful for generating comparisons between different configurations of a system during preliminary design [158].

The failure rate of the system is calculated by multiplying the generic failure rate of a generic part by a quality factor, and then summing the failure rates for number of generic parts. The generic failure rates for specific components can be obtained from failure rate databases, such as the electronic equipment failure rate database contained in MIL-HDBK-217F [104]. The quality factor can be applied when quality level data exists, and defaultd to 1 when it does not [104]. Equation 4 below gives the generic form of the PCM equation for the total system failure rate [104].

$$\lambda_{System} = \sum_{t=1}^{n} N_i(\lambda_g \pi_Q)_i \tag{4}$$

where λ_{System} is the overall system failure rate, n is the number of different generic parts, N_i is the quantity of the i^{th} generic part, λ_{g_i} is the failure rate for the i^{th} generic part, and π_{Q_i} is the quality factor for the i^{th} generic part. The PCM approach could be used to estimate launch vehicle reliability for architecture comparisons, but the actual reliability estimates have been shown to be imprecise [116, 158].

These common reliability techniques have all been considered for this research. While FMEA provides a considerable amount of information on specific risks, its qualitative nature makes it unsuitable for use as a TPM. FTA, RBD, PRA, and PCM could be used probabilistically by determining appropriate probability distributions for the component failures, but they are not typically used to predict reliability over time.

2.5.2.6 Reliability Growth Models

Another quantitative reliability projection technique is reliability growth models [20]. The idea behind these models is that reliability is increased as defects are uncovered and corrected, also referred to as the test-analyze-and-fix (TAAF) process. The existing reliability growth models are classified as either continuous or discrete. Continuous models typically consider mean time between failure (MTBF) data to track reliability over time. In discrete reliability growth models, the data represents reliability in terms of number of trials, or a Bernoulli process, where the possible outcomes

Table 4: Practical reliability growth rate parameters [103]

α	Reliability Effort	Launch Vehicle	α
0-0.2	No priority given. Corrective action only taken for critical modes.		
0.2	Routine attention given to reliability improvement.	Atlas [12]	0.2
0.3-0.4	Priority. Analysis and corrective action for important failure modes.		
0.4-0.6	Program dedicated to failure elimination. It has top priority, and corrective action is given for all failures.		

are either success or failure [58]. The following paragraphs discuss continuous and discrete reliability growth models that have been successfully applied to launch vehicle systems or subsystems.

Duane's model is one of the most widely used continuous reliability growth models [46]. His formulation for reliability growth is shown below,

$$\lambda = KT^{\alpha} \tag{5}$$

where λ is the total number of failures per total test time, K is a proportionality constant, T is total test time, and α is the growth rate parameter. In practical use, Duane asserts that a log-log plot of test hours vs. test failures can graphically provide α and K. Practical growth rate parameter values were later suggested based on the reliability effort required to attain a desired growth rate [103], seen in Table 4.

While Duane's model is used to track reliability across test phases, the Army Material Systems Analysis Activity (AMSAA) model was developed by Crow to track reliability growth within a test phase [23]. Reliability growth is modeled as a non-homogeneous Poisson processes that results from design fixes being introduced into the system. The expected cumulative number of failures at time t is given by:

$$E[N(t)] = \theta(t) = \lambda t^{\beta} \tag{6}$$

where λ and β are shape parameters, t is the cumulative test time, and $N(t)$ is the cumulative number of failures [23]. A discrete version of the AMSAA model was

Table 5: Morse reliability growth model parameters [97]

Parameter	Definition
d_k	Initial number of defects of type k.
λ_k	Conditional probability a defect is triggered if it is present.
τ_k	Conditional probability a defect leads to loss of mission if it is present and triggered.
ν_k	Conditional probability a defect is observed if it caused a partial anomaly.
φ_k	Conditional probability a defect is reported if it is triggered and observable.
γ_k	Conditional probability a defect is eliminated if it is uncovered.
p_{min}	Minimum probability of failure of the system.

later developed by Crow, referred to as the AMSAA-Crow model [38]. The discrete model substitutes cumulative test time with trial number and is calculated using this equation:

$$R_k = 1 - \lambda(N_k^{\beta} - N_{k-1}^{\beta})/n_k \tag{7}$$

where k is the configuration number, n_k is the number of trials in configuration k, λ and β are shape parameters, and $N_0 = 0$. The derivation of this model is based on the assumption that the number of trials is fixed for each configuration, and the distribution of successes and failures is random. These assumptions imply that, for each configuration, the full set of planned tests are carried out, regardless of the number of failures that occur. The result of the test series is then used to determine design fixes at the end of that phase.

Another discrete approach, introduced by Morse, models launch vehicle reliability growth as defect elimination by directly identifying the drivers of reliability growth in new systems [97]. In this model, the failure probability is derived from a set of probabilities for each identified defect type. The calculation of system reliability follows the flow chart shown in Figure 11, with the characteristic parameters required to define the model provided in Table 5 .

Following the diagram flow in Figure 11, reliability is derived using the following

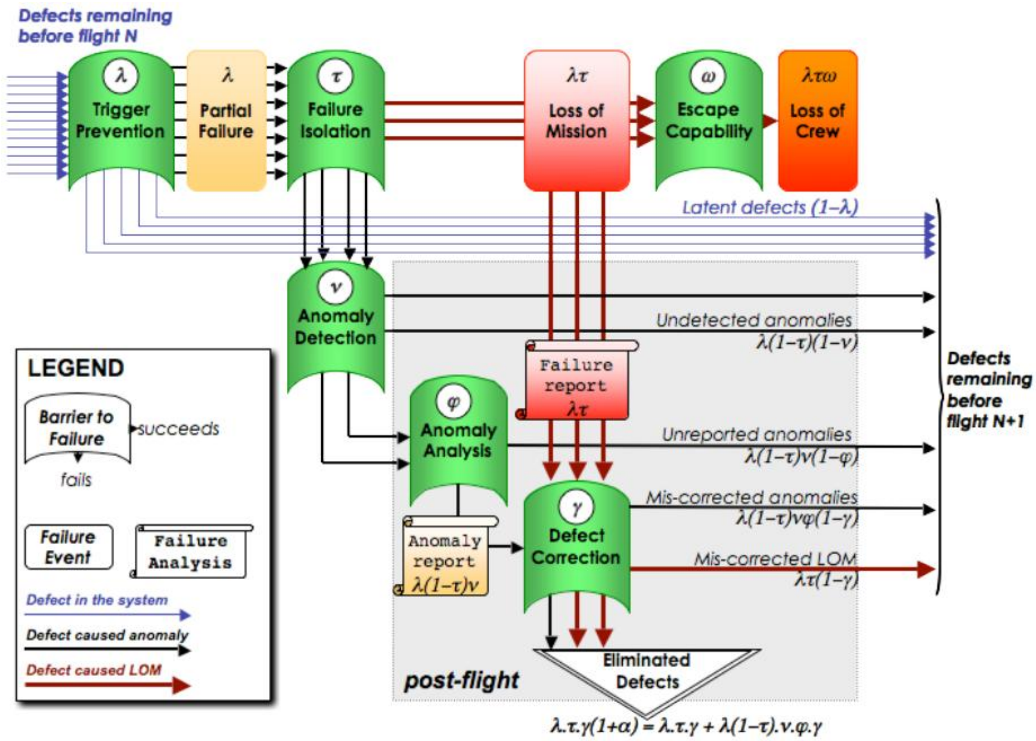


Figure 11: Morse reliability growth flow chart [97].

equation [97]:

$$R(N) = (1 - p_{min}) \prod_{k=1}^{D} [1 - \delta_k(N) p_k]^{d_k} \tag{8}$$

where p_{min} is the mature probability of failure of the system with D defect types, $\delta_k(N)$ is the probability that a defect of type k remains in the system at flight or test number N, p_k is conditional probability of system failure from a defect of type k if it is still present, and d_k is the initial number of defects of type k.

While Morse's reliability growth model is primarily a discrete model, projecting vehicle reliability versus number of flights, it also offers suggestions on how it could be applicable during testing phases. The same defect detection mechanisms are present during testing, implying that the same model equations will hold with small changes to the input parameters. Those changes are listed below:

1. Reduce a portion of risk due to the limited number of defects that can be exposed during ground tests, when fully integrated test flights are not viable. Risk is the contribution of a defect to the probability of loss of mission (LOM).

2. Improve the probability of defect detection and reporting due to extensive instrumentation available during testing:

$$\rho = \nu\varphi \tag{9}$$

   where ν is the conditional probability that a defect is observed after it occurs, φ is the conditional probability that it is reported after it is observed, and ρ is the conditional probability that a defect is eliminated after it is observed and reported.

3. Optionally, increase the probability of triggering some defect types, λ_k during certain test types.

Hall's discrete reliability growth model for one-shot systems has also been shown to accurately represent launch vehicle systems [62, 158]. The characteristic parameters to define the Hall model are listed in Table 6, with the five main assumptions used to derive the model being [62]:

1. A trial results in a dichotomous success/failure outcome, such that $N_{i,j}\sim$ Bernoulli(p_i) for each failure mode i and each trial j

2. Distribution of the number of failures in T trials for each failure mode is binomial, such that $N_i\sim$ Binomial(T,p_i) for each failure mode i

3. Initial failure mode probabilities of occurrence, $p_1,...,p_k$, constitute a realization of a simple random sample, $P_1,...,P_k$, such that $P_i\sim$ Beta(n,x) for each i=1,...,k.

4. Potential failure modes occur independently of one another and their occurrence is considered to constitute a failure

Table 6: Hall reliability growth model parameters [62]

Parameter	Definition
k	Initial number of failure modes in the system.
T	Number of trials during testing phase.
d_i	Fix effectiveness factor.
$p_i \sim Beta(n, x)$	Failure mode probabilities of occurrence, Beta distribution shape parameters.

5. There is at least one repeat failure mode

Each trial, or test, is considered an independent Bernoulli trial according to assumption 1, and the history of failure mode occurrences is tracked using an indicator function defined as:

$$I_i(t) = \begin{cases} 1 & \text{if failure mode } i \text{ is observed on or before trial } t \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Assuming the trials are statistically independent, the expected value of the indicator function at trial t is:

$$E[I_i(t)] = 1 - (1 - p_i)^t \tag{11}$$

Reliability on trial t is given by the following equation, given failure mode mitigation has occurred:

$$r(t|\overrightarrow{p}) = \prod_{i=1}^{k}(1 - [1 - I_i(t-1) * d_i] * p_i) \tag{12}$$

where d_i is the Fix Effectiveness Factor (FEF) for failure mode i, a measure of how effective the corrective action was in eliminating the failure mode once it has occurred. $FEF = 1$ implies the failure mode was completely eliminated from the system, and the resulting probability of failure occurrence is 0. $FEF = 0$ indicates that no corrective action was taken when the failure mode occurred.

Zwack's reliability growth model for conceptual launch vehicles utilizes Hall's mathematical model [158]. Hall's model is applied at the subsystem level, and a fault tree analysis is used to propagate the subsystem reliabilities to the system level for the entire launch vehicle. By generating lower level reliability growth curves, Zwack provides more insight into the reliability of the system than other models, and enables reliability to be considered in architecture trades at the conceptual design phase [158]. However, the additional reliability growth curves require more assumptions to be generated, i.e. Hall's FEF, $p(i)$, and the number of failure modes for each subsystem.

2.5.3 Hypothesis for Research Question 2a

The reliability growth models reviewed in Section 2.5.2.6 have been applied to either a full launch vehicle or a launch vehicle subsystem, primarily liquid rocket engines. The first two models — Duane and AMSAA-Crow — were continuous, and have been used to represent the Space Shuttle Main Engine reliability over its 110,000 sec hot-fire engine test history [142]. Both models use a growth parameter approach, which can be derived using data from a similar or surrogate system. AMSAA-Crow also requires an assumption about the number of test configurations and number of tests per configurations, which may not be known in advance. Although they show an accurate prediction of the SSME Mean Time Between Failure (MTBF), these models do not provide any insight on the effect of individual tests on the reliability of the system.

The other models discussed, Hall, Zwack, and Morse, were discrete reliability growth models that have been shown to accurately predict launch vehicle reliability. The primary difference in these models is the assumptions that they are built on. Morse requires more probability parameters to calculate the reliability of the system. These include the number of failure modes, probability of occurrence, detection, action, and correction. Hall's model only requires the number of failure modes, probability of occurrence, and the fix effectiveness factor. The number of failure modes and probability of occurrence are common to both models. Morse's probability of correction can be equated to Hall's fix effectiveness factor. An assumption could be made for the additional Morse parameters of action and correction by considering that this methodology is modeling activities during testing. It can be assumed that a significant amount of special test equipment and instrumentation is being used to detect any anomalies, and that action is always taken due to the increased reliability requirements of the system. The Morse model detection and action probabilities could be defaulted to 100% based these assumption, negating the additional parameters. However, the traceability of this methodology is negatively affected by adding these assumptions. The complexity of this model is also a drawback. Implementing this model across testing phases requires the evaluation of each input according to the specific activities. By requiring fewer inputs, Hall's model would more easily be adjusted to varying testing activities. While CONTRAST utilizes Hall's model, it requires that the input assumptions be generated for each subsystem that is being considered. The added complexity of Zwack's model is not necessary for the RIVVTS methodology because the subsystems are not changing. Based on this evaluation, Hall's reliability growth model was selected for assessing reliability during VVT.

Zwack, Hall, and Morse state that their respective models can be used during the testing phase, but are primarily used to model launch vehicle test flights where all failure modes are observable in a fully integrated system. Development testing for launch vehicles and launch vehicle subsystems does not always allow for fully integrated tests. Due to the extreme operating environment, lower fidelity testing is more common because it is less cost-prohibitive, and only a handful of flight tests are conducted. The fidelity level of development tests gets progressively higher as the system develops. This implies that each 'trial' or test cannot be treated equally. To enable the use of Hall's model during development testing, when the fidelity of VVT activities varies, it will have to be adapted to include a function of test fidelity, leading to a hypothesis for this research question.

Hypothesis: 2a

If Hall's reliability growth model is adapted to include defect elimination as a function of test fidelity, it will provide reliability projection with quantitative insight into individual VVT activities.

2.6 Schedule

Schedule slippage was identified as primary contributor to the cancellation of launch vehicle development programs in Section 1.1, and was determined to be a necessary metric for the evaluation of the rework impact on VVT strategies in Section 2.2. Rework can affect program schedule on multiple levels depending on the severity of the defect being corrected. Figure 12 illustrates the full potential impact of a single rework cycle on schedule slippage. Once a fault is detected, the cause of the fault is traced back to either a design or manufacturing defect. If it is a design defect, the rework cycle will start with redesign, and then remanufacturing, and finally retesting the design change for verification of fault correction. Between each of these activities, any number of delays could occur. For example, required parts could be unavailable or test facilities could be occupied. Because rework cycles are typically unplanned, the impact on total development schedule can be significantly impacted [64].

Due to the stochastic nature of failure modes and rework cycles, the technique used to model VVT schedule must be flexible and have the ability to incorporate the stochasticity of rework cycles. The overall objective of this research to develop a quantitative means to compare alternative VVT strategies, imposes the additional criterion that the method selected produce a quantitative schedule estimate. The following research question is posed to determine the most appropriate schedule estimating technique for this methodology.

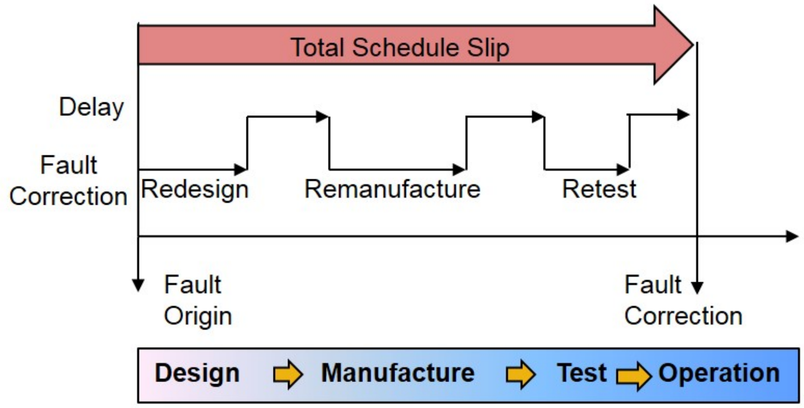


Figure 12: Total impact of a single rework cycle on schedule slippage.

Research Question: 3

How can the schedule of a VVT strategy be evaluated to include the impacts of rework cycles?

The following section will review current schedule requirements and development techniques, including the use of activity network diagrams to represent the sequential relationships of activities.

2.6.1 Schedule Requirements

NASAs Schedule Management Handbook details what is required to develop the Integrated Master Schedule (IMS) for a program. The IMS is considered the baseline schedule and is created using industry best practices [102]. A Logic Network is also created as a time-phased sequence of project tasks and milestones. The 'best practice' referred to here is called the Critical Path Method (CPM). The level of schedule detail and insight increases throughout the life cycle of a program. During the formulation phases, typical schedules will only include major milestones and general time phasing of high-level tasks. As the program flows into the implementation phases, and system details become formalized, more detailed tasks and milestones are required.

Developing a project schedule starts with determining which tasks are required.

This task definition is typically derived from a Work Breakdown Structure (WBS), where each input is broken down into measurable tasks. The level of detail required in the schedule depends on the current design phase and intended stakeholders. Program managers, for example, would need less schedule detail than a project manager [102]. These tasks or activities are then logically arranged into a network based on their relationships and constraints. Basic activity relationships are shown in Figure 13.

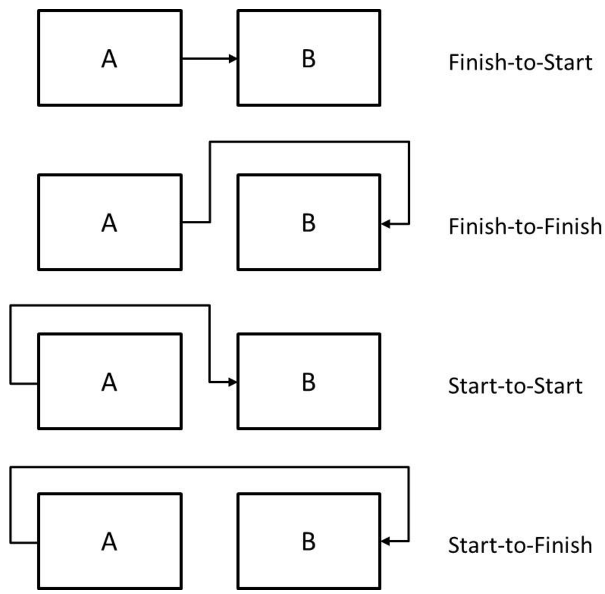


Figure 13: Types of activity interdependencies.

The next step is to estimate the duration of each activity. There are several ways to accomplish this. One way to estimate activity duration is to use historical data, either strictly using the historical data to compare activity durations of previously completed similar activities or with parametric analysis. Parametric analysis incorporates the duration of historical activity data and other related project data to estimate activity duration, for example system mass, power, cost, etc. Parametric analysis is a commonly used method to estimate high level tasks during conceptual design, i.e. development and production. Another method to estimate activity duration using historical data is the analogy method, where duration is directly associated to relevant historical data and adjusted for complexity or other system metrics. The last two methods to mention for activity duration estimation are team brainstorming and Subject Matter Experts (SME). With these methods, it is important to pull from a highly experienced work force. Team members that are familiar with the nature of the project can provide quality duration estimates. It is recommended that three durations be estimated for each activity, a pessimistic, optimistic, and most likely time. Having a range for the durations enables uncertainty calculation in the schedule. The last step is to determine the sequencing of the tasks and activities. Relationships between activities, i.e. finish-to-start, start-to-start, start-to-finish, etc., and activity constraints, i.e. as soon as possible, must start on date, must start before date, etc., are the kind of information required to determine activity sequencing.

Current industry standards rely heavily on subject matter expert opinion and do not incorporate any activity iterations. The following section will describe the most commonly used scheduling methods, all of which do not consider unplanned iterations. The remaining section will introduce methods that do consider iterations, but are less common in practice.

2.6.2 Non-Iterative Scheduling Methods

The most well-known scheduling method is the Critical Path Method (CPM), which has been used for project planning since the 1950s [47]. The main elements for CPM are the same as the basic elements discussed above, e.g. a list of required activities (WBS), duration for each activity, and activity interdependencies. Using those elements, CPM calculates the longest path to project completion using the relationships between the planned activities. It essentially determines which activities are critical, or are on the longest path, and which can 'float', or be delayed, without extending to total project completion time. An important distinction for CPM is that the activity durations are fixed and deterministic. Iterations are not modeled with CPM unless two of the same activities are called out in the WBS initially. While this method is widely used due to its direct and easily understood nature, these aspects limit its usefulness when uncertainty and risk calculations are necessary.

There are many risks that lead to uncertain activity durations, i.e. late deliveries, changes in project scope or requirements, unplanned rework, resource constraints, etc. The PERT method, Program Evaluation and Review Technique, was developed to deal with this imprecise data [88]. Also a widely used technique in industry, PERT requires three time estimates for each activity, a pessimistic, optimistic, and most likely. These estimates, usually given by SMEs, are used to determine the expected value and variance for each activity duration. PERT then calculates a probability distribution for the likely overall project duration based on the probability distributions of the activities. The system is visually represented by a series of nodes and arcs. A sample of a PERT activity network can be seen in Figure 14, where each arc represents an activity and each node represents an event.

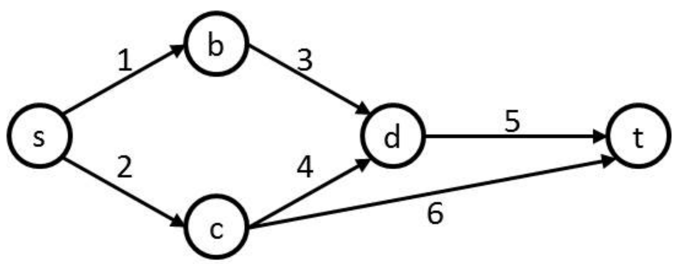


Figure 14: Notional PERT network.

There are a few recognized drawbacks to PERT in the literature. One significant criticism is in the ability to generate meaningful estimates for the activity durations. Obtaining estimates for each activity can be time consuming and it may be difficult to find experts that are familiar with the problem when dealing with novel concepts. Because the three time estimates are given by SMEs, they are purely subjective and sensitive to the judgment of the expert. A study by Swanson and Pazer (1971) was conducted to determine the sensitivity of the expected value and variance for the project duration to imprecise estimates and determined that the upper and lower bounds are ambiguous [141]. Another disadvantage of this method is the visual nature of the system representation. For complex systems it can quickly become overly complicated, making it difficult to compare alternative network options.

Another non-iterative technique is the Probabilistic Network Evaluation Technique (PNET) [12]. PNET evaluates overall project duration based on the number of failure modes in the network. A notional PNET chart is shown in Figure 15. Each independent path is represented as a different color, and each represents a possible failure mode in the network. 'Failure' in this context is anytime the project takes longer to complete than the initial estimate. Similarly to PERT, PNET starts with a probability distribution for the duration of each individual activity. Using these distributions, the probability distribution of each complete path duration is calculated. A correlation matrix is calculated based on the number of common activities between two paths and their respective standard deviations. This correlation matrix is used to reduce the number of paths by eliminating ones that are highly correlated with paths that have a longer duration. From this reduced set, the probability of the network duration being longer than the target duration, T, is found using the following equation:

$$P(t > T) = 1 - p(t_1 < T) * p(t_2 < T) * ... * p(t_n < T), \tag{13}$$

where $p(t_i < T)$ is the probability that the i^{th} path will have a duration less than T [42].

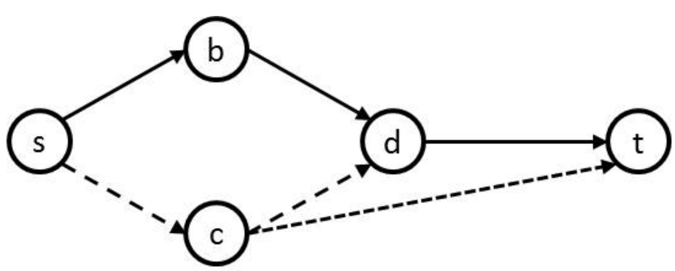


Figure 15: Notional PNET network.

2.6.3 Iterative Schedule Methods

There are a handful of scheduling methods that do include iterations. One of these methods is the Graphical Evaluation and Review Technique (GERT), a network analysis technique which was developed as an extension of PERT [119]. To overcome the PERT shortcomings, GERT was designed to allow for looping and probabilistic branching. These features make it more versatile than CPM or PERT. Figure 16 provides an example of a GERT network. Each arc is considered individually with its own probability of choosing that arc. The sum of probabilities from a single node is always equal to one. The visual system representation comes with the same disadvantage as PERT, in that it quickly becomes overly complicated for complex systems. Also, the order of each activities remains fixed, limiting its ability to compare alternative system process plans. Although it improves upon certain aspects of PERT, GERT has not been embraced by industry and is not often used [143].

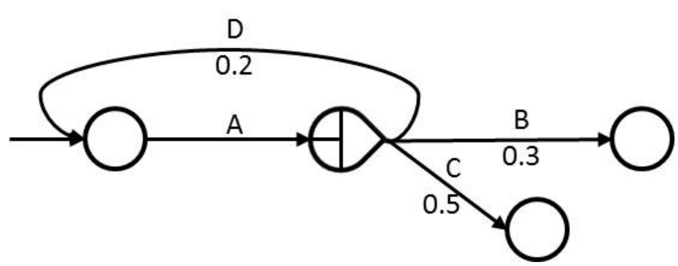


Figure 16: Notional GERT network.

Markov Chains are another method used to represent iteration in a system [11]. Markov Chains uses a state space representation that consists of various states and the transitions to/from those states. Figure 17 provides a simple example of a Markov Chain. The states $\{A, B\}$ represent activities in the process flow, and the arcs represent the transition between states. The numbers associated with the states are activity duration, and the fractions associated with the arcs are the transition probabilities. An advantage of this method is the ability to account for stochastic activity durations and branching. It is also an established model that is mathematically simple and well understood. However, like GERT, it can quickly become overly complicated for complex systems. Another disadvantage is that the activities must be predefined and cannot be altered once the model is created.

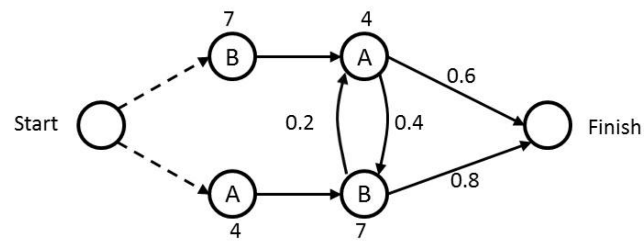


Figure 17: Notional Markov Chain.

The most commonly used tool to model iterations is the design structure matrix (DSM). The DSM is designed to represent the interactions between the matrix components, showing how tasks and information flow affects other tasks [139]. A basic DSM consists of a square matrix with activities down the left side and identically across the top, as shown notionally in Figure 12. The matrix entries represent a dependency between the tasks. This creates a manageable format for representing complex processes graphically. From this representation, the schedule can be converted into an activity network diagram, like GERT or PERT. Karniel and Reich published a comprehensive review of DSM-based planning which can be referred to for more detail [72].

A traditional DSM only establishes the existence of a relationship between tasks. The upper half of the matrix represents feed-forward dependency, and the lower half represents feed-back dependency (or vice versa depending on convention). The program DeMAID (Design Manager's Aide for Intelligent Decomposition) used this form of DSM to improve product development by rearranging the tasks to minimize feedback, a process known as partitioning [125]. This has been shown to reduce the sensitivity of overall process time to changes in individual activity durations [76], but

		1	2	3	4	5	6	7	8
Set Requirements	1	■	■						■
Concept Proposal	2	■	■						■
Propulsion	3	■	■	■					
Aero	4	■	■	■	■	■			
Weights	5	■				■			
Controls	6	■		■	■	■	■		
Performance	7	■		■	■	■		■	
Assess Concept	8	■	■	■	■	■	■	■	■

Figure 18: Notional DSM.

further evaluation has shown that this strategy does not always yield the minimum project duration [8, 25, 99]. DSM was improved by developing the numeric DSM (NDSM), where the matrix entries were replaced by numbers representing the level of relationship between activities [52]. Browning used the DSM off-diagonal entries as rework probabilities between the activities and utilized the on-diagonal entries as the activity duration [25]. His methodology was assessing the impact of architecture selection on the performance, cost, and schedule risk in product development, of which VVT is just a small piece. Internal rework was assumed to be included in the on-diagonal activity duration probability distributions, and only inter-activity rework was considered. Internal rework is more of a concern during VVT due to the probabilities of test failures causing rework [53]. Including these probabilities in the VVT activity durations limits the fidelity of the uncertainty measures. The utility of DSMs to model variations in activity relationships is an advantage for this study. Smith and Eppinger have shown it can be used to model only sequential activities, only parallel, or a combination of both [135]. A disadvantage of the DSM is its inability to represent stochastic activity durations and rework probabilities. Another disadvantage is its limited use on VVT activities, where most of the research has been done on overall product development processes.

2.6.4 Hypothesis for Research Question 3

The first set of schedule management techniques introduced in Section 2.6.2 is the non-iterative methods, which include CPM, PERT, and PNET. CPM and PERT are the most commonly schedule management tools used in industry today. CPM allows for the identification of 'critical activities in the schedule, and PERT extends that to include the calculation of an overall project duration probability distribution. PNET, while being less popular, also includes probabilistic by considering all the possible failure modes in the network. These methods are well established and understood, but can be dismissed because they do not allow for iterations and require a predetermined set of activities.

The graphical techniques introduced are all capable of stochastic assessment, and differ only on how they represent the system. GERT is a basic iterative model that is based on the PERT method, but has the ability to model iterations. Markov chains serve a similar purpose, but use a state space representation of the system. GERT is generally only used as an academic tool, and has not really gained traction in industry, while Markov chains is a well established and understood activity network model. These techniques all share the same weaknesses, however. Their graphical representation can be difficult to generate for complex systems and quickly becomes intractable. This limits their flexibility and traceability for use on launch vehicle systems.

The final schedule management technique that was introduced was the design structure matrix. DSMs are a matrix representation of the information that is graphically displayed in the other methods, making it much more concise and easily understood. The transition to numerical DSMs by Eppinger further increased its usefulness and led the way for other studies to adapt the DSM in different ways. Brownings use of DSM to represent rework probabilities and impacts in the product development process is particularly useful. The matrix representation makes DSM more flexible and traceable than the other techniques reviewed previously, leading to the development of hypothesis 3, stated below.

> **Hypothesis: 3**
> If a DSM is adapted to explicitly account for the probability of internal rework, it will provide a stochastic and quantitative model of rework impacts that is more accurate for VVT processes than if internal rework is implicitly included in the activity duration distribution.

2.7 Cost

Cost overrun due to rework cycles was identified as another primary contributor to the cancellation of launch vehicle development programs in Section 1.1, and determined to be a necessary metric for the evaluation of the rework impact on VVT strategies in Section 2.2. Like schedule, rework cost can vary due to when the fault is detected and the severity level of the fault. As discussed in Section 1.1, up to 75% of development cost can be spent on eliminating failure modes through unplanned rework. Figure 19 shows the percent of peak funding that is spent during the F-1 engine development as a function of development time. The amount spent on rework cycles increases through the first 6 years, where it reaches its peak. The last three years show a decrease in the amount of total spending attributed to rework, but the cost per rework cycle is not obvious. Reliability growth models show that failure modes occur more frequently during early testing, and then slow down as confidence in the design increases. It is necessary to determine the cost per rework cycles by itemizing the total cost for eliminating failure modes based on the rate that they occur.

The cost per rework cycle for the F-1 engine development program is illustrated in Figure 20. The first two years are excluded because the rework cycles that are being considered occurred during engine-level testing, which did not begin until 2 years

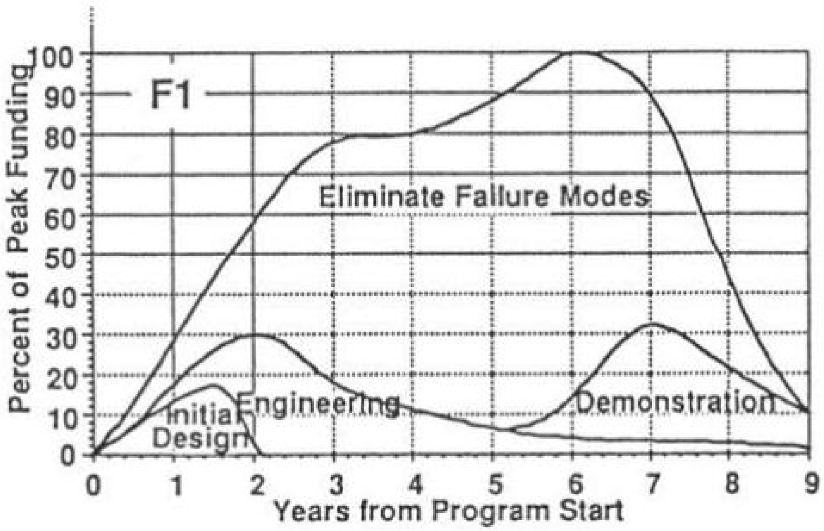


Figure 19: F-1 engine percent funding used to eliminate rework cycles during development.

into the program. Except for the one data point above 80%, the previous assumption can be confirmed. An exponential fit for this data shows that the cost per rework increases from 7% to as high as 42% near the end of the development cycle. This is to be expected because when the engine enters certification tests, any failures require that the engine start the certification cycle from the beginning.

While the data used in this example is from the F-1 engine, it illustrates a trend commonly found when dealing with complex systems. The general assumption is that the cost per rework cycle increases as the design progresses and nears production. The foundation for this assumption is based on the well-known design curves shown in Figure 21. This illustrates that the design freedom, or ability to make changes to a design, decreases rapidly from the start of the design process. The other curve illustrates that actual design knowledge increases slowly at first, then rapidly increases in the middle of the process, and gradually levels out towards the end. The early decrease in design freedom implies that any changes to the system, like those that would occur during rework cycles, are more and more difficult to implement as the

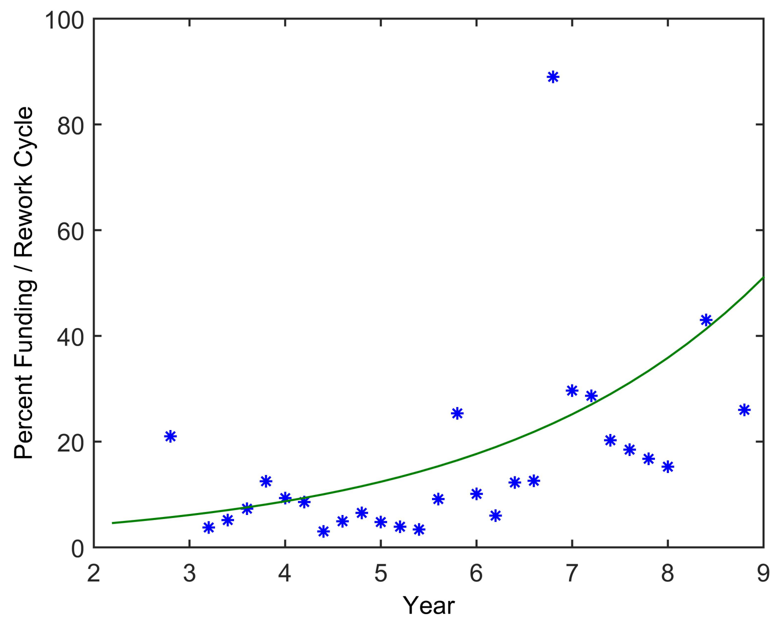


Figure 20: Percent of funding used per rework cycle during development.

system progresses in the design process.

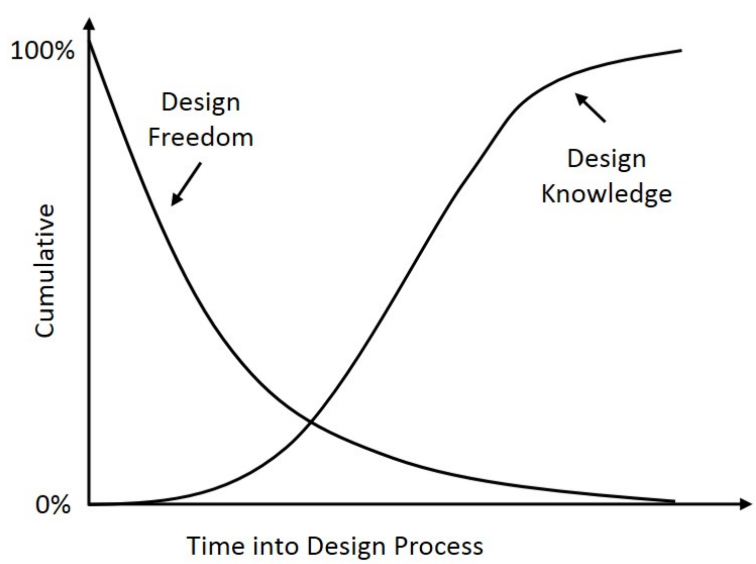


Figure 21: Design freedom and design knowledge during development.

When selecting a cost estimating approach for this methodology, it is important that the method be detailed enough to account for an individual rework cycle, but also flexible enough to allow for the stochastic nature of failure modes and rework cycles. Like the criteria for schedule estimation, the cost estimating approach must produce a quantitative result to support the overall research objective. The following research question is posed to determine the most appropriate cost estimating technique for this methodology.

> **Research Question: 4**
> How can the cost of a VVT strategy be quantified to include the impacts of rework cycles?

Section 2.4 described various approaches to determining the cost of a VVT strategy. Some only considered the direct cost of activities, while some modeled every VVT risk driver as a system cost. Section 2.7.1 reviews the general cost estimating methodologies currently being used to determine if existing methods can satisfy the criteria for this methodology. In addition to traditional methods, reliability-based cost estimating methods are reviewed in Section 23 due to the choice of reliability as a measure of quality.

2.7.1 Cost Estimating Methodologies

The three primary cost estimating methods for launch vehicles are parametric, analogy, and engineering build-up [101]. Figure 22 shows that these methods are used during different stages of the design process. Parametric and analogy methods are top-down approaches geared toward generating a gross estimate [147]. Conversely, engineering build-up is a bottom-up approach conducted at the lowest level of available detail. These main cost estimating methods are described in detail below.

Parametric cost estimation is applied predominantly during the early phases of the design process when little detail is known. Notably, Pre-Phase A and Phase A cost estimates are used by NASA to secure funding [101, 131]. The foundation of this

	Pre-Phase A	Phase A	Phase B	Phase C/D	Phase E
Parametric	●	●	◐	◐	○
Analogy	●	◐	◐	◐	○
Engineering Build Up	◐	◐	●	●	●
Legend: ● Primary	◐ Applicable	○ Not Applicable			

Figure 22: Standard cost estimating methods [101].

approach is the mathematical relationships, known as cost estimating relationships (CERs), developed to relate historical cost data to physical and performance parameters that are proven to be cost drivers for the system. Examples of these parameters include, weight, power, design life, and technical maturity, among others. The implicit assumption here is that the same cost drivers of past systems will continue to drive cost in the same manner for new systems. Parametric cost estimates can be versatile and quick once a sufficient amount of relevant data has been collected. The reliance on historical data gives the method defensibility by increasing objectivity and eliminating the need for expert opinion. This reliance is also a detriment due to the time and effort required to initially collect the data. Availability of such data can also be a challenge, especially when working with novel concepts. Because launch vehicles, and all space systems, are at the least proprietary and at the most classified, a sufficient amount of relevant data can be challenging or even impossible to gather. Another drawback of this method is that parametric relationships lose their predictive capability when applied to inputs outside of the data ranges used to create them. This can limit the usefulness of the cost model when a system is applying new technologies [73, 126]. This is the case for reusable launch vehicles (RLV), as the majority of LVs are expendable.

Analogy cost estimation is performed by identifying an existing system that is technically similar to the new system, and adjusting the cost data up or down to account for any differences in their complexity or requirements. This method is also used during early design phases because it does not require full program details. It relies on historical data to provide a defensible and traceable estimate. If a strong analogy can be found, then adjustments on the technical parameters related to cost will be small, further increasing the accuracy of the cost model. However, this method is contingent on there being a sufficient amount of technical and programmatic detail available for the analogous system. Lack of relevant historical data is one reason why it can be difficult to find an appropriate analogy. Another drawback of this method is that the cost estimate relies on a single data point for predicting the cost of a new system. This limits the accuracy of the cost model if the cited analogous system is unsuitable. In this sense, the method is largely subjective. It relies on expert judgment to determine not only the analogy, but also to make the relative comparison between the two systems. This can create problems when initial technical or heritage estimates are too optimistic to maintain throughout the design life cycle, resulting in cost or schedule overruns [31].

Engineering build-up, also referred to as a grassroots or bottom-up, is performed by creating cost estimates for the system at the lowest level of detail and rolling those up to create an overall estimate. These detailed cost estimates come from the work breakdown structure (WBS) that includes material and labor costs, often with added overhead costs and fees. This method is typically used during detailed design because the required detail is not available during early design. Because it requires so much detail to create, engineering build-up cost models are intuitive and defensible. Their credibility is provided from the visibility of the WBS. The low level cost estimates often come from the cost engineer working directly with technical experts who are familiar with the activities. However, this also makes this method inherently costly, requiring a significant amount of time and effort to collect all of the necessary information. The breadth of the WBS also makes it easy to either duplicate or omit cost elements from the estimate. Another disadvantage of this method is its inability to easily adapt to any design changes or answer 'what if' questions. Each time there is design modification or alternative scenario, a new estimate must be built from the beginning.

Other, less renowned methods include expert opinion, extrapolation, and process-based estimating [82]. The expert opinion approach solicits opinions from subject matter experts. While often used, it is generally not preferred because it is prone to errors due to bias from the expert [126]. Extrapolation uses past program costs to estimate a future program costs. Learning curves are an example of a technique used to extrapolate cost. This is only applicable when little has changed from the previous project. Process-based, or activity-based, costing uses the relationships between processes and the resources used to complete that process to build a cost estimate [112]. It is similar to the engineering build-up method in that they both roll up smaller costs to create an overall cost estimate.

2.7.2 Reliability-Based Cost Estimating Methodologies

Based on the identification of reliability as a system-level performance metric, a review of reliability-based cost estimates was also conducted. Figure 23 provides a notional illustration of the cost vs. reliability curve, also referred to as 'contractor's cost vs. reliability' and 'dependability vs non-dependability cost' [20, 69]. This figure shows that investing in increasing the reliability of a product corresponds to a decrease in the operation and support cost and vice versa. Another interpretation is that development cost represents the cost spent to avoid failure, or prevention cost, and operations and support represent the cost of failure or having to correct mistakes [118]. This leads to a total cost curve with an optimal total life cycle cost. Classic examples of this include Juran's Cost-of-Quality Model and Crosby's Cost-Of-Quality Model [37, 71]. Crosby's model considers quality as conformance costs. Conformance costs include appraisal and prevention costs, where non-conformance costs include the cost of rework or scrapping failed parts. Juran's model introduces the idea of benefits to the system, considered intangible or opportunity costs. A weakness of these qualitative cost models is that they do not allow for differentiation between VVT strategies. There is no indication of how or when a certain quality or reliability is reached, so two very different VVT strategies could have the same cost but using vastly different methods.

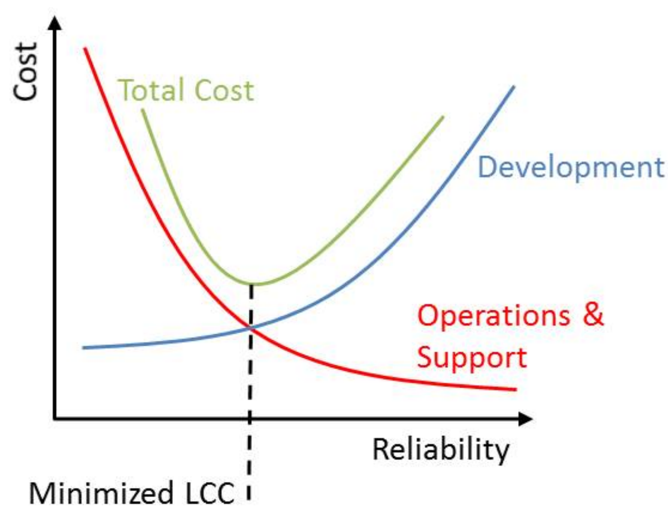


Figure 23: Generic cost of quality model.

One example of a quantitative cost model uses generic cost functions to relate cost and reliability. These can be used when the actual cost function is unknown. They are used to enable comparisons between alternatives with different reliabilities by using the same cost function on the alternatives. These functions have four requirements [10]:

1. The cost must be a monotonically increasing function of reliability.

2. The cost of high reliability is very high.

3. The cost of low reliability is very low.

4. The derivative of cost (with respect to reliability) is a monotonically increasing function of reliability.

While other cost functions have been used, one example of applying the generic cost function is the use of an exponential cost function in the reliability cost model developed by the ReliaSoft Corporation [93]. In their model the component reliabilities of a system are optimized to construct the maximum reliability for the minimum cost. The exponential cost function used for each component contains three parameters, f_i, $R_{(i,min)}$, and $R_{(i,max)}$:

$$c_i(f_i, R_{i,min}, R_{i,max}) = e^{\left[(1-f_i)\frac{R_i - R_{i,min}}{R_{i,max} - R_i}\right]} \tag{14}$$

The feasibility parameter, f_i, also known as the traditional rate parameter of an exponential distribution, represents the difficulty in increasing the component reliability, $R_{(i,min)}$ is the initial or current component reliability, and $R_{(i,max)}$ is the maximum achievable reliability. The feasibility here can depend on the design complexity, technological limitations, or weighting factors. The lower the feasibility value, the faster it approaches infinity. ReliaSoft has created a well-developed model for assessing the cost of increasing reliability. Although this model does not include any performance impacts on the reliability or cost, the feasibility parameter is a useful tool in differentiating the system components.

Another study performed by Krevor optimizes launch vehicle architecture selection based on performance, reliability, and cost [74]. The goal of Krevor's environment is to select the optimal cost and final reliability configuration for a given performance requirement. Beginning with feasible launch vehicle configurations, the reliability is tuned by altering the number of engines, increasing the thrust-to-weight ratio and adding redundant subsystems. The development and production costs are determined using an initial NAFCOM (NASA Air Force Cost Model) estimate for the system as is, and then multiplying the subsystems by their level of redundancy, r_i. For example the final cost of a launch vehicle with subsystem redundancies $r_i = 1, 2, 3$ would be:

$$C_{Total} = \sum_i r_i C_i \tag{15}$$

Although the main focus of Krevor's methodology is the mature reliability of the system, a reliability growth technique is included. Using Duane's reliability growth model, the reliability for the optimal architectures are compared using the same growth rate parameter, α, and the same number of flights to maturity [46]. The only differences between the configurations are the initial and mature reliabilities. The initial costs for each configuration include the development and theoretical first unit cost. The cost for each subsequent flight is calculated by adding the cost of the average production unit.

Another method to link reliability and cost is the Technical Uncertainty Rework Cycle (TURC) and Production Development Control Lever (Prodecol) developed by Rocketdyne to help control cost and schedule of their technically innovative product [64, 65, 66]. This model is an extension of their TURC model, which uses a technical uncertainty factor (TUF) to estimate the number of rework cycles during development. To create the chart in Figure 24, SMEs were asked to reflect on the J-2 and F-1 advanced technology engine programs and estimate the starting TUF. Then the cost of rework (CRW) required during each program was researched along with the causes of each rework cycle. The result is the relationship between the TUF and CRW in the Prodecol chart.

Although this study shows the strong correlation between increasing cost and rework cycles, the process to create the TURC and Prodecol charts is very labor

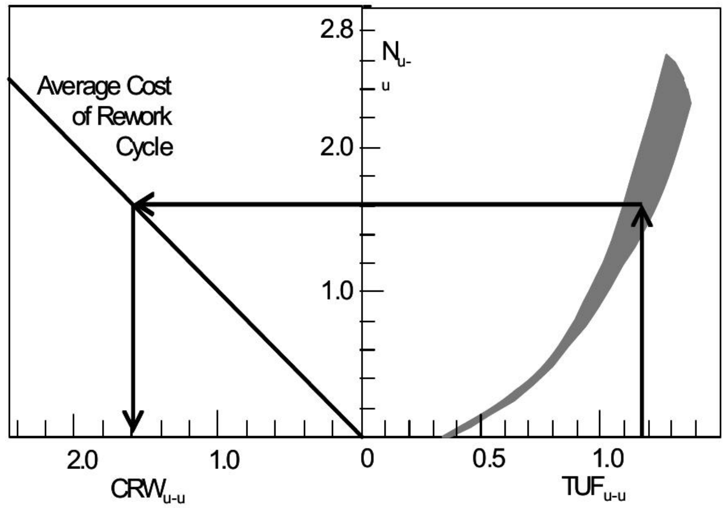


Figure 24: Rocketdyne TURC-Prodecal chart [65].

intensive. It requires sorting through historical documents to obtain all of the rework cycles and their causes and costs for an entire program, which are not always available [65]. Rocketdyne suggests other methods to estimate these values, but they are subject to bias and the experience limitation of the focus group [66]. Another limitation of this model is the cost of rework cycles. To calculate the cost of rework cycles, the total development cost was multiplied by the percent of cost that was used for corrective actions, and then that total was divided by the number of rework cycles documented. The benefit to this cost assessment is its simplicity, but in reality all rework cycles are not created equal. Depending on how late in the phase the defect is found or how critical the component, some rework cycles can cost much more to complete [144]. Finally, despite the fact that schedule slips are noted as a consequence of unplanned rework, this study does not directly address the schedule impact.

2.7.3 Conjecture for Research Question 4

To select the most appropriate method for determining the impact of rework cycles on cost during VVT, the techniques discussed in Sections 2.7.1 and 2.7.2 are compared to the criteria introduced in the beginning of this chapter — quantitative, stochastic, and flexible. The first three methods that were reviewed in Section 2.7 are the three standard cost estimating methodologies. Parametric and analogy cost estimating is typically used in the earlier design phases and rely on relevant historical data. Once the model is developed, these methods produce estimates rapidly and are flexible enough to use for rapid design trades. However, their accuracy is contingent on the quality and quantity of historical data collected. A lack of relevant data for launch vehicles and VVT processes limit the use of these methods. The engineering build-up, or grass-roots, method is a more detailed approach that would increase the accuracy of the cost estimate, but decreases the flexibility compared to parametric and analogy costing. It could be used to account for the stochasticity in rework cost because the individual rework cycles can be assessed individually and rolled-up to a final cost estimate.

The reliability based cost models are less established, but directly assess reliability. The generic cost functions provide a flexible and rapid approach to cost estimating based on reliability improvements. While convenient for understanding the effects of increasing reliability, they only allow for comparisons between different levels of reliability. They also do not consider how the reliability is attained. For example, two different VVT strategies could reach the same reliability using vastly different activities, but still appear to have the same cost. This and the qualitative nature of the approach make it less suitable to this problem.

Rocketdynes Prodecol method explicitly addresses the impact of rework cycles on cost. The development of the chart requires extensive effort, but once developed it is a rapid tool that would be flexible enough to determine how sensitive the system is to rework efforts. It does provide a quantitative cost estimate, but only for rework cycles. It does not consider the cost of testing to uncover the need for rework, reducing its accuracy for the complete VVT process. It also includes two simplifying assumptions that would not apply to launch vehicles. The first is the use of a linear cost curve, which would only be accurate over many rework cycles. The second is using a constant average cost per rework cycle, when in reality the costs for rework cycles can vary widely, as discussed at the beginning of this section. The Prodecol method could be used to provide an estimate of the average cost per rework cycle, and then an assumption could be made on how rework costs increase throughout VVT based on historical trends.

After comparing these five cost estimation approaches, a conjecture for research question 4 can be made. Engineering build-up method would produce the most accurate cost estimate compared to the other four. Although it is less flexible and rapid than the others, it can account for the stochasticity of rework cycles by assessing their costs individually. it has been selected as the most suitable costing approach to use for this methodology.

Conjecture: 4
Using engineering build-up to calculate cumulative cost will give a quantitative estimate that is more accurate than historical data based methods and accounts for the stochastic nature of rework cycles.

2.8 Risk

The final VVT evaluation metric identified in Section 2.2 is risk, which yields the following research question.

Research Question: 5
How can cost, schedule, and reliability risk be quantified?

In Systems Engineering, risk is considered to be the combined effect of the probability of an undesirable event and the consequence of that event. According to NASA's Risk Management Handbook, risk is characterized by the following questions [41]:

1. What can go wrong?

2. What is the likelihood that it will go wrong?

3. What are the consequences if it does go wrong?

Figures 25 and 26 illustrate the difference between uncertainty and risk in a risk analysis of alternatives example [41]. In this example, the outcome is analyzed for all significant possible input decision alternatives. The input variable ranges are defined based on the uncertain conditions of these alternative scenarios. Given the presence of uncertainty, any one decision alternative will produce only one outcome in a range of forecasted outcomes. A distribution of outcomes is characterized by a probability density function over the performance measures. Figure 26 illustrates the performance risk portion of the risk analysis of alternatives example [41], or the probability of not meeting a performance requirement. By incorporating the individual risks and aggregating their effects, the risk of not meeting the requirement can be quantified.

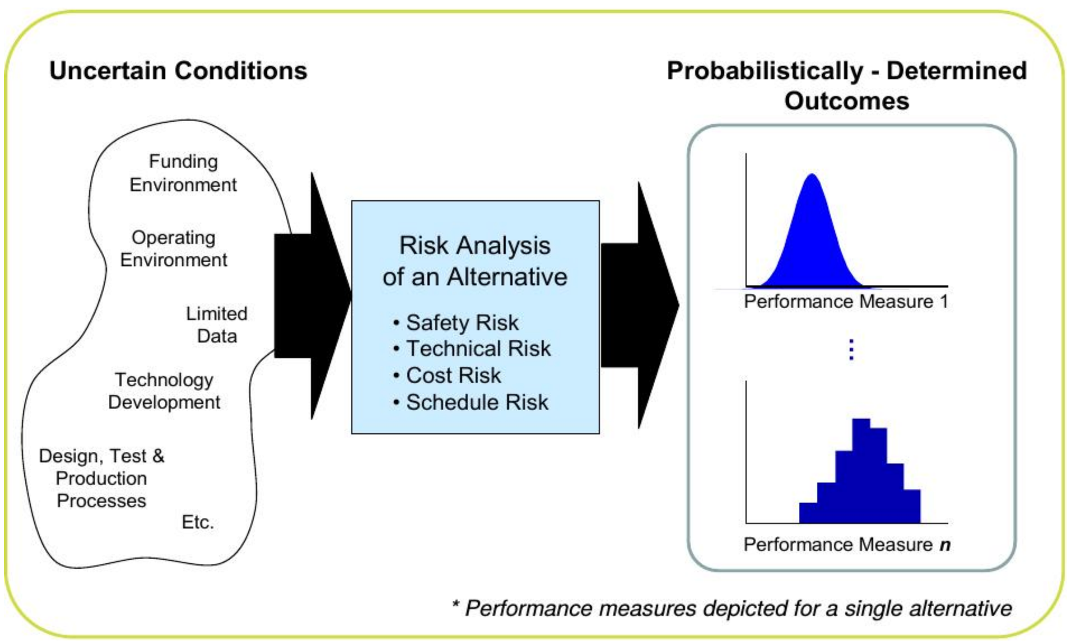


Figure 25: Performance uncertainty defined in the NASA Risk Management Handbook [41].

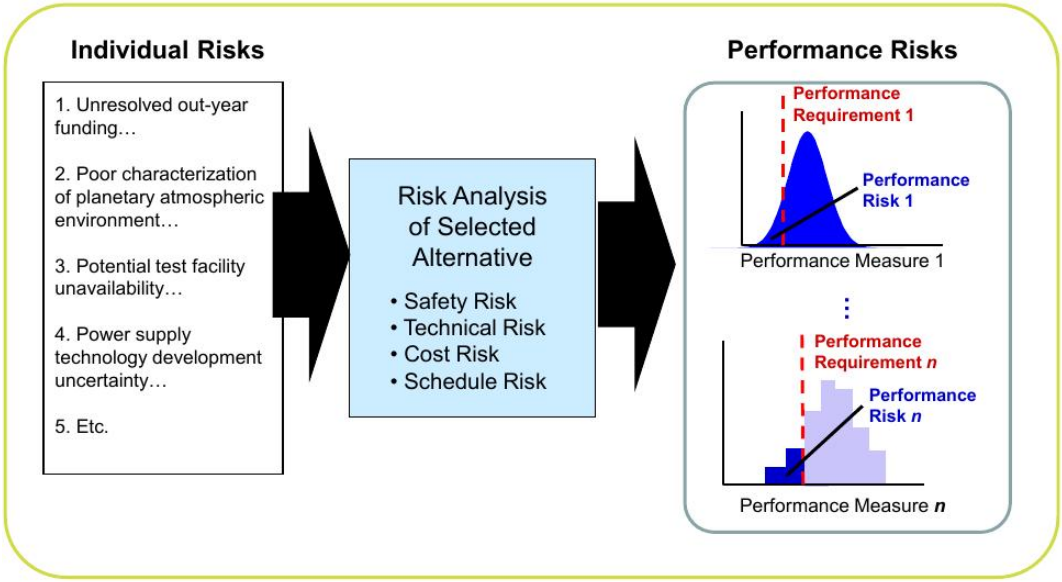


Figure 26: Performance risks defined in the NASA Risk Management Handbook [41].

The risk of a single event is quantified by multiplying the uncertainty by its consequence. Many studies that address either product development processes or VVT processes directly, quantify 'risk', but actually define risk as uncertainty. It is important to consider both uncertainty and its impacts to accurately estimate risk. The following sections will address both of these topics.

2.8.1 Uncertainty

Chapter 1 discussed the uncertainty in launch vehicle design over multiple disciplines. While cost and schedule overrun and performance shortfalls do happen, the degree to which they happen are uncertain. It is not a launch vehicle specific problem. For example, schedule slippage is a prevalent issue for construction companies and shipbuilding is sensitive to mass growth uncertainty [113, 123]. To quantify these uncertainties and better estimate them, it is necessary to understand where they come from and how they relate to each other.

2.8.1.1 Sources of Uncertainty

Morgan and Granger provide a good discussion on sources of uncertainty relating to policy and how to handle them, including probability distributions and statistical techniques [96]. They state that a major benefit of addressing uncertainty is that it requires the planners and experts to focus on the problem and think specifically about the issues involved with proceeding. Schrader, Riggs, and Smith discuss the uncertainty involved in technical problem solving and make the distinction between uncertainty and ambiguity [128]. They specify uncertainty as lack of knowledge about the value of recognized variables relevant to the problem. In this context uncertainty is related to a problem with a well-understood model and representation. Ambiguity arises from a lack of problem definition, where the functional relationships between variables are not understood and sensitivities have not been determined [128]. Thunnissen also recognizes ambiguity as a primary type of uncertainty, but defines it as a linguistic imprecision where terms are not clearly defined leading to uncertainty [145]. His framework includes three other primary types of uncertainty for a complex system, e.g. interaction, aleatory, and epistemic. Interaction uncertainty is defined as unknown or unanticipated interactions between disciplines or subsystems. Aleatory and epistemic are two of the most commonly used uncertainty categorizations [90, 115, 123]. Aleatory uncertainty consists of variables that cannot be precisely known or are inherently random. Examples include the wind speed on launch day or manufacturing imprecision. These variables can be treated as random variables [69]. Epistemic uncertainty is due to a general lack of knowledge about the system and can be reduced through testing. While other frameworks exist, a general consensus on aleatory and epistemic uncertainty can be made. Ambiguity is treated separately in some instances, but can be considered as epistemic since it comes from a general lack of knowledge.

Epistemic uncertainty is often further partitioned into more specific classes of uncertainty. Thunnissen divided it into model, phenomenological, and behavioral uncertainties [145]. Model uncertainty is related to the accuracy of a mathematical model used to describe the system and includes uncertainty from approximations or programming errors. Phenomenological uncertainty is defined as the lack of knowledge of particular phenomena behavior of the system. The third classification is behavioral uncertainty which is uncertainty related to choices made by people or organizations involved in system development, including design errors or requirements changes. Robertson uses some of these same sources of epistemic uncertainty, but categorized them broadly as either exogenous or endogenous before individually addressing them [123]. Exogenous uncertainty is defined as being out of the control of the program development office, such as funding availability or requirements changes. Conversely, endogenous uncertainty is within the control of the program. It includes uncertainty associated with technical development challenges, test failures, overly optimistic assumptions about heritage, and technology readiness levels [123]. This endogenous epistemic uncertainty is directly reduced through efficient VVT processes. These sources of uncertainty can be quantified and modeled in different ways to determine the overall uncertainty level in a project. The next section discusses some of the methods that are used to represent and propagate uncertainty to the system level.

2.8.1.2 Quantifying Uncertainty

Overall uncertainty in the system can be quantified using a variety of techniques. These techniques can vary in accuracy, traceability, and speed. While some are rooted in historical data that can be difficult to acquire, others require an extensive framework to implement. The first set of techniques determines a range of uncertainty based on a deterministic initial prediction. The rest of the methods are probabilistic, treating the future value as a random variable and using statistical techniques to quantify uncertainty.

Deterministic uncertainty analysis techniques are based off a single point estimate or a set of point estimates. Then the uncertainty is informally evaluated using historical data or expert opinion. Historical analogies are used to determine a range of uncertainty based on historically similar projects [13]. Based on a projected final value, the uncertainty is determined by the range of values for historically similar projects. While simple to implement, this method is sensitive to the availability of historical data, which can be difficult to obtain for launch vehicles. Additionally, the selection of 'similar' projects can be subjective and hard to define. Another method that uses a deterministic initial estimate is empirically derived growth factors [13]. This method typically uses a linear regression of initial estimates to final value ratios from similar historical projects to determine the mean and variance of the new estimate uncertainty. This method has been used by the Air Force to assess weapon system cost growth [45]. Similar to the historical analogy technique, growth factors are easy to use, but are entirely dependent on the availability of historical data.

Expert opinions are commonly elicited to determine uncertainty values for a particular project [41]. This method is used in many organizations and utilizes the experience of subject matter experts who are familiar with the system. Uncertainty estimates can be made at the system level or at the subsystem level and rolled up to the system level [94]. This flexibility enables it to be used at any level of detail. While the use of experts eliminates the need for historical data, its accuracy can be affected by the subjectivity of the expert [134]. Cost estimation is particularly susceptible to overly optimistic assumptions due to program pressures to reduce cost [67].

Probabilistic uncertainty techniques are used to develop a probability distribution function of the final value. They typically begin with an estimate of the probability distribution for each input variable, which can be gathered using expert opinion as discussed previously. The more detailed the problem breakdown can be, the better the estimates will be. For example, the engineering build-up cost method discussed in Section 2.7.1 uses individual cost estimates for each entry in the work break down structure. The probabilistic techniques differ in their method to aggregate these individual uncertainties.

Propagation of errors is an analytic version of a sensitivity analysis. It is a summation of the errors in each input multiplied by the partial derivatives of the function with respect to that input variable [13]. It is a well-known method that is commonly used for calculating error and uncertainty related to spatial modeling with geographic information systems [95]. One advantage of this method is that it does not require the computational overhead of simulation. However, for complex systems the partial derivatives can be difficult to evaluate analytically. Another analytic technique is the method of moments. This method is used in the parametric cost model NAFCOM, NASA Air Force Cost Model [35]. The total cost estimate is equal to the sum of the individual cost estimates, so the final cost probability distribution is similarly equal to the sum of the individual distributions. This method can be used to sum the means and variances assuming any reasonable distribution [13]. Similar to propagation of errors, the method of moments does not require simulation and can be easy to compute for less complex systems with few components.

Simulation is another probabilistic technique that is being used more and more as the computational effort required to complete it is being reduced. It is especially suited for problems with no known closed-form solution that cannot be analytically evaluated. A Monte Carlo Simulation (MCS) generates a random number from each input variable distribution and computes the final value. This is considered a single run, which is then repeated thousands of times to generate a probability distribution of the final value. The probability distribution can also be displayed as a cumulative distribution illustrating the probability that the variable is less than or equal to that value. An advantage of using MCS is that it is a common numerical technique that is well-understood and accepted. Also, advances in computer simulation have reduced the computational expense of using this method. One assumption that is often used in MCS is that the inputs are independent of each other, which may not always be true. To account for the relationship between input variables, a correlation matrix must be defined, adding to the complexity of this method. Another disadvantage is that it is dependent on the accuracy of the input distributions. Another simulation method is a discrete event simulation (DES) which models the time-based, or dynamic, behavior of a system. Using mathematical or logical models of the physical system, DES portrays state changes at precise points in simulated time. Similarly to MCS, it also can handle probability distributions for the input variables to generate a probability distribution of the output variables. The advantage of DES is the addition of a time component that does not exist for MCS, resulting in more information about the system. Consequently, this additional information requires more effort in the beginning to develop the model. The various distributions that can be used for inputs into these simulation models are discussed in the next paragraph.

Cost and schedule distributions tend to have a right hand skew, implying that they are more likely to go above the expected value than below [34]. Because of this, a normal distribution will not be used. Three commonly used distributions for cost and schedule uncertainty are the Beta, Weibull, and Triangular distributions [13]. The Beta probability density function is shown in Figure 27 and defined by the equation below [150]:

$$f(x;\alpha,\beta) = \begin{cases} \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)} & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{16}$$

where α and β are shape parameters that define the skew and variance of the distribution. $B(\alpha,\beta)$ is Beta function that is shown here:

$$B(\alpha, \beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt \tag{17}$$

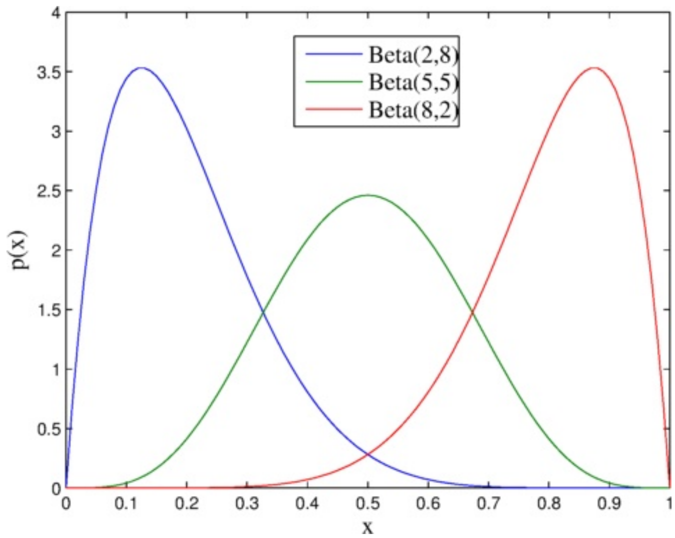


Figure 27: Beta probability density function.

The Weibull probability distribution function is shown in Figure 28 and described by the following equation [150]:

$$f(x; \lambda, k) = \begin{cases} \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k} & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{18}$$

where k and λ are shape parameters.

The simplest PDF that can represent the cost and schedule skewness, is the Triangular distribution, which is shown in Figure 29 and described by the following equation [150]:

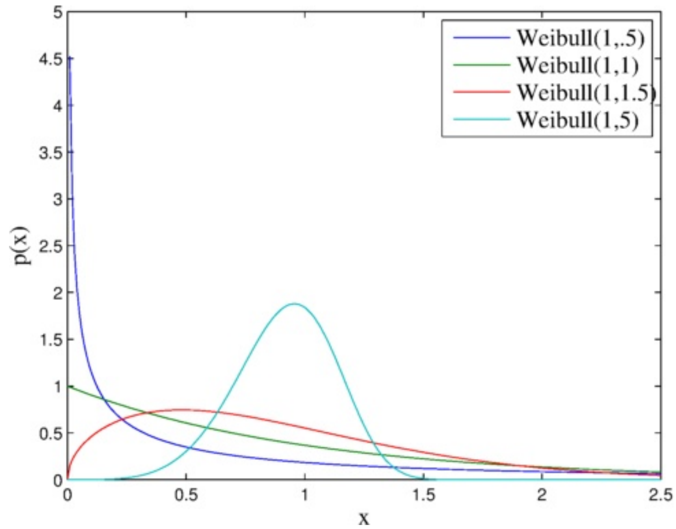


Figure 28: Weibull probability density function.

$$f(x;a,b,c) = \begin{cases} 0 & \text{for } x < a \text{ or } x > b \\ \frac{2(x-a)}{(b-a)(c-a)} & \text{for } a \le x < c \\ \frac{2}{b-a} & \text{for } x = c \\ \frac{2(b-x)}{(b-a)(b-c)} & \text{for } c < x \le b \end{cases} \tag{19}$$

where a is the pessimistic estimate, c is the most likely estimate, and b is the optimistic estimate.

2.8.2 Consequence of Uncertainty

The other half of risk is the consequence of undesirable events occurring. One method for evaluating these consequences is the risk matrix introduced in Section 2.4. Based on the previous identification of individual risk scenarios, that method directly measures the impact of each scenario. Advantages of this method are that it is very traceable and simple to develop. However, it is limited by the ability of experts to predict possible risk scenarios.

Another method to determine the consequences of undesirable events is the use

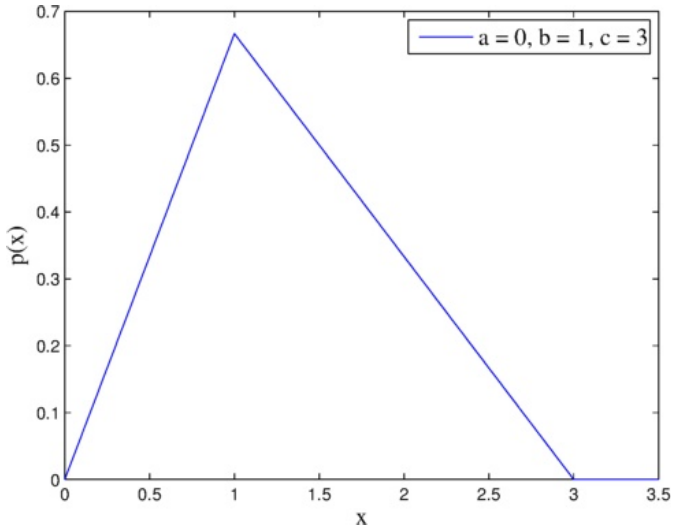


Figure 29: Triangular probability density function.

of risk impact functions, as shown below.

$$R = \int_{T}^{\infty} f(x_0) I dx_0 \tag{20}$$

where R is the total risk of exceeding the target value, T, $f(x_0)$ is the PDF of variable x, and I is the impact function. Impact functions can be constant, linear, or quadratic depending on the nature of the problem. Quadratic quality loss functions were first highlighted by Taguchi, and have been used in similar models [26, 68]. This implies that the risk experiences quadratic growth as it gets further away from the target, which would be appropriate for a cost and schedule model.

2.8.3 Conjecture for Research Question 5

The first section for the literature review on risk discussed the general definition of risk, and identified that both uncertainty and consequences of undesirable outcomes are necessary to evaluate risk. Section 2.8.1.1 discussed different frameworks for uncertainty. Although a wide variety of uncertainty sources are described by different authors, in general, most agree that uncertainty can be broadly classified as epistemic uncertainty and aleatory uncertainty. Aleatory uncertainty can only be classified and always exists within the system. VVT activities are designed to reduce sources of endogenous epistemic uncertainty.

Section 2.8.1.2 reviews deterministic and probabilistic methods to quantify uncertainty. The deterministic techniques include historical analogy, growth factors, and expert opinion. For all of these methods, uncertainty is estimated based on a single deterministic value. While most of these are fast and easy to implement, they depend heavily on the availability of historical data, which is not always available for novel systems. Expert opinion can be used for deterministic risk analysis or probabilistic risk analysis. For the probabilistic approach, the expert provides a distribution for the individual inputs, compared to providing a distribution for the final value. In both cases it can be a fast technique, but is not very traceable. While it does eliminate the reliance on historical data, it is affected by the biased experiences of the expert. The two analytic probabilistic techniques reviewed were propagation of errors and method of moments. These methods are not well suited for complex models and would not fare well for launch vehicle risk analysis. The remaining probabilistic uncertainty quantification technique is simulation. It is a fast and traceable technique that provides a quantitative measure of uncertainty. The ability to represent a time component with DES, enables the evaluation of stochastic rework cycles and adds one more level of comparison between alternative VVT strategies, making it the most suitable choice.

The final discussion for uncertainty quantification is the choice of input probability distribution functions. For cost and schedule it is necessary to use a distribution with the ability to represent skew. The three distributions discussed were triangular, beta, and Weibull. While all of them are used in practice, the triangular distribution is the simplest of the three with the most intuitive inputs. If the cost and duration estimates are gathered from experts, it is more reasonable to ask for a pessimistic, most likely, and optimistic estimate than for shape parameters. This will increase the traceability of the uncertainty quantification.

Finally, Section 2.8.2 discusses the evaluation of the impact or consequence of undesirable outcomes. A quadratic impact function has been used for activity process modeling, and to assess VVT cost risk. Due to its application in similar problems, it can be chosen for the proposed research based on the literature review. This review of uncertainty and uncertainty quantification techniques leads to the following conjecture:

> **Conjecture: 5**
>
> - Using triangular input distributions, the assumptions required will be more traceable than if beta and Weibull distributions are used.
> - If DES is used for simulation, the results will allow for quantitative comparisons between VVT strategies and account for stochasticity of rework cycles.

With the structure in place to model the effects of rework cycles, the remaining research question addresses how rework cycles can be estimated.

2.9 Rework Probabilities

The VVT methodologies currently in existence lack a consistent method for identifying the rework probabilities during VVT. While many authors acknowledge the effect rework cycles have on a product development life cycle [51, 64, 68, 110], very little attention is given to determining their likelihood and impact. This leads to the final research question:

> **Research Question: 6**
>
> How can the probability of rework cycles be estimated?

The following section provides a review of the causes for rework and discusses some of the methods for determining rework probabilities during various phases of development.

2.9.1 Rework Drivers

Iteration can be a useful and necessary process during product development, but when unplanned, it often leads to cost and schedule overruns. In order to determine rework probabilities, it is first important to understand what drives the need for rework. Arundachawat conducted a literature review to determine the causes of rework during design phases [15]. The first driver captured from this review is project complexity. Programs that utilize lessons learned from previous projects by taking advantage of design heritage for future designs are considered less complex and see fewer rework cycles. This approach has been taken by NASA SLS, which is designing the launch vehicle to progress with block upgrades. Programs that take advantage of computer aided engineering tools to eliminate critical defects are also considered less complex. Subsystem dependency was also identified as a design rework driver [15]. Rework effort was found to be reduced when programs managed the sequence of subsystem design based on relevant subsystem dependencies.

The last two rework drivers identified can be classified as programmatic design issues. Early communication between design teams was shown to affect the amount of rework. Teams that were more cooperative and had clearly defined objectives were able to reduce the amount of rework that arose during development. Lastly, the presence of resource constraints was also shown to affect rework during design [15].

Complexity was also found to be a rework driver during the later development phases [64]. Another driver that caused rework during design and manufacturing was design maturity. Designs that had a higher degree of heritage experienced fewer unplanned iterations during all development phases. The operating environment and addition of advanced technologies were two rework drivers identified during the implementation phases, but not explicitly identified as drivers during design. These could be related to complexity, but are addressed separately. The last rework driver identified in the literature is the amount of activity overlap [151]. It was found that dependent activities create more rework as the overlap between them increases. The purpose of overlap is to reduce the amount of time required for the project, but it decreases the amount of information that is available at the start of the later activities. This idea of information exchange appears in some of the approaches used to determine the probability of rework effort. These are discussed in the next section.

2.9.2 Approaches to Determine Rework Probabilities

A Rocketdyne study of their F-1 and J-2 engines discussed in Section 1.1 developed a quantitative measure of technical uncertainty for estimating the number of rework cycles during the development of technically innovative products [64]. The Technical Uncertainty Factor (TUF) is a function of design maturity, complexity, technology, and system environment. TUF is intended to be an improvement over the Technology Readiness Level (TRL) implemented by NASA which some consider inefficient due to its lack of cardinal meaning [34]. The four criteria used to determine the TUF value are similarly categorized, but the ratings assigned to the system are a relative measure of the respective criteria. A heuristic relationship is then created between TUF assessments and rework cycles, as seen in Figure 9. In future projects, once a TUF assessment is made, the TUF-Rework Cycle Chart will give an estimate of the rework cycles that can be expected during VVT. The simplicity of this approach is appealing in estimating rework cycles. However, this relationship is only valid for the Rocketdyne team and would need to be recreated for use by other teams. Creating the relationship between TUF and expected rework cycles would require extensive effort and rely heavily on subjective input by SMEs. Another disadvantage of this approach is the lack of detail in the uncertainty definition and absence of any time components. While efforts can be made to reduce the overall number of rework cycles, this method generalizes the rework cycles, making it impossible to differentiate between the rework cycles that have a higher impact on the system.

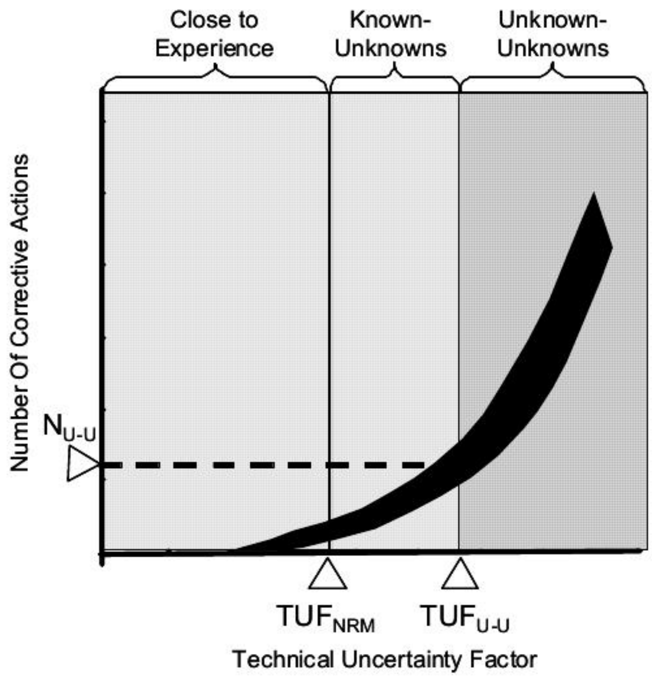


Figure 30: Rocketdyne TUF-Rework cycle chart [64].

An analogous methodology to estimate design rework cycles based on the relative influence of common drivers was suggested by Arundacahawat et al [14]. After a survey of three automotive industry projects, six rework drivers were identified and used to predict the probability of rework occurrence and degree of rework efforts. The relative influence of the rework drivers to a new product can then be determined, and analogous rework probabilities and impacts evaluated. The narrow sampling for data collection limits the applicability of this method, but a unique method to represent the rework probabilities was also introduced. A design structure matrix (DSM), which was discussed in Section 2.6.3 was used to model the system components. More commonly seen in the literature is an activity-based DSM where the system's activity network relationships are represented [72]. Here, it is used to model component interactions. The component-based DSM entries represent the probability that change in one component will cause rework in another.

Engel and Barad suggest the use of historical data to determine risk probabilities, but highlight the lack of relevant historic data related to systems VVT [51]. Another method for estimating rework probabilities during product development processes uses a subjective expert assessment [157]. The risk of rework in this method contains two components, the variability of the input information for a given task and the sensitivity of the task to that change in information. SMEs are asked to categorize each task as low, medium, or high. Then these values are mapped and calibrated to probabilities using project specific proportionality constants. This representation of rework as two information components is similar to the approach used by Browning and Eppinger, but no method to determine these components is given in that study [25]. Another subjective approach used in literature is a straightforward subject matter expert opinion of rework probabilities. Roemer et al. and Krishnan et al. use this strategy to reduce cost and schedule for projects with overlapping design and development phases [75, 124]. The lack of structure to this approach, however, creates the opportunity for bias.

While discussing reliability analysis techniques, Failure Mode and Effects Analysis (FMEA), or Failure Mode, Effects and Criticality Analysis (FMECA) was introduced in Section 2.5.2.3. The purpose of FMEA is to identify potential failure modes and determine the cause and effect of each mode. Three risk factors are defined during FMEA [105]:

- Occurrence (O) the probability a failure mode will occur
- Severity (S) the impact a failure mode will have on the system if it occurs
- Detection (D) the probability that a failure mode will be detected during

inspection or test

These three factors are used to prioritize the failure modes identified by calculating a Risk Priority Number (RPN) [44]:

$$RPN = O * S * D \tag{21}$$

The RPN is used to categorize failures by their level of criticality. Some analysts use 3 levels of criticality, from the most critical level 1 to the least critical level 3, while others further differentiate from 1-10 [44, 146]. In either case, critical 1 failure modes will lead to top level risk events, like Loss of Crew (LOC) or Loss of Vehicle (LOV). Lower critical failure modes can lead to damage or reduce performance of the system. FMEA has historically been an important step in preventing failures from occurring and increasing the reliability of the system [84].

The information provided by FMEA can be directly applied to rework cycles during VVT. The probability of failure modes can be used in the reliability growth model to predict when rework will occur. The impact of rework cycles can also be determined by the RPN or criticality level of the failure modes. The Complex systems can have a vast amount of components that require major effort to analyze. The failure modes with higher RPN pose the most risk to the system and need to be reexamined. These critical 1 failure modes can be assumed to result in redesign efforts due to their increased probability of LOC or LOV. The lower risk failure modes can be assumed to be resolved with remanufacturing. The percentage of failure modes that fall into these categories can be used in the DSM to represent the relationship between testing activities and design or manufacturing activities, enabling differentiation between the impact of redesign and remanufacturing on reliability, cost, and schedule.

2.9.3 Hypothesis for Research Question 6

From the literature review, it appears that many of the methods used to estimate rework cycles are largely subjective and subject to bias by the expert. The expert opinion method can lack traceability. It has been shown that SMEs are often hesitant to provide any quantitative predictions and are biased by their own experiences. This can reduce the accuracy of the probabilities they provide. Basing the estimates on historical data can be limiting for launch vehicles because there is a lack of relevant data available. The Rocketdyne TUF method requires either historical data or expert opinions based on relevant previously completed projects to develop, limiting its usefulness. The analogy method has similar draw backs due to the lack of available historical projects to use as the foundation of the estimate for launch vehicles.

Another method that was discussed is the representation of rework as two information components. Again, SMEs are used to determine the qualitative assessment initially, but these estimates are then mapped to quantitative values. This model of rework is more suitable to design than VVT activities, but the use of SME knowledge to map qualitative opinions to quantitative rework probabilities improves on the weakness of the expert opinion method. A disadvantage of this method is the lack of quantitative impacts on cost and schedule.

While conducting the literature review of reliability methods to derive Hypothesis 2, FMEA/FMECA was discussed. This method was identified as a possible technique for identifying rework probabilities. FMEA/FMECA data provides a more structured approach for garnering expert opinions. As a well-established and commonly used technique, it is less sensitive to data availability than the other methods. It also is more traceable than simply asking an expert for direct estimates of rework probabilities. Cost and schedule impacts can also be determined based on FMECA data. For each failure mode identified, the severity and effects are given. The completeness of this technique makes it ideal for gathering rework probabilities and impacts.

FMEA appears to provide a more traceable approach to gather rework probabilities and assess their impacts than the other methods and will be the least sensitive to data availability. Consequently, the following hypothesis for research question 6 can be formulated.

> **Hypothesis: 6**
> If subsystem and system level FMEA is performed, then the resulting data will provide quantitative rework probabilities that are more traceable than expert opinion and the data will be more readily available than expert opinion and all historical data based methods.

2.10 Background Research Summary

The research questions posed in this chapter identify the components necessary to build a methodology to meet the research objective. Each question addresses a specific element of the RIVVTS methodology. After the need for a methodology to assess the impact of rework during VVT was established, the first research question was used to determine how the value of a VVT strategy can be defined. A review of industry standard VVT planning approaches and academic studies on improving VVT and reducing rework cycles led to the formulation of conjecture 1 that selected quality, cost, schedule, and risk as the four metrics that would provide the most complete assessment of rework impact on VVT activities and enable a quantitative comparison between alternatives. Through literature search and additional research questions in Sections 2.5 - 2.8, hypotheses and conjectures were developed to further define these metrics and select existing techniques or suggest improvements on existing techniques. The last research question is posed to address the gap identified in Section 2.4.3 for estimating the probability of rework cycles. Section 2.9 discusses the primary rework drivers during development, and reviews existing techniques for estimating rework probabilities and impacts. Hypothesis 6 was formulated to provide a more traceable approach to estimating rework during VVT.

Chapter 3 discusses these research questions and introduces experiments that have been design to test the hypotheses formulated in this chapter. The setup of each experiment is introduced first, followed by the results and observations of that experiment, and conclusions are drawn to either support or refute the hypothesis. Using a combination of literature review and experimentation, the individual components of this methodology are developed. Following the formulation of these components, a detailed description of the complete methodology is provided in Chapter 4.

CHAPTER III

METHOD DEVELOPMENT

The development of the research objective for this thesis identified the need for improvement in the planning and evaluation of VVT activities during Phases C/D. The previous chapter determined the components needed to model and support trade-offs between distinct VVT strategies and presented hypotheses for some of these components. This chapter discusses the experiments that were designed to test these hypotheses, and either support or refute them based on the results. The first section describes the system that has been selected for use in these experiments.

3.1 Launch Vehicle Subsystem Design Problem

A launch vehicle subsystem design problem is used to test the accuracy of the components of this methodology during development and enable a comparison of the model to actual vehicle data. The Space Shuttle Main Engine (SSME) is identified for use in the following three experiments. The SSME is an example of a liquid rocket engine, which makes it an ideal candidate for this methodology because the largest percentage of historical launch vehicle failures can be attributed to failures of liquid-rocket propulsion systems, approximately 40-47% [89]. Additionally, the SSME has a lengthy and well documented development history. An introduction to the SSME system and its testing history is given in this section.

3.1.1 SSME

The Space Shuttle Main Engine, also known as the RS-25, is the first large, reusable liquid rocket engine built [17]. It burns liquid oxygen (LOX) and liquid hydrogen

(LH2) in a staged-combustion cycle. Figure 31 illustrates the simplified power propellant flow schematic and identifies the major components, which include two turbopumps for the fuel and oxidizer (i.e. low pressure and high pressure), hot gas manifold, fuel and oxidizer preburners, main combustion chamber, heat exchanger, and nozzle. The staged combustion cycle partially combusts a portion of the propellants at a fuel-rich mixture ratio, and then uses that mixture to drive the high pressure turbopump turbines prior to being completely burned in the main combustion chamber. This cycle was chosen for increased efficiency, but utilizing the high and low pressure turbopumps also increased complexity. These four turbopumps contained 47.3% of the total 5,807 component parts for the original SSME configuration, known as the First Manned Orbital Flight configuration [87]. Over the course of its 40-year history, it underwent six major 'phase' or 'block' changes [148]. The final engine configuration performance parameters are listed in Table 7.

Table 7: SSME Performance Parameters [17]

Propellants	LOX/LH2
Rated power level (RPL)	469,448 lb
Nominal power level (104.5% RPL)	490,847 lb
Full power level (109% RPL)	512,271 lb
Chamber pressure (109% RPL)	2,994 psia
Specific impulse at altitude	452 sec
Weight	7,748 lb
Service life	55 flights
	27,000 sec
Total program hot-fire time	3,171 starts
	1,095,677 sec

3.1.2 SSME Development

Since the first Space Shuttle flight in 1981, the SSME has been used in clusters of three to provide propulsion during the entire program. At a cost of $40 million each, a total of 46 engines were flown. Because of its long flight history, NASA plans to continue using the SSME to power the SLS core stage [40]. The flight-proven reliability of the

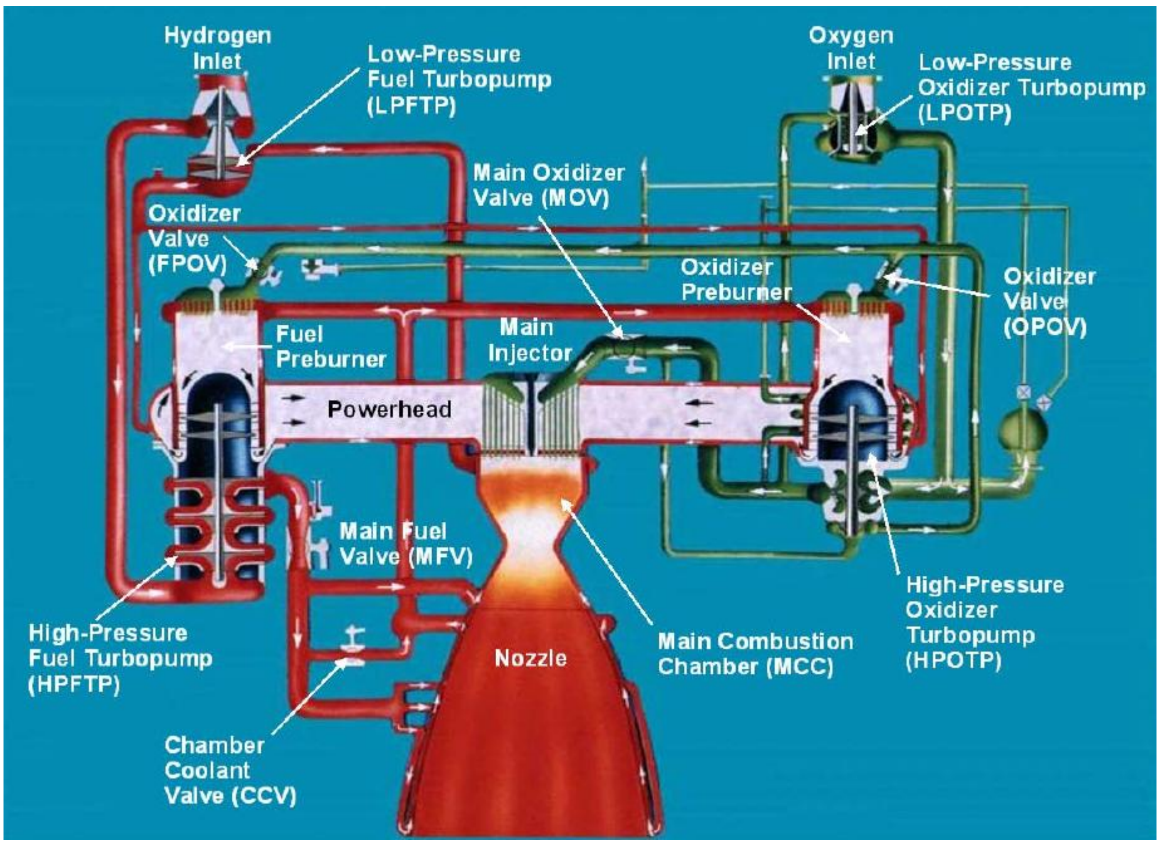


Figure 31: Simplified SSME Schematic [148].

SSME can be attributed to its extensive test program [17].

Aerojet Rocketdyne began development of the engine in 1971 when it was awarded the contract to design, develop, and produce the SSME. Engine testing began in 1975 with a series of short ignition tests. The dual fuel and oxidizer turbopumps required a sensitive initial control sequence to safely start and shut down the engine [17]. A summary of the major test failures for the program is shown in Figure 32. Other engine cutoffs resulted in additional rework and redesign, but the eight problems shown here were identified as the most critical to ensuring flight safety. Management instituted a dedicated effort for solving these problems by assigning full-time 'special team' members to eliminate failure modes. These multi-disciplinary teams were tasked with identifying the cause of the problem, establishing a path to safely resume testing, and ultimately redesigning and reworking a solution [17].

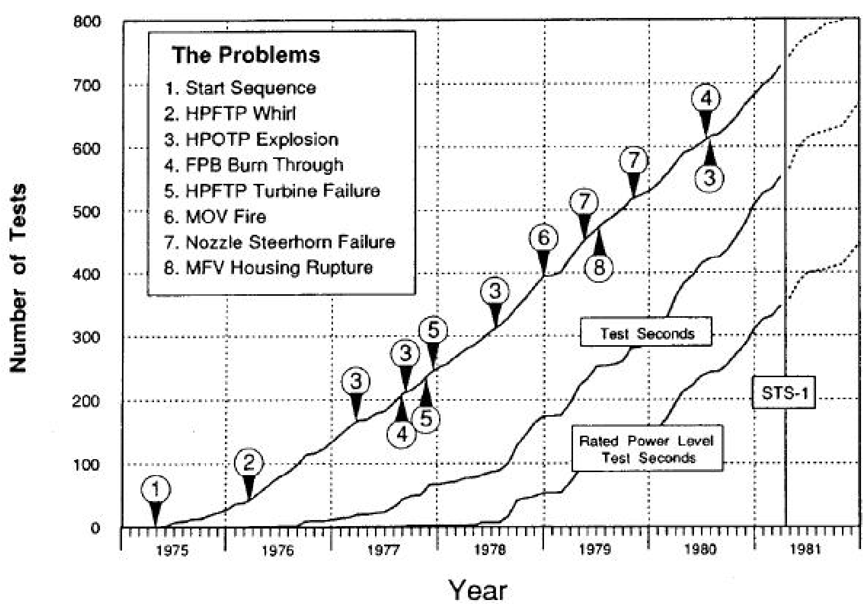


Figure 32: Summary of engine test problems during development [17].

The complete failure history of the SSME during testing is shown in Figure 33, recreated from [138]. The cumulative failures plotted against test numbers show the reliability growth expected during development as engine improvements are made and test-analyze-and-fix cycles correct problems. The growth curve is steeper at the beginning and flattens out as the number of tests increase. The causes of these premature engine cutoffs include failure modes of all criticality levels.

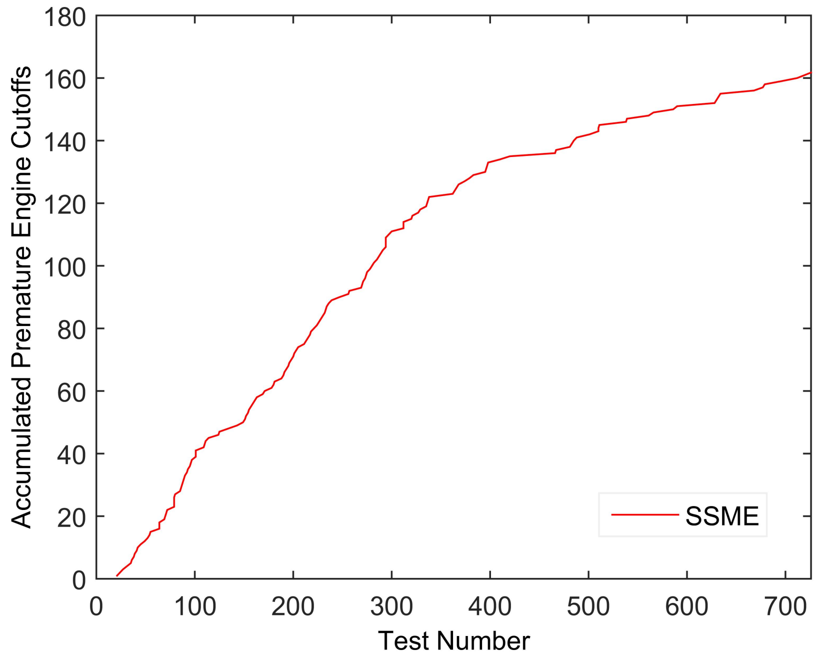


Figure 33: Accumulated engine cutoffs verses test number for the SSME development.

When the Space Shuttle's first flight occurred, the SSME had accumulated 726 hot-fire engine tests and over 110,000 seconds of operation [148]. Table 8 lists the number of tests conducted and average test duration per test for each group. Thirteen tests and 5,000 seconds of operation were required to certify the engine, including both normal and the longer abort mode flight profiles. Should any test fail during certification, the entire cycle would have to be repeated.

This review of the SSME development history is provided to support the three experiments presented in the remaining sections of this chapter. Due to its well-documented test history, it provides an excellent foundation for testing the hypotheses formulated in Chapter 2. Each experiment will address a different aspect of the SSME test program. The first experiment tests the use of fidelity levels to improve

Table 8: SSME Test History

Average Duration (s)	Number of Tests
2	27
21	107
97	184
158	132
183	121
283	128
400	21
520	6

the reliability growth projection during VVT by comparing Hall's discrete reliability growth model with and without test fidelities. The demonstrated SSME reliability growth during testing is used to evaluate the result. The second experiment tests the use of FMEA to provide traceable and accurate probabilities for rework cycles. A comparison of FMEA to historical-based data and expert opinion is provided to evaluate the traceability of each approach. Following this discussion, the SSME FMEA data is used to test the accuracy of estimating the total number of rework cycles and the occurrence of rework cycles. The results of this experiment are compared to the actual SSME rework history to determine the validity of this approach. The final experiment addresses the weakness of current DSM approaches to estimate overall schedule during testing phases. A discrete event simulation is run first with the probability of internal rework explicitly included in the DSM, and then with the probability of internal rework implicitly included in the individual activity duration. The results are compared the SSME development schedule to either support or refute the hypothesis made in Section 2.6.4. The results of these three experiments are used in conjunction with the conjectures made in Chapter 2 to formalize the RIVVTS methodology.

3.2 Experiment 1

After reliability was selected as an appropriate measure of quality in Conjecture 2, research question 2a was posed to determine which analysis technique is most suited for tracking reliability during VVT. A review of several techniques identified reliability growth models, specifically Hall's discrete model for one-shot systems, as a possible approach for predicting launch vehicle reliability during the testing phases. The assumption that every test is equivalent was identified as a weakness in the Hall model when applied over the entire testing phase. Hypothesis 2a was formulated in Section 2.5.3 to address this weakness, and is restated below. To test this hypothesis, Experiment 1 addresses two primary considerations for the growth model: 1) the ability of Hall's reliability growth model to accurately assess reliability during development testing, and 2) given that the accuracy of the model is acceptable, the level of insight provided into individual testing activities.

> **Hypothesis: 2a**
> If Hall's reliability growth model is adapted to include defect elimination as a function of test fidelity, it will provide reliability projection with quantitative insight into individual VVT activities.

The first part of Experiment 1 is set up to run Hall's reliability growth model without any modifications for the SSME. The results are compared to the SSME demonstrated reliability. It is expected that as is, the model will over predict reliability due to the varying fidelity level of tests that occur during development. Section 3.2.3 discusses existing approaches that are used to apply Hall's and other reliability growth models during testing phases. Typically, these take the form of combining multiple tests into a single trial to support the assumption made in the model of identical trials. This approach limits the insight into the individual testing activities

Table 9: Hall reliability growth model parameters [62]

Parameter	Definition
k	Initial number of failure modes in the system.
T	Number of trials during testing phase.
d_i	Fix effectiveness factor.
$p_i \sim Beta(n, x)$	Failure mode probabilities of occurrence, Beta distribution shape parameters.

in favor of modeling the overall test phase. An adaptation to Hall's model that includes a test fidelity measure for each test is introduced at the end of Section 3.2.3. The intent is to allow the model to be used for the full testing phase and maintain insight into individual tests by including the test fidelity level. Three fidelity measures are tested with Experiment 1b to determine their accuracy in predicting reliability growth during test phases. The amount of information provided by the approaches in Experiment 1a, 1b and the existing approaches in Section 3.2.3 is compared to determine if more insight into testing activities can be gained by including the test fidelity level during reliability projection.

3.2.1 Experiment 1a Setup

After the SSME was identified for use in the following experiments, the first step is to set up the Hall reliability growth model and determine appropriate ranges for the required parameters listed in Table 6, which are restated here. As the first large, reusable rocket engine, it is modeled as a new design with no historical data. To avoid any bias in the experiment, the only engine data that are used to compare model results are the demonstrated reliability data.

The first parameter to be considered is the number of failure modes that lead to loss of crew, loss of vehicle, or loss of mission — the top level failure for launch vehicle systems. Without historical data to generate this assumption, Zwack showed that the simple parts count method, discussed in Section 2.5.2.5, can be used to estimate number of failure modes [158]. In this context, the major components are used as the physically separate parts that contribute to the system level reliability. It is assumed that each of these components will contain at least 1 catastrophic failure mode, and can be equated to the number of top level failure modes for the system [158].

Using the PCM approach, the ten major components listed in Section 3.1.1 represent ten failure modes. This estimate can be compared to the actual failure mode count by looking at the detailed FMEA [146]. 190 failure modes are identified in the report and are assigned a risk factor from 0 to 1.0, shown in Table 10. 7.5% of those are given risk factors above 0.25, 14 failure modes, meaning they will likely lead to the loss of the vehicle or loss of engine [146]. Ten of those failure modes are determined to lead to a probable loss of vehicle. The other four lead to probable engine loss, and are included for a conservative estimate. That gives a range of 10-14 critical 1 failure modes. The parts count estimate falls within this range, confirming that it can be used as a substitute for estimating the number of critical 1 failure modes if FMEA data were not available.

Table 10: SSME FMEA risk factor definitions [146].

Severity	Description
1.000	Loss of vehicle
0.500	Probable loss of vehicle
0.333	Loss of engine
0.250	Probable loss of engine
0.200	Extensive engine damage
0.167	Local engine damage
0.143	Minor damage
0.125	Very minor damage
0.111	Piece part damage
0.100	Part still ok

The next parameter to consider is the probability of occurrence for these failure modes. The Morse model provides estimates for launch vehicles based on general history [97]. Three defect groups are identified by their frequency of occurrence: high, medium, and low. Table 11 lists the number of defaults, and probability of occurrence for each frequency based on historical launch vehicle reliability data. The probability of occurrence is a conditional probability of loss of mission, given that the failure mode is triggered. Table 11 lists this probability of occurrence calculated by multiplying λ_k and τ_k from Table 5. Based on this assessment of failure probabilities, approximately half will fall in the low frequency group, a third in the medium frequency group, and the remainder in the high frequency group. If the SSME is assumed to have 10-14 failure modes as discussed previously, there will be 5-8 low frequency, 3-4 medium frequency, and 0-2 high frequency.

Table 11: Morse general launch vehicle defect assumptions [97].

Characteristic	High	Medium	Low
Number of defects	0-5	1-12	2-20
Prob. of occurrence	18%-71%	.25%-7.5%	.05%-1.5%

Hall provides a method to determine the failure probability of occurrence based on failure data [62, 63]. These probabilities are modeled as a Beta distribution, with shape parameters α and β. The procedures to estimate these parameters are established for two cases. The first case is for a known number of failure modes, k, and the second is for the number of failure modes approaching infinity. Hall uses an air-to-ground missile system to demonstrate this procedure. A sample of the shape parameters predicted for these one-shot systems is listed in Table 12.

Table 12: Hall Beta shape parameters for one-shot systems [62].

Beta Parameters (α,β)	Mean	Maximum
0.19, 23.31	0.008	0.3
0.36, 14.99	0.023	0.4
0.19, 8.03	0.022	0.52
0.22, 8.75	0.024	0.54

If Hall's method of estimating failure mode probabilities is used with the SSME data in Figure 32, these values can be compared to the Morse assumptions to determine an accurate range for this Experiment. The last two Beta distributions given by Hall fit well with the Morse failure probabilities. The majority of the failure modes would fall within the low frequency failures, with a variance that provides 0-1 failure modes in the high frequency range. Using the Method of Moments Estimate procedure suggested by Hall for the critical failures of the SSME, $E[p_i] = .02\%$. The assumed number of failures used was $x = 14$ according to Figure 32, and the number of trials was based on equivalent full duration tests. To determine equivalent tests, the total number of test seconds is divided by the designed mission duration, 520 seconds. This aligns well with the Hall and Morse probability of occurrence ranges. The distribution chosen for this experiment is $Beta(0.22, 8.75)$ to allow for the medium and high frequency failure modes.

The final Hall model parameter to determine is the Fix Effectiveness Factor. As discussed in Section 2.5.2, this is the percent reduction in probability of occurrence of a failure mode, given that the mode has occurred and a fix has been implemented. This parameter is equal to Morse parameter γ_k, which represents the probability that a defect is eliminated from the system given that it has been detected and reported. In practice, this value is difficult to derive from vehicle data. Hall states that the fix effectiveness factor is typically assessed by subject matter experts and assigned during failure prevention review boards [62]. For the purpose of this research, this value is determined from the literature.

Morse states that the probability of eliminating a failure mode from the system, γ_k, is high for launch vehicles. This is due to the large amount of post launch flight data analysis that is typically performed after a launch. The large expense of flight tests and the extreme reliability requirements for crewed vehicles demand that any and all defects that have been detected within the system should be investigated and mitigated. Morse calls out a range of between 75% and 90% for this value [97]. Hall provides a table of fix effectiveness factors from an air-to-ground missile program. These factors are said to have been developed during a failure prevention and review board for the program. The table presented by Hall shows a range of 70% to 95% for this value.

A range for the fix effectiveness factor is determined for Experiment 1 based on the proposed Hall and Morse models. As previously mentioned about verification and validation testing, development tests will have a significant amount of instrumentation and sensors to collect data. Engine development takes up a significant portion of launch vehicle development costs and is held to the same reliability requirements because it is a crewed launch vehicle subsystem. Zwack suggests that the fix effectiveness factor is impacted by a long flight history, which allows the designer to assess the system many times and implement failure mode corrective actions [158]. Due to the fact that the SSME had a long and detailed test program, a high fix effectiveness is assumed, and a uniform range of $U(90\%, 95\%)$ is used for Experiment 1.

All of the necessary parameters for the Hall model have now been defined, and the model can be implemented for the SSME. The equations given by Hall [62] have been coded in MATLAB for the initial analysis of Experiment 1. A Monte Carlo simulation was run under these conditions. To determine the accuracy of this model during development testing, it is compared to the SSME reliability calculated in reference [138], where the AMSAA model was used to calculate the Mean Time Between Failure (MTBF) based on failure modes of criticality 1. For an initial test of this method, it is run as described, using the total number of SSME development tests as the number of trials. The reliability is expected to be over estimated under these conditions. This model assumes that each trial has an equal probability of uncovering all of the failure modes. However, when considering a test phase that contains tests at different fidelity levels, each test will not have an equal probability of detecting all of the failure modes. Table 12 lists the values of each parameter in the initial run of the Hall model.

Table 13: Reliability growth assumptions for Experiment 1

Parameter	SSME Value
Number of Modes	10-14
Probability of Occurrence	$\beta(0.22,8.75)$
FEF	90%-95%
Number of Tests	726

3.2.2 Experiment 1a Results

The results of the initial Hall run are plotted in Figure 34. To account for uncertainty in the input assumptions, a Monte Carlo simulation was performed. For each of the 10,000 runs, a random number was drawn from each of the input variable distributions in Table 13. The resulting output allowed for the calculation of a mean and percentiles for the model to be used for comparison. From this plot, it is clear that the Hall model over predicts the SSME reliability during the entire development phase. The result of this initial run confirms the expectation that using the same probability of failure occurrences for each flight will result in over prediction of reliability. There are some methodologies to avoid this, but they lose insight into the effect of individual tests on reliability. These methods are discussed in the following section, and a follow-on experiment to determine the best way to model test fidelities for a more accurate prediction of reliability during testing is presented.

3.2.3 Reliability During Testing

There are three methods to account for different levels of testing during development. The first is to consider a group of similar tests as a phase and determine reliability during that phase, and repeat for the number of test phases planned. The main drawback for this approach is the additional work in predicting reliability growth parameters for each phase. AMSAA and Duane models, discussed in Section 2.5.2.6, use a growth parameter. This growth parameter is determined based on the reliability growth of similar historical systems. While this model is simple enough to implement,

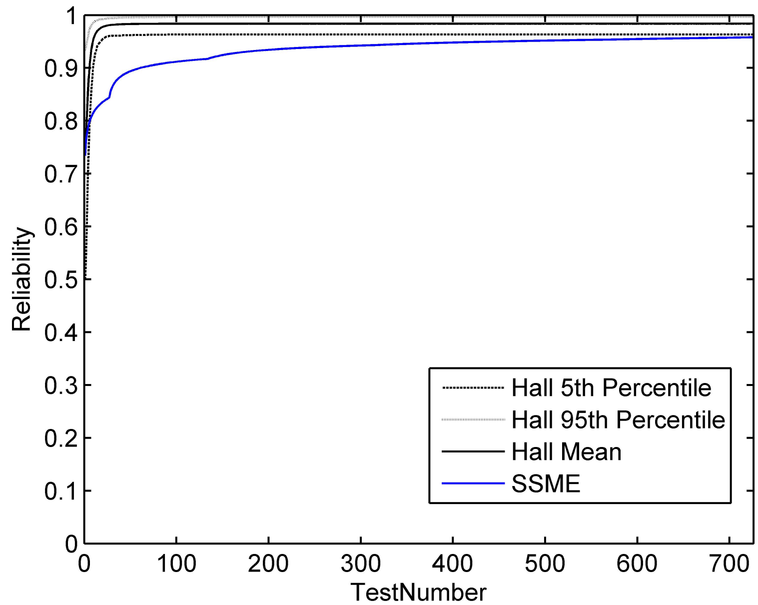


Figure 34: Hall model growth predictions versus SSME data.

it does not provide any insight into how individual tests are affecting the reliability of the system. It also does not allow for any changes in the development plan. The last method to be discussed, is the use of equivalent flights. Zwack uses this approach during the development of CONTRAST [158]. For engine development, Zwack determines a number of equivalent flights to contribute to the reliability growth before it is fully integrated into the launch vehicle. The total number of test seconds is divided by the duration of a full mission to calculate this value. For the SSME, this would be Eq. Flight Number $= 110,000s/(520s/flight) = 211$ flights. Then the reliability growth model is evaluated for 211 trials, instead of the full 726 because it is assumed that there is an equivalent amount of testing effort applied. While this approach can be used to model reliability growth during testing, it does not provide any information about individual tests.

Another possibility for defining test fidelity is to consider the system performance. For the SSME, the complicated start-up sequence due to the dual turbopumps would imply that more failure modes are uncovered earlier in the cycle. Instead of using a linear fidelity model based on the test-duration/full-duration ratio, the fidelity would show a growth curve similar to reliability growth, with a steeper increase in fidelity during start-up and then a flatter increase as the test continues. This is supported by SSME failure occurrence experience, where 60% of failures occur during start-up, 20% occur during the first third of the full duration, and the last 20% occur up to flight mission completion [156].

Based on this discussion, Hall's model is adapted to account for lower fidelity tests. The probability of occurrence for failure modes is determined assuming a full duration test. Lower fidelity tests will not be able to uncover all of the failure modes. For example, a computer simulation would not be able to detect manufacturing failure modes, and component testing would not detect any failure modes due to integration. The definition of fidelity for this methodology is the percent of failure modes that are able to be uncovered during a single test, f_t. The test fidelity is defined for the specific system and set of testing activities being modeled.

Mathematically, this is represented in the calculation of Hall's indicator function, Equation 11, restated here:

$$E[I_i(t)] = 1 - (1 - p_i)^t.$$

For the initial evaluation of Hall's model in Section 3.2.2, the indicator function is evaluated for each failure mode, i. It is calculated incrementally, until failure mode i occurs, then the test number of the first occurrence, t_n, of that failure mode is set, $I_i(t_n) = 1$. For example, the indicator function is evaluated at test t_1, if the failure mode does not occur, then the indicator function is evaluated for t_2, t_3, and so on until the failure mode occurs (i.e. $E[I_i(t_1)]$, $E[I_i(t_2)]$, $E[I_i(t_3)]$). In the original implementation, every test is considered for every failure mode. So at test t_3 the indicator function for failure mode i would be $E[I_i(t_3)] = 1 - (1 - p_i)^3$. When this model is adapted to include test fidelity, a test with only 50% fidelity will only be considered for 50% of the failure modes.

A simple example system with 10 failure modes can be used to further illustrate this concept. The fidelity levels for the first three tests are $t_1 = 100\%$, $t_2 = 50\%$, and $t_3 = 100\%$. Test t_2 can only be evaluated in the indicator function for the first 5 failure modes because it is a 50% fidelity test. At test t_2, the indicator function for failure mode 6, assuming it did not occur in test t_1 will be kept at 0, $I_6(t_2) = 0$. At test t_3, the indicator function for failure mode 6 will be $E[I_6(t_3)] = 1 - (1 - p_6)^2$. Experiment 1b is setup to determine an accurate way to represent test fidelities for this system.

3.2.4 Experiment 1b Setup

Three fidelity measures are tested with Experiment 1b to determine their accuracy in predicting reliability growth during test phases. The Hall parameter values used in Experiment 1a, listed in Table 13, are kept for Experiment 1b. Only the test fidelity levels, f_i, are changed. The first two cases are based on a linear fidelity profile, calculated by dividing the test duration by the full mission duration:

$$f_i = \frac{d_i}{D}, \tag{22}$$

where d_i is the duration, in seconds, of test i, and $D = 520s$ is full mission duration. The profile of fidelity levels over the length of a full mission is illustrated in Figure 35. This profile is a similar approach used in generating equivalent flights that was discussed in Section 3.2.3. When 520 seconds is considered one test flight, this implies a linear fidelity profile.

Using these values for fidelity, the first case will assume all tests are of an average length and as such, an average fidelity. For the 726 tests and 110,000 seconds of SSME development, this means the Hall model is evaluated for 726 tests of 151 seconds duration and 29% fidelity. This is not expected to yield promising results, because only 1/3 of the failure modes will ever have the opportunity to be uncovered, and reliability is not expected to show much growth.

The second model run also uses the linear fidelity profile, but the fidelity levels are set using the average yearly test durations listed in Table 8, which steadily increase. This is expected to provide a better reliability prediction capability than the first case, but will likely still under predict reliability.

The third case will use a different profile of fidelity levels from the first two cases. Here, the fidelity profile is modeled to emulate a reliability growth curve which takes the form:

$$1 - e^{-at} \tag{23}$$

where t is test number and a is a fit coefficient. The profile of fidelity levels over the length of a full mission is shown in Figure 35. This curve more closely follows the SSME failure history experience, and is therefore, expected to perform better than the first two cases.

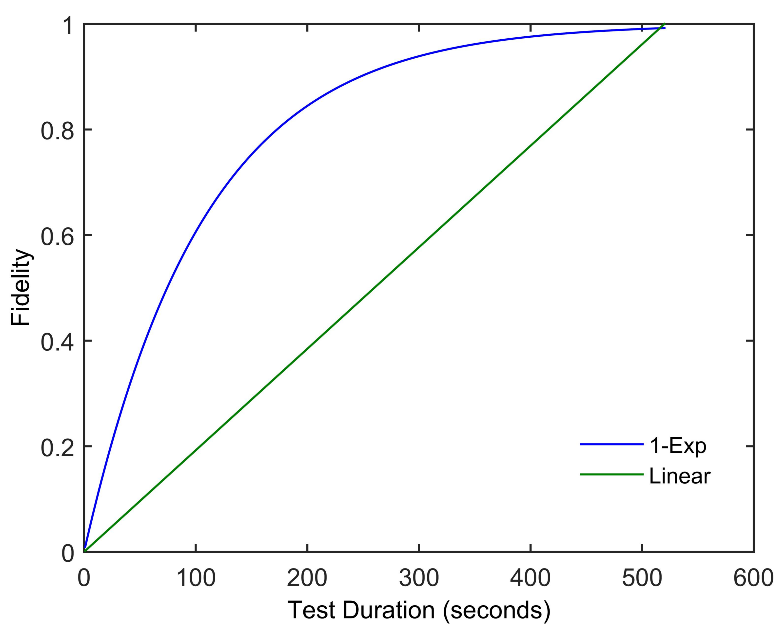


Figure 35: Linear and '1-Exponential' fidelity level profiles.

Now that the fidelity profiles for the three cases have been defined, the models can be implemented for the SSME. A Monte Carlo run was performed, providing a mean

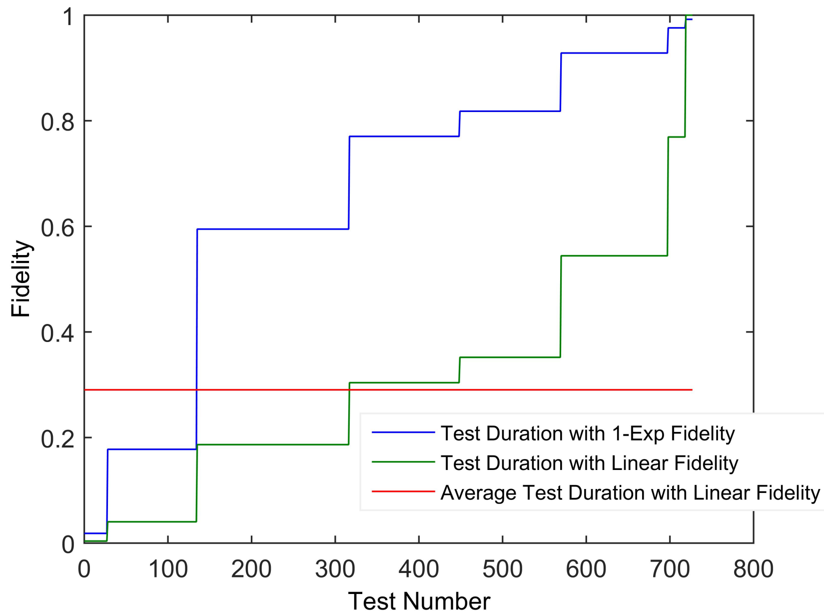


Figure 36: Test profiles with three different fidelity definitions.

and percentiles for each case to be compared to the vehicle data. A root mean square error was taken across the development testing to quantitatively compare profiles. This is discussed in more detail in the following section.

Table 14: Fidelity profiles for Experiment 1b.

Case	Fidelity Profile	Test Durations
1	Linear	Average
2	Linear	Table 8
3	1-Exp	Table 8

3.2.5 Experiment 1b Results

Section 3.1.1 presented a launch vehicle subsystem for use in testing the adaptation of the Hall growth model for use during development testing with variable test fidelities. After determining the initial assumptions required for the model in Section 3.2.1, the three fidelity profiles to be used in Experiment 1b were defined in Section 3.2.4. These assumptions were used to generate reliability growth models for each case using

MATLAB. A Monte Carlo simulation was run for each case, and the resulting mean and percentiles for each case are presented here.

To quantitatively determine the accuracy of these fidelity profiles, the Root Mean Square Error (RMSE) of the model mean and SSME data is calculated using the following equation:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{N}(\hat{Y}_i - Y_i)^2}{N}} \tag{24}$$

where $\hat{Y}_i$ is the SSME data at test i, Y_i is the model mean at test i, and N is the number of tests — 726 in this experiment. The errors for the three fidelity cases are shown in Table 15. This table shows that Case 3 performs the best in terms of RMSE. The 1-Exp fidelity profile used in this case has nearly a full magnitude improvement over Case 1 and 2. Based on this observation, it is expected that Case 3 will more closely predict the SSME data.

Table 15: Mean square error for Experiment 1b.

	Case 1	Case 2	Case 3
RMSE	0.1327	0.1367	0.0287

The results of the adapted Hall model using average fidelity levels as represented in Figure 37 show a poor agreement with the actual reliability values. It shows an early increase in reliability, up to test 10, where it then flattens out and no longer follows the SSME reliability values. This can be explained by considering the test fidelities. Each test has an average fidelity of 30%. As fidelity is defined for this engine system, each test can only uncover the first third of all failure modes. The remaining failure modes are never detected, and therefore reliability does not increase. This is not a practical approach to development testing, and as illustrated by the model results it does not accurately predict reliability.

The results of Case 2 are plotted in Figure 38. The linear fidelity levels increase over the course of the development testing, and as a result the reliability shows more growth than Case 1. Early reliability growth is expected to be steep, however, this

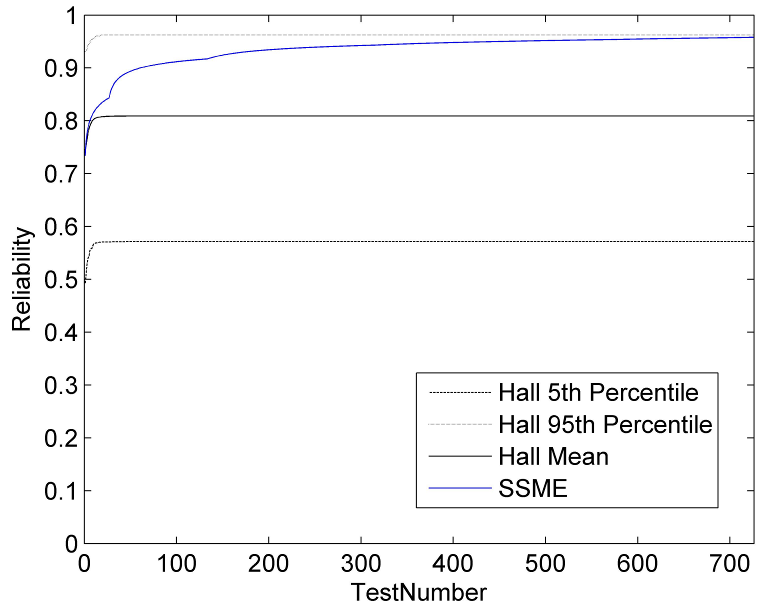


Figure 37: Hall model growth predictions using average fidelity values.

case shows a slower, steadier increase. While the agreement to SSME data is better than Case 1, it is still not a promising option because the values for reliability are well below the actual. The gradual increase in reliability indicates that the linear fidelity profile does not detect failure modes as quickly as the actual system.

Figure 39 shows the results of Case 3, where a 1-Exp fidelity profile is used. In this case, the model captures the growth trend of the SSME very well. In the early development tests, 1-400, the mean value follows the steep reliability curve closely. The model appears to slightly under predict reliability towards the end of the development program, but this provides a conservative estimate of mature reliability, which is preferred to a risky over prediction. The low RMSE for this case listed in Table 15 can easily be understood after plotting the results.

After considering the SSME results from these 3 cases, the appropriate fidelity profile has become apparent. The adapted Hall model using 1-Exp test fidelities tracks the actual data very well. For both cases where a linear definition of fidelity was used, the reliability was grossly under predicted through the entire development

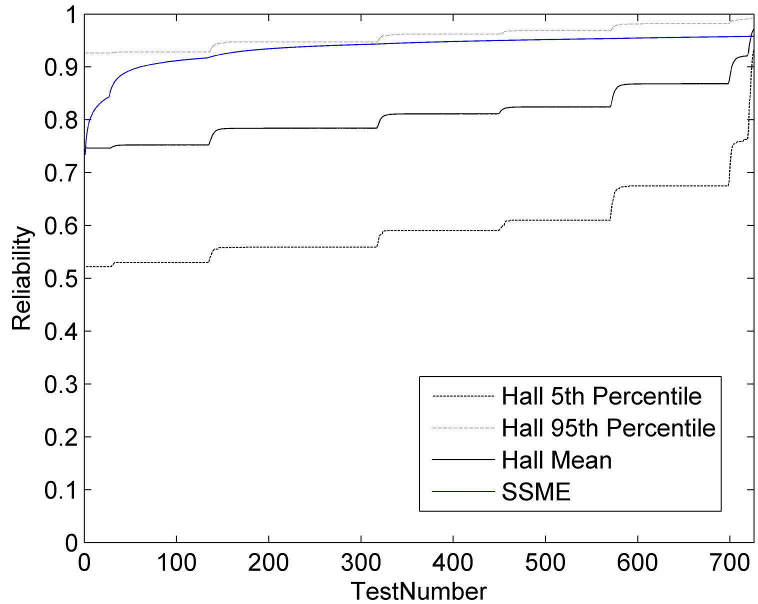


Figure 38: Hall model growth predictions using linear fidelity values.

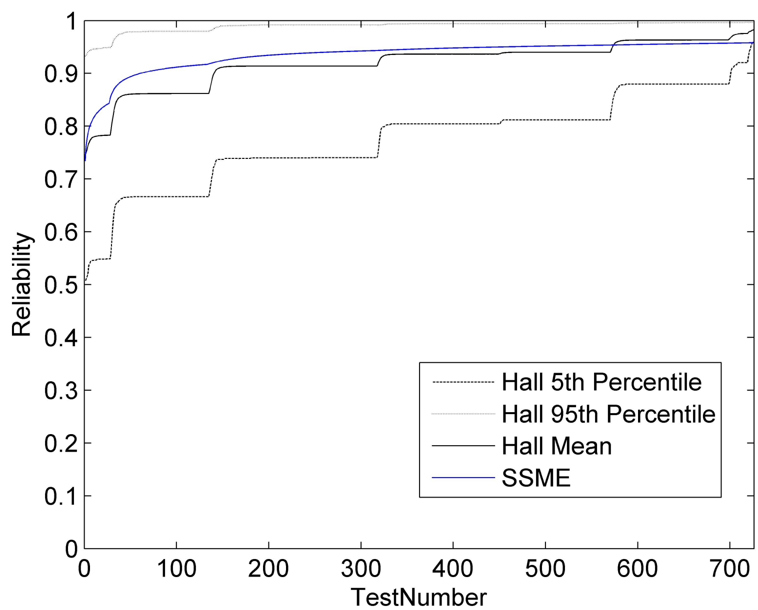


Figure 39: Hall model growth predictions using exponential fidelity values.

testing history. From these results it is clear that the 1-Exp fidelity profile is able to provide more accurate reliability growth predictions. In terms of accuracy, it is the most appropriate for this methodology.

3.2.6 Experiment 1 Summary

Experiment 1 tested the adaptation of Hall's reliability growth model with three different fidelity profiles, to determine which would be used in this method. Experiment 1a first showed the use of Hall's model over the full development testing history of the SSME without the use of fidelity levels. The results showed that this model over predicts reliability because it assumes all tests are weighted equally. Section 3.2.4 describes the three options of fidelity that were tested to adapt the Hall model. The results showed that Case 3 performed significantly better in terms of accuracy over Case 1 and 2. Due to the fact that it was able to more accurately predict the actual SSME reliability, it can be officially chosen for use in this methodology. This conclusion is in line with the original hypothesis, which identified the adaptation of Hall's model to incorporate test fidelity as a method for providing more insight into VVT activities. Therefore, hypothesis 2 can be accepted based on the results of Experiment 1.

3.3 Experiment 2

Section 2.9 reviewed the current approaches for estimating rework probabilities. Hypothesis 6, which is restated below, was formulated to address the weaknesses identified in those techniques. Experiment 2 is designed to test the use of FMEA to provide traceable and accurate probabilities for rework cycles. This experiment will address the primary considerations for estimating rework cycles, the traceability and availability of the data, and determine the accuracy of this approach. First, historical data and expert opinion approaches for estimating rework cycles are compared to FMEA data to evaluate the traceability of each approach. Following this discussion, the SSME FMEA data is used to test the accuracy of estimating the total number of rework cycles and the occurrence of rework cycles. The number of rework cycles is similar to the number of failure modes used in the Hall model, but also includes modes with lower risk factors, or lower criticality failures. The results of this experiment are compared to the actual SSME rework history to determine the validity of this approach.

> **Hypothesis: 6**
> If subsystem and system level FMEA is performed, then the resulting data will provide quantitative rework probabilities that are more traceable than expert opinion and the data will be more readily available than expert opinion and all historical data based methods.

3.3.1 Experiment 2 Setup

The three approaches to estimating the number of rework cycles are subject matter expert opinion, similarity to historical systems, and failure mode and effect analysis. All of these options rely on historical systems to some degree, but are considered separately. The traceability and availability of data using these methods is discussed first.

The first option for estimating rework cycles is to use the opinion of a subject matter expert. This approach uses assumptions about the system based on engineering judgment. To apply this technique, a SME for the system needs to be identified, and then the SME estimates the number of expected rework cycles that will occur during development based on their prior knowledge and experience. The lack of structure in this approach limits its traceability. There is no established standard for documenting the justification or reasoning behind a SME opinion, nor is there a standard practice in determining who is qualified to make assumptions about a system. The traceability of the assumptions used in generating that data is limited to the fact that it comes from an accepted expert. The nature of using previous experiences to apply engineering judgment also introduces bias into the estimate. For these reasons, this approach is considered less traceable than the direct comparison to historical systems and FMEA.

The next approach is comparison to a similar system. This method also requires the use of engineering judgment, but provides an anchor point for comparison. The first step in utilizing this method is to select an existing system that is similar to the current system. An ideal candidate would be one that successfully completed development and was taken into production. After a historical system is identified, the number of rework cycles that occurred during Phase C/D for that system are observed. Finally, someone that is knowledgeable about both systems, such as a SME, predicts the number of rework cycles that will occur during development of the current system based on similarities and differences between the two programs. Systems can be compared based on things like design maturity at different life cycle stages, complexity, commitment to reliability improvements, or programmatic differences (i.e. management style or funding confidence).

The traceability of this approach is improved slightly over direct SME input due to the fact that the existing system can be revisited to determine the reasoning behind the estimate. It is still not ideal, however, because it is not a structured or quantifiable comparison. Indeed, many of the program descriptors that are used for comparison are qualitative and subjective in nature. The data availability of this approach is also a concern, as this method is strongly dependent on the availability of a similar system. This can be a particular problem for launch vehicle systems and subsystems because so few are developed and even fewer are produced.

The last option considered for estimating rework cycles is the use of failure mode and effect analysis data. As discussed in Section 2.5.2, FMEA is an existing reliability technique that identifies all failure modes in the system and determines their effect and criticality level. This information directly relates to the number of expected rework cycles, making it well suited for this purpose.

The number of failure modes can be determined directly from the FMEA data. FMEA produces a comprehensive list of failure modes that result in various degrees of damage. When assessing reliability, as in Experiment 1, only those that lead directly to loss of mission and loss of crew are included, but even failure modes that result in minimal engine damage will require rework. A comprehensive FMEA also provides an assessment of the probability of occurrence. These are either quantitative probabilities or a qualitative assessment that categorizes the failure modes from unlikely to frequently occurring [44]. The quantitative probability of occurrence for each mode can be estimated from the qualitative categories.

The traceability of generating an estimate of rework cycles from FMEA is clearly an improvement over expert opinion or a similarity comparison. The detailed worksheets, shown in Figure 10 are very structured and each failure mode is well documented. Like the other two methods, it can also be based off of historical data. FMEA worksheets are continually developed and improved upon throughout development. Unlike the system comparison method, even if the historical system was canceled before development was completed, the FMEA data are still be relevant. In terms of data availability, a comprehensive set of FMEA worksheets may not be completed when Phase C begins for a new engine with no design heritage.

If a complete FMEA is not available, another approach must be used to estimate rework cycles. Havskjold argued a relationship between rework cycles and a complexity metric he defined as the Technical Uncertainty Factor (TUF), shown in Figure 30 [64]. This metric is relatively subjective and requires a SME to accurately estimate. Another possible complexity metric is the number of qualification tests required. Qualification tests, also referred to as certification tests, are designed to formally verify compliance with performance requirements and specifications. An engine that has a less complex design will require fewer tests to accept that the development program is producing flight hardware that meets specification, and design heritage gives the program a significant advantage by allowing "qualification by similarity" to reduce the number of qualification tests required [5].

The previous discussion determined that FMEA was the preferred method to estimate the number of rework cycles a system will incur during development in terms of traceability. The structured and standardized approach provides a quantified estimate of both rework cycles and probability of occurrence. Hall's model is designed primarily to account only for the probability of high-level faults, like loss of mission and loss of crew. The mathematical model it is based on, however, can be used in this experiment to determine the accuracy of FMEA data to predict rework cycles. The adapted indicator function described in Section 3.2.3 is used with the input assumptions in Table 16 which are derived from the SSME FMEA [146]. To ensure that the test fidelity levels are appropriate for use on failure modes of all criticality levels, the Hall model is run with and without test fidelities included. A Monte Carlo simulation was performed in both cases to account for the uncertainty in the input distributions and to provide a distribution for when rework cycles occur. The results are compared to the number of accumulated test failures that occurred during the SSME development program which are shown in Figure 33 as a function of test number.

Table 16: Reliability growth assumptions for Experiment 2

Parameter	SSME Value
Number of Modes	160-182
Probability of Occurrence	β(0.22,8.75)
Number of Tests	726

If detailed FMEA is not available, another complexity metric can be used to estimate the number of rework cycles. The number of qualification tests, which are called out in the Verification Requirements Matrix, discussed in Section 2.2.1, have been identified as a possible metric to represent the complexity of the engine. To determine the appropriateness of this metric, the qualification tests and rework cycles of four liquid rocket engines are compared: SSME, J-2, F-1, and RS-68. The SSME was introduced in Section 3.1.1, and a brief introduction to the other three engines is given next. The number of rework cycles is determined based on publicly available data on the number of engine failures during development [60, 64].

The J-2 was developed by Rocketdyne in the 60's to power two stages on the Saturn V launch vehicle in the Apollo program. Five engines were used on the S-II second stage, and a single engine was used on the S-IV-B third stage. Currently, an updated version of this engine, the J2-X, is being developed for use on the SLS Earth Departure Stage [54]. The gas generator engine cycle burned liquid oxygen/liquid hydrogen propellants and was designed for a 500 second full duration flight. During its 6 year development, 1,700 tests were completed through qualification [54]. The operating schematic is illustrated in Figure 40, which identifies the primary components: fuel and oxidizer turbopumps, heat exchanger, gas generator, feed control system, main combustion chamber, and nozzle.

The F-1 engine was used to power the first stage of the Saturn V launch vehicle to the moon in 1969. A total of 65 engines were used on 13 Saturn V flights, all with no failures [36]. Also developed by Rocketdyne, the gas generator cycle used rocket propellant-1 (RP-1) and liquid oxygen as propellants. Figure 41 shows the operating cycle that consists of 8 primary subsystems: fuel feed, oxidizer feed, igniter fuel, gas generator, vehicle pressurization, hydraulic control, electrical, and flight instrumentation [54]. A single turbompump, powered by the gas generator, is used to supply the fuel and oxidizer to the thrust chamber. The RP-1 is also used to fuel the thrust vector control system. During its development, the F-1 went through 1,081 tests through qualification [54]. Only 278 of those tests were for 150+ seconds, full mission duration tests.

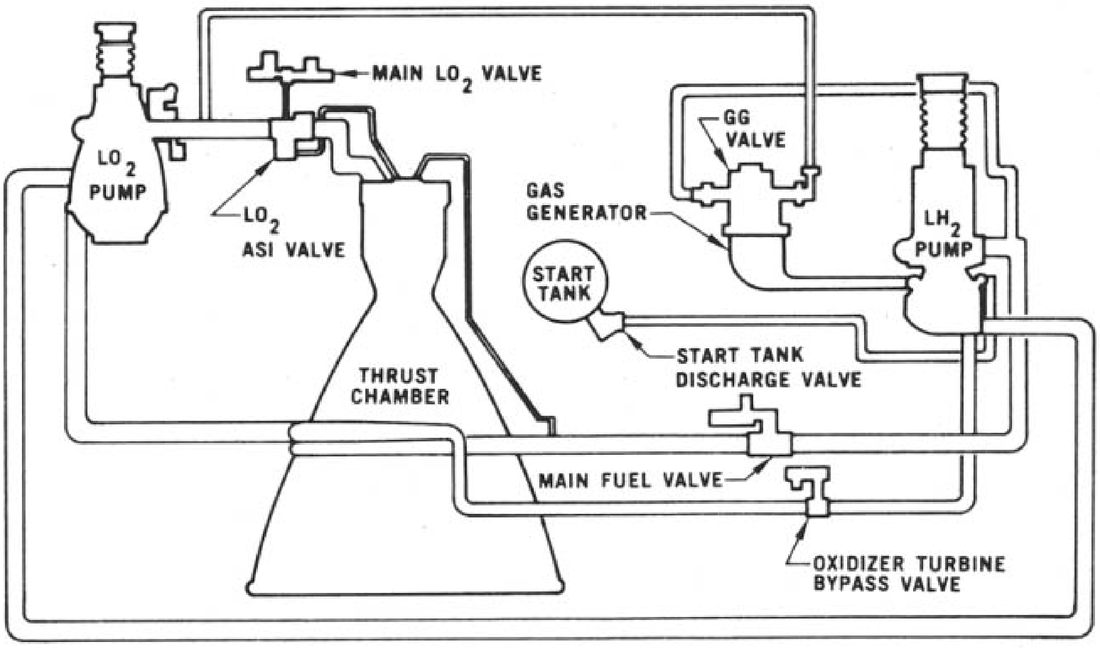


Figure 40: J-2 operating schematic [18].

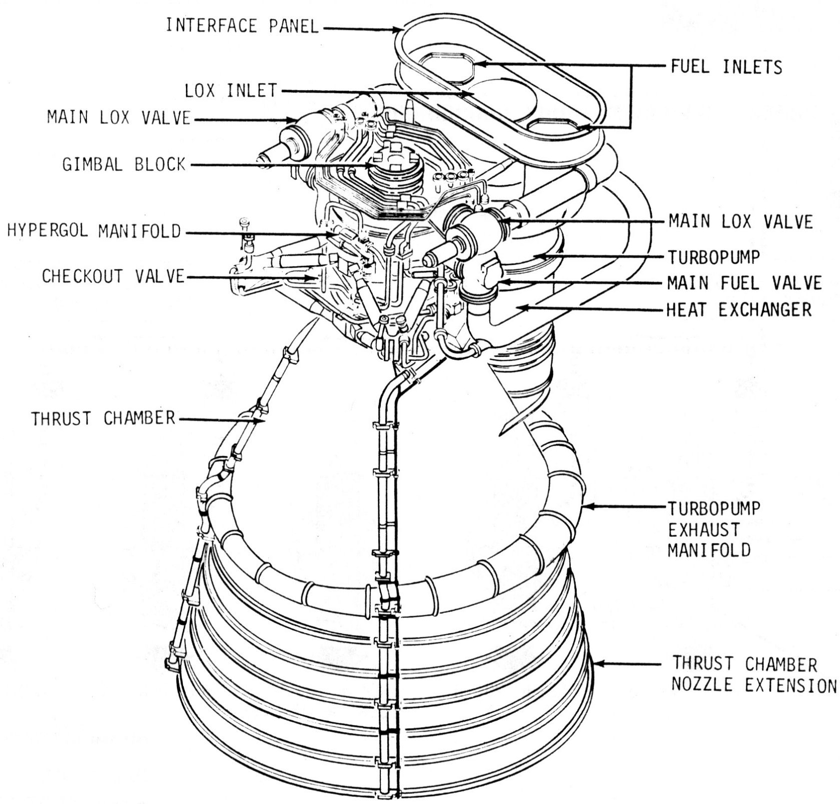


Figure 41: F-1 operation schematic [155].

The last engine that is used in this experiment is the RS-68. Rocketdyne began development of this engine in 1997, and it was certified for use on the Delta IV launch vehicle in 2001 [154]. The gas generator cycle, which is still in production today, uses a liquid oxygen/liquid hydrogen propellant architecture that is based on conceptual design studies of the NASA Space Transportation Main Engine (STME) study. The focus of that program was to develop a new liquid rocket engine using a cost-as-the-independent variable approach, but canceled in 1994. To reduce the total development cost, the primary intent was to simplify the design by using fewer unique components and reducing the overall parts count. The final design included 80% fewer parts than that SSME, and was produced with 92% less touch labor [154]. The result of this design was lower risk and higher reliability than a typical new engine development program. The RS-68 was certified with only 183 tests and 18,945 seconds of operation [154]. A simplified schematic of the RS-68 operations is shown in Figure 42. The primary components of the engine are two turbopumps, LOX and LH2, gas generator, LOX tank pressurization system, combustion chamber, nozzle, and flow control valves [154].

3.3.2 Experiment 2 Results

Figure 43 shows the results of the Hall model without including test fidelity levels. From this plot, the same over prediction of reliability that was seen in Experiment 1a is evident. When all tests are assumed to have an equal probability of uncovering all of the failure modes, they are uncovered more quickly than they occur in the SSME data. The total value of rework cycles that occur is in agreement with the actual value, 150-170. This implies that the failure mode probability of occurrence values are representative of the actual system.

Figure 44 shows a much better agreement for the number of rework cycles and

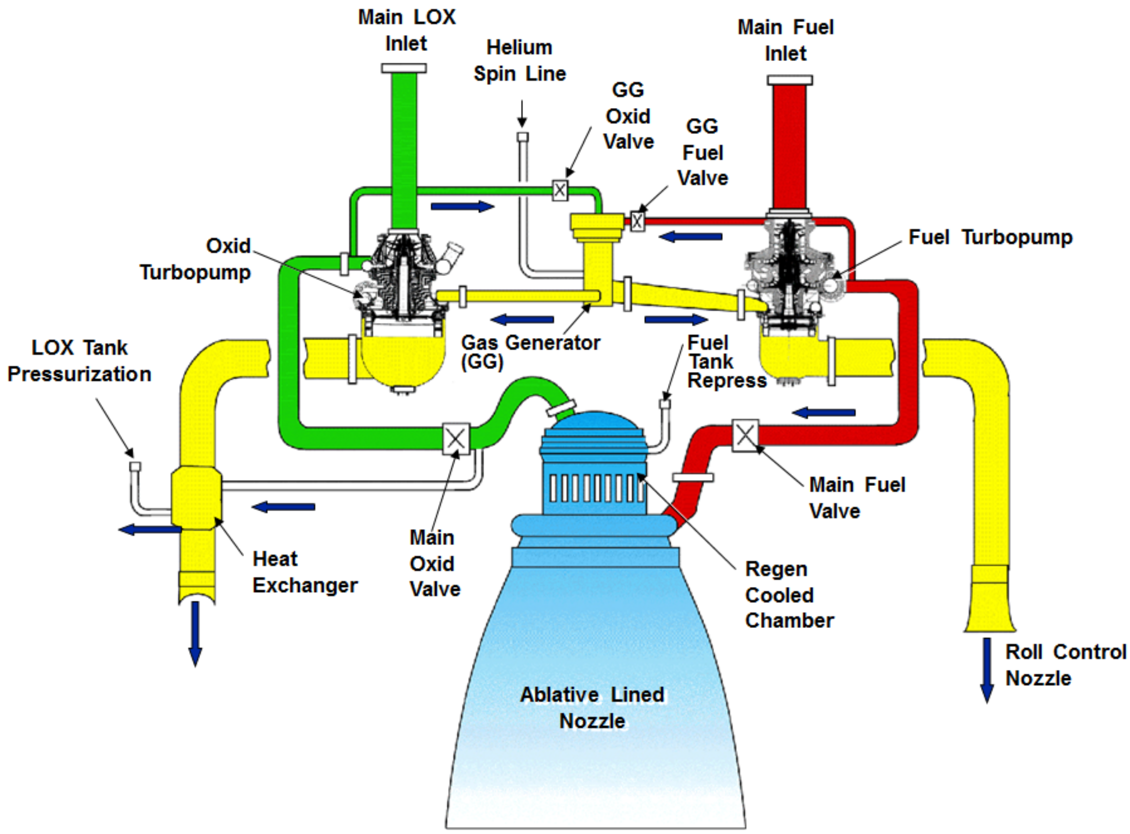


Figure 42: RS-68 operating schematic [154].

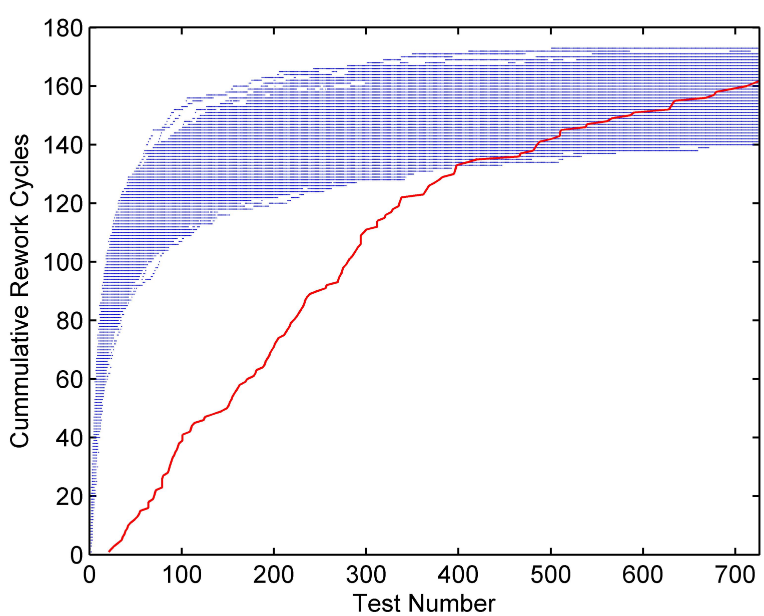


Figure 43: SSME rework cycle predictions with fidelity.

when they occur during development. The fidelity levels that were identified in Experiment 1b results in rework prediction that follows the actual engine data much more closely. The model does over predict slightly between tests 180-280. This could be due to the use of yearly average test duration as a measure of fidelity. The actual test durations could change throughout that year, but this model is not capturing that. Without being able to obtain the actual length of each individual test, it is not possible to adjust the fidelity of each test. However, the overall-growth trend for the failure modes of all criticality levels is captured well, providing more insight into testing activities than when each trial is considered equally. The total number of rework cycles that occur is also in agreement with the actual value. This shows that the test fidelity levels can be utilized to model failure modes of all criticality levels.

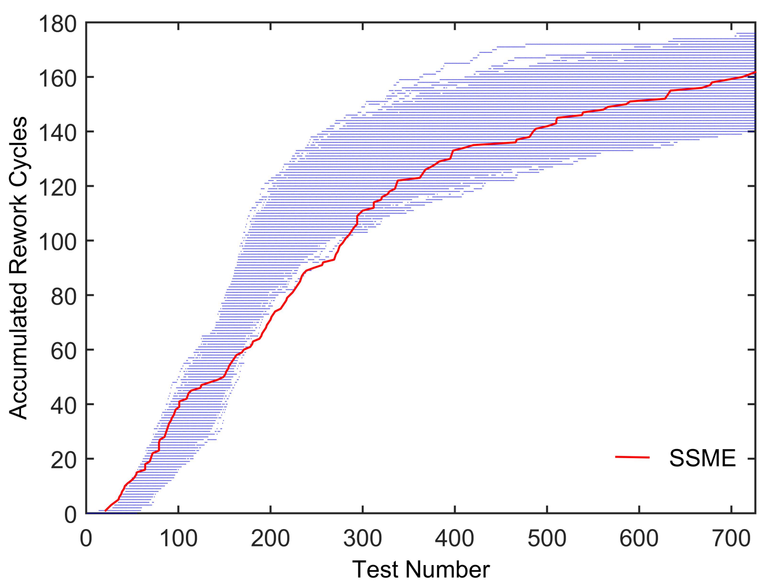


Figure 44: Hall model growth predictions using exponential fidelity values.

The last part of this experiment consists of identifying a traceable complexity metric that can be used to estimate the number of rework cycles when a comprehensive FMEA is not available. One possible option identified is the number of qualification tests defined in the RVM. Figure 45 illustrates the relationship between the number of qualification tests and number of rework cycles from the four historical engine programs discussed in Section 3.3.1: the SSME, RS-68, F-1, and J-2. This plot shows that there is a trend in number of rework cycles and qualification tests. While the RS-68 stands out, it is still within the general trend. One of the reasons the RS-68 had a reduced number of rework cycles during Phase C/D was the advance of computer analysis techniques that allowed the program to eliminate failure modes before hardware production was initiated [154]. Because it is still in production, the failure modes and probabilities for that engine are not publicly available for review.

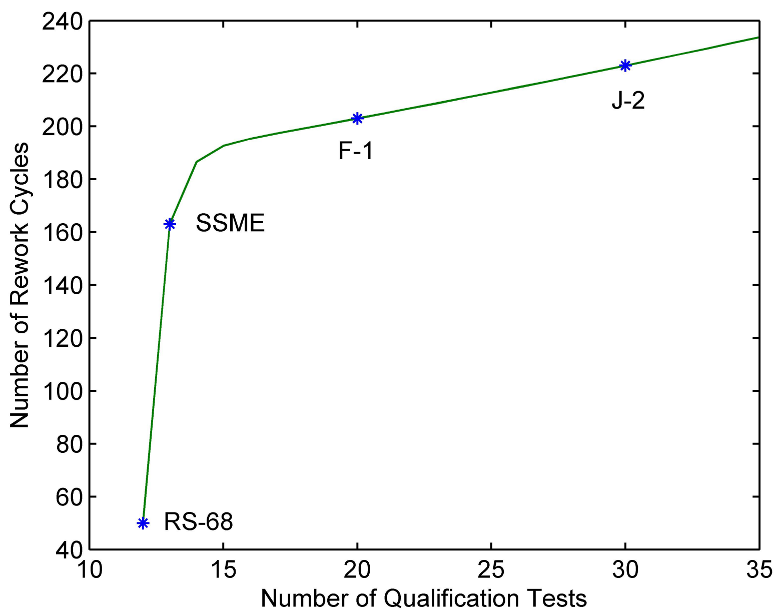


Figure 45: Number of qualification tests versus number of rework cycles for historical liquid rocket engines.

While there does appear to be a correlation between the number of qualification tests and the number of rework cycles that occur during engine development, this does not imply causation. With only four data points for comparison and no previously established link between rework and qualification tests, there is not evidence to substantiate the use of qualification tests can be used as a complexity metric. Based on this assessment, the previously identified Rocketdyne approach is used as a secondary method for estimating rework cycles if FMEA is unavailable. The Technical Uncertainty Factor (TUF), discussed in Section 2.9.2, is a measure of the design maturity, system complexity, technology level, and environment [64]. Rework cycles are caused by failure modes, both known and unknown, which directly relate to the system metrics used to quantify the TUF value.

3.3.3 Experiment 2 Summary

After considering the results from Experiment 2, it is apparent that FMEA data can provide quantitative rework probabilities that are more traceable than expert opinion or comparison to historical systems. It was also confirmed that the test fidelity levels are applicable when failure modes of all criticality levels are considered. When fidelity is not included in the Hall model, the reliability is over predicted. This can be seen by the number of rework cycles that are predicted to occur early on in development testing when compared to the actual SSME data. Using test fidelity levels to adapt the indicator function in Hall's model more closely follows the accumulated test cutoffs from the SSME program history. Due to this improved performance, FMEA data can be selected for use in this methodology as a method for estimating rework cycles and their probability of occurrence. The results of this experiment are in line with the original hypothesis, which identified FMEA as a method for providing quantitative rework probabilities. Therefore, hypothesis 6 is valid based on the results of Experiment 2.

An additional analysis was done to determine an appropriate complexity metric that can be used if FMEA data is unavailable. The number of qualification tests outlined in the program's RVM were shown to create a general trend when plotted against the number of rework cycles for four historical engine programs. While these two values are correlated, there is not enough evidence to support the use of qualification tests as a complexity metric for estimating rework cycles. The TUF metric defined in [64] will be used as a secondary approach if FMEA data are unavailable.

3.4 Experiment 3

The final experiment addresses three aspects of the RIVVTS methodology. The first is to test Hypothesis 3 that was introduced in Section 2.6.4, and is restated below. Experiment 3 will also provide support for the use of triangular distributions and improve the accuracy of activity relationships in the DES which were both identified in Conjecture 5, discussed in Section 2.8.3. The final goal of this experiment is to support the cost distribution assumption for the cost of rework as a function of test number that was discussed in Section 2.7. The two main tasks in Experiment 3 are:

1. To develop the DSM, which includes identifying the testing activities, defining their fidelities, cost and schedule distributions, and rework probabilities
2. To develop the discrete event simulation in order to propagate the uncertainty of the input variables to generate a distribution on the outputs.

> **Hypothesis: 3**
> If a DSM is adapted to explicitly account for the probability of internal rework, it will provide a stochastic and quantitative model of rework impacts that is more accurate for VVT processes than if internal rework is implicitly included in the activity duration distribution.

Conjecture: 5

- Using triangular input distributions, the assumptions required will be more traceable than if beta and Weibull distributions are used.

- If DES is used for simulation, the results will allow for quantitative comparisons between VVT strategies and account for stochasticity of rework cycles.

3.4.1 Experiment 3 Setup

The first step in Experiment 3 is to set up the DSM for the SSME VVT activities. A truncated version of this DSM is shown in Table 18. The first two rows are design and manufacturing. The rest of the rows are the 726 hot-fire engine tests. The off diagonal rows indicate the probability that when a failure mode occurs, if it will require redesign, rework or retest. These probabilities were determined based on the failure mode criticality levels in the SSME FMEA data during Experiment 2.

Redesign implies the failure was introduced during product design and the rework cycle will include a redesign, rework or re-manufacture, and retest. Rework implies a manufacturing defect of some kind and the rework cycle only requires rework and retest. There is also the possibility that the failure is unrelated to the test article, for example a test facilities failure, and the rework cycle will only require a retest. Between 7-8% of failure modes are assigned a risk factor around 0.250, meaning those failures would lead to loss of crew, loss of mission, or loss of engine. Failure modes with a risk factor of 0.100 or less do not cause damage and can be excluded from consideration. A risk factor between 0.100 and 0.111 means there is possible piece part damage and may or may not require rework. Consequently, a probability between 70-85% can be assumed for rework, which would include the failure modes with a risk factor below 0.250 and the upper half of the percentage of failure modes with a risk factor 0.111 and 0.123. This would mean the remaining 7-23% could be considered facility failures. An analysis of the SSME premature engine cutoffs found that 16% of the cutoffs were the fault of the test facility or controller [146]. The percentage of SSME facility or controller failures would fall directly into this range.

Table 17: Cumulative percentages of SSME FMEA risk factors [146].

Severity	Cumulative Percentage
1.000	1.5%
0.500	5.0%
0.333	7.0%
0.250	7.5%
0.200	10%
0.167	17.5%
0.143	43%
0.125	71%
0.111	96.5%
0.100	100%

Table 18: DSM for SSME activities

.

		1	2	3	4	...	728
Design	1			U(.07,.10)	U(.07,.10)		U(.07,.10)
Mfc.	2			U(.70,.85)	U(.70,.85)		U(.70,.85)
Test 1	3			U(.07,.23)			
Test 2	4				U(.07,.23)		
...						U(.07,.23)	
Test 726	728						U(.07,.23)

The next step is to define the cost, duration, and fidelity for each activity. The fidelities were determined in Experiment 1 based on the test fidelity levels. Cost and schedule distributions require a pessimistic, most likely, and optimistic estimate to generate the triangular distributions discussed in Section 2.8.1.2. The schedule durations in this experiment are defined by the actual test length plus the time between tests. The amount of time between tests is a number of days, which will dominate the actual test time. By looking at the number of tests per year in Table 8, a distribution can be determined. The first test was conducted in June 1975 and last test was in March of 1981, just prior to STS-1 on April 12, 1981 [17]. By dividing the number of tests per year by the number of days testing occurred during that year, a range of 3 to 6 days between tests is found. The tests were more often closer to 3 days apart than 6 days, so the triangular distribution can be set to $f(x;3,4,6)$ days. This time scale is similar to testing schedules for other engine development programs with at least two dedicated test facilities [154, 120].

Test costs can be estimated by considering the overall cost of the SSME development program and the percent of that cost that is associated with testing. The distribution of development costs for rocket engine programs by discipline is shown in Figure 46 [60]. The following costs are in 1996 dollars to enable comparison with reported development costs [64]. Design will include the cost of engineering and management, approximately 25% of development cost: $\$2.5B * 0.25 = \$625M$. Manufacturing cost is between 50% and 55%: $\$2.5B * [0.5, 0.55] = [1.25, 1.38]M\$$. Testing is approximately 20-25% of total development cost, giving the following range for testing costs:

$$\text{SSME Cost/Test} = (\$2.5B * [.20, .25])/726 = \$[0.69, 0.86]M. \tag{25}$$

The actual most likely value for testing costs of liquid rocket engines is not available. In lieu of this data, the median of the test cost range is chosen as the most likely estimate to complete the triangular distribution for testing cost. This results in an estimated test cost range of $f(x;0.69,0.77,0.86)\$M$.

The total cost of rework, or the 'test-fail-fix' portion of a development program has previously been reported as the total cost of testing, engineering, and hardware that is required during development. A further breakdown of cost distributions during development is shown in Table 19. The weakness in this approach, as applied to the RIVVTS methodology, is the limitation in evaluating alternative testing strategies. By grouping the engineering, hardware, and test costs associated with rework into a single value, the mitigating effects of alternative VVT strategies cannot be determined. Rocketdyne's Prodecol methodology, introduced in Section 2.9.2 used

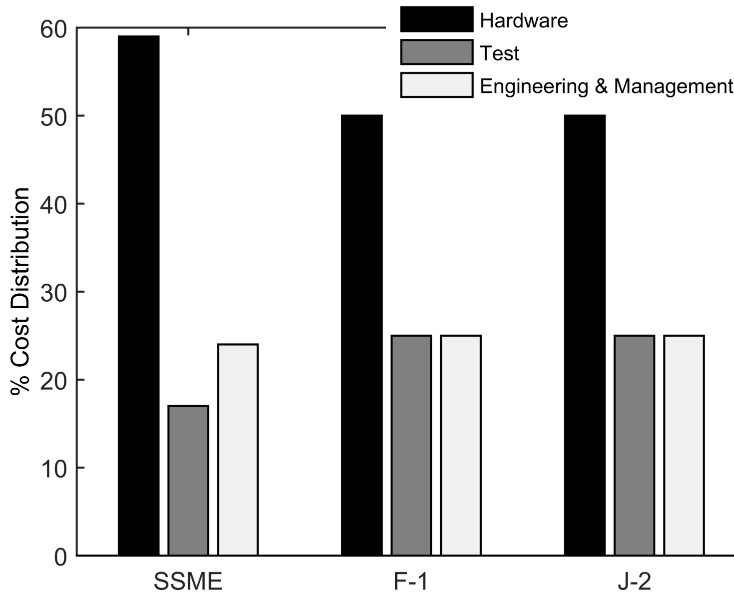


Figure 46: Historical development cost distributions recreated from [60].

this assumption in estimating rework cost [64] as follows:

$$C_{TotalRework} = \text{Total Development Cost} * 0.75 \div \text{Number of Rework Cycles}$$

\$ / Rework Cycle = (Total Development Cost * 75%) / Number of Rework Cycles

where 75% is a conservative estimate for the percent of rework cost. To allow the assessment of alternative VVT strategies, the cost of rework is divided into test costs and rework cost for this methodology. The average cost per rework is calculated with the same equation using 54% in place of the 75% used by Rocketdyne, based on the cost breakdown of engineering and hardware cost during the 'test-fail-fix' phase given in Table 19.

The average cost of rework using the previous formulation is \$8.2 M in 1996 dollars. To evaluate the change in rework cost throughout development testing, the F-1 cost per rework cycle curve is fit using an exponential equation: $a * exp^{b*x}$, where x is the percentage of tests completed, a is the fit coefficient that changes the mean value of the curve and b is the fit coefficient that changes the shape of the cost distribution

Table 19: Historical development cost breakdown [154]

	Conceptual Design %	Final Design %	Validation %	Fail-Fix &	
Engineering	2	15	1	7	25
Hardware			3	47	50
Test			9	19	25
	2	15	10	73	

over time. After establishing the fit from the data in Figure 20, the fit coefficient b is kept constant, while a is adjusted according to the estimated average rework cost. The indicator function determines when failure modes occur based on the probability of occurrence for each failure mode, and this curve determines the cost per rework cycles at that time.

$$C_{PerRework}(t) = 3.45 * \exp^{2.52t}$$

The triangular distribution for the duration of tests is determined from historical data of the SSME program illustrated in Figure 47. This figure shows the average number of days between tests during the SSME development testing program. The minimum value is approximately 3 days, the maximum value is 6 days, and the average of the seven points provided is 4 days. Based on these values, the triangular distribution for duration is $f(x; 3, 4, 6)$ days. This distribution is used to represent the duration for Case A, where internal rework is explicitly considered. Case B represents the duration with internal rework implicitly considered in the distribution. For Case B, the maximum estimated duration is 12 to represent this implicit possibility of rework.

Developing the discrete event simulation is the last primary task for this Experiment. Simio is used to develop the DES environment. Simio is a production planning and scheduling software that is well-suited to this problem. The Simio Standard Library contains the common objects that are required for a typical simulation, i.e. entities, resources, servers, nodes and connectors. The standard object behaviors are

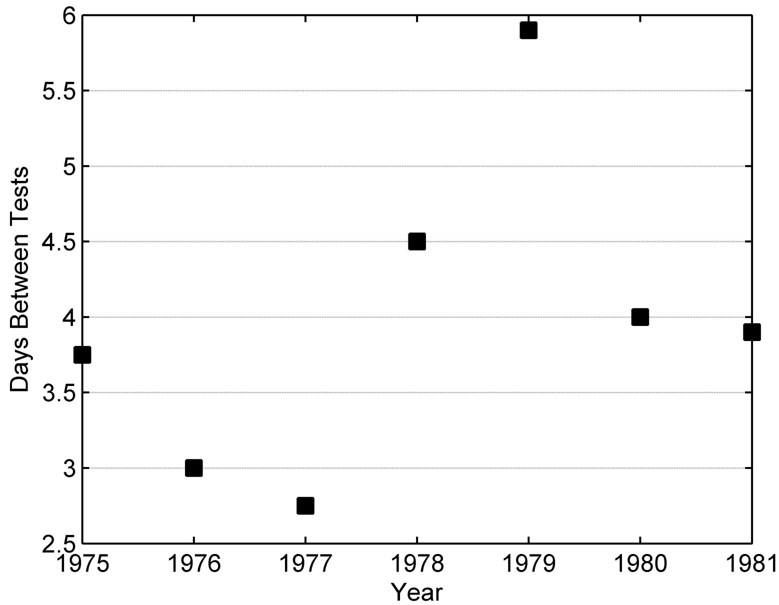


Figure 47: SSME average number of days between tests during development [17].

used in this model, but the option exists to augment objects as necessary. The process logic is developed specifically for this scenario and is outlined in Figure 48. During the simulation an internal clock is managed to track the passage of simulated time and resources used. Each case is replicated 1,000 times using the Simio experiment tools.

The DES inputs are the fully defined activities (i.e. fidelity, cost, and duration) and activity relationships (i.e. DSM). The DES logic is outlined in Figure 48. After the completion of a test, the indicator function is evaluated as described in Section 3.2.3. If a failure mode does not occur based on the failure mode probabilities defined in Section 3.3.2, there are no state changes in the DES and it continues onto the next activity. If a failure mode event is triggered, the DSM probabilities are used to determine the total impact of the failure mode. If the failure mode requires redesign, it is considered a criticality 1 failure mode that would lead to LOC or LOM and its impact is included in reliability calculation, cost and schedule. If the failure mode requires rework or retest, it is not included in the reliability model and will only impact cost and schedule.

To determine the more accurate way to represent the impact of internal rework, the DES is evaluated under two conditions. As mentioned in Section 2.6.3, when DSMs are used for product design processes, this internal rework is represented in the overall uncertainty for the cost and duration of that particular activity. For this methodology, if a test fails and requires retest, the entire test will need to be repeated as opposed to just a portion of a design process. An example of this would be a premature engine cutoff due to something unrelated to the actual test article, like a test facilities failure. For Experiment 3, the discrete event simulation is run with two different distributions on the activity durations. Only duration is changed because it is assumed that the results also apply to the cost distribution. The activity duration distribution in Case A is narrower and does not implicitly include the probability of rework. For this case, tests are repeated based on the separate probability of retest. The DES is run 1,000 times to determine an output distribution on the schedule. Then, the activity distributions are rest for Case B, with the possibility of internal rework included in the individual activity durations A separate probability for retesting is not included in the second case. The duration distributions and retest probabilities for each case are listed in Table 20.

Table 20: Internal rework assumptions for Experiment 3

Case	Duration Distribution (days)	Retest Probability
A	$f(x; 3, 4, 6)$	16%
B	$f(x; 3, 4, 12)$	N/A

3.4.2 Experiment 3 Results

Section 2.6.4 identified a DSM as an appropriate method to model activity relationships during VVT. Most often used for product design, the probability of internal rework is included in the actual activity cost and schedule duration. After evaluating

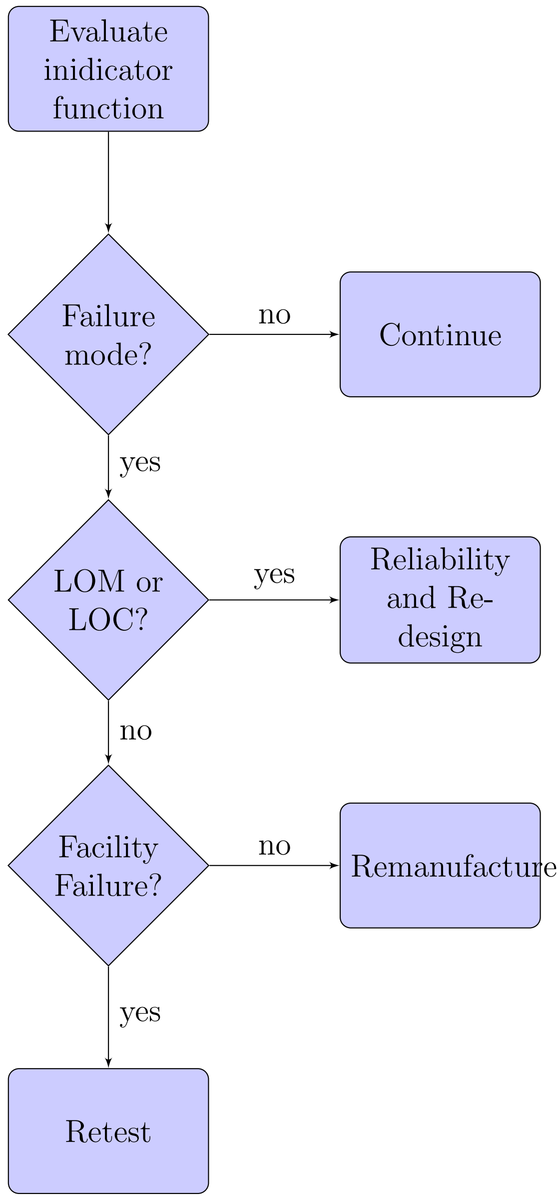


Figure 48: Discrete event simulation logic.

its use for VVT processes, it was suggested that internal rework be considered independently of the activity distributions to generate a more accurate cost and schedule model. A DES was developed to stochastically determine the impact of internal rework in both cases. The fidelity, cost, and duration for each activity and the activity relationships were defined in Section 3.4.1. 1,000 runs for each case were run, and the resulting mean and percentiles are presented here.

Figure 49 shows the results of Experiment 3, and the mean and variance for each case are listed in Table 21. In this plot, it is clear that including the probability of internal rework, or retest, decreases both the mean and the variance of the schedule estimation. The actual SSME development test program lasted 6 years, from the first ignition test in 1975 to the first Space Shuttle flight in 1981 [17]. By limiting the duration distributions in Case A, and determining retests by probability, the schedule estimate is more accurate and less uncertain. The mean schedule estimate for Case A is 6.25 years, and the variance is 8.24e-4 square years. The overly broad duration distribution for every activity in Case B, results in an over estimation of the schedule. The mean for Case A is 8.45 years, and the variance is 0.0051 square years.

Table 21: Schedule distributions with and without the probability of internal rework.

Case	**Mean**	**Variance**
A	6.25	8.24e-4
B	8.45	0.0051

The total rework cost and test cost results are plotted in Figures 50 and 51, respectively. Using the rework cost distribution assumed in Section 3.4.1 provides a mean total rework cost of $1,388 million in 1996 dollars. This value is 2.8% higher than the $1,350 million projection based on the historical distribution of rework costs over development phases. The mean total test cost is $561 million in 1996 dollars, which falls with in the estimated range of $500 to $625 million based on the same distributions. The test cost is expected to be accurate because it does not vary

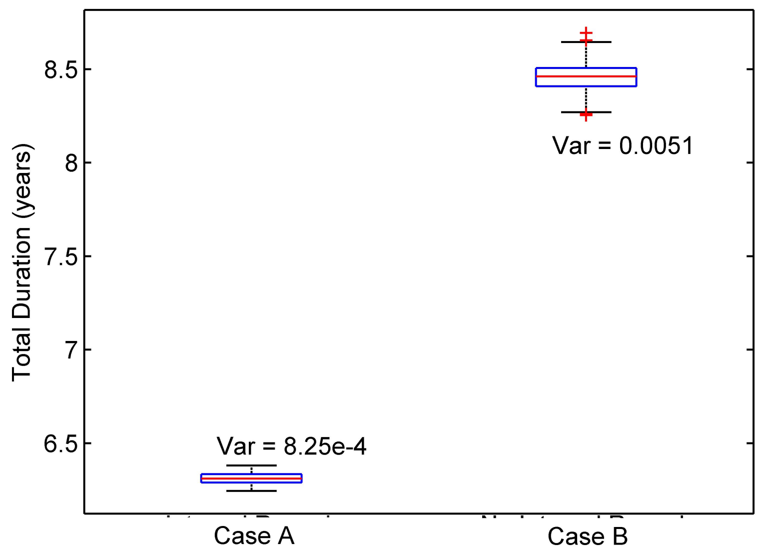


Figure 49: Schedule distributions with and without the probability of internal rework.

stochastically, and the input distributions are based on the historical distribution range. The result does provide verification that the test cost is modeled accurately in the discrete event simulation.

The result of the total rework cost is more significant because it does vary stochastically with the occurrence of rework cycles based on the failure mode probability of occurrence. Experiment 2 confirmed that using Hall's indicator function accurately predicted rework cycles when compared to the SSME test history. When used in conjunction with the assumed rework cost distribution, the resulting rework cost is within 3% of the estimated rework cost. This confirms the assumption of rework cost increase as a function of test number identified in Section 3.4.1.

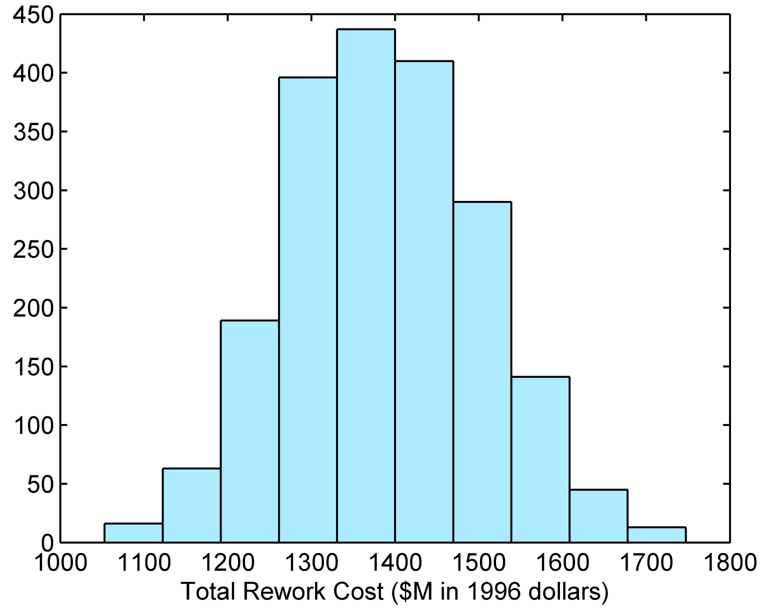


Figure 50: SSME total rework cost distribution in 1996 dollars.

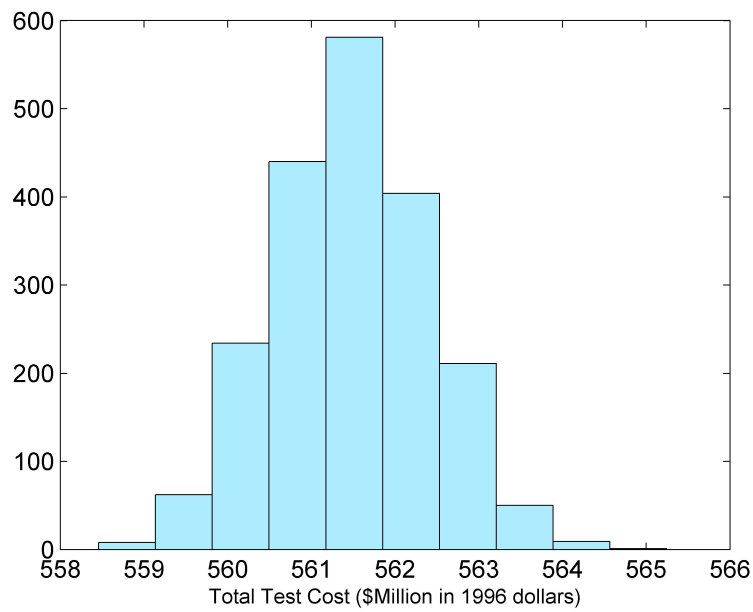


Figure 51: SSME total rework cost distribution in 1996 dollars.

The results of Experiment 3 were primarily intended to support hypothesis 3, but an observation about conjecture 5 and the assumption for rework cost distribution can also be made. The triangular distribution was chosen for the cost and duration of VVT activities because generating the assumptions for the distribution is more traceable than for a Weibull of Beta distribution. The importance of traceable and accurate assumptions is illustrated in this experiment. When an overly broad distribution is used to generate a schedule estimate, the additional uncertainty is carried over into the output distributions. It is necessary then, to ensure that reasonable assumptions can be made to determine inputs. When the distribution for an activity is unknown or too few data points exist to generate a statistically significant distribution, as is often the case, an expert can be consulted to provide an assessment. The triangular distribution requires a minimum, maximum, and most likely value. These are fairly reasonable estimates for a person to make because they are intuitive. A Beta distribution, on the other hand, requires two measures of central tendency, a mode and a mean, and two percentiles. Subjective assessments of these values make it harder to guarantee an accurate result because they are not as intuitive. In practice, many experts have a difficult time producing this information and their responses can vary widely. This experiment supports the use of a triangular distribution by illustrating the importance of using accurate input distributions.

3.4.3 Experiment 3 Summary

Experiment 3 tested the adaptation of a design structure matrix to explicitly include the probability of internal rework, referred to as retest. After defining the fidelity, cost, and duration of each activity, the activity relationships were represented with the DSM. A discrete event simulation was created to run the model for two cases. The first with narrower distributions and an explicit retest probability, and the second with a broad duration distribution on each activity and no probability of retest.

The result showed that the second case provided a more accurate and less uncertain schedule estimate than the first case. The inclusion of retest probabilities is a more realistic model of how testing activities would occur and gives more insight into the actual processes. This conclusion is in line with the original hypothesis, and this adaptation of the DSM to explicitly consider internal rework can be selected for use in this methodology.

3.5 Method Development Summary

The experiments presented in this chapter addressed specific components of RIVVTS methodology by testing the hypotheses formulated in Chapter 2. The first experiment compared the use of Hall's reliability growth model in four different set ups to the demonstrated SSME reliability growth during development testing.

1. No fidelity levels
2. Average fidelity levels
3. Linear fidelity levels
4. '1-Exp' fidelity levels

The results in Section 3.2.2 demonstrated the over prediction of reliability using Hall's model over testing phases when no fidelity levels were included. Section 3.2.5 showed a similar result with the average and linear fidelity definitions. The '1-Exp' fidelity level was able to predict reliability more accurately, which is attributed to its physical representation of failure mode occurrence. Ultimately, hypothesis 2a was accepted based on these results.

Experiment 2 tested the use of FMEA to provide traceable and accurate probabilities of rework cycles. Using the indicator function from Hall's methodology to track rework cycles during development testing proved to be an accurate approach when the fidelity levels identified in Experiment 1 were included. The model results closely followed the accumulated test cutoffs from the SSME program history. This result allowed hypothesis 6 to be accepted. A secondary approach to estimating the number of rework cycles was also considered, in case FMEA data were not available during VVT planning. The number of qualification tests were considered as a more traceable metric for complexity than the Rocketdyne TUF value. To determine the validity of this metric, the number of qualification tests for four liquid rocket engine test programs were plotted against the number of rework cycles those programs incurred. While the result was interesting, the relationship was not strong enough to support the use of qualification tests as a metric for program complexity. The TUF metric was, instead, chosen as a secondary approach for estimating rework cycles due to the established relationship between rework cycles and technical uncertainty of a program.

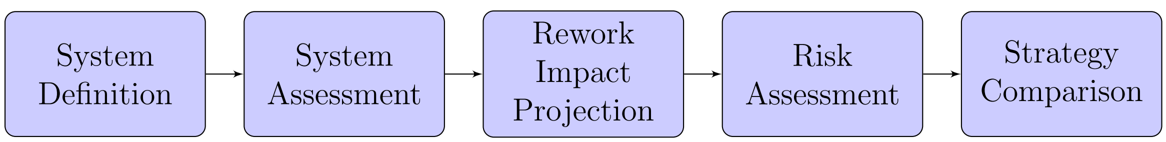
Experiment 3 tested the adaptation of a design structure matrix to explicitly include the probability of internal rework, referred to as retest. After defining the fidelity, cost, and duration of each activity, the activity relationships were represented with the DSM. The result confirmed that the explicit probability of internal rework provided a more accurate schedule estimate. When the probability of internal rework was implicitly included in the individual activity duration distribution, the schedule was over estimated. The result of this experiment allowed hypothesis 3 to be accepted.

The results of these three experiments, and the conjectures formulated in Chapter 2 are used to solidify the components of the RIVVTS methodology. Chapter 4 provides a detailed discussion of the complete method and describes how the components work together to evaluate the impact of rework on VVT. A case study is conducted to further validate the use of this method on a different liquid rocket engine, the RS-68.

CHAPTER IV

METHODOLOGY

Chapter 3 presented three experiments to define specific steps within this methodology. A combination of research questions and literature review presented in Chapter 2, along with observations from the three experiments in Chapter 3 help define the steps of the methodology presented herein. This chapter discusses in detail each of these steps and how they work together to help meet the overall research objective. This methodology is divided up into five primary elements discussed in more detail in the following sections.

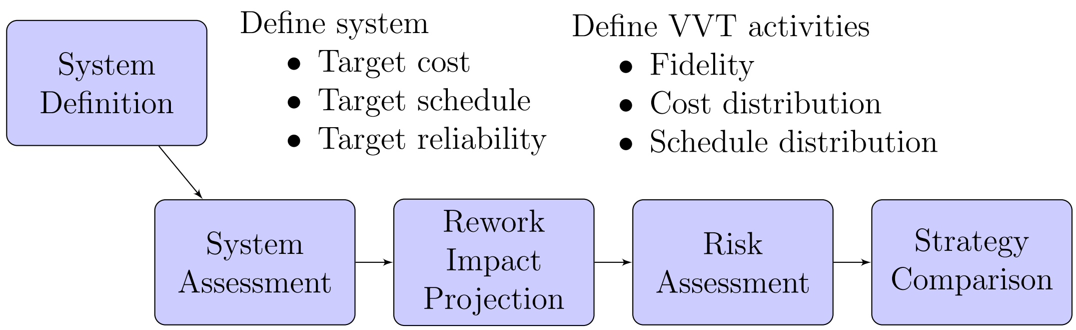


4.1 System Definition

The first element of this methodology is to define the system that is being analyzed. A system can be characterized at different levels. INCOSE defines these levels as a system, subsystem, assembly, subassembly, or component [153]. The levels create a hierarchical decomposition of the system from the detailed part level to the top system level. For this methodology, any of these levels can be included as long as the test fidelity can be accurately determined to that level of detail. The SSME used for comparison in the Chapter 3 is an example of a launch vehicle subsystem. The functional assemblies include the propellant feed assembly, and the pressurization assembly, among others [17]. The parts count method used to identify critical 1 failure modes (i.e. high and low pressure turbopumps) in Section 3.1.2 were examples of components. The analyst must define the system and functionally decompose it to the appropriate level, which can be determined by the VVT activities being considered.

Based on the system selected, a list of VVT activities need to be identified and defined. The verification requirements matrix and designed test plan for that system should be used to identify the baseline set of activities. To understand the impact of rework on different test plans, alternative sets of VVT activities should be selected for comparison. The next step is to define each activity being considered. For this methodology, a fully defined activity requires a fidelity level based on the estimated percent of failure modes that can be uncovered during that specific activity, and a pessimistic, optimistic, and most likely estimate for cost and schedule to generate triangular distributions.

The most reliable way to assess fidelity levels is by previous failure mode occurrence experience of the system. For new systems, the previous failure mode occurrences of similar systems can be used and adjusted based on an analogy approach. If a similar system does not exist, expert opinion or a physical decomposition of the system could be used. An example would be defining the fidelity based on the number of individual components that are used during a given test — either a percentage of the components, or using the individual component failure contributions if they are known. The last approach is utilized in the sample problem presented in this chapter. Cost and schedule distributions can also be elicited from experts, comparison to historical programs, or can be generated using industry standard tools like NAFCOM or SEER.



4.2 System Assessment

After the VVT activities are defined, the analyst must assess the system to generate the data required to project the impact of rework. The system data are used to define the VVT activity relationships and to estimate the number of rework cycles that will occur during development.

Two sets of probabilities are required to define the system. The first set is the probability of failure mode occurrences required for the indicator function and reliability growth model, shown below:

$$p_i = p_1, ..., p_k \text{ for } k \text{ failure modes} \tag{26}$$

The second set of probabilities are conditional probabilities that determine the type of rework cycle given that a failure mode has occurred. These probabilities, listed below, are used in the DSM to represent the relationship between testing and development activities — design and manufacture.

$$\begin{aligned} P(redesign) &= P(c=1) &&= p_{c1} \\ P(remanufacture) &= P(c=2) &&= p_{c2} \\ P(retest) &= P(c=3) \text{ or else} &&= p_{c3} \end{aligned} \tag{27}$$

Finally, the probability for a specific rework cycle is given by the following equation:

$$P(RWC_c) = (p_i)(p_c) \tag{28}$$

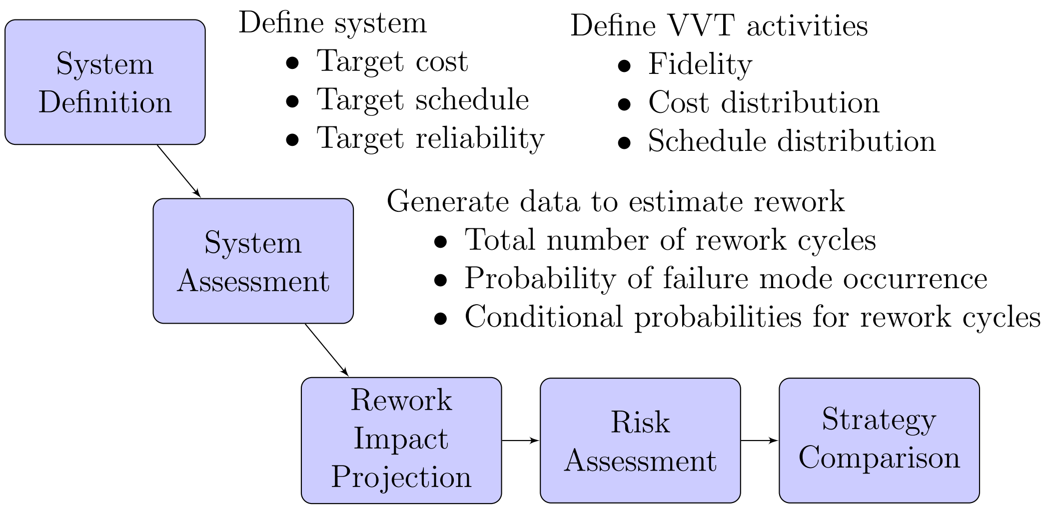
where p_i is the probability of occurrence for failure mode i, and p_c is the conditional probability of redesign (p_{c1}), remanufacture (p_{c2}), or retest (p_{c3}) given that a failure mode has occurred.

The method to generate this data depends on the design heritage of the system and its elements. Systems with extensive flight history and derivative systems, for example, could utilize prior development, test, and production experience. FMEA data are ideal for estimating the number of rework cycles and the activity relationships. A complete FMEA can also provide the probability of occurrence ($p_1, ..., p_k$ for k rework cycles) assumptions for the reliability growth model used in this methodology.

If FMEA does not exist, two methods can be used in conjunction to generate the required data. The failure modes that directly lead to loss of crew, loss of mission, or loss of vehicle, can be assessed using the part count method based on the physical decomposition of the system. The total number of failure modes can then be estimated based on the Technical Uncertainty Factor (TUF), which was identified in Section 3.3.2 as a secondary approach should FMEA data be unavailable. The TUF assessment of a system was shown to be a good indicator for the number of failure modes of all criticality levels. The probability of occurrence assumptions can be determined based on previous test data or can be developed based on information in the literature. An example of this is the derivation of failure probabilities used in Experiment 1.

The activity relationships used in the DSM can be derived using the number of critical 1 failure modes and the total number of failure modes. If the FMEA is available, the percentage of critical 1 failure modes can be used as the probability of redesign. The percentage of the remaining failure modes, not including the ones that do not cause any damage, gives the probability of rework, and the remaining percentage is assigned to retest. In the absence of FMEA data, dividing the number of critical 1 failure modes by the number of total failure modes estimated using the

TUF metric can be used for the probability of redesign. The probability of retest can be determined based on previous systems' testing failures or expert opinion, and the remaining failures will result in rework. Because the actual distributions of redesign, rework, and retest are unknown, these probabilities should be expressed as a uniform distribution with a max and min, $U(min, max)$.



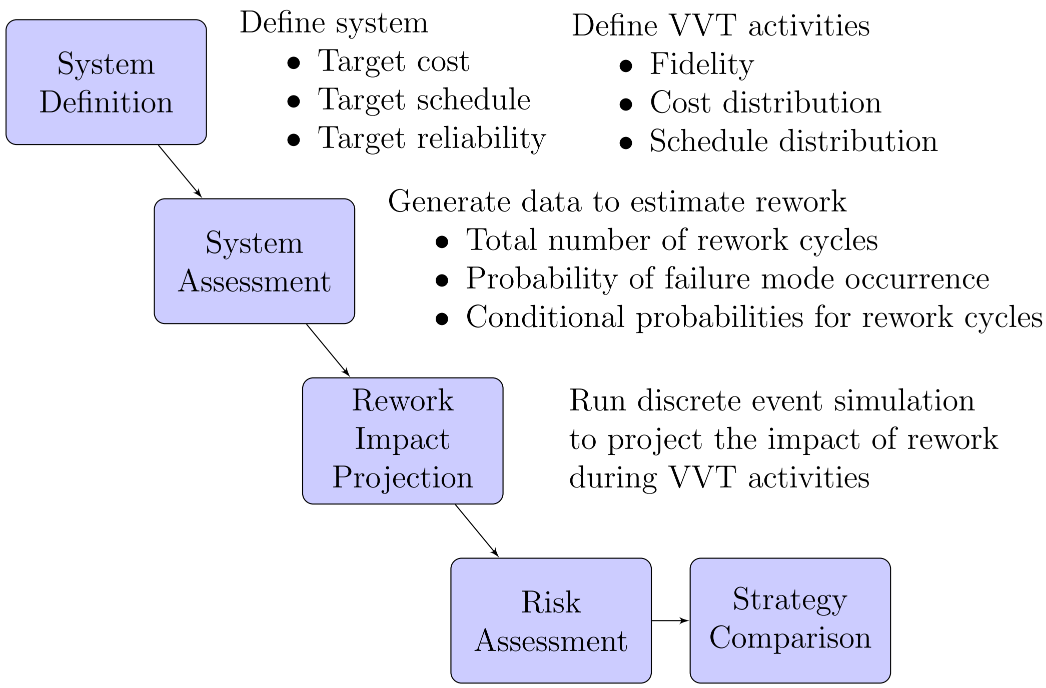
4.3 Rework Impact Projection

The next step in this methodology is to run a discrete event simulation to assess the impact of rework on reliability, cost, and schedule projections. At the beginning of each case, a random number is drawn from the Beta distributions for the probability of occurrence for each rework cycle. At each step in the DSM, the state of the system is evaluated. The indicator function, given in Equation 11, is used to determine if a failure mode occurs during the current state using the array of probabilities generated at the beginning of the case. If a Failure Mode Event is flagged, based on the probability of occurrence and current fidelity level, then the probability of redesign, rework, or retest is assessed using a random draw from the uniform distributions.

The system evaluates the column of the current activity from row 1 in the DSM. The row at which the rework cycle occurs determines how far back the system must go to correct that failure mode. If the failure mode requires redesign, based on the DSM probabilities, a Reliability Event is flagged. This event is used to track the number of critical 1 failure modes, their probability of occurrence and the first test in which they occur for calculating reliability. When the rework cycle is completed, the system flags a return event to continue onto the next VVT activity in the DSM. The cumulative cost and schedule are tracked during the simulation. When the model is evaluating an activity, a random number for cost and duration is drawn from the respective distributions. The cost of a rework cycle is dependent upon how far along the system is in the overall VVT plan. The assumption for cost distribution of rework cycles during development is described in Section 2.7.

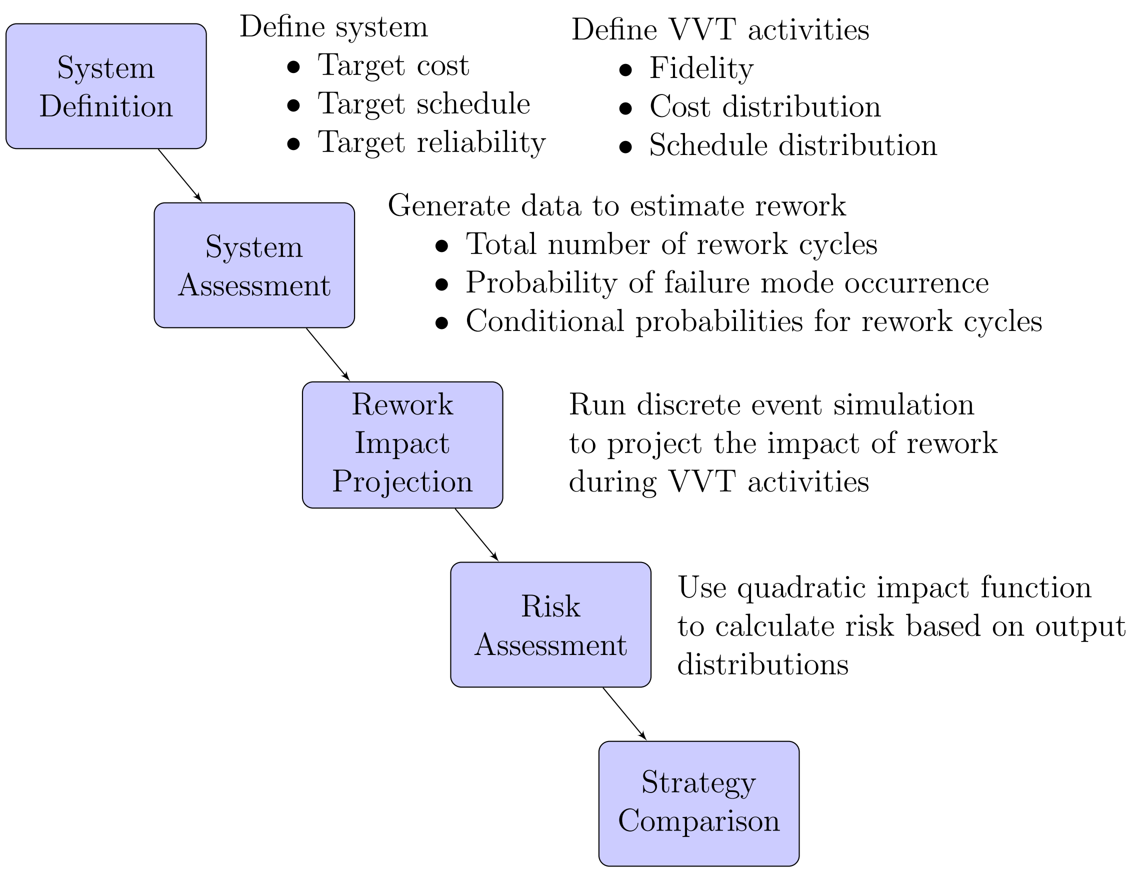
At the end of each case, the adapted Hall growth model is called. This is done at the end of the simulation because the number of critical 1 failure modes that occur will not be determined until the end of the simulation due to the probability of redesign. Each critical 1 failure mode that was flagged during the simulation has two properties: the probability of occurrence that was drawn randomly from the Beta distribution and the test number where it occurred for the first time. The reliability model inputs include the number of primary failure modes, their associated properties, the number of tests, and the fidelity of each test. The reliability growth model is evaluated at each test, where the indicator function for each failure mode is assigned $I = 1$ if the failure mode occurred before that test, and $I = 0$ if it did not. The fix effectiveness factor is determined by randomly drawing a number from its uniform distribution.

The result of one case is a reliability growth curve, cumulative rework cycles, cost, and schedule estimate. The simulation is repeated 1,000 times to generate distributions for each of these outputs. The number of repetitions is the number of random draws taken from the failure mode probability distributions, and the activity cost/duration distributions. This parameter will have an effect on the overall run time of the model and the granularity of the output. A large enough number must be used to ensure the resulting distributions are accurately represented.



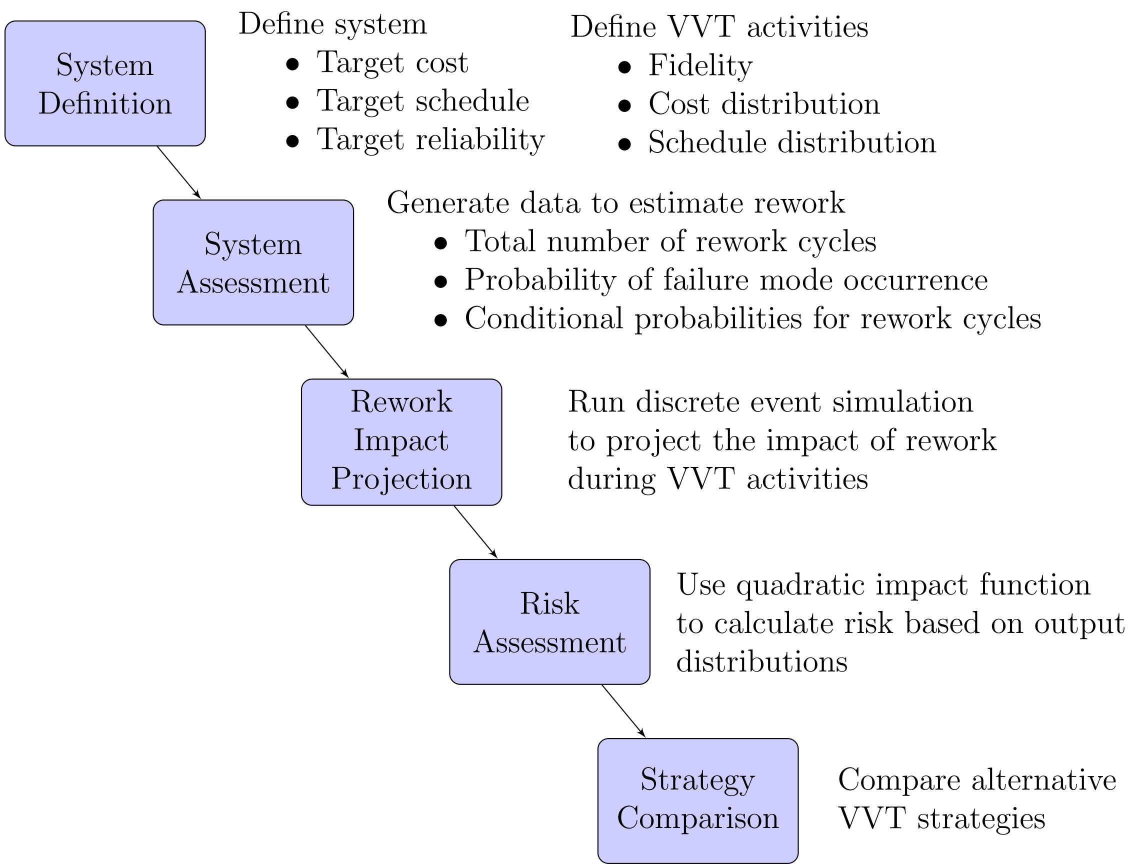
4.4 Risk Assessment

The next step in this methodology is to assess the risk for the selected VVT activities and to evaluate the testing strategy. The risk is assessed using the quadratic impact function chosen in Section 2.8.3. Risk assessment outputs a single value based on the simulation outputs for mature reliability, cost, and schedule. In the rework impact projection step, the discrete event simulation generates distributions for these values that are used to calculate the risk that these metrics do not meet the required baseline. The baseline for cost and schedule can be determined using NAFCOM or SEER — industry standard cost estimating tools. It is assumed that the baseline cost accounts for 2.5 years of engine-level testing at two dedicated testing facilities conducting 30 tests each per year. The cumulative distribution functions of the cost and schedule outputs from the simulation are used to determine the risk of exceeding that baseline. The mature reliability goal for the system is used as the reliability baseline. Unlike cost and schedule, the reliability risk is calculated based on the probability of not reaching the baseline, as opposed to exceeding it.



4.5 Strategy Comparison

After the risk has been assessed for a given VVT strategy, a number of alternative strategies can be evaluated for comparison. The primary research objective for this methodology is to produce a quantitative estimate of the impact of rework cycles on alternative VVT strategies to assist in the decision making process. The rework impact projection and risk assessment steps provide metrics that allow alternative VVT strategies to be compared to one another. The following section discusses the implementation of this methodology in order to demonstrate the approach on an actual system. Chapter 5 will present a full application of this method to further demonstrate the attributes and trends that can be utilized by the decision maker.



4.6 Case Study

The method is applied to the RS-68 liquid rocket engine, which was introduced in Section 3.3.1. This example serves to validate the capability of this methodology to accurately predict the impact of rework cycles on VVT activities. The RS-68 was chosen because it was developed relatively recently, but has a complete development testing history that can be used for comparison.

4.6.1 System Definition

The first step of this methodology is to define the system. A brief description of the RS-68 liquid rocket engine was given in Section 3.3.1. Certified in 2001, the RS-68 is used on the Delta IV launch vehicle and is still in production today [154]. The engine burns liquid hydrogen and liquid oxygen in a gas generator cycle that can oscillate between power settings of 101% or 58% during different mission profiles. The turbopumps are driven by a single shaft with direct drive turbines. The turbines are powered in parallel by high-pressure hot gases from the gas generator. The thrust chamber assembly consists of a combustion chamber and ablative nozzle designed to dissipate heat as the engine is running. Thrust vector and roll control is performed by gimbaling the thrust chamber assembly and the fuel turbine exhaust roll control nozzle [154]. The engine's operating characteristics are listed in Table 22 and the schematic is shown in Figure 42.

Table 22: RS-68 Operating Characteristics [154].

	Full Power	Min Power
Thrust, vac (KN)	3,341	1,922
Thrust, s/l (KN)	2,918	1,499
Chamber pressure (MPa)	9.79	5.62
Propellants	LOX/LH2	
Engine mixture ratio	6.0	
I_{sp} vac (sec)	409	
I_{sp} s/l (sec)	357	

The development program was 'designed to cost' in an attempt to reduce the non-recurring costs associated with the typical rework cycles seen in other engine programs. A concentrated effort was made to reduce risk prior to engine-level testing. The testing began with 71 component-level tests that progressively increased in fidelity by adding components. This incremental testing approach started with gas generator component testing, then advanced to the turbopump assembly, and finally powerpack (gas generator and turbupump subassembly) testing. The number of tests completed at each stage are shown in Table 23. These are the VVT activities that are used for this example problem.

Table 23: RS-68 development test program [154].

Test Level		Number of Tests
Component	Gas Generator	62
Component	Turbopump	11
Component	Power Pack	6
Prototype Engine		7
Engine		176

After identifying the activities that have been selected to verify and validate the design, each activity needs to be further defined. This can be done on an individual basis, or in groups if they can be categorized. Only one VVT strategy is analyzed to validate this model against the actual RS-68 test history.

The first 71 tests were component tests. To define the fidelity of those tests, the historical contribution of components to US liquid rocket propulsion failures is reviewed in Table 24 [89]. A significant percent of the failures, 34.2%, cannot be directly attributed to a single component. The largest component contributors to propulsion failures are the fuel feed and control subsystem, and the hydraulic/pneumatic control subsystem which contribute to around 15% each. The pressurization, electrical control, and oxidizer feed and control are the next largest, contributing around 10%. The remaining subsystems contribute less than 5% each. The components tested first can all be attributed to the fuel and oxidizer feed and control assemblies. The maximum fidelity for these components tests is then the sum of those two component contributions from Table 24, 22.5%.

The remaining 183 tests are engine-level tests. The fidelity can be determined based on the duration of the test. The percent of failure modes that can be uncovered as a function of operating duration could not be found in the literature. This is likely

Table 24: Component Contribution to US Liquid Rocket Propulsion Failure [89].

Subsystem	Percent Contribution
Fuel feed and control	15.0%
Oxidizer feed and control	7.5%
Combustion Chamber	4.2%
Nozzle	0.8%
Pressurization	10.0%
Lubrication	1.7%
Electrical Control	8.3 %
Hydraulic/pneumatic control	16.7%
Thrust vector control	0.8%
Engine structure	0.8%
Others	34.2%

because the engine is still in operation, and some data has not been released. Like the SSME, the J-2 engine shows a similar early growth in the number of failure modes that can be uncovered during the early operating environment, which was observed to accurately represent engine fidelity in Experiment 1 in Section 3.2.5. The failure experience for the J-2 is used to estimate the RS-68 test fidelities because they are both gas-generator cycle engines, while the SSME is a staged combustion engine. The failure experience for the J-2 given in [156] is plotted versus the actual engine test duration. To adapt this profile for the case study, the data are replotted versus normalized mission duration and scaled to the RS-68 mission duration of 250 seconds. The resulting RS-68 fidelity profile is provided in Figure 52. The RS-68 test groups and associated fidelities are listed in Table 25 [154, 156, 140].

The rework and test costs are combined for the case study to enable comparisons to the reported cost data in [154]. The Rocketdyne Prodecol method was applied to the RS-68 to estimate the number of rework cycles and the average cost per rework cycle prior to development. From this assessment, the rework cost curve fit discussed in Section 3.4.1 is adjusted to the estimated $5 million per rework cycle in 2001 year dollars. No distribution was provided to estimate testing costs. This does not affect the case study because only one testing strategy is evaluated, and the divided test

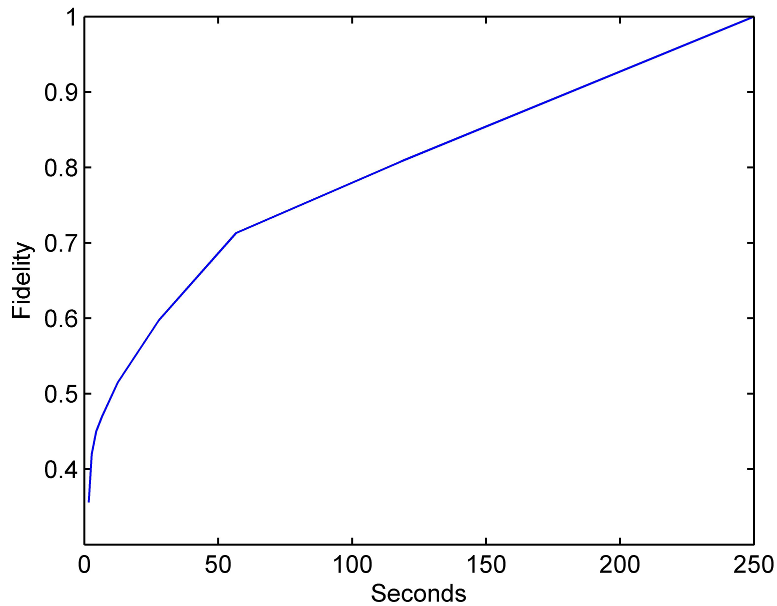


Figure 52: RS-68 fidelity profile versus test duration.

Table 25: RS-68 test durations and fidelity levels [154, 156, 140].

Test Level	Number of Tests	Average Duration (s)	Fidelity
Component	62		0.08
Component	11		0.15
Component	6		0.225
Engine	78	28	0.58
Engine	18	136	0.8
Engine	28	139	0.8
Engine	24	163	0.84
Engine	15	173	0.88
Engine	20	195	0.91

and rework costs would be summed for comparison regardless. The rework cost is calculated as follows:

$$C_{PerRework}(t) = 2.1 \exp 2.523t$$

The triangular distribution for the activity duration is determined using the publicly available data on the RS-68 engine development history time line illustrated in Figure 53. From this time line, the average number of days between system-level engine tests is 6 days, the least number of days between tests is 4 days, and the most days between tests is 9. The resulting distribution is $f(x; 4, 6, 9)$ days. The test schedule for component-level tests is approximately twice as frequent as engine-level tests, resulting in a distribution of $f(x; 2, 3, 4.5)$ days for component-level tests [154].

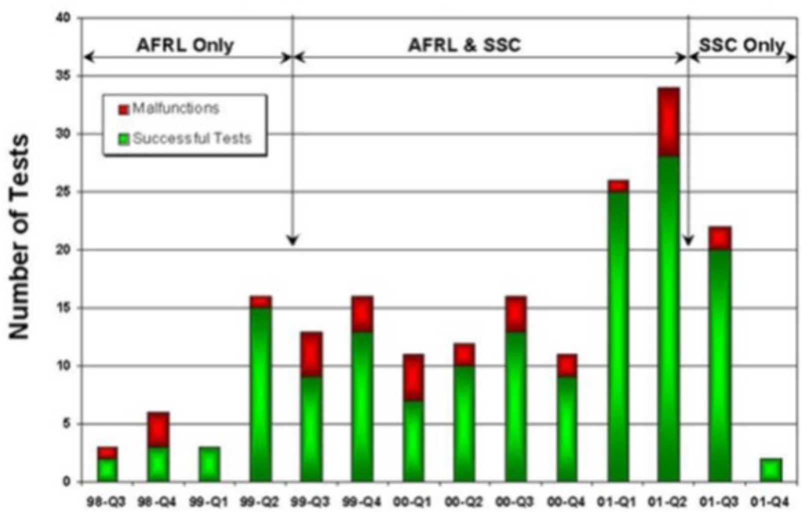


Figure 53: RS-68 engine development history [154].

4.6.2 System Assessment

The next step in this methodology is to assess the system and estimate the number of critical 1 failure modes, total number of rework cycles, and the activity relationships. As mentioned in the previous section, this engine is still in production and a comprehensive FMEA has not been released. The parts count approach is utilized to determine a range for the critical 1 failure modes. The primary components are:

- Gas generator
- Fuel turbopump
- Oxidizer turbopump
- LOX tank pressurization system
- Combustion Chamber
- Nozzle
- Heat Exchanger

Without the full FMEA, the relationship between the technical uncertainty factor assessment and number of rework cycles for historical engines can be used to determine the total number of rework cycles. The RS-68 was assessed using the TUF metric, and 30 rework cycles were estimated based on the risk level using the Rocketdyne Prodecol chart [154, 64]. Based on this assessment, the probability of redesign ($p_{c=1}$) can be determined: $7/30 = 23\%$. To account for uncertainty, a range of 20-25% for redesign is used. The historical range for test failures is used because no actual data is available. The SSME showed 16% of premature engine cutoffs were attributed to test facility failures, so a slightly lower range is used to the RS-68 because of the emphasis on reducing complexity and improving the test facility [154]. A range of 5-10% is used for this example, and the remaining percentages are the probability of rework, 65-75%. The RS-68 test program had 32 test malfunctions, and only 20 engine failures [154]. The other 12 malfunctions are attributed to facility failures, approximately 7% of the 183 tests. This is consistent with the 5-10% used for the case study.

The other inputs required for the reliability growth model are the probability of failure occurrence and the fix effectiveness factor. Again, without a comprehensive

FMEA, these values have to be based on similar systems or found in the literature. Woods assumed a mature reliability of 0.9982 based on a system comparison to the SSME [154]. The assessment was based on total parts count, design complexity, fabrication processes, and operating environment. This is slightly higher than the SSME reliability at first flight of 0.9952 [17]. The reliability at first flight is close enough that the same system-level probability of occurrence used for the SSME in Experiment 1 can be used for the RS-68, $Beta(0.22, 8.75)$. This can be verified by considering the test and failure history of the RS-68. Summing the fidelity levels defined in Figure 52 gives a weighted number of equivalent flights. For the 254 tests at the fidelity levels listed in Table 25, the equivalent number of flights is 146. The mean demonstrated probability of failure for the system is calculated as follows: $numberoffailuremodes/equivalentflights = 26/146 = 0.1844$. Assuming the individual failure mode probability of occurrences are randomly selected from the beta distribution in Table 26, the reliability of the system is calculated using the following equation:

$$R_{system} = (1 - p_{system}) = (1 - (p_1 + ... + p_k)) \tag{29}$$

where p_{system} is the probability of failure at the system level, p_i is the probability of failure for each critical 1 failure mode k [158]. While the summation of these probabilities isn't strictly correct, it is a 'rare event' approximation that be made for these independent top level failure modes. The number of failure modes ranges from 5 to 7 based on the probability of redesign determined during system assessment. To estimate the system-level probability of failure, first an integer is randomly selected from a uniform distribution of primary failure modes, k, then k random probabilities are drawn from the beta distribution and summed. This is repeated 10,000 to generate a distribution for the system-level probability of failure. The resulting mean of this exercise is 0.1772, which is 4% less than the demonstrated reliability of the system during development testing. This difference is negligible, and the assumed

$beta(0.22.8.75)$ distribution can be accepted.

The last input assumption that needs to be determined for the reliability projection is Hall's Fix Effectiveness Factor. Expert opinion was identified as the ideal way to estimate this value in Section 2.5.2.6. In Experiment 1 a uniform distribution between 90-95% was used based on the increased focus to eliminate defects during development testing. This value is kept for the RS-68 because the same focus on reliability can be assumed for its development program. In addition, because this value is difficult to estimate, keeping it the same will reduce any bias in the results.

Table 26: RS-68 reliability growth assumptions.

Parameter	**Value**
Number of Rework Cycles	25-35
Probability of Occurrence	β(0.22,8.75)
FEF	90%-95%
Number of Tests	254

The RS-68 engine and test program are fully defined in Sections 4.6.1 and 4.6.2. The next step in this methodology is to use the information gathered in the first two steps and determine the impact of rework cycles on the reliability, cost and schedule for the program. Once output distributions for these metrics are generated using the techniques defined in Chapter 3, the final step is a risk assessment to determine the program risk with respect to the target goals.

4.6.3 Rework Impact Projection

The discrete event simulation model inputs are defined during the system definition and assessment. The DSM represents how the model entity traverses the simulation according to the flow chart illustrated in Figure 48. The cost, schedule, and reliability input distributions are used to evaluate the system at each state during the simulation. The events are test failure, redesign, remanufacture, and retest. The probability of the test failure event being triggered is determined by the Hall indicator function and the fidelity level of the current activity state. If the event is triggered, then the DSM probabilities determine which level of fault correction will occur: redesign, remanufacture, or retest. After the fault correction is complete, the system continues on to the next activity and the state is evaluated again.

The DES was run 10,000 times to generate distributions on the model outputs. These distributions enable the calculation of output means, percentiles, and risk levels that are assessed in the next section. Figure 54 shows the results of the cumulative number of rework cycles predicted by the model during developmental testing compared to the actual premature cutoffs due to engine anomalies [154, 138]. The actual RS-68 data do not begin until engine-level testing, but the number of failures that occurred during component testing are reported. The red line represents the actual data that start after the 6 failures that occurred during the 71 component-level tests. The model follows the actual RS-68 engine cutoffs fairly closely. The component tests capture as many as 8 failure modes, which is more than the actual component tests uncovered, but this can be attributed to the range of rework cycles used as input. A range of rework cycles is used to model the uncertainty in rework prediction. The simulation gave a final number of rework cycles ranging from 20-35. The number of rework cycles that occurred, 26, falls directly in this range. This indicates that the model was able to capture the actual cumulative rework cycles for the RS-68 engine using test fidelity levels.

Figure 55 illustrates the results of the reliability prediction from the DES, which include the mean, 5th percentile, and 95th percentile. No RS-68 demonstrated reliability data has been released except for the mature reliability assessment of 0.9982 [154]. The resulting mature reliability prediction provided by the model is $P_5 = 0.9685$, $P_{50} = 0.991$,and $P_{95} = 0.9991$. The 0.9982 reliability assessment falls within this range. Another verification of the model accuracy is the number of redesign or criticality 1 failure modes that are predicted. The parts count method determined 7

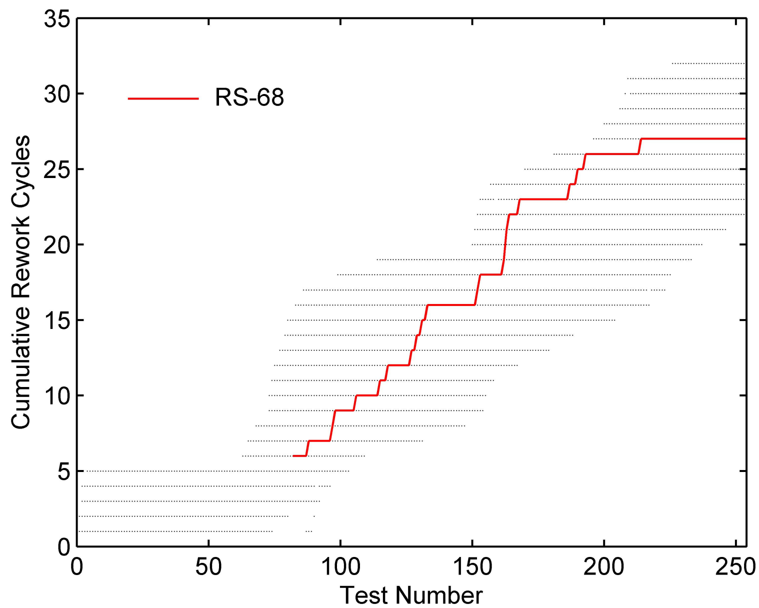


Figure 54: RS-68 cumulative rework cycles versus test number.

primary failure modes and the model shows an agreement with between 6-9 redesign cycles and an average of 6.9, or approximately 7. This suggests that the DSM percentages used to calculate reliability are accurately estimating the reliability growth during development testing for the RS-68 engine.

The final results from the DES are cost and schedule estimates. Figure 56 shows a histogram of the total rework costs. The rework costs from the F-1 engine were normalized to represent the rework cost as a function of the percent total development time. This exponential curve fit was then adjusted to scale for the RS-68 program costs of \$355 million in 2001 dollars [154]. The actual program reported a total of \$156 million for the fix-fail cycles. This is 1.3% lower than the average of \$158 million resulting from the simulation, which is negligible. This is the only cost value, other than total program cost, provided by the RS-68. This alignment of the cost prediction for rework cycles confirms the cost distribution used for rework cycles accurately predicts the total rework cost during development.

Figure 57 shows a histogram for the total testing schedule for the RS-68. Including

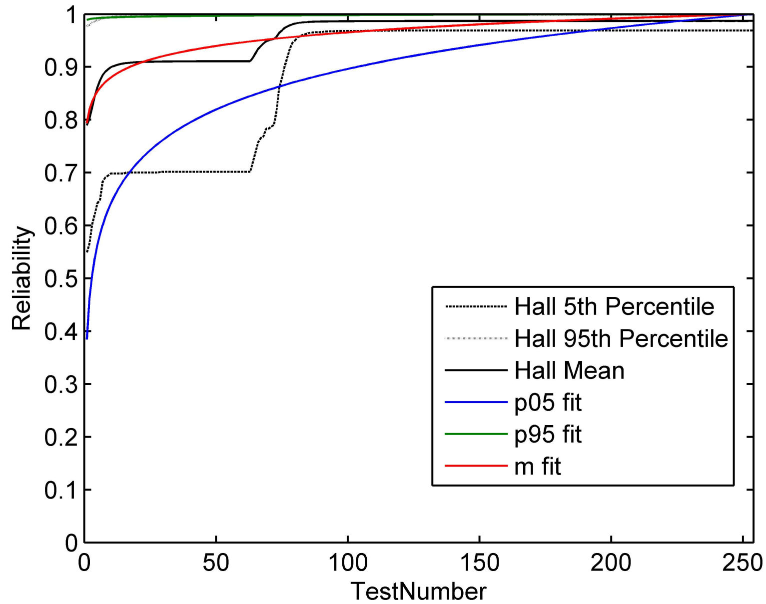


Figure 55: RS-68 reliability growth versus test number.

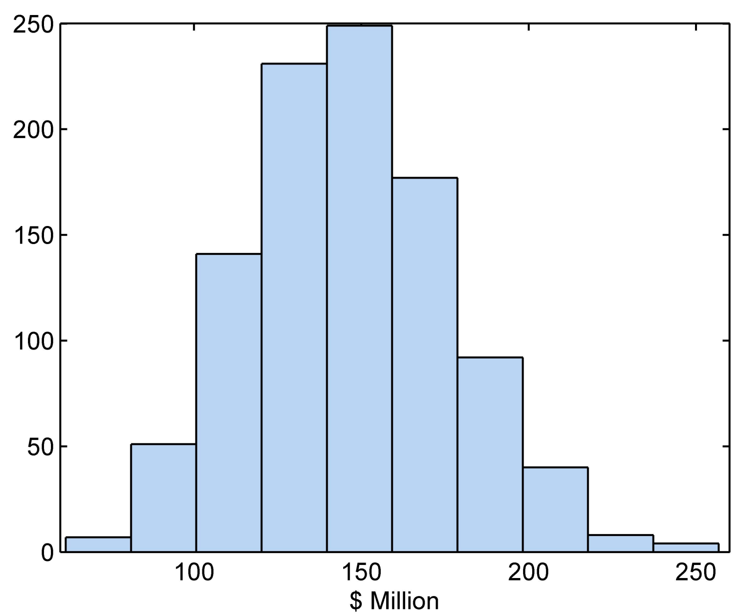


Figure 56: RS-68 total rework cost in 2001 dollars.

component testing, the engine was certified for first flight in 4.5 years. This is 2% lower than the average simulation schedule of 4.6 years. The result of the schedule duration including the probability of internal rework accurately predicts the total development time for this program.

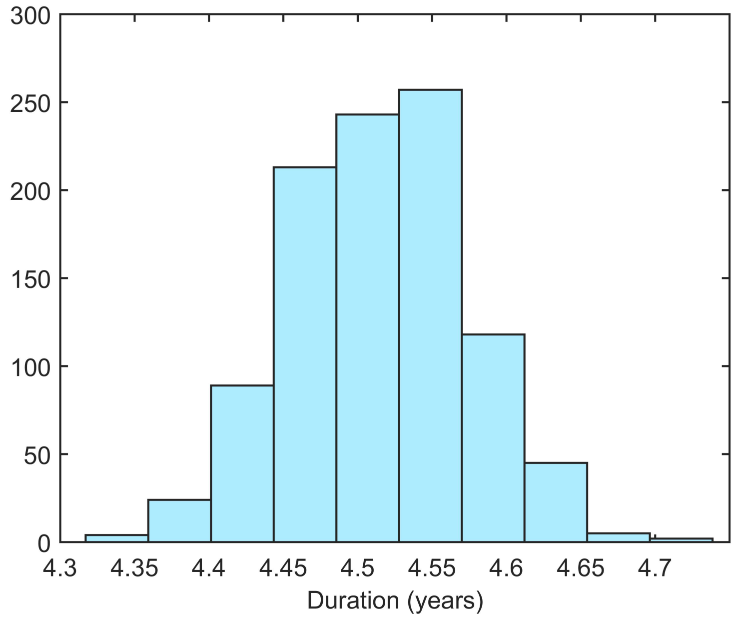


Figure 57: RS-68 schedule.

4.6.4 Risk Assessment and Strategy Comparison

The final steps in this methodology is to use the quadratic risk impact function to determine the overall risk in reliability, cost, and schedule for the RS-68 engine. The risk impact function is used to calculate the risk that a target value is not met and applying a quadratic risk growth as the estimate moves away from the target. For cost and schedule, the risk is going over the target values. The risk for reliability is to not meet the target value. Figures 58 to 60 illustrate the reliability, cost, and schedule risk for this program, respectively. On their own, the risk values do not hold much meaning. Their primary purpose is to allow comparisons of PDFs for different VVT strategies. They are shown here to illustrate the meaning behind the risk value.

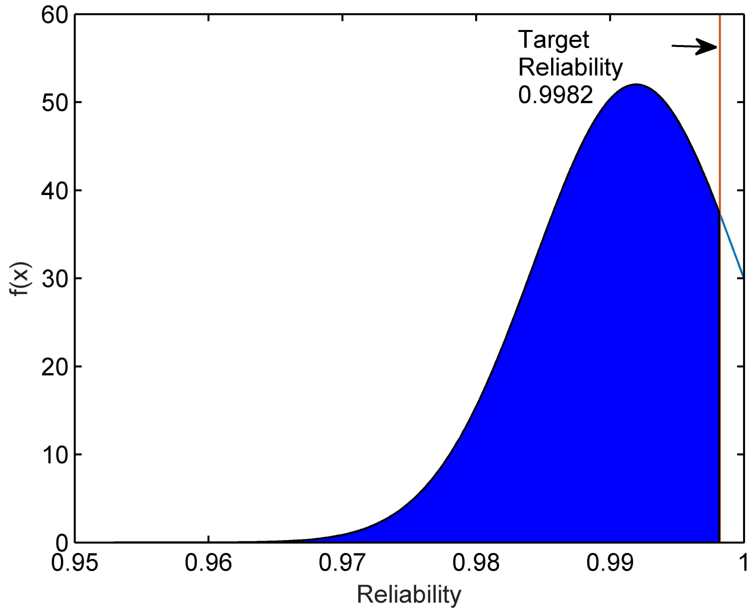


Figure 58: RS-68 reliability risk, R_R.

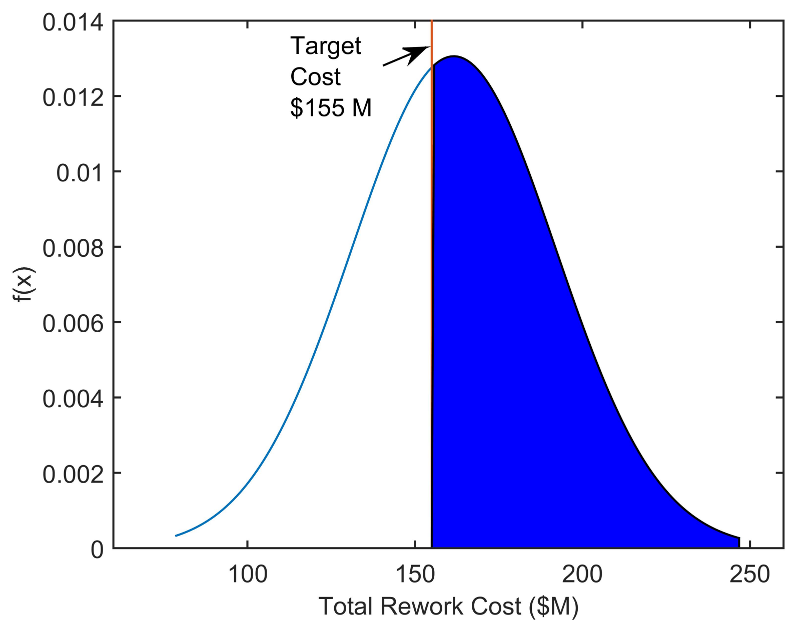


Figure 59: RS-68 cost risk, R_C.

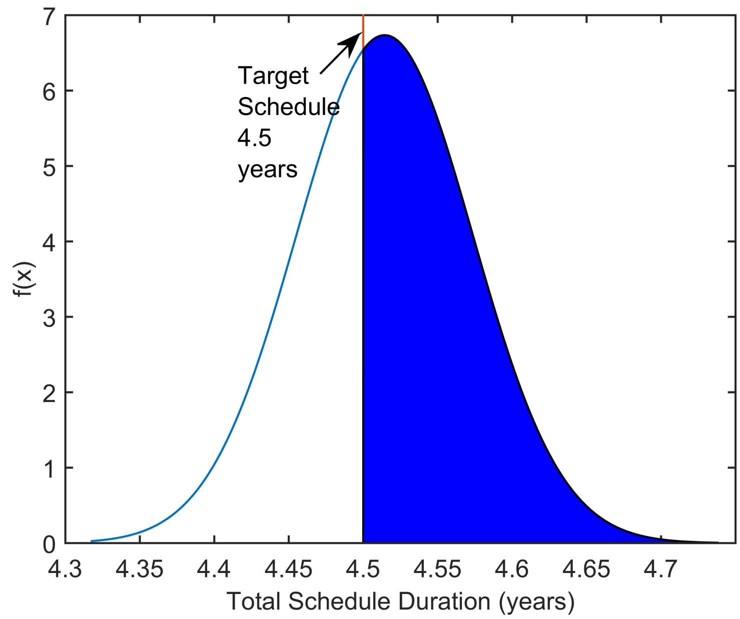


Figure 60: RS-68 schedule risk, R_S.

4.6.5 Case Study Conclusions

The RS-68 case study was set up as a validation exercise for this methodology. The primary goal was to demonstrate the ability of the method to accurately predict the impact of rework cycles on reliability, cost, and schedule, and to assess the risk of a VVT strategy. The example was also used to further support the conclusions drawn from the experimental observations in Chapter 3.

First, the case study illustrated the accuracy of this methodology to predict the occurrence of rework cycles. Figure 54 shows the cumulative rework cycles predicted by this method compared to the actual RS-68 rework cycles during development. The simulated results match the actual data very well. By using test fidelity levels in conjunction with the reliability indicator function, the level of insight into the effects of individual tests is increased. The accuracy of the model to predict rework cycles also validates the probability of occurrence assumption for the reliability growth model.

Because FMEA data for this system were not publicly available, alternative methods were used to estimate the number of total rework cycles and the percentage of critical 1 failure modes. The number of rework cycles is estimated based on the Rocketdyne Technical Uncertainty Factor metric. For this case study, the total number of rework cycles was known. The percent of failure modes that result in redesign was estimated based on the parts count approach, where the number of primary components was used as a substitute for the number of failure modes that would lead to LOC or LOV. When using this percentage in the DSM, the average number of redesign cycles simulated was used as the number of failure modes input in the reliability growth model.

The final conclusion from the case study was the confirmation in assumptions and techniques used to estimate rework cost and development schedule. When compared to the actual cost of rework reported from the RS-68 program, the exponential growth of rework costs used in this methodology very closely predicted the actual rework. Similarly, the schedule penalty applied to rework also accurately predicted the total testing duration. This further supports the conclusions drawn from Experiment 3.

The RS-68 case study illustrated the utility of this methodology and verified its ability to accurately assess the impact of rework cycles on the reliability, cost, and schedule of VVT activities. This method will now be applied to different VVT strategies for one system to confirm that the overall research objective has been met. The following chapter will discuss the details of this larger scale problem.

CHAPTER V

APPLICATION & RESULTS

Chapters 3 and 4 presented the development of this methodology through experimentation and observations made from the literature review in Chapter 2. The experiments were designed to test the research questions posed in Chapter 2 and either accept or reject the associated hypotheses. An additional experiment is necessary to verify that the overall research objective has been met and demonstrate the RIVVTS methodology. The research objective and derived requirements are restated below:

> **Research Objective**
> Reduce cost and schedule overruns by modeling the effects of unplanned rework on the verification, validation, and testing of launch vehicle systems, and determining how VVT strategies can mitigate those effects.

The research objective can be achieved through the formulation and implementation of a structured process or methodology that meets the following requirements:

Derived Requirements:

1. The method shall produce quantitative means for comparing alternative VVT strategies.

2. The method shall produce quantitative estimates for the impact of rework cycles on cost and schedule during VVT.

3. The method shall be scalable and flexible enough to enable use for large complex systems.

In order to test the completion of the research objective, this methodology is applied to a real world system and alternative VVT strategies are compared to determine the impact of unplanned rework during VVT. The baseline VVT strategy used for this problem is the Space Shuttle Main Engine, which was introduced in Section 3.1.1. Its well-documented test history makes it ideal for this application problem. Although originally developed in the 70's, the current SSME configuration is still in use today. The SLS plans to modify RS-25D engines for the core stage on early flights, and transition to a cheaper, expendable version at a later date [133]. For this reason, the SSME was determined to still be a relevant example system for the application of this method. First, a review of historical liquid rocket engine test programs is discussed in the following section to determine how different VVT strategies are created.

5.1 Liquid Rocket Engine Test Strategies

The objectives of a testing program vary depending on experience, analytical capabilities, and the technological maturity of the program, measured by Technology Readiness Level (TRL). Low TRL level programs, i.e. TRL 1-3, are focused on gathering sufficient test data for proof-of-concept hardware to support the development of a more sophisticated test article. Mid TRL level programs, i.e. TRL 4-6, use prototype hardware and engineering test units that closely resemble actual hardware [1]. Once the program reaches high TRL levels, i.e. TRL 7-9, testing of engine components and systems is done with emphasis on quality and rigor applied to both the test facilities and test hardware. For Phase C/D VVT, it is assumed that only upper-mid to high TRLs are considered, e.g. TRL 6-9.

An overview of the potential testing elements for a new engine development program are illustrated in Figure 61 [120]. Each element reduces risk and generates data to support the next testing phase — from prototype testing, to development testing, qualification testing, and finally integrated system testing. The confidence gained by incrementally increasing the testing fidelity validates the commitment of additional resources for full-scale hardware to be built and tested. Subscale component tests likely include testing of subscale combustion devices, such as pumps, preburners, or the thrust chamber. After subscale testing is completed on hardware with the desired attributes, the resulting data are used to reduce risk for the full scale component tests. Complex components, such as turbopump assembly and combustion devices, can only undergo development testing due to the high level of system interaction at the engine-level. These components cannot be qualified on an individual basis because the component-level testing environment cannot adequately represent the intended operating conditions [5]. From full-scale component testing, prototype engines are built and tested. The full scale engine development and qualification tests are used to demonstrate that the engine can operate under flight representative conditions. The 'test-as-you-fly' philosophy suggests that the test program should encompass as much of the flight envelope as possible, including worst-case scenarios [5]. Flight readiness is often determined based on both the development and qualification test efforts due to the complexity of liquid rocket engines. Upon completion of flight engine qualification tests, also referred to as certification tests, an integrated systems flight stage qualification test can be performed. Each unique flight engine must be acceptance tested before it is committed to a flight vehicle.

Designing an engine test program is largely subjective. General guidelines have been created to help successfully develop, test, qualify, and accept liquid rocket engines for launch vehicles [5]. The primary cost drivers for these programs are the number of engine samples and total number of tests performed. NASA-STD 2015, a technical standard for liquid-fueled space propulsion engines, requires six qualification units for pump-fed engines and a minimum of one for pressure-fed engines [6]. Standards like this one describe qualification and acceptance test guidelines for space

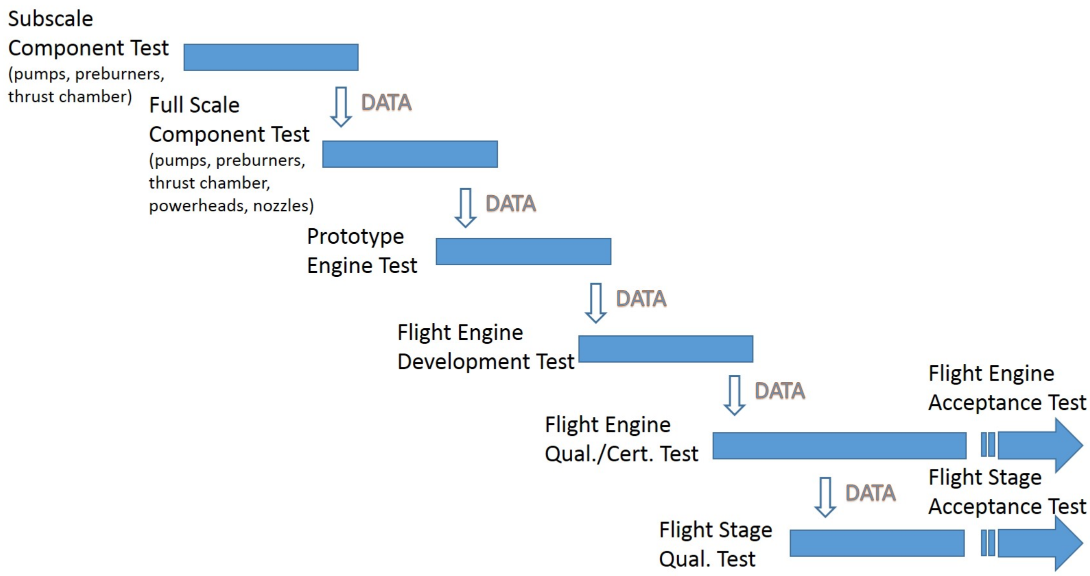


Figure 61: Notional test program elements recreated from [120].

systems, but are difficult to apply to testing.

The necessary number of tests can be difficult to ascertain. The test program must be comprehensive enough to ensure that all performance requirements and functional objectives have been met, but is also subject to programmatic influences. Two approaches have been suggested to determine the total number of tests: statistically relevant based and functional objective based [5]. The first approach uses statistical analysis to determine the minimum number of tests that must be performed to statistically demonstrate engine reliability based on an assumed failure distribution. Weibull failure distributions with different shape parameters are used to model the occurrence of random and wear-out failure modes [9]. Reliability is shown to increase faster for wear-out failure modes with longer duration testing, and for random failure modes with more test units. The number of additional test units required to demonstrate statistically high reliability for random failure modes of launch vehicles, however, often exceeds budget requirements [5]. The second approach is to verify that all functional objectives have been met efficiently and with minimum testing. In doing so, it is still necessary to test the engine within the full flight envelope, both nominal and off-nominal conditions [5]. The specific test plan and functional objectives will vary based on the type of engine. A complete list of functional objectives, and the recommended development and qualification tests to verify them, are given in Appendix A.

Early hot-fire test programs used a formal reliability demonstration approach to qualification, e.g. F-1 and J-2. During these programs, each engine accumulated thousands of starts and improved reliability directly through test-fail-fix cycles [48]. The SSME opted to use design verification specifications as a foundation for test program development [17]. The goal of each test was not to simply demonstrate reliability, but to verify that the design had met a specific requirement. This approach was intended to reduce the cost of testing [92]. The prohibitively high cost of hot-fire engine tests has continued to drive the evolution of liquid rocket engine test programs. The RS-68 was successfully certified in 183 tests, the least number of tests to qualification to-date [154]. This can be attributed to the advancement of computer analysis techniques that identify failure modes prior to hardware fabrication and their design-to-cost, objective based variable test/time approach [154].

A summary of the J-2, F-1, SSME, and RS-68 test programs is provided in Table 27. The testing philosophy evolution that occurred between these projects can be seen by the total number of tests, and total seconds of operation during development and qualification. The formal reliability demonstration approach of earlier engines required a significantly larger number of tests than later programs. The design verification specifications of the SSME reduced the total number of tests despite being a human-rated, reusable engine. The RS-68 further reduced the number of tests and test seconds by learning from past programs and focusing on risk reduction during design. The objective-based approach used during testing allowed multiple objectives to be accomplished in a single test, essentially increasing the fidelity of a given test [154]. The trend that can be seen in these programs is the push to reduce rework cycles and increase the fidelity of tests earlier in the development process.

Table 27: Engine development and qualification summaries [49, 48, 154, 17].

Engine	**Cycle**	**Year**	**Burn Time (s)**	**Development Tests**		**Qualification Tests**	
				Tests	Seconds	Tests	Seconds
J-2	GG	6	450	1,700	160,000	30	3,807
F-1	GG	8	165	2,805	252,958	20	2,255
SSME	SC	9	520	726	110,253	13	5,000
RS-68	GG	4	250	183	18,945	12	

5.2 Baseline Strategy

To determine how a VVT strategy affects the reliability, cost, and schedule risk of a program, alternative strategies are compared to the SSME baseline. Parts of the SSME baseline were introduced in Chapter 3, but a complete definition of the system is discussed in this section to provide the baseline for this application.

5.2.1 System Definition

The first step in defining the baseline is to define the system. Section 3.1.1 provided a discussion on the operating characteristics and major components of the SSME. Section 3.1.2 discussed the development test program details. The total number of development tests, including qualification tests, is 726. For this application, the tests are grouped by year and their fidelity is defined by the average duration. The number of tests per year, average duration, and fidelity are restated in Table 28.

5.2.2 System Assessment

The next step in the process is to assess the system and determine the reliability growth model assumptions and activity relationships. The availability of a comprehensive FMEA for the SSME provides most of this information in detail, enabling a

Table 28: SSME Test History

Average Duration (s)	Number of Tests	Fidelity
2	27	0.15
21	107	0.51
97	184	0.74
158	132	0.82
183	121	0.84
283	128	0.91
400	21	1.0
520	6	1.0

well defined baseline for this application. The reliability growth model requires three parameters: number of failure modes, probability of occurrence, and fix effectiveness factor. The number of critical 1 failure modes was determined to be 7-8% of the total failure modes based on the number of high risk failure modes identified in the FMEA that would lead to LOC or LOV. The probability of occurrence beta distribution used in Experiment 1 is used for these failure modes. The results of Experiment 1 and 2 demonstrated the accuracy of this assumption for predicting rework cycles during SSME development history. These values are listed in Table 29.

These probabilities could be extracted directly from FMEA, if provided. Assumption 3 for the Hall model, listed in Section 2.5.2, states that the initial failure mode probabilities of occurrence constitute a realization of a simple random sample such that $P_i \sim Beta(n, x)$. To represent the FMEA probabilities as a beta distribution, maximum likelihood estimation is used to estimate the shape parameters x and n. More current FMEA worksheets directly provide these probabilities or likelihoods of failure modes. The SSME FMEA, however, provides a frequency of failure factor based on the number of Unsatisfactory Condition Reports (UCR) [146]. A separate study of SSME and J-2 engine failure data found that no empirical relation between the number of UCRs and the number of premature engine cutoffs exists [33]. The number of UCRs is instead driven by the number of inspections or tests that occurred. For this reason, the assumed beta distribution given by Hall is used in this application.

Table 29: Reliability growth assumptions for application baseline

Parameter	**Value**
Number of Failure Modes	160-182
Probability of Occurrence	β(0.22,8.75)
FEF	90%-95%
Number of Tests	726

The activity relationships can also be extracted from the FMEA data. The total number of failure modes was determined to be between 160 and 182 based on the number of failure modes that would cause at least some damage. As stated previously, the probability of those failure modes requiring redesign is 7-8%, i.e. the same number of high risk failure modes used to calculate reliability growth. The percentage of those failure modes that fall into the criticality 2 category is 70-85%. The remaining 7-23% are considered the probability of retest. The SSME DSM used for this application problem is provided in Figure 30.

Table 30: DSM for application baseline

		1	2	3	4	...	728
Design	1			U(.07,.10)	U(.07,.10)		U(.07,.10)
Mfc.	2			U(.70,.85)	U(.70,.85)		U(.70,.85)
Test 1	3			U(.05,.18)			
Test 2	4				U(.05,.18)		
...						U(.05,.18)	
Test 726	728						U(.05,.18)

After the baseline VVT strategy is fully defined, alternative strategies are identified for the SSME. Section 5.3.1 discusses the system definition and assessment of the VVT alternatives for this application problem. The rework impact projection and risk assessment of the baseline strategy and the alternative strategies are compared in Section 5.4.

5.3 Alternative Strategies

5.3.1 System Defintion

The RIVVTS methodology is used to evaluate each of the individual fidelity profiles. The first step of system definition is not required because the alternative strategies use the same system as the baseline VVT strategy. While this is necessary to enable a fair comparison of the rework impact on different VVT activities, defining the VVT activities for each strategy is still required.

The alternative VVT strategies are generated by deviating from the baseline strategy. A curve fit was created using the test fidelity profile of the baseline using a $f(t) = 1 - exp(a * t)$ fit in MATLAB, where $f(t)$ is the fidelity for test $t = 1, ..., 726$, and fit coefficient $a = 0.0059$. The fit coefficient determines the shape of the fidelity profile. For example, a lower coefficient would increase reliability at a less steep slope and a higher coefficient would increase the fidelity earlier with a steeper slope. A range of a values are used to generate the alternative testing strategies. Figure 62 illustrates the test fidelity profiles, where $a_i = m_i a$ for each alternative $i = 1, ..., 35$. The multiplier, m_i, ranges from 0.30 to 2 in 0.1 increments, resulting in 35 alternative test profiles. For the baseline strategy $a = 1$. The minimum value for the multiplier was selected as the test profiles begin to become less distinguishable from one another as the multiplier approaches zero. At the minimum value of 0.3, the VVT strategies are still distinguishable from one another. The maximum value was selected because it is the first strategy that reaches over 1,000 tests, which is assumed to be excessive for modern test programs as determined by the review of historical test strategies in Section 5.1.

To determine the total number of tests for a given profile, T_{a_i}, an Effective Test Effort (EFE) is calculated for the baseline test program. EFE is calculated by integrating the area under the curve of the test profile, which is defined by the fidelity

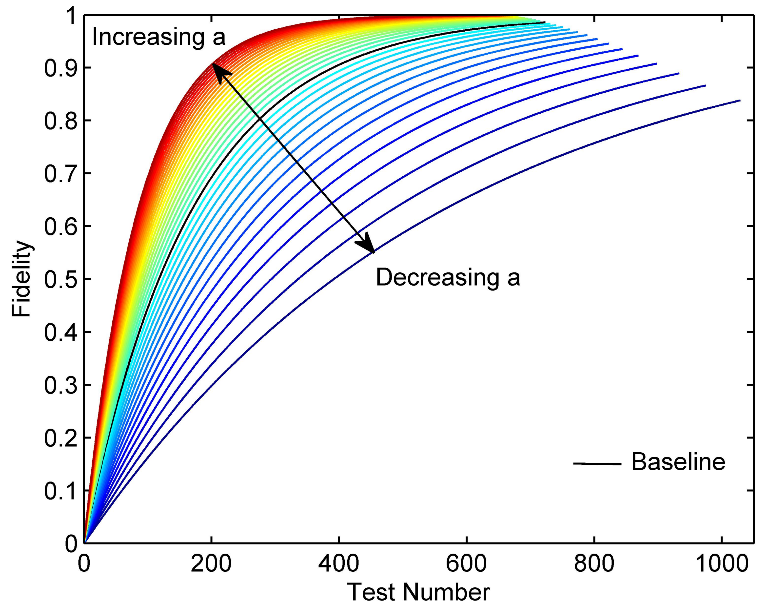


Figure 62: Alternative test fidelity profiles for SSME testing.

level of each test in the baseline VVT strategy:

$$EFE = \int_{1}^{T} f(t)dt \tag{30}$$

where T is the total number of tests for the baseline case, and $f(t)$ is the fidelity at each test t. This value gives an indication of the level of effort that was applied throughout testing, or a weighted equivalent flight calculation that accounts for the nonlinear nature of fidelity during mission duration that was observed in Experiment 1b. The alternative fidelity profile is then integrated from 1 to t until the baseline EFE is reached, and then $T_{a_i} = t$ for profile a_i. The 35 profiles plotted in Figure 62 range in total number of tests from 640 to 1029. After the fidelity level of each activity in each of the alternative VVT strategies is defined using this method, the cost and duration of the activities is determined.

The generic testing strategies developed for this application problem require assumptions to be made for cost and duration based on historical testing costs. A review of historical rocket propulsion testing determined the following to be the primary cost drivers for a test program [92, 121]:

1. Engine thrust class

2. Facility upgrade requirements

3. Project-specific modifications to the test facility

4. Number of tests and test duration

A comparison of various propulsion testing projects showed a positive correlation between test project cost and the engine thrust level [121]. This does not apply to this analysis because only a single engine is being considered. Similarly, the non-recurring costs associated with the second and third drivers are equivalent to any testing strategy for a given engine. The recurring cost of testing is addressed in the last item. Increasing the number of tests allows the non-recurring costs to be amortized over a greater number of tests, resulting in a lower cost per test when more tests are conducted. The test duration increases cost due to the length of time the test facility is used.

The use of generic testing strategies leads to two assumptions in regards to testing cost and schedule. In this context, lower fidelity tests could represent different component-level tests, short duration engine-level tests, or single-objective tests. Additionally, not enough information is provided to assess the cost of component-level tests. To enable a fair comparison between the alternative strategies, the first assumption is that test costs will vary with fidelity level in relation to the test cost distribution used in Experiment 3, Section 3.4.1. The equation for test cost is shown below:

$$\text{Test Cost}_i = f_i^2 c_i \tag{31}$$

where f_i is the fidelity level of test i, and $c_i \sim triangular(0.50, 0.55, 0.63)\$M$ in \$1996. The fidelity squared term is included to represent the project testing cost driver associated with recurring costs.

The second assumption is related to the duration input for testing activities. The testing schedule for the RS-68 program indicates that component-level tests occur roughly twice as frequently compared to engine-level tests. This is illustrated in the program gantt chart of the development testing phase [154]. An assumption for this application problem is made based on the RS-68 test history. VVT activities with a fidelity less than 0.225 are given a duration discount based on the component testing rate experienced during the RS-68 development program. The fidelity value is also selected based on the RS-68 case study, where component-level tests reached a fidelity of 0.225. Activity duration is represented as follows:

$$\text{Test Duration}_i = \begin{cases} f_i <= 0.225 & \sim triangular(3,4,6) * 0.5 \text{ days} \\ f_i > 0.225 & \sim triangular(3,4,6) \text{ days} \end{cases} \tag{32}$$

5.3.2 System Assessment

In the system assessment, any required parameters that define the system are also held constant. This includes the number of failure modes and the failure mode probability of occurrence. The fix effectiveness factor is also kept the same since it is assumed that the same approach to eliminating failure modes and increasing reliability is used for any testing strategy for the same system. The only parameter that changes is the total number of tests. This value is determined using the effective test effort, and is different for each alternative. The reliability growth assumptions for the alternative strategies are listed in Table 31. The DSM probabilities used in evaluating the baseline are used for the alternative strategies.

The final two steps of the RIVVTS methodology, rework impact projection and risk assessment, are presented in the following section. The results of these two steps are compared to the baseline strategy and conclusions are drawn on the effect of VVT strategies to mitigate the impact of rework cycles.

Table 31: Reliability growth assumptions for alternative VVT strategies.

Parameter	**Value**
Number of Failure Modes	160-182
Probability of Occurrence	$\beta(0.22,8.75)$
FEF	90%-95%
Number of Tests	vary

5.4 Results

The results of the RIVVTS evaluation of alternative VVT strategies are provided in Figures 63-68. In each of these figures, the x-axis is labeled test number. This value is the total number of tests required for each alternative VVT strategy to reach the baseline Effective Test Effort, Equation 30. The total test number for a given alternative is common to all figures.

Figure 63 provides the mean reliability projections for all of the strategies. From this plot, it is clear that using the EFE to determine the total number of tests ensures that every alternative meets the same reliability target, within 3%. The reliability risk is presented in Figure 64 and demonstrates the similar reliability risk for each alternative when compared to the baseline. This result is expected due to the tight range of mature reliability estimates.

Figures 65 and 66 represent the total rework cost and total test cost, respectively, for each of the 35 alternatives. The total cost of rework increases at the total number of tests increases. The baseline rework cost for the SSME is \$1.35 B in 1996 dollars. The shaded region indicates the 5% and 95% confidence levels of rework cost for the alternatives. As the total test number increases towards the baseline, the cost of rework increases rapidly. This continues until approximately 800 tests, when the cost of rework continues to increase with the test number, but at a less rapid pace. The uncertainty of rework also increases with test number. The uncertainty range at the lowest test number is 35% smaller than at the highest test number. This result implies that the fewer number of tests required, the less total rework cost is incurred

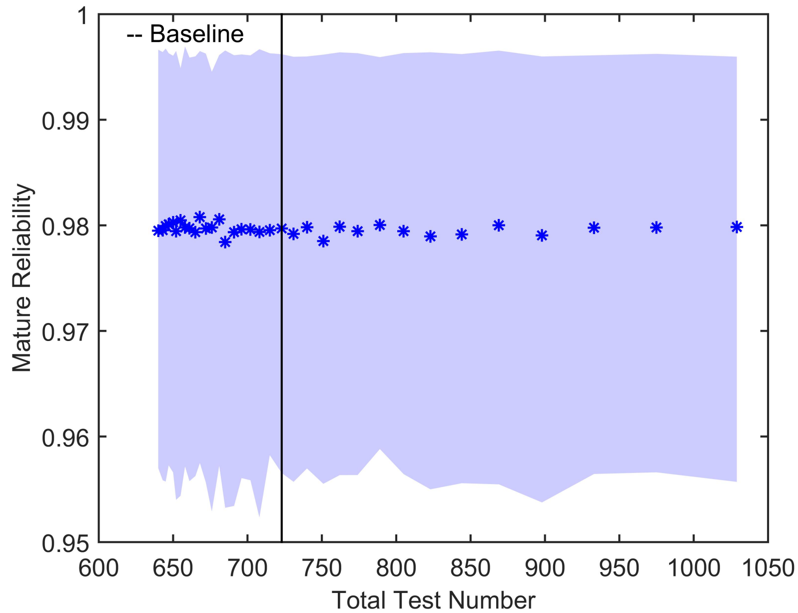


Figure 63: Mature reliability estimate for alternative VVT strategies.

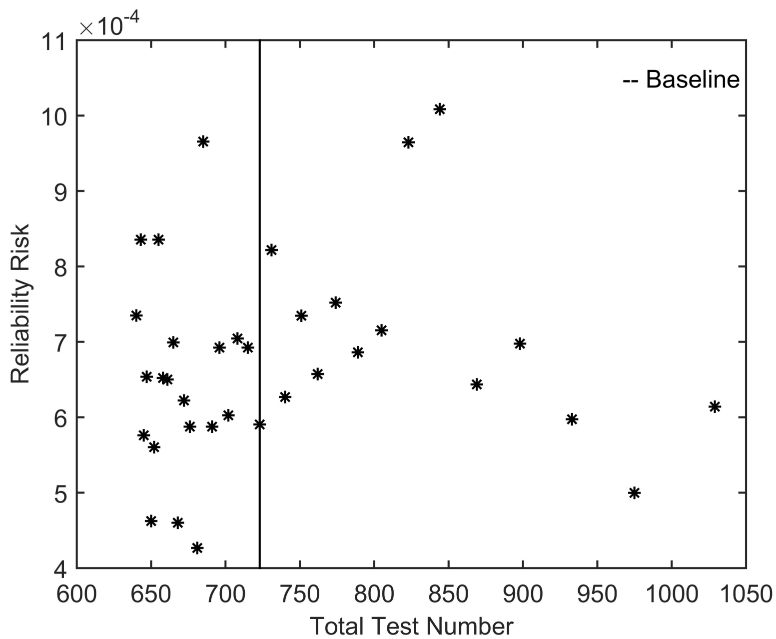


Figure 64: Reliability risk for alternative VVT strategies.

and this cost can be estimated with a greater level of certainty.

The alternative strategies that require a greater number of total tests, correspondingly include a greater number of lower fidelity tests, which can be seen from the alternative test profiles in Figure 62. This is relevant when considering the rate at which rework cycles are uncovered. Based on the assumption established in Section 2.7, that rework cycles cost more the later they occur, slower progressing test programs will not perform as well in terms of rework cost. The F-1 and J-2 engines are examples of this testing strategy, completing extensive component testing and a large number of single-objective engine-level tests [92].

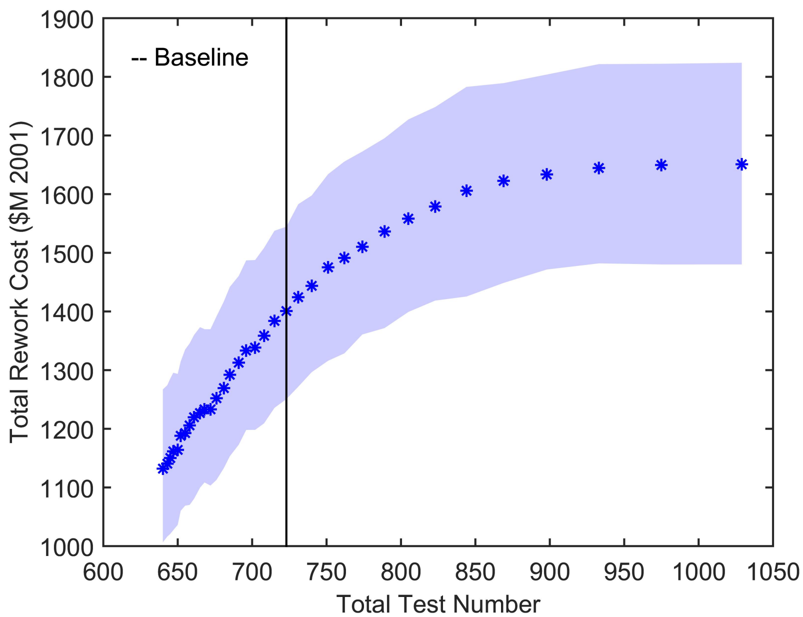


Figure 65: Total rework cost for alternative VVT strategies.

Figure 66 provides the total test cost for each alternative. The alternatives that requirer fewer total tests have a higher total test cost. This is expected based on the test cost assumption stated in Section 5.3.1. The more high fidelity tests in a given test program, the higher the overall test cost. The lower fidelity tests are assumed to cost less, and therefore, the slower progressing test programs will have a lower overall test cost. The uncertainty on the test cost is low and consistent regardless of the total number of tests. The magnitude of test costs is significantly lower than the total cost of rework for all of the alternatives. The impact of rework cost on the program is in line with the historical cost distribution that was provided in Figure 46. The test cost is typically 25% of the total development cost, almost half of the hardware costs [154]. From this assessment, the rework cost is expected to drive the cost risk.

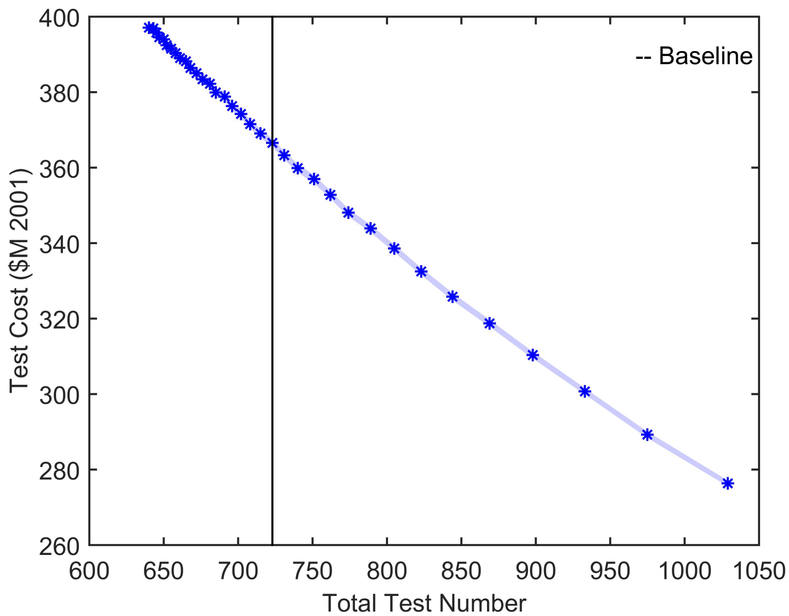


Figure 66: Total test cost for alternative VVT strategies.

The total duration for the alternative VVT strategies is provided in Figure 67, and is plotted with the 5% and 95% percentiles. The test program duration increases with total number of tests, as expected. The shorter duration, lower fidelity tests do not provide enough of a schedule discount to overcome the higher number of tests for the slower progressing test strategies. While the schedule prediction results are not particularly interesting, the schedule risk results provide additional information when compared to the cost risk.

The cost and schedule risks are plotted in Figure 68 versus the total test number. The shape of the cost risk curve confirms the expectation that rework cost drives the overall cost risk of the system. As the total number of tests required begins to

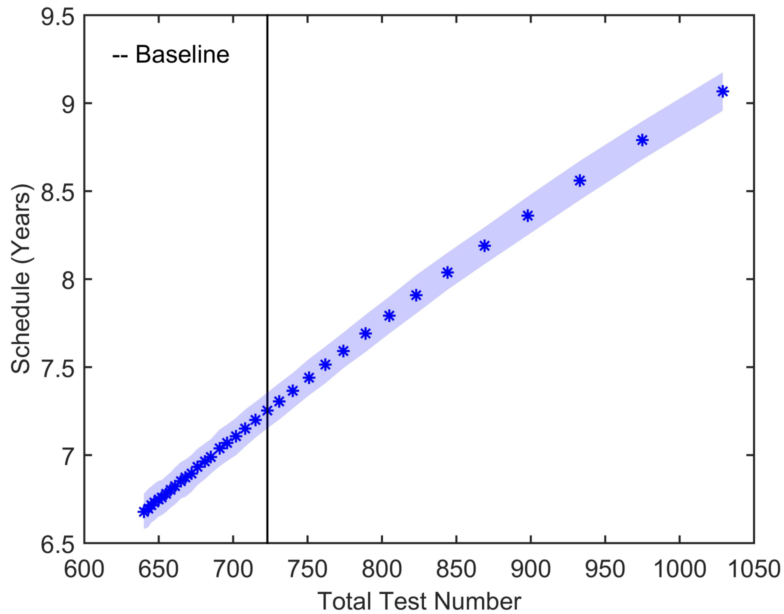


Figure 67: Total duration for alternative VVT strategies.

exceed the baseline, the cost risk increases suddenly. Towards the end of the curve, the last five or six scenarios, the cost risk begins to level out. The rework cost of those alternatives also begins to level out, but the testing cost actually decreases. Consequently, the cost risk increases less rapidly towards the slower progressing test program alternatives and decreases slightly for the four alternative strategies with the most tests. The total cost risk for the four strategies with the most tests decreases due to the continuing decrease in test cost and lack of corresponding increase in rework cost. By evaluating multiple alternative VVT strategies, a conclusion can also be drawn on how VVT strategies can mitigate the impact of rework cycles. The total cost and total cost risk curves indicate that fewer total tests with higher fidelities can reduce the cost of unplanned rework. By increasing the fidelity of tests when possible, the program incurs less rework cost.

The schedule risk axis is on the right axis of Figure 68. The schedule risk also increases as total test number increases, but to a different degree than the cost risk. The linear relationship between schedule and test number allows the quadratic risk impact function relationship to become evident in the schedule risk plot. While cost risk begins to increase suddenly around the baseline value and level off towards the longer test programs, the schedule risk does not increase until approximately test number 850, where it begins to rapidly increase and continues to rapidly increase towards the longer test programs. The schedule risk is based on the total duration for the baseline case. The strategies with fewer tests than the baseline do not incur schedule risk because they do not exceed the total baseline duration. This comparison implies that cost risk due to rework is a bigger driver in liquid rocket engine test programs than both test cost and schedule.

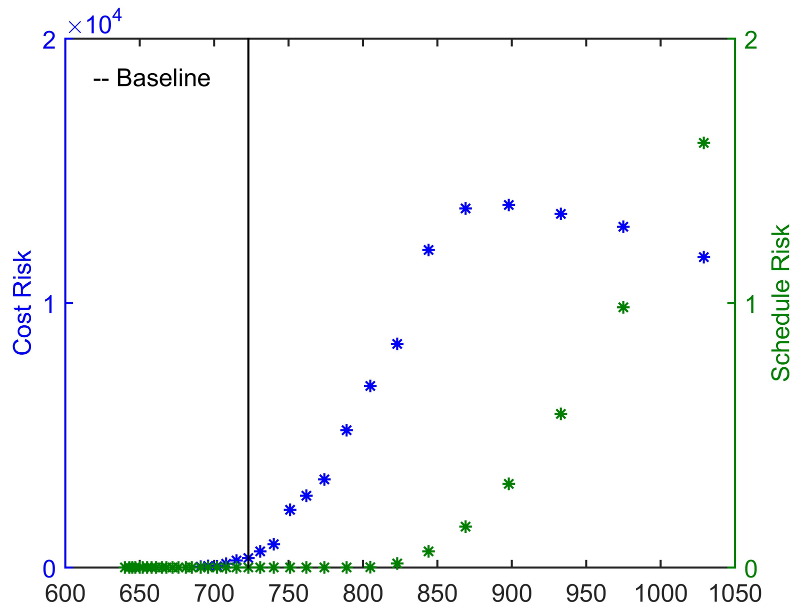


Figure 68: Cost and schedule risk for alternative VVT strategies.

The previous results for the alternative VVT strategies consider the individual outputs of the RIVVTS methodology. The final step of the generic decision-making process discussed in Section 2.3 is to select a VVT strategy. The 'best' VVT strategy can depend on the overall risk profile for the program and the program priorities. For example, some programs are more sensitive to cost than schedule, and can accept additional tests to reduce overall test cost. Figure 69 provides an easier way to compare the RIVVTS output metrics by arranging them in a scatter plot matrix. The scatter plot matrix is used to assess the relationship between multiple variables simultaneously.

Reliability and reliability risk are excluded due to their uniformity for the alternative strategies. From this matrix, it can be shown that test cost is the only metric with a negative correlation to the other metrics. Since early test planning is typically based on only test cost and test schedule, a trade-off between increasing schedule and increasing test cost could be made. If a program is being pressured to produce results quickly, the VVT strategies with fewer total tests can be chosen up to the point where the test cost becomes prohibitively high. Similarly, if cost is the primary driver of VVT planning, the VVT strategies with more tests at lower fidelity have a lower test cost, but require more time to complete.

While rework cost drives the overall program risk, as discussed previously, it is not considered during test planning because it results from unplanned rework [64]. This indicates that early VVT planning is performed with insufficient information regarding the impact of rework cycles. Figure 70 illustrates the benefit of incorporating the cost of rework during VVT planning. The highlighted point represents the strategy with the fewest total number of tests. The total test cost for this strategy is the highest, due to the increased number of high fidelity tests included in the strategy. However, it has the lowest overall cost and schedule. It can be seen here that a short test program with more high fidelity testing can mitigate the impact of unplanned rework on the program. By including the impact of rework on a VVT strategy, the scatter plot matrix allows for an additional metric to be used for strategy comparisons and ultimately, the selection of a ‘best’ VVT strategy.

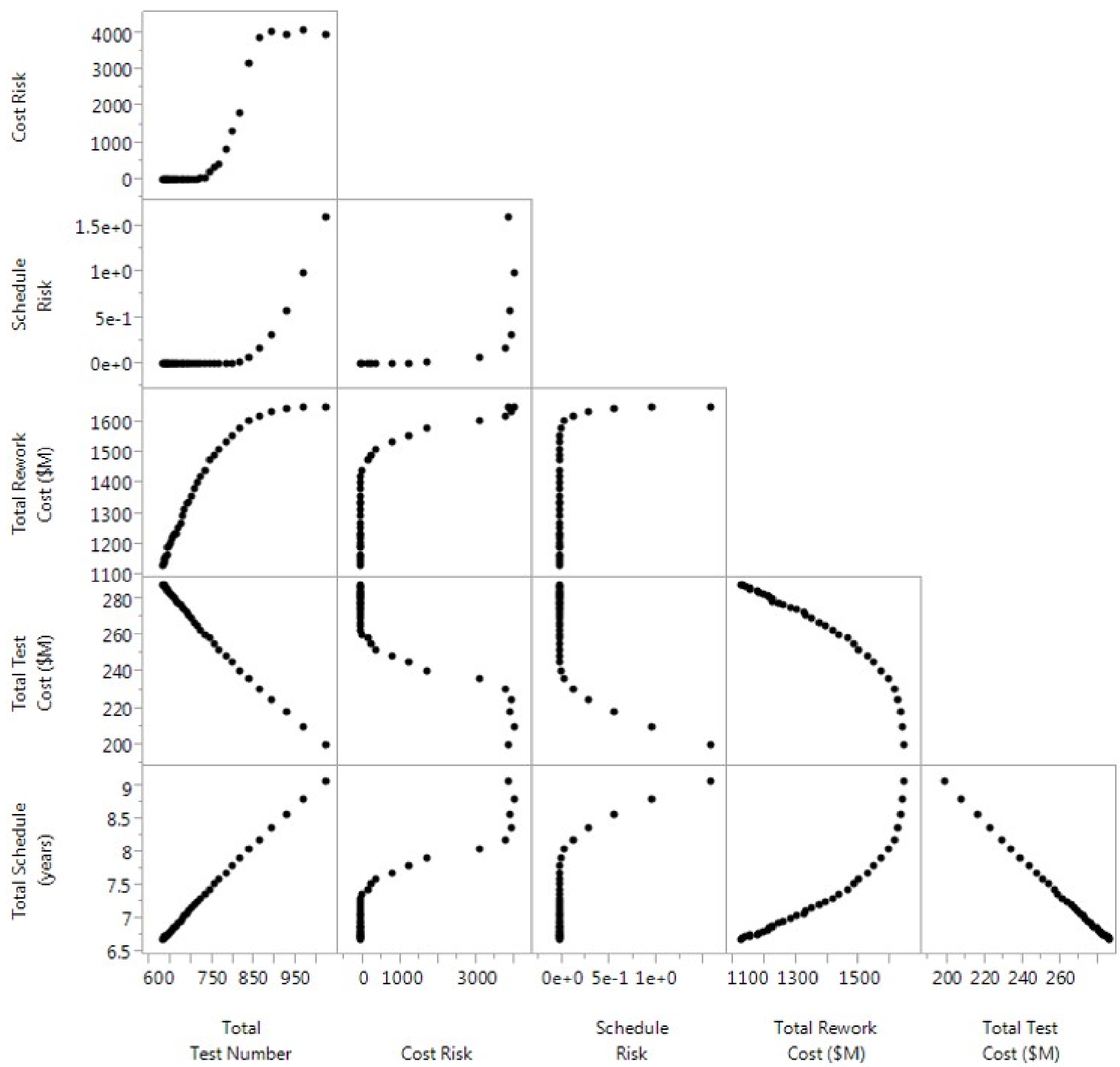


Figure 69: Scatter plot matrix for cost, schedule, reliability, and risk of alternative VVT strategies.

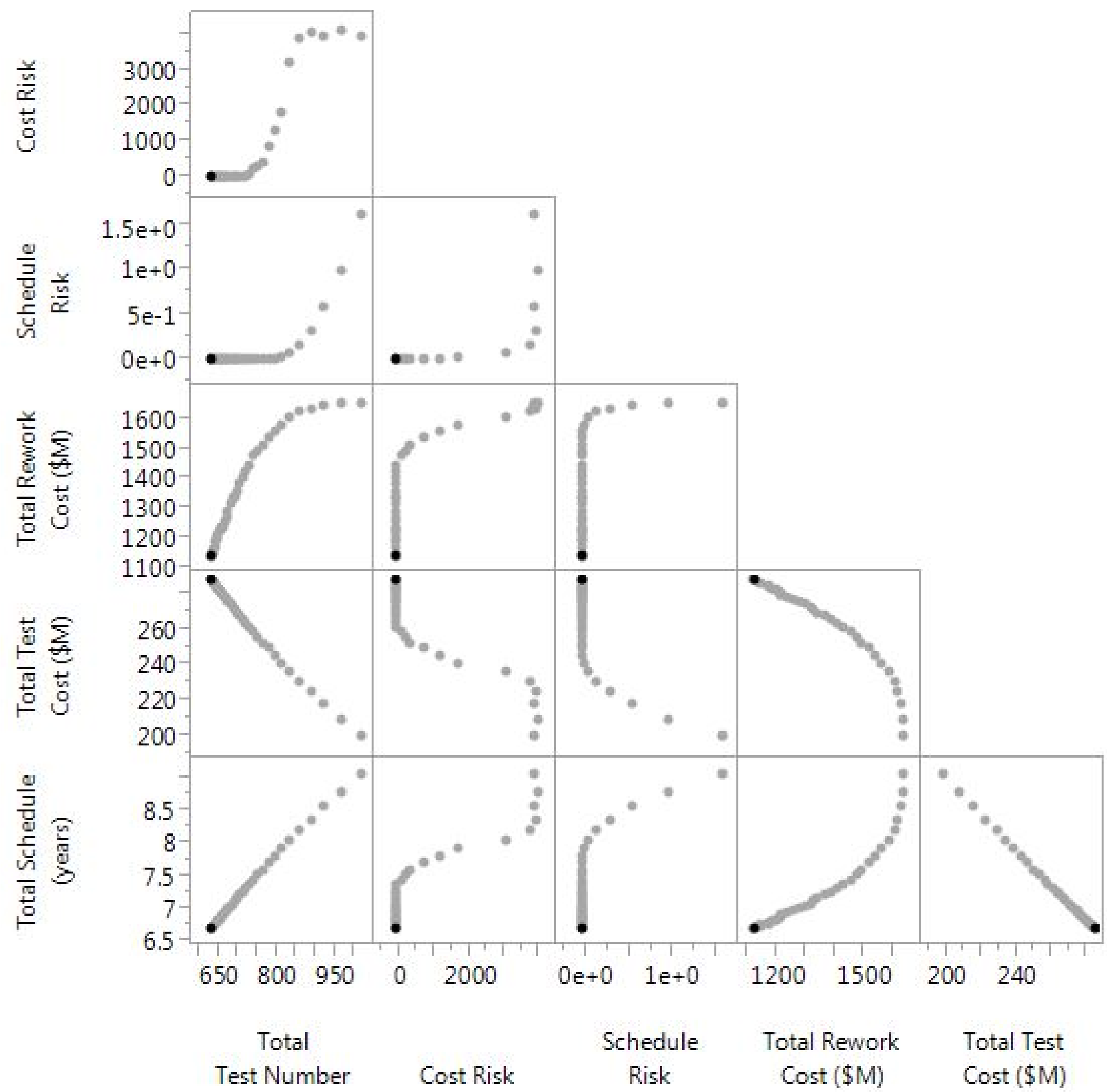


Figure 70: Scatter plot matrix for cost, schedule, reliability, and risk of alternative VVT strategies.

5.5 *Application Problem Summary*

The application problem was designed to confirm that the research objectives were met and to demonstrate the potential benefits of the RIVVTS methodology in reducing the impact of unplanned rework cycles during VVT. Section 5.5.1 provides a summary of the results from Section 5.4, and Section 5.5.2 reviews the overall research objective requirements to confirm that they have been met by the application problem.

5.5.1 Summary of Results

The SSME was selected as the system for this problem due to its long test history which allowed for a detailed baseline to be used for comparison. The alternative strategies were generated by first creating a curve fit of the actual SSME test fidelity profile, and then perturbing that by small increments using a fit coefficient, which equals one for the baseline. Fit coefficients less than one created VVT strategies that consisted of more lower fidelity tests, and longer overall test programs. These test strategies can be compared to the F-1 and J-2 engine programs that had a comprehensive component test program and a high number of engine-level tests aimed at reliability demonstration. Fit coefficients greater than one created VVT strategies that required fewer overall tests, and increased test fidelity earlier in the test program. These strategies can be compared to the RS-68 test program that utilized multiple-objective tests and was able to certify the engine in a historically low number of tests.

A total of 35 alternative test strategies were modeled for this application problem using the RIVVTS methodology. These alternatives were compared based on the estimated mature reliability, rework cost, test cost, and overall program schedule. The mature reliability projection was relatively stable, all of the alternatives were within 1% of the target value. This was expected due to the total number of tests being determined by the effective test effort. The total cost of rework increased rapidly as the total number of tests increased, and continued to increase at a slower rate after approximately 850 total tests. The uncertainty of rework cost also increased as the test programs got longer, with the shortest test program uncertainty range being 35% smaller than the uncertainty range for the longest program.

The cost of rework was determined to be the primary driver of risk because it was the most sensitive to the total number of tests and the time at which rework occurred. By using lower fidelity tests, failure modes took longer to uncover and therefore, rework cycles to fix them were more costly. A greater number of tests also resulted in an increase in schedule risk, but this increase was not observed until the last 6 alternatives, wherein the total number of tests was greater than 850.

5.5.2 Research Objective Requirements

The application problem presented in this chapter was designed for two primary purposes: first, to verify that the overall research objective and criteria are met, and second, to demonstrate the possible benefits of the RIVVTS methodology on a relevant launch vehicle subsystem. To verify that the research objective has been met, the application problem outputs are compared to each criteria, and then to the overall research objective.

The first criterion states that the method must produce quantitative means for comparing alternative VVT strategies. This criteria can be confirmed by observing the application problem results in Section 5.4. The use of four metrics to assess the value of a single VVT strategy provides four distinct quantitative means for decision-makers to compare strategies depending on the program priorities. A criterion for quantitative outputs was included for all four metrics, having been derived from the overall objective, to ensure this result.

Figures 63 - 68 illustrates the quantitative result of mature reliability, reliability risk, total rework cost, total test cost, total program schedule, and cost/schedule risk for each VVT alternative strategy. Each point in Figure 63 provides the average mature reliability prediction which accompanies a complete reliability growth prediction over the full test program. Similarly, the total rework cost plotted in Figure 65 can be calculated as a cumulative cost over the length of the test program for additional information. Figure 68 illustrates a quantitative trade that can be made by comparing cost and schedule risk. By observing these outputs, it can be confirmed that the first overall research criterion is met.

Figures 65 and 67 can also be used to confirm that the second overall criterion is met. It states that a quantitative estimate for the impact of rework on cost and schedule be provided by the RIVVTS methodology. These figures demonstrate the ability of the engineering build-up cost methodology and DSM schedule representation to provide realistic and quantitative estimates for the impact of rework during VVT.

The final requirement is that the method be scalable and flexible enough to use for large complex systems. The application problem included testing strategies that range from 600 to over 1,000 tests at different fidelity levels. This range demonstrates the flexibility and scalability of the RIVVTS methodology. The run-time for changing the discrete event simulation model was evaluated to address the issue of flexibility. While a single simulation run only takes 2-4 seconds for a given test strategy, changing the model to a different test strategy takes between 15 minutes to an hour, depending on the number of activities being modeled. A Simio add-on allows an automated transition between strategies. The full test problem included 1,000 runs each for 35 alternative strategies. At approximately 3 seconds per run and 30 minutes between alternatives, the test problem was completed in less than 2 days. This physical evaluation of the RIVVTS model demonstrates its adaptability to a wide range of fidelities and activities.

The results of the application problem can therefore be used to verify that the original requirements for the research objective have been met. The application problem provides quantitative metrics to select between VVT alternatives, two of these metrics include quantitative estimates for the impact of rework during VVT, and the overall methodology is scalable and flexible for large complex systems.

CHAPTER VI

CONCLUSION

6.1 Summary of Findings

The goal of this research was to develop a methodology for evaluating the impact of rework cycles on the verification, validation, and testing of launch vehicle systems. In Chapter 1 the difficulty of achieving first-flight for launch vehicle programs in recent history was discussed. Excessive cost overruns and schedule slippages have resulted in program cancellations and left the U.S. without the ability to independently launch people into space since 2011. The unplanned rework that occurs during VVT was shown to be a significant contributor to these overruns, but has not been explicitly considered during VVT planning. This observation led to the overall research objective for this thesis with is restated below.

> **Research Objective**
> Reduce cost and schedule overruns by modeling the effects of unplanned rework on the verification, validation, and testing of launch vehicle systems, and determining how VVT strategies can mitigate those effects.

The first research question posed in Section 2.3 addressed the foundation of this methodology by asking what are the necessary components to provide a complete assessment of the value of a VVT strategy. In Section 2.4 the academic efforts to improve VVT planning and the current industry standard approach were discussed and categorized into cost or cost versus benefit approaches. The cost versus benefit approaches were determined to provide a more comprehensive value according to the formal definition of VVT stated in Section 2.2.1. To provide a complete assessment, four metrics were chosen to evaluate a VVT strategy: quality, cost, schedule, and risk. The literature review and background research conducted to determine the necessary output of this methodology led to the following conjecture:

> **Conjecture: 1**
> If quality, cost, schedule, and risk are used as metrics to evaluate the impact of rework during VVT, it will provide the most complete assessment of VVT activities, and will enable a quantitative comparison of alternative VVT strategies.

After selecting the four metrics for evaluating a VVT strategy in conjecture 1, existing techniques for evaluating each metric were reviewed and discussed to further develop those components of the overall methodology. The measure of benefit provided by VVT — referred to as quality — was inconsistently defined in the existing VVT planning approaches. Technical performance measures, risk reduction, rework reduction and other measures of quality were discussed in Section 2.5. Section 2.5.1 discussed the current priorities for NASA's SLS development program to provide context in determining an appropriate quality measure for this research. NASA's current focus on improving the reliability of SLS by an order of magnitude over previous launch vehicles led to the formulation of conjecture 2.

> **Conjecture: 2**
> If reliability is used as a quality metric for launch vehicle systems, it will provide a quantitative representation and accurate measure of quality for VVT activities.

The selection of reliability as a measure of quality for launch vehicle VVT strategies led to a review of the following commonly used reliability analysis techniques in Section 2.5.2: fault tree analysis, reliability block diagrams, failure mode and effect analysis, and reliability growth models. The merits of these techniques were discussed, and ultimately, discrete reliability growth projection was chosen to evaluate reliability during development testing phases. A number of models that had already been proven to accurately estimate launch vehicle reliability growth were then reviewed, and Hall's discrete reliability growth model was selected for use in this methodology. Experiment 1a was designed to test the accuracy of Hall's model during VVT. The assumption that all trials or tests were equivalent was found to be a weakness when using Hall's approach across the full testing phase, where tests of different fidelity levels were conducted. An adaptation of Hall's model was suggested in Section 3.2.3, in which the fidelity of each test would be defined by the percentage of failure modes that could be uncovered during a given test. Experiment 1b was conducted to test the use of different fidelity levels to determine if an appropriate fidelity measure could be identified and used to improve upon Hall's model for use during VVT. Three fidelity definitions were used as comparison, based on identified approaches to defining equivalent flights in the literature. The results substantiated hypothesis 2a.

Hypothesis: 2a
If Hall's reliability growth model is adapted to include defect elimination as a function of test fidelity, it will provide reliability projection with quantitative insight into individual VVT activities.

After it was determined that the adapted discrete reliability growth model could be used to accurately project reliability during VVT, additional research questions were posed to further define the remaining evaluation metrics. Section 2.6 reviewed common schedule projection techniques and discussed their strengths and weaknesses as applied to this problem. For VVT schedules, a DSM was identified as a concise way to represent activity relationships and allow for the stochastic nature of rework cycles to be modeled. A weakness in the current use of DSMs to represent rework was identified in the implicit evaluation of internal rework within an activity. Experiment 2 was designed to address this weakness, and verify that the use of internal rework probabilities would result in a more accurate schedule estimate. The results of Experiment 2 substantiated hypothesis 3.

> **Hypothesis: 3**
> If a DSM is adapted to explicitly account for the probability of internal rework, it will provide a stochastic and quantitative model of rework impacts that is more accurate for VVT processes than if internal rework is implicitly included in the activity duration distribution.

The cost of VVT activities is a necessary consideration when choosing a VVT strategy. Additionally, the cost of rework during VVT was identified as a significant contributor to the cancellation of launch vehicle programs in recent history. Therefore, a cost metric is included in VVT strategy assessment. Current cost estimating techniques and reliability-based cost estimating techniques were reviewed for use in this methodology. The reliability-based methods were generic in general, and the Rocketdyne method did not account for the stochasticity of rework cycle occurrence. The bottom-up, engineering build-up approach was determined to be flexible enough to account for stochasticity, and more accurate than historical data-based methods. The following conjecture was made to assess cost of VVT activities and the impact of rework cost.

> **Conjecture: 4**
> Using engineering build-up to calculate cumulative cost will give a quantitative estimate that is more accurate than historical data based methods and accounts for the stochastic nature of rework cycles.

After techniques for reliability, cost, and schedule assessment were identified, research question 5 was asked the most appropriate use of different input distributions and uncertainty quantification techniques for risk assessment. Section 2.8.1.1 discussed the commonly used uncertainty categorizations to determine the type of uncertainty present during VVT. The presence of aleatory uncertainty, by definition, cannot be eliminated, while VVT activities are designed to reduce the endogenous epistemic uncertainty that exists during early design. Section 2.8.1.2 reviewed the necessary components for quantifying uncertainty to determine an approach for this methodology, starting with the input variable distributions. The triangular distribution parameters — pessimistic, optimistic, and most likely values — are more intuitive than the abstract shape parameters required for a Beta or Weibull distribution. For this reason, the triangular distribution was determined to be a more traceable option for assessing the input distributions of the cost and schedule for an individual activity.

In quantifying the uncertainty of rework impact, a discrete event simulation was chosen to allow for the stochastic nature of rework cycle occurrence and impact to be captured. Section 2.7 discussed the change in rework cycle cost as the system progresses through the development phase. The DES evaluates the system at discrete steps in time to determine if rework has occurred and assess the impact. The ability to account for the stochastic nature of rework cycles, as well as the quantitative output of reliability, cost, and schedule distributions makes the DES the best choice for this methodology. The resulting output distributions are then used to calculate the risk level of a given VVT strategy using a quadratic impact function. This led to the formulation of conjecture 5 in response to this research question.

Conjecture: 5

- Using triangular input distributions, the assumptions required will be more traceable than if beta and Weibull distributions are used.

- If DES is used for simulation, the results will allow for quantitative comparisons between VVT strategies and account for stochasticity of rework cycles.

The last research question was posed in Section 2.9 to determine how the probability of rework can be determined. Many approaches to estimating design rework cycles were presented in Section 2.9.2, but few address the rework during VVT. Historical data-based methods and expert opinion were the most commonly used approaches, but can be subject to bias by the estimator. The Rocketdyne TUF metric was a more relevant approach, but is limited to estimating the total number of rework cycles and neglects their stochasticity. FMEA was identified as a possible technique that is more traceable than expert opinion and historical data-based methods and can be used to predict when rework occurs. As a commonly used reliability analysis technique, utilizing FMEA for this purpose requires no additional effort on the part of the designer. A comprehensive FMEA provides total number of failure modes, failure probabilities, and severities. Experiment 3 was designed to test the use of FMEA for estimating the probability and impact of rework cycles. Utilizing Hall's indicator function to determine when rework cycles occur based on the failure mode probabilities proved to be an accurate prediction of rework cycles when compared to the actual rework data from an existing system. The results of this experiment substantiated hypothesis 6, which is restated below.

Hypothesis: 6

If subsystem and system level FMEA is performed, then the resulting data will provide quantitative rework probabilities that are more traceable than expert opinion and the data will be more readily available than expert opinion and all historical data based methods.

These research questions, literature review, and experiments were used to develop the individual components of the RIVVTS methodology. The complete method was tested on a case study, where the output was compared to actual data from the RS-68 engine. The purpose of this example was to illustrate that this methodology accurately captures the occurrence and impact of rework cycles seen during VVT. The secondary purpose of the example was to validate the development of the cost, schedule, and reliability assumptions for future applications of this method.

Finally, this method was applied to alternative VVT strategies for a relevant launch vehicle subsystem, the Space Shuttle Main Engine. The alternative VVT strategies were evaluated to determine how the impact of rework can be mitigated through VVT activities. The results give interesting observations regarding the benefit of comprehensive component testing versus early integrated testing. Ultimately, this final application problem demonstrates the merits of this methodology in evaluating VVT strategies and provides a risk-informed decision making environment to reduce the impact of rework cycles on the verification, validation, and testing process of launch vehicle systems and subsystems.

6.2 Contributions

The work in this thesis provides multiple contributions to improve the test planning process for launch vehicles. These contributions relate to the overall research objective and to the individual techniques used to address it.

The first contribution is in the field of reliability projection across testing phases.

A discussion of reliability growth models in Section 2.5.2 identified weaknesses in the use of discrete models during testing. Many authors suggest that their method can be used during testing, but the assumption that each test has an equal probability of uncovering all failure modes is a limiting assumption when applying it to tests with varying fidelity levels. A hypothesis was made to adapt an existing discrete reliability growth model to account for test fidelity levels to provide more insight into the effect individual tests have on reliability. Experiment 1 tested the use of different test fidelity definitions to model the reliability of an actual system during development testing, and ultimately substantiated that a functional fidelity level definition could be used to accurately project reliability growth across test phases. The following case study in Chapter 4 further validated the use of this adapted reliability growth model on an actual system. This contribution is an improved method of predicting reliability growth during testing with varying levels of fidelity while the effects of individual tests are represented.

The second contribution is in the prediction of unplanned rework occurrence and impact. In Section 2.9, research question 6 was introduced and options for estimating the probability of rework cycles were discussed. During this discussion, weaknesses in the existing methods were identified. FMEA was suggested as a method to overcome these weakness and improve the traceability of rework estimates. Experiment 2 was designed to test the use of FMEA for estimating the number of rework cycles, and probability of failure occurrences when combined with Hall's adapted reliability indicator function. Previous attempts to estimate rework failed to account for the stochasticity of their occurrence and impact. The results of this experiment illustrated the usefulness of this approach. The case study presented in Section 4.6 further demonstrated the ability of this approach to accurately estimate the cumulative rework cycles of the RS-68 engine development program. The results from the experiment and case study validated the use of FMEA and the adapted indicator function as an accurate and traceable approach to predicting rework cycles during VVT.

The third contribution is provided by the actual RIVVTS methodology. The research objective was to reduce cost and schedule overruns by modeling the effects of unplanned rework on the verification, validation, and testing of launch vehicle systems, and determining how VVT strategies can mitigate those effects. The criteria for this objective were generated through a discussion on what constitutes a complete assessment of a VVT strategy and what is the desired output of this methodology. Ultimately, it was determined that this research must provide a quantitative means of comparing alternative strategies that assesses the impact of rework cycles for a risk-informed decision making environment. These derived requirements were based on identified weaknesses of the industry standard approach in VVT planning, and therefore, represent improvements on the current state-of-the-art approach.

The RIVVTS methodology provides a much needed link between VVT planning and the impact of rework cycles. Typically, VVT planning is based one subjective expert opinion while considering overall cost, schedule, and risk to the program. A lack of structured assessment and no explicit rework impact projection limits the ability to quantitatively compare alternative VVT strategies. With the RIVVTS method, decision makers can now consider the impact of rework cycles throughout the development testing process when planning VVT activities. The application problem presented in Chapter 5 demonstrates that this method is well suited for conducting cost versus benefit trades for alternative strategies, which can help decision makers during VVT planning.

This method also provides a flexible and traceable approach to assessing a VVT strategy depending on the current phase of the design process. For early design test planning, the assumptions can be made at the system level and test definitions can be grouped according to test phases. This provides the design team with a quantitative assessment of rework impact on a given system earlier in the design process. As the program progresses and enters detailed design, the assumptions can be generated at the component level and test can be defined individually. This flexibility allows the method to be applied throughout the development process, and updated as more information becomes available.

6.3 Future Work

During the development of this research, several additional areas of interest were identified for future work. These areas concern both the specific components within the RIVVTS methodology, but also directions to expand upon the foundation this method has created.

The first area for improvement of the RIVVTS methodology is to expand the DSM to incorporate parallel testing activities. Testing often occurs simultaneously at multiple sites. For example, engines testing is assumed to be conducted at a minimum of two sites. While this reduces the overall schedule, it increases the uncertainty in reliability and rework estimation. By assuming sequential testing activities, the full capability of discrete event simulation was not utilized. The benefit of using DES over Monte Carlo Simulation is the ability represent state changes at discrete times in the system, making it ideal for navigating two parallel test executions that start and stop at different time steps. Increasing the capability of the RIVVTS methodology to include parallel testing would provide a more accurate assessment of reliability in real-time.

The second area for potential research would be to implement a more complex reliability growth model. The adapted Hall model represents component testing as a subset of system reliability. A more complex reliability growth model could enable reliability projections at the component level. The CONTRAST method discussed in Section 2.5.2 would be well suited to supplement this research. Already based on the Hall model, CONTRAST uses the reliability growth models at the subsystem level, and then uses FTA to aggregate the lower level reliabilities to the system level. It was considered for the initial development of this methodology, but the additional assumptions required to generate the lower level reliability growth curves were deemed unnecessary. Combining CONTRAST and RIVVTS would provide more insight into the cost, schedule, and reliability of complex systems during development testing.

The third, and perhaps, simplest area for future work would be to include an optimization routine to find an optimal VVT strategy based on the rework impact evaluation provided by RIVVTS. A discrete, multi-objective optimization routine like the NSGA-II would be ideally suited for this problem. NSGA-II is a popular non-domination based genetic algorithm that was identified as a potential option for optimizing this problem during preliminary research. With the model in place, an optimization based on weighted objectives of minimizing cost, schedule, and reliability risks would provide another tool for risk-informed decision making.

APPENDIX A

DEVELOPMENT AND QUALIFICATION TEST MATRIX

[5]

		Component		Engine	
	Objective	Dev	Qual	Dev	Qual
Performance	Steady State Performance Characterization			X	X
	Repeatability			X	X
	Run-time Trends			X	X
	Engine Influence Coefficients			X	X
	Mixture Ratio Excursions			X	X
	Thrust/Mixture Ratio Margin Demonstration			X	X
	Ignition System	X	X	X	X
	Turbomachinery	X		X	X
	Combustion Devices	X		X	X
Life	Operational Life/Durability	X	X	X	X
	Single Burn Endurance Test			X	X
	Service Live, Number of Starts			X	X
	Acceptance Test Validation			X	X
Functional Characteristics	Cold Shock Tests			X	
	Cold Flow Tests	X	X	X	
	Propellant Conditions				
	- Pre-start Chilldown			X	X
	- Start Propellant Conditions			X	X
	- Steady State			X	X
	- Shutdown Propellant Conditions			X	X
	Transient Characterization				
	- Start Transient			X	X
	- Restart			X	X
	- Throttle Transient			X	X
	- Shutdown Transient			X	X
	- Abort Shutdown			X	X
	NPSP Margin and Cavitation	X		X	X
	Pogo and Compliance Characterization	X		X	X
	Ancillary Subsystems				
	- Autogenous Pressurization			X	X
	- Valve Actuation	X	X	X	X
	- Purges	X	X	X	X
	- Electrical Power and Integration	X	X	X	X
	Thrust Vector and Gimballing				
	- Gimbal Limits	X	X	X	X
	- Roll Control Limits			X	X
	- Ambient Environment			X	X
	- Inspections	X	X	X	X
	- Heat Flux			X	X
	- Clearance			X	X
	- Interface Compatibility			X	X
	- Thrust Vector Alignment			X	X

		Component		Engine	
	Objective	Dev	Qual	Dev	Qual
Controls	Functional Tests	X	X	X	X
	Eng. Control System Malfunction Logic Check	X	X	X	X
	Engine Health Management	X	X	X	X
Operations	Pre-Test Inspections and Checkouts	X	X	X	X
	Leakage	X	X	X	X
	Post-Test Inspections	X	X	X	X
	Drying Purges			X	X
	Line Replaceable Unit Demonstrations			X	X
	Reusability	X	X	X	X
	Operability			X	X
	Preflight Procedures and Flight Sequences			X	X
Environments	Thermal Environment	X	X	X	X
	Climatic Tests	X	X	X	X
	Vibration/Shock/Acoustics				
	- External Vibration	X	X	X	
	- Self-induced Vibration	X	X	X	
	Modal Surveys/Testing			X	X
	Vehicle Interface Loads			X	X
	Electromagnetic Interference/Compatibility	X	X	X	
Design	Proof Pressure	X	X		
	FOD/DOD Tolerance	X		X	
	Structural Model Validation			X	X
	Margin Testing			X	X
	Human Rating	X	X	X	X
	Hardware Discrepancy Tracking	X	X	X	X
Physical	Gas Liquefaction Control	X		X	X
	External Icing Control	X		X	
	Mass Properties				
	- Mass	X	X	X	X
	- Center of Gravity	X	X	X	X
	- Moments of Interia	X	X	X	X

REFERENCES

[1] "Nasa marshall space flight center - verification documentation examples," *MSFC-HDBK-2221*, 1994.

[2] "Nasa marshall space flight center - verification handbook," *MSFC-HDBK-2221*, 1994.

[3] "Planning, developing, and managing an effective reliability and maintainability (r&m) program," *NASA-STD-8729*, 1998.

[4] "Space engineering verification," *European Cooperation for Space Standardization*, 2009.

[5] "Test and evaluation guideline for liquid rocket engines," *JANNAF*, 2011.

[6] "Strength and life assessment requirement for liquid-fueled space propulsion system engines," tech. rep., NASA Technical Standard 5012A, 2015.

[7] "Nasa authorization act of 2010," Oct. 11, 2010.

[8] Abdelsalam, H. M. and Bao, H. P., "A simulation-based optimization framework for product development cycle time reduction," *Engineering Management, IEEE Transactions on*, vol. 53, no. 1, pp. 69–85, 2006.

[9] Abernethy, R., Breneman, J., Medlin, C., and Reinman, G., "Usaf durability design handbook, afwal-tr-83–2079," *Prepared for the Aero Propulsion and Power Laboratory, Wright-Patterson AFB, Ohio*, 1983.

[10] Aggarwal, K. and Gupta, J., "On minimizing the cost of reliable systems," *IEEE Transactions on Reliability*, vol. 3, p. 205, 1975.

[11] Ahmadi, R. and Wang, R. H., "Managing development risk in product design processes," *Operations Research*, vol. 47, no. 2, pp. 235–246, 1999.

[12] Ang, A. H., Chaker, A. A., and Abdelnour, J., "Analysis of activity networks under uncertainty," *Journal of the Engineering Mechanics Division*, vol. 101, no. 4, pp. 373–387, 1975.

[13] Arena, M. V., *Impossible Certainty: Cost Risk Analysis for Air Force Systems*, vol. 415. Rand Corporation, 2006.

[14] Arundacahawat, P., Roy, R., and Al-Ashaab, A., "An analogy based estimation framework for design rework efforts," *Journal of Intelligent Manufacturing*, vol. 24, no. 3, pp. 625–639, 2013.

[15] Arundachawat, P., Roy, R., Al-Ashaab, A., and Shehab, E., "Design rework prediction in concurrent design environment: current trends and future research directions," in *Proceedings of the 19th CIRP Design Conference–Competitive Design*, Cranfield University Press, 2009.

[16] Aven, T. and Jensen, U., *Stochastic models in reliability.* Springer, 1999.

[17] Biggs, R. E., *Space shuttle main engine: The first twenty years and beyond*, vol. 29. American Astronautical Society, 2008.

[18] Bilstein, R. E., *Stages to Saturn: A Technological History of the Apollo/Saturn Launch Vehicle.* DIANE Publishing, 1999.

[19] Bjorkman, E. A., Sarkani, S., and Mazzuchi, T. A., "Using model-based systems engineering as a framework for improving test and evaluation activities," *Systems Engineering*, vol. 16, no. 3, pp. 346–362, 2013.

[20] Blischke, W. R. and Murthy, D. P., *Reliability: modeling, prediction, and optimization*, vol. 767. John Wiley & Sons, 2011.

[21] Boehm, B. W. and others, *Software engineering economics*, vol. 197. Prentice-hall Englewood Cliffs (NJ), 1981.

[22] Braun, R., Kroo, I., and Moore, A., "Use of the collaborative optimization architecture for launch vehicle design," in *Proceedings of the 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 1996.

[23] Broemm, W. J., Ellner, P. M., and Woodworth, W. J., "Amsaa reliability growth guide," tech. rep., DTIC TR-652, 2000.

[24] Browning, T. R., "On customer value and improvement in product development processes," *Systems Engineering*, vol. 6, no. 1, pp. 49–61, 2003.

[25] Browning, T. R. and Eppinger, S. D., "Modeling impacts of process architecture on cost and schedule risk in product development," *Engineering Management, IEEE Transactions on*, vol. 49, no. 4, pp. 428–442, 2002.

[26] Browning, T. R., "Modeling and analyzing cost, schedule, and performance in complex system product development," 1998.

[27] Butts, G. and Center, K. S., "Procurement lessons learned,"

[28] Castellini, F., Lavagna, M. R., Riccardi, A., and Büskens, C., "Quantitative assessment of multidisciplinary design models for expendable launch vehicles," *Journal of Spacecraft and Rockets*, vol. 51, no. 1, pp. 343–359, 2013.

[29] CB-04-044, M., "From cb/chief, astronaut oce, to ca/director, flight crew operations," 2001.

[30] Chang, I.-S., "Space launch vehicle reliability," *Crosslink*, vol. 2, no. 1, pp. 22–32, 2001.

[31] Chaplain, C. T., Bothwell, B., Campbell, G., Chan, J., Echard, J., Gallegos, A., Haynes, B., Hobson, A., Lee, J., and McGinty, S., "Space acquisitions: Dod needs to take more action to address unrealistic initial cost estimates of space systems," tech. rep., Government Accountability Office, GAO-07-096, 2006.

[32] Chin, K.-S., Wang, Y.-M., Poon, G. K. K., and Yang, J.-B., "Failure mode and effects analysis using a group-based evidential reasoning approach," *Computers & Operations Research*, vol. 36, no. 6, pp. 1768–1779, 2009.

[33] Christenson, R., Whitley, M., and Knight, K., *Comprehensive Design Reliability Activities for Aerospace Propulsion Systems.* National Aeronautics and Space Administration, Marshall Space Flight Center, 2000.

[34] Conrow, E. H. and CMC, C., "Space program schedule change probability distributions," in *AIAA Space 2010 Conference and Exposition, paper AIAA*, vol. 8834, p. 30, 2010.

[35] COVERT, R. P. and COVARUS, L., "Analytic method for probabilistic cost and schedule risk analysis," *National Aeronautics and Space Administration (NASA), Washington, DC*, 2013.

[36] Crocker, A., Doering, K., Cook, S., Bachtel, R., St Germain, B., and Schaffer, M., "The benefits of advanced booster competition for nasa's space launch system," in *Joint Propulsion Conference*, 2013.

[37] Crosby, P. B., *Quality is free: The art of making quality certain.* Signet, 1980.

[38] Crow, L. H., "Amsaa discrete reliability growth model," *AMSAA, Aberdeen Proving Ground, MD, Office Note*, pp. 1–83, 1983.

[39] Crumbly, C. M., May, T. A., and Dumbacher, D. L., "Nasas space launch system advanced booster engineering demonstration and risk reduction efforts,"

[40] Davidson, M., "Rs-25: The clark kent of engines for the space launch system," 2013.

[41] Dezfuli, H., Benjamin, A., Everett, C., Maggio, G., Stamatelatos, M., Youngblood, R., Guarro, S., and others, "Nasa risk management handbook," tech. rep., Technical report, NASA, 2011.

[42] Diaz, C. F. and Hadipriono, F. C., "Nondeterministic networking methods," *Journal of construction engineering and management*, vol. 119, no. 1, pp. 40–57, 1993.

[43] Dick, S. J. and Launius, R. D., *Critical issues in the history of spaceflight.* Government Printing Office, 2006.

[44] Dodson, B. and Nolan, D., *Reliability engineering handbook.* Marcel Dekker New York, NY, 1999.

[45] Drezner, J. A., Jarvaise, J. M., Hess, R. W., Hough, P. G., and Norton, D., "An analysis of weapon system cost growth," tech. rep., DTIC Document, 1993.

[46] Duane, J., "Learning curve approach to reliability monitoring," *IEEE Transactions on Aerospace*, vol. 2, no. 2, pp. 563–566, 1964.

[47] Elmaghraby, S. E., "Activity networks," 1977.

[48] Emdee, J., "A survey of development test programs for hydrogen oxygen rocket engines," in *39th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, AIAA Paper*, vol. 749, 2001.

[49] Emdee, J., "A survey of development test programs for lox/kerosene rocket engines," in *37st Joint Propulsion Conference, Salt Lake City, UT. AIAA Paper*, vol. 3985, 2001.

[50] Engel, A., *Verification, validation, and testing of engineered systems*, vol. 84. John Wiley & Sons, 2010.

[51] Engel, A. and Barad, M., "A methodology for modeling vvt risks and costs," *Systems engineering*, vol. 6, no. 3, pp. 135–151, 2003.

[52] Eppinger, S. D., "Model-based approaches to managing concurrent engineering," *Journal of Engineering Design*, vol. 2, no. 4, pp. 283–290, 1991.

[53] Eron, M., "Cost analysis of rework and test," *MICROWAVE JOURNAL-EUROGLOBAL EDITION-*, vol. 42, pp. 116–122, 1999.

[54] Fisher, S. C., Rahman, S. A., and Center, J. C. S. S., "Remembering the giants," 2009.

[55] Fragola, J. R., "How safe must a potential crewed launcher be demonstrated to be before it is crewed?," *Journal of Loss Prevention in the Process Industries*, vol. 22, no. 5, pp. 657–663, 2009.

[56] Freaner, C. W., Bitten, R., Bearden, D., and Emmons, D., "An assessment of the inherent optimism in early conceptual designs and its effect on cost and schedule growth," *Aerospace Corporation, Presented to the Planetary Science Subcommittee, NASA Advisory Council*, 2008.

[57] Frey, D., Palladino, J., Sullivan, J., and Atherton, M., "Part count and design of robust systems," *Systems engineering*, vol. 10, no. 3, pp. 203–221, 2007.

[58] Fries, A. and Sen, A., "A survey of discrete reliability-growth models," *Reliability, IEEE Transactions on*, vol. 45, no. 4, pp. 582–604, 1996.

[59] Gebicke, M., "Space missions require substantially more funding than initially estimated," tech. rep., GAO/NSAID-93-97. General Accounting Office. Washington DC, 1992.

[60] Goracke, B. D. and Meisl, C. J., "Launch vehicle engine development in hindsight," in *Space technology and applications international forum*, vol. 361, pp. 653–658, AIP Publishing, 1996.

[61] Haimes, Y. Y., Kaplan, S., and Lambert, J. H., "Risk filtering, ranking, and management framework using hierarchical holographic modeling," *Risk Analysis*, vol. 22, no. 2, pp. 383–397, 2002.

[62] Hall, J. B. and Mosleh, A., "An analytical framework for reliability growth of one-shot systems," *Reliability Engineering & System Safety*, vol. 93, no. 11, pp. 1751–1760, 2008.

[63] Hall, J. B. and Mosleh, A., "A reliability growth projection model for one-shot systems," *Reliability, IEEE Transactions on*, vol. 57, no. 1, pp. 174–181, 2008.

[64] Havskjold, G., "Developing innovative products on budget and on schedule–part 1: Identifying and measuring cost drivers correlates technical uncertainty with rework cycles. aiaa joint propulsion conference," in *45th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, p. 5436, 2009.

[65] Havskjold, G., "Developing innovative products on budget and on schedule–part 2: Using prodecol charts to control development of an innovative advanced technology system," in *45th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, p. 5437, 2009.

[66] Havskjold, G. and by Institutions, F., "Developing innovative products on budget and on schedule–part 3: Generating the prodecol diagram," in *45th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, p. 5438, 2009.

[67] Hertzfeld, H. R., Williamson, R. A., and Peter, N., "Launch vehicles: An economic perspective," *George Washington University Space Policy Institute, September*, 2005.

[68] Hoppe, M., Engel, A., and Shachar, S., "Systest: Improving the verification, validation, and testing processassessing six industrial pilot projects," *Systems Engineering*, vol. 10, no. 4, pp. 323–347, 2007.

[69] Huang, H.-Z. and Zhang, X., "Design optimization with discrete and continuous variables of aleatory and epistemic uncertainties," *Journal of mechanical design*, vol. 131, no. 3, p. 031006, 2009.

[70] Johnson, J., "Fault propagation timing analysis to aid in the selection of sensors fro health management systems," 2008.

[71] Juran, J. M., "Basic concepts," in *Quality control handbook*, p. 2, McGraw-Hill New York, 1974.

[72] Karniel, A. and Reich, Y., "From dsm-based planning to design process simulation: a review of process scheme logic verification issues," *Engineering Management, IEEE Transactions on*, vol. 56, no. 4, pp. 636–649, 2009.

[73] Keller, S., Collopy, P., and Componation, P., "What is wrong with space system cost models? a survey and assessment of cost estimating approaches," *Acta Astronautica*, vol. 93, pp. 345–351, 2014.

[74] Krevor, Z. C., *A methodology to link cost and reliability for launch vehicle design*. PhD thesis, Citeseer, 2007.

[75] Krishnan, V., Eppinger, S. D., and Whitney, D. E., "Iterative overlapping: Accelerating product development by preliminary information exchange.," *ASME DES ENG DIV PUBL DE., ASME, NEW YORK, NY(USA), 1993,*, vol. 53, pp. 223–231, 1993.

[76] Kusiak, A., Wang, J. R., He, D. W., and Feng, C.-X., "A structured approach for analysis of design processes," *Components, Packaging, and Manufacturing Technology, Part A, IEEE Transactions on*, vol. 18, no. 3, pp. 664–673, 1995.

[77] Larson, W. J. and Wertz, J. R., "Space mission analysis and design," tech. rep., Microcosm, Inc., Torrance, CA (US), 1992.

[78] Larson, W., Kirkpatrick, D., Ryan, R., and Weyers, V., *Space Launch and Transportation Systems*. 2005.

[79] Launius, R. D. and Jenkins, D. R., *To Reach the High Frontier: A History of US Launch Vehicles*. University Press of Kentucky, 2015.

[80] Lee, R. A. and Steincamp, J. W., "Reliability analysis techniques for engine-out failures in main propulsion systems," in *AIAA 92-1337, AIAA space prgrams and technologies conference*, 1992.

[81] Lee, W.-S., Grosh, D. L., Tillman, F. A., and Lie, C. H., "Fault tree analysis, methods, and applications a review," *Reliability, IEEE Transactions on*, vol. 34, no. 3, pp. 194–203, 1985.

[82] Leonard, B., *GAO Cost estimating and assessment guide: best practices for developing and managing capital program costs.* DIANE Publishing, 2009.

[83] Lévárdy, V., Hoppe, M., Wenzel, S., and Vollerthun, A., "Process modeling procedure for verification, validation and testing planning," in *Proceedings of the ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2003/DTM-48682) Chicago, USA*, 2003.

[84] Liu, H.-C., Liu, L., and Liu, N., "Risk evaluation approaches in failure mode and effects analysis: A literature review," *Expert systems with applications*, vol. 40, no. 2, pp. 828–838, 2013.

[85] Loch, C. H., Terwiesch, C., and Thomke, S., "Parallel and sequential testing of design alternatives," *Management Science*, vol. 47, no. 5, pp. 663–678, 2001.

[86] Logsdon, J. M., Launius, R. D., Onkst, D. H., and Garber, S. J., "Exploring the unknown: selected documents in the history of the us civilian space program," *NASA*, no. 19980202741, 1998.

[87] London III, J. R., "Leo on the cheap. methods for achieving drastic reductions in space launch costs.," tech. rep., DTIC Document, 1994.

[88] Malcolm, D. G., Roseboom, J. H., Clark, C. E., and Fazar, W., "Application of a technique for research and development program evaluation," *Operations research*, vol. 7, no. 5, pp. 646–669, 1959.

[89] McFadden, R. H. and Shen, Y., "An analysis of the historical reliability of us liquid-fuel propulsion systems," 1990.

[90] McManus, H. and Hastings, D., "3.4. 1 a framework for understanding uncertainty and its mitigation and exploitation in complex systems," in *INCOSE International Symposium*, vol. 15, pp. 484–503, Wiley Online Library, 2005.

[91] McManus, H. L., Hastings, D. E., and Warmkessel, J. M., "New methods for rapid architecture selection and conceptual design," *Journal of Spacecraft and Rockets*, vol. 41, no. 1, pp. 10–19, 2004.

[92] Meisl, C. J., "Life-cycle-cost considerations for launch vehicle liquid propellant rocket engine," *Journal of Propulsion and Power*, vol. 4, no. 2, pp. 118–126, 1988.

[93] Mettas, A., "Reliability allocation and optimization for complex systems," in *Reliability and Maintainability Symposium, 2000. Proceedings. Annual*, pp. 216–221, IEEE, 2000.

[94] Modarres, M., *Risk analysis in engineering: techniques, tools, and trends.* CRC press, 2006.

[95] Moncrieff, J., Malhi, Y., and Leuning, R., "The propagation of errors in long-term measurements of land-atmosphere fluxes of carbon and water," *Global change biology*, vol. 2, no. 3, pp. 231–240, 1996.

[96] Morgan, M. G., Henrion, M., and Small, M., *Uncertainty: a guide to dealing with uncertainty in quantitative risk and policy analysis.* Cambridge University Press, 1992.

[97] Morse, E., Fragola, J. R., and Putney, B., "Modeling launch vehicle reliability growth as defect elimination," in *AIAA Space 2010 Conference & Exposition*, p. 8836, 2010.

[98] Muessig, P. R., Laack, D. R., and Wrobleski, J., "Optimizing the selection of vv&a activities a risk/benefit approach," in *wsc*, pp. 60–66, IEEE, 1997.

[99] Nagano, S., "Space systems verification program and management process," *Systems Engineering*, vol. 11, no. 1, pp. 27–38, 2008.

[100] NASA, "Preliminary report regarding nasa's space launch system and multipurpose crew vehicle," 2011.

[101] NASA, N., "Cost estimating handbook," *Washington, DC*, 2008.

[102] NASA, N., "Schedule management handbook," *National Aeronautics and Space Administration NASA Headquarters, Washington, DC*, 2010.

[103] O'Connor, P. and Kleyner, A., *Practical reliability engineering.* John Wiley & Sons, 2011.

[104] of America: Department of Defense, U. S., *Military Handbook: Reliability Prediction of Electronic Equipment: MIL-HDBK-217F: 2 December 1991.* Department of defense, 1991.

[105] of Defense, U. D., "Military standard: Procedures for performing a failure mode, effects and criticality analysis," *MIL-STD-1629A*, 1980.

[106] of Defense, U. D., "Test requirements for ground equipment and associated computer software supporting space vehicles," *MIL-STD-1833*, 1989.

[107] of Defense, U. D., "Military standard: System safety program requirements," *MIL-STD-882C*, 1993.

[108] of Defense, U. D., "Test requirements for launch, upper-stage, and space vehicles," *MIL-STD-1540*, 2006.

[109] of US Human Spaceflight Plans Committee, R., Augustine, N. R., Austin, W. M., Bajmuk, B. I., Chiao, L., Chyba, C., Crawley, E. F., Greason, J. K., Kennel, C. F., Lyles, L. L., and others, *Seeking a*

human spaceflight program worthy of a great nation. National Aeronautics and Space Administration, 2009.

[110] Osborne, S. M., *Product development cycle time characterization through modeling of process iteration*. PhD thesis, Massachusetts Institute of Technology, 1993.

[111] Otto, K. N. and Antonsson, E. K., "Trade-off strategies in engineering design," *Research in Engineering Design*, vol. 3, no. 2, pp. 87–103, 1991.

[112] Özbayrak, M., Akgün, M., and Türker, A., "Activity-based cost estimation in a push/pull advanced manufacturing system," *International journal of production economics*, vol. 87, no. 1, pp. 49–65, 2004.

[113] Öztaş, A. and Ökmen, Ö., "Judgmental risk analysis process development in construction projects," *Building and Environment*, vol. 40, no. 9, pp. 1244–1254, 2005.

[114] Paté-Cornell, E. and Dillon, R., "Probabilistic risk analysis for the nasa space shuttle: a brief history and current work," *Reliability Engineering & System Safety*, vol. 74, no. 3, pp. 345–352, 2001.

[115] Paté-Cornell, M. E., "Uncertainties in risk analysis: Six levels of treatment," *Reliability Engineering & System Safety*, vol. 54, no. 2, pp. 95–111, 1996.

[116] Pecht, M. and Kang, W.-C., "A critique of mil-hdbk-217e reliability prediction methods," *Reliability, IEEE Transactions on*, vol. 37, no. 5, pp. 453–457, 1988.

[117] Pegden, C. D., "Introduction to simio," in *Simulation Conference, 2008. WSC 2008. Winter*, pp. 229–235, IEEE, 2008.

[118] Plunkett, J. and Dale, B., "A review of the literature on quality-related costs," *International Journal of Quality & Reliability Management*, vol. 4, no. 1, pp. 40–52, 1987.

[119] Pritsker, A. A. B., "Gert," 1966.

[120] Rahman, S. A. and Hebert, B. J., "Large liquid rocket testing–strategies and challenges," in *41st Joint Propulsion Conference, Tucson, AZ. AIAA Paper*, vol. 3564, 2005.

[121] Ramirez, C. and Rahman, S., "An analysis of rocket propulsion testing costs," in *46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, p. 6797.

[122] Rao, K. D., Kushwaha, H., Verma, A. K., and Srividya, A., "Quantification of epistemic and aleatory uncertainties in level-1 probabilistic safety assessment studies," *Reliability Engineering & System Safety*, vol. 92, no. 7, pp. 947–956, 2007.

[123] Robertson, B. E., "A hybrid probabilistic method to estimate design margin," 2013.

[124] Roemer, T. A., Ahmadi, R., and Wang, R. H., "Time-cost trade-offs in overlapped product development," *Operations Research*, vol. 48, no. 6, pp. 858–865, 2000.

[125] Rogers, J. L., *DeMAID: A Design Manager's Aide for Intelligent Decomposition User's Guide*. National Aeronautics and Space Administration, Langley Research Center, 1989.

[126] Rush, C. and Roy, R., "Expert judgement in cost estimating: Modelling the reasoning process," *Concurrent Engineering*, vol. 9, no. 4, pp. 271–284, 2001.

[127] Sandoval-Chavez, D. A. and Beruvides, M. G., "Using opportunity costs to determine the cost of quality: A case study in a continuous-process industry," *The Engineering Economist*, vol. 43, no. 2, pp. 107–124, 1998.

[128] Schrader, S., Riggs, W. M., and Smith, R. P., "Choice over uncertainty and ambiguity in technical problem solving," *Journal of Engineering and Technology Management*, vol. 10, no. 1, pp. 73–99, 1993.

[129] Schrage, D. P., "Technology for rotorcraft affordability through integrated product/process development (ippd)," 1999.

[130] Shabi, J. and Reich, Y., "Developing an analytical model for planning systems verification, validation and testing processes," *Advanced Engineering Informatics*, vol. 26, no. 2, pp. 429–438, 2012.

[131] Shishko, R. and Aster, R., "Nasa systems engineering handbook," *NASA Special Publication*, vol. 6105, 1995.

[132] Singer, C. E. and Dumbacher, D., "Getting to first flight: Equipping space engineers to break the start-stop-restart cycle," in *AIAA SPACE 2010 Conference & Exposition*, p. 8736.

[133] Singer, J. A., Cook, J. R., and Singer, C. E., "Nasa space launch system operations strategy," in *SpaceOps 2012 Conference, AIAA*, 2012.

[134] Slovic, P., Fischhoff, B., and Lichtenstein, S., "Facts and fears: Understanding perceived risk," in *Societal risk assessment*, pp. 181–216, Springer, 1980.

[135] Smith, R. P. and Eppinger, S. D., “A predictive model of sequential iteration in engineering design,” *Management Science*, vol. 43, no. 8, pp. 1104–1120, 1997.

[136] Sörqvist, L., “Poor quality costing,” 1998.

[137] Stamatelatos, M., Dezfuli, H., Apostolakis, G., Everline, C., Guarro, S., Mathias, D., Mosleh, A., Paulos, T., Riha, D., Smith, C., and others, “Probabilistic risk assessment procedures guide for nasa managers and practitioners,” *NASA/SP-2011-3421*, 2011.

[138] Stanley, D., “Nasas exploration systems architecture study (esas), final report,” tech. rep., Technical Report NASA-TM-2005-214062, The National Aeronautics and Space Administration (NASA), 2005.

[139] Steward, D. V., “The design structure system: a method for managing the design of complex systems,” *Engineering Management, IEEE Transactions on*, no. 3, pp. 71–74, 1981.

[140] Strunz, R. and Herrmann, J. W., “Reliability as an independent variable applied to liquid rocket engine hot fire test plans,” *Journal of Propulsion and Power*, vol. 27, no. 5, pp. 1032–1044, 2011.

[141] Swanson, L. A. and Pazer, H. L., “Implications of the underlying assumptions of pert,” *Decision Sciences*, vol. 2, no. 4, pp. 461–480, 1971.

[142] Taneja, V. and Safie, F. M., “An overview of reliability growth models and their potential use for nasa applications,” *NASA STI/Recon Technical Report N*, vol. 93, p. 15433, 1992.

[143] Taylor III, B. W. and Moore, L. J., “R & d project planning with q-gert network modeling and simulation,” *Management Science*, pp. 44–59, 1980.

[144] Thomke, S. and Bell, D. E., “Sequential testing in product development,” *Management Science*, vol. 47, no. 2, pp. 308–323, 2001.

[145] Thunnissen, D. P., *Propagating and mitigating uncertainty in the design of complex multidisciplinary systems.* PhD thesis, California Institute of Technology, 2005.

[146] Tischer, A. and Glover, R., “Studies and analyses of the space shuttle main engine,” *Prepared for NASA George C. Marshall Space Flight Center, Contract No. NAS8-3737, December*, 1987.

[147] Trivailo, O., Sippel, M., and Şekercioğlu, Y., “Review of hardware cost estimation methods, models and tools applied to early phases of space mission planning,” *Progress in Aerospace Sciences*, vol. 53, pp. 1–17, 2012.

[148] Van Hooser, K. and Bradley, D., "Space shuttle main enginethe relentless pursuit of improvement," in *AIAA Space Conference & Exposition*, 2011.

[149] Vesely, W. E., Goldberg, F. F., Roberts, N. H., and Haasl, D. F., "Fault tree handbook," tech. rep., DTIC Document, 1981.

[150] Walpole, R. E., Myers, R. H., Myers, S. L., and Ye, K., *Probability and statistics for engineers and scientists*, vol. 5. Macmillan New York, 1993.

[151] Wang, J. and Lin, Y.-I., "An overlapping process model to assess schedule risk for new product development," *Computers & Industrial Engineering*, vol. 57, no. 2, pp. 460–474, 2009.

[152] Wang, Y.-M., Chin, K.-S., Poon, G. K. K., and Yang, J.-B., "Risk evaluation in failure mode and effects analysis using fuzzy weighted geometric mean," *Expert Systems with Applications*, vol. 36, no. 2, pp. 1195–1207, 2009.

[153] Wiley, *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. John Wiley & Sons, 2015.

[154] Wood, B. K., "Propulsion for the 21st centuryrs-68," in *38th Joint Liquid Propulsion Conference*, 2002.

[155] Woods, D., O'Brien, F., and MacTaggart, K., "The apollo 11 flight journal," 2011.

[156] Worlund, A., Monk, J., and Bachtel, F., "Nls propulsion design considerations," in *AIAA Aerospace Design Conference, Irvine, CA, AIAA Paper*, vol. 1181, pp. 3–6, 1992.

[157] Yassine, A. A., Whitney, D. E., and Zambito, T., "Assessment of rework probabilities for simulating product development processes using the design structure matrix (dsm)," 2001.

[158] Zwack, M. R., "Contrast: A conceptual reliability growth approach for comparison of launch vehicle architectures," 2014.