

DISSERTATION

A NOVEL FRAMEWORK FOR IMPROVING FMEA APPLICATION IN COMPLEX
V-MODEL DEVELOPMENT

Submitted by

Daniel Perreault

Department of Systems Engineering

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Summer 2026

Doctoral Committee:

Advisor: Erika Gallegos

Edwin Chong

Ali Pezeshki

Vincent Paglioni

Copyright by Daniel Perreault 2026

All Rights Reserved

ABSTRACT

A NOVEL FRAMEWORK FOR IMPROVING FMEA APPLICATION IN COMPLEX V-MODEL DEVELOPMENT

Management of design risk has persisted as a topic for as long as formal design processes have existed. In 1949, based on the lessons from World War II development efforts, the U.S. Department of Defense published the first standard for Failure Modes, Effects, and Criticality Analysis (FMECA), which later shortened in common use to Failure Modes and Effects Analysis (FMEA). Over the following 70 years, this approach became arguably the most ubiquitous tool in the design risk toolkit, with standards and application across virtually every aspect of product and process design. FMEA has also become an active topic for academic research, in part because of persistent problems with its application, effectiveness, and usability. Counterintuitively, the bulk of research into the topic has done little to stem the continued reporting of these problems. In fact, the last 35 years have seen continued publication of peer-reviewed articles documenting recurring challenges in FMEA application.

At roughly the same time that these challenges became more visible, the same Department of Defense published standards that contributed to the origins of the systems engineering V-model as a means of managing the growing complexity of modern weapon systems. In short order, this model migrated into software development and then into broader product development, where it has become similarly influential. This dissertation argues that the persistence of FMEA application challenges is not coincidental with the growth of the V-model as a dominant development approach. Rather, many recurring problems in FMEA can be

understood as symptoms of a structural mismatch between a traditionally bottom-up risk analysis method and a top-down staged development model.

This dissertation addresses that argument through three stages of research. First, a literature review is performed to identify recurring criticisms of FMEA and examine commonality in these issues over time. Second, a structural analysis is performed comparing the FMEA methodology, as defined by internationally accepted standards and common practice, against the information structure and staged logic of V-model development. This analysis identifies systemic mismatches between the two methods, including proliferation of failure modes, incompatibility between FMEA information needs and the timing of information availability in the V-model, and difficulty aligning FMEA decomposition with systems engineering decomposition. Third, a novel framework is developed to address these problems while preserving the core value of FMEA as a design risk management tool.

The proposed framework improves FMEA application in complex V-model development through four related elements. First, a New, Unique, and Difficult (NUD) design risk filter is used to identify the areas of a system architecture that warrant focused risk analysis. Second, requirements-based definitions of failure modes and causes are used to align FMEA more directly with systems engineering requirements and architecture. Third, mitigations may be expressed as new or revised requirements and recursively reviewed to determine whether they introduce additional design risk. Fourth, Critical-to-Quality (CtQ) designations are reframed as a mechanism for tracking mitigations and requirements that cross architectural or engineering discipline boundaries.

A key portion of this framework, the NUD design risk filter, is validated in an industrial setting through application to a complex technology development project. The validation trial

demonstrates that the method can be taught and deployed with limited training and facilitation, and that users perceive value in its ability to identify key risk areas in a system architecture. The results provide initial evidence that targeted design risk filtering can help address FMEA proliferation and improve the practical deployment of FMEA in complex V-model environments.

As FMEA continues to be a fundamental expectation of sound design risk practice, the proposed framework does not seek to replace FMEA. Instead, it improves how FMEA is scoped, timed, decomposed, and connected to systems engineering development activities. From a research perspective, the framework opens opportunities for continued improvement in FMEA through integration with Model-Based Systems Engineering (MBSE), requirements management tools, and improved measures of design maturity and design risk management throughout the V-model process.

ACKNOWLEDGEMENTS

I want to acknowledge the assistance of my advisor Dr. Erika Gallegos both as an advisor and a co-author on the publication submissions. As someone coming from industry, academia is a very different world and I have appreciated her guidance and advice.

DEDICATION

Dedicated to my wife Carrie. I'm looking forward to the next 11 years of Saturdays with you.

TABLE OF CONTENTS

ABSTRACT	ii
ACKNOWLEDGEMENTS.....	v
DEDICATION.....	vi
LIST OF TABLES	xi
LIST OF FIGURES	xii
Chapter 1. Introduction	1
1.1. Problem Statement	1
1.2. Research Objective, Aims, and Questions.....	4
1.2.1. Research Aim 1: Identify Challenges of FMEA in V-Model Development.....	4
1.2.2. Research Aim 2: Develop Improved Framework for FMEA Application.....	5
1.2.3. Research Aim 3: Validate Proposed Framework for Improving FMEA	5
Chapter 2. Background on FMEA	7
2.1. Introduction.....	7
2.2. Origins and Evolution of FMEA and V-Model	7
2.3. Failure Mode and Effects Analysis (FMEA)	10
2.3.1. Core Logic of FMEA.....	10
2.4. Mechanics of FMEA	13
2.4.1. Initial Columns: Requirements Decomposition, Identification of Functions and Failures	13
2.4.2. Middle Columns: Scoring	16
2.4.3. Final Columns Mitigations and Evidence	17
2.5. Current Applications of FMEA and Research Directions	19
2.5.1. Span of Application	20
2.5.2. Fuzzy Methods and AI in FMEA	22
2.6. Summary and Research Implications	23
Chapter 3. Aim 1 - Criticisms of FMEA in V-Model Development	25
3.1. Introduction.....	25
3.2. History of Criticisms of FMEA.....	26
3.3. Systems Engineering V-Model	32
3.3.1. Overview	32
3.3.2. Historical Development of the V-Model.....	34
3.3.3. Evolution from DOD-STD-2167A to the V-Model	35
3.3.4. Connecting the V-Model to Systems Engineering as a Development Model.....	37
3.4. Combining the Two: Using FMEA within the V-Model	39
3.5. Implementation Challenges with FMEA	41

3.6. Conclusion	44
Chapter 4. Aim 1 - Systemic Gaps in FMEA Application	45
4.1. Introduction.....	45
4.2. Gaps in the Literature: Missing the Systematic Problems.....	45
4.3. Information Challenges in V-Model Decomposition.....	46
4.4. Case Studies	48
4.4.1. Case Study 1: Motion Compensation for an Imaging System - Inadequate Risk Assessment.....	48
4.4.2. Case Study 2: System Temperature Monitoring - Vague Failure Definition.....	50
4.4.3. Case Study 3: UI Controls Requirement - Unnecessary FMEA	51
4.4.4. Case Study 4: Border Computation in Image - Complexity and Novelty	53
4.4.5. Synthesis	54
4.5. Conclusion	55
Chapter 5. Research Aim 2 – Proposed Solution.....	57
5.1. Introduction.....	57
5.2. Specific Problems to Target for a Solution.....	58
5.3. Requirements	59
5.4. Proposed Framework Overview.....	60
5.5. Design Risk Filtering Using New, Unique, and Difficult Analysis	62
5.5.1. Purpose of the NUD Filter	62
5.5.2. Definition of New	63
5.5.3. Definition of Unique	65
5.5.4. Definition of Difficult	66
5.5.5. Reducing Scope, Cost, and Low-Value Analysis.....	67
5.6. Requirements-Based Definitions for FMEA Information	69
5.6.1. Failure Modes as Requirement Failures.....	70
5.6.2. Causes of Failure as Allocated Architectural Sources.....	70
5.6.3. Improving Consistency in Risk Assessment	71
5.6.4. Contribution to the Proposed Framework.....	72
5.7. Decomposition, Recursion, and Requirements as Mitigations	73
5.7.1. Shifting FMEA Toward System Performance	73
5.7.2. Mitigations as New or Revised Requirements	74
5.7.3. Recursive Application of the NUD Filter	75
5.7.4. Closure of the Recursive Process	76
5.7.5. Design Risk Mapping Across Decomposition	76
5.8. Managing Cross Discipline Complexity Through CtQ Designation	77
5.8.1. Limitation of Traditional CtQ Use	77
5.8.2. CtQ as a Cross Discipline Tracking Mechanism.....	78

5.8.3. CtQ as a Design Coupling Indicator	79
5.9. Relationship to Existing FMEA Improvements	79
5.10. Meta Analysis of Previous FMEA Examples using the new Framework	80
5.10.1. Image Stabilization Feature Case Study Reviewed	81
5.10.2. System Over Temperature Management Failure Reviewed.....	83
5.10.3. UI Image Optimization Controls Reviewed.....	85
5.10.4. Border Computation Case Study Reviewed.....	85
5.11. Conclusion and Proposal for Validation.....	87
Chapter 6. Aim 3 - Validate Design Risk Filtering	89
6.1. Introduction.....	89
6.2. NUD Framework for Validation	90
6.3. Assumptions and Constraints.....	90
6.4. Definitions of New, Unique, and Difficult	91
6.5. Application and Execution of the NUD Framework.....	92
6.6. Methods	96
6.6.1. Case Study Context.....	97
6.6.2. Participants	97
6.6.3. Procedure.....	98
6.6.4. Survey Design and Data Collection.....	99
6.7. Results	100
6.7.1. Participant Background with FMEA	100
6.7.2. Training Effectiveness	101
6.7.3. Perceived Value Relative to Existing Process.....	101
6.7.4. Qualitative Feedback	102
6.8. Discussion.....	104
6.9. Conclusion	106
Chapter 7. Conclusions	107
7.1. Introduction.....	107
7.2. Research Contributions.....	107
7.3. Summary of Research Findings	109
7.3.1. Research Aim 1 Findings: Identify Challenges of FMEA in V-Model Development	109
7.3.2. Research Aim 2 Findings: Develop Improved Framework for FMEA Application .	110
7.3.3. Research Aim 3 Findings: Validate Proposed Framework for Improving FMEA....	112
7.4. Limitations and Future Work.....	113
7.4.1. Limitations.....	113
7.4.2. Future Work	114
7.5. Publications of Results	116

References	118
LIST OF ABBREVIATIONS.....	128

LIST OF TABLES

Table 1: First Set of Columns in a Sample FMEA Table	14
Table 2: Second Set of Columns in a Sample FMEA Table	16
Table 3: Third (and Final) Set of Columns in a Sample FMEA Table	17
Table 4: Comparison of McKinney (1991) Deficiencies vs. MIL-STD-1629A	27
Table 5: Comparison of Issues within FMEA Identified by Three Independent Articles	28
Table 6: Alignment of FMEA Information Needs with Information Available in the V-Model ..	40
Table 7: Sample Row from FMEA for Case Study 1	49
Table 8: Sample Row from FMEA for Case Study 2	51
Table 9: Sample Row from FMEA for Case Study 3	52
Table 10: Sample Row from FMEA for Case Study 4	53
Table 11: Image Stabilization Case Study	81
Table 12: Image Stabilization under New Framework	82
Table 13: System Over Temperature Management Case Study	83
Table 14: System Over Temperature Management Reviewed	84
Table 15: UI Image Optimization Controls Case Study	85
Table 16: Border Computation Algorithm Case Study Case Study	86
Table 17: Border Computation Case Study Reviewed	87
Table 18: Participant Team Representation	98
Table 19: Participant Experience with FMEA.....	100

LIST OF FIGURES

Figure 1: Systems Engineering V-Model Adapted from INCOSE (2015) and Forsberg et al. (2005)	33
Figure 2: Adapted from DOD-STD-2167A Development Logic (U.S. DOD,1988).....	35
Figure 3: Adapted from Rook's (1986) V-Model with Annotations (gray boxes) Overlaid from DOD-STD-2167A (U.S. DOD, 1988).....	36
Figure 4: Adapted from Forsberg & Mooz (1992) V-Model with Annotation (grey boxes) from Rook (1986) and DOD-STD-2167A (U.S. DOD, 1988).....	38
Figure 5: Comparison of V-Model Stages to FMEA Information Requirements	47
Figure 6: Example NUD Flow from Training	96
Figure 7: Participant Evaluation of NUD Framework (n = 6).....	102

Chapter 1. Introduction

1.1. Problem Statement

Failure Modes and Effects Analysis (FMEA) emerged from U.S. military efforts to improve design risk management following World War II. The method was first published by the U.S. Department of Defense (DOD) as MIL-P-1629A (U.S. DOD, 1949) and later updated under MIL-STD-1629A (U.S. DOD, 1980). The purpose of FMEA was, and still is, to identify potential design failures before they occurred, prioritize failure modes, and trace identified risks to mitigations and evidence of effectiveness.

FMEA is now broadly used across product, process, system, reliability, safety, and risk management applications. Failure Modes, Effects, and Criticality Analysis (FMECA) extends the method by incorporating criticality assessment. However, FMEA and FMECA share a common methodology and frequently overlap in practice and standards, and as such, this dissertation uses the term FMEA broadly. Although widely adopted, FMEA remains inherently heuristic and has known limitations related to consistency, scalability, and interpretation (Bouti & Kadi, 1994; Liu et al., 2013). Nevertheless, FMEAs are commonly treated as mandatory deliverables in regulated industries and government contracts.

Over the same period, system development practices evolved in response to increasing system complexity. By the mid-1980s, software had become a standard component of military and commercial systems, requiring more formal coordination between hardware and software development. DOD-STD-2167 and DOD-STD-2167A formalized staged software development practices and contributed to the development logic later associated with the modern V-model (U.S. DOD, 1985; U.S. DOD, 1988). Subsequent work by Rook (1986) and Forsberg and Mooz

(1992) strengthened the relationship between staged development activities and corresponding test, verification, and validation activities.

The parallel development of FMEA and the V-model is important because both methods respond to increasing complexity in engineered systems. However, they do so in different ways. The V-model provides a staged structure for decomposing requirements, developing architecture and design, and building evidence through integration, verification, and validation. FMEA provides a structured way to reason about failures, effects, causes, mitigations, and evidence. While system development models evolved to manage the staged maturation of complex hardware and software systems, the core FMEA process remained largely unchanged.

This difference creates the central problem addressed in this dissertation. FMEA is expected to support design decisions early enough to reduce risk before major commitments are made, but conventional FMEA often depends on information that becomes available only later, including detailed architecture, allocated requirements, specific failure mechanisms, existing controls, mitigation strategies, and verification evidence. In V-model development, where design information matures progressively, this timing mismatch can make FMEA difficult to apply effectively.

When FMEA is performed early, the analysis may rely on incomplete architecture, vague failure definitions, speculative causes, or immature controls. When FMEA is delayed until detailed design information is available, it may occur too late to influence major design decisions without costly rework. In complex systems involving hardware, software, firmware, algorithms, user interaction, and cross-disciplinary interfaces, this mismatch can contribute to unclear scope, proliferation of failure modes, inconsistent analysis, weak traceability to design action, and the perception that FMEA is primarily a compliance activity. It is particularly the case that in

complex systems, the interaction and connection of different layers of system can fail leading to the potential for the failure modes being on one side, the other or associated with the interaction, making these both harder to identify and increased in quantity. This is further complicated by the addition of design reuse, considered a best practice in modern systems engineering. Reuse of design components allows for stability in a new project in areas which are not the focus of new capabilities. However, under classic FMEA approaches, this unbalanced level of design maturity leads teams to focus on failure analysis of known elements where risk is low, distracting limited resources from less mature areas of the design which are still in early development. The emphasis in classic FMEA on completeness drives resources towards the design reuse where design detail is available, as opposed to the portions of the design which represent higher risk, but are less developed. This leads to frustration by the personnel involved, as well as taking time away from the areas where FMEA is most needed. It is true that some cases of design reuse merit FMEA attention, because the design context for the reuse is different from predicates, but classic FMEA processes do not have a process for differentiating these situations.

The problem addressed by this dissertation is that conventional FMEA practice is not adequately structured to align with the staged information availability, decomposition logic, and verification structure of V-model development. This dissertation investigates recurring FMEA challenges in V-model development and proposes a framework to improve how FMEA is scoped, decomposed, timed, and connected to requirements, architecture, mitigations, and verification evidence.

1.2. Research Objective, Aims, and Questions

The objective of this dissertation is to analyze these challenges and develop potential improvements to FMEA, then validate whether those improvements better address recurring issues in its application. Accordingly, the dissertation is organized around three research aims.

1.2.1. Research Aim 1: Identify Challenges of FMEA in V-Model Development

Research Aim 1 examines published criticisms of FMEA to identify recurring patterns and identify systemic misalignments between FMEA practices and the V-model methodology. A literature review indicates that, over the last 30 to 35 years, criticisms of FMEA have been published regularly, with each article proposing some level of corrective action or methodological improvement. In addition, relevant standards provide cautions and recommendations intended to address many known pitfalls. The purpose of analyzing these criticisms is to address the following questions:

- **RQ1.1:** What recurring patterns of problems are evident in published criticisms of FMEA, particularly those that appear resistant to solutions proposed by standards, academia, or industry?
- **RQ1.2:** What underlying systemic causes contribute to recurring FMEA problems when the method is applied within modern development models such as the V-model?

This analysis was presented at and published in the proceedings for the International Council on Systems Engineering (INCOSE) Human Systems Integrations (HSI) Conference (Perreault et al., 2024).

1.2.2. Research Aim 2: Develop Improved Framework for FMEA Application

Research Aim 2 develops an improvement to FMEA that addresses the conflicts identified between classic FMEA and the V-model methodology. Building on the findings from Research Aim 1, this aim recognizes that lower-level improvements to the FMEA process, as evidenced in the literature, have not been effective in producing significant improvements against the identified problems. Therefore, Research Aim 2 approaches the problem by using the findings from the first aim to develop a method that targets the underlying systemic causes and constraints discovered, rather than addressing only isolated procedural weaknesses. The following research questions are addressed by this research aim:

- **RQ2.1:** What high-level requirements define the solution space for improvements to FMEA that address the systemic causes and constraints identified in Research Aim 1?
- **RQ2.2:** How can FMEA be augmented to better align with the V-model based on the requirements developed in Research Aim 2?

The resulting artifact provided case study examples of the systemic problems identified in Research Aim 1 and established a set of requirements and a framework for addressing them. This work was presented at and published in the proceedings of the INCOSE Western States Regional Conference (Perreault & Gallegos, 2025).

1.2.3. Research Aim 3: Validate Proposed Framework for Improving FMEA

Research Aim 3 validates the improved FMEA approach developed in Research Aim 2 by testing the proposed solution in a high-complexity development project and gathering data on its benefits. A key problem consistently identified in the FMEA literature is proliferation, which becomes increasingly problematic as product complexity and feature density increase. Therefore,

this aim evaluates whether the improved framework can help address proliferation in the context of a complex, real-world development effort.

This validation was conducted in collaboration with Access Laser, a small company developing a novel laser system for a specific customer. The laser system differed significantly from Access Laser's prior products, making it an appropriate case for evaluating the improved FMEA process. As part of the collaboration, the author hosted a series of workshops on the proposed framework for Access Laser employees, who then applied the process to the new laser project and provided feedback regarding its usefulness, implementation needs, and perceived value. The following research questions are addressed in this aim:

- **RQ3.1:** How difficult is it to teach the proposed methodology to a team that is entirely unfamiliar with it?
- **RQ3.2:** To what extent do users perceive value in the proposed methodology and express willingness to use it in future projects?
- **RQ3.3:** What level of facilitation and contextually relevant examples are needed to support effective deployment of the proposed methodology?

The results of this research aim have been submitted to a peer-reviewed journal and are under review at the time of this dissertation's submission (Perreault & Gallegos, 2026).

Chapter 2. Background on FMEA

2.1. Introduction

This chapter provides background on FMEA to establish a foundational, common understanding of the method before critiquing and improving upon it in later chapters. The discussion begins with the basic logic of FMEA as a structured approach for identifying potential failures, evaluating their effects, prioritizing risks, and documenting mitigations. It then describes the typical mechanics of FMEA, including the organization of an FMEA table, the use of scoring to prioritize risks, and the role of mitigation and evidence in closing the analysis. Finally, the chapter summarizes current applications of FMEA and recent research efforts to extend or improve the method, particularly through fuzzy methods, expert systems, and artificial intelligence.

2.2. Origins and Evolution of FMEA and V-Model

In 1949, following the extensive military research and development activity undertaken in the United States during World War II, the DOD published MIL-P-1629A, Failure Modes Effects and Criticality Analysis, as a method for improving product development (U.S. DOD, 1949). In particular, the method focused on predicting potential failure modes in designs before those failures occur. This approach was intended to augment the prototype and test model that had been predominant during the war. The method was originally developed to achieve three primary objectives:

- Predict design failures before committing to designs that are costly to change.

- Prioritize potential failure modes using a semi-heuristic, semi-data-driven approach to ensure that the most serious failure modes receive appropriate attention during the development process.
- Provide a means of tracing proposed solutions or mitigations to specific failure modes so they can be tested to confirm that the failure modes have been eliminated.

By design, the process is intended to be iterative and to align with design cycles, allowing teams to evaluate and improve the design before committing to a design approach. It also incorporates prototyping data to support the prioritization of failure modes that need to be eliminated. Despite many generations of standards, including current standards (ISO 2019; IEC, 2018; ISO 2017; IATF, 2016), the core intent and methodology of FMEA remain largely unchanged. FMEA has expanded to incorporate the design of devices (D-FMEA), processes (P-FMEA), usability (U-FMEA) and specific subcategories, such as safety elements of devices, or service processes. The range of application areas has also become broad enough that FMEA is now considered a core element of risk management, reliability, device design, and process design.

By the mid 1980's, the development of software as part of products for the U.S. military had become standard. The combination of software and hardware development created a level of complexity that required an improved development model for system software, which the DOD published in 1985 (U.S. DOD, 1985). This standard documented the first origins of the modern V-model for development and aligned hardware and software development. Only a year later, Paul Rook published what is considered the full origin of the V-model (Rook, 1986). Rook's work augmented MIL-STD-2167 (U.S. DOD, 1985), Defense System Software Development, by integrating testing at each level of the development cycle to support proven and controlled design

maturity throughout the process. Similar to FMEA, the V-model has become a widely used standard method for managing complex, mixed-discipline engineering development projects. Also similar to FMEA, it has become an expectation in many regulated industries, such as medical devices, where it has a clear connection to governing regulations related to design controls (U.S. FDA, 2026). As another similarity to FMEA development, the V-model has remained remarkably unchanged in its fundamental principles. As technology has changed, specifics of implementation have improved, and the model has been applied to the development of a wide range of systems, but the underlying methodology remains the same. Some recent research efforts have explored approaches to integrate hybrid methodologies into the V-model, borrowing from the agile development space to allow for more iterative and responsive development, although these adaptations generally preserve the V-model's core emphasis on requirements traceability, verification, validation, and disciplined progression through development stages (Gracias & Gallegos, 2025; Gracias & Gallegos, 2024).

While there have been challenges with the application of FMEA since its inception, academic, government, and industry attention to these challenges has increased over the last 35 years. Peer reviewed literature, as well as industry periodicals, have regularly documented issues with FMEA application and suggested improvements to different aspects of the method, often with little lasting impact. Some of these improvements have been incorporated into standards over the years, in many cases leading to challenges greater than the problems they were intended to solve.

2.3. Failure Mode and Effects Analysis (FMEA)

2.3.1. Core Logic of FMEA

FMEA is a structured, bottom-up risk analysis methodology used to identify possible failure modes for a system, subsystem, process or component, evaluate the effects of those failures, and prioritize actions to reduce risk. The approach is typically implemented in a tabular format, where subject matter experts evaluate each failure mode using three primary risk factors: severity, occurrence, and detectability (Stamatis, 2003; IEC, 2018). These factors are generally assessed using ordinal scales and combined, most commonly through multiplication, into a Risk Priority Number (RPN), which is used to rank risks and prioritize mitigation actions. Its purpose is not simply to document possible failures, but to support design and process decision-making by making potential failure pathways visible before they occur in use. In this way, FMEA provides a systematic means of connecting system functions, potential failure modes, causes, effects, existing controls, and mitigations.

The core FMEA methodology was introduced by the U.S. Military 77 years ago (U.S. DOD, 1949). Once the administrative groundwork is complete (e.g., putting together a team, securing resources, etc.), the classic methodology is deceptively simple. It can be summarized by the following steps:

1. Select a system or subsystem for analysis.
2. Determine the functionality of the system in question.
3. Brainstorm ways that the system might fail to function, noting that there may be nuances in the failure, such as the degree to which a function is violated.

4. Determine what impact the failure might have on a user or dependent system, noting that the impact may differ based on the degree or exact nature of the failure.
5. Analyze what in the system could cause that failure to occur, noting that there may be many possible causes.
6. Apply a scoring rubric. Typically, this involves three orthogonal factors: (1) the severity of the failure, based on its impact; (2) the likelihood of the failure, based on the probability of the cause; and (3) the detectability of the failure, based on the premise that a user would seek to avoid the impact if they were aware of it. These scoring rubrics generally lead to a single value for each failure.
7. Sort the failures based on the scoring to create a priority order. In general, the scoring rubric includes various levels of required action depending on the priority score. Some scores may require no action, while others, such as those associated with safety-related failures, may require extensive analysis to address the scoring.
8. Perform one or more design improvements to mitigate the causes of failures and, therefore, typically reduce the probability of failures. In general, it is much more difficult to reduce the severity and much less effective to improve the detectability. However, depending on the circumstances, any or all of the three scoring factors may be improved.
9. Rescore the post mitigated design against the identified failures to determine the level of improvement. In principle, mitigation work should continue until the rescored value is below the predetermined action limit. As a result, more than one mitigation may exist for a given cause of failure.

10. Document the analysis, typically in tabular form, though other approaches also exist.

When all intended mitigations have been incorporated into the design and proven effective through appropriate testing or verification activities, the final FMEA update can be completed and the process closed for a given project. However, essentially all standards and guidelines recommend that this work be maintained throughout the life cycle of the product along with sustaining data so that practitioners do not start over with each new project.

Since its origin, with the growth in FMEA use, dozens of standards, texts, and handbooks (NASA, 2024; U.S. FDA, 2023; SAE, 2021; AIAG & VDA, 2019; ISO 2019; IEC, 2018; ISO 2017; IATF, 2016; Stematis, 2003) have been published to formalize and refine the method. However, the fundamental logic, process, and execution of FMEA have not substantially changed. A review of modern texts on the subject shows that, in many areas, the methods remain unchanged from the original work in 1949. Refinements from standards and literature have generally focused on the following areas:

- *Description of the organizational commitment required for FMEA.* Essentially all standards, guidance documents, and texts discussing FMEA emphasize the need for cross-functional teams, early initiation of the FMEA process during design, careful planning of the FMEA, and continued updates throughout design maturation.
- *Development of different algorithms for performing risk assessment.* Spreafico et al. (2017) provide a literature review identifying a wide range of research into different scoring models. AIAG & VDA (2019) augmented the classic scoring model, the Risk Priority Number, or RPN, with a hierarchy of scores to provide a form of tie-breaking when identical scores occur. Some approaches have become sufficiently complex to incorporate linguistic models as part of the scoring process (Liu et al., 2014).

- *New applications of FMEA.* FMEA research has been published in areas ranging from nuclear reactor reliability (Guimaraes & Lapa, 2004) to medication safety (Anjalee et al., 2021), with no indication that the expansion of applications is slowing.

Anecdotally, discussions with peers in this research area indicate increasing interest in using artificial intelligence (AI) in FMEA, further extending the next generation of applications for the tool.

2.4. Mechanics of FMEA

The process of conducting an FMEA can be broken down into three major steps: (1) planning the FMEA, which includes defining the objectives, scope, system boundaries, decision criteria, and documentation; (2) performing the analysis, which includes subdividing the system into elements, identifying functions, failure modes, failure causes, scoring failures, and defining actions; and (3) documenting the FMEA, which includes synthesizing the results and generating a living document (IEC, 2018).

2.4.1. Initial Columns: Requirements Decomposition, Identification of Functions and Failures

Typically, the output of the FMEA process is documented in tabular form, though many variants and alternatives are proposed in the research literature. For illustration, an example FMEA table is presented in three sections of columns: Table 1, Table 2, and Table 3. In a full FMEA, these columns would be bound together. Each of the columns across these tables represents items associated with Step 2, “Perform the analysis,” as described above. Each table includes the FMEA column header, a brief description of the column, and an illustrative example related to a cooling fan used to remove heat from internal electronics.

The first eight items are shown in Table 1 and represent decomposition of the system and identification of functions and failures. The first row provides the column headers, and the second row provides the typical information that would be populated to support those headers.

Table 1: First Set of Columns in a Sample FMEA Table

ID	Element	Functions & Performance	Failure Mode	Local Effect of Failure	Final Effect of Failure	Cause of Failure	Existing Controls
#	<i>Component of the system</i>	<i>What the element does</i>	<i>How the element fails to meet its function</i>	<i>What happens at the element level</i>	<i>What happens at the system/ user level</i>	<i>What creates the failure mode</i>	<i>Mitigations that already exist</i>
1	Cooling fan	Remove heat from internal electronics during operation	Fan does not provide sufficient airflow	Internal electronics exceed target operating temp	Device shuts down or has reduced service life	Fan motor failure or blocked airflow path	Thermal sensor monitors internal temp

It is important to note that these columns represent the initial state before new design mitigations are applied. While there often are iterations to this portion of the FMEA, at some point it is expected to be completed such that the remainder of the process can be executed. A more detailed description of these columns is as follows:

- *ID*: A unique number for the line. Generally, it is only unique within a given table to differentiate lines.
- *Element*: A discrete portion of the system identified for analysis.

- *Functions & Performance:* The expected behavior of the element, often including acceptance criteria for this behavior.
- *Failure Mode:* A way that the element can violate the stated function and performance expectations. This does not represent what failed in the system, but rather the unacceptable behavior that may occur.
- *Local Effect of Failure:* The element-level effect of the identified failure mode. For example, if a circuit board signal is expected to trigger a software monitoring function and the circuit board fails to send the signal, the local effect would be that the software monitoring function is not triggered.
- *Final Effect of Failure:* The system-level and/or user-level effect of the failure mode, or how the failure propagates through the system. The intent is to assess the chain of events leading to an effect on the system or user. Continuing the previous example, if the software monitoring function is not triggered, a warning to the user may not be presented regarding the system's error state.
- *Cause of Failure:* The physical or digital problem in the element that creates the failure mode. Continuing the circuit board example, this could include damage to a component on the board.
- *Existing Controls:* Proven controls that already exist in the new design. Most commonly, these come from well-proven elements in a predicate product or from well-established off-the-shelf designs that already include proven controls for a possible failure mode.

2.4.2. Middle Columns: Scoring

The next three typical elements, which comprise the initial state scoring table, are presented in Table 2, along with typical values to continue the example. These are used to calculate the RPN, which is used to sort the failure modes. RPN is typically calculated by multiplying the numeric scores for each category. Severity, occurrence, and detection commonly range from 1 to 10, meaning RPN can range from 1 to 1000.

Table 2: Second Set of Columns in a Sample FMEA Table

Severity Score	Occurrence Score	Detection Score	RPN (S x O x D)
<i>Impact of the failure effect on the system or user</i>	<i>Likelihood that the failure mode will occur</i>	<i>Ability to detect the failure mode before the final effect occurs</i>	<i>Prioritization score calculated from severity, occurrence, and detection</i>
10	5	10	500

For both severity and detection, the scoring is related to the “Failure Mode” column rather than “Cause of Failure” column. This accommodates circumstances in which there may be a many-to-one relationship between causes and failure modes. These columns can be explained in more detail as follows:

- *Severity Score*: This is related to the “Final Effect of Failure” column from Table 1 and is designed to measure the impact of the effect on the system if the failure mode occurs.
- *Occurrence Score*: This is related to the “Cause of Failure” and the likelihood of the Cause of Failure occurring. The premise being that if the Cause of Failure occurs then

Failure Mode occurs. Thus, in many to one relationships between Cause of Failure and Failure Mode, the true likelihood of the Failure Mode is a function of the combination of the likelihoods of all causes.

- *Detection Score*: This is related to the “Failure Mode” and the ability to detect it in time to mitigate the “Final Effect of Failure.” It is important to note that the detection score is typically inverted in comparison to the other factors. For example, on a 10-point scale, a score of 10 would indicate the lowest level of detectability, while a score of 1 would indicate the highest level of detectability.
- *RPN*: This is used to prioritize the risks, with higher values indicating a higher priority for mitigation. Since the resulting RPN value is based on a combination of factors, the scales are typically the same for all factors to avoid structurally weighting one factor over another.

2.4.3. Final Columns Mitigations and Evidence

Lastly, Table 3 presents the final set of columns in the FMEA, which relate to mitigation actions in response to identified tasks.

Table 3: Third (and Final) Set of Columns in a Sample FMEA Table

Future State Mitigation	Post Mitigated Severity Score	Post Mitigated Occurrence Score	Post Mitigated Detection Score	Post Mitigated RPN	Evidence of Effectiveness	Coding
<i>Design change or action to reduce risk</i>	<i>Severity score after mitigation</i>	<i>Occurrence score after mitigation</i>	<i>Detection score after mitigation</i>	<i>Updated RPN after mitigation</i>	<i>Evidence mitigation is effective</i>	<i>Tracking info</i>

Add airflow obstruction detection	10	1	1	10	Verification test confirms shutdown warning before overheating	Safety related
--	----	---	---	----	---	-------------------

To complete an FMEA, the expectation is that, for the appropriately selected high-RPN score lines, the design is modified to introduce a mitigation. The columns presented in Table 3 can be further described as follows:

- *Future State Mitigation:* These are design changes made to address the “Failure Mode” identified in Table 1. Typical examples include design redundancy or the use of higher-grade components to reduce the “Occurrence Score.” It is typically not the case that design mitigations can reduce “Severity Scores,” because severity is scored based on the assumption that the failure mode occurs.
- *Scoring Columns:* These represent the score for the line after the mitigation in the first column is applied. Essentially, they represent the expected effect of the design change on the “Failure Mode,” which is presumed to be an improvement. Typically, improvements are identified that reduce the opportunities for failure by reducing the likelihood that the “Cause of Failure” from Table 1 occurs. This, in turn, reduces the “Occurrence Score” for the “Failure Mode.”
- *Evidence of Effectiveness:* Particularly in regulated industries, FMEA can be an essential aspect of design controls. Therefore, some form of formal verification of the mitigation’s effectiveness is often performed.

- *Additional Coding:* Similarly, there may also be additional coding elements, where designators such as “critical to safety” are placed on the mitigation. These designators may lead to additional activities to ensure that the mitigation is present in the product.

At this point, the FMEA process is considered complete for a given design or set of designs. However, all standards dating back to the origins of FMEA commonly state that the FMEA should be a living object. It should be referenced and updated when design changes are made, when aspects of the design are incorporated into a new design, when new failure modes are discovered, or when the scoring proves incorrect (U.S. DOD, 1980; U.S. DOD, 1949). Thus, while there is an expected completion point for a given FMEA analysis, the FMEA is also expected to be reused throughout the life of the system.

2.5. Current Applications of FMEA and Research Directions

As discussed previously, FMEA has become nearly ubiquitous in risk analysis and reliability analysis. Its continued use across regulated industries, safety-critical systems, healthcare, design optimization, and commercial applications demonstrates the adaptability of the method. However, recent research also reveals a recurring pattern, although FMEA remains valuable as a basic framework for identifying and prioritizing failure-related risk, researchers frequently modify or extend the method to address limitations in conventional scoring, prioritization, scalability, and knowledge management. Thus, the current literature demonstrates both the continued relevance of FMEA and the persistence of challenges in its practical application.

2.5.1. Span of Application

FMEA has become a de facto requirement in many regulated industries. In the automotive industry, the Automotive Industry Action Group (AIAG), which includes major U.S. and Japanese automobile manufacturers, collaborated with Verband der Automobilindustrie e. V. (VDA), which represents major German automobile manufacturers, to create a common standard for FMEA application in 2019 (AIAG & VDA, 2019). This collaboration is notable because the represented companies are global competitors, yet they agreed to establish a common approach to FMEA rather than develop independent methodologies that could be used for competitive advantage. Other industries rely on standards such as IEC 60812, which was most recently updated in 2018 (IEC, 2018).

Beyond these regulated contexts, recent research demonstrates the continued expansion of FMEA into new application areas. Across these applications, researchers tend to preserve the basic FMEA framework while modifying aspects of the method to fit the needs of a particular domain. For example, Ayyildiz et al. (2025) apply FMEA to university laboratory safety and argue that conventional methods, including classic FMEA, are insufficient for assessing and preventing laboratory risks. Their use of fuzzy logic techniques reflects a broader trend in which researchers retain FMEA's underlying structure while modifying the risk assessment portion of the method.

A similar pattern appears in healthcare and maritime safety applications. En-Naaoui et al. (2024) apply FMEA to hospital sterilization risks and identify limitations in conventional FMEA scoring models. Their approach uses fuzzy inference for risk evaluation and extends into decision-making around “risk tolerance” through an artificial neural network algorithm. Ceylan (2024) similarly applies FMEA to safety risks associated with marine diesel engine turbocharger

fouling and uses fuzzy analysis and expert systems to improve prioritization of causal structures. These studies suggest that FMEA remains useful for structuring risk analysis, but that conventional scoring and prioritization methods are frequently viewed as insufficient for complex or specialized applications.

FMEA has also been extended beyond traditional safety-related contexts. Singer and Ozcelik (2024), for example, apply the FMEA framework to classify and optimize stainless steel properties for biomedical applications. Their work again treats classic scoring models as inadequate and uses a fuzzy FMEA approach, followed by an intuitionistic fuzzy TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution)-sort method, to support prioritization. This reflects a recurring concern in the literature regarding the ordinal structure of traditional RPN and the mathematical limitations of using RPN as the primary basis for prioritization.

Commercial applications show a similar pattern. Kadena et al. (2022) apply fuzzy FMEA methods to customer dissatisfaction and cell phone replacement decisions. In this application, the cell phone is treated as a system in which subsystems can produce customer-facing failures or dissatisfiers. This use of FMEA differs somewhat from the traditional bottom-up concept of FMEA, which typically begins at the component failure level. However, the authors primarily focus on improving the scoring and prioritization model rather than examining the broader implications of adapting FMEA to a different system framing.

Overall, the span of recent FMEA applications demonstrates the flexibility and durability of the method. Researchers continue to apply FMEA across increasingly diverse domains, suggesting that the basic logic of identifying functions, failure modes, effects, causes, and mitigations remains useful. However, these applications also repeatedly identify limitations in

conventional FMEA, particularly in scoring, prioritization, and the treatment of complex causal relationships.

2.5.2. Fuzzy Methods and AI in FMEA

A common theme across the recent literature is the use of fuzzy methods, expert systems, and artificial intelligence to improve specific aspects of FMEA application. These approaches are typically not intended to replace FMEA. Instead, they attempt to improve how FMEA is executed, especially in areas where conventional practice is viewed as too subjective, labor-intensive, difficult to scale, or mathematically limited.

Fuzzy methods are especially common in efforts to improve FMEA scoring and prioritization. These methods generally seek to address concerns about the use of ordinal scoring scales, the multiplication of severity, occurrence, and detection scores into RPN, and the difficulty of representing uncertainty in expert judgment. Across safety, healthcare, biomedical, and commercial applications, fuzzy FMEA approaches are used to improve the consistency or mathematical defensibility of prioritization while preserving the broader FMEA framework.

More recently, AI and large language models (LLMs) have been explored as tools for improving FMEA efficiency, completeness, and knowledge reuse. Xia et al. (2024) examine the use of LLMs to enhance and mine FMEAs for improved risk management in technical processes and products. Their prototype uses an LLM agent to enhance an FMEA knowledge database, with the intent of reducing cost, improving completeness, and addressing limitations in expertise. Notably, the expertise limitation discussed in this work is framed less as a lack of individual competence and more as a problem of organizational communication and access to knowledge, which aligns with the long-standing emphasis on cross-functional participation in FMEA.

Hassani et al. (2025) similarly examine the use of LLMs in FMEA and propose a human-in-the-loop model to address collaboration, complexity, and scalability issues. Their work emphasizes the potential for LLMs to reduce manual effort, support large datasets, improve prioritization, and maintain knowledge reuse across differences in engineering experience. This focus is particularly relevant to FMEA proliferation, where the size and complexity of the analysis can become difficult to manage as products and systems become more complex. Cognitive overload of human supervisors and the need for transparent, explainable, and trustworthy AI therefore become necessary considerations for supporting effective human-AI collaboration in complex engineering decision processes (Ortolano & Gallegos, 2025; Botyarov & Gallegos, 2025).

Research on fuzzy methods, expert systems, and AI suggests that FMEA continues to be viewed as valuable, but also difficult to execute effectively in modern development contexts. Much of the literature attempts to improve specific application problems, especially scoring, prioritization, knowledge reuse, and scalability. However, these improvements often leave the underlying structure of FMEA largely intact. As a result, they may address symptoms of FMEA difficulty without fully addressing the systemic sources of recurring problems in the method's application.

2.6. Summary and Research Implications

The recent literature demonstrates that researchers continue to recognize both the value and the limitations of FMEA. On one hand, FMEA remains widely used and adaptable across regulated industries, safety-related applications, healthcare, design optimization, and commercial decision-making. On the other hand, the same literature repeatedly identifies problems in

execution, particularly related to scoring, prioritization, uncertainty, scalability, knowledge reuse, and proliferation.

This pattern is important because it suggests that many FMEA improvement efforts focus on lower-level methodological modifications rather than the broader system of constraints in which FMEA is applied. Spreafico et al. (2017) similarly identify that much of the research focuses on specific application problems while paying less attention to broader systemic issues with the methodology overall. Therefore, the systemic sources of recurring FMEA problems represent the core focus this dissertation seeks to address.

Chapter 3. Aim 1 - Criticisms of FMEA in V-Model Development

3.1. Introduction

Research Aim 1 examines recurring criticisms of FMEA to determine whether persistent problems in its application reflect isolated execution issues or deeper systemic constraints in the method. Although FMEA has been widely used for more than 70 years, published critiques repeatedly identify similar challenges, including incomplete failure identification, inconsistent severity classification, inadequate use of data, narrow scope definition, late application, and difficulty translating results into effective design action. Many of these issues have also been anticipated in standards, handbooks, and guidance documents, suggesting that their persistence cannot be explained solely by a lack of procedural instruction.

This chapter presents the first part of the Research Aim 1 analysis. It reviews the history of published criticisms of FMEA, identifies recurring patterns across independent sources, and examines how those criticisms relate to existing FMEA standards and guidance. The chapter then introduces the systems engineering V-model and evaluates whether the recurring problems may be associated with a mismatch between the information required to perform FMEA and the timing of information availability within V-model development. The analysis presented in this chapter provides the foundation for the subsequent chapter, which further examines how this systemic mismatch is treated in the literature and how it appears in practical FMEA application. These chapters together address the following research questions:

- **RQ1.1:** What recurring patterns of problems are evident in published criticisms of FMEA, particularly those that appear resistant to solutions proposed by standards, academia, or industry?

- **RQ1.2:** What underlying systemic causes contribute to recurring FMEA problems when the method is applied within modern development models such as the V-model?

3.2. History of Criticisms of FMEA

Over more than 70 years of use across many industries, practitioners and researchers have identified persistent challenges in the application of FMEA. These criticisms are notable not only because they recur across time, but also because many of the identified problems are addressed, at least procedurally, in FMEA standards and guidance documents. This suggests that the issues may not be caused simply by a lack of instruction, but by deeper difficulties in applying the method within real development contexts.

As an early example, McKinney (1991) examined the application of Failure Mode, Effects, and Criticality Analysis (FMECA) in a case study of an approach radar system and identified seven categories of deficiencies. Ten years prior to McKinney's article, the U.S. military had further codified the FMEA process in MIL-STD-1629A (U.S. DOD, 1980). In that revised standard, the Department of Defense had already addressed the types of issues that McKinney (1991) later identified as deficiencies. However, McKinney (1991) evaluated FMEAs from nine different vendors and found that each included some portion of the seven deficiencies, with at least one vendor's FMEA exhibiting all seven. These deficiencies occurred despite MIL-STD-1629A being included as part of the contract language governing the generation of the FMEAs.

A comparison of the deficiencies identified by McKinney (1991) and the related sections of MIL-STD-1629A (U.S. DOD, 1980) is shown in Table 4. McKinney (1991) emphasized that

the intent of the activity was the “early identification of all catastrophic, critical and safety related failure possibility so they can be eliminated or minimized through design changes” [emphasis in the original] (McKinney, 1991). This intent closely mirrors the purpose of FMEA from its military origins, including nearly verbatim language in the original standard (U.S. DOD, 1949). Thus, even though the military standard provided specific procedures, instructions, and references to related standards, the same categories of errors occurred in practice.

Table 4: Comparison of McKinney (1991) Deficiencies vs. MIL-STD-1629A

McKinney (1991) Issue	MIL-STD-1629A Related Section
Lack of defined failure causes	<i>Section 5.4 Failure modes and causes</i> – includes the potential for lower-level failures to cause failures at higher levels in the system
Reckless and improper severity classification	<i>Section 3.3 Failure mode severity classification</i> – provides both a severity scale and a procedure for assessing different failure scenarios in the system
No data sources	<i>Section 4.4.1.4 Reliability data</i> – specifically identifies requirements for the use of reliability data in the analysis and references standards governing data generation
Failure to recognize deficiencies and failure modes from earlier systems of similar design	<i>Section 4.5.2 Reliability criticality item list part e)</i> – specifically calls out the use of the history of the design or similar designs
Lack of recommendations pertaining to operation and support of the system	<i>Section 4.1.1 Mission functions and operational modes</i> – calls for beginning at a high level and decomposing to lower system levels; it also calls for identifying alternative operational modes
Narrow scope of analysis	<i>Section 4.4.2 FMEA part a) Define the system to be analyzed</i> – articulates the need for complete system definition, including internal and interface levels as well as performance expectations at all levels of the system design
Untimely submission	<i>Appendix A 50.1.2 Timing</i> – states that the analysis should provide actionable information “at or before a project decision point.” It further indicates that the FMECA plan must support design during system development

More recently, Bluvband & Grabov (2009) performed a direct analysis of the FMEA process and identified specific pitfalls. Although their terminology differs somewhat from McKinney's, they identified issues that are remarkably similar to those reported by McKinney (1991). Notably, Bluvband & Grabov (2009) did not reference McKinney's work, suggesting that their analysis was conducted independently.

Silverman & Johnson (2013) also performed an independent analysis of FMEA, but their emphasis differed from the previous two studies. Rather than focusing primarily on direct execution problems, they examined organizational implementation issues that contribute to FMEA failure. Even with this different framing, several of the problems they identified align with those discussed by both McKinney (1991) and Bluvband & Grabov (2009). Table 5 shows how these three independent analyses converged on similar problems in FMEA application.

Table 5: Comparison of Issues within FMEA Identified by Three Independent Articles

Issue Category	McKinney (1991)	Bluvband & Grabov (2009)	Silverman & Johnson (2013)
Data	No data sources	Use of irrelevant statistics	No direct equivalent
Severity Classification	Reckless and improper severity classification	Wrongly defined criteria for high priority items and undefined risk acceptance criteria	Scoring system not customized/ scoring system not developed in advance
Design Stage Use	Failure to recognize deficiencies and failure modes from earlier systems of similar design	Missing failure modes and lack of guidelines for the optimal choice	Identifying failure modes but not prioritizing or mitigating the correct ones
FMEA Scope	Narrow scope of analysis	Narrow definition of failure related to scope	Agenda not clear from the start/attempting too large an FMEA

Additional evidence of recurring criticism appears in the meta-analysis by Spreafico et al. (2017), who synthesized published criticisms of FMEA across 220 scientific papers and 109 patents. They identified four major problem classifications affecting FMEA. The first, applicability, appeared in 86 citations and included concerns related to FMEA being time-consuming, expensive, difficult to repeat, subjective, difficult to manage from a data and information standpoint, and often applied too late. The second, cause and effects, appeared in 38 citations and related to shortcomings in failure determination. The third, risk analysis, appeared in 45 publications and related to ambiguous criteria for quantifying risks. The fourth, problem solving, appeared in 22 papers and related to poorly defined problems and poor implementation of solutions. Consistent with previous periods, Alruqi, et al. (2022) raise problems with the extensive resource requirements to deal with the proliferation of issues in complex systems as well as the other areas that Spreafico identified in 2017. Similarly, Gaikwaid, et al. (2023) expand on the issue of scale of failure modes and complexity by using AI to organize it, rather than filter it, thus making the volume more manageable to examine, but still requiring the resources to address.

The same themes continue. Ouyang et al. (2022) identified three major flaws in applying FMEA to real-world problems: (1) difficulty obtaining accurate values for risk factors for every failure mode; (2) failure to consider weighting among risk factors; and (3) problems with the use of RPN methodology. These flaws clearly trace to the scoring and prioritization issues raised by McKinney (1991) more than 30 years earlier. As with earlier researchers, Ouyang et al. (2022) proposed methods to address these flaws, largely through improvements to scoring methodology. While these proposals are technically sophisticated, the continued recurrence of similar

criticisms suggests that scoring improvements alone may not resolve the broader problems associated with FMEA application.

Scoring methodology remains one of the most active areas of FMEA research, particularly in relation to the limitations of common scoring approaches (Spreafico et al., 2017). When using either criticality matrix or RPN methods, equivalent resulting scores may not represent equivalent levels of concern to the business or design team. Although RPN appears to be a mathematical expression, the resulting values function primarily as an ordinal sorting mechanism for FMEA line items rather than as a mathematically meaningful measure of relative risk. Therefore, the difference between scores does not necessarily represent the relative importance of the issue to the system design or the business.

Researchers have continued to examine these scoring limitations through more sophisticated methods, including models for measuring the effectiveness of FMEA risk assessment and approaches intended to augment RPN-based prioritization (Panyukov et al., 2022; Lo & Liou, 2018). These problems appear across FMEA application areas. For example, in usability research, similar challenges related to prioritization, scoring, and identification of risk factors have become increasingly common. Researchers in that space have taken paths similar to those used in other FMEA research areas to address these challenges (Li & Zhu, 2020).

Research in the medical device space provides additional insight into why FMEA remains common despite its known limitations. Nolan and McDermott (2025) surveyed experienced medical device risk management professionals and each participant indicated that FMEA, or a hybrid variant of FMEA, was the most used risk analysis tool. At the same time, each participant described problems with FMEA that were largely consistent with prior assessments of the method. When asked about the apparent contradiction between FMEA's

ubiquity and its persistent problems, a strong theme of stakeholder familiarity emerged. Regulators expect FMEA and generally do not question the methodology, engineers are familiar with it, and the traceability from risk to mitigation meets the needs of quality and risk professionals. Respondents were familiar with other risk assessment methods, and in some cases advocated for or used them, but generally as augmentations to FMEA rather than replacements. As medical devices become embedded within broader medical device ecosystems, the interconnectivity among devices, manufacturers, healthcare professionals, patients, regulators, researchers, and data flows further complicates the application of proactive risk management methods (Coronado & Gallegos, 2025). Nolan and McDermott (2025) also note that FMEA does not completely address the requirements of ISO 14971:2019 (ISO, 2019). Despite these limitations, FMEA is still viewed as an indispensable component of risk management in this context.

The body of research shows that many of the same criticisms have persisted since McKinney's 1991 analysis. Moreover, a review of standards dating back to the original MIL-P-1629 indicates that many of these problems were anticipated, with cautions and guidance provided to help users avoid them. These cautions and guidance have continued to be refined in standards literature, including IEC 60812:2018 (IEC, 2018) and the joint AIAG-VDA FMEA handbook (AIAG & VDA, 2019). The persistence of these problems, despite long-standing procedural guidance, suggests a possible systemic mismatch between FMEA logic and typical system development logic. The next section therefore introduces the systems engineering V-model as one of the most common development models currently used in practice.

3.3. Systems Engineering V-Model

3.3.1. Overview

The Systems Engineering V-model is a structured development model used to organize the progression from high-level user needs to detailed design and then back upward through integration, verification, and validation. It is commonly represented as a “V” because the left side of the model describes the decomposition of the system from user needs and system requirements into architecture, subsystem requirements, module requirements, and detailed design. The right side of the model describes the corresponding integration and verification activities used to demonstrate that the implemented design satisfies those earlier requirements. Figure 1 illustrates the systems engineering V-model.

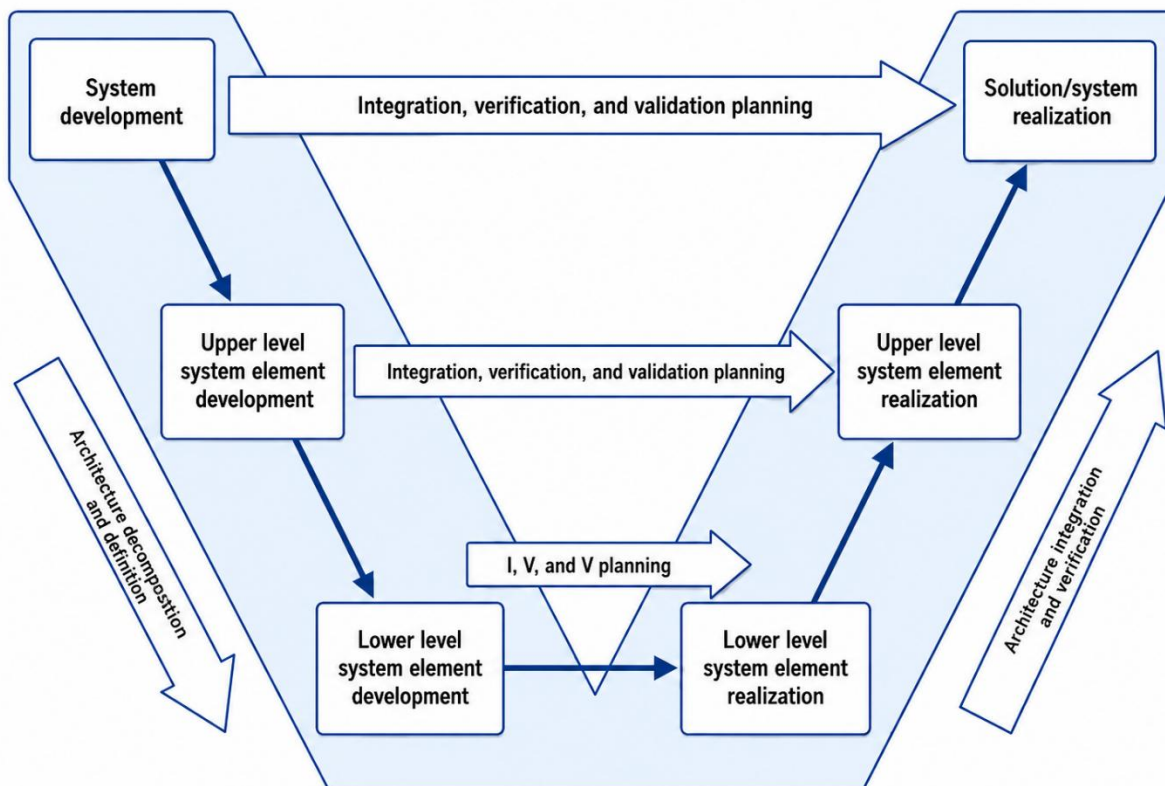


Figure 1: Systems Engineering V-Model Adapted from INCOSE (2015) and Forsberg et al. (2005)

A central feature of the V-model is the relationship between design decomposition and verification planning. As requirements are decomposed on the left side of the V, corresponding verification or validation activities are planned for the right side. This structure is intended to ensure that design decisions are traceable to requirements and that evidence of design maturity is developed as the system is integrated. In this way, the V-model provides both a development logic and a traceability structure for managing complex systems across the lifecycle, which is important because long-lived systems require proactive planning not only for verification and validation, but also for sustainment challenges such as obsolescence management (Chellin & Miller, 2023a).

A related strength of the V-model is that it begins with user needs rather than immediately with technical design. This ordering grounds the development process in the operational context, constraints, and expectations of the people who will use or be affected by the system. Prior research on technology adoption and human-systems integration emphasizes that technically capable systems may still fail to achieve sustained use when user needs, cultural fit, maintenance burden, resource access, and local operating conditions are not adequately considered (Ahmed & Miller, 2023; Ahmed & Miller, 2022). By placing user needs at the beginning of the development process, the V-model embeds a human-centered starting point for requirements development and provides a structure for tracing those needs through design, verification, and validation.

For this dissertation, the V-model is important because it defines when different types of design information become available. Early in the V-model, teams may understand user needs, intended functions, and high-level system requirements, but they may not yet know the detailed architecture, component specifications, failure mechanisms, or verification results. Later stages provide more technical detail, but by then major design commitments may already have been made. This staged availability of information is central to the argument that conventional FMEA can be difficult to apply effectively within V-model development.

3.3.2. Historical Development of the V-Model

The historical development of the V-model helps explain why these decomposition and verification concepts became central to modern systems engineering practice. As software development activities became more significant in the 1980s, the need for formal management of software development became prevalent. In 1985, the U.S. DOD issued DOD-STD-2167, which was superseded three years later by DOD-STD-2167A, to define expectations for the management of defence software development (U.S. DOD, 1985; U.S. DOD, 1988). This standard introduced many systems engineering concepts associated with staged development, including parallel paths for both software and hardware development, as summarized in Figure 2. This standard also provided detailed expectations for military project work products, including system decomposition, system block diagrams for defining architecture, and related development artifacts.

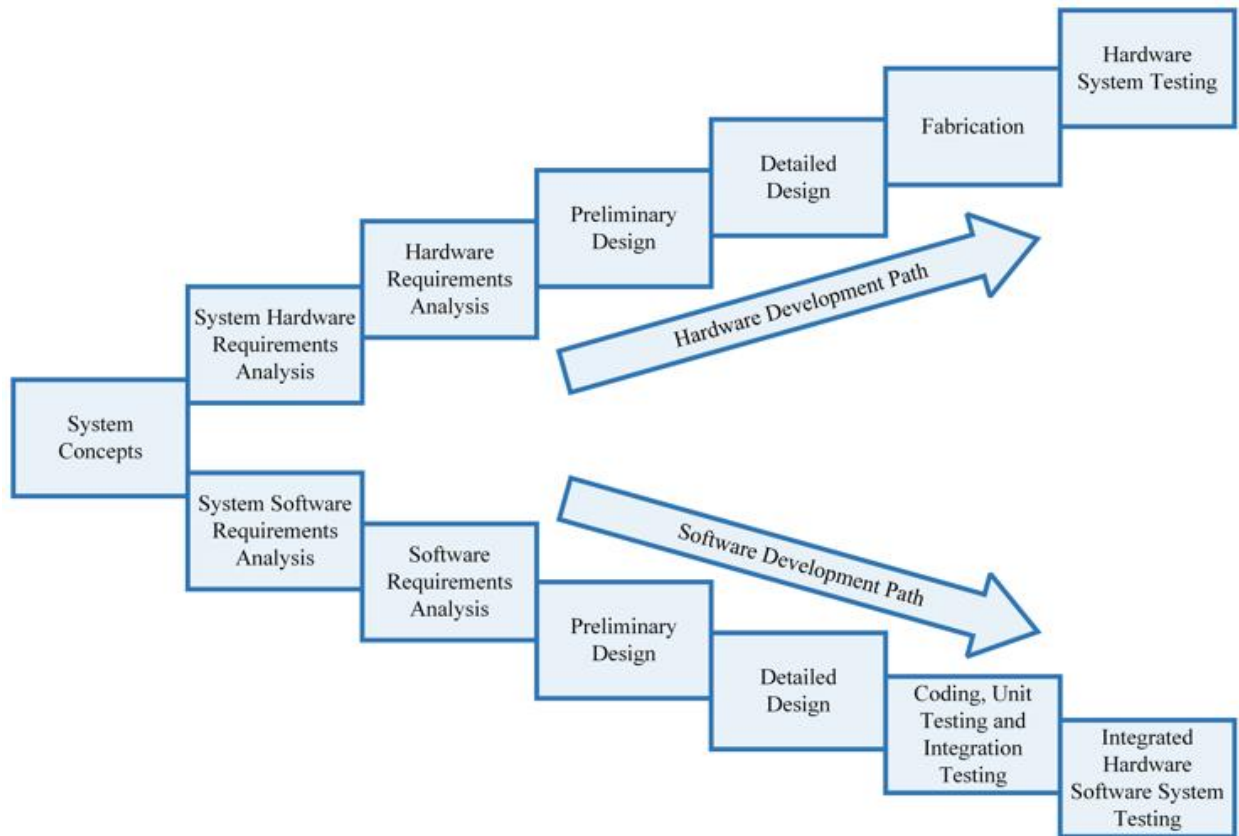


Figure 2: Adapted from DOD-STD-2167A Development Logic (U.S. DOD,1988)

3.3.3. Evolution from DOD-STD-2167A to the V-Model

Rook (1986) introduced one of the early papers on the management of software development, based on the idea that software development should be treated as a science rather than an art. His paper introduced the V-model as a development approach. A significant contribution of Rook's work, relative to the ideas in DOD-STD-2167A, was the identification of confidence points within the development process. These confidence points related testing and evaluation activities to corresponding development activities, an idea that was not clearly articulated in the standard.

Rook (1986) also recognized that, in software development, the stages identified in DOD-STD-2167A were not as distinct or linear as the standard implied. Instead, different levels of confidence were needed at different points in the development process. Nevertheless, the development logic in Rook's model closely matched the logic proposed in DOD-STD-2167A. A replication of this V-model is provided in Figure 3, with annotations from related elements in DOD-STD-2167A.

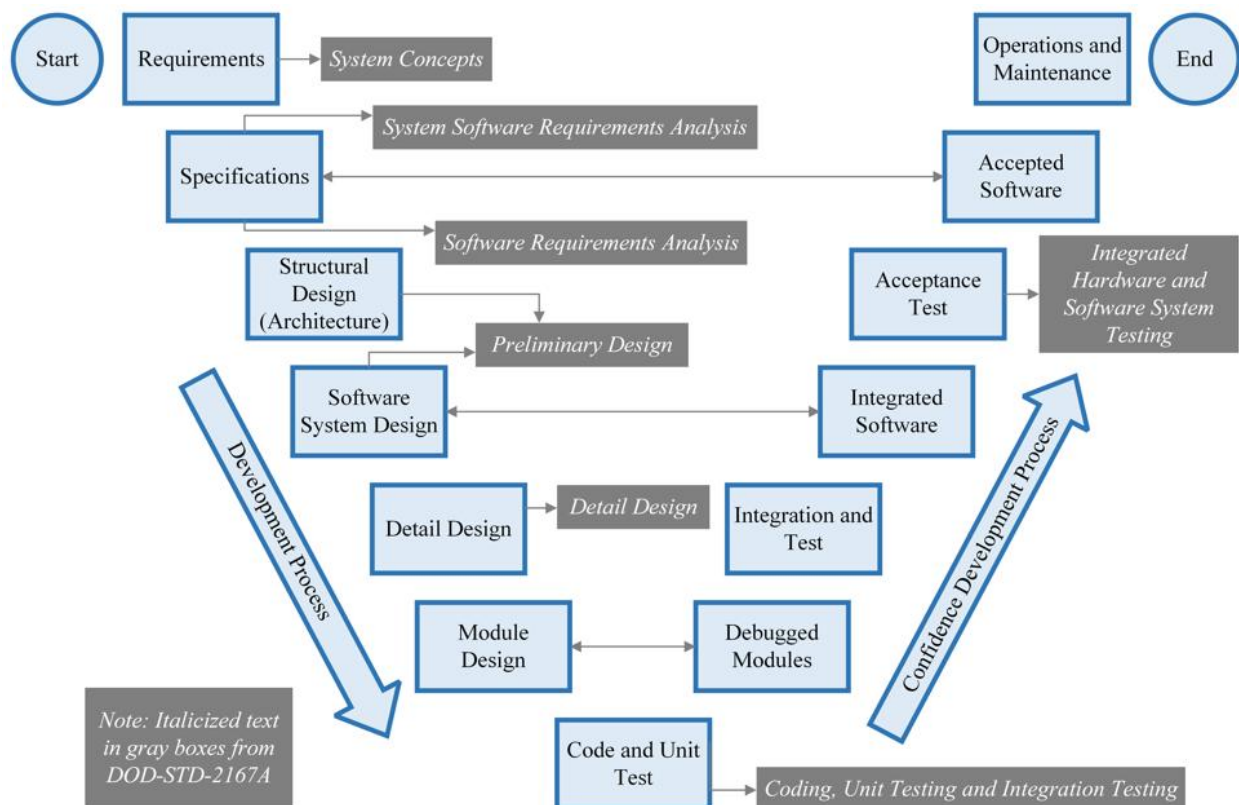


Figure 3: Adapted from Rook's (1986) V-Model with Annotations (gray boxes) Overlaid from DOD-STD-2167A (U.S. DOD, 1988)

Rook's V-model illustrates the importance of connecting testing and development throughout the development process in order to build confidence in the design. By making the

arrows bidirectional, Rook also represented the iterative nature of software development, which was not explicitly recognized in DOD-STD-2167A. Similarly, Juristo et al. (2007) demonstrate that software design must include usability no later than the design stage, because failure to do so can result in unusable software and intolerable costs. Horskey et al. (2010) also discuss the need for recurring usability evaluations during system design.

3.3.4. Connecting the V-Model to Systems Engineering as a Development Model

In 1992, Forsberg & Mooz (1992) discussed the connection between the V-model and systems engineering, and the development of complex systems with hardware and software, not only software. Rook's (1986) "confidence development path," which related development activities to testing activities, was recognized both in research by Forsberg & Mooz (1992) and in practice by the U.S. government (U.S. DOD, 1991). Specifically, the Department of Defense called for prototypes as part of development activities to demonstrate design maturity at different levels (U.S. DOD, 1991).

Forsberg & Mooz (1992) generalized Rook's original V-model to apply more broadly to systems development, as shown in Figure 4. In their enhancement of the model, they condensed some steps while preserving the essential idea that design begins with the definition of end-user needs. From there, designers decompose and define the design from the system level down to individual elements and components.

Forsberg & Mooz (1992) also emphasized that, at each level of design, the corresponding plan for inspection, verification, or validation should be created. This introduces discipline into the development process because any verification plan requires three elements: (1) a requirement or specification that must be met; (2) a method of inspection, verification, or validation; and (3)

an acceptance criterion for dispositioning the outcome of that activity. These elements, including the “off-core” activities discussed by Forsberg & Mooz (1992), drive a level of rigor in design that may not otherwise be achieved.

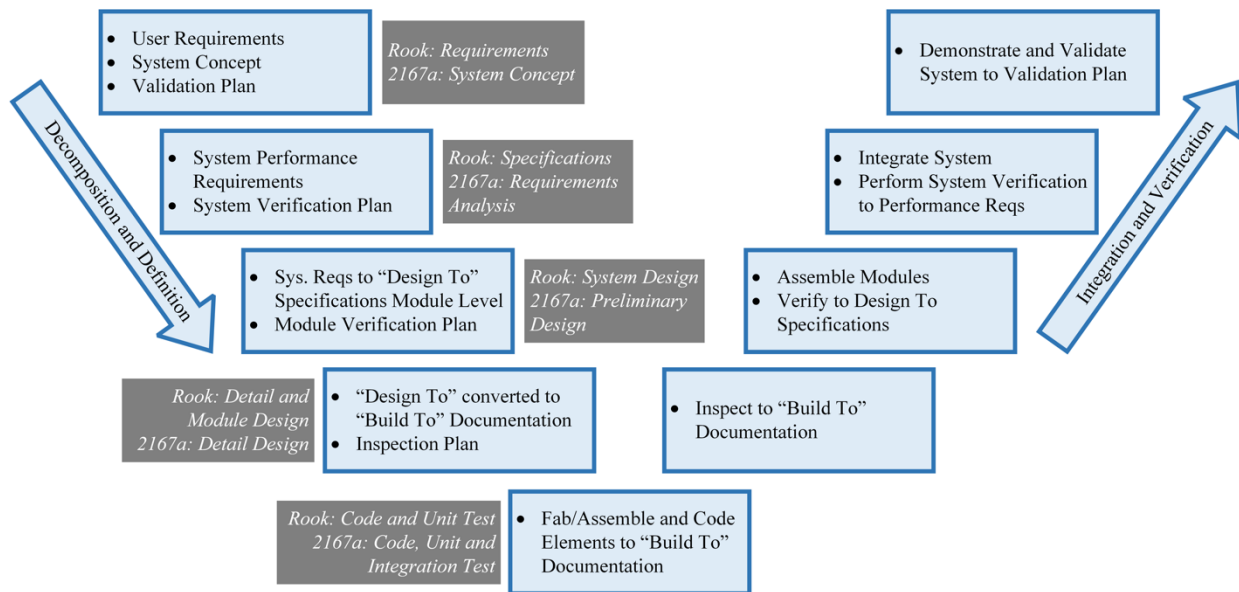


Figure 4: Adapted from Forsberg & Mooz (1992) V-Model with Annotation (grey boxes) from Rook (1986) and DOD-STD-2167A (U.S. DOD, 1988)

In review of the three works from DOD-STD-2167A (U.S. DOD, 1988), the initial defense software development framework, to Rook’s (1986) V-model, the creation of the V-model structure [for software development], to Forsberg & Mooz’s (1992) V-model, the application of the V-model to both software and hardware development, the V-model is refined, but the core elements remain the same. The V-model continues to be relevant in academic research and industry. Reflecting this in the usability research space, as an example, shows that addressing usability in the development process is of great interest and as discussed previously,

industry, particularly those that are closely regulated, have a need to incorporate usability into their development models.

3.4. Combining the Two: Using FMEA within the V-Model

Building on the literature discussed above, a generic V-model can be described with the following basic steps and resulting objects, contextualized for a human factors perspective:

1. *User Needs*: User requirements, user workflows, feature and function descriptions in user language, and the basic system concept.
2. *System Requirements*: High-level system architecture or platform definition, major subsystem decomposition and interfaces, and system-level performance requirements. This is also where the intended user interface for a system would be defined at the needs level.
3. *System Design*: Detailed system architecture, identification of subsystem design reuse elements, decomposition, and allocation of system requirements to subsystems.
4. *Detail Design*: Design of subsystems, decomposition, and allocation of requirements to modules.
5. *Module Design*: Design of modules, decomposition, and allocation of requirements to elements.
6. *Fabrication, Coding, Initial Integration*: Matured prototypes tested with initial integration and lower-level module testing.

These six steps represent the left side of the V and are generalized to apply to both hardware and software systems, based on the foundational literature discussed above (Forsberg

& Mooz, 1992; U.S. DOD, 1988; Rook, 1986). To synthesize these arguments, Table 6 compares the typical columns of an FMEA, as detailed in Table 1, Table 2, and Table 3, with the information required to perform an FMEA, as described in IEC 60812:2018, and the earliest step in the V-model at which that information is typically available for a system-level FMEA (IEC, 2018).

Table 6: Alignment of FMEA Information Needs with Information Available in the V-Model

FMEA Column	Information Necessary to Perform FMEA	When Information is Available in a Generic V-model
Element	Element definition within architecture	Step 3 (System Design)
Functions and performance standard	Requirements defining the expected performance for the element and its interfaces	Step 2 (System Requirements)
Failure mode	All possible ways the element can fail to meet the requirements	Step 1 (User Requirements) and Step 2 (System Requirement)
Local effect of failure	Impact of the failure at the local level	Step 3 (System Design)
Final effect of failure	Impact of the failure at the user or system level	Step 1 (User Requirements)
Cause of failure	Design detail to determine how the element can physically fail to cause the failure mode	Step 4 (Detail Design)
Initial state mitigation	Detailed understanding of design reuse to determine mitigations that have proven effectiveness	Step 3 (System Design) and Step 4 (Detail Design)
Initial state scoring (S x O x D = RPN)	Detailed understanding of design with data to understand cause of failure, to failure mode logic with data to determine scores	Step 4 (Detail Design) and Step 5 (Module Design)
Future state mitigations	Detail understanding of what will be incorporated into the design to address the failure mode	Step 4 (Detail Design)

Post mitigation rescore (S x O x D)	Detail understanding with test data to determine the score of the new mitigations	Step 6 (Fabrication and Initial Testing)
Evidence of effectiveness	Completed test reports showing mitigation effectiveness	Right side of the V (Verification Testing)

International standards, previous versions of FMEA guidance, and U.S. military contracts may require or advocate performing FMEA early in the design process, before major design commitments have been made, so that mitigations can be incorporated into the design. However, Table 6 shows that even for a system-level FMEA, the information required to complete the analysis becomes available across multiple stages of the V-model. Subsystem- or module-level FMEAs, which are also advocated within the standards, may not have the necessary information available until the detail design stage at the earliest.

This timing issue becomes more significant when FMEA is treated as an iterative process involving cycles of mitigation, testing, revised mitigation, and additional testing. The method is expected to inform design before design commitments are finalized, yet many of the inputs needed to complete the analysis emerge only after the design has matured. This creates a systemic tension between the early-use expectations of FMEA and the staged information availability inherent in the V-model. That tension helps explain why many criticisms of FMEA persist despite decades of procedural guidance, standards development, and methodological refinement.

3.5. Implementation Challenges with FMEA

Despite decades of refinement and widespread adoption, the effective implementation of FMEA remains a persistent challenge in practice. These issues have been recognized in the literature for several decades, including early work identifying difficulties in representing failure

modes that may lead to multiple potential outcomes or severities (Agarwala, 1990). Subsequent reviews and meta-analyses have continued to synthesize recurring limitations and proposed improvements to the method (Perreault et al., 2024; Spreafico et al., 2017; Bouti & Kadi, 1994).

Much of this research has focused on how risk is evaluated within FMEA, particularly the use of ordinal scoring and the calculation of the RPN. Prior reviews have identified limitations in the consistency, mathematical validity, and interpretability of RPN rankings (Liu et al., 2013). These concerns have motivated a wide range of extensions, including fuzzy logic approaches, multi-criteria decision frameworks, and more recent artificial intelligence methods (Ouyang et al., 2022; Xia et al., 2024). However, while these efforts address aspects of risk scoring and prioritization, they do not fully resolve broader challenges associated with applying FMEA to increasingly complex systems.

As system complexity has increased, limitations related to the scope and scalability of FMEA have become more pronounced. The methodology emphasizes comprehensive identification of failure modes, including combinations of faults and interactions across system elements (Stamatis, 2003). In modern products characterized by high feature density and tight integration of hardware, software, and firmware, this expectation can lead to a combinatorial expansion of potential failure modes. In practice, this proliferation can make the analysis difficult to execute and sustain, shifting FMEA toward a documentation or compliance-oriented activity rather than a focused tool for risk reduction.

Risk analysis methods can also be difficult to sustain when they are applied to long-lived systems whose operating conditions, components, and support environments change over time (Chellin & Gallegos, 2024). In these contexts, risk management must account not only for design-time failure modes, but also for lifecycle factors that affect whether a system can continue

to perform as intended. Related sustainment research shows that proactive, risk-based management is critical for long-lived systems, where obsolescence can negatively affect cost, schedule, performance, readiness, availability, and maintainability (Chellin & Miller, 2023b).

These challenges are compounded by the way FMEA is positioned within contemporary development processes. The V-model emphasizes a top-down progression from user needs through system architecture to detailed design, whereas FMEA is often applied as a bottom-up analysis that depends on detailed design information (Forsberg & Mooz, 1992; U.S. DOD, 1988; Rook, 1986). Although guidance recommends applying FMEA early in the development lifecycle, the information required for effective execution is often not available at those stages. The systemic misalignment between FMEA information needs and V-model information availability therefore contributes significantly to the practical difficulties encountered when applying FMEA in V-model development environments (Perreault & Gallegos, 2025).

The persistence of these issues has led some researchers to question whether FMEA, in its current form, remains an effective primary tool for design risk management (Nolan & McDermott, 2025). However, regulatory expectations, industry standards, and organizational practices continue to reinforce its use, making it difficult to replace in practice. As a result, the practical question is less whether FMEA should be abandoned and more how its application can be adapted to modern development environments.

Risk assessment in product development has also been studied through alternative frameworks, including probabilistic and Bayesian network approaches (Goswami, 2018) and decision-making models that incorporate competitive or strategic considerations in new product development (Mousavi et al., 2022). These methods provide valuable perspectives on development risk, but they are generally not aligned with the structure, workflow, or

organizational integration of FMEA as it is commonly practiced. Consequently, they are difficult to apply as direct substitutes in environments where FMEA remains the primary risk assessment tool.

3.6. Conclusion

This chapter examined recurring criticisms of FMEA and compared the information required to perform a conventional FMEA with the timing of information availability in the V-model. The analysis showed that many long-standing criticisms of FMEA, including incomplete failure identification, inconsistent scoring, narrow scope definition, late application, and difficulty translating results into effective design action, have persisted despite decades of procedural guidance in standards and handbooks.

The comparison with the V-model suggests that these recurring problems may reflect more than inconsistent execution. FMEA is expected to begin early enough to influence design decisions, yet many of the inputs required to complete a conventional FMEA are not available until later stages of decomposition, detailed design, integration, or verification. This creates a structural tension between the early-use expectations of FMEA and the staged information availability of V-model development.

The next chapter extends this analysis by examining how this systemic mismatch is treated in the existing literature and by using case examples to show how the mismatch appears in practice. This additional analysis completes the diagnostic work of Research Aim 1 and establishes the problem basis for the improved FMEA framework developed in Research Aim 2.

Chapter 4. Aim 1 - Systemic Gaps in FMEA Application

4.1. Introduction

Chapter 3 identified recurring criticisms of FMEA and connected many of those criticisms to a potential mismatch between the information required to perform FMEA and the timing of information availability in V-model development. This chapter extends that analysis by examining how the existing literature addresses these problems and where important gaps remain. In particular, the chapter argues that much of the literature focuses on symptoms of FMEA difficulty, such as scoring, prioritization, and risk classification, rather than the underlying systemic constraints that make FMEA difficult to apply in modern development environments.

The chapter first discusses the information challenges created when FMEA is applied within a V-model development process. It then presents redacted case study examples that illustrate how these challenges appear in practice. Together, these examples support the conclusion that the recurring problems identified in Research Aim 1 are not merely procedural errors, but reflect deeper incompatibilities between traditional FMEA expectations and the staged decomposition of modern systems development. This analysis provides the basis for Research Aim 2, which develops requirements for an improved FMEA approach.

4.2. Gaps in the Literature: Missing the Systematic Problems

Despite the increasing recognition of fundamental problems in the effective use of FMEA, very little research into structural solutions which would address the underlying process can be found. One of the more recent examples is more than a decade old and attempted to

address the complexity of software and hardware within a system (Ozarin, 2013). This lack of systemic research was also noted by Spreafico et al. (2017) in their literature review.

When reviewing the research around the issues and solutions presented, the topic of information availability, as identified in the previous chapter, is not explicitly discussed. Instead, the literature generally focuses on symptoms that may be driven by this mismatch. Spreafico et al. (2017) specifically note that the majority of the research they surveyed, whether from industry or academia, relates to detailed aspects of FMEA execution, including improvements in scoring, the use of other tools to augment cause-and-effect analysis, and improvements to prioritization based on scoring. However, none of the works surveyed by Spreafico et al. (2017) offered an in-depth discussion of the mismatch between the information required to perform FMEA and the information available at the relevant phase of the V-model, beyond general discussion of the applicability of FMEA as a tool.

In the V-model literature, system complexity is typically represented through discussions of decomposition and methods for managing it. In FMEA standards and literature, the need for decomposition mirrors the V-model. However, this same literature provides few, if any, methods for managing the resulting scale of the analysis. FMEA literature and standards emphasize capturing all failure modes and causes, but provide limited guidance on how to adapt this expectation to large design efforts. In even moderately sophisticated products, it is often practically impossible to identify every failure mode and every failure cause.

4.3. Information Challenges in V-Model Decomposition

It is a well-established expectation in FMEA literature, standards, and guidance that FMEA should be performed as early as possible in the design process, and certainly before major

design commitments are made. In many organizations, the development process expects the FMEA process to begin as early as Stage 1 or Stage 2 of the V-model, as illustrated in Figure 5.

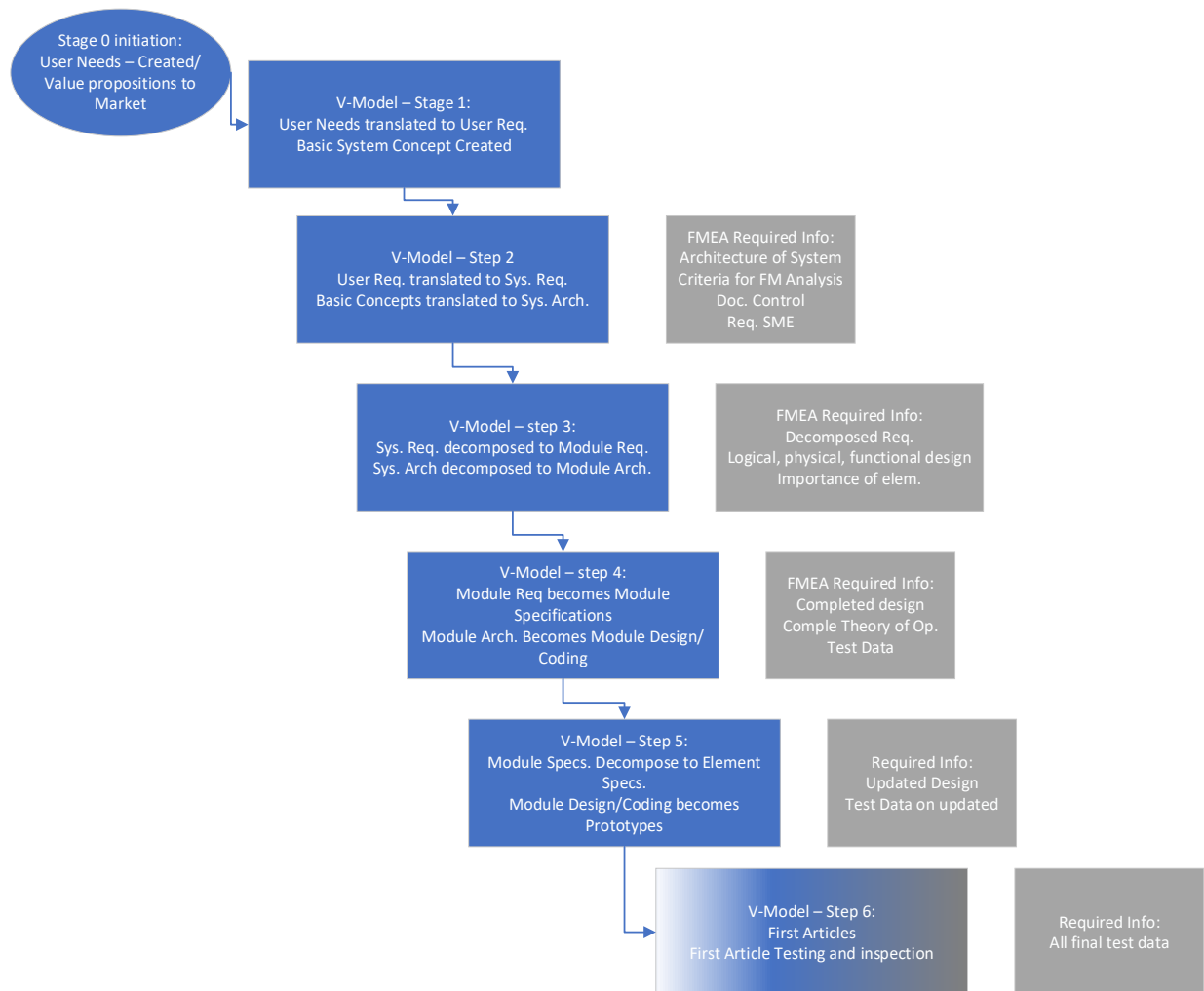


Figure 5: Comparison of V-Model Stages to FMEA Information Requirements

This figure illustrates that, at virtually any stage of the V-model during a development project, the expected FMEA deliverables are exceedingly difficult or impossible to complete because the required information does not yet exist. FMEA standards and texts tend to be

thorough in defining requirements and deliverables, but comparatively sparse in providing methodology for addressing this mismatch.

In retrospect, systems engineering frameworks such as the V-model emphasize a top-down progression from user needs through system architecture to detailed design (Katzier et al., 2025; Beasley et al., 2022). Classic FMEA, by contrast, is often described as a bottom-up risk management methodology (Kirke et al., 2015). Thus, the systemic inconsistencies between the design maturity needed to perform FMEA and the stage of the V-model at which FMEA activity is expected are not surprising.

4.4. Case Studies

As a further examination of this hypothesis, redacted case study examples are examined from FMEAs to which the author had access. In the company's development model, system-level FMEAs were required to be performed in the early stages of the V-model, as illustrated in Figure 5. Example lines from a sample of these FMEAs are reviewed here. These case studies are not intended to be exhaustive, but rather representative of how the challenges identified in the research appear in application.

4.4.1. Case Study 1: Motion Compensation for an Imaging System - Inadequate Risk Assessment

In this example, shown in Table 7, the image compensation option had not yet been developed, although pseudo-code had been developed for the intended algorithm. It was recognized that previous hardware platforms might not have adequate processing power. However, the cause of failure, "inefficient coding of algorithm," provided little value in supporting mitigation. Similarly, the use of system integration testing with what were perceived to be "extreme" imaging cases was ineffective.

Table 7: Sample Row from FMEA for Case Study 1

Failure Mode Summary								Current Design Control		
Item/ Function	Requirement	Potential Failure Mode	Potential Effects of Failure	Severity	Cause of Failure	Current Prevention	Occurrence	Current Detection	Detection	RPN
Motion compensation and persistence	The Image Compensation option provides the ability for the user to enable digital image stabilization	CPU cannot keep up with algorithm processing demands	System crashes	8	Inefficien t coding of algorithm	None	5	System integratio n testing of extreme imaging cases	1	40

Ultimately, the risk was viewed as acceptable based on the RPN score. However, this proved problematic and had to be addressed in a sustaining release. This case illustrates how an FMEA performed before sufficient design maturity can create the appearance of risk assessment without providing actionable risk information. The failure mode identified in the FMEA row, “CPU cannot keep up with algorithm processing demands,” was directly related to system performance, but the stated cause of failure did not sufficiently explain the architectural or design source of that performance limitation. “Inefficient coding of algorithm” may have been plausible, but it did not distinguish among several materially different possibilities, including insufficient processor capacity, inefficient algorithm structure, inadequate hardware allocation, or unrealistic performance requirements for the selected platform. As a result, the FMEA row did not provide a clear basis for selecting a mitigation.

The detection control also demonstrates the timing problem identified earlier in this chapter. System integration testing of extreme imaging cases could only identify the failure after

enough of the algorithm, hardware platform, and integration environment existed to perform the test. By that point, major architectural choices may already have been made. In this sense, the FMEA captured a real risk but did so in a way that depended on downstream information and downstream test activity. The low RPN score therefore created a misleading sense of acceptability, even though the underlying uncertainty had not been resolved.

4.4.2. Case Study 2: System Temperature Monitoring - Vague Failure Definition

In this example, shown in Table 8, the row item is poorly constructed because it simply documents how the temperature monitoring system is designed to be used. At this point in the design process, there was no clarity regarding whether the condition would be met by the system or what failure of the temperature monitoring system itself would mean. The system architecture decomposition had not yet been completed, leading to confusion on the FMEA team's part about how to process the line.

The resulting row combines aspects of a requirement, a system response, and a failure outcome in a way that makes the analysis difficult to interpret. The requirement states that the system warns the user of an overtemperature condition and then shuts down within a specified time period. However, the potential failure mode is recorded as "System Shuts down," which is not necessarily a failure if shutdown is the intended protective behavior. This creates ambiguity about whether the FMEA is analyzing failure of the temperature monitoring function, failure of thermal management, excessive heat generation, or the user impact of a protective shutdown. Since the failure mode is not clearly tied to a violated requirement, the subsequent effect, cause, and scoring are also difficult to interpret.

Table 8: Sample Row from FMEA for Case Study 2

Failure Mode Summary								Current Design Control		
Item/ Function	Requirement	Potential Failure Mode	Potential Effects of Failure	Severity	Cause of Failure	Current Prevention	Occurrence	Current Detection	Detection	RPN
System Temperature Monitoring	System warns user of overtemp. condition then shuts down within time period	System Shuts down	System not availabl e	8	Heat buildup due to extended system use at high GPU utilization	Temperatu re monitorin g.	5	Design Qualificat ion Testing	5	200

This example demonstrates how early-stage FMEA can become unstable when architectural responsibility has not yet been allocated. The cause of failure, “Heat buildup due to extended system use at high GPU utilization,” may describe a plausible operating condition, but it does not identify whether the design concern resides in the GPU workload, thermal design, cooling capacity, sensor placement, shutdown threshold, warning logic, or system-level performance expectation. As a result, the row does not clearly guide design action. The FMEA team was effectively being asked to analyze a lower-level design problem before the system architecture had decomposed the requirement into accountable subsystems or design elements.

4.4.3. Case Study 3: UI Controls Requirement - Unnecessary FMEA

In this example, the RPN score of 72 was low enough, according to the business’s scoring rubric, that no additional mitigation was required. The FMEA team, when interviewed, felt this was an example of a “paperwork exercise” because the UI was design reuse, the AppUI library was well proven on the platform, and the only difference was the size of the images to which the optimization controls were being applied, as shown in Table 9.

Table 9: Sample Row from FMEA for Case Study 3

Failure Mode Summary								Current Design Control		
Item/ Function	Requirement	Potential Failure Mode	Potential Effects of Failure	Severity	Cause of Failure	Current Prevention	Occurrence	Current Detection	Detection	RPN
HD Image UI Controls	System has requirements for specific UI controls to support user image optimization of the dynamic video	Incompatible system state created by the UI Controls	System Crash	8	Incompatible system state created by the UI Controls	Existing AppUI Library supports controls.	3	Integration Software Testing	3	72

Therefore, before the row was ever populated, it was understood that no further action was likely to be required because of the standardization of the UI image optimization controls from predicate products. This case demonstrates a different form of mismatch, the FMEA process was applied to an element that the team already understood to be mature design reuse. Unlike the previous examples, the issue was not that the risk was poorly understood or that the design information was unavailable. Rather, the available design history suggested that the feature was unlikely to warrant focused FMEA effort. The AppUI library had already been implemented and tested on the platform, and the change involved applying established controls in a slightly different image context. The FMEA row therefore consumed effort without changing the design decision, risk understanding, or mitigation plan.

This example is important because it shows how proliferation of failure modes in FMEAs as well as the proliferation of FMEAs across systems can occur even when individual rows are technically defensible. A team can often imagine a failure mode for a reused design element,

such as an incompatible system state created by UI controls. However, if the architecture, implementation pattern, and verification approach are already well established, the FMEA may add little value beyond documenting that the team does not intend to take further action. In practice, this contributes to the perception of FMEA as a paperwork exercise and diverts attention from areas of the design where uncertainty is higher.

4.4.4. Case Study 4: Border Computation in Image - Complexity and Novelty

This example is similar to Case Study 1, but in this instance the issue was whether the algorithm itself was adequate for performing the detection. In this case, shown in Table 10, the algorithm had AI elements associated with it, so detection as a mitigation could only have value after the AI was trained on sample video clips. This required substantial development work to create curated training materials for the AI. As a result, the project experienced delays because the team could not predict when the algorithm would be trained to a suitable level.

Table 10: Sample Row from FMEA for Case Study 4

Failure Mode Summary								Current Design Control		
Item/ Function	Requirement	Potential Failure Mode	Potential Effects of Failure	Severity	Cause of Failure	Current Prevention	Occurrence	Current Detection	Detection	RPN
Border Computation Algorithm	During continuous video, the Border Computation algorithm provides real time boundaries for desired artifacts in complex video	Incorrect border identification	User Misinterpretation of the result	10	Algorithm cannot accurately identify boundaries in image	None	3	System integration Testing	3	90

This case illustrates a risk that was more fundamentally tied to novelty and uncertainty than to a known component failure mechanism. The failure mode, “Incorrect border identification,” was directly meaningful at the system and user level because it could cause the user to misinterpret the result. However, the cause of failure, “Algorithm cannot accurately identify boundaries in image,” reflected an unresolved capability question rather than a specific design defect. At the time the FMEA was created, the team did not yet have sufficient information to determine whether the algorithm would meet the requirement after training, what training data would be sufficient, or what operating conditions would expose weaknesses in the model.

The detection control, “System integration testing,” also highlights the limitation of relying on late-stage verification to manage early-stage algorithmic uncertainty. For AI-enabled or training-dependent functions, the ability to test the system depends on first developing the training set, training the algorithm, and defining representative cases. Until those activities occur, the team may not know whether poor performance is caused by inadequate training data, limitations in the algorithm, insufficient computational resources, or an unrealistic requirement. The FMEA row therefore identifies an important risk, but the information needed to characterize and reduce that risk does not yet exist at the point where the FMEA is expected to support design decision-making.

4.4.5. Synthesis

The four cases illustrate different ways the same systemic mismatch appears in practice. Case Study 1 shows a risk that was recognized but not adequately characterized because detailed

design and performance information were not yet available. Case Study 2 shows how unclear architectural decomposition can produce vague or internally inconsistent FMEA rows. Case Study 3 shows how comprehensive FMEA expectations can drive analysis of low-value design reuse, contributing to proliferation. Case Study 4 shows how novel or complex algorithmic functions may require feasibility, training, and performance evidence before conventional FMEA controls can be meaningfully evaluated. These examples support the broader argument that recurring FMEA problems are not merely the result of poor execution, but often arise from applying a detail-dependent, bottom-up method within a staged, top-down development process.

4.5. Conclusion

One possible conclusion from the analysis performed here is that FMEA and V-model development are incompatible, requiring a choice between one or the other. However, particularly for human factors, this conclusion would be problematic. FMEA has recognized value as a tool in use error and safety analysis, while the V-model provides a structured means of integrating usability analysis into the systems development lifecycle. Further, as previously discussed, many regulated industries require both a structured development process, such as the V-model, and the use of FMEA to support safety risk assessment and reliability by design.

Therefore, this dissertation rejects the premise that FMEA and the V-model are fundamentally incompatible. Instead, the analysis suggests that FMEA logic must be adapted to better align with the staged information availability and decomposition logic of the V-model. The recurring problems identified in the literature and illustrated through the case studies are not simply failures of practitioner execution. Rather, they reflect systemic tensions between what traditional FMEA requires and what the development process can provide at different stages.

The findings from Chapters 3 and 4 complete the diagnostic work of Research Aim 1. They show that persistent criticisms of FMEA can be understood, at least in part, as symptoms of systemic mismatch between conventional FMEA expectations and V-model development. These findings establish the basis for Research Aim 2, which develops requirements and a proposed framework for improving FMEA application under V-model development.

Chapter 5. Research Aim 2 – Proposed Solution

5.1. Introduction

Research Aim 2 develops a proposed improvement to FMEA that addresses the systemic problems identified in Research Aim 1. The previous chapters showed that recurring criticisms of FMEA are not simply isolated execution issues, but are associated with deeper constraints in applying a traditionally bottom-up risk analysis method within a top-down V-model development process. These constraints include proliferation of failure modes, incompatibility between FMEA information needs and the timing of information availability in the V-model, and difficulty aligning FMEA decomposition with systems engineering decomposition.

This chapter addresses Research Aim 2 by first translating these systemic problems into high-level requirements for an improved FMEA approach. It then proposes a framework that augments FMEA through New, Unique, and Difficult (NUD) design risk filtering, requirements-based definitions of failure modes and causes, recursive decomposition of mitigations, and cross-discipline tracking through Critical-to-Quality (CtQ) designations. In doing so, the chapter addresses the following research questions:

- **RQ2.1:** What high-level requirements define the solution space for improvements to FMEA that address the systemic causes and constraints identified in Research Aim 1?
- **RQ2.2:** How can FMEA be augmented to better align with the V-model based on the requirements developed in Research Aim 2?

5.2. Specific Problems to Target for a Solution

Using good design practices, the specific problem statements regarding the application of FMEA under the V-model represent the starting point for developing a solution. Based on the analysis in the previous chapters, three systemic problems are targeted: (1) proliferation of failure modes; (2) incompatibility between FMEA information needs and V-model information availability; and (3) difficulty aligning FMEA with V-model decomposition.

The first problem is the proliferation of failure modes. As indicated in the literature, a core challenge in FMEA application is the number of FMEAs, failure modes, and failure causes that may need to be considered as the complexity of features and functions increases in modern product design. This challenge is compounded by the generational nature of many products, where development often begins with an existing predicate product and advances from that established design. This creates an unbalanced level of design risk and maturity in the product development. This distracts FMEA teams from the less developed areas of a product which inherently are likely to be higher risk due to the unknown and focuses them on the predicate reuse which is well developed and therefore more difficult to change as well as should represent inherently lower development risk.

The second problem is the incompatibility between the information required by classic FMEA and the information available at different stages of the V-model. As shown through the standards analysis and case study examples, the information needed to support classic FMEA is often not available at the points in the development process where FMEA is expected to occur.

The third problem is the difficulty of following V-model decomposition with FMEA. The V-model is structured as a top-down development model, moving from system-level needs and requirements toward lower-level design detail. Classic FMEA, by contrast, is often structured as

a bottom-up risk management method. As a result, performing FMEA in a way that aligns structurally with the V-model is problematic.

These three problems define the solution space for the proposed improvement to FMEA. From here, a set of high-level requirements can be established.

5.3. Requirements

At a high level, the proposed solution must modify FMEA in ways that directly respond to the systemic problems identified in the previous section. First, the improved approach must include a filtering process for evaluating the risks associated with an intended new design at an architectural level. The purpose of this filter is to determine where FMEA provides the greatest benefit by distinguishing between new or higher-risk elements of the design and stable elements with strong design provenance. This requirement addresses the problem of *proliferation* by reducing unnecessary analysis of low-risk design reuse while focusing attention on areas where uncertainty is greater.

Second, the improved approach must allow the FMEA process to be more flexible in its information requirements and outputs depending on where it is being applied within the V-model. Since different types of design information become available at different stages of development, the method must support useful risk analysis without requiring information that does not yet exist. This requirement addresses the *information incompatibility* between classic FMEA and V-model development.

Third, the improved approach must provide a structure for iterating and decomposing FMEA in a way that works synergistically with the V-model rather than in conflict with it. Instead of forcing FMEA to operate as a purely bottom-up method, the revised approach should

support decomposition from higher-level requirements and architecture into lower-level design detail. This requirement addresses the difficulty of aligning FMEA with V-model *decomposition*.

In addition, most regulated industries that require FMEA as part of risk management and reliability development also require evidence that mitigations are effective. This evidence must be *traceable*, showing the linkage between the mitigation and the failure mode it is intended to address. Therefore, the improved approach must take advantage of the right side of the V-model to provide evidence of mitigation effectiveness in a way that is consistent with the verification and validation logic of the V-model.

Finally, the improved approach must be *compatible* with ongoing improvements to FMEA rather than replacing or preventing them. As discussed previously, nearly every article that identifies a problem with FMEA also proposes some form of improvement. Spreafico et al. (2017) categorize many of these solutions as improvements in scoring, risk categorization, failure mode elicitation, and related areas. This also does not account for the likely continued development of artificial intelligence methods to improve specific elements of FMEA execution. Therefore, the revised model should operate as a framework for applying FMEA under the V-model while still allowing improved methods developed by other researchers to be used during specific execution phases.

These requirements translate the problem elements identified in the previous section into performance expectations for the improved FMEA model.

5.4. Proposed Framework Overview

To address the high-level nature of the challenges raised by the requirements, the proposed solution is structured as a *framework* that can be applied beginning at the user needs or

requirements phase of the V-model. The framework is intended to preserve the core value of FMEA while adapting its execution to better align with the information available during staged system development. At a high level, the proposed framework includes the following elements:

- **A New, Unique, and Difficult (NUD) filter** to identify which areas of the design warrant focused risk analysis based on requirements and architecture information rather than requiring design detail.
- **A scoped FMEA plan** that focuses FMEA effort on higher-risk areas rather than attempting to analyze all possible failure modes across the entire system.
- **Requirements-based definitions of failure mode and cause of failure** so that failure analysis is tied to expected system performance and architectural allocation.
- **A top-down, decomposable FMEA structure** that follows the V-model by beginning with higher-level requirements and architecture, then decomposing only where additional analysis is needed.
- **Mitigations expressed as requirements** when design details are not yet available, allowing risk reduction actions to be allocated into the architecture before detailed design is complete.
- **Recursive review of mitigations through the NUD filter** to determine whether a mitigation introduces new design risk that requires further analysis.
- **Use of the right side of the V-model** to provide verification and validation evidence that mitigations are effective.

- **Critical-to-Quality (CtQ)** designation as a cross-discipline tracking mechanism for requirements or mitigations that cross architectural or engineering discipline boundaries.

These elements create a framework in which FMEA is not replaced, but repositioned within the V-model. The NUD filter helps determine where FMEA is most valuable, while the modified FMEA structure allows the analysis to proceed using the level of information actually available at each stage of development. Further, at the bottom of the V-model when detail design becomes available, classic FMEA can be performed strategically if needed against the risk context already established.

5.5. Design Risk Filtering Using New, Unique, and Difficult Analysis

5.5.1. Purpose of the NUD Filter

The first component of the proposed framework addresses the problem of FMEA proliferation through the creation of a design risk filter. Complex system development is rarely “green field,” or starting from no predicate design. Instead, most systems grow in complexity by adding innovation to existing, proven platforms. This creates significant differences between the maturity of the predicate design being reused and the innovations being added.

In many cases, the result is not simply an enhanced version of the predicate design, but a new product or system in its own right. This creates significant challenges in scoping the FMEA process, because the combinatorial proliferation of failure modes can rapidly become unmanageable, even with digital tools. Therefore, a filtering process is needed to determine where FMEA provides the greatest value.

The proposed filter is the New, Unique, and Difficult (NUD) analysis. The NUD process is intended to evaluate user requirements, system concepts, and early architecture information to determine which areas of the design should be included in the FMEA scope. Items that meet one or more NUD criteria are flagged for further risk analysis. Items that do not meet the criteria are treated as sufficiently proven or low-risk design reuse and can proceed through normal design practices.

The purpose of this filter is not to claim that unflagged items have no risk. Rather, the purpose is to distinguish between areas where standard design controls practices are likely sufficient and areas where additional structured risk analysis is warranted. As an example, all regulated industries have regulations requiring a defined design controls process which address design risk. In this way, the NUD filter provides a disciplined mechanism for narrowing the FMEA scope while preserving attention on areas of the design with greater uncertainty.

5.5.2. Definition of New

In the proposed framework, **New** refers to a user requirement, feature, technology, or aspect of the system concept that is new to the business or design team, but not necessarily new to the industry. This often includes adding a feature to a product that already exists in competing products but has not previously been used by the organization. It may also involve incorporating a technology that is understood across the industry but has not been used in the organization's predicate products.

An illustrative example can be drawn from image processing system design. In many applications, core image processing functionality has traditionally been implemented using Application Specific Integrated Circuits (ASICs), such as in ultrasound medical imaging

systems. While ASICs can be highly optimized for a given application, they are typically costly to develop and manufacture. With the continued evolution of digital image processing technologies, general-purpose computing platforms, such as graphics processing units (GPUs), have emerged as viable alternatives. The adoption of such technologies often represents a shift in system architecture, which can introduce uncertainty in performance, interfaces, and integration, thereby warranting further analysis.

A similar situation arises in the application of established algorithms to enhance image quality or enable feature identification. Although these algorithms may be well defined and documented, their implementation in a new system context may impose additional requirements related to data, computational resources, or hardware support, introducing further sources of design uncertainty.

As a basic criterion, a New feature should have some level of reference performance information defining customer expectations. Similarly, a New technology should have a reference design or other technical basis for how it is expected to be incorporated. These references provide the starting point for determining whether the design team understands the feature or technology well enough to address potential failure modes and causes of failure.

When an item is classified as New, the FMEA effort should focus on whether the team has sufficient knowledge to translate the reference information into design requirements, integration constraints, and risk controls. From a feature standpoint, this may require competitive analysis or user expectation analysis. From an architecture standpoint, it may require review of reference designs, errata, derating guidance, interface constraints, or other information needed to establish appropriate limits around the design.

5.5.3. *Definition of Unique*

In the proposed framework, **Unique** refers to a feature or technology that is truly first of kind. These are cases where no reference design, established market data, or proven implementation exists. Unique items often involve new intellectual property or emerging technologies whose behavior and limitations are not yet fully known.

For Unique items, the primary question for the FMEA is whether the technology or feature is mature enough for the target market or application environment. Unlike New items, where the concern is the organization's unfamiliarity with an established technology, Unique items involve uncertainty in the technology itself. The design team may not have access to reliable performance boundaries, environmental limits, failure modes, or operational data.

An illustrative example can be drawn from the evaluation of liquid lens technology for use in a compact, variable-focus imaging system in the barcode scanning domain. While the technology demonstrated acceptable performance under nominal conditions, it failed to maintain integrity under required environmental conditions, including temperature, shock, and vibration extremes. Notably, these limitations were not fully understood at the time of evaluation, including by the technology provider, highlighting the uncertainty associated with immature technologies.

A similar form of uncertainty is observed in emerging algorithmic approaches, such as newly developed artificial intelligence methods, where training processes and performance characteristics may be difficult to predict across operating conditions. In such cases, the lack of established performance boundaries and failure modes necessitates additional analysis prior to integration.

A Unique classification therefore points the analysis toward feasibility, maturity, and unknown operating limits. The purpose is to identify uncertainty early enough that the design can pivot, mature the technology, or add appropriate constraints before major design commitments are made.

5.5.4. Definition of Difficult

In the proposed framework, **Difficult** refers to the formal evaluation of design reuse. These are design elements that may be based on known or previously validated technology, but are being reused in a new context that may introduce risk. The design is not necessarily new to the organization, and it may not be first of kind, but the application may still be difficult because of changes in scale, environment, architecture, integration, materials, or performance expectations.

An illustrative example can be drawn from power management architectures. Early smartphones used a power management architecture for single processor devices and wired trickle charging for batteries under 3000 mAh in capacity. Modern tablet devices use batteries with more than 10,000 mAh capacity, have high current wireless fast charging, and support much more complex radio, camera, and processing architecture, yet many of the logic elements, and user facing telemetry remains the same.

Similarly, in the software space, many client/server architectures for software services have been or are being migrated to cloud based Software as a Service (SaaS) using the internet as the basis for connectivity instead of a Local Area Network (LAN). In this sort of porting process the core algorithms and code base remain the same, but the connectivity becomes more difficult.

Difficult items may also include elements of the design that are undergoing scale changes, have had yield or failure issues in previous products, are being refactored with different materials, or are being applied in a more demanding environment. These conditions indicate that prior design provenance may not be sufficient to justify excluding the item from further analysis.

Unlike the New and Unique categories, which focus on unfamiliarity and technological novelty, the Difficult category addresses uncertainty arising from the application of known designs in new contexts. Evaluation is guided by several key considerations:

- Whether there is evidence of stable performance at the required scale;
- Whether prior performance satisfies current requirements or requires improvement;
- Whether system interfaces remain consistent in the new architecture; and
- Whether sufficient historical data exists to support assumptions about performance and integration.

A lack of evidence or uncertainty in these areas does not preclude the use of prior designs; rather, it indicates that the element may require further evaluation. This classification therefore captures cases where reuse is feasible but not straightforward, and where additional analysis is warranted to assess associated design risk.

5.5.5. Reducing Scope, Cost, and Low-Value Analysis

The NUD filter directly addresses several symptomatic problems repeatedly identified in the literature criticizing FMEA, including lack of organizational support, questions about applicability, and the excessive cost of performing FMEAs. These concerns are especially

relevant in incremental or generational design, where a product begins with a proven predicate design and adds selected innovations or modifications.

FMEA standards discuss scope, but they tend to emphasize inclusion rather than exclusion in order to support completeness. While this is understandable from a risk management standpoint, it can lead teams to spend substantial effort analyzing well-proven predicate architecture and design details that may add little value to the current project. In these situations, FMEA can become a documentation exercise rather than a focused design risk activity.

The NUD filter provides a structured way to make exclusion decisions. By focusing on what is New, Unique, or Difficult, the framework directs FMEA effort toward the aspects of the design most likely to contain unknown or elevated risk. At the same time, it allows stable design reuse with strong provenance to proceed through normal design controls rather than being repeatedly reanalyzed in full.

This approach also aligns the FMEA process more closely with the business and engineering rationale for a new project. New projects are rarely funded unless they are expected to provide some value improvement, and those value improvements typically arise from new features, new technologies, new integrations, or challenging reuse. By biasing the filter toward requirements, functionality, and architectural change, the FMEA process becomes focused on the elements of the project most likely to justify the development effort and most likely to introduce design uncertainty.

Finally, application of the NUD criteria can occur using information available in the early stages of the systems V-model. New items generally require reference performance or reference design information. Unique items generally require feasibility analysis or early test data. Difficult items generally require historical performance data or evidence that the reuse context

remains valid. Since the process remains semi-heuristic, the evaluation should be performed by an appropriately cross-functional team to reduce bias and improve the quality of design risk judgment.

These features make NUD a practical front-end filter for FMEA. It reduces unnecessary analysis, preserves attention on the highest-risk areas of the design, and provides a structured basis for deciding where FMEA should be applied within V-model development.

5.6. Requirements-Based Definitions for FMEA Information

A second component of the proposed framework addresses the incompatibility between the information required by classic FMEA and the information available at different stages of the V-model. As shown in the case studies, this incompatibility is often compounded by unclear or inconsistent definitions of key FMEA terms, particularly “failure mode” and “cause of failure.” Ironically, allowing FMEA to be more flexible across stages of the V-model requires stricter definitions of these terms.

In the proposed framework, FMEA information is tied directly to requirements and architecture. This allows the analysis to begin earlier in the development process, when detailed design information may not yet exist, while still maintaining a disciplined structure for identifying failures, causes, effects, and mitigations. The framework therefore shifts the analysis away from asking only how components might fail and toward asking how the system, subsystem, or element might fail to meet its allocated requirements.

5.6.1. Failure Modes as Requirement Failures

In this framework, a failure mode is defined as a failure to meet a requirement. The failure mode should be specific about how the requirement is not met. If it is not clear which requirement the failure mode applies to, then the failure mode is either poorly described or may not actually be a failure mode.

This definition is important because it allows FMEA to align more naturally with the V-model. In early stages of development, the design may not yet be detailed enough to identify all physical or technical causes of failure. However, user needs, system requirements, and allocated performance expectations may already be available. By defining failure modes as failures to meet requirements, the FMEA can begin with the information that exists at that stage of development.

For example, a system-level requirement may state that the system must operate for a specified period of time on battery power. A corresponding failure mode would be that the system does not operate for the required duration when starting from a full battery charge. This failure mode is tied directly to the requirement and describes the unacceptable performance condition, rather than prematurely assuming the specific part or mechanism that failed.

5.6.2. Causes of Failure as Allocated Architectural Sources

Once failure modes are tied to requirements, causes of failure must be tied to the portion of the architecture responsible for satisfying those requirements. In other words, the cause of failure should be associated with the system element, subsystem, module, or design feature responsible for generating the required performance.

This requires that the system architecture exists at a sufficient level of maturity to support at least first-level allocation of requirements to major architectural elements. It also requires a distinction between a requirement and a specification. A requirement defines expected performance, while a specification defines how a portion of the design is instantiated. These distinctions are well established in systems engineering and are especially important in more advanced approaches such as Model-Based Systems Engineering.

Returning to the battery example, if the system fails to meet the required operating time, one possible cause of failure may be that the battery subsystem has inadequate power capacity. This shifts the cause from a vague restatement of the failure condition to a more specific architectural source. Another possible cause could be that another subsystem exceeds its allocated power budget while operating on battery. In that case, the cause of failure is tied to a lower-level requirement allocated to a different part of the architecture.

If the design detail is already mature, the cause of failure may also be expressed as a failure of a specification. For example, if a specific battery design has already been selected, then the risk may involve whether that selected battery meets its specification under expected operating conditions. If the design has not yet been selected, however, the analysis should remain at the requirement or architectural allocation level rather than pretending that detailed design information already exists.

5.6.3. Improving Consistency in Risk Assessment

This requirements-based structure also addresses another recurring problem in FMEA application, which is inconsistent risk assessment. In complex designs, multiple teams may perform FMEAs on different portions of the system. Without a consistent definition of failure

modes and causes, similar failures may be scored differently even when they have the same effect on system performance or the user.

In many cases, inconsistency occurs because teams attempt to evaluate the effect of a cause of failure rather than the failure mode. Put another way, teams may ask, “What happens if part X breaks?” rather than, “What is the impact to the user or system if the system fails to do Y?” The first question can lead to inconsistent scoring because it depends heavily on the assumed failure mechanism. The second question creates a more stable basis for assessment because it evaluates the impact of failing to meet expected performance.

By forcing failure modes to be described in the language of requirements and expected performance, the framework creates a more consistent basis for severity evaluation. A lower-level requirement failure can be evaluated in relation to its impact on the higher-level parent requirement. This supports decomposition because the effect of a lower-level failure can be traced upward through the architecture to determine its system-level or user-level consequence.

This approach does not prevent the use of existing or improved risk assessment methods. Teams may still use RPN, criticality matrices, fuzzy scoring, AI-supported methods, or other prioritization approaches. The framework instead provides a more consistent structure for defining what is being scored. By tying failure modes to requirements and causes of failure to allocated architecture, the analysis becomes better aligned with the V-model and less dependent on inconsistent interpretations across teams.

5.6.4. Contribution to the Proposed Framework

The requirements-based definition structure is central to making FMEA usable earlier in the V-model. It allows teams to perform meaningful risk analysis before complete detailed

design information is available, while avoiding vague or poorly constructed FMEA lines. Failure modes are grounded in requirements, causes of failure are grounded in architectural responsibility, and risk assessment is grounded in the impact of unmet performance expectations.

This structure also supports the later recursive elements of the framework. Once a failure mode is identified and a cause of failure is associated with a lower-level architectural element, mitigations can be expressed as new or revised requirements allocated into the architecture. Those requirements can then be evaluated again through the NUD filter to determine whether the mitigation itself introduces new design risk. In this way, requirements-based definitions provide the bridge between early-stage risk identification, FMEA decomposition, and V-model development.

5.7. Decomposition, Recursion, and Requirements as Mitigations

A third component of the proposed framework addresses the difficulty of aligning FMEA with V-model decomposition. Classic FMEA is often applied as a bottom-up analysis, beginning with components or elements and asking how they might fail. In contrast, the V-model begins with user needs and system-level requirements, then decomposes those requirements into lower-level architecture and design detail. To better align FMEA with this structure, the proposed framework shifts the analysis toward system performance and allows FMEA to decompose recursively as the design matures.

5.7.1. Shifting FMEA Toward System Performance

A key “left shift” in this framework is to focus first on failures of the system to do something, rather than immediately focusing on ways that elements of the system can fail. In

other words, the analysis begins with expected performance, expressed through requirements, and then asks how the system, subsystem, or element might fail to meet those requirements.

This shift provides several improvements in alignment with the V-model. At early stages of development, teams may have user needs, system requirements, and high-level architecture, but they may not yet have detailed design information. A traditional bottom-up FMEA is difficult to perform at this point because the design elements, interfaces, and failure mechanisms may not be fully defined. A requirements-based FMEA, by contrast, can begin earlier because it uses the information available at that stage of development.

The tradeoff is that early-stage FMEA may not always identify a tangible design feature to modify. Instead, it may identify a gap, uncertainty, or risk in the requirements or architecture. This is a useful outcome because it allows the team to improve the design definition before committing to detailed design decisions.

5.7.2. Mitigations as New or Revised Requirements

When the analysis is performed at the system or architectural level, the mitigation to a cause of failure may not be a specific design change. Instead, the mitigation may be a new or revised requirement allocated to a lower level of the architecture. This allows risk reduction to occur before detailed design is complete.

For example, a system-level requirement may state that the system must operate on battery power for a specified number of hours. A corresponding failure mode could be that the system does not operate for the required duration starting from a full battery charge. One possible cause of failure could be inadequate power capacity in the battery system. If detailed battery design has not yet occurred, the mitigation may be written as a new lower-level

requirement, such as requiring the battery subsystem to provide a specified capacity margin relative to the projected power budget.

By treating the mitigation as a requirement, the framework allows the lower-level design to be defined correctly from the beginning rather than requiring rework after a conventional FMEA identifies a problem later in the process. This approach does not prevent the use of fault tree analysis, reliability modeling, prototype testing, or other tools. Rather, it improves the quality of the inputs to those activities by clarifying what the lower-level design must accomplish.

5.7.3. Recursive Application of the NUD Filter

Once a mitigation is expressed as a new or revised requirement, that requirement must be evaluated through the NUD filter. This recursive step is important because a mitigation intended to reduce one risk may introduce another design risk.

Continuing the battery example, a mitigation may require the battery subsystem to provide 200% of the projected power budget in order to meet the required operating time. At that point, the new requirement must be reviewed against the NUD criteria. If the required power density is well within proven design capability, then the requirement may proceed through standard design practices. However, if the required power density exceeds proven technology for a high-reliability battery system, the mitigation itself becomes a new design risk.

In that case, the NUD filter flags the mitigation for further analysis. The team may then perform a more focused, lower-level FMEA, conduct a feasibility analysis, pursue targeted testing, or evaluate alternative design approaches. This ensures that the process does not simply move risk from one part of the system to another without recognizing it.

5.7.4. Closure of the Recursive Process

The recursive process continues until the mitigation no longer introduces a New, Unique, or Difficult element requiring additional analysis, or until the analysis reaches a level of design detail where conventional FMEA can be effectively applied. This creates a natural closure point for the FMEA process within the V-model.

If a new requirement is evaluated through NUD and does not introduce additional design risk, then standard design practices can proceed. If the requirement does introduce design risk, then the process decomposes to the next level of architecture and the analysis continues. In this way, the framework supports decomposition without requiring the entire system to be analyzed at the same level of detail.

This recursive structure also helps focus expertise where it is most needed. Rather than requiring teams to perform detailed FMEA across all system elements, the framework identifies where uncertainty or design risk remains and directs analysis toward those areas. This reduces unnecessary analysis while preserving the ability to apply conventional FMEA where it provides value.

5.7.5. Design Risk Mapping Across Decomposition

An additional benefit of this recursive process is that it creates a “design map” of the areas of the system that require lower-level FMEAs and the depth to which those analyses must proceed. This map provides the systems engineer with an evolving view of where design risks are being resolved and where new risks are being introduced as the project progresses.

For complex projects, this design map can provide a practical way to monitor risk across the architecture. Areas that do not trigger NUD criteria can proceed through normal design

practices, while areas that continue to trigger New, Unique, or Difficult classifications can be tracked for deeper analysis, additional testing, or more careful architectural management.

In this way, decomposition and recursion are not simply procedural features of the proposed framework. They provide a mechanism for aligning FMEA with the V-model, managing design risk as information becomes available, and maintaining visibility into how risk evolves across the architecture.

5.8. Managing Cross Discipline Complexity Through CtQ Designation

A final component of the proposed framework addresses the difficulty of managing risk across increasingly interconnected engineering disciplines. This issue appears in multiple forms in FMEA practice, particularly as modern systems become more dependent on interactions among hardware, software, firmware, user interfaces, manufacturing processes, and external integrations. These interactions are especially important in systems where software and hardware are tightly coupled, and where design decisions in one discipline may create requirements or constraints in another.

5.8.1. Limitation of Traditional CtQ Use

In classic FMEA, the term “Critical to Quality,” or CtQ, is often associated with some form of importance designation. In many cases, this designation is tied to severity scoring or qualitative descriptions such as “loss of critical function” or “extreme dissatisfier.” These types of criteria can be difficult to apply consistently and may become redundant with the severity score already present in the FMEA.

As a result, CtQ designations may not provide much additional information about how risk is distributed across the architecture or how mitigations affect other parts of the system, and are be difficult to apply consistently In the context of complex V-model development, however, this kind of cross-architecture visibility is important because mitigations may create new requirements outside the discipline or subsystem where the original risk was identified.

5.8.2. CtQ as a Cross Discipline Tracking Mechanism

In the proposed framework, CtQ is repurposed as a classifier for mitigations or requirements that cross architectural or engineering discipline boundaries. Rather than using CtQ only as a general importance label, the framework uses it to identify when a mitigation in one part of the design creates a requirement in another part of the system.

For example, a high-power-density battery system may be feasible, but the FMEA may identify that charge and discharge rates must be strictly controlled to prevent hazardous battery behavior. The resulting requirement for charge and discharge measurement and control may then become a software requirement, even though the original risk was associated with the battery subsystem. In this case, the software control requirement would be designated as CtQ because it crosses the boundary from hardware risk mitigation into software implementation.

This use of CtQ provides a more specific and actionable purpose. It helps identify requirements or mitigations that must be tracked across disciplines because their effectiveness depends on coordination between different parts or layers of the architecture. Additional hardware requirements created to support the battery mitigation could also be designated as CtQ if they are outside the original battery subsystem.

5.8.3. CtQ as a Design Coupling Indicator

Using CtQ in this way creates a secondary map of cross-discipline dependencies within the design. This map can help the systems engineer understand where mitigations create coupling between architecture elements, subsystems, or engineering disciplines. Such coupling is not necessarily undesirable, but it should be visible because it can introduce integration risk, verification complexity, or coordination requirements.

This CtQ map also provides insight into the degree of loose coupling and independence between different elements of the architecture, which is a well-established consideration in good systems design. A design with many CtQ designations crossing subsystem or discipline boundaries may require additional systems engineering attention, interface management, verification planning, or cross-functional coordination. Thus, CtQ becomes not simply a quality label, but a mechanism for tracking how risk mitigations propagate through the system architecture.

5.9. Relationship to Existing FMEA Improvements

The proposed framework is not intended to replace existing improvements to FMEA scoring, prioritization, failure mode elicitation, or risk categorization. Instead, it provides a higher-level structure for determining where and when those methods should be applied within V-model development.

As discussed in previous chapters, much of the FMEA improvement literature focuses on execution-level problems, such as improving RPN, refining scoring models, applying fuzzy methods, using expert systems, or incorporating artificial intelligence to support failure identification and prioritization. These methods may improve specific aspects of FMEA

execution, but they do not necessarily address the broader structural mismatch between FMEA and V-model development.

The proposed framework addresses that higher-level mismatch. The NUD filter helps determine where FMEA is most needed, the requirements-based definitions clarify what is being analyzed, and the recursive decomposition structure allows FMEA to follow the staged availability of information in the V-model. Within that structure, improved scoring models, fuzzy FMEA methods, expert systems, AI-supported tools, or other methods can still be used during specific execution phases.

In this sense, the framework is intended to be complementary to existing FMEA research. It does not prescribe a single scoring model or failure identification technique. Rather, it provides a systems engineering structure for applying those techniques more effectively by aligning FMEA scope, timing, and decomposition with the development process.

5.10. Meta Analysis of Previous FMEA Examples using the new Framework

Chapter 4 presented several case study examples that illustrated common challenges in FMEA implementation. Building on the FMEA model introduced in Chapter 5, this section revisits those case study examples using the proposed framework as a form of meta-analysis. The purpose is to demonstrate how the framework can be applied and how the issues identified in the case studies may be avoided.

5.10.1. Image Stabilization Feature Case Study Reviewed

Table 11 reproduces Table 7 for ease of reference before applying the proposed framework.

Table 11: Image Stabilization Case Study

Failure Mode Summary								Current Design Control		
Item/ Function	Requirement	Potential Failure Mode	Potential Effects of Failure	Severity	Cause of Failure	Current Prevention	Occurrence	Current Detection	Detection	RPN
Motion compensation and persistence	The Image Compensation option provides the ability for the user to enable digital image stabilization	CPU cannot keep up with algorithm processing demands	System crashes	8	Inefficien t coding of algorithm	None	5	System integratio n testing of extreme imaging cases	1	40

Following the new framework, the first step would be to classify the requirement, “The image compensation option provides the ability for the user to enable digital image stabilization”, using the New, Unique and Difficult criteria. Image stabilization algorithms are well documented in literature for a variety of types of images. Therefore, it would not fall under Unique. In this specific case, the author is aware that the algorithm in question was novel to the company and therefore would classify this as New. However, even assuming this was an enhancement to an existing algorithm, based on review of the predicate performance and the expectation of data to be managed in the implementation this would have reasonably classified as Difficult and therefore merit FMEA consideration.

Table 12 shows how the new framework could be applied to this requirement. Rather than treating the primary concern as a CPU crash caused by inefficient coding, the revised

FMEA focuses on whether the image stabilization requirement can be achieved and allocates the potential causes across multiple architectural layers, including the data pipeline, software algorithm, and hardware compute capability.

Table 12: Image Stabilization under New Framework

Failure Mode Summary								Current Design Control		
Item/ Function	Requirement	Potential Failure Mode	Potential Effects of Failure	Severity	Cause of Failure	Current Preventio n	Occurrence	Current Detection	Detection	RPN
Motion compensation and persistence	The Image Compensation option provides the ability for the user to enable digital image stabilization	Image cannot be stabilized, continuing to leave a jittery image	Customer cannot resolve clinically relevant image detail due to jitter.	8	Lack of adequate data processing capacity (pipeline)	None	5	None	10	400
Motion compensation and persistence	The Image Compensation option provides the ability for the user to enable digital image stabilization	Image cannot be stabilized, continuing to leave a jittery image	Customer cannot resolve clinically relevant image detail due to jitter.	8	Lack of adequate data processing capacity (Software)	None	5	None	10	400
Motion compensation and persistence	The Image Compensation option provides the ability for the user to enable digital image stabilization	Image cannot be stabilized, continuing to leave a jittery image	Customer cannot resolve clinically relevant image detail due to jitter.	8	Lack of adequate data processing capacity (Compute)	None	5	None	10	400

Under the new framework, integration testing is no longer treated as an appropriate detection control because it occurs after the relevant subsystems have already been developed. Instead, the revised analysis drives mitigation activities toward the architectural layers where the issue would need to be prevented. Although not shown in this portion of the table, a clear action

item would be to perform early design analysis to understand the data processing requirements for the application with enough fidelity to define appropriate requirements at each relevant layer of the architecture.

It is also important to note that, in the original case study, the failure mode was described as a system crash. While a crash is a possible outcome, it is not directly tied to the requirement itself. By focusing on the need expressed by the requirement, the framework helps identify and reduce the conditions that could eventually lead to a crash. A failure mode stating that the system crashes when attempting to enable image compensation could still reasonably indicate that the requirement is not met, but it would be processed through the same requirement-focused logic shown in the revised example.

5.10.2. System Over Temperature Management Failure Reviewed

Again, for ease of reference, Table 13 reproduces Table 8 regarding the consideration of a possible system over temperature condition.

Table 13: System Over Temperature Management Case Study

Failure Mode Summary								Current Design Control		
Item/ Function	Requirement	Potential Failure Mode	Potential Effects of Failure	Severity	Cause of Failure	Current Prevention	Occurrence	Current Detection	Detection	RPN
System Temperature Monitoring	System warns user of overtemp. condition then shuts down within time period	System Shuts down	System not available	8	Heat buildup due to extended system use at high GPU utilization	Temp monitoring	5	Design Qualification Testing	5	200

As before, the first step would be to apply the New, Unique or Difficult filter. In this instance, over temperature conditions for high use electronics, particularly in image processing is a known design issue and predicate products had mechanisms for monitoring and managing them which were being reused. However, in this case, where new high utilization features, such as image stabilization were being added, the additional feature of warning the user of an impending over temperature condition so they could make choices in their clinical application before a more severe problem occurred was being added. Therefore, this feature could be classified as a Difficult requiring further analysis, as shown in Table 14 below.

Table 14: System Over Temperature Management Reviewed

Failure Mode Summary								Current Design Control		
Item/ Function	Requirement	Potential Failure Mode	Potential Effects of Failure	Severity	Cause of Failure	Current Prevention	Occurrence	Current Detection	Detection	RPN
System Temperature Monitoring	System warns user of overtemp. condition then shuts down <u>after</u> time period	System does not provide warning	System shuts down unexpectedly (to the user)	10	Software does not pick up temp sensor value	None	5	None	5	250

In this case, the case study highlighted a problem in clear definition of a failure mode leading to an incorrect mitigation approach. In application of the new framework, the first problem would be the lack of clarity in the requirement. The intent of the feature is to provide a warning to the user with a time frame of when the system would shut down due to an over temperature condition. The previous wording confused this. Thus, in this case the analysis would drive a change in the requirement to clarify and a clear failure mode that the system does not provide the warning. This then drives a revised effect of failure where the system shuts down

unexpectedly to the user, which is a higher severity problem due to the potential that the user could be in a critical aspect of a clinical procedure. As in the previous case study, multiple causes could lead to the same failure mode. Further, a failure mode of the amount of time being inaccurate could be associated with the same requirement leading to its own line and scoring.

5.10.3. UI Image Optimization Controls Reviewed

Table 15 reproduces Table 9 for ease of reference and summarizes the UI controls case study. In applying the NUD process, the addition of new buttons to the user interface using the existing AppUI Library would not be expected to introduce sufficient incremental design risk to require FMEA. This rationale would be documented, allowing the design team to focus FMEA activities on higher-risk areas.

Table 15: UI Image Optimization Controls Case Study

Failure Mode Summary								Current Design Control		
Item/ Function	Requirement	Potential Failure Mode	Potential Effects of Failure	Severity	Cause of Failure	Current Preventi on	Occurrence	Current Detection	Detection	RPN
HD Image UI Controls	System has requirements for specific UI controls to support user image optimization of the dynamic video	Incompatible system state created by the UI Controls	System Crash	8	Incompatible system state created by the UI Controls	Existing AppUI Library supports controls.	3	Integration Software Testing	3	72

5.10.4. Border Computation Case Study Reviewed

As in the previous examples, Table 16 reproduces Table 10 to allow comparison between the original case study and the updated analysis. Using the NUD process, the Border

Computation Algorithm would be classified as a novel machine learning AI capability intended to support physician identification of clinically relevant pathology artifacts in video. While image artifacts associated with clinically relevant pathology are well documented in ultrasound, real-time automated identification of those artifacts is not. In addition, the feature was not available in competitor products, nor was reference or research data available to support the implementation. Although academic work on boundary identification in ultrasound images existed, it was not available in an implementable form. Therefore, this feature would clearly be classified as Unique, meriting further analysis.

Table 16: Border Computation Algorithm Case Study Case Study

Failure Mode Summary								Current Design Control		
Item/ Function	Requirement	Potential Failure Mode	Potential Effects of Failure	Severity	Cause of Failure	Current Preventio n	Occurrence	Current Detection	Detection	RPN
Border Computation Algorithm	During continuous video, the Border Computation algorithm provides real time boundaries for desired artifacts in complex video	Incorrect border identification	User Mis-interpretation of the result	10	Algorithm cannot accurately identify boundaries in image	None	3	System integration Testing	3	90

Table 17 applies the stricter rules (test processes or test data cannot represent mitigation, mitigation must exist as a design capability) regarding Prevention and Detection in the current state and therefore increases their scores from the original case. As a note, the company's practice was to assign cases with no current controls to the midpoint of the scoring scale unless a

different score was justified. Simply maintaining this discipline moves the RPN to 250, requiring mitigation. In this case, the natural mitigation would be to define a requirement limiting allowable algorithm misclassification error, which would support evaluation of the algorithm's maturity against test data sets. This would also likely drive the need for tools to perform that evaluation, as well as other design activities that were not originally scoped in the project.

Table 17: Border Computation Case Study Reviewed

Failure Mode Summary								Current Design Control		
Item/ Function	Requirement	Potential Failure Mode	Potential Effects of Failure	Severity	Cause of Failure	Current Prevention	Occurrence	Current Detection	Detection	RPN
Border Computation Algorithm	During continuous video, the Border Computation algorithm provides real time boundaries for desired artifacts in complex video	Incorrect border identification	User misses or miss classifies clinically relevant image artifacts	10	Algorithm cannot accurately identify boundaries in image	None	5	None	5	250

5.11. Conclusion and Proposal for Validation

The framework proposed in this chapter addresses the systemic problems identified in Research Aim 1 by modifying how FMEA is scoped, structured, decomposed, and connected to V-model development. The framework begins with the NUD filter, which identifies New, Unique, and Difficult areas of the design that warrant focused risk analysis. It then uses requirements-based definitions of failure modes and causes of failure to support analysis at

stages where detailed design information may not yet be available. As mitigations are identified, they may be expressed as new or revised requirements and reviewed again through the NUD filter to determine whether they introduce additional design risk. Finally, CtQ designations provide a mechanism for tracking mitigations and requirements that cross architectural or engineering discipline boundaries.

These elements provide a framework for adapting FMEA to the realities of complex V-model development. The framework does not reject FMEA or replace its core logic. Instead, it preserves the value of FMEA while addressing the timing, scope, decomposition, and information-availability problems identified in the previous chapters.

The next step is to evaluate whether this framework provides practical benefit in application. Since proliferation is one of the most persistent and practically significant problems identified in the literature, the NUD filtering portion of the framework was selected for initial validation. This portion can be evaluated using early-stage requirements and architecture information, making it suitable for testing in a real development project. The following chapter presents the validation of the NUD design risk filtering approach in a high complexity industrial development context.

Chapter 6. Aim 3 - Validate Design Risk Filtering

6.1. Introduction

Research Aim 3 validates the design risk filtering portion of the improved FMEA framework developed in Research Aim 2. As discussed in the previous chapters, one of the most persistent challenges in applying FMEA to complex systems is proliferation, where the number of potential FMEAs, failure modes, and causes of failure can grow rapidly as product complexity and feature density increase. The proposed framework addresses this issue first through the New, Unique, and Difficult (NUD) filter, which is intended to identify areas of a design that warrant focused risk analysis before detailed FMEA activity begins.

This chapter presents an initial validation of the NUD design risk filtering process in a high-complexity industrial development context. The validation examines whether the method can be taught to a team unfamiliar with it, whether users perceive value in the method and express willingness to use it again, and what level of facilitation or contextually relevant examples are needed to support deployment. Accordingly, this chapter addresses the following research questions:

- **RQ3.1:** How difficult is it to teach the proposed methodology to a team that is entirely unfamiliar with it?
- **RQ3.2:** To what extent do users perceive value in the proposed methodology and express willingness to use it in future projects?
- **RQ3.3:** What level of facilitation and contextually relevant examples are needed to support effective deployment of the proposed methodology?

The chapter first summarizes the NUD framework as applied in the validation study, including its purpose, assumptions, constraints, definitions, and execution process. It then describes the industrial case study, participants, procedure, and survey design. Finally, the chapter presents the results, discusses the findings in relation to the research questions, and identifies limitations and opportunities for future work.

6.2. NUD Framework for Validation

The NUD framework is the design risk filtering portion of the broader FMEA improvement framework proposed in Chapter 5. It was selected for initial validation because it directly addresses FMEA proliferation and can be applied using information available early in the V-model, including user needs, system requirements, and high-level architecture. This makes it appropriate for validation in an active development project, before detailed design information is fully available.

The purpose of the NUD framework is to reduce the scope of downstream FMEA analysis by identifying areas of a design that are most likely to contribute to elevated risk. Rather than relying on comprehensive enumeration of all possible failure modes, the framework provides a structured approach for determining where detailed analysis is most warranted. The intent is to focus limited engineering and risk management resources on areas where uncertainty is greatest, thereby improving the efficiency and effectiveness of risk assessment.

6.3. Assumptions and Constraints

The NUD framework was developed within the context of modern product development, where designs are rarely “green field” and instead incorporate varying degrees of technology reuse. The framework assumes that system elements that are well understood, previously

validated, and used in a similar context do not require the same level of failure analysis as elements that introduce uncertainty.

However, reuse does not inherently imply low risk. The applicability of prior designs depends on factors such as scale, context, performance requirements, interfaces, operating environment, and integration conditions. The framework therefore distinguishes between routine reuse and reuse that introduces uncertainty requiring further analysis.

The framework is intended for application during early stages of development, when available information is limited to user needs, system requirements, and high-level architecture (i.e., early stages of the V-model). Detailed design information is not assumed. This constraint is important because it aligns with the intended timing of FMEA, which is often recommended early in the development lifecycle but is frequently limited by the lack of detailed system information.

6.4. Definitions of New, Unique, and Difficult

The NUD framework classifies system elements into three categories: New, Unique, or Difficult. Each category represents a distinct source of uncertainty and potential design risk. These classifications are not limited to the system level and may be applied at any level of architectural or requirements decomposition. This flexibility allows the framework to be used iteratively throughout development and supports integration with the top-down FMEA approach proposed in Chapter 5.

New refers to system elements that are established in the state of the art but unfamiliar to the designing organization. In these cases, reference designs, algorithms, or implementations may be available, but uncertainty arises from the organization's limited experience applying

them within the intended context. The key question is whether the organization has sufficient knowledge to implement the technology without additional analysis to reduce design risk.

Unique refers to system elements that represent novel or proprietary technologies for which no established reference implementation exists. These elements are typically associated with emerging technologies, new intellectual property, or immature design concepts. Unlike New elements, where uncertainty comes from organizational unfamiliarity, Unique elements involve uncertainty inherent to the technology itself, including unknown behavior under operating conditions or limited validation data.

Difficult refers to system elements that are based on existing or previously validated designs but present challenges in reuse due to changes in context, requirements, scale, interfaces, or operating conditions. The purpose of this classification is to evaluate whether design reuse can be applied reliably within the proposed architecture or whether the reuse introduces uncertainty that warrants further analysis.

6.5. Application and Execution of the NUD Framework

The NUD framework is applied during early stages of system development, when user needs, system-level requirements, and high-level architecture have been defined but detailed design information remains limited. At this stage, system elements are evaluated to distinguish between those representing established design reuse and those introducing incremental or novel aspects within the system.

The application of the framework consists of three primary activities: identification, classification, and disposition. First, system elements are identified and separated into those representing established or predicate design elements and those introducing incremental changes

relative to prior designs. Elements identified as incremental are then evaluated using the NUD classification criteria, while elements based on established or predicate design are considered low design risk and therefore can be addressed with normal design practices. It is a relevant note that those aspects classified as outside the scope of the NUD analysis do not represent *no* risk, rather that the organization's normal design practices can address them without incremental focus. Where an element satisfies one or more of the New, Unique, or Difficult conditions, it is flagged for further consideration.

Following classification, identified elements are evaluated to determine whether further analysis is required and, if so, whether FMEA is an appropriate method. This separation between classification and disposition is intentional, as it allows the identification of potential risk areas to remain independent of downstream decisions regarding mitigation approach. In practice, this distinction supports more consistent identification of risk-prone elements prior to resource-driven prioritization.

The process is typically performed collaboratively by a cross-functional team with expertise spanning system architecture, domain-specific engineering disciplines, and user or operational considerations. While specific roles may vary by organization, effective execution relies on the ability to evaluate system elements from multiple perspectives, including functional intent, implementation approach, and integration context.

In practice, execution often includes preparatory analysis (i.e., "pre-work") to distinguish between predicate and incremental aspects of the design, particularly at the requirements and architectural levels. This pre-work may involve identifying elements associated with known design reuse, as well as those introducing new interfaces, technologies, or performance expectations. Reused elements with known operational concerns, such as manufacturing yield

limitations, supply chain risks, or adverse field performance, may also be included for evaluation and are often classified as Difficult.

During classification, stakeholders contribute domain-specific insight to assess whether elements meet the criteria for New, Unique, or Difficult. In many cases, classification is straightforward; for example, incremental feature additions based on existing implementations may not trigger any category, whereas changes that introduce new architectural demands or performance constraints may be classified as New.

Determining the appropriate response for classified elements is more complex. For this reason, classification is intentionally separated from disposition, allowing potential risk areas to be identified independently of downstream mitigation decisions. The framework is designed to bias toward further analysis for classified elements, with the expectation that clear evidence or supporting analysis is required to justify exclusion from additional risk assessment.

The application of the framework is inherently iterative and may inform additional engineering activities, such as trade studies, architectural refinement, or targeted testing. In some cases, classification may lead to further analysis prior to final design decisions, followed by application of FMEA or other risk assessment methods on the resulting design.

In many cases, classification is relatively straightforward when considered in the context of existing system capabilities. For example, in an imaging system, the addition of a new measurement capability to an existing suite of measurements may not be classified as New, Unique, or Difficult, even if it represents a novel feature from the user perspective. In such cases, the implementation may rely on established classes of algorithms already present within the system, and therefore does not introduce meaningful design uncertainty.

In contrast, changes that affect underlying system performance requirements may trigger classification. For instance, increasing system resolution or refresh rate can impose significantly higher data throughput demands, potentially requiring modifications to the imaging hardware or a transition to a different processing architecture. Such changes introduce uncertainty in system behavior and integration, and would typically be classified as New.

The overall outcome of the process is a reduced and prioritized set of system elements requiring focused risk assessment. These elements may then be analyzed using FMEA or alternative methods (e.g., prototype testing, fault tree analysis, or simulation), depending on the nature of the identified uncertainty. In many cases, top down FMEA may be used to better articulate the possible architectural causes for a failure mode leading to an alternative tool to provide information toward an effective mitigation early in design. In this way, the NUD framework serves to guide both the scope and timing of risk analysis activities within the development process.

Figure 6, taken from a NUD training course developed by the author, illustrates the conceptual activity flow of the framework. The table on the left represents the system requirements, while the block diagram on the right represents a simplified top-level architecture, including an example of how one portion of the architecture may be decomposed into a lower-level block diagram. In practice, the process can be simplified into the three primary steps shown.

6.6.1. Case Study Context

The validation study was conducted at a company operating in the scientific and industrial laser domain, representing a different industry from the original development environment. The company was engaged in the development of a novel laser system involving both significant design reuse and new architectural elements.

The organization maintained an established design control process, including formal approaches to design risk assessment and FMEA. However, participants reported challenges with the efficiency and effectiveness of these processes, providing a relevant context for evaluating the NUD framework.

No financial compensation was exchanged as part of this collaboration. Instead, one of the authors provided training on the NUD framework and facilitated its application within the laser development project. In return, participants completed surveys regarding their experience with the process.

6.6.2. Participants

The NUD process was applied by a cross-functional engineering team consisting of six participants, see Table 18. The team collectively represented expertise in system architecture, engineering disciplines, manufacturing, and quality. While no dedicated product marketing representative was included, detailed information regarding the target customer and application was available, as the product was being developed for a specific use case. One of the authors served as facilitator during initial execution of the process, leveraging prior experience with both the NUD framework and the application domain (i.e., gas laser technology).

Table 18: Participant Team Representation

Role at Company	Role in NUD Process
Director of Laser Engineering	Lead Systems Architect
Manufacturing Engineering Manager	Lead Operations Engineer
Senior Laser Engineer	Discipline Engineer Lead
Senior Electrical Engineering Manager	Discipline Engineer Lead
Mechanical Development Manager	Discipline Engineer Lead
Quality Engineering Manager	Scribe and Project Manager

6.6.3. Procedure

The validation study consisted of three phases: training, initial facilitated execution, and independent application. Prior to execution, preparatory materials were provided, including system-level requirements and a high-level architecture diagram. The facilitator developed and delivered a structured training session introducing the NUD definitions, application process, and representative examples. The training was conducted remotely using a slide deck (12 slides) and included approximately 30 minutes of instruction followed by 15 minutes of questions and discussion.

Following training, an initial working session (approximately 45 minutes) was conducted in which the facilitator guided the team through application of the framework to selected system elements. This session was intended to establish a shared understanding of the classification criteria and overall process through representative examples. After this initial session, the team continued applying the NUD framework independently over the course of several weeks. During this period, three additional follow-up sessions were conducted with the facilitator to address questions and clarify edge cases encountered during application.

These follow-up discussions focused on several recurring areas of uncertainty, including the interpretation of supplier-related risks and their relationship to FMEA, the evaluation of

environmental requirements (such as shock and vibration) in the context of system scaling, and the consideration of reliability requirements and latent failure risks. In these cases, the framework supported structured discussion of both classification and appropriate downstream analysis. For example, supplier-related issues were identified as appropriate to flag for evaluation but were determined to be better addressed through supplier-specific quality processes rather than FMEA. Similarly, environmental requirements associated with increased system scale were classified as Difficult due to changes in mass and operating conditions, while reliability concerns related to high uptime requirements led to classification of certain elements as New or Difficult, depending on whether uncertainty arose from new technology or reuse under more demanding conditions.

These interactions illustrate how classification decisions informed subsequent determination of appropriate analysis methods, including when FMEA was applicable and when alternative approaches, such as trade studies or targeted testing, were more suitable.

6.6.4. Survey Design and Data Collection

Upon completion of the NUD process, participants were asked to complete a structured survey consisting of eleven questions. These included two questions assessing participant background and familiarity with FMEA, two questions evaluating the training and facilitation, four questions assessing the NUD process using Likert response scales, and three open-ended questions for qualitative feedback.

To compare the NUD framework with the organization's existing risk assessment process (i.e., traditional FMEA), responses were collected using a seven-point scale ranging from -3 to

+3. A value of 0 indicated parity with the existing process, positive values indicated improvement, and negative values indicated reduced effectiveness.

All survey questions were accompanied by descriptive anchors to promote consistent interpretation. Participants were instructed to complete the survey independently.

6.7. Results

6.7.1. Participant Background with FMEA

All participants reported having prior experience with FMEA, ranging from moderate familiarity to advanced proficiency, including individuals with experience facilitating FMEA activities (see Table 19). This distribution of experience provided a relevant basis for evaluating the NUD framework relative to established practices.

Table 19: Participant Experience with FMEA

Familiarity	Description	Count (%)
Basic	I have participated in a few FMEAs as a contributor but only one kind (e.g., DFMEA or PFMEA) on only one design, product or process.	0 (0%)
Moderate	I have participated in FMEAs on a couple of different designs, processes or products. But still only one kind of FMEA.	2 (33.3%)
Skilled	I have participated in different kinds of FMEAs and am comfortable with executing them.	1 (16.7%)
Advanced	I have facilitated FMEAs of at least one type and am comfortable doing so, including teaching & answering questions about the method.	2 (33.3%)
Mastery	I am an SME in the company on FMEA and feel comfortable applying it in most situations. I have written procedures and taught how to do FMEA.	1 (16.7%)

6.7.2. Training Effectiveness

Effective evaluation of the NUD framework required that participants first understood the underlying concepts and application process. To assess this, participants were asked whether the training in NUD risk filtering effectively communicated the method.

All participants (100%) indicated that the training was effective. Responses were distributed between those who found the concept clear and directly applicable to design tasks, and those who emphasized the importance of facilitated examples in developing a practical understanding. Specifically, half of the participants indicated that the examples and guided application were essential to achieving useful understanding, while the remaining participants reported that the method was sufficiently clear to apply directly, with some noting the ability to extend the approach beyond design into process-related contexts.

These results suggest that the NUD framework can be effectively communicated within a relatively short training session, particularly when supported by representative examples and initial facilitation.

6.7.3. Perceived Value Relative to Existing Process

Participant responses to the Likert-scale questions are summarized in Figure 7. All responses were non-negative, indicating that the NUD framework was perceived as equivalent to or better than the organization's existing risk assessment process across all evaluated dimensions. All but one response were positive, with a single neutral rating (17%) observed for design risk insight, reflecting consistently favorable evaluation across participants.

The survey assessed five aspects of framework performance. NUD effectiveness captures how well the process supported cross-functional discussion and decision-making, while

understanding of NUD reflects participants' ability to apply the method following training. FMEA guidance reflects the extent to which the framework supports selection of appropriate analysis methods. Risk filtering confidence evaluates whether the framework appropriately distinguishes between areas requiring further analysis and those that can be deprioritized. Lastly, design risk insight captures whether the process improves understanding of where risk resides within the system.

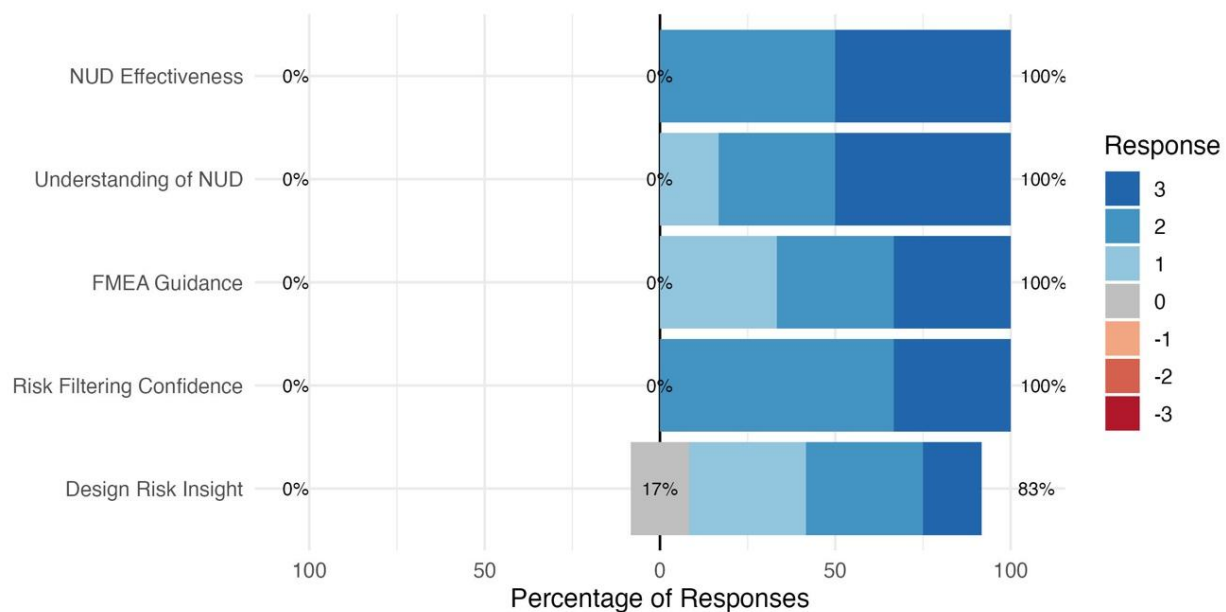


Figure 7: Participant Evaluation of NUD Framework (n = 6)

6.7.4. Qualitative Feedback

Open-ended responses were generally positive and provided additional insight into how the NUD framework was perceived in practice. All participants indicated that they would use the approach in future design efforts. One respondent noted that the process would have been more valuable if applied earlier in development, before key design decisions were committed: “In

future programs, I would incorporate it earlier, at the architectural definition stage. Applying it upfront would enable earlier identification of key risks, assumptions, and interface boundaries, and would make the subsequent FMEA process more efficient and focused. I would use it again, particularly as a structured alignment tool early in the design phase.”

Participants consistently described the primary value of the framework in terms of improved efficiency and focus. Responses emphasized that the process enabled rapid alignment between system requirements and architecture, while helping to identify the most significant areas of design risk. As one participant noted, “the NUD process added value ... and was a very effective use of our time. In just a couple of hours we had paired the architecture with the requirements and identified our most concerning risks.” Another participant similarly observed that the framework “helped narrow down what actually needed attention instead of trying to analyze everything.”

While the overall reception was positive, some feedback highlighted areas for refinement. One participant noted that the concept of architectural levels introduced additional complexity and was perceived as a distraction during application: “I did come away feeling like the “levels of architecture” concept was a bit nebulous and sort of distracted us.” Additionally, although the process was viewed as effective, only one respondent explicitly described the experience as enjoyable, suggesting that the value of the framework lies primarily in its utility rather than user experience.

Several responses provided additional context for their ratings on the Likert scale. Multiple respondents indicated that their selection of +2 rather than +3 for risk filtering confidence reflected characteristics of the design rather than limitations of the NUD process itself. For example, one participant stated, “I picked 2, but that's only because lasers are very

unpredictable beasts to design.” Similarly, another noted, “I score this '2' only because in my experience, failures often occur where you least expect them. ...This is where I feel NUD provides an improved rational for identifying areas to receive heightened focus regardless of the design engineers’ initial confidence level in them.”

6.8. Discussion

This study introduced and evaluated the NUD framework as a structured approach for scoping and prioritizing design risk analysis in early-stage system development. The framework was applied in an industrial case study involving a cross-functional engineering team, and its performance was assessed through a combination of structured survey responses and qualitative feedback.

The results indicate a consistently positive evaluation of the NUD framework across all assessed dimensions. All responses were non-negative, with only a single neutral response observed, suggesting that participants perceived the approach as at least equivalent to, and generally more effective than, their existing risk assessment practices. Importantly, this improvement was not attributed solely to increased identification of risk areas, but to improved efficiency and focus in reaching those outcomes. As noted by one participant, the framework did not necessarily uncover more risks but enabled the team to arrive at conclusions more efficiently/quickly, highlighting a key advantage over traditional FMEA approaches.

This finding aligns with well-documented challenges in the application of FMEA, particularly in early-stage system development, where the method can become time-intensive and difficult to scope effectively (Spreafico et al., 2017; Carlson, 2012; Stamatis, 2003). More recent work has further emphasized that traditional risk analysis methods often struggle to keep

pace with increasing system complexity and may not fully capture all relevant forms of risk, motivating the development of hybrid or complementary approaches (Ibarra et al., 2024; Shafiee et al., 2019). These efforts, while improving analytical coverage, can also increase methodological complexity and analysis burden, reinforcing the need for approaches that improve the scoping and prioritization of analysis rather than expanding its breadth.

Participants consistently emphasized the value of the framework in improving focus and facilitating cross-functional discussion. This aligns with systems engineering principles emphasizing interdisciplinary integration and communication in the identification and management of system risk (INCOSE, 2015). The NUD classification approach appears to provide a shared language for discussing design uncertainty, enabling more efficient convergence on areas requiring further analysis. This is particularly relevant given that many traditional methods, including FMEA, adopt a bottom-up perspective that may not fully capture system-level interactions or dynamic behavior in complex systems (Sulaman et al., 2019).

While responses related to process effectiveness and filtering confidence were uniformly strong, slightly greater variability was observed in responses related to downstream analysis, particularly in determining when FMEA was the appropriate tool. This suggests that, although the framework is effective in identifying and categorizing risk, additional guidance or experience may be required to consistently translate these classifications into specific analysis strategies. This observation is consistent with prior work indicating that no single risk analysis method is sufficient to identify all hazards and that multiple approaches are often required in practice (Sulaman et al., 2019).

Overall, the results suggest that the primary contribution of the NUD framework lies not in replacing FMEA, but in improving its application by providing a structured approach to

scoping and prioritization. In the context of increasingly complex systems and evolving risk analysis practices, there is a growing need for methods that support practical decision-making in addition to theoretical completeness (Nolan & McDermott, 2025). By integrating risk-informed decision-making earlier in the development process, the framework addresses a gap between high-level systems engineering practices and detailed failure analysis methods.

6.9. Conclusion

This chapter validated the NUD design risk filtering process as the first applied test of the improved FMEA framework developed in Research Aim 2. In a high-complexity industrial development context, the process was teachable, applicable by a cross-functional team, and perceived as useful relative to existing risk assessment practices. These findings support the use of NUD as a practical front-end filter for focusing FMEA effort on areas of greater design uncertainty.

These results address Research Aim 3 by showing that the NUD portion of the proposed framework can be deployed in a real development environment with limited training and facilitation. While additional validation is needed for the broader FMEA framework, this study provides initial evidence that targeted design risk filtering can help address the proliferation problem identified throughout this dissertation.

Chapter 7. Conclusions

7.1. Introduction

This dissertation examined persistent challenges in the application of FMEA within complex V-model development. Although FMEA remains one of the most widely used tools for design risk management, the literature and case examples reviewed in this dissertation show that many of the same problems have persisted for decades. These problems include proliferation of failure modes, difficulty applying FMEA early enough to influence design, inconsistent risk assessment, and challenges aligning FMEA with staged systems engineering decomposition.

The dissertation addressed these challenges through three research aims. First, it analyzed published criticisms of FMEA and identified systemic sources of recurring problems when FMEA is applied within V-model development. Second, it developed an improved framework intended to better align FMEA with the timing, decomposition, and information constraints of the V-model. Third, it validated the New, Unique, and Difficult (NUD) design risk filtering portion of the framework in a high-complexity industrial development project. This chapter summarizes the major contributions of the dissertation, reviews the findings associated with each research aim, discusses limitations and future work, and concludes with the dissemination of these findings.

7.2. Research Contributions

This dissertation has advanced body of knowledge regarding FMEA in three primary ways. First, through analysis of relevant FMEA standards and standard practices for V-model development, this dissertation demonstrates that structural incompatibilities exist between the two processes. These incompatibilities appear to have a causal relationship with several recurring

issues identified in peer reviewed FMEA research over the last 35 years. Specifically, the analysis shows that classic FMEA requires a level of completeness in failure mode identification that becomes increasingly difficult in modern systems characterized by high feature density, high engineering discipline complexity, and extensive design reuse. This contributes to a combinatorially unmanageable number of potential failure modes to evaluate in a typical design effort.

A second structural incompatibility is created by the difference in development logic between the two methods. V-model development operates primarily in a top-down manner, beginning with user needs and system requirements before decomposing into architecture and detailed design. Classic FMEA, by contrast, is often applied as a bottom-up analysis that depends on detailed knowledge of components, causes, and failure mechanisms. As a result, the point at which there is adequate information to perform a classic FMEA often occurs too late in the V-model for the FMEA to fully influence design decisions. This finding helps explain why many common areas of FMEA research, particularly those focused on risk scoring and prioritization, cannot by themselves address the systemic problems identified in this dissertation.

Second, this dissertation develops a two-part framework intended to address these structural problems. The first part of the framework is the New, Unique, and Difficult (NUD) design risk filter. NUD allows teams to identify areas of design risk before executing a full FMEA, enabling low-risk or well-proven design elements to be removed from detailed consideration. Importantly, this filtering can be performed using information available in the initial stages of the V-model, allowing risk assessment activity to begin earlier in the design planning process.

The second part of the framework is a modified approach for executing FMEA after NUD filtering has established the key design risk areas. This modified FMEA framework uses the information available at the system level and supports the decomposition of requirements and architecture as part of the analysis. As mitigations are established, they are reassessed using the NUD approach to determine whether the mitigation introduces new design risk. The framework also allows for the use of improved risk classification models, scoring approaches, and failure mode identification methods, making it compatible with current areas of FMEA research rather than replacing them.

Third, this dissertation provides initial validation of the NUD portion of the framework through application in the design of a complex laser system with a design team that was not involved in the development of the method. While the full framework has not yet been validated to the same extent, the NUD validation demonstrates that the design risk filtering process can be taught, deployed, and perceived as useful in a real development environment. This provides evidence that the proposed framework has practical value and establishes a foundation for future validation of the broader approach.

7.3. Summary of Research Findings

7.3.1. Research Aim 1 Findings: Identify Challenges of FMEA in V-Model Development

RQ 1.1: What recurring patterns of problems are evident in published criticisms of FMEA, particularly those that appear resistant to solutions proposed by standards, academia, or industry?

Answer: The recurring problems identified in the literature include proliferation of failure modes for analysis (the significant majority of which do not merit further analysis), inaccurate or

inconsistent cause-and-effect analysis, inconsistent risk scoring, and limited effectiveness in identifying high risk areas in design.

RQ 1.2: What underlying systemic causes contribute to recurring FMEA problems when the method is applied within modern development models such as the V-model?

Answer: Two primary systemic causes were identified. First, modern system complexity and predicate design reuse create a combinatorially challenging set of possible failure modes, many of which do not merit extensive additional analysis. Second, the information required for classic FMEA is often unavailable in the early, requirements- and architecture-focused stages of V-model development, leading to inaccurate assumptions and errors in judgment.

7.3.2. Research Aim 2 Findings: Develop Improved Framework for FMEA Application

RQ2.1: What high-level requirements define the solution space for improvements to FMEA that address the systemic causes and constraints identified in Research Aim 1?

Answer: The high-level requirements developed for the improved framework are to:

- Create an architectural-level filtering process for evaluating risks associated with an intended new design, to determine where FMEA provides the greatest benefit. This addresses proliferation by focusing FMEA on new or higher-risk design elements rather than stable design reuse with strong provenance.
- Modify FMEA so that its information requirements and outputs can vary depending on where it is applied within the V-model. This addresses the incompatibility between classic FMEA and the staged availability of information in V-model development.

- Create a structure for FMEA to iterate and decompose in alignment with the V-model rather than in conflict with it. This addresses the mismatch between top-down V-model decomposition and bottom-up FMEA execution.
- Take advantage of the right side of the V-model to provide traceable evidence of mitigation effectiveness. This addresses the need, particularly in regulated industries, to show clear linkage between identified failure modes, the mitigations intended to address them, and the verification or validation evidence demonstrating that those mitigations are effective.
- Create the revised model as a framework for applying FMEA under the V-model without preventing the use of other FMEA improvement methods. This ensures that improvements in scoring, risk categorization, failure mode elicitation, artificial intelligence, and other execution-level methods can still be incorporated within the broader framework.

RQ2.2: How can FMEA be augmented to better align with the V-model based on the requirements developed in Research Aim 2?

Answer: FMEA can be augmented through a two-part framework. The first part is a structured design risk filter, the New, Unique, and Difficult (NUD) process, which identifies areas of the design that warrant focused analysis. The second part is a modified, top-down FMEA framework that uses requirements and architecture information available early in the V-model, supports decomposition of requirements and architecture, and reassesses mitigations through the NUD filter to determine whether they introduce new design risk.

7.3.3. Research Aim 3 Findings: Validate Proposed Framework for Improving FMEA

RQ3.1: How difficult is it to teach the proposed [NUD] methodology to a team that is entirely unfamiliar with it?

Answer: The validation study indicated that NUD could be taught with limited training and facilitation. A one-hour training session with relevant examples was sufficient for the team to begin applying the NUD process independently, without continued facilitation by the author. Follow-up discussion was needed for more complex topics, including environmental issues, supplier-based issues, latent defects, and reliability concerns. However, these are complex issues within FMEA generally, and the follow-up discussions totaled less than eight hours over several weeks of work.

RQ3.2: To what extent do users perceive value in the proposed methodology and express willingness to use it in future projects?

Answer: Participants perceived value in the NUD process and indicated willingness to use it again. Survey responses showed that the method improved efficiency in identifying design risks and helped create a clearer connection between requirements and architecture.

RQ3.3: What level of facilitation and contextually relevant examples are needed to support effective deployment of the proposed methodology?

Answer: The validation study suggests that contextually relevant examples and initial facilitation are important for effective deployment. The author had extensive knowledge of gas laser design and was therefore able to provide examples directly relevant to the team when teaching the NUD definitions and application. This also supported facilitation by helping guide discussion around challenge points encountered by the team. Participant feedback indicated that

these examples were valuable for understanding the method and helped the team continue applying it without the author's assistance. Therefore, it is difficult to determine what the outcome would have been without such examples. However, based on other uses of the method with the author's former and current employers, the likely result is that teams can still learn and apply the concepts, but more time may be required.

7.4. Limitations and Future Work

7.4.1. Limitations

The research presented in this dissertation supports the conclusion that structural challenges exist between classic FMEA and development under the V-model or similar staged development processes. The analysis also suggests that many common areas of FMEA research, particularly failure mode identification and risk classification, do not fully address these structural concerns, even though they have value. Therefore, the development of a framework that can operate top down and perform risk filtering without full articulation of all possible failure modes contributes to the body of knowledge.

However, several limitations remain. First, the sample size for validation testing is small. Although the NUD process has shown value in different environments, the formal validation was limited to a single case study with one organization and one development project. The consistency of participant responses is encouraging, but broader validation across additional organizations, industries, and project types is needed to assess robustness, repeatability, and generalizability.

Second, all cases to date, including the validation case, have been taught, deployed, and facilitated by the author. Therefore, independent validation has not yet occurred. A stronger

validation would involve practitioners unconnected to the author applying the framework and attempting to replicate the results in other settings.

Third, the validation in this dissertation tested the NUD design risk filtering portion of the framework rather than the full improved FMEA framework. This is appropriate as an initial validation because NUD directly addresses the proliferation problem and can be tested using early-stage requirements and architecture information. However, the broader framework, including requirements-based FMEA execution, recursive decomposition, mitigation reassessment, and CtQ tracking, still requires formal validation.

Fourth, the research and testing were performed in product design contexts. FMEA is used much more broadly, including in processes, systems of systems, services, and other applications. Theoretically, there is no clear reason why the framework could not extend to these contexts, but this has not yet been demonstrated through research or testing.

Finally, the validation was not longitudinal. The study assessed participant perceptions and application of the NUD process during a development effort, but did not track how NUD classifications evolved over the full lifecycle of the project or how the resulting risk filtering affected downstream FMEA execution, mitigation decisions, verification activities, or product outcomes.

7.4.2. Future Work

A basic area of future work is to perform additional case studies to further validate the strengths and weaknesses of the framework. These studies should include additional organizations, industries, project types, and facilitators. In particular, future validation should test

whether practitioners who were not involved in developing the framework can learn, deploy, and reproduce similar outcomes.

Future work should also include longitudinal application of the framework within ongoing development efforts. This would allow researchers to evaluate how NUD classifications evolve as system designs mature, how risk is reduced or transferred through mitigation decisions, and how the framework supports downstream FMEA execution. Integration with downstream FMEA is especially important, including the use of NUD classifications to guide mitigation strategies and assess whether design changes introduce new risks.

Another important area of future work is the development of quantitative metrics for design risk based on NUD classification. For example, the distribution of New, Unique, and Difficult elements relative to overall system scope may provide a useful indicator of development risk from a systems engineering perspective. These metrics could also support evaluation of risk distribution by architectural element, performance requirement, risk type, or stage of development. Over time, this could support assessment of risk burn down as a project progresses.

There are also significant opportunities to integrate the framework with Model-Based Systems Engineering (MBSE). Since the framework is built from systems engineering principles of requirements allocation, architecture, and decomposition, integration into MBSE environments represents a natural extension. Future research could examine how SysML or related modeling approaches might be extended to represent NUD classifications, FMEA scope decisions, mitigations, evidence of effectiveness, and risk propagation across architecture, consistent with prior work demonstrating the use of MBSE to incorporate human factors

requirements into early requirements analysis and architecture definition within the V-model (Corl & Gallegos, 2024).

Similarly, the framework may support integration with requirements management tools. Unlike some areas of systems engineering, where requirements management tools have become common and have relatively standard expected functionality, FMEA tooling remains less standardized. Since the proposed framework aligns FMEA with requirements and architecture, there is a feasible opportunity to support traceability from failure mode, to mitigation, to verification or validation evidence within existing requirements management environments.

Finally, the framework currently does not take advantage of artificial intelligence tools. However, current research into AI-supported elicitation of failure modes, design analysis for causes, and related FMEA activities suggests a rich area for future work. AI tools may be particularly useful within the proposed framework because the framework provides structure around when and where such tools should be applied. Rather than using AI to expand the number of possible failure modes indiscriminately, future work could explore how AI might support focused analysis within NUD-flagged areas or assist in identifying whether mitigations introduce additional New, Unique, or Difficult design risks.

7.5. Publications of Results

The findings from this dissertation were split into three peer-reviewed products. Two of which were presented at and published in proceedings for conferences associated with the International Council on Systems Engineering (INCOSE). The third product has been submitted and is under review with a journal focused on engineering reliability. These products are as follows:

1. Perreault, D. C., Gallegos, E. E., Paglioni, V., & Bradley, T. H. (2024). Usability challenges of failure mode and effects analysis (FMEA) within the V-model.
Proceedings of the International Council on Systems Engineering (INCOSE) Human Systems Integration (HSI) Conference. Jeju, Korea: Aug 2024.
 - a. This paper focused on Research Aim 1.
2. Perreault, D. C., & Gallegos, E. E. (2025). Analysis of systemic failures in application of failure modes and effects analysis in V-model based complex design.
Proceedings of the INCOSE Western States Regional Conference. Bothell, WA: Sept 2025.
 - a. This paper focused on Research Aim 2.
3. Perreault, D. C., & Gallegos, E. E. (submitted April 2026). *Design risk filtering in complex V-model development to support FMEA*. Manuscript submitted for publication.
 - a. This paper focused on Research Aim 3.

REFERENCES

- Agarwala, A. S. (1990). Shortcomings in MIL-STD-1629A guidelines for criticality analysis. *Proceedings of the Reliability and Maintainability Symposium*, 494-496. Los Angeles, CA: Jan 1990. <https://doi.org/10.1109/ARMS.1990.68007>
- Ahmed, H., & Miller, E. E. (2022). Human-systems integration of agricultural machinery in developing economy countries: Perceptions of adoption. *Proceedings of the INCOSE Human Systems Integration Conference*, 18-28, San Diego, CA: Nov 2021. <http://doi.org/10.1002/iis2.12868>
- Ahmed, H., & Miller, E. E. (2023). Quantifying the economic impact on farmers from agricultural machinery: A case study of farmers in Sudan. *World*, 4(2), 347-359. <https://doi.org/10.3390/world4020022>
- Alruqi, Mansoor, Baumer, Martin, Branson, David T., Girma, Sourafel (2022), The Challenge of Deploying Failure Modes and Effects Analysis in Complex System Applications – Quantification and Analysis. *Sustainability*, 14(1397), 1-25. <https://doi.org/10.3390/su14031397>
- Anjalee, J. A., Rutter, V., & Smaranayake, N. R. (2021). Applications of failure mode and effect analysis (FMEA) to improve medication safety: A systemic review. *Postgrad Medical Journal*, 97(1145), 168-174. <https://doi.org/10.1136/postgradmedj-2019-137484>
- Automotive Industry Action Group (AIAG) & Verband der Automobilindustrie (VDA). (2019). *AIAG & VDA FMEA handbook: Failure mode and effects analysis* (1st ed.). Automotive Industry Action Group.

- Ayyildiz, E., Syhan, M., Kara, B., & Seyis, G. I. (2025). Investigation of risk factors in university Laboratories: An improved fuzzy based FMEA approach. *Safety Science*, 191, 1-13. <https://doi.org/10.1016/j.ssci.2025.106933>
- Beasley, R., Woodcock, H., & Eastwood, P. (2022). Mindful maturation matters: The V-model as a passport to engagement. *INCOSE International Symposium*, 32(1), 277-292. <https://doi.org/10.1002/iis2.12931>
- Bluvband, Z., & Grabov, P. (2009). Failure Analysis of FMEA. *Proceedings of the Annual Reliability and Maintainability Symposium*. Fort Worth, TX: Jan 2009. <https://doi.org/10.1109/RAMS.2009.4914700>
- Botyarov, M., & Gallegos, E. E. (2025). Generative design solution space parsing: An evaluation of user experience, workload, and performance. *Journal of Systems Science and Systems Engineering*, 34, 129-155. <https://doi.org/10.1007/s11518-024-5626-8>
- Bouti, A., & Kadi, D. A. (1994). A state of the art review of FMEA/FMECA. *International Journal of Reliability, Quality and Safety Engineering*, 1(4), 515-543. <https://doi.org/10.1142/S0218539394000362>
- Carlson, C. S. (2012). *Effective FMEAs: Achieving Safe, Reliable, and Economical Products and Processes Using Failure Mode and Effects Analysis*. Wiley. <https://doi.org/10.1002/9781118312575>
- Ceylan, B. O. (2024). Marine diesel engine turbocharger fouling phenomenon risk assessment application by using fuzzy FMEA method. *Journal of Engineering for the Maritime Environment*, 238(3), 514-530. <https://doi-org/10.1177/14750902231208848>

- Chellin, M. D., & Gallegos, E. E. (2024). Visualization tool for obsolescence management: Reducing supply chain risk for C5ISR systems. *Defense Acquisition Research Journal*, 31(1), 36-61. <https://doi.org/10.22594/dau.23-914.31.01>
- Chellin, M. D., & Miller, E. E. (2023a). Frustrated with obsolescence – Try changing your mental model. *Defense Acquisition Research Journal*, 30(3), 228-249. <https://doi.org/10.22594/dau.23-903.30.03>
- Chellin, M. D., & Miller, E. E. (2023b). Proactive obsolescence management methods for C5ISR systems: Insights from practitioners. *Defense Acquisition Research Journal*, 30(1), 24-44. <https://doi.org/10.22594/dau.21-886.30.01>
- Corl, K., & Gallegos, E. E. (2024). A method for human systems integration requirements within model based systems engineering. *Proceedings of the International Council on Systems Engineering (INCOSE) International Symposium*, 34(1), 1787-1806. Dublin, Ireland: July 2024. <https://doi.org/10.1002/iis2.13237>
- Coronado, P. J., & Gallegos, E. E. (2025). Defining medical device ecosystems: A systematic literature review. *IEEE Access*, 13, 173065-173080. <https://doi.org/10.1109/ACCESS.2025.3615110>
- En-Naaoui, A., Kaicer, M., & Aguezzoul, A. (2024). Predicting risk criticalities of hospital sterilization using fuzzy FMEA and artificial neural network. *Proceedings of the International Conference on Intelligent Systems and Computer Vision (ISCV)*. Fez, Morocco: May 2024. <https://doi-org/10.1109/ISCV60512.2024.10620085>

Forsberg, K., & Mooz, H. (1992). The Relationship of systems engineering to the project cycle. *Engineering Management Journal*, 4(3), 36-43.

<https://doi.org/10.1080/10429247.1992.11414684>

Forsberg, K., Mooz, H., & Cotterman, H. (2005). *Visualizing project management: Models and frameworks for mastering complex systems* (3rd ed.). John Wiley & Sons.

Gaikwad, A., Antara, F., Gangu, K., Agarwal, R., Jain, S., Vashishtha, S. (2023). Innovative Approaches to Failure Root Cause Analysis Using AI-Based Techniques. *International Journal of Progressive Research In Engineering Management and Science*, 3(12), 561-592. <https://www.doi.org/10.58257/IJPREMS32377>

Goswami, M. (2018). An enterprise centric analytical risk assessment framework for new product development. *Cogent Business and Management*, 5(1).

<https://doi.org/10.1080/23311975.2018.1540255>

Gracias, M. H., & Gallegos, E. E. (2024). Transitioning Perspectives: Agile and Waterfall Perceptions in the Integration of Model-Based Systems Engineering (MBSE) within Aerospace and Defense Industries. *ITEA Journal of Test and Evaluation*, 45(4).

<https://doi.org/10.61278/itea.45.4.1006>

Gracias, M. H., & Gallegos, E. E. (2025). Balancing structure and flexibility: evaluating agile, waterfall, and hybrid methodologies in aerospace and defense projects. *ITEA Journal of Test and Evaluation*, 46(3). <https://doi.org/10.61278/itea.46.3.1003>

Guimaraes, A. C., & Lapa, C. M. (2004). Fuzzy FMEA applied to PWR chemical and volume control system. *Progress in Nuclear Energy*, 44(3), 191-213.

[https://doi.org/10.1016/S0149-1970\(04\)90010-8](https://doi.org/10.1016/S0149-1970(04)90010-8)

- Hassani, I. E., Masrour, T., Kourouma, N., & Tavcar, J. (2025). AI-driven FMEA: integration of large language models for faster and more accurate risk analysis. *Design Science*, 11, e10. <https://doi.org/10.1017/dsj.2025.7>
- Horsky, J., McColgan, K., Pang, J. E., Melnikas, A. J., Linder, J. A., Schnipper, J. L., & Middleton, B. (2010). Complementary methods of system usability evaluation: Surveys and observations during software design and development cycles. *Journal of Biomedical Informatics*, 43(5), 782-790. <https://doi.org/10.1016/j.jbi.2010.05.010>
- Ibarra, L. A. C., Carvalho, E. F. U., & Durazzo, M. (2024). Overcoming FMEA shortcomings: Proposal of an adapted risk analysis matrix. *Engineering Failure Analysis*, 166, 108796. <https://doi.org/10.1016/j.engfailanal.2024.108796>
- International Council on Systems Engineering (INCOSE). (2015). *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities* (4th ed.). Wiley.
- International Automotive Task Force (IATF). (2016). *IATF 16949:2016: Quality management system requirements for automotive production and relevant service parts organizations*.
- International Electrotechnical Commission (IEC). (2018). *Failure modes and effects analysis (FMEA and FMECA)* (IEC 60812:2018).
- International Organization for Standardization (ISO). (2019). *Medical devices - Application of risk management to medical devices* (ISO 14971:2019). <https://www.iso.org/standard/72704.html>
- International Organization for Standardization (ISO). (2017). *Plain bearings - Quality assurance of thin-walled half bearings - Design FMEA* (ISO 12132:2017). <https://www.iso.org/standard/68623.html>

- Juristo, N., Moreno, A. M., & Sanchez-Segura, M.-I. (2007). Analysing the impact of usability on software design. *Journal of Systems and Software*, 80(9), 1506-1516. <https://doi.org/10.1016/j.jss.2007.01.006>
- Kadena, E., Kocek, S., Takacs-Gyorgy, K., & Keszthelyi, A. (2022). FMEA in smartphones: A fuzzy approach. *Mathematics*, 10(3), 513. <https://doi.org/10.3390/math10030513>
- Katzir, L., Yoel, S. R., & Dori, D. (2025). A unified requirements-architecture system model-development and empirical evaluation. *Systems Engineering*, e70035. <https://doi.org/10.1002/sys.70035>
- Kirke, M. S., Santosh, R. B., & Jadhav, J. R. (2015). Risk management in medical product development process using traditional FMEA and fuzzy linguistic approach: A case study. *Journal of Industrial Engineering International*, 11, 595-611. <https://doi.org/10.1007/s40092-015-0113-y>
- Li, Y., & Zhu, L. (2020). Risk analysis of human error in interaction design by using a hybrid approach based on FMEA, SHERPA and fuzzy TOPSIS. *Quality and Reliability Engineering International*, 36(5), 1657-1677. <https://doi.org/10.1002/qre.2652>
- Liu, H.-C., Liu, L., & Liu, N. (2013). Risk evaluation approaches in failure mode and effects analysis: A literature review. *Expert Systems with Applications*, 40(2), 828-838. <https://doi.org/10.1016/j.eswa.2012.08.010>
- Liu, H.-C., Li, P., You, J.-X., & Chen, Y.-Z. (2014). A novel approach for FMEA: Combination of interval2-tuple linguistic variables and gray relational analysis. *Quality and Reliability Engineering International*, 31(5), 761-772. <https://doi.org/10.1002/qre.1633>

- Lo, H.-W., & Liou, J. J. H. (2018). A novel multiple-criteria decision-making-based FMEA model for risk assessment. *Applied Soft Computing Journal*, 73, 684-696. <https://doi.org/10.1016/j.asoc.2018.09.020>
- McKinney, B. T. (1991). FMECA, The Right Way. *Proceedings of the Annual Reliability and Maintainability Symposium*, 253 - 259. Orlando, FL: Jan 1991.
- Mousavi, S., Hafezalhotob, A., Ghezavati, V., & Abdi, F. (2022). A new fuzzy multi-criteria decision-making approach for risk assessment of competitor's cooperation in new product development projects. *Journal of Business and Industrial Marketing*, 37(11), 2278-2297. <https://doi.org/10.1108/JBIM-06-2021-0298>
- National Aeronautics and Space Administration (NASA). (2024). *Guideline for Failure Modes and Effects Analysis and Risk Assessment (GSFC-HDBK-8004)*.
- Nolan, N., & McDermott, O. (2025). Failure mode effect analysis use and limitations in medical device risk management. *Journal of Open Innovation: Technology, Market and Complexity*, 11(1), 100439. <https://doi.org/10.1016/j.joitmc.2024.100439>
- Ortolano, L. F., & Gallegos, E. E. (2025). Advancing human-AI collaboration in small and medium-sized enterprises: A systems engineering approach. *Systems Engineering*, 29(3), 477-494. <https://doi.org/10.1002/sys.70031>
- Ouyang, L., Yan, L., Han, M., & Gu, X. (2022). Survey of FMEA methods with improvement on performance inconsistency. *Quality and Reliability Engineering International*, 38(4), 1850-1868. <https://doi.org/10.1002/qre.3059>

- Ozarin, N. W. (2013). Bridging software and hardware FMEA in complex systems. *Proceedings of the Annual Reliability and Maintainability Symposium*. Orlando, FL: Jan 2013.
<https://doi-org/10.1109/RAMS.2013.6517739>
- Panyukov, D. I., Kozlovskii, V. N., Aidarov, D. V., & Shakurskii, M. V. (2022). Effectiveness of FMEA Risk Analysis. *Russian Engineering Research*, 42, 1070-1072.
<https://doi.org/10.3103/S1068798X22100203>
- Perreault, D. C., & Gallegos, E. (2025). Analysis of systemic failures in application failure modes and effects analysis in V-model based complex design. *Proceedings of the INCOSE Western States Regional Conference*. Bothell, WA: Sept 2025.
- Perreault, D., & Gallegos, E. E. (2026). *Design risk filtering in complex V-model development to support FMEA*. Unpublished Manuscript.
- Perreault, D. C., Gallegos, E. E., Paglioni, V., & Bradley, T. H. (2024). Usability challenges of failure mode and effects (FMEA) within the V-model. *Proceedings of the INCOSE Human Systems Integration (HSI) Conference*. Jeju, Korea: Aug 2024.
- Shafiee, M., Enjema, E., & Kolios, A. (2019). An integrated FTA-FMEA model for risk analysis of engineering systems: A case study of subsea blowout preventers. *Applied Science*, 9(6), 1192. <https://doi.org/10.3390/app9061192>
- Silverman, M., & Johnson, J. R. (2013). FMEA on FMEA. *Proceedings of the Annual Reliability and Maintainability Symposium*. Orlando, FL: Jan 2013. <https://doi-org/10.1109/RAMS.2013.6517629>
- Singer, H., & Ozcelik, T. O. (2024). Classifying the properties of stainless steel materials for biomedical applications under an intuitionistic fuzzy environment: An FMEA-based

- TOPSIS-sort methodology. *Materials Today Communications*, 40, 110183. <https://doi.org/10.1016/j.mtcomm.2024.110183>
- Society of Automotive Engineers International (SAE). (2021). *Potential Failure Mode and Effectis Analysis (FMEA), Including Design FMEA, Supplemental FMEA-MSR, and Process FMEA (J1739_202101)*. https://doi.org/10.4271/J1739_202101
- Spreafico, C., Russo, D., & Rizzi, C. (2017). A state-of-the-art review of FMEA/DMECA including patents. *Computer Science Review*, 25, 19-28. <https://doi.org/10.1016/j.cosrev.2017.05.002>
- Stematis, D. H. (2003). *Failure Mode Effect Analysis: FMEA from Theory to Execution (2nd ed.)*. ASQ Quality Press.
- Sulaman, S. M., Beer, A., Felderer, M., & Host, M. (2019). Comparison of the FMEA and STPA safety analysis methods – a case study. *Software Quality Journal*, 27, 349-387. <https://doi.org/10.1007/s11219-017-9396-0>
- Rook, P. (1986). Controlling software projects. *Software Engineering Journal*, 1(1), 7-16. <https://doi.org/10.1049/sej.1986.0003>
- U.S. Department of Defense (U.S. DOD). (1949). *Procedure for Performing a Failure Mode, Effect and Criticality Analysis (MIL-P-1629)*.
- U.S. Department of Defense (U.S. DOD). (1980). *Military Standard: Procedures for Performing a Failure Mode, Effects, and Criticality Analysis (MIL-STD-1629A)*.
- U.S. Department of Defense (U.S. DOD). (1985). *Defense System Software Development (DOD-STD-2167)*.

- U.S. Department of Defense (U.S. DOD). (1988). *Military standard: Defense system software development* (DOD-STD-2167A).
- U.S. Department of Defense (U.S. DOD). (1991). *Defense acquisition management documentation and reports* (DoD Manual 5000.2-M).
- U.S. Food and Drug Administration (U.S. FDA). (2026). *Quality management system regulation, 21 C.F.R. pt. 820*. Electronic Code of Federal Regulations.
<https://www.ecfr.gov/current/title-21/chapter-I/subchapter-H/part-820>
- U.S. Food and Drug Administration (U.S. FDA). (2023). *Q9 (R1) Quality Risk Management*.
<https://www.fda.gov/regulatory-information/search-fda-guidance-documents/q9r1-quality-risk-management>
- Xia, Y., Jazdi, N., & Weyrich, M. (2024). Enhance FMEA with large language models for assisted risk management in technical processes and products. *Proceedings of the IEEE 29th International Conference on Emerging Technologies and Factory Automation*. Padova, Italy: Sept 2024. <https://doi.org/10.1109/ETFA61755.2024.10710996>

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
AIAG	Automotive Industry Action Group
ASIC	Application Specific Integrated Circuit
CtQ	Critical-to-Quality
D-FMEA	Design Failure Modes and Effects Analysis
DOD	Department of Defense
FMECA	Failure Modes, Effects, and Criticality Analysis
FMEA	Failure Modes and Effects Analysis
GPU	Graphics Processing Unit
HSI	Human Systems Integration
IATF	International Automotive Task Force
IEC	International Electrotechnical Commission
INCOSE	International Council on Systems Engineering
ISO	International Organization for Standardization
LAN	Local Area Network
LLM	Large Language Model
MBSE	Model-Based Systems Engineering
NUD	New, Unique, and Difficult

P-FMEA	Process Failure Modes and Effects Analysis
RPN	Risk Priority Number
SaaS	Software as a Service
U-FMEA	Usability Failure Modes and Effects Analysis
VDA	Verband der Automobilindustrie