INCOSE

WESTERN STATES REGIONAL CONFERENCE

Analysis of Systemic Failures in Application of Failure Modes and Effects Analysis in V-Model Based Complex Design

INCOSE WSRC · 2025

AUTHORS
Dan Perreault, and Erika E. Gallegos

PUBLICATION DATE
July, 2026

SUBMISSION #
22

INCOSE PUBLICATIONS

Analysis of Systemic Failures in Application of Failure Modes and Effects Analysis in V-Model Based Complex Design

Dan Perreault
Colorado State University
6029 Campus Delivery
Fort Collins, CO 80523
Dan.Perreault@colostate.edu

Erika E Gallegos
Colorado State University
6029 Campus Delivery
Fort Collins, CO 80523
Erika3@colostate.edu

Abstract

Originating more than 70 years ago within the US Department of Defense, Failure Modes and Effects Analysis (FMEA) has become one of the most ubiquitous design risk assessment tools in use today. For many, FMEA has outstripped fault tree and other forms of design failure analysis as the primary tool. FMEA has become a standard requirement of military contracts, automotive design processes, medical device development, and commercial product development. The methodology has broadened from discrete hardware module level design analysis to software analysis, full mixed system level analysis, system of systems, usability, service, workflows, process analysis - essentially any type of product or process for which a design risk assessment is desired. Despite its widespread use, the core methodology of FMEA has remained largely unchanged over the past seven decades, and for the last 30 years, research has increasingly documented problems with its applicability and effectiveness in various ways. Recent research suggests that these problems are systemic to the approach in modern complex product development. This paper explores four case studies using anonymized FMEAs from a leading medical device manufacturer, illustrating several of the known FMEA challenges in real-world practice. This paper also introduces a novel framework for improving the effectiveness and applicability of FMEA in design environments, aligning with the systems engineering V-model.

Keywords

FMEA, Risk analysis, Requirements, Criticality, Risk prioritization.

Introduction

Failure Modes Effects Analysis (FMEA), represents the foundation of a family of related methods (PFMEA, FMECA, UFMEA, etc.), that originated as an approach to achieve three core objectives:

- Predict design failures before committing to designs that are costly to change.
- Prioritize these failures based on a semi-heuristic model to ensure that the most severe failures receive attention in the design process.

- Provide a means of determining whether proposed solutions adequately address the identified failures.

Over the past 30 years, however, peer reviewed research has consistently highlighted that all three of these aims have become increasingly challenging to achieve. Much of the scholarly and industry-focused effort has centered either on the administrative aspects of FMEA or on attempts to better systematize the scoring model for prioritization. Despite these efforts, there is little evidence of any substantive improvement in the practical application of FMEA. As the case studies presented in this paper demonstrate, FMEA in industry has increasingly devolved into a digital "paperwork" exercise, offering limited practical value. With system designs now routinely involving complex interactions across multiple engineering disciplines, there is an urgent need for more effective and value-added approaches to design risk analysis.

Background

Failure Mode and Effects Analysis

The core FMEA methodology, with the three defined goals stated previously, was introduced by the US military 76 years ago (Department of Defense, 1949). Once the administrative groundwork is complete (e.g., forming a team, securing funding, etc.), the classic methodology is deceptively simple, can be summarized by the following steps:

- Select a system or subsystem for analysis.
- Determine the functionality of the system.
- Identify ways in which the system might fail to function.
- Determine the impact of each failure on the user or dependent system, recognizing that impacts may differ based on the nature or severity of the failure.
- Analyze potential causes within the system that could lead to each failure, noting that multiple causes may exist.
- Apply a scoring rubric, typically involving three orthogonal factors: severity (based on impact), likelihood (based on the probability of the cause), and detectability (based on the assumption that users can avoid impact if aware). These factors are combined into a single score for each failure.
- Sort failures based on their scores to establish a priority order. Scoring rubrics generally define the corresponding levels required action. Some failures may require no action, while others (e.g., safety related failures) may require extensive analysis to justify the score.
- Implement one or more design improvements to mitigate failure causes, typically reducing the probability of failures. Reducing severity is generally more difficult, and improving detectability is often less effective; however, any of the three factors may be addressed depending on the context.
- Rescore the mitigated design to determine the level of improvement. In premise, mitigation continues until the score falls below a predetermined action threshold, and multiple mitigations may be required for a given failure cause.
- Document the analysis, typically in tabular form, although other approaches exist.

Once all intended mitigations have been incorporated into the design and are proven to be effective through appropriate testing or verification activity, the final FMEA update can be completed and the process closed. Nearly all standards and guidelines recommend maintaining this information as part of sustaining data, so practitioners are not required to start from scratch with each new project.

Since its origin, and with the growth in FMEA applications, dozens of standards (International Electrotechnical Commission, 2018), texts (Stematis, 2003), and handbooks (Automotive Industry Action Group, 2019) have been published to formalize and refine the method. However, the fundamental logic, process, and execution of FMEA have not changed. In many respects, modern texts are substantively similar to the original work from 1949. Refinements in the literature have generally focused on three areas:

Organizational commitment to FMEA. Most standards and guidance emphasize the need for cross functional teams, early initiation of FMEA in the design process, and maintaining the analysis as a living document throughout design maturation.

Development of alternative risk-scoring algorithms. Spreafico et al. (2017) identify extensive research into alternative scoring models. The VDA augmented the traditional Risk Priority Number (RPN) with hierarchical tie-breaking rules (Automotive Industry Action Group, 2019). Some approaches extend further, incorporating linguistic or fuzzy models to improve scoring resolution (Liu et al., 2014).

Expansion to new application domains. FMEA has been applied to domains ranging from nuclear reactor reliability (Guimaraes & Lapa, 2004) to medication safety (Anjalee et al., 2021). Anecdotally, discussions within the academic community increasingly point to the use of artificial intelligence in FMEA, suggesting further expansion of the method's application space.

Despite the growing recognition of fundamental challenges in the effective use of FMEA, relatively little research has explored systemic solutions. Among the more recent efforts is work addressing the complexity of integrated software and hardware systems (Ozarin, 2013), now over a decade old. This lack of systemic research was also noted by Spreafico et al. (2017) in their literature review.

V-Model for Development

The systems engineering V-model for development has a history similar to that of FMEA. It originated in the mid 1980s within the US Department of Defense as new development increasingly required the integration of software and hardware (Department of Defense, 1985). As a methodology, the software development aspect migrated to the commercial sector soon after (Rook, 1986). The fundamentally intuitive aspects of the method led to its broad adoption across many types of system development.

As a summary, the V-model involves staged decomposition of design on the left-hand side, paired with planning for testing, verification, and validation on the right-hand side, as shown in Figure 1. Numerous variants of the model exist in the literature. Figure 1 provides a generic variant adapted from the original Mil-Std 2167 (Department of Defense, 1985). In this illustration, the blocks represent the stages of the V-model with the maturation steps occurring at each. This will become relevant when discussing the systemic challenges in application of FMEA.

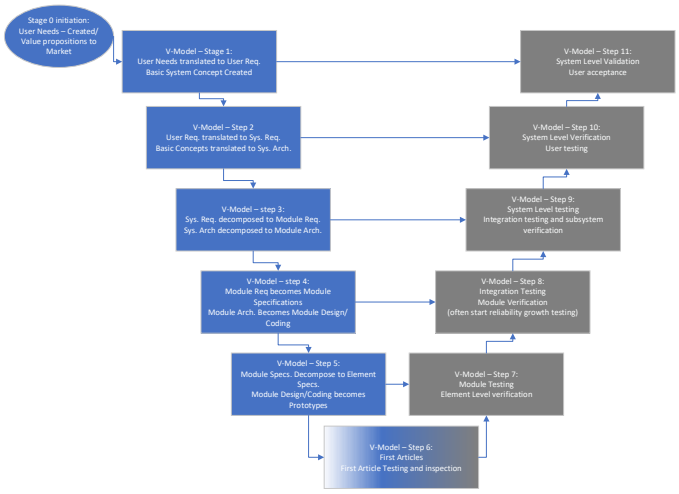

Figure 1. Description of the Systems V-Model

Using FMEA within V-Model

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

In many organizations, the development process expects the FMEA process to start as early as Stage 1 or Stage 2 in Figure 1. Perreault et al. (2024) compared FMEA standards with the V-model and identified systemic inconsistencies between the level of design maturity required to perform FMEA and the stages of the V-model at which FMEA is expected.

It is also well established, particularly in regulated industries such as medical devices, that both a formal, structured development model, such as the V-model (Food and Drug Administration, 2025), and a formal, structured design risk methodology, such as FMEA, are required (Nolan & McDermott, 2025). In retrospect, the V-model represents a top-down development approach, decomposing system level requirements into manageable subsystems, whereas classic FMEA is often described as a bottom-up risk management methodology (Kirkire et al., 2015). From this perspective, the results identified by Perreault et al. (2024) are not surprising.

Figure 2 provides a visual comparison between standards based FMEA information requirements (International Electrotechnical Commission, 2018) and the stages of the V-model shown in Figure 1.

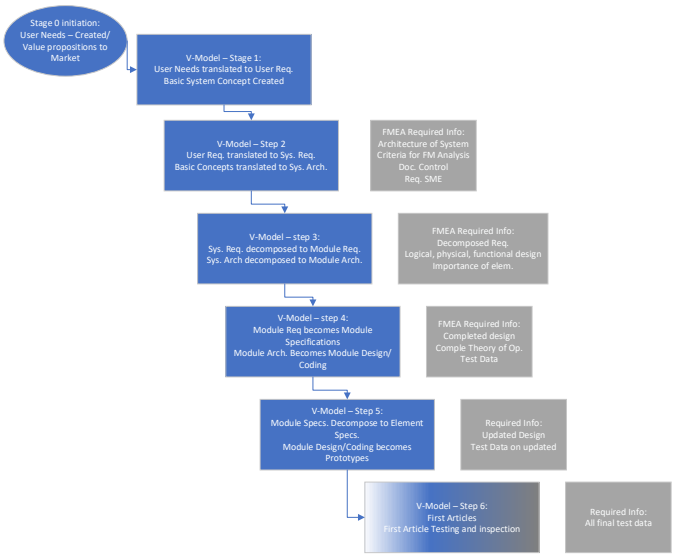

Figure 2. Comparison of V-Model Stages to FMEA Information Requirements

This figure illustrates how 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 information does not yet exist. While FMEA standards and text tend to be thorough on defining requirements and deliverables, they are comparatively sparse in guidance on methodology.

To further examine this hypothesis, this analysis incorporates redacted FMEAs from a regulated organization that employs a V-model based product development process. In this organization, system level FMEAs are required early in the V-model, as illustrated in Figure 2. Selected example entries from these FMEAs are reviewed in the following sections. These case studies are not intended to be exhaustive, but rather representative of how the challenges identified in the literature manifest in practice.

Case Study 1: Inadequate Risk Assessment

In this example (see Table 1), the image compensation option (not the marketed name) had not yet been developed, although pseudo-code existed for the intended algorithm. It was recognized that previous hardware platforms may not have adequate processing power. However, the identified cause of failure (i.e., inefficient coding of algorithm) provided little value in guiding mitigation, and system integration testing focused on what were perceived as "extreme" imaging cases proved ineffective. As a result, the risk was judged as acceptable based on the RPN score, a determination that later proved problematic and required correction in a sustaining release.

Item/Function	Motion compensation and persistence
Requirement	The Image Compensation option provides the ability for the user to enable digital image stabilization
Potential Failure Mode	CPU cannot keep up with algorithm processing demands
Potential Effects of Failure	System crashes
Severity	8
Cause of Failure	Inefficient coding of algorithm

Current Prevention	None
Occurrence	5
Current Detection	System Integration testing of extreme imaging cases
Detection	1
RPN	40

Table 1. Row from FMEA for Case Study 1

Case Study 2: Poor Failure Definition

In this example (see Table 2), the FMEA row is poorly constructed, as it merely documents the intended use of the temperature monitoring system rather than a potential failure. At this stage of the design process, there was no clarity regarding whether the condition would be met by the system or what a failure of the temperature monitoring function would represent. Additionally, system architecture decomposition had not yet been completed, leading to confusion within the FMEA team about how to evaluate the entry.

Item/Function	System Temperature Monitoring
Requirement	The System provides a warning to the user that the system is in an overtemperature state and will shut down within a specific period of time
Potential Failure Mode	System Shuts down
Potential Effects of Failure	System not available
Severity	8
Cause of Failure	Heat buildup due to extended system use at high GPU utilization
Current Prevention	Temperature monitoring.
Occurrence	5
Current Detection	Design Qualification Testing
Detection	5
RPN	200

Table 2. Row from FMEA for Case Study 2

Case Study 3: Low Value FMEA Activity

In this example, the RPN score of 72 was low enough according to the scoring rubric to not require additional mitigation (the rubric is defined within the organization's FMEA procedure). During interviews, the FMEA team described this case as a "paperwork exercise," as the user interface was a design reuse and the AppUI library was well proven on the platform. The only difference from prior implementations was the size of images the optimization controls were being applied to (see Table 3). As a result, it was well understood before ever populating the row that no further action would be required, given the standardization of the UI image optimization controls from predicate products.

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

Table 3. Row from FMEA for Case Study 3

Case Study 4: Complexity and Novelty

This 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 (see Table 4) the algorithm in question 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 of curated training materials for the AI. This led to delays in this project because of the lack of ability to predict when the algorithm could be trained to a suitable level.

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

Table 4. Row from FMEA for Case Study 4

Proposed Solution

A previous paper by the authors introduced the possibility of a framework-level improvement to the FMEA process to address the challenges identified earlier. This section expands on that concept and provides greater detail on a proposed approach.

Scope Definition via NUD Filtering

Complex system development is rarely "greenfield." Most systems evolve by adding innovation to existing, proven platforms, resulting in significant differences in maturity between predicate elements and new functionality. In many cases, the resulting product must be treated as a new system rather than a simple enhancement. This creates substantial challenges in scoping FMEA, as the number of potential failure modes can quickly become unmanageable, even with digital tools. To address this, a filtering process is required. We propose the *New, Unique, and Difficult (NUD)* analysis.

The NUD process establishes criteria for determining which elements of a system should fall within FMEA scope based on user requirements and system concept information. Elements meeting one or more criteria are included in the FMEA scope; those that do not are considered sufficiently proven to present minimal design risk. The criteria also provide directional guidance on the appropriate depth of analysis.

New refers to a user requirement or system concept that is new to the organization, though not necessarily new to the industry. Examples include features present in competitor products or the adoption of established technologies not previously used in a predicate design, such as migrating from an ASIC to a GPU for image processing. For inclusion, features require reference performance expectations, while technologies require reference design information.

FMEA guidance and team structure must focus on whether the design team sufficiently understands these features or technologies to identify credible failure modes and causes. This often requires competitive analysis, as well as evaluation of errata, derating, and reference design limitations to define appropriate architectural bounds.

Unique represents truly first-of-a-kind features or technologies for which no reference designs or market data exist, often involving new intellectual property. In these cases, the central FMEA concern

is feasibility and maturity relative to the target application environment. For example, a novel liquid lens technology evaluated for mobile optical scanning exhibited attractive capabilities but failed to meet temperature and vibration requirements (limitations not initially known). As a result, the FMEA focus shifted toward feasibility rather than traditional failure modes.

Difficult addresses design reuse under changed conditions. Predicate elements are flagged when reused in different architectural contexts (e.g., migrating from a local application to a web-based system), when scaled beyond prior use, when prior yield or failure issues exist, or when materials or processes are altered.

Application of NUD criteria can occur early in the V-model using limited available information. Each criterion defines the type of information required for evaluation: reference designs for New elements, feasibility data for Unique elements, and historical performance data for Difficult elements. Because the process is semi-heuristic, evaluation should be performed by cross-functional teams to reduce bias in assessing design risk.

Failure Mode and Cause Definition

To create the required definitions at this stage of the analysis, the system architecture must exist, with the ability to perform at least first level allocation of requirements to major elements of the architecture. It also requires that a clear definition of requirement as representing a definition of performance vs. a specification representing the definition of how a portion of the design is instantiated be present. These concepts are well understood in the realm of INCOSE and systems engineering textbooks. Further, these concepts are essential to the application of more advanced approaches such as Model Based Systems Engineering (MBSE).

Within this framework, a failure mode is defined as a failure to meet a requirement and must specify how that requirement is violated. If a failure mode cannot be traced to a requirement, it is either not a valid failure mode or is poorly defined. A cause of failure must then be associated with the architectural element responsible for meeting that requirement. If sufficient design detail exists, the cause may represent failure of a specification rather than a requirement.

For example, a system level requirement may specify operation on battery power for a defined duration. A failure mode would be inability to meet that duration. Causes could include inadequate battery capacity or excessive power consumption by a subsystem – each tied to different decomposed requirements. Once a battery design is selected, insufficient capacity becomes a specification failure, whereas excessive subsystem power draw remains a requirement failure.

While similar concepts exist in reliability literature, two gaps remain: these distinctions are often unclear to practitioners (Spreafico et al., 2017), and the extension of requirement-based failure modes to architectural allocation and decomposition is poorly addressed in FMEA texts (Stematis, 2003).Therefore, there is a clear need to improve the FMEA methodology.

Consistent Risk Assessment

In many instances of problems in risk assessment associated with failures, the issue is caused by attempting to evaluate the effect of the *cause of failure* rather than the *failure mode*, i.e. what is the effect if part X breaks, as opposed to what the impact to the user is if the system fails to do Y.

By forcing failure modes to be contextualized in the language of expected performance, a consistent criterion can be devised to evaluate them. Further, as the FMEA process is decomposed by the allocation or assignment of requirements to lower levels of the architecture, these lower-level requirements failures are contextualized by the resulting failure caused to the higher-level requirement and its resulting impact which provides a consistent reference. Further, taking this approach does not prevent the use of any general approach to risk assessment, it simply forces a

consistency in approach by tying it to the top-level system performance.

Reducing Cost and Improving Applicability

Having a filtering model at all to define the scope of an FMEA plan reduces the cost of FMEA. The relevant standards discuss scope but focus on *inclusion* rather than *exclusion* to attempt to ensure completeness of the FMEA process. However, this overlooks what typically represents well proven predicate architecture and detail design leading to relatively low value-added analysis. Focusing on the novel aspects of a design inherently focuses on the unknown risks. Further, by performing the sorting activities to bias towards requirements and functionality, inherently the FMEA process becomes focused on the elements of a project that typically are the justification for existence. Rarely do new projects get funded without a projected ROI based on some value improvement. Finally, by having the FMEA process adapt to follow the requirements and allocation of requirements model it better supports the required resources to make the FMEA valuable.

An important aspect regarding the application of the NUD process is that inherently, it represents a form of design risk assessment to ensure that the highest design risk parts of a developing system are identified prior to design commitments are made. This allows more optimal allocation of design resources in the design process. It may appear that this represents an approach to managing cost of the FMEA process in the development activity, but this is not the primary intent. The purpose of the NUD process is identification of high risk in design and the classification of that risk in the design to better apply resources in having the most impact in reducing those risks. Further, the iteration of the NUD process being applied to the mitigations, *before being committed to* continues to manage the risk throughout the development cycle, preventing otherwise unseen design risk from being introduced at lower levels of the product.

Timing, Decomposition, and Recursive FMEA

In this framework, a significant key "left shift" is to focus on failures of a system *to do* something, as opposed to focusing on ways the elements of the system can fail. This provides numerous improvements in alignment with the V-model, but at the cost of having tangible design to improve. Continuing with this perspective, when analyzing the top-level system with its architecture, the mitigation to a cause of failure becomes a *new requirement* allocated to the module within the system, that *is yet to be detail designed* to ensure that its lower-level requirements are complete. By approaching it in this manner the lower-level detail design is better defined initially, rather than "reworked" as part of the FMEA analysis. None of this prevents the use of tools such as fault tree analysis or other methods to design in reliability, rather it supports that in the initial design by improving the quality of the inputs to the design. This then reduces the level of design rework required.

Further, when a new requirement is created as a mitigation to a cause of failure and assigned to a subsystem, the entire FMEA process is *restarted at the NUD filter*. Essentially, the filter is reused to determine whether the requirement mitigation is attempting to reduce risk in the design by adding a new risk element. Using a previous example:

- Requirement: Operate under battery for X hours.
- Failure Mode: Does not operate for the X hours starting from full battery charge.
- Cause of Failure: Inadequate power capacity in the battery system.
- Design Mitigation: Set requirement for battery capacity at 200% of projected power budget to meet X hours.

At this point, this new requirement must be reviewed against the NUD criteria. Essentially, once the requirement has been set, the question of

feasibility must be addressed. The NUD criteria forces consideration of whether there is design risk associated with this requirement. If there is not, then the FMEA process can stop, and standard design practices executed to the added requirement. If there is, then the NUD processes guides what that risk is, and a next level decomposed FMEA is performed. The example continues:

- Requirement: 200% power capacity reviewed against NUD shows that the power density required for the space in the system is beyond proven technology for high reliability battery systems.

Thus, in this example, the mitigation is flagged for feasibility in a structured way before the design is committed to allowing the time for a design pivot if needed. Following this example further, if it is determined that the requirement should stand, and a battery technology chosen, conventional FMEA can be applied to that aspect of the detail design specifically, again focusing expertise in the areas needed.

As an additional benefit, the "design map" of the areas of the system requiring these lower-level FMEAs and their depth provides the systems engineer an evolving metric of the design risks being resolved vs. created as a complex project evolves and what portions of the design are incurring those risks.

Managing Cross-Disciplinary Complexity

As a final aspect of the model, a meta complaint that manifests itself in different ways about FMEA is the difficulty in managing the increasingly interconnected but different engineering disciplines involved in design. This is increasingly common between software and hardware, as software becomes more demanding and Moore's law and the need for backward compatibility have difficulty maintaining pace.

Traditionally, in classic FMEA the term "Critical to Quality" or CtQ has been associated with some form of *importance* definition typically associated with severity scoring, often using language of "loss of critical function" or "extreme dissatisfier" as qualitative terms. These types of criteria are notoriously difficult to apply and typically become redundant to the severity score. In this model the term is repurposed as a classifier to identify mitigations that are placed on a *different area or engineering discipline of the design*.

Continuing our power system example, we can posit that the high-power density battery system is feasible, but its FMEA indicates that to prevent the risk of the battery exploding the charge and discharge rates must be strictly controlled. This requirement of charge and discharge measurement and control now become a software element, crossing the boundary from hardware to software. Thus, this new requirement on software becomes a CtQ. Essentially, there is no inherent feature level software performance requirement for such controls. However, it is necessary to ensure that the mitigation of charge and discharge management for the top-level battery requirement is met in a safe and effective manner. It may also be that additional hardware is required that is not intended for the battery subsystem. These requirements would also flag as CtQ.

Now the secondary map of CtQ designations provides the system engineer a metric of what degree of loose coupling and independence between different elements of the architecture exist, which is a well-known consideration for good systems design.

Conclusions

This paper builds on previous work in examining systemic challenges in application of FMEA under V-model development. While not explicitly stated, without systemic solutions to these challenges, they will only become worse as complexity in design increases. From that work, this paper provides the structure and detail of a proposed approach to address these systemic challenges. With explanation of how it can be used.

In principle, the methods described here are additive to the principles currently in use with FMEA. As described, classic FMEA methodology is described as a 'bottom-up' approach. Therefore, a level of design maturity required to perform this sort of 'bottom-up' analysis cannot exist early in V-model development approaches. The structure provided by the authors creates a 'top-down' approach to the FMEA concept. This allows for the assessment of the design risks to be managed prior to design commitment, driving more robust design activity. Once the level of design maturity is reached, if there are remaining high risk portions of a design, classic FMEA or other variants with 'bottom-up' principles can be employed to provide confirmation of the design before final commitment.

This model has been deployed and tested in industry with positive results. These will be discussed in a future paper.

References

Anjalee, J. A. L., Rutter, V., & Samaranayake, N. R. (2021). Application of failure mode and effect analysis (FMEA) to improve medication safety: A systematic review. *Postgraduate Medical Journal*, *91*(1145), 168-174.

Automotive Industry Action Group. (2019). *AIAG & VDA FMEA Handbook.* https://www.aiag.org/training-and-resources/manuals/details/FMEAAV-1?utm_source=chatgpt.com

Department of Defense. (1949). *Procedure for performing a failure mode, effects, and criticality analysis* (MIL-P-1929).

Department of Defense. (1985). *Defense system software development* (DOD-STD-2167).

Food and Drug Administration. (2025). *Quality System Regulation* (21 C.F.R. § 820).

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

International Electrotechnical Commission. (2018). *Failure modes and effects analysis (FMEA and FMECA)* (IEC 60812:2018).

Kirkire, M. S., Rane, S. B., & Jadhav, R. S. (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.

Liu, H. C., Li, P., Y., J. X., & Chen, Y. Z. (2015). A novel approach for FMEA: Combination of interval 2-tuple linguistic variables and gray relational analysis. *Quality and Reliability Engineering International, 31*(5), 761-772.

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.

Ozarin, N. W. (2013). Bridging software and hardware FMEA in complex systems. In *Proceedings of the 2013 Annual Reliability and Maintainability Symposium* (pp. 1-6).

Perreault, D. C., Gallegos, E. E., Paglioni, V., & Bradley, T. H. (2024). Usability challenges and failure modes and effects analysis (FMEA) within the V-model. In *Proceedings of the International Council on Systems Engineering Human Systems Integration Conference*.

Rook, P. (1986). Controlling software projects. *Software Engineering Journal, 1*(1), 7-16.

Spreafico, C., Russo, D., & Rizzi, C. (2017). A state-of-the-art review of FMEA/FMECA including patents. *Computer Science Review, 25*, 19-28.

Stematis, D. H. (2003). *Failure Mode and Effect Analysis: FMEA from Theory to Execution.* ASQ Quality Press.

Biography

Dan Perreault

Mr. Perreault is a PhD candidate in the Systems Engineering Department at Colorado State University. He is also the Director of Product Performance and Post Market Surveillance for Philips Ultrasound.

Erika E Gallegos

Dr. Gallegos is an Associate Professor in the Department of Systems Engineering at Colorado State University. She holds a BS in Civil Engineering from Oregon State University and an MS and PhD in Civil Engineering from the University of Washington.