Advances in Failure Mode and Effects Analysis for Reliability Engineering in Complex Hardware-Software Systems

Francis Ojeakhena[1], Emmanuel Chukwuma Okeke[2*], Eromosele Asibor[3], Oluranti Ogundapo[4]
[1]TotalEnergies Upstream Nigeria, Port Harcourt, Nigeria
[2]The University of Texas at Austin, Austin, TX, USA
[3]Nigeria National Exploration and Production (NEPL) Ltd, Benin City, Nigeria
[4]Huawei Technologies, Nigeria
* Corresponding author: Emmanuel Chukwuma Okeke, emmanuelokeke68@gmail.com

ABSTRACT

Failure Mode and Effects Analysis (FMEA) has been a foundation of reliability engineering for more than six decades, yet its classical form was designed for systems dominated by mechanical and electrical components with well-characterized physical failure modes. Modern products such as automotive electronic control units, avionics, medical devices and industrial controllers are hardware-software systems in which a large share of failures originate in requirements, logic, timing and component interaction rather than in wear-out of parts. This paper reviews the advances in FMEA for such systems that emerged to date. Using an integrative review methodology, it screened standards, peer-reviewed literature, practitioner sources and a consolidated seed list, retaining sources published to date. The literature synthesis covers seven themes: the limitations of classical FMEA; the third edition of IEC 60812, the AIAG-VDA harmonized handbook review draft with Action Priority and FMEA for Monitoring and System Response, the second edition of ISO 26262 and the STPA Handbook; model-based FMEA from SysML; software FMEA taxonomies; fuzzy, multi-criteria and severity-led prioritization; hybridization with system-theoretic and security analyses; and related literature on maintenance, inspection, process safety, power systems and information security. Building on the synthesis, the paper proposes the Integrated Hardware-Software FMEA (IHS-FMEA) conceptual model with eight elements and five testable propositions, and illustrates it with a brake-by-wire controller. The discussion shows that the advances to date are complementary and that their value depends on a shared architecture model and evidence-grounded ratings. Theoretical, practical and standards-related implications are drawn, limitations are stated and directions for future research are proposed.

Keywords: failure mode and effects analysis; FMECA; software FMEA; hardware-software systems; reliability engineering; model-based systems engineering; SysML; Action Priority; fuzzy FMEA; STPA; ISO 26262; IEC 60812; integrative review; conceptual model.

DOI: 10.21590/ijhit.v3.i1.03

1. INTRODUCTION

FMEA is a structured, bottom-up technique for identifying how the elements of a product or process can fail, what effects those failures produce at local and system levels, and which treatments should be prioritized. Since its formalization in the U.S. military procedure MIL-P-1629 in 1949 and its adoption across aerospace, automotive, nuclear and medical industries, FMEA has become one of the most widely required analyses in reliability and safety engineering. Its strengths are well known: it is systematic, it forces cross-functional teams to reason about causes and consequences, and it produces a traceable record that links design decisions to risk.

Those strengths rest on assumptions that do not hold comfortably in modern systems. Classical FMEA presumes that the system can be decomposed into components whose failure modes can be enumerated, that failures are largely independent, and that the likelihood of each failure can be estimated from physical failure data. In a hardware-software system these assumptions weaken. Software does not wear out; it fails because a latent design fault meets a triggering input or state. Many hazardous behaviors arise when every component works exactly as specified but the interaction among them is unsafe. Occurrence ratings for software causes cannot be read from a parts-count handbook. And the volume of possible failure paths in a system with dozens of networked controllers overwhelms manual spreadsheets.

By the middle of the 2010s these tensions had produced a substantial body of criticism and proposed fixes. The year 2018 is notable because several of these threads converged at once into published standards, handbooks and peer-reviewed methods. This paper examines that convergence. Its aims are to:

1. set out a transparent review methodology for identifying and synthesizing the relevant literature;
2. summarize the standards and industry guidance on FMEA that were published, revised or reviewed in 2018 and explain their significance for hardware-software systems;
3. review the main research advances of the period in model-based FMEA, software FMEA, risk prioritization and hybrid methods;

4. develop a conceptual model that integrates these advances into a single hardware-software FMEA framework and demonstrate it with an illustrative example; and
5. discuss the implications, limitations and future research directions that follow from the synthesis.

The review draws on standards documents, industry handbooks, conference proceedings and journal articles published to date, with emphasis on work that was published, released for review or presented in 2018. Earlier foundational work is cited where it is needed to explain the 2018 contributions. Because FMEA is applied well beyond automotive and aerospace electronics, the review also positions these advances against a broad body of related and adjacent literature on reliability engineering, predictive maintenance, asset integrity, process and occupational safety, power and communication systems, operational technology security, enterprise IT reliability and artificial intelligence, drawn from work published to date. That literature is synthesized in Section 3.7.

The remainder of the paper is organized as follows. Section 2 describes the research method. Section 3 presents the results and discussion: the literature synthesis (Sections 3.1 to 3.7), the conceptual model with an illustrative application (Section 3.8), the discussion of findings against the research questions (Section 3.9) and the implications for theory, practice and standards (Section 3.10). Section 4 concludes the paper and states its limitations and directions for future research.

2. RESEARCH METHOD

2.1 Review design

This study adopts an integrative literature review design. Integrative reviews are suited to topics in which the evidence base combines standards, industry guidance, conceptual papers and empirical studies, and in which the aim is to synthesize a field and propose a conceptual framework rather than to estimate a pooled effect. The review was structured in five stages: definition of the research questions, identification of sources, screening against inclusion criteria, thematic coding and synthesis, and development of a conceptual model.

Three research questions guided the review:

- **RQ1.** What standards, guidance documents and research methods relevant to FMEA of complex hardware-software systems were published, revised or reviewed in 2018?
- **RQ2.** How do these advances address the known limitations of classical FMEA in software-intensive and interaction-dominated systems?

- **RQ3.** How can these advances be integrated into a coherent conceptual model for reliability engineering practice?

2.2 Sources and search strategy

Four groups of sources were searched. The first comprised the publications of standards bodies and industry associations, specifically the International Electrotechnical Commission, the International Organization for Standardization, the Institute of Electrical and Electronics Engineers, the Automotive Industry Action Group and the German Association of the Automotive Industry. The second comprised peer-reviewed journals and conference proceedings in reliability, safety, systems and software engineering. The third comprised practitioner material such as handbooks, white papers and conference presentations from systems engineering and quality organizations. The fourth was a consolidated seed list of related publications supplied for this study, which was screened for related and semi-related work in reliability, safety, maintenance, security and engineering systems. Search terms combined the method names (FMEA, FMECA, FMEDA, software FMEA, STPA) with qualifiers for the system type (hardware-software, embedded, software-intensive, mechatronic) and for the topic of each stream (SysML, model-based, risk priority number, fuzzy, Action Priority, functional safety, cybersecurity).

2.3 Inclusion and exclusion criteria

Sources were included if they were published to date, or were released for public review in late 2017 and reviewed during 2018; if they addressed FMEA, a closely related failure or hazard analysis, or the data and controls that support such analyses; and if they were relevant to complex engineered systems. Foundational works published before 2016 were included only where they were necessary to explain the 2018 contributions. Sources published after 2018 were excluded so that the review reflects the state of knowledge in the year under study. Sources in unrelated domains, such as agriculture, education, law and clinical care, were excluded.

2.4 Screening of the seed literature

The consolidated seed list contained 1,056 entries across many subject areas. Screening by publication year identified 52 entries published to date. These were then screened for topical relevance to reliability, safety, risk, maintenance, power and communication systems, inspection technology, security and engineering governance. Twenty-eight entries met both criteria and were retained as related or semi-related literature; the remaining 24 were excluded because their subject matter, such as livestock nutrition, special-needs pedagogy or tax policy, did not bear on failure analysis.

Table 1. Summary of the review protocol

Stage	Description	Outcome
Research questions	Three questions on 2018 advances, their response to FMEA limitations and their integration	RQ1 to RQ3
Source identification	Standards bodies, journals and proceedings, practitioner material, consolidated seed list	Four source groups
Date screening	Publication to date (drafts under review in 2018 included)	52 of 1,056 seed entries eligible by date
Relevance screening	Related or semi-related to failure, hazard or risk analysis of engineered systems	28 seed entries retained; core FMEA sources retained
Thematic coding	Coding of each source to one or more analytic streams	Seven themes (Sections 3.1 to 3.7)
Synthesis and modeling	Integration of themes into a conceptual model with propositions	Conceptual model (Section 3.8)

2.5 Analysis and synthesis

Each retained source was read and coded inductively against the research questions. The codes were grouped into seven themes: limitations of classical FMEA, standards and guidance, model-based FMEA, software FMEA, risk prioritization, hybridization with system-theoretic and security analyses, and cross-domain related literature. Within each theme the synthesis identified what changed in 2018, which limitations the change addressed and what remained unresolved. The themes were then compared to identify the functional elements that a combined method would need, which formed the basis for the conceptual model in Section 3.8. An illustrative application with invented ratings was used to test the internal coherence of the model rather than to provide empirical validation.

3. RESULTS AND DISCUSSION

The results are presented in four parts. Sections 3.1 to 3.7 report the literature synthesis in seven themes, beginning with the limitations of classical FMEA that motivated the 2018 advances, then examining standards, model-based methods, software FMEA, risk prioritization and hybrid methods, and closing with related literature from adjacent domains. Section 3.8 presents the conceptual model developed from the synthesis and illustrates it with an example. Section 3.9 discusses the findings against the research questions, and Section 3.10 sets out their implications.

3.1 Classical FMEA and its limitations in hardware-software systems

3.1.1 The classical procedure

A conventional design FMEA proceeds by defining the scope, decomposing the item into a hierarchy of systems, subsystems and components, listing the functions of each element, identifying failure modes for each function, and tracing each failure mode to its effects and causes. Each failure mode is then rated on three ordinal scales, usually from 1 to 10: Severity (S) of the effect, Occurrence (O) of the cause, and Detection (D), the ability of current controls to find the cause or failure mode before it reaches the customer. The product S x O x D gives the Risk Priority Number, which has traditionally been used to rank failure modes and to trigger corrective action above a threshold. A criticality analysis (FMECA) replaces or supplements the RPN with a quantitative or matrix-based criticality measure.

3.1.2 Known weaknesses of the RPN

The RPN has been criticized for decades. A frequently cited literature review by Liu, Liu and Liu (2013) examined 75 FMEA papers from 1992 to 2012 and grouped them by the shortcomings of the conventional RPN they tried to overcome. The most common objections are that:

- the three factors are treated as equally important, so a catastrophic but rare failure can rank below a trivial but frequent one;
- the factors are ordinal, so multiplying them is mathematically questionable and the resulting numbers are not evenly distributed (only 120 distinct values exist between 1 and 1000);
- very different risk profiles produce the same RPN, which hides important distinctions;
- team members find it hard to express uncertain judgments as precise integers; and
- the method ignores dependencies among failure modes and among the risk factors themselves.

3.1.3 Challenges specific to software and interaction

Hardware-software systems add a further set of difficulties. Software failure modes are not physical: a software element may produce a wrong value, a late value, no value, a value at the wrong time or in the wrong sequence, or an output that is correct for its specification but unsafe in context. Software causes are systematic design faults, so a statistical occurrence rating has little meaning; the relevant question is whether the fault is present and whether it will be triggered. Interfaces between hardware and software are often the weakest point of an analysis because hardware and software teams analyze their own domains separately. Finally, complex controllers fail through feedback and control problems, for example when a controller acts on a process model that no longer matches reality, which a component-by-component worksheet rarely captures.

These challenges shaped the agenda for the advances described in the following sections.

3.2 Advances in standards and industry guidance in 2018

3.2.1 IEC 60812:2018, third edition

The International Electrotechnical Commission published the third edition of IEC 60812, *Failure modes and effects analysis (FMEA and FMECA)*, on 10 August 2018, replacing the 2006 second edition. The new edition explains how FMEA and FMECA are planned, performed, documented and maintained. Its listed technical changes are directly relevant to hardware-software systems. The normative text was made generic so that it covers all applications; informative annexes were added with examples for safety, automotive, software and service processes; tailoring of FMEA for different applications was described; reporting formats, including a database information system, were introduced; alternative ways of calculating the RPN were added; a criticality-matrix method was provided; and the relationship of FMEA to other dependability methods was described.

Three aspects of this revision deserve emphasis. First, by moving domain-specific content into annexes, the standard acknowledged that a single worksheet template cannot serve mechanical parts, embedded software and service processes equally well; analysts are expected to tailor. Second, the explicit inclusion of a software example in a generic dependability standard gave software FMEA a recognized place alongside hardware FMEA. Third, offering alternatives to the simple RPN product, together with a criticality matrix, formally conceded the long-standing criticisms summarized in Section 3.1.2. The standard also notes that, although FMEA can support safety analysis, it does not itself provide specific guidance for safety applications, which places it alongside rather than in competition with sector safety standards such as IEC 61508 and ISO 26262.

3.2.2 The AIAG-VDA harmonized FMEA handbook

The automotive industry had long operated with two incompatible FMEA references: the AIAG FMEA Manual, fourth edition (2008), used by North American manufacturers, and VDA Volume 4 (2012), used by German manufacturers. Suppliers serving both markets had to maintain different rating tables and worksheet formats. In November 2014 AIAG and VDA began a joint project to harmonize the two. On 27 November 2017 they released a draft of the co-copyrighted AIAG-VDA FMEA Handbook for a 90-day stakeholder review, and the comment period closed on 27 February 2018. AIAG and VDA spent the remainder of 2018 resolving the stakeholder feedback ahead of a final edition.

The 2018 review draft introduced three changes that matter for hardware-software systems:

- **A structured step-wise approach.** The draft organized the analysis into six steps: Planning and Preparation (scoping), Structure Analysis, Function Analysis, Failure Analysis, Risk Analysis and

Optimization, with documentation of results running through all of them. Structure, function and failure analysis are performed explicitly as linked trees, an approach inherited from the software-supported VDA method. This makes the analysis closer to a model than to a free-form table.

- **Action Priority (AP) in place of the RPN.** Instead of multiplying S, O and D, the handbook uses a three-dimensional lookup table that assigns each combination a High, Medium or Low priority for action. Severity dominates the logic, so high-severity effects cannot be masked by favorable occurrence or detection ratings.
- **FMEA for Monitoring and System Response (FMEA-MSR).** This supplemental analysis examines how diagnostic monitoring detects faults during customer operation and how the system responds, for example by entering a safe state or a degraded mode. It was designed for mechatronic and software-controlled systems where safety depends on on-board diagnostics rather than on end-of-line inspection, and it creates a natural bridge between the FMEA and the safety mechanisms required by ISO 26262.

The draft was not universally welcomed. During the 2018 review period some practitioners argued in published commentary that the draft amounted to adopting the VDA approach rather than truly harmonizing the two, and some members of the SAE J1739 committee indicated that the SAE standard would not simply follow it. This debate is itself evidence of how seriously the industry treated the revision.

3.2.3 ISO 26262:2018, second edition

The second edition of ISO 26262, *Road vehicles: Functional safety*, was published on 20 December 2018. It grew from ten to twelve parts and extended coverage to trucks, buses, trailers and motorcycles. For FMEA practice the most important addition is Part 11, *Guidelines on application of ISO 26262 to semiconductors*. Part 11 explains how to decompose integrated circuits for safety analysis, discusses failure modes of digital, analog and mixed-signal blocks, addresses transient faults, multicore devices and silicon intellectual property, and gives worked guidance on base failure rates and diagnostic coverage. These are the inputs to the Failure Modes, Effects and Diagnostic Analysis (FMEDA), the quantitative FMEA variant used to compute hardware architectural metrics.

ISO 26262 continues to call for inductive analyses such as FMEA at system, hardware and software levels, with Part 9 covering safety-oriented analyses and dependent failure analysis. The second edition also refined the treatment of the probabilistic metric for random hardware failures and the verification of safety analyses. At the same time, the standard explicitly does not address hazards caused by the nominal performance of a system, as opposed to malfunctions. That gap was being addressed in parallel by a separate publicly available specification on the safety of the intended functionality

(SOTIF), which was under development during 2018. The separation is significant for FMEA: hazards arising from sensor limitations or algorithm behavior in a fault-free system fall outside a failure-based analysis and call for complementary methods.

3.2.4 Software reliability guidance: IEEE 1633-2016 in practice

IEEE Std 1633-2016, *Recommended Practice on Software Reliability*, was approved in 2016 but its influence on FMEA practice was felt mainly in the following years. Clause 5.2 of the recommended practice covers the development of a failure modes model through software FMEA and software fault tree analysis, and an annex provides supporting detail on SFMEA. The working group was chaired by Ann Marie Neufelder, whose guidance on software FMEA was widely referenced and used in training during this period. The recommended practice explicitly addressed a cultural barrier: reliability engineers often do not understand software, and software engineers often do not understand reliability. By 2018, IEEE 1633 provided a common vocabulary that let hardware FMEA and software FMEA be performed within one reliability program.

3.2.5 The STPA Handbook

In March 2018 Nancy Leveson and John Thomas published the *STPA Handbook*, a practitioner guide to System-Theoretic Process Analysis. STPA is not a variant of FMEA; it is a hazard analysis based on systems theory, in which the system is modeled as a hierarchy of control loops and the analysis searches for unsafe control actions and the scenarios that produce them. The handbook argued that STPA is needed for complex, software-intensive systems because it includes software and human operators in the analysis and captures accidents caused by unsafe interactions rather than component failures. Its publication made STPA accessible to industry teams and stimulated a wave of comparative and hybrid studies with FMEA, discussed in Section 3.6.

Table 2. Summary of standards and guidance relevant to FMEA in 2018

Document	**2018 status**	**Key contribution for hardware-software FMEA**
IEC 60812:2018 (3rd ed.)	Published 10 Aug 2018	Generic normative text; software, safety and automotive annexes; tailoring; alternative RPN methods; criticality matrix; database reporting
AIAG-VDA FMEA Handbook	Draft review Nov 2017 to Feb 2018; comments resolved during 2018	Structure, function and failure trees; Action Priority replaces RPN; FMEA-MSR for diagnostics and system response
ISO 26262:2018 (2nd ed.)	Published 20 Dec 2018	Part 11 semiconductor guidance (failure modes, transient faults, IP, diagnostic coverage) feeding FMEDA

Document	2018 status	Key contribution for hardware-software FMEA
SOTIF specification	Under development in 2018	Addresses hazards from intended functionality, outside the scope of failure-based FMEA
IEEE 1633-2016	In active adoption	Software FMEA and software fault trees within a software reliability program
STPA Handbook	Published March 2018	Systems-theoretic hazard analysis to complement FMEA for interaction and control failures

Sources: IEC (2018); AIAG (2017, 2018); ISO (2018); IEEE (2016); Leveson and Thomas (2018).

3.3 Model-based and automated FMEA

3.3.1 Motivation

A manually maintained FMEA spreadsheet is disconnected from the design it describes. When the architecture changes, the worksheet becomes stale, and in a complex system the effort to update it is so large that FMEA is often performed late, once, and mainly for compliance. Model-based systems engineering (MBSE) offered a way out: if failure information is attached to the architecture model itself, the FMEA can be regenerated whenever the model changes, and failure propagation can be computed rather than reasoned by hand.

3.3.2 SysML-based FMEA

Research on deriving FMEA from SysML models had been building since the early 2010s. Li and Wang (2011) noted that hardware and software safety analyses were traditionally done separately, leaving the hardware-software interface as a weak point, and proposed heuristic rules for transforming SysML diagrams into FMEA and fault tree analysis. Scippacercola and colleagues (2016) integrated FMEA into a model-driven methodology for space systems, introducing custom FMEA diagrams in SysML from which worksheets and qualitative and quantitative analyses could be generated automatically.

By 2018 this line of work had become practical. Earlier work had already shown that software FMEA could be computer-aided from UML design models (Hecht, An and Hecht, 2004), and the SysML approaches extended the idea to whole hardware-software architectures. In these methods an FMEA profile annotates the system model: the analyst defines component failure modes and how each one propagates to its neighbors, and the tool traces propagation through the architecture to generate the worksheet, including effects at each level. The advantages are instructive: only local parameters need to be entered rather than complete worksheet rows, the FMEA can be regenerated almost instantly,

many design alternatives can be evaluated, and the same propagation framework can in principle carry threat effects as well as failure effects. At the INCOSE Great Lakes Regional Conference in October 2018, Hummell presented the case for modeling FMEAs in SysML so that failure analysis is traceable to the systems model. Commercial safety tools also adopted the approach, deriving FMEA worksheets directly from SysML elements and keeping worksheet entries synchronized with the underlying model.

3.3.3 Significance for hardware-software systems

Model-based FMEA addresses several of the challenges in Section 3.1.3. Because hardware blocks, software components and their interfaces appear in one architecture model, propagation across the hardware-software boundary is analyzed explicitly rather than falling between two teams. Because effects are computed from the model, the analysis scales to architectures that would produce hundreds or thousands of worksheet rows. And because regeneration is cheap, FMEA can be used as a design tool early and repeatedly, which is where it provides most value. The structure, function and failure trees of the AIAG-VDA approach point in the same direction: the worksheet becomes a view of an underlying model.

The approach also has limits. Its results are only as good as the failure propagation rules, and those rules must still be written by engineers who understand both the hardware and the software. Behavioral failures that depend on timing, state or sequence are harder to express in structural block diagrams than value failures. Executable models and model-level fault injection, explored for software FMEA before 2018, offered one route to these behavioral effects but were not yet widely used.

3.4 Advances in software FMEA

3.4.1 From black box to failure mode taxonomies

Early guidance often treated software as a black box with a few generic failure modes. By 2018 practice had moved toward structured taxonomies of software failure modes and root causes. Neufelder's work, referenced by IEEE 1633, catalogued several hundred failure mode and root cause pairs and organized software failure modes into categories such as faulty functionality, faulty state management, faulty sequencing, faulty timing, faulty data definition, faulty error handling, faulty processing and faulty algorithms. Analyses can be performed from several viewpoints, including requirements, interfaces, detailed design and code, vulnerabilities, corrective actions, serviceability, usability and process.

3.4.2 Key principles established by the period

Several principles of software FMEA were widely articulated by 2018:

- **Software causes are systematic, not random.** A software FMEA is not used to calculate failure rates. Once a root cause is correctly removed, the corresponding failure does not recur, so the value of the analysis lies in finding and eliminating causes rather than in estimating probabilities.
- **Severity depends on the feature, not the failure mode.** Any software failure mode can be catastrophic in a mission-critical function and harmless elsewhere, so severity must be assessed in the context of the function the code implements.
- **Software must be analyzed against hardware failures.** A key purpose of software FMEA in a hardware-software system is to check how software reacts to sensor faults, communication loss, memory corruption and power transients, which is where detection and system response are actually implemented.
- **Software engineers must participate.** Facilitators may run the session, but the people who understand the code and its failure behavior must identify the failure modes.

3.4.3 Empirical comparisons

Empirical studies published in this period helped calibrate expectations. Sulaman, Beer, Felderer and Höst (2017), in a case-study comparison of FMEA and STPA published online in 2017 in the *Software Quality Journal*, applied both software FMEA and system FMEA to software-intensive systems and found that FMEA and STPA delivered broadly similar results in their case, while differing in the effort required and the kinds of hazards emphasized. They also summarized the core point of software FMEA: software components often do not fail in the traditional sense but instead produce incorrect behavior. Earlier controlled experiments by Abdulkhaleq and Wagner, by contrast, found that STPA identified more software safety requirements than FTA and FMEA. Together these studies suggested that software FMEA is effective for component-level behavioral faults but should be complemented for interaction-level hazards.

3.5 Advances in risk prioritization beyond the RPN

3.5.1 Fuzzy FMEA

Fuzzy set theory had been applied to FMEA since the 1990s, and 2018 saw a steady flow of applications and refinements. In fuzzy FMEA, team assessments of severity, occurrence and detection are expressed as linguistic terms (for example "low", "moderate", "high") mapped to fuzzy numbers, and an expert-defined rule base of IF-THEN statements infers a fuzzy risk priority number (FRPN) that is then defuzzified for ranking. Studies published in 2018 applied this approach to a wide range of systems, including turbocharger failures using a fuzzy inference model (MATEC Web of Conferences, vol. 183), operational risk in a creative business with computer failures ranking highest (MATEC Web of Conferences, vol. 154), maintenance decision-making for a reach stacker crane

(Rabbi, 2018), and maintenance scheduling with risk clustering (International Journal of Industrial Engineering and Production Research, 2018). These studies typically reported that the FRPN separated failure modes that the crisp RPN tied, and that it reflected the relative importance of the risk factors more faithfully.

More advanced uncertainty representations also appeared, notably intuitionistic fuzzy numbers, which capture hesitation in expert judgment as well as the degree of membership, and Petri-net-based reasoning, which can represent dependencies among failure modes.

3.5.2 Multi-criteria decision making

A parallel stream treated failure mode ranking as a multi-criteria decision problem. Methods such as TOPSIS, VIKOR, COPRAS, the analytic hierarchy process and grey relational analysis, often combined with fuzzy or soft-set representations, were used to weight the risk factors and to rank failure modes. Liu and colleagues, for example, combined trapezoidal fuzzy soft sets with a modified COPRAS method and used the Choquet integral to treat the risk factors as interdependent. These approaches directly address the equal-weighting and interdependence criticisms of the RPN.

3.5.3 Action Priority and criticality matrices

Standards bodies took a different route from the research community. Rather than replacing the RPN with a more complex calculation, the AIAG-VDA handbook replaced it with a simpler but better-structured rule: the Action Priority table. IEC 60812:2018 similarly added alternatives to the RPN and a criticality-matrix method. The common idea is that severity should act as a gate, not merely as one of three multiplied factors, and that prioritization should drive action rather than produce a number to be compared with an arbitrary threshold.

Table 3. Comparison of risk prioritization approaches

Approach	How risk is ranked	Strengths	Limitations
Classical RPN	S x O x D on 1 to 10 scales	Simple, familiar, easy to audit	Equal weights; ordinal multiplication; ties; masks high severity
Action Priority (AIAG-VDA)	Lookup table gives High, Medium or Low	Severity-led; consistent across suppliers; action-oriented	Coarse; table is fixed; still relies on ordinal ratings
Criticality matrix (IEC 60812)	Severity versus likelihood grid	Transparent; aligns with safety risk matrices	Ignores detection unless extended; coarse categories

Approach	How risk is ranked	Strengths	Limitations
Fuzzy FMEA	Linguistic ratings, fuzzy rule base, FRPN	Handles vague judgment; separates RPN ties	Rule base must be elicited; results harder to explain
Fuzzy MCDM (TOPSIS, VIKOR, COPRAS, etc.)	Weighted multi-criteria ranking	Weights and interdependence of factors; group decisions	Computationally heavier; sensitivity to weights; limited industrial uptake

Compiled by the author from the sources discussed in Sections 3.1 and 3.5.

3.5.4 Relevance to software causes

These prioritization advances were developed mostly for hardware and process FMEA, and their transfer to software is not straightforward. Occurrence ratings for software causes are inherently judgmental, which favors fuzzy and linguistic methods over crisp numbers, but they are also less meaningful, since a systematic fault either exists or does not. The more defensible practice emerging in this period was to rate software occurrence by proxies, such as the maturity of the development process, the novelty and complexity of the code, and evidence from testing and static analysis, and to let severity and the strength of detection and system response drive priority. The Action Priority and FMEA-MSR concepts fit this practice well.

3.6 Hybridization with system-theoretic and security analyses

3.6.1 FMEA and STPA

The publication of the STPA Handbook in 2018 accelerated work comparing and combining STPA and FMEA. Case studies in automotive and medical domains reported that FMEA had difficulty identifying unsafe interactions between systems, design flaws, flawed requirements and human-interaction behaviors, while STPA found causes that overlapped with FMEA results and additional causes at higher levels of abstraction.

Rather than treating the two as rivals, a complementary pattern emerges from this work. FMEA is weak at handling component interactions, while STPA, as presented in the 2018 handbook, does not explicitly rank risk. STPA can therefore work top-down to identify unsafe control actions and loss scenarios at the system level, and FMEA can work bottom-up to examine the hardware and software components that implement each control loop and to prioritize them (Abdulkhaleq and Wagner, 2015; Leveson and Thomas, 2018).

3.6.2 FMEA and cybersecurity

Connected vehicles and networked industrial systems made it clear that a malicious input can produce the same effect as a random failure. Model-based FMEA frameworks of the kind described in Section

3.3.2 can, in principle, carry attack propagation alongside failure propagation, using known vulnerability and weakness catalogues for software components. The second edition of ISO 26262 also strengthened the interface between functional safety and cybersecurity. In this view, FMEA becomes one element of a broader dependability analysis in which failure modes, threats and functional insufficiencies are examined together.

Enterprise security research of the period supports this view, offering conceptual models for insider threat classification and risk modeling in complex digital systems and lessons from the offline assessment of security-critical systems (Akeju et al., 2018; Ladapo et al., 2018).

3.7 Related reliability, safety and security literature across domains

The 2018 advances did not develop in isolation. FMEA shares its core logic, identifying how elements can fail, what follows and which risks deserve action, with a wide family of reliability, safety and security methods used across engineering sectors. This section synthesizes related and semi-related literature published to date that either applies FMEA-like reasoning in other domains, supplies the data and models that make Severity, Occurrence and Detection ratings more defensible, or addresses failure sources that a hardware-software FMEA must consider.

3.7.1 Maintenance, asset integrity and reliability data

A persistent weakness of FMEA is that Occurrence and Detection ratings rest on judgment. Work on predictive maintenance and condition monitoring in critical power and energy infrastructure shows how sensor data and diagnostic analytics can detect degradation before functional failure, which is precisely the evidence an analyst needs to justify a detection rating (Adaramola et al., 2018). Risk-based inspection, a close relative of FMECA that ranks equipment by the consequence and likelihood of degradation, has been applied to fitness-for-service assessment of ageing offshore pressure vessels, and condition-based maintenance concepts for buoy bearings and swivels link failure mechanisms directly to maintenance actions that preserve availability (Obetta, 2018; Obetta & Ninduwezuor-Ehiobu, 2018). Reviews of corrosion inhibition for mild steel describe the degradation mechanisms that underlie many hardware failure causes (Atta et al., 2018). Studies of procurement optimization in oil and gas operations and of inventory availability for plant uptime show that the effects column of an FMEA extends to the supply chain, since spare-parts availability determines how long a failure keeps a system down (Adaramola et al., 2018; Atta et al., 2018; Obetta, 2018; Obetta & Ninduwezuor-Ehiobu, 2018; Okonkwo et al., 2018a; Okonkwo et al., 2018b).

3.7.2 Process safety, occupational risk and hazard modeling

Process and occupational safety research applies the same bottom-up hazard identification and cause-and-effect reasoning that underlies FMEA. Frameworks for operational integrity in high-risk drilling balance productivity against safety compliance in a way that mirrors the trade-offs in FMEA optimization, while integrated heat stress risk models and data-driven occupational safety risk control for large energy infrastructure projects show how hazard causes, exposures and controls can be structured and quantified (Arumosoye & Obriki, 2018; Nwangwu et al., 2018; Obriki & Arumosoye, 2018). Data-driven and spatially explicit risk modeling frameworks for subsurface contaminant migration demonstrate how uncertain, data-limited environments can still support risk ranking, a problem directly analogous to rating occurrence for novel hardware-software designs (Azeez & Badmus, 2018; Lamidi & Olamide, 2018).

3.7.3 Electrical machines, power and telecommunication systems

Electrical machines and power networks are core hardware elements of most hardware-software systems, and their failure behavior is shaped by design optimization. Multi-objective and multi-algorithm design studies of single-phase induction motors address losses, weight and efficiency, which drive the thermal and wear-out failure mechanisms that a hardware FMEA must capture (Amayo, 2017; Amayo & Popoola, 2017a; Amayo et al., 2015). Work on voltage control in the presence of distributed generation describes control interactions in which software-based regulation and power hardware jointly determine system behavior, and studies of telecommunication deployment for network quality address the communication links on which networked controllers depend (Amayo & Popoola, 2017b; Oshevire et al., 2017).

3.7.4 Inspection technologies and unmanned systems

Unmanned aerial vehicles are themselves complex hardware-software systems and are also increasingly used as detection tools in asset management. Conceptual frameworks for integrating UAVs into national power grid inspection programs and reviews of UAV methods for transmission line inspection describe both the technologies that improve detection of physical faults and the operational and regulatory considerations that shape how such systems are deployed (Dagodzo, 2018a; Dagodzo, 2018b).

3.7.5 Cybersecurity, information risk and enterprise IT reliability

As Section 3.6.2 noted, a malicious action can produce the same effect as a random failure. Conceptual models for insider threat classification and risk modeling in complex digital systems, frameworks for legal and ethical risk modeling in enterprise data protection governance, and lessons from offline assessment of security-critical directory systems offer structured approaches to identifying threat-

derived failure modes and ranking them, steps that parallel FMEA failure and risk analysis (Akeju et al., 2018; Ladapo et al., 2018; Mbonu et al., 2018). On the IT side, work on scalable and secure cloud architectures for enterprise messaging, energy-efficient virtual machine placement, performance of multi-tenant collaboration platforms under high concurrency and CI/CD pipeline strategies for platform deployment characterizes the capacity, configuration and release failure modes of software-intensive services, and shows how automated pipelines act as detection controls for software defects before deployment (Ahmed & Odejobi, 2018a; Ahmed & Odejobi, 2018b; Badmus et al., 2018; Odejobi & Ahmed, 2018b).

3.7.6 Risk governance, digital operations and aviation safety

FMEA is effective only when its findings are acted on and audited. Technology-enabled risk assessment frameworks for internal audit quality and reviews of digital transformation in operations management, covering automation and analytics, describe the governance and digital infrastructure through which risk analyses are tracked and verified (Agboada et al., 2018; Akomolafe & Agu, 2018). In aviation, a sector that pioneered FMEA, predictive compliance monitoring based on consumer complaint data has been proposed as a way to detect systemic safety risks early, illustrating how field data can feed back into hazard identification and risk assessment (Adeyelu, 2018).

3.7.7 Synthesis of the literature

Across these domains three patterns stand out. First, the bottom-up logic of FMEA recurs under different names, such as risk-based inspection, hazard modeling and threat modeling, which suggests that cross-domain transfer of rating scales, taxonomies and tools is both possible and under-exploited. Second, the main contribution of condition monitoring, inspection technology and data-driven risk modeling is to make Occurrence and Detection more measurable, addressing the subjectivity criticized in Section 3.1.2. Third, the spread of software, connectivity and automation across power, process, IT and aviation systems confirms that the hardware-software focus of the 2018 advances was well placed.

3.8 Conceptual model

3.8.1 Rationale

The synthesis shows that each 2018 advance addresses a different limitation of classical FMEA, but none addresses all of them. Model-based methods address scalability and traceability; software FMEA taxonomies address the identification of software failure modes; Action Priority and fuzzy methods address prioritization; FMEA-MSR addresses diagnostics and system response; STPA addresses interaction and control failures; and the cross-domain literature supplies evidence for occurrence and detection. A practitioner who adopts only one of these gains little against the others. This subsection

therefore proposes an Integrated Hardware-Software FMEA (IHS-FMEA) conceptual model that combines them into one analysis framework.

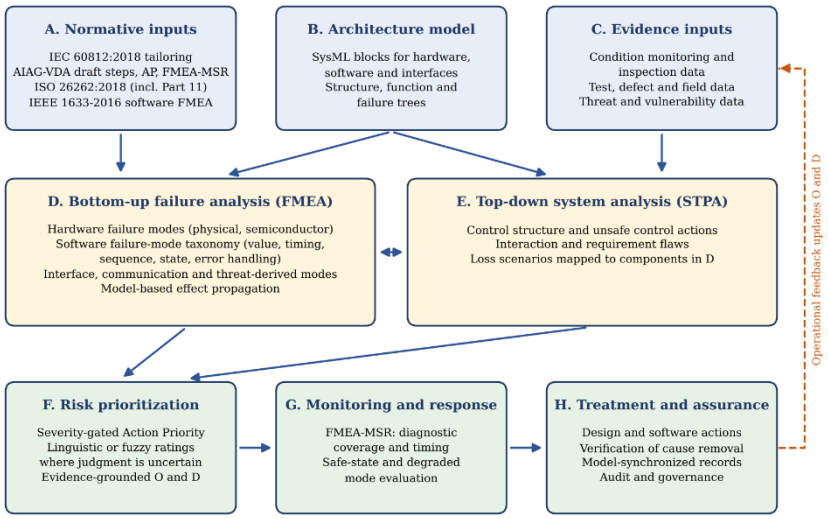


Figure 1. Integrated Hardware-Software FMEA (IHS-FMEA) conceptual model

Source: developed by the author from the literature synthesized in Section 3.

Table 4. Elements of the IHS-FMEA model and their basis in the literature

Element	Function in the model	Main basis in the literature
A. Normative inputs	Define scope, steps, rating logic and required analyses; enable tailoring	IEC 60812:2018; AIAG-VDA draft; ISO 26262:2018; IEEE 1633-2016 (Section 3.2)
B. Architecture model	Single SysML model of hardware, software and interfaces from which the worksheet is generated	SysML-based FMEA and structure, function and failure trees (Section 3.3)
C. Evidence inputs	Field, test, inspection and threat data that ground occurrence and detection ratings	Condition monitoring, inspection and risk modeling literature (Section 3.7)
D. Bottom-up failure analysis	Hardware, software, interface and threat-derived failure modes with model-based propagation	Software FMEA taxonomies; ISO 26262-11; security integration (Sections 3.4, 3.6.2)
E. Top-down system analysis	Unsafe control actions and interaction flaws traced to components in D	STPA Handbook; FMEA and STPA comparisons (Section 3.6.1)
F. Risk prioritization	Severity-gated priority with linguistic ratings where judgment is uncertain	Action Priority; criticality matrices; fuzzy and MCDM methods (Section 3.5)

Element	Function in the model	Main basis in the literature
G. Monitoring and response	Evaluation of diagnostic coverage, timing and safe-state behavior	FMEA-MSR; FMEDA and diagnostic coverage (Sections 3.2.2, 3.2.3)
H. Treatment and assurance	Design actions, verification of cause removal, model-synchronized records and audit	Software FMEA principles; risk governance literature (Sections 3.4.2, 3.7.6)

3.8.2 Structure of the model

The model, shown in Figure 1, has eight elements arranged in three layers. The input layer contains the normative inputs that define how the analysis is performed (A), the architecture model on which it is built (B) and the evidence that grounds its ratings (C). The analysis layer contains a bottom-up failure analysis (D), which is the FMEA proper, and a top-down system analysis (E), which applies STPA to the same architecture; the two are linked so that loss scenarios found top-down are traced to component causes and component failure modes are checked for system-level effects. The decision layer contains risk prioritization (F), monitoring and response evaluation (G) and treatment and assurance (H). A feedback loop carries operational and test data back into the evidence inputs, so that occurrence and detection ratings are revised as the system matures.

3.8.3 Propositions

The model implies five propositions that can guide practice and be tested in future research:

- **P1.** Generating the FMEA from a shared architecture model (B to D) increases the completeness of identified hardware-software interface failure modes compared with separate hardware and software worksheets.
- **P2.** Linking bottom-up and top-down analysis (D with E) identifies interaction and requirement-related causes that neither method identifies alone.
- **P3.** Severity-gated prioritization (F) reduces the masking of high-severity failure modes relative to the multiplicative RPN.
- **P4.** Grounding occurrence and detection in evidence inputs (C), and updating them through operational feedback, improves the stability of risk rankings across analysis teams and over time.
- **P5.** Explicit evaluation of monitoring and system response (G) improves the alignment between FMEA findings and the safety mechanisms required by functional safety standards.

To show how the model operates, the following subsections sketch an analysis of a simplified electronic brake-by-wire controller. The example is illustrative, and its ratings are invented for demonstration; it is not drawn from a real product.

3.8.4 Illustrative application: system description

The item consists of a pedal position sensor pair, a brake electronic control unit (ECU) with a dual-core microcontroller, control software that converts pedal input into a clamping force request, a CAN bus link to the vehicle stability controller, and an electromechanical actuator at each wheel. A diagnostic monitor compares redundant pedal sensors, checks CAN message counters and timeouts, and supervises the control software with a watchdog.

3.8.5 Illustrative application: applying the model

1. **Normative inputs and architecture model (elements A and B; AIAG-VDA steps 1 and 2).** The architecture is captured as a SysML block model containing hardware blocks (sensors, microcontroller, actuator), software components (input processing, force calculation, diagnostics) and the interfaces between them.
2. **Bottom-up failure analysis (element D; steps 3 and 4).** Failure modes are attached to each block: hardware modes such as sensor drift, stuck-at faults in microcontroller logic (with transient faults considered per ISO 26262-11), and software modes from a taxonomy such as wrong value, late output, omitted output and faulty state transition. Propagation rules generate the effect chain to the vehicle level.
3. **Top-down system analysis (element E).** An STPA control structure identifies unsafe control actions, for example "brake force not provided when the driver presses the pedal" or "brake force provided when the stability controller requests release", and scenarios arising from interaction, such as the ECU acting on a stale CAN value that appears valid.
4. **Risk prioritization and monitoring and response (elements F and G).** Each cause is rated. For diagnostic-covered faults, FMEA-MSR evaluates whether the monitor detects the fault in time and whether the system response (for example a degraded mode with reduced assistance and a driver warning) keeps the vehicle in a safe state.
5. **Treatment and assurance (element H).** High-priority items receive design actions; results are kept in the model so that the analysis is regenerated when the design changes.

Table 5. Excerpt of an illustrative hardware-software FMEA for the brake-by-wire controller

Element	Failure mode	Cause	Effect (vehicle level)	S	O	D	RPN	AP
Pedal sensor A	Output drifts high	Sensor ageing	Unintended braking	9	3	3	81	H
Force calculation SW	Wrong value	Unit conversion error in calibration table	Excessive clamping force, wheel lock	9	2	4	72	H
Input processing SW	Late output	Task overrun under high CAN load	Delayed braking response	10	2	5	100	H
CAN interface	Stale message accepted	Counter check not implemented for one message	Acts on outdated stability request	8	3	6	144	H
Diagnostics SW	Omitted warning	Fault flag not latched after reset	Driver unaware of degraded mode	6	3	4	72	M
Actuator driver	Intermittent drive	Solder joint fatigue	Uneven braking, pull to one side	7	4	3	84	M
HMI display	Icon flicker	Display refresh timing	Minor annoyance	3	6	4	72	L

Ratings are invented for illustration. AP values follow the general severity-led logic of the AIAG-VDA approach and are not taken from the official tables.

3.8.6 Illustrative application: observations

Table 5 shows why the 2018 changes mattered. Three very different failure modes share an RPN of 72, ranging from a severity-9 software calculation error to a severity-3 display annoyance. Ranked by RPN alone, the stale CAN message (144) leads, the severity-10 late output (100) follows, and the severity-9 force calculation error sits level with the cosmetic display fault. A severity-led Action Priority separates these cases immediately. The analysis also illustrates the value of software failure mode taxonomies: three of the four high-priority items are software or interface causes that a hardware-only worksheet would have missed. Finally, the stale CAN message case is a classic interaction problem; it is found most readily by the STPA view of the control loop and then pinned to a specific software cause by the FMEA.

3.9 Discussion

3.9.1 A converging picture

Taken together, the developments of 2018 describe a coherent evolution. FMEA moved from a stand-alone spreadsheet to a view of a system model; from a hardware-centric method to one with recognized software procedures; from a single multiplicative index to severity-led and uncertainty-aware prioritization; and from a self-contained analysis to one component of an integrated set that includes FMEDA, fault tree analysis, STPA, SOTIF analysis and security analysis. Standards (IEC 60812:2018, the AIAG-VDA handbook, ISO 26262:2018) codified the parts of this evolution that were mature, while research explored the parts that were not.

3.9.2 Answering the research questions

With respect to RQ1, the review identified four normative developments in or around 2018 (IEC 60812:2018, the AIAG-VDA review draft, ISO 26262:2018 and the continuing adoption of IEEE 1633-2016), one major methodological publication (the STPA Handbook) and active research streams in model-based FMEA, software FMEA and uncertainty-aware prioritization. With respect to RQ2, these advances map closely onto the classical limitations identified in Section 3.1: model-based methods address scale and traceability, software taxonomies address non-physical failure modes, Action Priority and fuzzy methods address the RPN, and STPA addresses interaction failures. The mapping is not complete, however; occurrence for software causes and behavioral failure modes remain weakly supported. With respect to RQ3, the IHS-FMEA model shows that the advances are complementary rather than competing, and that their value depends on how they are connected, particularly through the shared architecture model and the feedback of evidence into ratings.

3.9.3 Positioning the conceptual model

The IHS-FMEA model does not introduce a new analysis technique. Its contribution is integrative: it specifies how existing techniques that were maturing in 2018 should be sequenced and linked. In this respect it is consistent with the direction of IEC 60812:2018, which describes the relationship of FMEA to other dependability methods, and with the structure, function and failure trees of the AIAG-VDA draft, which already treat the worksheet as a view of a model. It extends those references by making the top-down complement and the evidence feedback loop explicit. The cross-domain literature suggests that the model is not specific to automotive electronics: risk-based inspection, condition monitoring and threat modeling in energy, process and IT systems perform the same functions as elements C, F and G.

3.9.4 Interpreting the illustrative application

The brake-by-wire example indicates that the model behaves as intended: severity-gated prioritization separated failure modes that the RPN tied, software and interface causes dominated the high-priority

set, and the most consequential interface fault was found most readily through the top-down view. Because the ratings were invented, the example demonstrates internal coherence only and should not be read as evidence of effectiveness.

3.10 Implications

3.10.1 Theoretical implications

The review has three theoretical implications. First, it reframes FMEA from a self-contained worksheet method to one component of a model-centered dependability analysis, which suggests that research on FMEA should study the interfaces between methods as much as the methods themselves. Second, it shows that the long-standing debate over the RPN is partly a debate about evidence: prioritization schemes cannot compensate for ratings that are not grounded in data, so research on prioritization should be coupled with research on how occurrence and detection are estimated. Third, the five propositions provide a basis for empirical testing of integrated approaches, which the 2018 literature largely lacked.

3.10.2 Practical implications

For reliability engineers working on hardware-software systems, the practical lessons of this period can be summarized as follows. Build the FMEA on an architecture model and keep it synchronized with the design. Analyze software explicitly using a recognized taxonomy and involve the software engineers. Let severity gate the priority, and use FMEA-MSR to evaluate diagnostics and safe-state responses. Use STPA or a similar top-down method to catch interaction and control hazards that a bottom-up worksheet will miss. And treat FMEA as one part of a coordinated set of dependability analyses rather than as a compliance document to be completed once.

3.10.3 Implications for standards and organizations

For standards bodies, the synthesis suggests value in guidance that connects FMEA explicitly to top-down hazard analysis and to security analysis, in software occurrence scales based on process and verification evidence rather than failure rates, and in data exchange formats that allow FMEA models to move between tools and between customers and suppliers. For organizations, adopting the advances of 2018 requires investment in modeling tools, in training that bridges reliability and software engineering, and in governance that ensures FMEA actions are verified and that operational data are fed back into the analysis.

4. CONCLUSION

The year 2018 marked a consolidation of FMEA for complex hardware-software systems. The third edition of IEC 60812 made the method generic, tailorable and open to alternatives to the RPN, with explicit software examples. The AIAG-VDA harmonization, reviewed throughout early 2018, introduced a model-like six-step structure, Action Priority and FMEA-MSR. ISO 26262:2018 added the semiconductor guidance needed for credible hardware failure mode and diagnostic coverage analysis. The STPA Handbook gave practitioners a mature complement for interaction failures. Research advanced model-based automation, software failure taxonomies and uncertainty-aware prioritization. None of these developments made FMEA a complete answer to the safety and reliability of software-intensive systems, and several important problems remained open. But together they transformed FMEA from a component worksheet into a model-connected, software-aware and better-prioritized analysis that can take its place within an integrated reliability and safety engineering process. Building on this synthesis, the paper proposed the IHS-FMEA conceptual model, which links normative inputs, a shared architecture model and evidence inputs to coordinated bottom-up and top-down analysis, severity-gated prioritization, monitoring and response evaluation, and verified treatment, with operational feedback that keeps the analysis current. The model and its five propositions offer practitioners a structured way to combine the 2018 advances and offer researchers a basis for empirical testing.

4.1 Limitations of the study

This study has several limitations. First, it is an integrative review rather than a systematic review with exhaustive database searches, so relevant studies may have been missed, particularly non-English publications and proprietary industry reports. Second, the restriction to sources published to date means that the review reflects the state of knowledge in that year and does not assess how the advances performed in later practice. Third, some primary documents, notably the AIAG-VDA review draft and parts of the ISO and IEC standards, are available only under licence, and parts of the synthesis rely on official summaries of their content. Fourth, the related literature drawn from the seed list is largely conceptual and comes from a limited set of venues, so its findings were used to illustrate cross-domain parallels rather than as strong empirical evidence. Fifth, the conceptual model has not been empirically validated; the illustrative application uses invented ratings and demonstrates coherence, not effectiveness. Finally, the examples are drawn mainly from automotive and energy contexts, and the model may require adaptation for domains such as medical devices or space systems with different regulatory regimes.

4.2 Future research directions

The synthesis points to the following directions for future research:

- **Occurrence for software and complex electronics.** No consensus emerged on how to rate occurrence for systematic software causes or for transient faults in complex semiconductors, and proxy-based ratings remain subjective.
- **Behavioral and temporal failure modes.** Structural models handle value failures well but represent timing, sequencing and state-dependent failures less naturally; executable models and simulation-based fault injection were promising but not yet mainstream.
- **Validation of fuzzy and MCDM methods.** Most published fuzzy and multi-criteria methods were demonstrated on single case studies, with limited evidence that they improve real-world outcomes compared with simpler severity-led rules, and they had limited uptake in industry standards.
- **Effort and scalability.** The AIAG-VDA method, STPA and model-based approaches each demand more upfront effort and skill than a traditional worksheet; organizations must invest in training and tools to benefit.
- **Machine learning and autonomy.** Neither ISO 26262:2018 nor classical FMEA addressed components whose behavior is learned from data. Failure modes of perception and decision algorithms, and the boundary between malfunction and functional insufficiency, were only beginning to be addressed through the emerging SOTIF work.
- **Interoperability.** Exchanging FMEA data between tools, and between customers and suppliers, remained difficult despite the database-oriented reporting described in IEC 60812:2018.
- **Empirical testing of the IHS-FMEA propositions.** Comparative case studies and controlled experiments are needed to test whether integrated, model-based and evidence-grounded FMEA improves completeness, ranking stability and safety outcomes relative to conventional practice.
- **Cross-domain transfer.** The parallels with risk-based inspection, condition monitoring and threat modeling suggest that rating scales, taxonomies and tools could be transferred between sectors; this transfer has not been studied systematically.

REFERENCES

Abdulkhaleq, A., & Wagner, S. (2015). A controlled experiment for the empirical evaluation of safety analysis techniques for safety-critical software. *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering (EASE)*. ACM.

Adaramola, T. S., Fadero, S., & Gideon, E. N. (2018). Predictive maintenance and condition monitoring in critical power and energy infrastructure. Iconic Research and Engineering Journals, 2(5), 454-477. https://doi.org/10.64388/IREV2I5-1722489

Adeyelu, O. O. (2018). A predictive compliance monitoring framework for detecting systemic safety risks through aviation consumer complaint data. Iconic Research and Engineering Journals, 1(8), 250-276. https://doi.org/10.64388/IREV1I8-1718478

Agboada, P., Ezeako, R., Okoli, I. E., & Uzondu, C. S. (2018). Digital transformation in operations management: A review of advances in automation, analytics, and process redesign. International Journal of Engineering Technology Research & Management, 2(11), 109-134.

Ahmed, K. S., & Odejobi, O. D. (2018a). Conceptual framework for scalable and secure cloud architectures for enterprise messaging. IRE Journals, 2(1), 1-15.

Ahmed, K. S., & Odejobi, O. D. (2018b). Resource allocation model for energy-efficient virtual machine placement in data centers. IRE Journals, 2(3), 1-10.

AIAG. (2017, December 8). New AIAG-VDA FMEA Handbook draft available for review and comments. Automotive Industry Action Group.

AIAG. (2018, January 18). There's still time! AIAG-VDA FMEA Handbook draft available through February 27, 2018. Automotive Industry Action Group.

Akeju, B., Edivri, J., Ogbole, J. I., Okoruwa, P. O., Fadayomi, O., & Abolaji, T. O. (2018). Conceptual model for insider threat classification and risk modeling in complex digital systems. Iconic Research and Engineering Journals, 1(9), 476-492. https://doi.org/10.64388/IREV1I9-1713778

Akomolafe, O., & Agu, M. U. (2018). A conceptual model for enhancing internal audit quality through technology-enabled risk assessment frameworks. Iconic Research and Engineering Journals, 1(9), 458-475.

Amayo, E. B. (2017). Multi algorithm of loss objective function of a single phase induction motor. International Journal of Development Research, 7(5), 12805-12810.

Amayo, E. B., & Popoola, T. T. (2017a). Multi algorithm of weight objective function of a single phase induction motor. International Journal of Trend in Research and Development, 4(2), 184-188.

Amayo, E. B., & Popoola, T. T. (2017b). Scientific study of telecommunication deployment in achieving quality network. International Journal of Trend in Research and Development, 4(5), 311-314.

Amayo, E. B., Ubeku, E., & Oshevire, P. (2015). Multi algorithm of a single objective function of a single phase induction motor. Journal of Multidisciplinary Engineering Science and Technology, 2(12), 3400-3403.

Arumosoye, O. M., & Obriki, O. D. (2018). Development of an integrated heat stress risk conceptual model for industrial operations in extreme environments. Iconic Research and Engineering Journals, 1(12), 141-160. https://doi.org/10.64388/IREV1I12-1714415

Atta, S., Asomani, E., & Adenuga, O. M. (2018). A systematic review of green corrosion inhibitors from agricultural extracts for mild steel protection in acidic and alkaline media. International

Journal of Engineering and Modern Technology, 4(1), 64-97. https://doi.org/10.56201/ijemt.vol.4.no1.2018.pg64.97

Azeez, L. O., & Badmus, O. B. (2018). Data-driven framework for predicting subsurface contamination pathways in complex remediation projects. Iconic Research and Engineering Journals, 2(5), 312-335.

Badmus, O., Dosunmu, A. A., & Ozowara, D. E. (2018). A systematic review of CI/CD pipeline strategies in Salesforce DevOps: Tools, practices, and deployment outcomes. Iconic Research and Engineering Journals, 2(6).

Dagodzo, D. (2018a). A conceptual framework for UAV integration into national power grid inspection programs. Iconic Research and Engineering Journals, 2(5), 391-412. https://doi.org/10.64388/IREV2I5-1716082

Dagodzo, D. (2018b). A review of UAV applications in electrical transmission line inspection: Methods, technologies, and challenges. Iconic Research and Engineering Journals, 2(6), 234-254. https://doi.org/10.64388/IREV2I6-1716083

Hecht, H., An, X., & Hecht, M. (2004). Computer aided software FMEA for unified modeling language based software. *Proceedings of the Annual Reliability and Maintainability Symposium (RAMS)*. IEEE.

Hummell, J. (2018, October 18). SysML and FMEA. *INCOSE Great Lakes Regional Conference 2018*. MBSE Solutions, LLC.

IEC. (2018). *IEC 60812:2018 Failure modes and effects analysis (FMEA and FMECA)* (3rd ed.). International Electrotechnical Commission.

IEEE. (2016). *IEEE Std 1633-2016, IEEE Recommended Practice on Software Reliability*. Institute of Electrical and Electronics Engineers.

ISO. (2018). *ISO 26262:2018 Road vehicles: Functional safety* (2nd ed., Parts 1 to 12). International Organization for Standardization.

Ladapo, O. O., Dosunmu, A. A., Jooda, D., & Abolaji, T. O. (2018). Lessons learned from offline assessment of security-critical systems: The case of Microsoft Active Directory. Iconic Research and Engineering Journals, 2(6), 277-299. https://doi.org/10.64388/IREV2I6-1717205

Lamidi, O. B. A., & Olamide, A. (2018). Spatially explicit risk modeling framework for tracking subsurface contaminant migration in data-limited remediation sites. Iconic Research and Engineering Journals, 2(6), 178-198.

Leveson, N. G., & Thomas, J. P. (2018). *STPA Handbook*. Massachusetts Institute of Technology.

Li, G., & Wang, B. (2011). SysML aided safety analysis for safety-critical systems. In *Artificial Intelligence and Computational Intelligence (AICI 2011)*, Lecture Notes in Computer Science, 7002, 270 to 275. Springer.

Liu, H.-C., et al. (2017). Failure mode and effect analysis using soft set theory and COPRAS method. *International Journal of Computational Intelligence Systems, 10*(1), 1002 to 1015.

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 to 838.

Mbonu, I. S., Aliliele, C., Iwuanyanwu, U., & Oluoha, O. M. (2018). A conceptual framework for legal and ethical risk modeling in enterprise data protection governance systems. Iconic Research and Engineering Journals, 2(2), 207-226. https://doi.org/10.64388/IREV2I2-1714911

Neufelder, A. M. (2014). *Effective Application of Software Failure Modes Effects Analysis*. Quanterion Solutions / SoftRel, LLC.

Nwangwu, C. V., Duvbiama-Owasanoye, O. A., & Onwumere, A. (2018). A framework for operational integrity in high-risk drilling: Balancing productivity and safety compliance. Iconic Research and Engineering Journals, 2(5), 436-453.

Obetta, P. C. (2018). A risk-based inspection concept for fitness-for-service assessment of ageing pressure vessels on offshore joint venture facilities. Iconic Research and Engineering Journals, 2(5), 497-520. https://doi.org/10.64388/IREV2I5-1722578

Obetta, P. C., & Ninduwezuor-Ehiobu, N. (2018). A condition-based maintenance concept for CALM buoy bearings and swivels to improve offshore crude export availability. Iconic Research and Engineering Journals, 2(6), 385-410. https://doi.org/10.64388/IREV2I6-1722579

Obriki, O. D., & Arumosoye, O. M. (2018). Conceptual modeling of data-driven occupational safety risk control in large-scale energy infrastructure projects. Iconic Research and Engineering Journals, 1(7), 169-189. https://doi.org/10.64388/IREV1I7-1714414

Odejobi, O. D., & Ahmed, K. S. (2018b). Performance evaluation model for multi-tenant Microsoft 365 deployments under high concurrency. IRE Journals, 1(11), 92-107.

Okonkwo, C. S., Ogunwole, O., & Okeke, O. T. (2018a). Framework for strategic procurement optimization in oil and gas operations. Iconic Research and Engineering Journals, 1(7), 153-168. https://doi.org/10.64388/IREV1I7-1713119

Okonkwo, C. S., Ogunwole, O., & Okeke, O. T. (2018b). Model for inventory availability and plant uptime improvement in energy facilities. Iconic Research and Engineering Journals, 2(4), 160-172. https://doi.org/10.64388/IREV2I4-1713120

Operational risk analysis with fuzzy FMEA (Failure Mode and Effect Analysis) approach (Case study: Optimus Creative Bandung). (2018). *MATEC Web of Conferences, 154*, 01084.

Oshevire, P., Eyenubo, O. J., & Amayo, B. (2017). Voltage control in the presence of distributed generation. ATBU Journal of Science, Technology and Education, 5(2), 165-173.

Rabbi, M. F. (2018). Assessment of fuzzy failure mode and effect analysis (FMEA) for reach stacker crane (RST): A case study. *International Journal of Research in Industrial Engineering, 7*(3), 336 to 348.

Scippacercola, F., Pietrantuono, R., Russo, S., Asper, A., & Silva, N. (2016). Integrating FMEA in a model driven methodology. *ESA SP-736, Proceedings of DASIA 2016*. European Space Agency.

Sotomayor Martínez, R. (2015). [Comparison of STPA and FMEA through an automotive case study] (Master's thesis, System Design and Management Program). Massachusetts Institute of Technology. http://hdl.handle.net/1721.1/105318

Sulaman, S. M., Beer, A., Felderer, M., & Höst, M. (2017). Comparison of the FMEA and STPA safety analysis methods: A case study. *Software Quality Journal*. Advance online publication. https://doi.org/10.1007/s11219-017-9396-0

The idea of fuzzy logic usage in a sheet-based FMEA analysis of mechanical systems. (2018). *MATEC Web of Conferences, 183*, 03009. https://doi.org/10.1051/matecconf/201818303009

Using fuzzy FMEA approach to improve decision-making process. (2018). *International Journal of Industrial Engineering & Production Research, 29*(3).