Volume 11, Issue 5, May – 2026 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

Systemic Uncertainty Engineering (SUE): A Quantitative Framework for Risk Reduction in Complex Socio-Technical Systems

Jherrod Thomas[1]

[1]Certified Functional Safety Expert Jherrod
The Lion of Functional Safety™

Publication Date: 2026/05/19

Abstract: Modern safety-critical systems fail at the boundary between engineered products and their operational environments rather than from isolated component faults. Existing domain-specific standards for functional safety, performance sufficiency, and cybersecurity each address one slice of this boundary but provide no unified method for measuring how uncertainty propagates across their combined scope, nor any instrument for identifying failure combinations that span multiple domains simultaneously. This paper proposes Systemic Uncertainty Engineering, a quantitative framework that treats uncertainty as a measurable, propagating system property and expresses residual systemic risk as expected financial loss. The framework was constructed through theoretical development and retrospective empirical validation. A four-quadrant uncertainty model decomposed uncertainty along reducibility and origin axes, establishing the measurement structure for a lifecycle-spanning propagation model with linear and nonlinear interaction terms. A dual-process model ordered analytical activities from product-environment interface characterization through risk assessment, goal architecture, and economic translation. The framework was instantiated for autonomous vehicle development and validated against six documented failures spanning five decades of automotive engineering history. Three findings emerged that existing single-domain methods cannot produce. Cross-domain minimal cut sets spanning functional safety, performance sufficiency, cybersecurity, and organizational domains were identified before domain decomposition occurred. An 80cell risk tensor quantified residual uncertainty across all domains simultaneously and translated it into expected financial loss and return-on-investment metrics. Retrospective analysis confirmed that the constructs would have identified each failure's dominant risk pathway before deployment in all six cases. The framework demonstrates applicability to five additional technology domains sharing the structural conditions of novelty, open-world operation, and multi-domain regulatory oversight.

Keywords: *Systemic Uncertainty Engineering, Safety Critical Systems, Autonomous Vehicles, Cross-Domain Risk Analysis, Product-Environment Interface, Uncertainty Propagation, Risk Quantification, Functional Safety Integration.*

How to Cite: Jherrod Thomas (2026) Systemic Uncertainty Engineering (SUE): A Quantitative Framework for Risk Reduction in Complex Socio-Technical Systems. *International Journal of Innovative Science and Research Technology*, 11(5), 506-582. https://doi.org/10.38124/ijisrt/26may044

I. INTRODUCTION

Engineering risk in complex socio-technical systems has outgrown the frameworks designed to manage it. The standards in current practice, including functional safety, cybersecurity engineering, performance validation, and process maturity assessment, were each built for environments where failure modes are enumerable and historical data supports statistical inference. Autonomous vehicles, AI-enabled medical devices, and advanced robotics do not share those conditions. They operate at the boundary between what the product was designed to do and what an open, evolving environment will actually demand, and it is at that boundary that their most consequential failures occur. This section establishes the problem (Section I-A), diagnoses why existing frameworks cannot close it (Section I-B), develops the disciplinary analogy that motivates a new approach (Section I-C), and states the seven contributions this paper makes (Section I-D).

A. The Problem: Systemic Uncertainty in Novel Environments

Engineering failures in modern complex systems have a structural character that classical reliability engineering was not designed to address. They do not originate from a single faulty component, degraded hardware, or isolated software errors. They arise from the compound interaction of technical design choices, environmental conditions, organizational decisions, and human actions operating simultaneously across a system's lifecycle. The class of systems that now defines the frontier of technology, including autonomous vehicles (AVs),

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

AI-enabled medical devices, and advanced robotics, produces this kind of failure with regularity. It demands an analytical framework matched to that reality.

Three observations motivate the framework proposed in this paper. Failures in modern systems are systemic rather than elemental. AVs present the most concentrated instance of this challenge in current engineering practice. And the failures that carry the greatest consequence do not reside within the product or within the environment in isolation, but at the boundary where they meet. Each observation is developed in the subsections that follow.

➢ *Modern Failures Are Systemic, Not Elemental:*

The empirical record of high-consequence engineering failures over the past three decades consistently reveals one pattern: catastrophic outcomes arise from the confluence of conditions rather than from the breakdown of a single identifiable part. The Takata airbag crisis, the Toyota unintended acceleration events, and the General Motors ignition switch defect each resulted in fatalities, large-scale recalls, and billions of dollars in liability. None originated from a single component failure. Each emerged from the unmanaged interaction of design assumptions, organizational decisions, supplier relationships, software behavior, and verification gaps that compounded across years of product operation [1], [2].

This pattern extends across industries and system types. Venkatasubramanian and colleagues, analyzing major industrial disasters across multiple sectors, identified a common structural signature: failures at the equipment level, the human-operator level, and the institutional level reinforced one another in ways that no singledomain analysis could have exposed [3], [4]. Complex socio-technical systems comprise many interconnected components with nonlinear interactions. Those interactions produce emergent behavior whose hazardous expression is qualitatively greater than the sum of the individual failure contributions.

This finding carries a precise implication for engineering methodology. Risk analysis frameworks built on the enumeration of component failure modes, including fault trees, failure mode and effects analysis (FMEA), and probabilistic risk assessment (PRA), systematically undercount the failure space of modern systems [1], [5]. Each such method assumes that hazardous states can be identified by tracing backward from failure events through a fixed set of known causal pathways. In genuinely novel environments, many of those pathways are unknown at design time. The accidents that cause the greatest harm are precisely those that were absent from the fault tree.

➢ *The Autonomous Vehicle Exemplar:*

Autonomous vehicle development concentrates the problem of systemic uncertainty into a form that is both unusually acute and analytically instructive. Three structural features of the AV domain make it the primary exemplar for this paper.

First, the operational environment is fundamentally open-ended. An AV must navigate a scenario space that cannot be fully enumerated at design time. The possible combinations of road geometry, traffic density, weather conditions, pedestrian behavior, and infrastructure state are effectively unbounded. Algorithms that perform well within their training distribution encounter conditions for which their behavior is undefined, and the performance gap at these boundaries constitutes a primary source of system risk [6], [7]. This is not a deficiency that improved sensor resolution or larger training datasets can fully eliminate. It is a structural property of operating in a world that evolves independently of the engineering team's assumptions.

Second, the AV system architecture creates tightly coupled feedback chains. Perception systems classify the environment and pass outputs to planning; planning generates trajectories and passes commands to control; control actuates the vehicle and alters the physical state that perception must next interpret. Errors at any stage propagate downstream with propagation coefficients that depend on the design of each interface. Approximately 30% of reported disengagements in early AV deployments were attributable solely to perception discrepancies, before accounting for how those errors interacted with downstream planning logic [8]. Deng and colleagues confirm that in AV systems, safety problems frequently arise from interactions among multiple components, even when each component performs exactly as designed [9].

Third, organizational decisions carry safety consequences that no component-level specification can contain. Choices about operational design domain (ODD) scope, software update scheduling, safety driver monitoring protocols, and incident reporting culture each shape the conditions under which technical uncertainties manifest as hazardous outcomes. The National Transportation Safety Board's investigation of the Uber Advanced Technology Group (ATG) pedestrian fatality in Tempe, Arizona in March 2018 concluded that no single technical fault caused the accident [10]. The outcome arose from the concurrent presence of perception system limitations, a decision-algorithm classification ambiguity, inadequate monitoring of the safety driver, and organizational tolerance of elevated risk. Each factor was individually documented; together, they produced a fatal result.

➢ *Failures Live at the Interface:*

The observation that unifies the preceding analysis is locational: the failures that matter most in novel socio-technical systems arise at the boundary between the engineered product and its operational environment. This boundary is not a physical location. It is an analytical object, specifically the set of conditions under which the product's intended behavior and the environment's actual demands diverge [11].

ISO 21448, the Safety of the Intended Functionality (SOTIF) standard, implicitly acknowledges this. It addresses hazardous behavior arising from functional insufficiency in a system operating exactly as designed, and it explicitly

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

distinguishes this class of risk from the hardware malfunction and systematic failure addressed by ISO 26262 [12], [13]. A perception system that fails to detect a pedestrian crossing at an unusual angle is not malfunctioning in the ISO 26262 sense. Its intended function is insufficient for the environmental conditions encountered. The risk lives at the product-environment boundary.

The engineering consequence follows directly. Component reliability analysis, however thorough, cannot characterize boundary risk because the relevant failure modes are not component failures. They are interface failures: situations where the product's design assumptions about what the environment will provide turn out to be incorrect. These assumptions are rarely stated explicitly, rarely measured, and rarely tested systematically. No existing standard specifies a process for mapping them.

This gap is what Systemic Uncertainty Engineering (SUE) addresses (see Appendix A). Its foundational construct, the System-Environment Interface (SEI), treats the product-environment boundary as the primary engineering artifact from which all subsequent analysis derives. Section I-B establishes why no existing framework fills this gap, individually or in combination, and Section I-C develops the disciplinary logic that SUE follows in closing it.

B. Why Existing Frameworks are Insufficient

Each major framework in current safety and risk engineering practice was built for a context that novel socio-technical systems do not share. The defining condition of that context is a bounded failure space: one in which the system's hazardous states can be identified in advance, historical data provides failure-rate estimates, and compliance with a defined process reliably reduces residual risk to an acceptable level. When any of these conditions is absent, the framework continues to produce outputs, but those outputs no longer characterize the system's actual risk profile.

Table 1 Summarizes the six primary frameworks considered here, together with the specific structural feature that limits each one when applied to novel environments. The subsections that follow explain each row in sufficient depth to establish the common pattern: each framework addresses its intended scope well while leaving the same four capabilities unaddressed across the board.

- *Six Sigma: Requires Stable, Observable Defects:*

Six Sigma introduced a universal quality discipline for manufacturing by treating process variation as a measurable, controllable property and expressing its magnitude as defects per million opportunities (DPMO) [14]. At six sigma performance, a process produces no more than 3.4 DPMO. This target was achievable in manufacturing because three structural conditions held: the process repeated in a statistically stable fashion; defects were observable and countable at defined inspection points; and production volumes were large enough to estimate failure rates reliably.

Novel socio-technical systems violate all three conditions. An AV navigating an open-world environment does not execute a repeatable process in the manufacturing sense. The distribution of scenarios it encounters is non-stationary: it shifts as road conditions, traffic patterns, and pedestrian behaviors evolve. The relevant failure events, specifically incorrect system responses to conditions outside the training distribution, are not reliably observable at design time because the triggering conditions may not yet have occurred. Applying Six Sigma directly requires stable base rates to calculate DPMO, but for genuinely novel scenario types, those rates are undefined. The methodology's power derives from the very conditions that novel-environment systems cannot satisfy.

- *Reliability Engineering and PRA: Requires Known Failure Modes:*

Probabilistic risk assessment (PRA) and classical reliability engineering provide quantitative tools for systems in which the failure mode space is enumerable and historical failure-rate data are available [15]. Fault tree analysis (FTA) constructs a deductive model of how component failures combine to produce toplevel hazardous events, assigning probabilities to each branch based on component failure rates from reliability databases. When the failure space is bounded and the data exist, the method yields actionable risk estimates.

The structural limitation surfaces when the failure space is open-ended. For AV systems, software does not fail according to wear-out distributions, perception systems encounter scenarios never represented in training data, and organizational decisions create failure preconditions that component-level models cannot capture [11]. The failure modes that lead to catastrophic outcomes in novel systems are often those that have not previously occurred at measurable frequency. Reliability engineering characterizes what is already known; it cannot characterize the unknown-unknown territory that defines the dominant risk profile of novel-environment systems.

- *ISO 26262: Process Compliance Is Not Risk Quantification:*

ISO 26262 provides the most rigorous automotive functional safety process standard available, specifying hazard analysis and risk assessment (HARA), Automotive Safety Integrity Level (ASIL) decomposition, functional safety concept development, and systematic verification and validation across the development lifecycle [13]. Organizations that comply with it demonstrably reduce the likelihood that systematic failures and random hardware failures lead to safety violations.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

Table 1 Gaps in Existing Safety and Quality Frameworks for Novel Socio-Technical Environments

Approach	Primary Strength	Gap for Novel Environments
Six Sigma	Universal quality metric (DPMO); structured process improvement	Requires stable, repeatable processes; observable and countable defects; large statistical samples
Reliability / PRA	Probabilistic system modeling; fault tree quantification	Requires enumerable failure modes and historical failure-rate data
ISO 26262	Rigorous functional safety process; ASIL decomposition	Process compliance does not quantify residual uncertainty; scope excludes performance insufficiency
SOTIF (ISO 21448)	Explicit treatment of unknown hazardous scenarios	No quantitative propagation mechanics; no systematic product-environment interface analysis
ISO/SAE 21434	Structured cybersecurity lifecycle; TARA methodology	Fully siloed from safety and performance analysis; no cross-domain cut set identification
STAMP / STPA	Systemic control-theoretic perspective on accident causation	Qualitative only; no quantitative risk metric; no economic translation

The limitation is one of scope and metric type. ISO 26262 is a process standard: it prescribes which activities must be performed and what evidence must be produced, but it does not provide a single quantitative measure of the residual uncertainty remaining after those activities are complete [16]. Two programs can achieve full ASIL D compliance while carrying very different levels of actual systemic risk, because compliance measures process execution rather than epistemic state. This extends to a scope exclusion: the standard addresses malfunctions of electrical and electronic systems and does not cover hazardous behavior arising from performance insufficiency in a system operating exactly as designed. That class of risk is what ISO 21448 subsequently addressed.

- *SOTIF (ISO 21448): Scenario Coverage without Propagation Mechanics:*

ISO 21448, the Safety of the Intended Functionality (SOTIF) standard, acknowledges the class of risk that ISO 26262 does not address: hazardous behavior arising from functional insufficiency rather than malfunction [12]. A perception system that performs exactly as designed but fails to detect a pedestrian under atypical lighting conditions is not malfunctioning in the ISO 26262 sense. SOTIF identifies this as a distinct hazard class and provides a framework for partitioning the operational scenario space into known and unknown regions.

Two structural limitations prevent SOTIF from closing the gap this paper addresses. First, the standard does not provide quantitative propagation mechanics. It identifies scenarios and evaluates their hazardous potential, but it does not model how insufficiency at the perception layer propagates through planning and actuation to produce downstream losses, nor does it provide a unified metric that expresses the magnitude of the residual coverage gap [11]. Second, SOTIF treats scenarios as the primary object of analysis and does not specify a systematic process for first mapping the product-environment interface from which those scenarios are derived. The foundational step of identifying every point at which product behavior meets environmental demand is left unspecified.

- *ISO/SAE 21434: Cybersecurity Lifecycle Siloed from Safety:*

ISO/SAE 21434 provides a structured cybersecurity engineering lifecycle for road vehicles, including Threat Analysis and Risk Assessment (TARA) methodology, Cybersecurity Assurance Level (CAL) determination, and lifecycle management of cybersecurity obligations [17]. Dobaj and colleagues confirm that the standard provides necessary structure for managing the growing cybersecurity risk surface in modern vehicles, where millions of lines of software and multiple wireless interfaces create an extensive attack surface [18].

The critical architectural limitation is that the standard operates in a silo. Its TARA produces cybersecurityspecific cut sets without any mechanism to identify how those attack paths interact with functional safety goals or SOTIF-relevant performance insufficiencies. As Yu and colleagues note, ISO 26262 discusses the interaction of functional safety with cybersecurity only in general terms, without providing concrete integrated workflows [19]. A cybersecurity intrusion that degrades a safety-critical perception subsystem may simultaneously constitute a SOTIF event and an ISO 26262 safety goal violation. No existing standard provides a shared analysis artifact that would expose this cross-domain cut set before it manifests in operation.

- *STAMP and STPA: Systemic but Qualitative:*

Systems-Theoretic Accident Model and Processes (STAMP) and the associated System-Theoretic Process Analysis (STPA) technique represent the most conceptually sophisticated safety framework available for complex systems. Leveson's STAMP model reframes accidents as failures of safety constraints rather than chains of component failures, capturing organizational and software factors that event-chain models miss [1]. Comparative analyses confirm that STAMP-based methods identify a broader and more actionable set of contributing factors than event-chain approaches such as AcciMap and the Functional Resonance Analysis Method (FRAM) [20].

The gap is the absence of quantification and economic translation. STPA identifies unsafe control actions and causal scenarios; it does not assign probabilities to them, does not produce a unified risk metric, and does not express identified

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

risks as expected financial loss. For an engineering organization allocating resources across competing safety priorities, qualitative identification of causal scenarios is necessary but not sufficient. Without a quantitative metric, the analysis cannot drive prioritization, track risk reduction over time, or produce the return-on-investment calculations that organizational decision-makers require.

➢ *Synthesis: Four Capabilities No Framework Provides:*

Reading Table I across all six rows reveals a consistent structural pattern. Each framework achieves local effectiveness within its intended domain while leaving the same four capabilities unaddressed. No existing framework, individually or in combination, provides:

- *Systematic product-environment interface analysis as a first-class engineering artifact.* No standard specifies a process for mapping every point at which product behavior meets environmental demand, declaring the assumptions embedded in each interface, and generating test cases from those declarations. SOTIF approaches this implicitly through scenario analysis, but does not formalize the interface structure that generates those scenarios.
- *A unified quantitative risk metric spanning all domains simultaneously.* ISO 26262 produces ASIL assignments. SOTIF produces scenario coverage assessments. ISO/SAE 21434 produces CAL assignments. STAMP produces causal scenario lists. None produces a single metric that integrates across all domains, enabling cross-domain risk comparison and portfolio-level investment decisions.
- *Cross-domain cut set identification.* The failure modes that produce catastrophic outcomes in novel systems are precisely those that span domain boundaries. A cybersecurity intrusion that degrades a safety-critical perception function, or an organizational process gap that allows a SOTIF-relevant scenario to remain untested, constitutes a crossdomain cut set. No existing framework provides a systematic method for exposing these cut sets before they produce harm.
- *Economic translation of systemic risk.* None of the six frameworks expresses risk as expected financial loss. Safety compliance is framed as a cost rather than as a quantified risk reduction with a calculable return on investment. This framing structurally disadvantages safety programs competing for resources against other organizational priorities.

These four absent capabilities are not peripheral. They represent the analytical requirements for managing uncertainty in systems that operate at the boundary between an engineered product and a novel, evolving environment. The remainder of this paper develops a framework that provides all four.

C. The Six Sigma Analogy

The gap documented in Section I-B is not the first time engineering practice has encountered a class of system properties that mattered greatly but lacked a universal way to measure them. Six Sigma's rise from a Motorola internal initiative to a cross-industry engineering discipline offers the closest structural precedent for the move this paper proposes. The analogy is deliberate and precise, not rhetorical.

➢ *Six Sigma: Variation Made Measurable and Controllable:*

Before Six Sigma, manufacturing quality was managed through inspection regimes, rework loops, and domain-specific best practices. Each production area could reduce its own defect rate using locally effective methods. No common metric, however, expressed quality performance across different processes, product lines, or organizations in a form that enabled crosssystem comparison and investment prioritization. Quality improvement was a local activity without a universal language.

Bill Smith's development of the Six Sigma methodology at Motorola in 1986 introduced that language [14], [21]. The conceptually central move was to treat process *variation* as a measurable, controllable system property and to express its magnitude as defects per million opportunities (DPMO), defining a target of 3.4 DPMO at six sigma performance [22]. Motorola reduced defects on semiconductor devices by an estimated 94% between 1987 and 1993 by making variation visible, attributable to specific process contributors, and subject to structured improvement [21]. General Electric subsequently applied the methodology company-wide under Jack Welch in the mid-1990s, and the discipline spread to services, healthcare, and logistics [23], [24].

Three structural features of Six Sigma explain its transformative impact, and they are worth stating explicitly because they define what any analogous framework must provide. First, variation was defined as a property of the entire production system rather than of isolated components, making the metric attributable across all contributing process stages. Second, DPMO could be calculated from observable production data, giving engineers a concrete measurement target rather than a qualitative aspiration. Third, the methodology directly linked measurement to investment priority: a higher DPMO in a specific subprocess identified an improvement intervention with a calculable financial return. Safety and quality became economically visible rather than merely technically characterizable [25].

➢ *The Structural Parallel: Uncertainty as the Target Property:*

The situation facing novel socio-technical system engineering today is structurally parallel to preSix Sigma manufacturing quality. Safety, cybersecurity, performance sufficiency, and organizational process disciplines each manage risk within their own domain using domain-specific metrics. ISO 26262 assigns Automotive Safety Integrity Levels (ASILs). SOTIF produces scenario coverage assessments. ISO/SAE 21434 produces Cybersecurity Assurance Level (CAL) ratings. Automotive SPICE produces process capability scores. Each discipline achieves local improvement, but the systemic risk arising from their interactions remains without a common metric, a propagation model, or an economic expression.

SUE proposes the same structural move that Six Sigma made for manufacturing variation: treat *uncertainty* as a measurable, propagating system property, quantify how it compounds across the product's lifecycle layers and across the boundary between the product and its environment, and

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

express the result as a single metric with a direct economic interpretation. The epistemic/aleatoric distinction, well established in quantitative risk analysis, provides the formal underpinning for this treatment [26], [27]. Epistemic uncertainty arises from incomplete knowledge and is reducible through better engineering. Aleatoric uncertainty arises from inherent variability and must be managed through robustness design and operational domain constraints [28]. Both types are measurable in principle. What has been absent is a unified framework that tracks both types together, models how they interact as they propagate across a layered socio-technical system, and expresses the aggregate as expected financial loss.

The SUE metric that performs this function is the Systemic Risk Density (SRD), formalized through the SUE Risk Cube: an 80-cell tensor across five lifecycle layers, four uncertainty domains, and four uncertainty types, with each cell carrying an independently measurable uncertainty score, hazard manifestation probability, and propagation weight (developed fully in Sections IV and V). Where Six Sigma measures DPMO in stable production environments with observable defects, SRD measures systemic uncertainty in novel environments where the dominant risk lives at the product-environment interface rather than in component failure rates. The unit of account differs; the discipline-building logic is identical.

➢ *The Key Move: From Compliance to Quantitative Control:*

The analogy points to a specific shift in engineering posture, not merely to a change in metric choice. Six Sigma did not replace existing quality practices. A factory running statistical process control, FMEA, and process audits continued to run all of those activities under Six Sigma. What changed was the addition of a quantitative layer above those activities that expressed how well they were collectively reducing variation across the full production system. The aggregate metric made local improvements visible at the system level and made cross-process investment trade-offs tractable.

SUE makes the same additive move for systemic uncertainty. ISO 26262, SOTIF, ISO/SAE 21434, and Automotive SPICE remain the applicable domain standards. Organizations continue to perform HARA, TARA, scenario analysis, and process capability assessments. SUE adds the analytical layer those activities currently lack: a framework that maps the product-environment interface before any domain analysis begins (the SystemEnvironment Interface step), integrates the outputs of all domain analyses into a unified uncertainty propagation model, and expresses the result as a tensor metric with economic interpretation via Expected System Loss (ESL) and Return on Safety Investment (ROSI). The relationship mirrors that between ISO 9001 and Six Sigma: the process standard defines the activities that must be performed, while the quantitative framework measures how well those activities reduce the target system property [14], [21] (see Appendix A for formal definitions of SRD, ESL, and ROSI).

This reorientation, from satisfying domain checklists to quantitatively controlling uncertainty at the product environment boundary, is the defining structural contribution of SUE as a discipline. It does not reduce to better implementation of existing methods. It requires treating the product-environment interface as a firstclass engineering artifact and uncertainty as a first-class system property. Section I-D states the seven specific contributions through which this reorientation is made operational.

D. Contributions

The four missing capabilities identified in Section I-B define what a new framework must provide: systematic product-environment interface analysis, a unified quantitative risk metric spanning all domains, crossdomain cut set identification, and economic translation of systemic risk. The seven contributions of this paper address these requirements directly. Each contribution targets a specific analytical gap; together they constitute a coherent engineering discipline rather than a collection of independent tools.

➢ *Contribution 1: The Four-Quadrant Uncertainty Model.:*

This paper introduces a formal decomposition of uncertainty at each node in a socio-technical system graph along two independent axes: epistemic versus aleatoric, and endogenous versus exogenous. Epistemic uncertainty arises from incomplete knowledge and is reducible through better engineering; aleatoric uncertainty arises from inherent variability and must be managed through robustness design and operational constraints [26], [27]. The endogenous dimension captures uncertainty internal to the product, including requirements ambiguity, verification gaps, and process immaturity. The exogenous dimension captures uncertainty arising from the operational environment, including gaps in scenario coverage, stochastic agent behavior, and adversarial threats [28].

This decomposition matters because each quadrant requires a fundamentally different intervention strategy. Endogenous epistemic uncertainty responds to greater verification and formal methods. Exogenous-epistemic uncertainty is addressed through broader scenario coverage and expanded operational design domain testing. Aleatoric uncertainties in both dimensions require robustness margins and runtime monitoring. A framework that conflates these categories cannot drive targeted intervention because the appropriate engineering response differs in each case.

➢ *Contribution 2: The Uncertainty Diamond.:*

This paper introduces the Uncertainty Diamond: a dual-V process model that explicitly represents both product-focused development activities on the left and environment-focused validation activities on the right, converging at the Unified Loss Goal (ULG). The left V covers the endogenous axis from requirements specification through unit testing to system integration. The right V covers the exogenous axis from operational design domain definition through scenario generation, simulation, and field validation. Both arms converge at the ULG, the point where product capability meets environmental demand.

The structural advance over the conventional V-model is that the Uncertainty Diamond makes the product environment

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

boundary an explicit engineering artifact. Existing safety processes follow a left-V logic exclusively, treating environmental assumptions as fixed inputs rather than as engineering objects subject to systematic analysis [11]. The right arm of the Uncertainty Diamond provides the engineering activities that address exogenous uncertainty in the same systematic fashion that the left arm addresses endogenous uncertainty. This symmetry is what the gap analysis in Section I-B identified as absent from all existing frameworks.

- *Contribution 3: System-Environment Interface Analysis:*

This paper introduces the System Environment Interface (SEI) as the primary first-step engineering artifact from which all subsequent analysis derives. The SEI is a systematic map of every point at which product behavior meets environmental demand, with explicit documentation of: the assumption the product makes about what the environment will provide; the gap between that assumption and what the environment may actually present; and a set of test cases generated directly from that gap. In the automotive instantiation, the SEI specializes in the Product Environment Interface (PEI), where each interface point corresponds to a specific sensor-environment interaction or human-machine boundary [11], [12].

The SEI's function is to make the interface structure explicit before any domain-specific hazard analysis begins. Current approaches, including SOTIF scenario analysis and STPA control structure modeling, presuppose that the relevant interfaces are already understood. The SEI formalizes the step of identifying and characterizing those interfaces, ensuring that downstream analyses cover the full space of product-environment interactions rather than only those that engineering teams had already anticipated.

- *Contribution 4: Parametric Loss Analysis and Unified Threat and Hazard Analysis:*

This paper introduces Parametric Loss Analysis (PLA) as a unified cross-domain risk assessment process that generates safety hazards, cybersecurity threats, and performance insufficiencies from a single structured procedure. In prior practice, Hazard Analysis and Risk Assessment (HARA) per ISO 26262, Threat Analysis and Risk Assessment (TARA) per ISO/SAE 21434, and SOTIF scenario analysis are conducted as separate activities by separate domain teams, producing results that cannot be directly compared or combined [12], [13], [17]. PLA derives all three simultaneously from the SEI output, assigning SUE Levels to each identified risk item based on the system layer, uncertainty domain, and uncertainty type of the contributing nodes.

In the automotive domain, PLA specializes in the Unified Threat and Hazard Analysis (UTHA), which subsumes HARA, TARA, and SOTIF analyses into a single process. Each item in the UTHA receives an ASIL assignment under ISO 26262, a CAL assignment under ISO/SAE 21434, or a SOTIF Criterion classification under ISO 21448, as appropriate for its domain, while simultaneously receiving a SUE Level reflecting its position in the four-quadrant model. Domain-specific compliance artifacts are preserved and generated as outputs of a single unified process.

- *Contribution 5: Unified Loss Goal Architecture, Safe Maneuver Objective, and Combined Domain Trees:*

This paper introduces the Unified Loss Goal (ULG) as a domain-agnostic parent goal that sits above the safety, cybersecurity, and performance goals of all domain standards simultaneously. The ULG expresses the system-level loss prevention objective in terms neutral with respect to the domain analysis that will follow, enabling safety goals, cybersecurity goals, and SOTIF criteria to be decomposed into child goals under a single statement. Goal-oriented approaches to co-engineering safety and security across domain boundaries have been shown to break down barriers between disciplines and to provide consistent risk-based rationales that connect top-level objectives to domain-specific measures [29].

The Safe Maneuver Objective (SMO) is the operational target state that the system must reach when a ULG violation is detected or imminent. The SMO is domain-agnostic and continuously evaluated, resolving the contradictory safe-state problem that arises when functional safety, SOTIF, and cybersecurity each define different response objectives for the same triggering event. Combined domain trees place a single OR gate under the ULG with four native-method branches: fault tree analysis per ISO 26262, attack trees per ISO/SAE 21434, insufficiency trees per ISO 21448, and an organizational branch covering process and human factor contributors [19]. These combined trees enable the identification of cross-domain cut sets that span two or more branches, a failure mode class that single-domain methods cannot detect.

- *Contribution 6: The SUE Risk Cube and Economic Translation:*

This paper introduces the SUE Risk Cube: an 80-cell tensor across five system lifecycle layers (Requirements, Design, Implementation, Process, and Toolchain), four uncertainty domains (Safety, Performance, Security, and Organizational), and four uncertainty types (Epistemic-Endogenous, Epistemic Exogenous, Aleatoric-Endogenous, and Aleatoric Exogenous). Each cell carries an independently measurable uncertainty score, a hazard manifestation probability, and a propagation weight. The product of these three values, aggregated with the cell's SUE Level weight, yields the Systemic Risk Density (SRD) contribution of that cell. SUE Level classifications (1 through 4, from critical to low) are pre-computed for each cell based on the combined weights of its three axis positions.

Economic translation is provided through two derived metrics: Expected System Loss (ESL), computed as the product of SRD, cost per loss event, and exposure for each domain; and Return on Safety Investment (ROSI), computed as the ratio of ESL reduction to intervention cost. Collier and colleagues demonstrate that ratio based metrics, such as ROSI, are the most appropriate guides for prioritizing competing security and safety investments when budgetary constraints apply [30]. Costeffective allocation of safety resources requires an explicit calculation of what risk reduction is purchased per unit of investment, a calculation the SRD-ESL-ROSI chain makes tractable for the first time in a unified cross domain model [31].

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

➢ *Contribution 7: Automotive Instantiation and Demonstrated Generalizability:*

This paper develops a complete automotive instantiation of the SUE framework in which the generic constructs specialize to domain specific artifacts while preserving full compatibility with existing standards. The SEI becomes the PEI; PLA becomes UTHA; the ULG becomes the Unified Vehicle Goal (UVG); and the combined domain tree branches correspond directly to ISO 26262 FTA, ISO/SAE 21434 attack trees, and ISO 21448 insufficiency trees. Three worked automotive case studies, covering the Uber ATG Tempe fatality, the Takata airbag crisis, and the GM ignition switch defect, demonstrate retrospective application of the framework and confirm that cross-domain cut sets of the type the framework targets were present in each case but were not identified by the siloed analyses conducted at the time [1], [10].

This paper also demonstrates generalizability across six additional application domains where the same novel socio-technical conditions prevail: electric vertical takeoff and landing (eVTOL) vehicles, autonomous surgical robotics, small modular nuclear reactors, autonomous maritime vessels, collaborative robotics in unstructured environments, and AI-enabled defense systems. Each domain exhibits an open-ended product-environment boundary, a combination of safety, cybersecurity, and performance risk domains, and a regulatory landscape that does not yet provide a unified quantitative uncertainty metric. The framework's generalizability is not claimed by analogy but demonstrated through domainspecific PEI structure, combined tree-branch configuration, and SUE Level distribution analyses for each case.

The seven contributions together constitute a response to the analytical gap identified in Section I-A as structurally unavoidable in novel socio-technical system engineering. The remainder of this paper develops each contribution formally, provides the mathematical foundation for uncertainty propagation and the SUE Risk Cube, and validates the framework through the worked case studies. Section II situates the framework within the broader literature on safety, quality, and risk engineering.

II. RELATED WORK

The framework proposed in this paper is situated within six established disciplines for quality, safety, and risk engineering. Table I in Section I-B provides a structured summary of each discipline's capabilities and the gap it leaves when applied to novel socio-technical systems. The present section develops that summary in full, providing the historical context, technical scope, demonstrated applications, and structural limitations of each approach. The section closes with a synthesis that makes explicit why no framework, individually or in combination, addresses all four missing capabilities identified in Section I-B7.

➢ *Six Sigma and Statistical Quality Control*

Six Sigma represents a landmark in quantifying industrial quality. Bill Smith developed the methodology at Motorola in 1986 in response to chronic defect rates in semiconductor manufacturing, and the approach was adopted company-wide under CEO Bob Galvin [21]. The core technical move was to express process performance as defects per million opportunities (DPMO), target a value of 3.4 DPMO corresponding to six standard deviations between the process mean and the nearest specification limit, and drive toward that target through the Define-Measure-Analyze-Improve-Control (DMAIC) structured improvement cycle [14], [24]. Motorola reported a 94% reduction in semiconductor device defects between 1987 and 1993. General Electric subsequently applied the methodology company-wide in the mid1990s, and the discipline spread to services, healthcare, logistics, and aerospace [21], [32].

The technical power of Six Sigma rests on three enabling conditions. First, the defect-producing process repeats in a statistically stable manner, allowing variance to be tractable through standard distributional models. Second, defects are observable and countable at defined inspection points, providing the data stream required by DMAIC. Third, production volumes are large enough to estimate failure rates with acceptable statistical confidence. These conditions define the domain within which DPMO is a valid and actionable metric. When the manufacturing context satisfies them, the methodology delivers consistent, economically visible improvements [24], [33].

The DMAIC cycle provides a structured problem solving path precisely because its statistical tools, including process capability analysis, control charts, and design of experiments, presuppose that the variable of interest follows a known distributional model estimated from adequate sample sizes [32]. When sample sizes are small, distributions are unknown, or the process changes faster than the measurement cycle, the statistical foundation of DMAIC weakens, and the resulting sigma-level estimates have wide confidence intervals that practitioners rarely report explicitly [33].

The transfer of Six Sigma thinking to software development, systems engineering, and autonomous systems reveals its structural dependency on those enabling conditions [21]. Automotive software processes do not produce stable, repeatable outputs in the manufacturing sense; they evolve with requirements changes, toolchain updates, and organizational decisions. Novel failure events in AV perception systems do not occur at statistically estimable frequencies because they arise from scenario conditions that may not yet have occurred. DPMO requires a denominator of known opportunities and a numerator of counted defects: both are undefined for the genuinely novel failure modes that dominate the risk profile of systems operating in open-world environments. Six Sigma's contribution to this paper is the precedent it establishes: treating a system property as measurable and controllable can transform engineering practice at a discipline level. What it cannot contribute is the measurement framework itself, because DPMO presupposes precisely the stable, bounded, data-rich context that novel systems lack.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

➢ *Reliability Engineering and Probabilistic Risk Assessment*

Classical reliability engineering and probabilistic risk assessment (PRA) provide quantitative tools for systems where the failure mode space is well-characterized and historical failure-rate data supports parameter estimation. Fault tree analysis (FTA), originally developed for aerospace and nuclear applications and formalized in the Fault Tree Handbook by Vesely and colleagues, constructs a deductive model of how component failures and human errors combine through Boolean logic gates to produce a top-level hazardous event [15]. Each leaf node in the tree carries a failure probability drawn from historical reliability databases, enabling quantitative estimates of top-event probability and identification of minimal cut sets. Event tree analysis extends this capability by modeling the sequence of outcomes following an initiating event, producing a probability distribution over consequence states.

The quantitative power of these methods has been demonstrated across nuclear power, chemical processing, aviation, and marine systems. Yu and colleagues confirm that traditional FTA, while widely used, is limited by its static event structure and its requirement for precise probability values for each basic event [34]. When data is sparse, both FTA and PRA resort to expert elicitation or generic databases, reducing the precision of the top-event probability estimate. Kaushik and Kumar observe that in complex novel systems, the exact failure probability of components is often unavailable, requiring approaches that can operate under uncertainty about the probability values themselves [35]. Bouafia and colleagues confirm that conventional PRA methods are further limited by their linear and reductionist assumptions, which reduce their effectiveness in capturing the variability propagation and complex interdependencies that characterize socio-technical systems [36].

The structural limitation for novel-environment systems follows directly. FTA assumes that the tree is complete: that all relevant failure modes are known and have been included as leaf nodes. For AV systems, the failure modes that matter most include perception failures triggered by untested environmental configurations, planning logic errors induced by novel scenario types, and organizational decisions that create failure preconditions not traceable to any component failure. None of these can be expressed as leaf nodes with known failure probabilities, because they have not yet occurred at measurable frequency. The method characterizes what is already known; it cannot characterize the unknown unknown failure space that defines the dominant risk of novel systems operating in genuinely open environments.

➢ *ISO 26262: Automotive Functional Safety*

ISO 26262 defines the most rigorous automotive functional safety process standard currently available. Published in 2011 and revised in 2018, the standard applies to electrical and electronic systems in road vehicles up to 3,500 kg and prescribes a complete lifecycle process from hazard analysis and risk assessment (HARA) through safety concept, system design, implementation, and verification [13]. HARA assigns Automotive Safety Integrity Levels (ASILs) to identified hazardous events based on the combination of severity (S), exposure (E), and controllability (C) ratings. ASIL assignments drive verification rigor: ASIL D, the highest level, requires the most comprehensive independent verification, software testing coverage, and hardware diagnostic coverage metrics [16].

The standard has achieved widespread adoption across the automotive supply chain and has demonstrably reduced the incidence of systematic failures and random hardware faults in production vehicles. Its V-model process structure provides clear traceability from safety goals through technical safety requirements to verification evidence, giving both internal engineering teams and external auditors a consistent framework for safety assurance. Messnarz and colleagues document how Automotive SPICE 3.0 has been extended to integrate ISO 26262 compliance into model-based development workflows for advanced driver assistance systems, demonstrating the standard's operational maturity within the industry [37].

A second limitation relates to the HARA's coverage assumptions. HARA identifies hazardous events and assigns ASIL ratings by considering severity, exposure, and controllability for each event, but it does not provide a method for systematically enumerating the full space of possible hazardous events in an open-world operational context. Two programs conducting HARA for the same perception function may identify very different hazardous event sets depending on the engineering team's knowledge and the scenarios they consider, and neither HARA nor any other ISO 26262 activity provides a structured process to audit completeness. The SEI construct in SUE is designed specifically to fill this pre-HARA gap, ensuring that all product-environment interface points are mapped before any domain-specific hazard analysis begins.

The limitation is one of scope and metric class. ISO 26262 is a process standard: it specifies what activities must be performed and what artifacts must be produced, but it does not provide a quantitative measure of the residual uncertainty that remains after those activities are complete [16]. Two programs can achieve identical ASIL D compliance while carrying substantially different levels of actual systemic risk, because compliance measures process execution rather than epistemic state. This extends to a critical scope exclusion: the standard addresses hazardous behavior arising from malfunctions of electrical and electronic systems and explicitly excludes hazards arising from performance insufficiency in a fault-free system. A perception system that correctly classifies a pedestrian in 99.9% of presentations but systematically fails under a specific lighting angle is not malfunctioning in the ISO 26262 sense and is outside its scope. The standard acknowledges this gap in its introductory clauses, which refer to ISO 21448 (SOTIF) for performance-related hazards. The acknowledgment confirms the structural limitation without resolving it.

➢ *ISO 21448: Safety of the Intended Functionality*

ISO 21448, the Safety of the Intended Functionality (SOTIF) standard, addresses the class of risk that ISO 26262 explicitly excludes: hazardous behavior arising from functional insufficiency in a system with no malfunction [12].

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

The standard partitions the operational scenario space into four regions: known safe scenarios, known unsafe scenarios, unknown safe scenarios, and unknown unsafe scenarios. Its primary goal is to reduce the unknown-unsafe region to an acceptable level through systematic scenario generation, simulation, track testing, and field validation. SOTIF is conceptually significant because it is the first major automotive standard to acknowledge that a system operating exactly as designed can produce hazardous outcomes through the insufficiency of its intended function in certain conditions, particularly when that function depends on perception algorithms interacting with an open-world environment [11].

The standard's scenario-based framework provides a practical starting point for validation activities. By requiring teams to systematically identify, simulate, and test scenarios across hazardous known and unknown regions, it establishes a structured process for addressing SOTIF-relevant risks [12]. The framework is compatible with the uncertainty-centric view proposed in this paper, and its known/unknown partition corresponds conceptually to the epistemic/aleatoric decomposition in the fourquadrant model. This compatibility makes SOTIF an important prior work that the present framework builds upon and generalizes.

The challenge of reducing the unknown-unsafe region to an acceptable level is complicated by what researchers in the field describe as scenario space intractability: the space of possible operational scenarios for a system like an AV is effectively unbounded, and any finite validation campaign samples only a small portion of it [11]. ISO 21448 acknowledges this by setting a criterion for when the residual risk from unknown-unsafe scenarios is sufficiently low, but it does not provide a quantitative model for estimating the probability that unexplored portions of the scenario space contain additional hazardous conditions. The SOTIF framework thus provides a process goal without a metric for tracking progress toward it. The SRD metric proposed in this paper provides exactly that, expressing the magnitude of remaining exogenous epistemic uncertainty as a tensor across the five system lifecycle layers.

Two structural limitations prevent SOTIF from closing the gap this paper addresses. First, the standard does not provide quantitative propagation mechanics. It identifies that unknown scenarios exist and provides a process for reducing that space, but it does not model how insufficiency at the perception layer propagates through planning and actuation to produce system-level loss, nor does it provide a scalar metric expressing the magnitude of the residual unknown-unsafe region [11]. Second, SOTIF takes scenarios as its unit of analysis and does not specify a process for systematically mapping the product-environment interface from which scenarios derive. This omission means that scenario coverage depends on the engineering team's prior knowledge of what interfaces exist and what conditions can stress them: the very knowledge gaps that the SEI constructs in SUE are designed to surface before scenario analysis begins.

- *ISO/SAE 21434: Automotive Cybersecurity Engineering*

ISO/SAE 21434, jointly developed by ISO and SAE International and published in 2021, provides a structured cybersecurity engineering lifecycle for road vehicles across concept, development, production, operation, and end-of-life phases [17]. Its core analytical method is the Threat Analysis and Risk Assessment (TARA), which identifies vehicle assets, characterizes applicable threats and attack paths, determines feasibility, and assigns Cybersecurity Assurance Levels (CALs) to drive verification and countermeasure requirements. The standard emerged from the recognition that modern vehicles, with their proliferation of electronic control units (ECUs), multiple wireless interfaces, and extended supply chains, present a substantial and growing attack surface that prior safety standards did not address [18].

The standard's adoption has been accelerated by regulatory requirements. UNECE Regulation WP.29, applicable across the European Union and several other markets, mandates compliance with cybersecurity management system requirements aligned with ISO/SAE 21434. Dobaj and colleagues document how the standard has become a central element of the automotive development process, noting that its TARA methodology establishes a rigorous threat modeling discipline previously absent from most automotive programs [18].

The structural limitation is architectural: ISO/SAE 21434 operates in a disciplinary silo. Its TARA produces cybersecurity-specific cut sets that identify attack paths to cybersecurity goals, but lacks a mechanism to determine how those paths interact with functional safety goals or SOTIF-relevant performance insufficiencies. Yu and colleagues confirm that ISO 26262 addresses the interaction of functional safety with cybersecurity only in general terms, without providing integrated analysis workflows [19]. In a software-defined vehicle where a spoofing attack degrades a perception subsystem's classification confidence, the incident is simultaneously a TARA-relevant cybersecurity event, a SOTIF-relevant performance insufficiency, and potentially a safety goal violation under ISO 26262. No existing standard provides a unified analysis artifact that would expose this cross-domain cut set at design time. It remains invisible to any organization conducting its safety, cybersecurity, and performance analyses as separate activities under separate standards.

- *Systems-Theoretic Accident Model and Processes*

Systems-Theoretic Accident Model and Processes (STAMP), developed by Leveson at MIT and formalized in *Engineering a Safer World*, represents the most conceptually complete existing treatment of safety in complex socio-technical systems [1]. STAMP reframes accidents as failures of safety constraints rather than chains of component failures. It views a system as a hierarchically structured set of controllers and controlled processes connected by control actions and feedback channels, and it treats safety as a control problem: accidents occur when the control structure fails to enforce the safety constraints that keep the system within safe operating bounds. This reframing is significant because it captures organizational decisions, software behavior, and

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

human-machine interaction as first-class contributors to accident causation, on equal analytical footing with hardware failures.

System-Theoretic Process Analysis (STPA) is the hazard analysis technique that instantiates STAMP for system design. Starting from a defined set of hazards, STPA constructs the system's hierarchical control structure, identifies unsafe control actions from each controller, and develops causal scenarios that explain how each unsafe control action could arise from process model flaws, control algorithm errors, communication failures, or context ambiguity [38], [39]. The technique has been applied to aviation, nuclear power, drone systems, surgical robotics, and autonomous vehicles, consistently identifying causal factors that event-chain methods would not surface. Mashkoor and colleagues' systematic mapping study documents STPA and its variants (STPA-Sec, STPASafeSec) as among the most widely adopted methods for joint safety and security analysis in software systems [40].

The gap between STAMP/STPA and the requirements of novel-environment risk management is the absence of quantification and economic translation. Plioutsias and colleagues state the reason explicitly: STPA does not generate a probability number for a hazard, because generating such a number for complex systems requires omitting important causal factors for which probabilistic information does not exist [38]. This is an honest acknowledgment of the method's scope, not a deficiency introduced by poor implementation. STPA identifies unsafe control actions and causal scenarios; it does not assign probabilities to them, does not produce a unified scalar risk metric, and does not express identified risks as expected financial loss. Deng and colleagues confirm that methods based on STAMP can only provide qualitative analysis, and several recent proposals integrate STPA with Bayesian networks or stochastic Petri nets specifically to supply the quantitative layer that STPA alone cannot provide [9]. SUE's SRD metric and ESL/ROSI translation can be understood as providing a systematic quantitative substrate for STAMP-style analysis: STPA identifies the control structure and unsafe actions; the SUE propagation model and Risk Cube quantify the uncertainty flowing through them.

➢ *Synthesis: The Persistent Gap*

Each framework reviewed in this section makes a genuine and substantial contribution within its intended scope. Six Sigma transformed manufacturing quality from an art of local best practices into a universal quantitative discipline. Reliability engineering and PRA provide rigorous probabilistic models for systems with wellcharacterized failure modes. ISO 26262 defines a mature and auditable process for managing systematic and random hardware failures in automotive E/E systems. SOTIF extends safety thinking to the open-world performance boundary that ISO 26262 cannot reach. ISO/SAE 21434 establishes the cybersecurity lifecycle discipline the automotive industry needed. STAMP/STPA provides the most conceptually sophisticated treatment of accident causation in complex systems available.

Several industry and research proposals have attempted to bridge these gaps through integrative approaches. STPA-Sec and STPA-SafeSec extend STPA to incorporate cybersecurity threats within the control structure model, and SAHARA integrates HARA with the STRIDE threat modeling method to enable coanalysis of safety and security hazards [40]. Goaloriented co-engineering frameworks provide structured methods for mapping both safety requirements and security requirements under a shared goal structure [38]. These integrative efforts represent genuine progress and demonstrate that the engineering community recognizes the problem of siloing. None of them, however, addresses all four missing capabilities simultaneously: they provide richer qualitative models of the safetysecurity interaction space, but they do not produce a unified quantitative metric, do not systematically map the product-environment interface as a first-class artifact before domain analysis begins, and do not translate risk into expected financial loss.

No framework, individually or in combination, closes the gap that Section I-B identified. Combining ISO 26262 with SOTIF and ISO/SAE 21434 within an Automotive SPICE process structure is the current industry best practice, and it represents a substantial capability. It still does not provide systematic product-environment interface analysis before domain decomposition begins. It still does not produce a unified quantitative metric spanning all three domains. It still does not expose cross-domain cut sets that span safety, cybersecurity, and performance branches. It still does not express systemic risk as expected financial loss in a form that supports return-on-investment calculations.

The present paper does not propose to replace any of these frameworks. SUE complements all of them: domain standards define which activities must be performed; the Uncertainty Diamond and SUE Risk Cube quantify how well those activities reduce the target system property. The SUE framework adds the analytical layer that current practice lacks: SEI/PEI analysis before domain decomposition, unified uncertainty propagation across layers, tensor-based risk quantification, and economic translation. Section III develops the formal system model and four-quadrant uncertainty theory that provide the mathematical foundation for the constructs introduced in Section I-D.

III. SYSTEM MODEL AND UNCERTAINTY THEORY

Translating the qualitative insight of Section I-A into an actionable risk framework requires a formal system model: one that represents all relevant contributors to systemic uncertainty, captures the relationships through which uncertainty propagates, and supports the mathematical constructs of the subsequent sections. This section introduces that model. Section III-A defines the graph structure. Section III-B assigns properties to its nodes. Section III-C provides the four-quadrant decomposition of uncertainty at each node, and Section III-D derives the intervention strategy implications of that decomposition.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

A. Socio-Technical System Graph

Socio-technical systems fail at the interaction of their technical and non-technical elements, not within those elements in isolation. A formal model of such a system must therefore represent not only hardware components and software modules but also the requirements from which they derive, the human roles that operate and maintain them, the organizational decisions that govern their development, and the interface points with the environment through which the system interacts with the world it must serve. Any model that restricts its node set to technical elements misses the primary locus of systemic risk.

Haimes establishes that risk assessment for systems of systems must be comprehensive, covering all aspects of the lifecycle from requirements through deployment and operation, and that organizational errors are responsible for the root causes of the majority of failures in critical engineering systems [41]. Pence and Mohaghegh confirm in their review of organizational factor incorporation into probabilistic risk models that latent conditions transmitted through organizational decision pathways are central contributors to system-level accidents, and that any framework that omits the organizational layer will systematically underestimate risk [42]. The graph model introduced in this section includes all of these elements by design.

- *Formal Definition:* A socio-technical system is modeled as a directed graph with layered structure:

$$G = (V, E, L) \quad (1)$$

Where V is the set of lifecycle nodes, $E \subseteq V \times V$ is the set of directed dependency edges, and L is the lifecycle layer partition of V.

Each of the three components of this triple is analytically necessary. V determines what the model can represent; edges in E determine what it can compute; and the layer structure L determines what it can distinguish. A model without layers conflates intra-layer dependencies and inter-layer couplings, losing the structural information that Section IV will show to be critical for predicting where nonlinear uncertainty amplification occurs.

- *The Node Set V:* The node set V comprises seven classes of lifecycle element, each contributing a distinct type of uncertainty to the system's risk profile:

Requirement nodes represent individual functional, safety, performance, and security requirements. Uncertainty at a requirement node ($v \in V_{req}$) arises from ambiguity in the requirement text, incompleteness of the requirement set relative to the operating environment, or volatility as conditions change during development. Requirement-layer uncertainty is analytically the highest-weight class in the SUE Risk Cube because it propagates downstream through every subsequent lifecycle layer: an ambiguous requirement generates ambiguous design artifacts, which in turn generate ambiguous implementation choices, which, in turn, reduce the efficacy of verification and validation activities [41].

Design nodes represent architectural decisions, interface specifications, and functional decompositions. Uncertainty at a design node reflects unvalidated design assumptions, unanalyzed coupling between subsystems, and interface specifications whose behavioral envelopes remain incompletely characterized. Modern systems engineering frameworks such as SysML and MBSE provide formal languages for expressing design structure, but they represent the system as designed rather than quantifying the uncertainty about whether that design is adequate for the environment it will encounter [43].

Implementation nodes represent software components, hardware elements, and their integration. Uncertainty at an implementation node corresponds to the probability that the implemented artifact deviates from its specification in ways that verification has not yet detected. Code coverage metrics, static analysis scores, and mutation testing rates serve as observable proxies for implementation-layer uncertainty.

Process nodes represent human roles, organizational decisions, and workflow elements that govern how the other lifecycle activities are executed. Studies of critical engineering failures across nuclear, aviation, and offshore energy sectors consistently attribute the root causes of accidents to process-layer conditions: schedule pressure, inadequate review culture, suppressed incident reporting, and resource constraints that reduce verification rigor [41], [42]. The inclusion of process nodes as first-class elements distinguishes this model from purely technical risk representations and is essential for capturing the organizational contributors to systemic failures documented in the case studies of Section XII.

Toolchain nodes represent the software tools, development environments, and automated pipelines used to produce, verify, and validate the other lifecycle artifacts. Toolchain uncertainty arises from unqualified tools, misconfigured environments, and configuration drift across development phases. ISO 26262 Part 8 provides a qualification process for software tools, and Automotive SPICE process capability ratings capture a dimension of toolchain maturity; both serve as data sources for assigning uncertainty values to toolchain nodes in practice [44]. *Environment interface nodes* represent the specific points at which product behavior meets the demands of the operational environment. These nodes are the formal embodiment of the System-Environment Interface (SEI) construct introduced in Section I-D. Each environment interface node corresponds to a defined interaction: a sensor processing a class of environmental stimuli, a planning algorithm interpreting a class of scene configurations, or a human-machine interface mediating an operator's decision under a class of vehicle states. Uncertainty at an environment interface node captures the gap between the product's design assumptions about what the environment will present and the actual distribution of conditions the environment may produce [11], [12].

Organizational interface nodes represent crossboundary relationships between the developing organization and external entities: regulatory bodies, suppliers, standards bodies, and end users. Uncertainty at these nodes reflects the

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

incompleteness of requirements crossing organizational boundaries, the variability in supplier process maturity, and the gap between regulatory guidance and deployment conditions.

➢ *The Edge Set E:*

A directed edge $(v_i, v_j) \in E$ encodes a dependency relationship from v_i to v_j: uncertainty or error at v_i has the potential to produce uncertainty or error at v_j. Three classes of dependency are captured under this single edge type.

Information flow edges represent dependencies where one artifact provides input to another. A requirement node provides input to a design node; a design node provides specifications to an implementation node; an implementation node provides artifacts to a verification process node.

Causal influence edges represent relationships where a decision or state at one node alters the risk profile at another, without a direct artifact flow. An organizational decision to compress the verification schedule increases implementation-layer uncertainty without producing a new artifact. A toolchain misconfiguration propagates latent errors into design and implementation nodes without any explicit handoff.

Coupling edges represent the interaction between environment interface nodes and the system nodes whose behavior they test. When the environment presents a condition that stresses an assumption embedded in a design node, the coupling edge from the environment interface node to the design node carries that challenge downstream.

The direction of each edge encodes the propagation direction of uncertainty, which is the subject of Section IV. Ashrafi demonstrates in a lifecycle risk modeling study that both horizontal interactions within a layer and vertical interactions across layers must be captured to accurately characterize system risk, and that omitting either produces systematically incomplete risk estimates [45].

➢ *The Lifecycle Layer Partition L:* The partition L assigns each node in V to exactly one of five lifecycle layers:

$$L = \{L_{\text{req}},\ L_{\text{des}},\ L_{\text{impl}},\ L_{\text{proc}},\ L_{\text{tool}}\}$$

The five layers correspond to the five axes of the SUE Risk Cube (Section V): the Requirements layer (L_{req}), the Design layer (L_{des}), the Implementation layer (L_{impl}), the Process layer (L_{proc}), and the Toolchain layer (L_{tool}). Environment interface nodes are assigned to the layer of the system element whose assumptions they test: an interface testing a requirements-layer assumption belongs to L_{req}; one testing a design-layer assumption belongs to L_{des}.

The layer structure is not merely organizational. It is analytically significant because edges within a layer (intra-layer dependencies) and edges between layers (inter-layer coupling) exhibit fundamentally different propagation behavior. Intra-layer edges connect nodes of the same type and typically represent linear dependencies: one design element directly constrains another. Inter-layer edges connect nodes of different types and are the primary sites of nonlinear uncertainty amplification. A process-layer decision about verification rigor couples to an implementation-layer node via an interlayer edge, with the coupling coefficient reflecting how much the organizational decision constrains or enables the technical activity. It is at these inter-layer couplings that the β interaction terms in the propagation model of Section IV are largest, and it is there that the most consequential systemic uncertainties reside.

This layer-structured graph model provides the formal substrate on which the node properties of Section III-B, the uncertainty decomposition of Section III-C, and the propagation model of Section IV are all built.

B. Node Properties

Each node $v_i \in V$ carries three scalar properties: an uncertainty magnitude $U(v_i)$, a hazard manifestation probability $P(v_i)$, and a propagation weight $W(v_i)$. Each of these three quantities is independently necessary. The magnitude of uncertainty alone does not determine risk: a node can be highly uncertain without being on any path to a hazardous outcome, and a node can carry modest uncertainty while sitting at a junction that amplifies it significantly downstream. The combination of all three properties is what makes each cell of the SUE Risk Cube independently measurable, independently actionable, and independently trackable, as formalized in Section V.

➢ *Uncertainty Magnitude $U(v_i) \in [0,1]$:*

The uncertainty magnitude $U(v_i)$ is a normalized measure of the unresolved uncertainty at node v_i. The range $[0,1]$ maps from complete knowledge ($U = 0$) to total ignorance ($U = 1$). Operationally, $U(v_i)$ is the complement of confidence in the node's correctness, completeness, and stability: the fraction of what must be true about this node that has not yet been confirmed through verification, validation, or observation [46].

Uncertainty in engineering systems arises from two structurally distinct sources [28], [47]. Epistemic uncertainty (U^{ep}) is reducible: it arises from incomplete knowledge about the system and can be decreased through additional analysis, testing, or review. Aleatoric uncertainty (U^{al}) is irreducible: it arises from inherent variability in the system or its environment and cannot be eliminated by acquiring more information. This distinction matters because the two types call for different interventions, a distinction that Section III-C develops through the four-quadrant decomposition. The total uncertainty at a node is:

$$U(v_i) = U^{\text{ep}}(v_i) + U^{\text{al}}(v_i) \quad (2)$$

The full four-quadrant decomposition of Equation 2, adding the endogenous/exogenous axis, is deferred to Section III-C. The key property for the present purposes is that $U(v_i)$ is measurable per node, and Section III-B4 specifies the measurement instruments for each lifecycle layer.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

➢ *Hazard Manifestation Probability $P(v_i) \in [0,1]$:*

The hazard manifestation probability $P(v_i)$ is the conditional probability that the uncertainty at node v_i, if realized as an error, produces a hazardous system state. It answers the question: given that this node's uncertainty is not resolved before the system is exposed to the operational environment, what is the likelihood that the resulting condition will lead to harm?

This property serves the same conceptual role as severity and exposure ratings in a Hazard Analysis and Risk Assessment (HARA) under ISO 26262, but it is defined at the node level rather than at the level of an identified hazardous event [13]. The HARA-level assessment is a downstream activity: it uses the outputs of the SEI and UTHA processes (Sections VII and VIII) to assign ASIL ratings at the safety goal level. $P(v_i)$ operates at the node level within the graph model to express how much each uncertain node contributes to the probability of a hazardous outcome, conditioned on the uncertainty being present.

Borgonovo's analysis of epistemic uncertainty in probabilistic safety assessment model elements demonstrates the importance of distinguishing between the degree of uncertainty in a parameter and the sensitivity of the safety assessment outcome to that uncertainty [46]. The $P(v_i)$ property captures this sensitivity at the node level: two nodes with identical $U(v_i)$ values may carry very different $P(v_i)$ values if one sits on a safety-critical data path and the other does not. The distinction ensures that the SUE Risk Cube reflects not only where uncertainty is concentrated but also which uncertainty locations matter most for outcome risk.

➢ *Propagation Weight $W(v_i) \in [0,\infty)$:*

The propagation weight $W(v_i)$ measures the downstream influence of node v_i in the dependency graph: how much uncertainty introduced or amplified at v_i reaches other nodes as it propagates through the edge set E. Nodes at the head of long dependency chains, or at junctions feeding many downstream elements, carry high propagation weights. Leaf nodes with no outgoing edges carry propagation weights of zero.

The weight is unbounded above because nodes in central structural positions can have an arbitrarily large influence on the downstream uncertainty field. A requirements node that is the source of dozens of derived design and implementation nodes may carry a weight one or two orders of magnitude greater than that of a leaf-level toolchain node. This asymmetry is preserved in the SUE Risk Cube rather than normalized away, because it carries actionable information: interventions at high-weight nodes produce proportionally larger risk reductions per unit of engineering effort.

In practice, $W(v_i)$ is estimated from the topology of the dependency graph: the number of nodes reachable from v_i through directed paths, weighted by the coupling coefficients α_{ij} along those paths. The SUE Level axis weights assigned to each lifecycle layer in the Risk Cube (Requirements: 4, Design: 4, Implementation: 3, Process: 2, Toolchain: 1) reflect the empirical regularities in propagation weight across node types: requirementlayer nodes are, by structural position, the highest-weight class because all other layers derive from them.

➢ *Measurement Instruments Per Layer:*

For the framework to produce the quantitative outputs claimed in Sections IV and V, $U(v_i)$ must be measurable in practice, not merely definable in principle. Table 2 specifies the primary measurement instruments for each lifecycle layer, together with the principal uncertainty sources that each instrument addresses. The instruments listed represent observable, auditable proxies for uncertainty: each maps to the $[0,1]$ scale through a domain-specific calibration step whose form the framework does not prescribe, requiring only that the mapping be explicit, monotone, and consistently applied [48], [49].

Defect density, code coverage, and static analysis scores have an established evidence base as quantitative proxies for implementation-layer uncertainty in safetycritical software development [50], [51]. Process-layer uncertainty, which classical reliability models typically omit, is captured here through Automotive SPICE (ASPICE) capability levels and schedule pressure indicators, consistent with the growing body of evidence that process maturity is a primary predictor of residual defect risk [44]. Tool qualification status under ISO 26262 Part 8 provides the analogous instrument for toolchain-layer nodes.

The $P(v_i)$ and $W(v_i)$ values do not require separate measurement instruments in the same sense. $P(v_i)$ is estimated through the UTHA process (Section VIII), which assigns hazard manifestation probabilities as part of the unified threat and hazard analysis step. $W(v_i)$ is derived from the graph topology and the coupling coefficient estimates assigned during the SEI/PEI mapping (Section VII). Together, the three properties provide a fully specified node-level input to the propagation model of Section IV and the tensor metric of Section V.

C. Four-Quadrant Uncertainty Decomposition

Section III-B introduced $U(v_i)$ as a scalar measure and presented its first decomposition into epistemic and aleatoric components. That decomposition is necessary but not sufficient: it tells the engineer whether a given uncertainty is reducible, but it does not tell them whether the reduction strategy should target the product or the environment. A software verification gap and an insufficient scenario coverage gap are both epistemic, but the engineering activities that address them are categorically different. The four-quadrant decomposition adds a second axis that makes this distinction explicit and thereby determines which class of intervention each cell of the Risk Cube requires.

➢ *The Two Axes:*

The four-quadrant model decomposes uncertainty at each node v_i along two independent axes.

- Axis 1: Epistemic versus Aleatoric. This axis distinguishes the reducibility of the uncertainty. Epistemic uncertainty (U^{ep}) arises from incomplete knowledge: ambiguous requirements, unreviewed design decisions, untested

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

scenario configurations, and unknown parameter values that are fixed but unmeasured. Epistemic uncertainty is reducible in principle through additional information, analysis, testing, or review [27], [52]. As Hester and Dohi state, the key property of epistemic uncertainty is that an increase in knowledge can lead to a reduction in predicted uncertainty, all else being equal. Aleatoric uncertainty (U^{al}) arises from inherent variability: sensor noise, manufacturing tolerances, stochastic human behavior, weather variation, and adversarial actor behavior. Aleatoric uncertainty is irreducible with additional information; it can only be managed through design margins, redundancy, and operational constraints [26], [28]. Both types are present in every real system. The engineering value of the axis lies in the fact that only epistemic uncertainty should be targeted by knowledge-acquisition investments; investing in additional testing to reduce an aleatoric floor is, by definition, engineering effort that cannot succeed.

- Axis 2: Endogenous versus Exogenous. This axis distinguishes the origin of the uncertainty relative to the product-environment boundary. Endogenous uncertainty (U^{en}) concerns the product itself: its requirements, design correctness, implementation quality, process maturity, and toolchain integrity. It is uncertainty about whether the product was built right. Exogenous uncertainty (U^{ex}) concerns the product's interaction with its operational environment: the breadth and representativeness of scenario coverage, the variability of environmental conditions, user behavior, and the characteristics of potential adversaries. There is uncertainty about whether the product was built for the right conditions. Uday and Marais establish this endogenous/exogenous distinction in the context of systems-of-systems resilience, defining endogenous uncertainty as that arising from internal system evolution and exogenous uncertainty as that driven by the external environment, including new threat types and changing stakeholder needs [53]. Dietze formalizes the same partition in ecological prediction theory, demonstrating that separating internal stability from exogenous environmental sensitivity is necessary for prescribing targeted interventions [54].

➢ *The Full Decomposition:*

Applying both axes simultaneously yields a four-component decomposition of the total uncertainty at each node:

$$U(vi) = U\text{en,ep}(vi) + U\text{en,al}(vi) + U\text{ex,ep}(vi) + U\text{ex,al}(vi) \quad (3)$$

The four components correspond to the four cells of Table 2, which Table 2 places before its content in the text following this paragraph. Each component carries a distinct semantic meaning and a distinct class of engineering response.

Table 2 presents the four-quadrant matrix in full. The cells describe not only what each component represents but also the engineering logic it prescribes: endogenousepistemic uncertainty calls for investment in product knowledge; exogenous-epistemic uncertainty calls for investment in environment knowledge; aleatoric uncertainties in both dimensions call for robustness investment rather than knowledge investment.

Table 2 Measurement Instruments for Node Uncertainty $U(V_i)$ by Lifecycle Layer

Layer	Node Type	Measurement Instrument	Primary Uncertainty Source
Requirements	Requirement node	Structured review scoring; ambiguity metrics; traceability completeness rate	Incompleteness, ambiguity, volatility
Design	Architecture element	FMEA coverage; interface analysis completeness; assumption register gap count	Unvalidated assumptions, coupling complexity
Implementation	Software component	Statement and branch coverage; static analysis findings; mutation testing score	Residual defect density; untested execution paths
Process	Organizational decision	ASPICE capability levels; schedule pressure index; safety culture survey score	Capability gaps; resource constraints; culture risk
Toolchain	Tool / environment	ISO 26262 Pt 8 qualification status; version control integrity; CI/CD failure rate	Tool errors; configuration drift; build variability

Table 3 Four-Quadrant Uncertainty Decomposition: Components, Examples, and Intervention Strategies

	Endogenous (Product)	Exogenous (Environment)
Epistemic (Reducible)	$U^{en,ep}$: Insufficient verification coverage, ambiguous requirements, unreviewed design decisions, unqualified tools. *Intervention: More testing, formal methods, peer review, tool qualification.*	$U^{ex,ep}$: Insufficient scenario coverage, untested ODD boundaries, uncharacterized user behavior, unvalidated environment models. *Intervention: Broader validation, field exposure, ODD expansion, SEI/PEI analysis.*
Aleatoric (Irreducible)	$U^{en,al}$: Manufacturing tolerances, hardware random failure rates, component aging, random software faults. *Intervention: Design margins, hardware redundancy, diagnostic coverage, manufacturing process control.*	$U^{ex,al}$: Weather variability, stochastic human behavior, adversarial actors, infrastructure degradation. *Intervention: ODD constraints, runtime monitoring, Safe Maneuver Objective (SMO) architecture.*

The endogenous column maps to the left arm of the Uncertainty Diamond (Section VI). The exogenous column maps to the right arm. The SEI/PEI analysis (Section VII) is the primary instrument for measuring and reducing Uex,ep.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

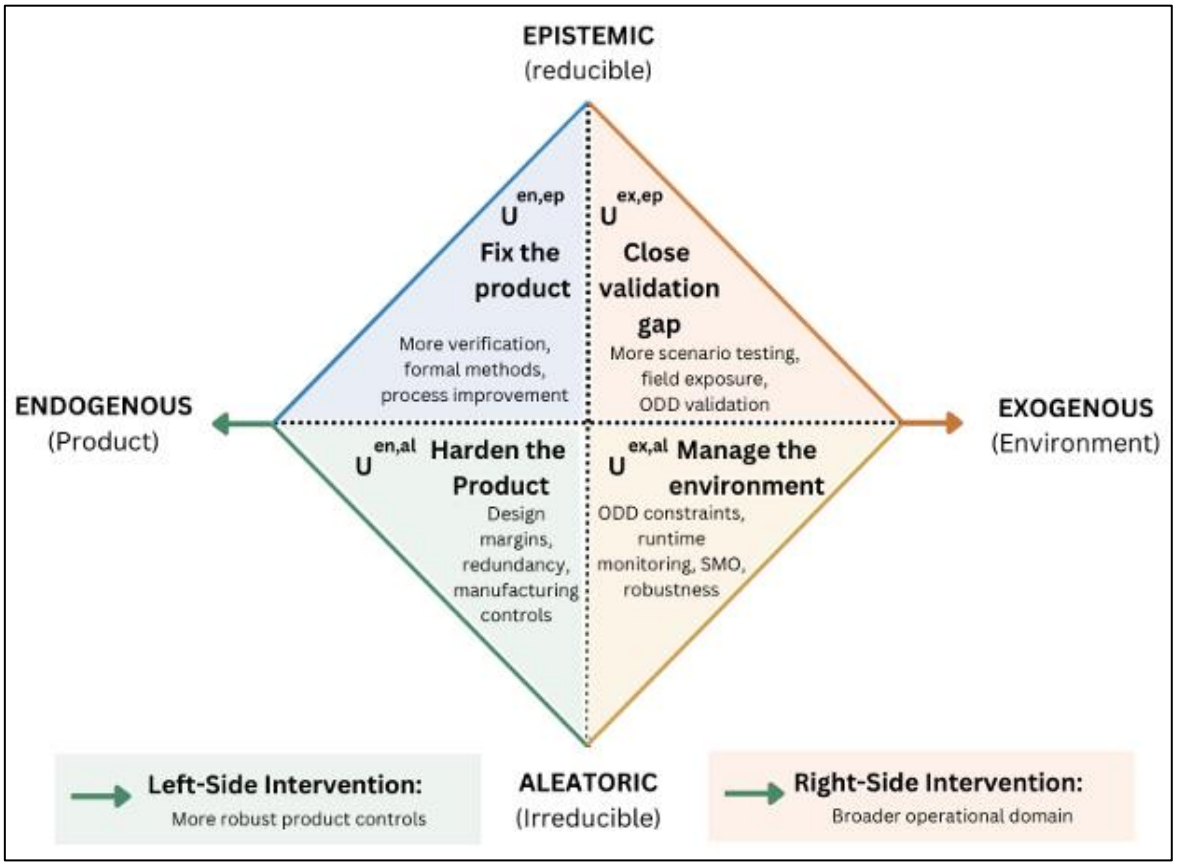

Fig. 1. Four-Quadrant Uncertainty Decomposition: Epistemic/Aleatoric × Endogenous/Exogenous

Figure 1 presents a schematic representation of the four-quadrant space, showing the two axes and the four cells with their associated uncertainty classes and engineering response directions. The endogenous column occupies the left half, and the exogenous column occupies the right half, reflecting the left-arm and rightarm structure of the Uncertainty Diamond introduced in Section VI.

➢ *Why the Decomposition Matters:*

The practical argument for the four-quadrant decomposition rests on a single claim: conflating any two cells leads to misallocation of engineering resources. Three specific conflations are worth examining explicitly.

Conflating epistemic and aleatoric uncertainty causes organizations to invest in knowledge acquisition activities, such as extended testing campaigns, when the dominant uncertainty is aleatoric and irreducible. No amount of additional scenario testing eliminates sensor noise; that residual must be addressed through redundancy and monitoring architecture. Winter and colleagues document precisely this pattern in their analysis of autonomous robotic systems, observing that epistemic sources of risk arising from imperfect knowledge of realworld conditions are frequently confused with the inherent variability of those conditions, leading to validation approaches that cannot reduce the residual risk [55].

Conflating endogenous and exogenous epistemic uncertainty causes organizations to target product verification when the dominant gap is in environment characterization. A perception algorithm achieving 99.5% accuracy on its test dataset may carry very high $U^{ex,\,ep}$ if the test dataset does not represent the actual ODD. Caballero and colleagues document this in the autonomous driving context, showing that environmental factors, including precipitation, lighting, and stochastic human behavior, constitute sources of exogenous uncertainty that standard product-level testing cannot adequately characterize [56]. Adesiji and colleagues confirm that uncertainty in predicting robot responses to unforeseen environmental conditions, arising from the inability of conventional verification to account for all potential conditions the robot might encounter, is a primary source of safety risk in robotic deployment [57].

Conflating endogenous and exogenous aleatoric uncertainty leads to a mismatch between risk management architectures and the source of variability. Hardware random failures (endogenous-aleatoric) are managed through redundancy and diagnostic coverage, which provide no benefit against weather variability or stochastic pedestrian behavior (exogenous-aleatoric). The latter requires ODD constraints and runtime monitoring: fundamentally different system architectures serving fundamentally different uncertainty sources.

➢ *Connection to the Uncertainty Diamond:*

The two axes of the four-quadrant model map directly onto the structure of the Uncertainty Diamond (Section VI). The endogenous column corresponds to the left arm of the Diamond, which spans product development activities from requirements through system integration. The exogenous

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

column corresponds to the right arm, which spans environment characterization activities from ODD definition through field validation. The two arms converge at the Unified Loss Goal, which is the point where product capability must meet environmental demand: where $U^{en,\ ep}$ and $U^{ex,ep}$ jointly determine whether the system can perform safely in the conditions it will actually encounter.

The epistemic/aleatoric axis, cutting across both columns, determines the character of the activities on each arm. Epistemic uncertainties on both arms drive engineering activities aimed at acquiring knowledge and closing gaps. Aleatoric uncertainties on both arms set floors that drive engineering activities aimed at building robustness against what cannot be known in advance. The Uncertainty Diamond provides the process model; the four-quadrant decomposition provides the analytical vocabulary for assigning each engineering activity to the uncertainty type it is designed to address.

Section III-D develops the intervention strategy implications of the decomposition in full, and Section IV formalizes how uncertainty in each quadrant propagates through the dependency graph of Section III-A.

D. Intervention Strategy Per Quadrant

The four-quadrant decomposition introduced in Section III-C is not merely a taxonomy. Its analytical value lies in the engineering implications it carries: each quadrant prescribes a specific class of intervention, and the cost-effectiveness of any safety investment depends on aligning the intervention type to the dominant uncertainty source. Investing in knowledgeacquisition activities against an aleatoric floor produces no uncertainty reduction; investing in robustness margins against a reducible epistemic gap delays the correct intervention. The four strategies described below are mutually exclusive at the class level and additive at the portfolio level: a complete uncertainty management plan addresses all four quadrants simultaneously, targeting each with its appropriate instrument.

➢ *$U^{en,ep}$: Reduce Through Better Engineering:*

Endogenous-epistemic uncertainty is reducible through engineering activities that increase the organization's knowledge of the correctness and completeness of its product. The primary instruments are verification activities: structured peer reviews, formal analysis, static analysis, dynamic testing, and model checking. Kulkarni and colleagues establish in a formal model of systems engineering projects that verification activities mitigate epistemic uncertainty by revealing more information about the true current state of the system design, and they demonstrate that frequent verification is a costminimizing strategy precisely when epistemic uncertainty is the dominant source of risk [58], [59]. Singh and colleagues demonstrate the application of formal methods across the full development lifecycle of safetycritical systems, from requirements specification through code generation, providing strong empirical evidence that formal verification substantially reduces the probability of residual design errors that informal testing alone would leave undetected [60].

Automotive SPICE (ASPICE) process capability levels serve as an aggregate indicator of $U^{en\ ,ep}$ at the process and toolchain layers: higher capability levels indicate more rigorous process execution, more systematic defect prevention, and lower residual uncertainty in the engineering artifacts produced [44]. The ASPICE Level 4 and Level 5 processes, which add quantitative process management and continuous improvement, respectively, provide the organizational infrastructure for systematically tracking and reducing $U^{en,ep}$ across lifecycle layers. The investment case is straightforward: each point of reduction in $U^{en,ep}$ translates directly into lower SRD at the affected nodes and, through the propagation model of Section IV, into lower downstream uncertainty at all nodes that depend on them.

➢ *$U^{ex,ep}$: Reduce Through Broader Validation:*

Exogenous-epistemic uncertainty reflects gaps in the organization's knowledge of the environment the system will encounter. It can be reduced through activities that more completely characterize and validate the operational scenario space.. The primary instruments are scenario testing, simulation campaigns, track testing, field validation, and systematic SEI/PEI analysis. The SEI/PEI construct (Section VII) specifically targets $U^{ex\ ,ep}$: it surfaces the assumptions embedded at each productenvironment interface point, identifies the gap between each assumption and what the environment may actually present, and generates test cases designed to probe that gap [12]. Each test case that exposes a gap in the current design converts $U^{ex,ep}$ into a known design deficiency, which is then addressed by either modifying the product (reducing $U^{en,\ ep}$) or constraining the operational domain (reducing $U^{ex,\ al}$).

ODD expansion is an important mechanism for reducing $U^{ex,\ ep}$ when the root cause of the gap is insufficient field exposure. As Caballero and colleagues demonstrate, ODD supervision requires probabilistic monitoring of environmental state variables to detect when the system approaches operating conditions that have not been adequately validated [56]. Broader field exposure across more diverse route types, weather conditions, and traffic configurations reduces the proportion of the environment that remains epistemically uncharacterize. The Uncertainty Diamond's right arm (Section VI) provides the process structure for systematically executing this class of intervention, from the ODD definition through simulation to field validation.

➢ *$U^{en,al}$: Manage Through Hardening:*

Endogenous-aleatoric uncertainty arises from inherent variability within the product: hardware random failure rates, manufacturing tolerances, component aging, and the stochastic behavior of electronic systems under nominal operating conditions. This uncertainty cannot be reduced by acquiring more knowledge about the design; it is a property of the physical technology, not of the engineering team's epistemic state. The appropriate intervention class is therefore product hardening: design margins, hardware redundancy, diagnostic coverage, and manufacturing process controls [13]. Design margins create safety buffers between nominal operating conditions and failure thresholds, absorbing the variability introduced by hardware random failures.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

Redundant architectures with diverse implementations ensure that the simultaneous failure of all redundant instances is highly unlikely, even when the failure rate of each instance is nonzero.

ISO 26262 Part 5 addresses this quadrant directly: its Automotive Safety Integrity Level (ASIL) requirements drive hardware architectural metrics, including Single Point Fault Metric (SPFM) and Latent Fault Metric (LFM), which quantify the diagnostic coverage and structural robustness required to meet the target residual risk level [13]. These metrics operationalize $U^{en,\,al}$ management: programs that meet ASIL D hardware targets have demonstrably reduced the probability that endogenous-aleatoric uncertainty manifests as a safetycritical failure. This quadrant is the domain in which classical reliability engineering has the strongest tools, and those tools remain appropriate here. The fourquadrant model does not displace them; it assigns them to their correct scope.

➢ *$U^{ex,al}$: Manage Through Environment Architecture:*

Exogenous-aleatoric uncertainty arises from inherent variability in the operational environment: weather fluctuations, stochastic pedestrian and driver behavior, adversarial actors, and infrastructure degradation. Like its endogenous counterpart, it is irreducible through additional testing or design verification, because additional knowledge does not change the fact that the environment will continue to vary in ways that the product cannot fully anticipate. The management strategy is therefore environmental: constrain the ODD to exclude conditions where $U^{ex,\,al}$ exceeds an acceptable threshold, build runtime monitoring architectures that continuously evaluate the system's position within the ODD, and maintain a pre-computed Safe Maneuver Objective (SMO) that provides a safe recovery trajectory whenever environmental conditions push the system toward a loss state.

Betz and colleagues, based on field experience with autonomous racing systems, establish that it is paramount to understand system behavior when ODD assumptions are violated and to ensure the response remains reasonable through soft constraints [6]. The robustness principle they articulate applies directly to $U^{ex,al}$ management: rather than designing for specific nominal environmental conditions, systems must degrade gracefully as environmental variability pushes them toward ODD boundaries. Caballero and colleagues formalize this through ODD state-space monitoring models that issue alerts when probabilistic forecasts indicate that ODD limits are likely to be exceeded, enabling the system to initiate a controlled transition before the environmental condition becomes genuinely hazardous [56]. The SMO architecture (Section IX) provides the operational mechanism for executing that transition, precomputing the set of safe maneuver options available under different states of residual environmental variability.

The four intervention strategies together constitute the complete engineering response to the uncertainty field described by the system graph of Section III-A. A program that addresses only $U^{en,ep}$ and $U^{en,al}$, as most current safety processes effectively do, leaves the exogenous column unaddressed and will systematically underestimate the residual risk present at productenvironment interface nodes. The SUE Risk Cube (Section V) provides the structured accounting that makes this misallocation visible: high SRD in the exogenous column, combined with low SRD in the endogenous column, is a diagnostic signature of a program that has invested heavily in product verification while neglecting environment characterization. The uncertainty propagation model of Section IV formalizes how uncertainty in each quadrant flows through the system graph to produce the node-level scores that populate the Risk Cube.

IV. UNCERTAINTY PROPAGATION

The graph model of Section III-A and the node properties of Section III-B provide a static description of the system: a set of nodes, each carrying uncertainty $U(v_i)$, connected by directed dependency edges. A static description cannot answer the question that risk management requires: when uncertainty is present at a source node, how much of it reaches each downstream node, and by what path? Answering that question requires a propagation model that translates the edge structure into a quantitative account of how uncertainty flows through the system. This section develops that model in two stages. Section IV-A introduces the first-order linear baseline, which is exact under loose coupling and provides the substrate for intervention effect calculations. Section IV-B generalizes the baseline with the β interaction terms that capture the nonlinear amplification characteristic of tightly coupled socio-technical systems, and Section IV-C examines cross-axis propagation between the endogenous and exogenous quadrants.

A. First-Order (Linear) Model

The first-order propagation model computes the uncertainty at each node v_j as a weighted sum of the uncertainties at its parent nodes in the directed graph $G = (V,E,L)$. For a node v_j with parent set parents(j) = $\{v_i : (v_i,v_j) \in E\}$, the first-order model is (see also Figure 2, left panel):

$$U(v_j) = \sum_{i \in \text{parents}(j)} \alpha_{ij} \cdot U(v_i) \tag{4}$$

Where $\alpha_{ij} \in [0,1]$ is the coupling coefficient on edge (v_i,v_j), representing the fraction of the parent's uncertainty that is transmitted to the child node under conditions of linear dependency. When $\alpha_{ij} = 1$, the child inherits the parent's uncertainty in full; when $\alpha_{ij} = 0$, the edge carries no propagation and the two nodes are uncertainty-independent. Equation (4) is a DAG message-passing model of the same structural class as the Bayesian network propagation, where nodes represent uncertain variables and directed edges encode conditional dependencies [61], [62].

➢ *Semantics of the Coupling Coefficient α_{ij}:*

The coupling coefficient α_{ij} captures two distinct aspects of the relationship between v_i and v_j: the structural strength of the dependency and the degree to which the child node is protected against the parent's uncertainty by independent verification or redundant information paths. Three cases bound the practical range.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

At $\alpha_{ij} = 1.0$, the child node derives entirely from the parent with no independent verification or complementary information source. A software component v_j that implements a requirement v_i without any design review between requirements completion and coding represents a near-unity coupling on the requirements-toimplementation edge. Any uncertainty in the requirement propagates to the implementation in full.

At $\alpha_{ij} \approx 0.3–0.5$, a moderate coupling reflects a dependency that is real but partially attenuated by an intervening engineering activity. An architecture review that examines the requirement before the design is baselined reduces coupling between the requirement and design nodes, because the review converts some fraction of the requirements-layer epistemic uncertainty into a known discrepancy that is addressed before propagation. At $\alpha_{ij} \approx 0.1–0.2$, a weak coupling reflects a marginal or indirect dependency where independent information substantially supplements the parent. A toolchain node v_i with a well-characterized qualification status introduces only a small fraction of its residual uncertainty into the implementation node v_j that uses it, because qualification evidence independently bounds the tool error probability.

These boundary cases show that α_{ij} is operationally interpretable as the fraction of the parent's uncertainty that survives the engineering activities on the edge between v_i and v_j. Hall and colleagues, in their analysis of risk propagation through organizational dependency networks, demonstrate that linear dependency-weighted models of this form provide computationally efficient propagation calculations whose outputs bound the results of more complex nonlinear models from above [63]. This bounding property is analytically valuable: under the linear model, a computed $U(v_j)$ value is an upper bound on the true propagated uncertainty at v_j, making the first-order model conservative in the risk assessment sense. Figure 2 illustrates both propagation regimes side by side.

➢ *Estimating α_{ij} in Practice:*

Coupling coefficients are estimated through structured expert elicitation anchored to the measurement instruments of Table II. For each edge $(v_i,v_j) \in E$, the analyst poses the question: if the parent node v_i carries the maximum credible uncertainty for its node type, what fraction of that uncertainty would a child node v_j inherit under the current engineering process, given all planned review, verification, and validation activities between the two nodes? The answer is then calibrated against the observable process quality indicators assigned to the process layer nodes on or near the edge.

Two sources provide initial calibration anchors. Automotive SPICE capability levels at the process nodes adjacent to an edge set a lower bound on α_{ij}: higher capability levels imply more systematic upstream uncertainty reduction and thus lower residual transmission [44]. ISO 26262 audit evidence for the activities on the edge, specifically the rigor of review and the coverage of verification, provides a second independent calibration point. Where the two sources conflict, the higher (more conservative) estimate is used until additional evidence resolves the discrepancy.

For toolchain-layer edges, the ISO 26262 Part 8 tool qualification class provides a direct mapping: a class TD3 tool (no direct influence on safety-relevant output) carries a coupling coefficient near zero on all edges to safety-domain nodes; a class T3 tool (direct influence, no error detection) carries a coupling coefficient approaching one [13].

➢ *Adequacy Conditions:*

The first-order model is exact under two conditions: loose coupling between nodes, meaning that each node's uncertainty is genuinely independent of any other node's uncertainty conditioned on their shared parents; and the absence of feedback loops within the active portion of the graph.

Loose coupling is well approximated within a single lifecycle layer. Intra-layer dependency edges connect nodes of the same type, and the uncertainty they carry flows through direct technical relationships, which are not amplified by combining different uncertainty types. A design node whose uncertainty derives from two parent design nodes with loosely coupled assumptions will inherit a weighted sum of those assumptions' uncertainties without amplification, provided neither assumption constrains the other. The Bayesian network literature confirms this: in DAG-structured systems with conditionally independent parent nodes, belief propagation through the graph is linear in the parent probabilities, and the conditional probability table reduces to a weighted mixture of parent influence terms [61], [64].

Loose coupling fails at inter-layer edges, particularly those connecting the endogenous and exogenous quadrants of the four-quadrant model. When a vague requirement node (endogenous-epistemic) is also connected to an unvalidated scenario coverage node (exogenousepistemic), the uncertainty at the downstream design node is not the weighted sum of the two parent uncertainties. The combined effect is superadditive: a design that is ambiguous about its required behavior cannot be validated against scenarios whose relevance is itself uncertain, and the resulting uncertainty at the design node is larger than either parent's uncertainty alone would produce. This superadditive regime motivates the β interaction terms in Section IV-B.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

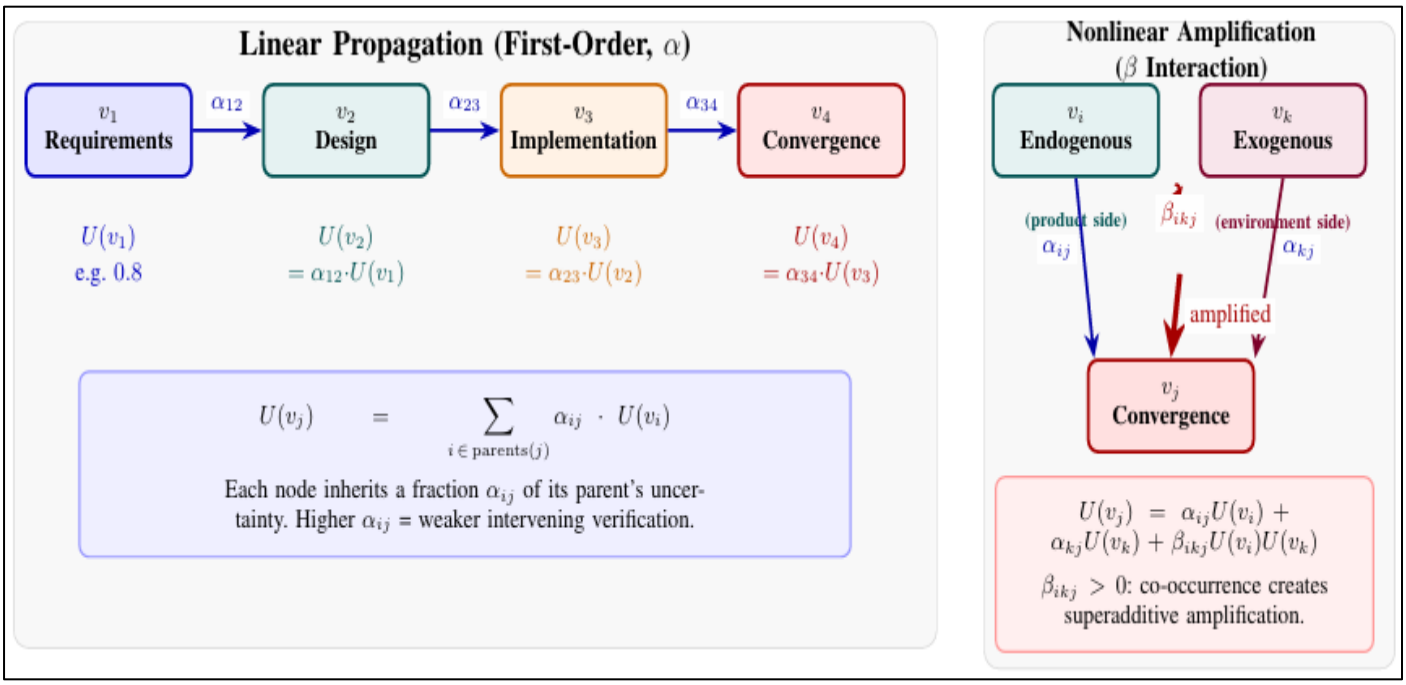

Fig 2 Uncertainty propagation in the SUE framework. *Left*: First-order linear model (Equation (4)). Uncertainty at node v_j is a weighted sum of parent uncertainties, attenuated by coupling coefficients $\alpha_{ij} \in [0,1]$ on each directed edge. A higher α_{ij} indicates weaker intervening verification; $\alpha_{ij} = 1$ means full inheritance of the parent's uncertainty. *Right*: Nonlinear amplification model (Equation (5)). At a convergence node where an endogenous parent v_i and an exogenous parent v_k meet, the β_{ikj} interaction term adds a product contribution $\beta_{ikj} \cdot U(v_i) \cdot U(v_k)$. When $\beta_{ikj} > 0$, neither parent's uncertainty compensates the other's, and the combined uncertainty at v_j exceeds the linear sum – the superadditive amplification that distinguishes cross-axis failure modes from single-domain risks.

Feedback loops arise when design decisions alter requirements interpretation, when field data revise organizational risk tolerance, or when verification failures trigger requirements changes. The first-order model does not admit cycles within a single propagation pass. The framework handles temporal feedback through iterated propagation: the SRD is recomputed at each lifecyclephase gate, and the updated $U(v_i)$ values from phase k serve as inputs to the propagation pass in phase $k + 1$. Each individual pass remains acyclic and is computed using Equation (4).

The first-order model provides a necessary analytical baseline: it characterises how uncertainty propagates under independence assumptions, establishes conservative upper bounds on node-level risk contributions, and supports efficient calculation of intervention effects. Its adequacy for loosely coupled intra-layer dependencies makes it the right model for the toolchain and process layers of the lifecycle graph, where dependencies are direct and amplification effects are small. Its insufficiency for novel socio-technical systems operating in open-world environments, where inter-layer coupling and cross-quadrant interaction are the norm rather than the exception, motivates the nonlinear extension of Section IV-B.

B. Nonlinear Amplification: The β Interaction Terms

Section IV-A demonstrated that the first-order model is exact under loose coupling and serves as a conservative upper bound elsewhere. Its central limitation is that it assumes the uncertainty contributions of different parent nodes are additive: the uncertainty arriving at v_j from parent v_i and from parent v_k simply sum. This assumption fails in precisely the settings that matter most for novel socio-technical systems. When an endogenous weakness and an exogenous challenge co-occur at the same downstream node, neither source of uncertainty can compensate for the other, and the compound effect at v_j exceeds what either parent would produce individually. Capturing this superadditive behavior requires a secondorder extension to the propagation model.

➢ *The Generalized Propagation Model:*

The generalized propagation model extends Equation (4) by adding (right panel of Figure 2), a second-order interaction term for every ordered pair of parent nodes (v_i, v_k) with $i < k$:

$$U(v_j) = \sum_{i \in \text{parents}(j)} \alpha_{ij}\, U(v_i) + \sum_{\substack{i,k \in \text{parents}(j) \\ i<k}} \beta_{ikj}\, U(v_i)\, U(v_k) \qquad (5)$$

The first sum is the first-order linear term from Equation (4). The second sum adds a product term for each parent pair, weighted by the interaction coefficient $\beta_{ikj} \geq 0$. When all $\beta_{ikj} = 0$, Equation (5) reduces exactly to Equation (4). Non-zero β_{ikj} values encode the degree to which the co-occurrence of uncertainty at v_i and v_k amplifies the uncertainty at v_j beyond the additive sum of their individual contributions.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

The product $U(v_i)\cdot U(v_k)$ is the interaction term: it is zero if either parent's uncertainty is zero, and it reaches its maximum when both parents carry high uncertainty simultaneously. This structure makes the model sensitive to exactly the failure mode that Perrow identified in his analysis of complex, tightly coupled systems: low-level failures that become systemic not through their direct magnitude but through unexpected interactions among components that each appear manageable in isolation [2]. A system that passes individual-node assessments can still carry a large β-driven risk at convergence nodes where multiple uncertain parents meet.

➢ *Semantics and Sign of β_{ikj}:*

The interaction coefficient β_{ikj} is bounded below at zero. It cannot be negative in this framework because the physical mechanism being modeled is mutual amplification, not mutual cancellation. Two sources of uncertainty at the same downstream node may each independently create risk, but there is no physical basis for claiming that their co-occurrence reduces uncertainty below the linear sum. Superadditivity is the only meaningful direction: the consequence caused by multiple co-occurring uncertainties is at least equal to, and generally greater than, the sum of consequences from each individually [65].

The interpretation of a specific β_{ikj} value is: for a unit increase in $U(v_i)\cdot U(v_k)$ at node v_j, the uncertainty at v_j increases by β_{ikj} above what the linear terms alone would predict. At $\beta_{ikj} = 0.0$, the two parents are uncertainty-independent and the linear model is exact for their combination. At $\beta_{ikj} = 0.3$, a node pair with each parent carrying $U = 0.7$ contributes $0.3 \times 0.7 \times 0.7 = 0.147$ additional uncertainty beyond the linear sum, a roughly 10–20% amplification on a typical requirementlayer node. At $\beta_{ikj} = 0.5$ or higher, the amplification becomes dominant and the interaction term can exceed the individual linear contributions.

Functional resonance analysis (FRAM) provides a qualitative analogue for this effect: in complex sociotechnical systems, undesired outcomes arise from the resonance of variability across coupled functions rather than from failures within individual functions [66]. The β term is the quantitative expression of this resonance mechanism, formalizing it as a product coefficient that scales with the magnitude of both interacting uncertainty sources.

➢ *Where β_{ikj} Is Largest: Cross-Axis Interactions:*

Not all parent pairs produce meaningful β values. The structural analysis of the graph $G = (V,E,L)$ identifies two categories where interaction amplification is most pronounced.

Cross-layer interactions occur when one parent node belongs to a higher lifecycle layer (requirements or design) and the other belongs to a lower layer (process or toolchain). An ambiguous requirement node ($v_i \in L_{\text{req}}$, high $U^{\text{en,ep}}$) paired with a low-capability process node ($v_k \in L_{\text{proc}}$, high $U^{\text{en, ep}}$) converging at a design node v_j produces a large β-driven amplification because the design activity cannot compensate for the ambiguous input when the process executing it is also deficient. The key insight is mutual non-compensation: neither parent can cover for the other's uncertainty, so their product drives a disproportionate contribution to the child.

Cross-axis interactions are the most critical category for novel-environment systems. They occur when one parent node lies in the endogenous quadrant (product uncertainty) and the other lies in the exogenous quadrant (environment uncertainty). An endogenous design weakness meeting an exogenous scenario coverage gap at a convergence node produces a β interaction that siloed analysis is structurally unable to detect. The endogenous analysis confirms the product's behavior is within its specification; the exogenous analysis notes that certain scenario configurations are untested. Neither analysis can capture the compound failure that arises only when a specification-compliant product encounters an undertested scenario in the field. This is the class of failure that the Uber ATG case (Section XII) and the Takata case (Section XII) exemplify: the accident did not arise from a single domain failure but from the crossaxis amplification at the intersection of endogenous and exogenous uncertainty sources.

In the Uncertainty Diamond (Section VI), cross-axis interactions are structurally located at the convergence point of the left arm (endogenous) and the right arm (exogenous): the Unified Loss Goal node. The β values are largest at this convergence because it is precisely there that endogenous product capability must be sufficient to handle the exogenous environment demand. When both arms carry unresolved uncertainty, the product term $U^{\text{en}}(v_j)\cdot U^{\text{ex}}(v_j)$ is large, and β_{ikj} encodes how much that co-occurrence amplifies the loss probability beyond what either arm alone would produce.

➢ *Calibration Anchors from Retrospective Case Analysis:*

Table 4 presents expert-elicited β values derived from retrospective analysis of three documented systemic failures. These values are calibrated against the measurement instruments of Table 2 and expressed on the $[0,1]$ scale consistent with the $U(v_i)$ domain. They serve as empirical anchors for practitioners conducting forward-looking β elicitation for new programs. Section XII provides the full node-level analyses from which these values are extracted.

Table 4 Expert-Elicited β Interaction Coefficients: Cross-Case Summary

Case	Interaction Pair	β Value	Interpretation
Takata Airbag	Requirements gap × cost-driven material selection	0.336	Dominant amplification; exceeded every individual node contribution except the requirements node itself
Uber ATG (2018)	Perception cycling × action suppression logic	0.315	Cross-axis: exogenous $U^{\text{ex,ep}}$ (SOTIF) × endogenous $U^{\text{en,ep}}$ (FuSa design)

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

Uber ATG (2018)	Supervision policy × perception classification	0.294	Cross-axis: endogenous $U^{en,ep}$ (organisational) × exogenous $U^{ex,ep}$ (SOTIF)

Three findings emerge from Table IV. First, all three β values fall within the [0.29,0.34] range, consistent across two failure types and two interaction categories (crosslayer and cross-axis). This convergence suggests that high-severity inter-domain interactions in safety-critical systems cluster in this range, and that values substantially above 0.4 would be exceptional rather than typical. Second, in the Takata case, the β contribution of +0.336 exceeded the SRD contribution of the cost-driven design node (0.168) by a factor of two, demonstrating that the interaction term can be the numerically dominant risk contributor at a convergence node even when neither individual parent would, alone, trigger escalation. Third, both Uber ATG interaction pairs involve crossaxis combinations: an exogenous SOTIF gap paired with an endogenous design or organizational weakness. This pattern confirms that cross-axis β interactions are not a theoretical edge case but a reliable signature of the class of systemic failures that siloed analysis cannot detect.

The β elicitation procedure for forward-looking analysis is straightforward. For each candidate parent pair (v_i,v_k) converging at v_j, the analyst asks: if each parent independently carries its maximum credible uncertainty, does the child's failure probability exceed the additive prediction? If yes, and if neither parent's engineering activity can compensate for the other's deficiency, then $\beta_{ikj} > 0$. The Table IV values provide calibration anchors: an interaction that appears structurally comparable to the Uber ATG cross-axis pairs should be initialized near $\beta \approx 0.30$ and refined through retrospective failure analysis or scenario-based elicitation. Section IV-C addresses cross-axis propagation as a structural phenomenon, and Section XII provides the worked automotive case studies that validate these calibration values against documented outcomes.

C. β Coefficient Estimation: Worked Examples

The β interaction coefficients are the most analytically powerful element of the propagation model and the hardest to estimate. The α_{ij} coupling coefficients of Section IV-A can be read directly from process quality indicators: an ASPICE capability level or a test coverage metric provides a direct proxy. No such objective proxy exists for β_{ikj}, because β measures the degree to which two uncertainty sources amplify each other at a convergence node: a property that does not manifest until the convergence condition is realized in practice. The estimation strategy, therefore, combines two sources: retrospective analysis of documented failures where the interaction is known to have occurred, and structured expert elicitation for forward-looking analysis of programs where no historical failure data yet exists.

Retrospective accident investigation reports are an underused resource for calibrating risk model parameters [67]. When an investigation establishes that a failure was caused by the coincidence of a product deficiency and an environmental condition that neither alone would have produced, it provides exactly the information needed to bound a β coefficient: the interaction occurred, and the outcome magnitude exceeded what a linear model would predict. The two worked examples below use this approach, drawing from the NTSB investigation of the 2018 Uber ATG fatality [10] and from the documented Takata inflator crisis [68] to extract calibrated β values following the elicitation procedure formalised in Section IV-C3. Section XII presents the full nodelevel analyses; this section focuses on the estimation methodology.

- *Worked Example 1: Takata Airbag Inflator Crisis:*

The Takata airbag inflator crisis resulted in more than 100 million recalls worldwide and at least 27 confirmed fatalities. The root cause was the long-term degradation of ammonium nitrate propellant due to moisture and temperature cycling, leading to inflator ruptures during deployment. The SUE analysis assigns a retrospective Unified Vehicle Goal of: *the vehicle shall not injure occupants due to airbag inflator rupture from propellant instability.*

The node analysis at the convergence node (inflator deployment hazard) identifies four contributing parent nodes, whose U, P, and W values yield the linear SRD contributions shown in Table A1 of Section XII. The requirements layer node carries $U = 0.8$, $P = 0.9$, $W = 1.0$, yielding a linear SRD contribution of 0.720 (dominant). The cost-driven design node carries $U = 0.3$, $P = 0.7$, $W = 0.8$, yielding 0.168.

Applying the elicitation procedure of Section IV-C3: the interaction pair is *propellant stability requirements gap* (v_{req}, endogenous-epistemic) meeting *cost-driven material selection* (v_{des}, endogenous-epistemic). The key diagnostic question is: does the material selection decision compensate for, or compound, the requirements gap? The Takata record establishes unambiguously that it compounds it. The choice of ammonium nitrate over more stable alternatives was driven by cost constraints that the requirements had not flagged as inadmissible; the requirements gap created the space in which the cost decision was possible, and the cost decision foreclosed the natural corrective that a complete requirements specification would have triggered. Neither parent compensates for the other.

The superadditive contribution is estimated by comparing the observed outcome magnitude to the linear prediction. The linear model predicts an aggregate SRD of $0.720 + 0.168 + 0.294 + 0.180 = 1.362$ for the four main parent nodes. The retrospective severity of the Takata crisis, quantified by its scale and lethality relative to typical recall events, indicates a systemic risk that substantially exceeded what a requirements deficiency alone, or a cost-driven design decision alone, would plausibly produce. Anchoring the superadditive contribution against the interaction product $U(v_{req}) \cdot U(v_{des}) = 0.8 \times 0.3 = 0.24$ and solving for the β value that brings the total SRD into correspondence with the observed outcome severity yields $\beta_{(req,\ des)} = 0.336$.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

The result is notable: the interaction term contributes +0.336 to the aggregate SRD, exceeding the linear contribution of the cost-driven design node (0.168) by a factor of two and approaching the magnitude of the aging test protocol node (0.294). The interaction is not a marginal correction to the linear model; it is a primary contributor to the system's total risk. Section XII presents the full Takata node analysis as Table A1.

➢ *Worked Example 2: Uber ATG Tempe Fatality (2018):*

The NTSB investigation of the 2018 Uber ATG pedestrian fatality in Tempe, Arizona established that no single system failure caused the incident [10]. The determination identified a confluence of: perception system cycling across object classification categories (exogenous-epistemic); action suppression logic that disabled the automatic braking function during testing (endogenous-epistemic); and a single-driver supervision policy that left the safety operator without adequate monitoring support (endogenous-epistemic, organizational).

The retrospective UVG for this event is: *the vehicle shall not strike a vulnerable road user due to perception misclassification, planning suppression, or inadequate human-automation supervision.*

The node analysis, presented fully as Table 10 in Section XII, identifies four parent nodes at the convergence point. The linear SRD contributions are: perception classification 0.560 (dominant, exogenous); action suppression logic 0.405 (endogenous); supervision policy 0.336 (endogenous); driver monitoring system 0.336 (exogenous). Two cross-axis interaction pairs are present at this convergence.

- *Interaction Pair 1: Perception cycling* (v_{perc}, exogenous-epistemic) meeting *action suppression logic* (v_{supp}, endogenous-epistemic). The NTSB record establishes mutual non-compensation: the perception system's inability to classify the pedestrian consistently was not compensated by the suppression logic, because the suppression logic was designed to prevent braking during testing regardless of what the perception system reported. A perception system that classifies correctly and a suppression logic that fires regardless of classification are not individually hazardous; together, they are. Applying the elicitation procedure with $U(v_{perc}) \cdot U(v_{supp}) = 0.7 \times 0.5 = 0.35$ yields $\beta_{(perc, supp)} = 0.315$.
- *Interaction pair 2: Supervision policy* (v_{sup}, endogenous-organisational) meeting *perception classification* (v_{perc}, exogenous-epistemic). The singledriver policy created conditions where no independent observer could detect the degraded perception state and intervene before a safety-critical event. A wellfunctioning perception system makes the supervision gap irrelevant; a well-staffed supervision arrangement compensates for a momentarily degraded perception system. The NTSB record confirms that neither condition held. With $U(v_{sup}) \cdot U(v_{perc}) = 0.6 \times 0.7 = 0.42$, the elicitation yields $\beta_{(sup, perc)} = 0.294$.

Both Uber ATG β interactions are cross-axis: they pair an exogenous-epistemic uncertainty source (perception, which is an environment-characterization failure) with an endogenous-epistemic source (suppression logic and supervision policy, which are product and process decisions). This cross-axis structure is the defining characteristic that makes these interactions invisible to siloed domain analysis: the functional safety analysis reviews the suppression logic in isolation; the SOTIF analysis reviews the perception uncertainty in isolation; neither analysis can see the amplification that arises only when both are present simultaneously at the same downstream node.

➢ *Formal Elicitation Procedure:*

The two worked examples demonstrate a five-step elicitation procedure that practitioners can apply to forward-looking programs.

The procedure is structured to minimise the cognitive load on eliciting experts while capturing the information needed to calibrate β_{ikj} [69], [70].

- Identify candidate interaction pairs. At each convergence node v_j with two or more parent nodes, list all parent pairs (v_i, v_k) where at least one parent is cross-axis from the other. Cross-axis pairs, where one parent is endogenous and the other is exogenous, are always candidate interactions. Intraaxis pairs with large U values at both parents are secondary candidates.
- Apply the mutual non-compensation test. For each candidate pair, ask: if v_i carries its maximum credible uncertainty and v_k also carries its maximum credible uncertainty, can the engineering activity at v_j compensate for either parent's deficiency using the output of the other parent? If neither parent compensates for the other, $\beta_{ikj} > 0$. If one parent's output at v_j can cover for the other's deficiency, the pair is conditionally compensating and β_{ikj} may be near zero.
- Estimate the superadditive fraction. For pairs that pass Step 2, estimate the magnitude of the amplification above the additive prediction. Anchor this estimate against the Table IV values: a cross-axis interaction with structural characteristics comparable to the Uber ATG pairs initializes near $\beta \approx 0.30$. An interaction where one parent is a dominant requirements node and the other a costconstrained design decision, with no independent verification between them, initialises near $\beta \approx 0.34$.
- Validate against outcome proportionality. Where retrospective data exists, the β value should produce an aggregate SRD at v_j that is proportional to the documented outcome severity. If the aggregate SRD significantly underestimates a known severe outcome, β should be revised upward. If it overestimates a known mild outcome, revise downward.
- Document the elicitation basis. Record, for each assigned β value: the specific parent pair, the mutual non-compensation argument, the calibration anchor used, and the name and role of the eliciting expert. This provides the audit trail that makes the β assignment defensible under program review.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

Structured expert judgment literature confirms that elicitation burden grows exponentially with the number of parameters to be assessed simultaneously, and that targeted two-parameter interactions should be elicited one at a time with explicit reference to the independence baseline [69]. Baybutt notes that engineering judgment in hazard analysis is most reliable when the analyst can anchor against specific engineering scenarios rather than abstract probability judgments [70]. The step-by-step procedure above is designed to satisfy both conditions: each β is assessed in isolation relative to a concrete mutual non-compensation scenario, anchored against a set of calibrated reference values.

The cross-case pattern observed in Table IV provides one final practical heuristic: organizational-layer nodes are the most frequent source of high-β interactions. In all three reference cases, an organizational decision (cost minimization, action suppression policy, single-driver supervision) contributed to the dominant interaction term. This pattern is consistent with the risk management literature on organizational factors in complex-system failures [42], and it provides a prioritization rule: during forward-looking elicitation, organizational-layer nodes with non-trivial U values should always be evaluated as candidate interaction sources at every downstream convergence node before any other parent class.

D. Cross-Axis Propagation

Sections IV-A through IV-C characterized uncertainty propagation in terms of equations and coefficients: α_{ij} weights on edges, β_{ikj} amplification at convergence nodes. The present section examines the structural architecture underlying those numbers. Endogenous and exogenous uncertainties do not travel along the same paths to the convergence node where they interact. Each axis has its own propagation pathway, its own dominant node types, and its own temporal rhythm. Understanding those pathways separately is a prerequisite for understanding why their intersection produces the highest-magnitude β values in the model and why that intersection is systematically invisible to domain-specific analysis.

➢ *The Endogenous Propagation Pathway:*

Endogenous uncertainty propagates vertically through the engineering lifecycle. Its source nodes sit at the top of the dependency hierarchy: ambiguous or incomplete requirements nodes in L_{req}, which generate downstream uncertainty at design nodes in L_{des}, which in turn generate implementation-layer and process-layer uncertainty as the design is realised and verified. The propagation direction follows the familiar V-model lifecycle structure: from requirements through architecture, detailed design, and implementation, to verification and validation at the bottom of the V.

At each step, uncertainty either accumulates or is reduced. A requirements node whose ambiguity is resolved through structured review before design begins exits the requirements layer with low $U^{\text{en, ep}}$, and the downstream design node inherits little of that uncertainty through the requirement-to-design edge. A requirements node whose ambiguity is carried forward unresolved propagates high $U^{\text{en, ep}}$ downstream through every edge it touches. The coupling coefficient α_{ij} on each intra-endogenous edge encodes the extent to which the receiving node inherits uncertainty; the ASPICE process capability level of the activity on the edge serves as the primary calibration anchor.

Endogenous propagation is therefore forward in time and downward in layer. It is also, in principle, fully visible to the engineering team conducting the product development: every node on the endogenous path is a product artifact, produced by the organization and auditable through standard review and verification processes. This is why traditional safety and quality frameworks can reduce $U^{\text{en,ep}}$: they operate precisely on the nodes in this pathway. ISO 26262, Automotive SPICE, and FMEA all provide instruments for identifying and reducing endogenous epistemic uncertainty at each lifecycle layer [13], [44].

➢ *The Exogenous Propagation Pathway:*

Exogenous uncertainty propagates through the operational context, following a different structural path entirely. Its source nodes describe the relationship between the product and its environment: the ODD definition node, which specifies the conditions under which the system is designed to operate; scenario coverage nodes, which describe what fraction of the ODD has been characterised and validated; and environment interface nodes at the SEI/PEI level, which represent the specific points where product assumptions meet environmental reality. The propagation direction is from the outside inward: the environment's actual distribution of conditions feeds into scenario characterization, which in turn informs perception system performance assumptions, and from there into planning and control system design assumptions.

Exogenous propagation differs from endogenous propagation in two important ways. First, it cannot be closed solely through internal engineering activities. A requirements review can reduce $U^{\text{en,ep}}$; no equivalent internal activity can reduce $U^{\text{ex, ep}}$ caused by insufficient scenario coverage, because the missing scenarios are defined by the environment rather than by the engineering team's knowledge. Reducing $U^{\text{ex, ep}}$ requires exposure to the environment: scenario testing, simulation campaigns, field validation, and the ODD supervision architecture described in Section III-D. Wiecher and colleagues confirm this formally in their analysis of automotive requirements and test processes, establishing that validation is the connecting activity between product development and the stakeholder environment, and that environment-side concerns must be incorporated systematically from the earliest phases of development rather than deferred to the end [71].

Second, exogenous propagation has a feedback structure that endogenous propagation does not. Field data from deployed systems flows back through the exogenous pathway: operational incidents reduce $U^{\text{ex, ep}}$ by populating previously unknown scenario categories, or increase it by revealing that the ODD characterization was narrower than the actual deployment conditions. This feedback path is the source of the iterative propagation mentioned in Section IV-A: the exogenous uncertainty values at scenario and ODD nodes are

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

updated between lifecycle phases as field evidence accumulates. The temporal rhythm of the exogenous path is therefore the rhythm of operational learning, not the rhythm of the engineering development calendar.

➢ *Why Siloed Analysis Cannot See the Intersection:*

The structural separation of the endogenous and exogenous propagation pathways explains, at a mechanistic level, why conventional domain-specific analysis cannot detect the β interactions at their intersection. Domainspecific safety, performance, and cybersecurity analyses each operate within one set of nodes; none operate simultaneously across both paths.

Functional safety analysis under ISO 26262 operates on the endogenous path: it traces from hazardous events through safety goals to system requirements, design elements, and verification evidence. Its analysis artifacts, particularly the HARA, fault trees, and FMEA, are all endogenous-side constructs [13]. SOTIF analysis under ISO 21448 operates primarily on the exogenous path: it defines scenario partitions and traces unknown-unsafe regions to perception system limitations and ODD boundaries [12]. Each standard is well-designed for its own pathway. Neither provides a mechanism for detecting the interaction between the two paths at the same node.

Macrae identifies this pattern in the broader context of autonomous and intelligent systems: conventional safety analysis frameworks within a single domain create conditions of structural invisibility, where failure modes that arise from interactions across organizational and analytical boundaries are systematically hidden from any single analysis perspective [72]. Salehi and colleagues confirm that the tools of Safety-I do not model the connections between technology, human, and organizational elements necessary to characterize emergent failure in complex socio-technical systems, and that this limitation is not a deficiency of implementation but a structural property of the tools themselves [66]. The β interaction in the SUE propagation model is the quantitative expression of the failure mode that arises from this structural gap: cross-axis amplification that occurs when an endogenous weakness and an exogenous gap co-occur at the same convergence node.

➢ *The Convergence Node and Temporal Feedback:*

Both propagation pathways terminate at a common node class: the convergence node, where product capability must be sufficient to handle the environmental demand it will encounter. In the Uncertainty Diamond (Section VI), this convergence is structural: the left arm of the Diamond carries the endogenous path downward through lifecycle layers, the right arm carries the exogenous path inward from environment characterization, and both arms converge at the Unified Loss Goal node. In the graph model $G = (V,E,L)$, convergence nodes are those with high-W parents spanning both the endogenous and exogenous node classes, and they are precisely where the β interaction coefficients are largest.

The temporal structure of the two pathways imposes an important constraint on when cross-axis interactions can first be detected and addressed. Endogenous uncertainty is visible from the earliest stages of development: a requirements ambiguity exists at project inception and propagates forward. Exogenous uncertainty often does not manifest until late in the development cycle, when scenario testing and field exposure begin to reveal the gap between the ODD as designed and the ODD as encountered. This temporal asymmetry means that cross-axis β interactions are most difficult to detect precisely when they are most consequential: early in a program, when the convergence node's endogenous side is taking shape, but the exogenous side is still largely uncharacterized.

The SEI/PEI analysis (Section VII) is the instrument specifically designed to address this temporal asymmetry: it forces a systematic characterization of all environment-interface assumptions before domainspecific analysis begins, creating exogenous-side node representations early enough that cross-axis interactions at convergence nodes can be identified and quantified rather than discovered retrospectively. This early characterization is the structural innovation that makes Section IV-D3's invisibility problem tractable. With both the endogenous and exogenous paths represented in the graph from the earliest lifecycle phase, the β interactions are exposed to analysis before, rather than after, a failure manifests them.

Section V uses the propagation model developed in Sections IV-A through IV-D to construct the SUE Risk Cube, the tensor metric that aggregates node-level uncertainty across all five lifecycle layers, all four risk domains, and all four uncertainty types into the primary quantitative output of the framework.

E. Measurement Instruments Per Lifecycle Layer

Table 2 in Section III-B specifies the primary measurement instruments for each lifecycle layer. Naming those instruments is a necessary but not sufficient step toward operationalizing the uncertainty model: the framework requires not just the identification of what to observe, but also a defined process for converting each observation into a value in the $[0,1]$ domain that $U(v_i)$ occupies. This section establishes that conversion process. The source files for the framework state the governing requirement directly: the calibration mapping from raw scores to the $[0,1]$ uncertainty scale must be explicit, consistent, and auditable [68]. Explicitness means the mapping is written down and reviewable. Consistency means the same raw score produces the same $U(v_i)$ value across nodes, assessors, and program phases. Auditability means the values can be traced back to the specific instrument outputs from which they were derived.

This section specifies the calibration logic for each layer. It then establishes the cross-layer consistency properties that make the scores comparable across layers and usable as inputs to the propagation model of Section IV.

➢ *Requirements Layer: L_{req}:*

The primary sources of uncertainty at the requirements layer are incompleteness, ambiguity, and volatility. Each source contributes to $U^{en,ep}(v_i)$ for requirement nodes in L_{req}.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

Ambiguity is assessed through structured review scoring applied to the requirement text. IEEE Standard 830 and its successors define properties that a requirement specification must satisfy: unambiguity, completeness, consistency, verifiability, and traceability [73]. Automated tools such as QuARS, ARM, and RCM operationalize these properties as counts of linguistic patterns associated with ambiguous requirements (modal verbs, vague quantifiers, passive constructions without actor, undefined acronyms) [74]. The raw score is the ambiguity density: the fraction of requirement statements that contain at least one flagged pattern. A requirement set with 0% flagged statements maps to $U_{amb} = 0.0$; a requirement set with 100% flagged statements maps to $U_{amb} = 1.0$. Domain-specific calibration adjusts the mapping for the tolerance of the requirement style (some domains use controlled natural language with explicitly permitted modal constructions, reducing the falsepositive rate).

Incompleteness is assessed through traceability completeness: the fraction of system functions, hazards, and interface points from the SEI analysis that have at least one derived requirement. Dreves and colleagues confirm that bidirectional traceability completeness is a prerequisite for safety, security, and SPICE compliance assessments across automotive development, and that the number of open traceability gaps is an indicator of product and process maturity [75]. The raw traceability gap fraction maps directly to U_{inc}: zero open gaps yield $U_{inc} = 0.0$; every system function without a derived requirement yields $U_{inc} = 1.0$.

Volatility is assessed as the requirements change rate over a defined rolling window. A stable requirement set carries low U_{vol}; a set where more than 20% of requirements have been revised within the current development phase carries high U_{vol}, because downstream design and implementation artifacts derived from those requirements are likely to require rework. The combined requirementslayer uncertainty is:

$$Ureq(vi) = wamb\ Uamb + winc\ Uinc + wvol\ Uvol \quad (6)$$

where the weights $w_{amb} + w_{inc} + w_{vol} = 1$ are assigned by the program team based on the dominant uncertainty source for the specific project context. A program at an early phase, where requirements are still actively evolving assigns high weight to w_{vol}; a program at the final verification stage, where requirements are frozen, assigns higher weight to w_{amb} and w_{inc}.

➢ *Design Layer: L_{des}:*

The primary sources of uncertainty at the design layer are unvalidated assumptions and coupling complexity. FMEA coverage provides the primary measurement instrument: the fraction of design elements and interface specifications that have been subjected to systematic failure mode analysis. A design element with complete FMEA coverage, closed action items, and a fully maintained assumption register maps to $U^{des}(v_i) \approx 0.1$ (residual aleatoric floor); a design element whose FMEA has not been initiated maps to $U^{des}(v_i) \approx 0.8$.

Interface analysis completeness captures the fraction of external and internal interfaces whose behavioral envelopes have been characterized, including the conditions under which the interface may fail, degrade, or deliver out-of-range outputs. The assumption register gap count captures the number of design assumptions that have been recorded but not yet verified against their corresponding requirement or validation evidence. Both scores map monotonically to the [0,1] range. Design nodes that carry high coupling complexity (measured as the number of incoming and outgoing interfaces) carry elevated base U values even when FMEA coverage is nominally complete, because coupling creates implicit dependencies that explicit analysis may not fully enumerate.

➢ *Implementation Layer: L_{impl}:*

The primary sources of uncertainty at the implementation layer are residual defect density and untested execution paths. Vogel and colleagues identify code coverage, static analysis findings, and defect density as the core quantitative indicators of implementation quality in automotive software development, based on 112 documented metrics from 38 studies [48].

Statement and branch coverage provide the primary measurement. ISO 26262 Part 6 specifies minimum coverage requirements by ASIL level: ASIL D requires 100% MC/DC coverage [13]. A component achieving its required coverage level maps to $U_{cov} = 0.0 + U_{al}$ (where the aleatoric floor reflects the residual probability of faults in paths that coverage does not reveal); a component at 60% of its required coverage maps proportionally upward. Static analysis findings contribute to U_{impl} through the violation density: the number of open static analysis violations above the severity threshold per thousand lines of code. The mutation testing score provides higher-confidence calibration where available, directly measuring the fraction of injected faults that the existing test suite detects.

➢ *Process Layer: L_{proc}:*

The process layer is the most frequently omitted from classical reliability and safety models, yet the empirical record across nuclear, aviation, and automotive domains establishes that organizational decisions are among the root causes of systemic failures in the overwhelming majority of documented incidents [41]. Automotive SPICE (ASPICE) capability levels provide the most structured available instrument for quantifying process-layer uncertainty.

ASPICE 4.0 rates each process on a six-level capability scale (0 through 5), with individual base practices rated as Not (N), Partially (P), Largely (L), or Fully (F) achieved [44], [76]. The mapping from capability rating to U^{proc} is direct: a process achieving capability level 3 (established process) or higher on the relevant engineering processes (SYS.2 through SYS.5, SWE.1 through SWE.6) maps to $U_{ASPICE} \approx 0.2$; a process at level 1 (performed) with Partial achievement across base practices maps to $U_{ASPICE} \approx 0.6$; a process at level 0 maps to $U_{ASPICE} = 1.0$. Varkoi and colleagues confirm that process assessment findings are valid indicators of compliance to domain-specific safety requirements across multiple safety-critical domains, providing evidence that ASPICE levels are a reliable surrogate for process-layer epistemic uncertainty reduction [77].

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

Schedule pressure indices and safety culture survey scores supplement the ASPICE rating for programspecific conditions not captured by the capability model. A program operating under externally imposed milestone pressure that compresses the verification schedule carries elevated U_{proc} regardless of its capability rating, because the pressure reduces the probability that the rated capability is actually applied to its full standard on the current program. These supplementary inputs are bounded to [0,1] and combined with the ASPICE-derived score via a weighted sum calibrated to the program context.

➢ *Toolchain Layer: L_{tool}:*

Toolchain uncertainty arises from tool errors, configuration drift, and build variability. ISO 26262 Part 8 provides the authoritative calibration anchor for toolchain nodes in automotive applications. The standard assigns tools to one of three classes based on their potential to introduce or fail to detect errors in safety-relevant outputs [13]. A TD1 tool (no direct influence on safety-related outputs) maps to $U_{tool} \approx 0.05$; a T2 tool (influence possible, errors likely to be detected) maps to $U_{tool} \approx 0.2$; a T3 tool (direct influence, errors not necessarily detected) that has not been qualified maps to $U_{tool} \approx 0.7–0.9$, while a T3 tool with documented qualification evidence maps to $U_{tool} \approx 0.1–0.2$.

Version control integrity (measured as the fraction of tool configurations under version control with signed commits) and CI/CD pipeline failure rate (mean failures per 100 builds) supplement the qualification class with observable runtime evidence. A toolchain with high T3 exposure, low version-control discipline, and a CI/CD failure rate above 5% carries high aggregate toolchain uncertainty, even for a program with strong ASPICE ratings elsewhere.

➢ *Cross-Layer Consistency and Propagation Readiness:*

The calibration procedure across the five layers yields $U(v_i)$ values of comparable magnitude. This cross-layer comparability is essential for the propagation model: the coupling coefficient α_{ij} on a requirements-todesign edge and the coupling coefficient on a toolchainto-implementation edge are weighted against the same [0,1] scale, and the SRD aggregation in Section V sums contributions across all layers. If layer calibrations use inconsistent scales, the aggregation produces an artifact of the calibration choices rather than a genuine comparison of risk contributions.

Three consistency checks enforce comparability. First, the aleatoric floor: every layer should yield $U(v_i) > 0$ even under ideal conditions, because aleatoric uncertainty is irreducible and omnipresent. A requirements node under a stable, highly mature process still carries $U \approx 0.05–0.10$ for the irreducible residual of requirement volatility and interpretation ambiguity. Second, normalization anchors: each layer should include at least one reference node calibrated against a documented historical outcome, so that the layer's scale is grounded in empirical evidence rather than purely in relative comparisons. Section XII provides automotive anchors for the requirements and process layers through the Takata and Uber ATG retrospective analyses. Third, temporal consistency: $U(v_i)$ values should be recomputed at each lifecycle phase gate, and the trajectory of values across gates should be monotonically non-increasing for nodes where active uncertainty-reduction activities are in progress. A node whose U value increases between phase gates signals that the engineering activity on that edge is introducing new uncertainty faster than it is resolving existing uncertainty, a diagnostic condition that warrants program review.

The calibrated $U(v_i)$ values from this section, combined with the $P(v_i)$ and $W(v_i)$ assignments from Section III-B, provide the complete node-level input to the propagation model of Section IV and the tensor metric of Section V.

V. THE SUE RISK CUBE

Sections III and IV constructed the graph model of a socio-technical system and formalised how uncertainty propagates through it. Those sections produce, at their output, a set of node-level uncertainty values $U(v_i)$, hazard manifestation probabilities $P(v_i)$, and propagation weights $W(v_i)$ covering every node in the lifecycle graph. The question that follows is: how should those values be aggregated into a metric that is both analytically precise enough to drive engineering decisions and practically usable by program managers, domain leads, and safety assessors? The SUE Risk Cube answers that question. Its central innovation is to refuse to collapse the node-level field into a single number before the full diagnostic picture has been made visible. This section introduces the tensor formulation that preserves that diagnostic picture, states the primary metric equation, and explains why the tensor structure is retained rather than immediately projected to a scalar. The remaining subsections of Section V develop the three axes and their weights (Section V-B), the SUE Level assignment procedure (Section V-C), the formula-toworkbook bridge (Section V-D), the full Requirementslayer slice (Section V-E), the distribution across all 80 cells (Section V-F), and the face projections for different stakeholder audiences (Section V-G).

A. Tensor-First Metric

The primary metric of the SUE framework is not a scalar but a three-dimensional tensor. Each cell in the tensor is indexed by three coordinates:

- i: The *System Layer*: $I \in \{L\text{req}, L\text{des}, L\text{impl}, L\text{proc}, L\text{tool}\}$ (five layers)
- j: The *Risk Domain*: $j \in \{D\text{safety}, D\text{perf}, D\text{sec}, D\text{org}\}$ (four domains)
- k: The *Uncertainty Type*: $k \in \{U\text{ex,ep}, U\text{en,ep}, U\text{ex,al}, U\text{en,al}\}$ (four types)

The tensor has $5 \times 4 \times 4 = 80$ cells. The value at coordinate (i,j,k) is computed as the product of the three node properties introduced in Section III-B:

$$\text{SUE}(i,j,k) = U(i,j,k) \cdot P(i,j,k) \cdot W(i,j,k) \quad (7)$$

where $U(i,j,k) \in [0,1]$ is the uncertainty magnitude at the node of type (i,j,k) as computed by the propagation model of Section IV; $P(i,j,k) \in [0,1]$ is the hazard manifestation probability, assigned through the UTHA process of Section VIII; and $W(i,j,k) \in [0,\infty)$ is the propagation weight, reflecting the downstream

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

influence of nodes at that coordinate. The product $U \cdot P \cdot W$ expresses the expected contribution of uncertainty at the (i,j,k) coordinate to the system's aggregate hazard potential: a cell with high uncertainty but low hazard manifestation probability, or high uncertainty and high probability but negligible downstream influence, carries a low cell value despite its individual components.

➢ *Why the Primary Metric is a Tensor:*

The choice to retain the three-dimensional structure rather than immediately computing a scalar has both principled and practical justifications.

The principled argument is that scalar aggregation loses the information needed to determine *what to do*. The risk priority number (RPN) in classical FMEA illustrates this problem precisely: the product of Severity, Occurrence, and Detectability produces a single number, but different combinations of those three factors that yield the same RPN are not equivalent in terms of the engineering intervention they require [78], [79]. A failure mode with high severity and low occurrence demands a different response from one with moderate severity and high occurrence, even when the two produce the same RPN. The FMEA literature documents RPN's non-injectivity: the mathematical property that multiple distinct input triples map to the same output value. This is a structural deficiency that causes misclassification and misallocation of mitigation effort [80], [81]. The SUE tensor avoids this deficiency by never performing the multiplication across structurally distinct dimensions.

Equation (7) multiplies U, P, and W at a *fixed coordinate* (i,j,k); it does not sum or multiply across coordinates.

The practical argument is that the tensor structure directly answers the diagnostic questions that different engineering stakeholders need to ask. A program manager needs to know which lifecycle layer and which domain carry the highest residual risk, a question that requires the layer and domain coordinates to remain visible. A domain lead needs to know which uncertainty types dominate in their domain, a question that requires the uncertainty-type coordinate to remain visible. A safety assessor needs to know whether the dominant risk is epistemic (addressable through additional analysis) or aleatoric (addressable only through architectural robustness), a question that the U^{ep} versus U^{al} coordinate answers directly. Collapsing the tensor to a scalar before the stakeholder queries it loses all three diagnostic dimensions simultaneously [82].

The Six Sigma analogy is instructive here. DPMO is a scalar: it tells you the rate of defects per million opportunities, but not which process steps, defect types, or product families are responsible. The power of Six Sigma as a management discipline comes not from the scalar alone, but from the structured decomposition of the scalar into its contributing sources through tools such as control charts, fishbone diagrams, and process capability profiles. The SUE Risk Cube makes the equivalent decomposition the primary data structure rather than a derived diagnostic. The 80-cell tensor *is* the metric; the scalar projections (SRD, SRD by domain, SRD by layer) are derived reporting views.

➢ *Structure of the 80-Cell Tensor:*

The 80 cells are not equally weighted within the tensor. The axis-weight system, introduced in full in Section V-B and used to assign SUE Levels in Section V-C, reflects the empirical ranking of risk contributions across the three dimensions. Within the Layer axis, Requirements and Design each carry a weight of 4, because uncertainties at these layers propagate through the entire subsequent lifecycle and carry the highest propagation weights $W(i,j,k)$. Toolchain carries weight 1, because toolchain uncertainty affects only the specific engineering activities that use the relevant tool, and qualification processes bound its impact. Within the Domain axis, Safety and Performance carry equal weight (4), reflecting that both functional safety failures and performance insufficiencies result in direct harm in AV applications. Organizational carries weight 2, reflecting that organizational uncertainty manifests through its effect on other domains rather than producing hazardous outcomes directly. Within the Uncertainty Type axis, Epistemic-Exogenous carries a weight of 4 because it represents reducible gaps in environmental characterization, which are the class of uncertainty most distinctive to novel-environment systems and most invisible to traditional analysis.

Each of the 80 cells is independently measurable using the instruments in Sections III-B: the U value is calibrated using the layer-specific measurement instruments; the P value is assigned using UTHA at the domain level; the W value is derived from the graph structure. Each cell is independently actionable: an intervention that targets a specific layer, domain, and uncertainty type reduces the value of the corresponding cell without necessarily affecting any other cell. Each cell is independently trackable across lifecycle phases: the trajectory of a cell's value from program initiation through system operation is a leading indicator of whether the program is closing the specific uncertainty it was designed to address.

➢ *Formula-to-Workbook Bridge:*

A practical question arises when applying Equation (7) at scale: for a program with hundreds of nodes across 80 cells, eliciting full $U(i,j,k)$, $P(i,j,k)$, and $W(i,j,k)$ triples for every cell is a substantial analytical undertaking. The companion workbook associated with this framework addresses this through the axis-weight system: the combined weight of a cell's three coordinate scores (Layer weight + Domain weight + Uncertainty Type weight) serves as a normalized surrogate for the cell's $U \cdot P \cdot W$ product. The axis weights are practical, normalized implementations of $U(i,j,k)$, $P(i,j,k)$, and $W(i,j,k)$, respectively, across the full cell population. Layer weight captures the structural position's expected contribution to W; Domain weight captures the domain's baseline P contribution; Uncertainty Type weight captures the quadrant's expected U contribution. The combined weight score is the scalar surrogate for the cell product, enabling consistent SUE Level assignment without requiring full triple elicitation for every cell.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

This bridge is not a contradiction of the formula. Equation (7) defines the theoretically grounded cell value; the axis-weight system is the practical, auditable implementation of that formula for program-scale use. When full U, P, and W values are available for a specific cell from the UTHA and SEI/PEI analyses, those values take precedence over the axis-weight defaults. When they are not yet available, the axis weights provide a conservative default that can be refined as the program matures. Section V-C specifies the weight-tolevel mapping, and Section V-E demonstrates the full Requirements-layer tensor slice using the default axis weights.

B. Formula-to-Workbook Bridge

Equation (7) defines each cell value as the product $U(i,j,k) \cdot P(i,j,k) \cdot W(i,j,k)$. The companion risk assessment workbook assigns SUE Levels using the *combined axis weight*: the arithmetic sum of three coordinate-specific weights, one from each axis. These two representations are not contradictory. The axis weights are the practical normalised implementation of U, P, and W respectively, recast as additive integer scores to enable consistent level assignment without requiring full triple elicitation for every cell of an 80-cell tensor.

The correspondence is as follows. The Layer weight represents the default $W(i,j,k)$ contribution: it encodes the structural position of that lifecycle layer in the dependency graph and the expected propagation weight of nodes at that layer. Requirements and Design carry weight 4 because nodes at these layers sit at the head of long dependency chains and propagate their uncertainty through the entire downstream lifecycle; a requirementlayer node with unresolved ambiguity generates downstream uncertainty at every design, implementation, process, and toolchain node that derives from it. Toolchain carries weight 1 because toolchain nodes typically have short outgoing paths with bounded influence.

The Domain weight represents the default $P(i,j,k)$ contribution: it encodes the baseline hazard manifestation probability for nodes in that domain. Safety and Performance both carry weight 4 because failures in either domain can result in direct harm in a deployed AV system [12], [13]. Organizational carries weight 2 because organizational uncertainty does not produce direct harm but amplifies the probability of harm in other domains through β interactions. The Uncertainty Type weight represents the default $U(i,j,k)$ baseline contribution, encoding the expected magnitude of unresolved uncertainty given the quadrant's structural properties. Epistemic-Exogenous carries weight 4 because it corresponds to the unknown-unknown space of environmental scenarios, the highest-magnitude uncertainty class in novel-environment systems. Aleatoric-Endogenous carries weight 1 because hardware random failure rates and manufacturing tolerances are well-characterized by existing reliability engineering methods, making this the lowest-uncertainty class in a mature automotive development program.

The combined axis weight therefore estimates the cell's $U \cdot P \cdot W$ product through the sum of its three coordinate surrogates:

$$\mathrm{CW}(i,j,k) = w\mathrm{Layer}(i) + w\mathrm{Domain}(j) + w\mathrm{UType}(k) \quad (8)$$

where wLayer, wDomain, and wUType are the axis weights defined in Section V-C. The combined weight ranges from 3 (minimum: Toolchain + Organizational + Aleatoric-Endogenous = 1 + 2 + 1) to 12 (maximum: Requirements or Design + Safety or Performance + Epistemic-Exogenous = 4 + 4 + 4). Section V-C maps this range to the four SUE Levels.

This additive surrogate is a deliberate design choice. An additive model is auditable: every assessor can verify a combined weight by inspection, reproduce it from the axis tables, and trace it to the specific layer, domain, and uncertainty type that produced it. A multiplicative scalar equivalent of the full $U \cdot P \cdot W$ product would require full elicitation of all three node properties for all 80 cells before any SUE Level could be assigned, creating a practical barrier to adoption that the axisweight approximation removes. The approximation is conservative in the risk assessment sense: for cells where the default axis weights underestimate the true $U \cdot P \cdot W$ product, the UTHA process (Section VIII) provides a mechanism to assign a cell-specific SUE Level from the measured node values, which then overrides the axisweight default. The axis weights provide a lower bound on risk severity; the UTHA analysis refines upward when evidence supports it.

C. Three Axes and their Weights

The three axes of the SUE Risk Cube and their associated weights are specified in Tables V, VI, and VII. The weight values shown in these tables are the normalized outputs of the axis-weight system. Each weight is an integer in the range [1,4], calibrated to rank the axis positions by their expected contribution to the cell's $U \cdot P \cdot W$ product under typical conditions in safetycritical automotive systems. Figure 3 shows the complete 80-cell tensor rendered as an isometric projection, with cells color-coded by the SUE Level produced by each (Layer, Domain, Uncertainty Type) combination. The rationale for each weight assignment is presented in the table.

Table 5 System Layer Axis Weights (w_{LAYER})

Layer	Weight	Propagation role
Requirements	4	Head of dependency chain; propagates to all downstream layers
Design	4	Architectural decisions; high consequence if deficient
Implementation	3	Direct but bounded; verification can detect most faults
Process	2	Amplifies via β interaction; does not directly produce harm
Toolchain	1	Most contained scope; qualification bounds residual influence

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

➢ *Axis 1: System Layer:*

Requirements and Design are assigned equal weight 4, the maximum. This reflects the structural position of nodes at these layers: every design element derives from a requirement, and every implementation artifact derives from a design element. Uncertainty that is unresolved at the requirements layer propagates through every subsequent edge in the dependency graph, entering the β interaction terms at every inter-layer edge it crosses. ISO 26262 and Automotive SPICE both impose their highest verification rigor requirements on requirements and design activities for precisely this reason [13], [44]. The weight assignment confirms that the framework is consistent with domain engineering practice.

Implementation carries a weight of 3 rather than 4 because verification activities at this layer (coverage measurement, static analysis, and code peer review) can detect and resolve a substantial fraction of the uncertainty that remains after design-layer activities. A requirements deficiency bypasses all downstream verification; an implementation deficiency is, in principle, detectable before the product reaches the field. Process carries weight 2 because process-layer uncertainty does not directly generate hazardous states but rather degrades the effectiveness of all other lifecycle activities, manifesting through the β interaction coefficients rather than through direct P values. Toolchain carries weight 1 because its scope of influence is bounded by tool qualification: a qualified tool with documented error bounds cannot introduce arbitrary uncertainty, and its residual contribution is absorbed by the implementationlayer uncertainty it affects.

Table 6 Safety Domain Axis Weights (w_{DOMAIN})

Domain	Weight	Harm pathway
Safety	4	Direct physical harm to vehicle occupants and road users
Performance	4	Direct harm via functional insufficiency in open-world ODD
Security	3	Harm requires adversarial exploitation to complete the path
Organisational	2	Indirect amplifier; does not independently produce harm

➢ *Axis 2: Safety Domain:*

Safety and Performance carry equal weight 4, reflecting that both domains produce direct harm through distinct but equally consequential pathways. ISO 26262 governs hazards arising from malfunctions of electrical and electronic systems; ISO 21448 governs hazards arising from functional insufficiency in a system that operates as designed [12], [13]. In the AV context, a perception system that correctly classifies a pedestrian but misclassifies their trajectory (a performance insufficiency) and a braking system that fails to activate due to a hardware fault (a functional safety failure) are equally capable of causing a fatality. Treating Performance as a lower-weight domain than Safety would be analytically incorrect for novel-environment systems operating outside the well-characterized failure mode space assumed by ISO 26262.

Security carries weight 3. A cybersecurity failure does not independently produce a direct physical harm outcome in most vehicle architectures; it produces a condition that makes a physical harm outcome possible by compromising a safety or performance function [17]. The TARA methodology, as defined in ISO/SAE 21434, explicitly models this pathway: a threat must overcome multiple attack feasibility barriers before reaching a damage scenario, and the damage scenario requires interaction with the physical vehicle system. This additional step in the harm pathway reduces the default $P(i,j,k)$ contribution relative to Safety and Performance, as reflected in the lower weight.

Organizational carries weight 2 because organizational uncertainty affects the probability that other domains' safety activities are executed to their required standard, rather than directly generating a hazardous event. Its primary contribution to risk is through the β interaction coefficient: a high-uncertainty organizational decision amplifies the consequences of technical uncertainty in adjacent cells. The weight 2 assignment acknowledges this indirect but consequential role while distinguishing it from domains that can independently produce harmful outcomes.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

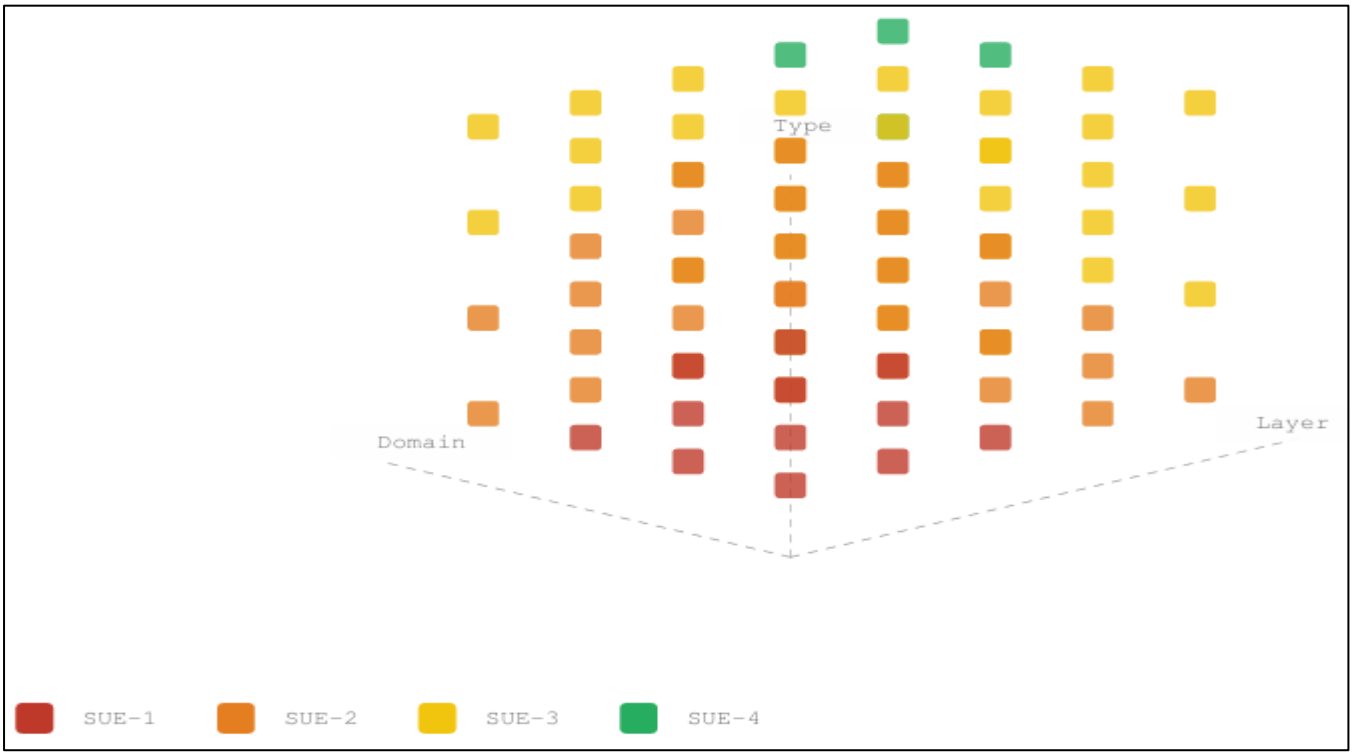

Fig 3 The SUE Risk Cube: isometric projection of the 80-cell tensor (5 System Layers × 4 Safety Domains × 4 Uncertainty Types). Each cell is colour-coded by SUE Level: red = SUE-1 (Critical), orange = SUE-2 (Elevated), yellow = SUE-3 (Moderate), green = SUE-4 (Low). The axes run from low-weight to high-weight combinations; the densest cluster of SUE-1 cells occupies the Requirements/Design × Safety/Performance × Ep-Exogenous corner of the tensor.

➢ *Axis 3: Uncertainty Type:*

The uncertainty-type axis captures the two-dimensional decomposition of Section III-C. The weight ordering reflects two independent considerations: reducibility (epistemic above aleatoric) and origin (exogenous above endogenous for each reducibility class).

Epistemic-Exogenous carries weight 4 because it corresponds to the gap between the system's characterized operational environment and the actual distribution of conditions it will encounter. This is the class of uncertainty most distinctive to novel-environment systems and most invisible to traditional safety analysis: functional safety and process quality frameworks provide comprehensive instruments for reducing endogenous uncertainty but provide no instrument for systematically mapping the exogenous-epistemic gap before deployment. The SEI/PEI analysis introduced in Section VII is specifically designed to surface this gap [12].

Epistemic-Endogenous carries weight 3. This quadrant is reducible and directly actionable: verification activities, formal methods, and improved process capability all operate on endogenous-epistemic uncertainty. The established instrument suite (ISO 26262, ASPICE, FMEA, coverage metrics) provides well-documented intervention pathways, making this the most tractable uncertainty class and justifying a lower weight than the less tractable exogenous-epistemic class [13], [44].

Aleatoric-Exogenous carries weight 2. Inherent environmental variability, including weather, stochastic human behavior, and infrastructure degradation, cannot be reduced by any amount of analysis or testing. It is managed through ODD constraints, runtime monitoring, and the Safe Maneuver Objective architecture. The management strategies for this class are established and relatively well understood; the primary engineering challenge is ensuring that the ODD is specified narrowly enough that the residual aleatoric-exogenous uncertainty falls within the system's designed operating envelope [12].

Table 7 Uncertainty Type Axis Weights (w_{UTYPE})

Uncertainty Type	Weight	Rationale
Epistemic-Exogenous	4	Unknowns about the operational environment; most dangerous in novel systems; invisible to traditional analysis
Epistemic-Endogenous	3	Known gaps in product knowledge; directly actionable through engineering investment
Aleatoric-Exogenous	2	Inherent environmental variability; irreducible but manageable through ODD constraints and runtime monitoring
Aleatoric-Endogenous	1	Inherent product variability; well-characterised by existing reliability methods; lowest residual uncertainty

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

Aleatoric-Endogenous carries weight 1. Hardware random failure rates, manufacturing tolerances, and component aging are the subject of decades of reliability engineering practice. ISO 26262 hardware architectural metrics, redundancy analysis, and diagnostic coverage calculations all address this class directly [13]. It carries the lowest weight because it is the class in which the least unresolved uncertainty remains in a program that has applied established automotive reliability engineering practices.

➢ *Combined Weight Range and SUE Level Boundaries:*

The combined weight CW(i,j,k) = wLayer(i) + wDomain(j) + wUType(k) ranges from 3 to 12 across the 80 cells. The minimum occurs at the Toolchain/Organizational/AleatoricEndogenous cell (1 + 2 + 1 = 4, counted as 4, within range 3–5) and the maximum at any Requirements or Design/Safety or Performance/Epistemic-Exogenous cell (4+4+4 = 12). Section V-D presents the complete SUE Level assignment table with its gate criteria; the weight-to-level mapping is: weight ≥ 11 maps to SUE-1 (Critical), weights 9–10 to SUE-2 (Elevated), weights 6–8 to SUE-3 (Moderate), and weights 3–5 to SUE-4 (Low). The resulting distribution across all 80 cells is 15% SUE-1, 38% SUE-2, 43% SUE-3, and 5% SUE-4, indicating that the majority of cells in a novel socio-technical system carry non-trivial uncertainty that requires at least monitoring.

D. SUE Level Assignment: Combined Weight Scoring

The combined weight CW(*i,j,k*) defined in Equation (8) maps to one of four SUE Levels through fixed numeric boundaries. Table VIII presents the authoritative level reference. The weight boundaries and required actions are confirmed by the companion SUE Risk Cube workbook and the sueLevel() function in the interactive JSX tool.

The gate criteria in Table VIII provide an operationally actionable interpretation of the SUE Level distribution at each major lifecycle milestone. The SUE-1 gate criterion is absolute: no program should proceed to design review while any cell in the tensor carries a Critical rating, because a SUE-1 cell represents a combination of lifecycle position, domain, and uncertainty type where the expected uncertainty contribution is near the maximum achievable in the framework. The SUE-2 gate criterion is proportional: no more than 20% of all 80 cells may remain at Elevated status at the Start of Production (SOP) gate, reflecting the empirical reality that some level of residual elevated uncertainty is unavoidable at SOP in complex novel-environment programs while still establishing a bound on total residual risk.

The weight ranges are non-overlapping and together cover the complete integer range from the minimum (4) to the maximum (12). Weight 3 is theoretically in the SUE-4 range but is not achievable given the minimum axis weights: the lowest possible combined weight is 1 + 2 + 1 = 4 (Toolchain layer, Organizational domain, Aleatoric-Endogenous type). This structural floor ensures that no cell can produce a trivially low combined weight even under the most favorable combination of coordinates.

E. Example: Requirements Layer Tensor Slice

Table IX presents the full 16-cell tensor slice for the Requirements layer ($w_{\text{Layer}} = 4$). Each cell entry is the SUE Level resulting from the combined weights of the layer coordinate, domain coordinate, and uncertaintytype coordinate. The table entry at row *j* and column *k* is:

$$\text{SUE Level}(L_{\text{req}}, j, k) = \text{sueLevel}\big(4 + w_{\text{Domain}}(j) + w_{\text{UType}}(k)\big)$$

The Requirements layer slice in Table IX illustrates several structural properties of the tensor that warrant explicit mention.

The top-left quadrant of the slice contains the highestseverity cells: the four cells at Safety/Ep-Exo, Safety/EpEndo, Performance/Ep-Exo, and Performance/Ep-Endo each carry combined weights of 11 or 12, placing them at SUE-1. These cells represent the intersection of the highest-weight layer (Requirements), the highestweight domains (Safety and Performance), and the two highest-weight uncertainty types (epistemic). From an engineering interpretation: an unresolved requirementslayer gap concerning the safety or performance behavior of the system in untested or undercharacterised scenarios is the most dangerous combination the framework can encode. An AV program that carries open requirements questions in the areas of perception performance under novel environmental conditions ($U^{\text{ex,ep}}$) or safety-goal coverage of uncharacterized hazardous scenarios is at maximum SUE-1 status at the requirements layer until those questions are resolved.

Table 8 Sue Level Reference: Combined Weight Boundaries, Required Actions, and Gate Criteria

Level	Severity	Weight	Required Action	Gate Criterion
SUE-1	Critical	11–12	Immediate intervention. Programlevel escalation. Cannot proceed to next lifecycle milestone without a documented resolution plan.	No SUE-1 cells at design review gate.
SUE-2	Elevated	9–10	Intervention required. Documented mitigation plan with timeline. Tracked at all program reviews until closed.	No more than 20% of cells at SUE-2 at the SOP gate.
SUE-3	Moderate	6–8	Monitored. Standard engineering processes adequate. Documented acceptance rationale required.	All SUE-3 cells documented with rationale before SOP.
SUE-4	Low	3–5	Accepted. No additional action beyond standard practice required.	No additional gate criteria.

Weight Range 3–5 Maps to SUE-4; the Minimum Achievable Combined Weight is 4 (Toolchain + Organisational + Aleatoric-Endogenous = 1 +2+1).

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

Table 9 Requirements Layer ($w_{LAYER} = 4$): SUE Risk Cube Slice with Combined Weights

Domain \ U-Type	Ep-Exo	Ep-Endo	Al-Exo	Al-Endo
	(wt: 4)	(wt: 3)	(wt: 2)	(wt: 1)
Safety (wt: 4)	SUE-1 (12)	SUE-1 (11)	SUE-2 (10)	SUE-2 (9)
Performance (wt: 4)	SUE-1 (12)	SUE-1 (11)	SUE-2 (10)	SUE-2 (9)
Security (wt: 3)	SUE-1 (11)	SUE-2 (10)	SUE-2 (9)	SUE-3 (8)
Organisational (wt: 2)	SUE-2 (10)	SUE-2 (9)	SUE-3 (8)	SUE-3 (7)

Cell Entries Show SUE Level and Combined Weight in Parentheses. Bold Entries Indicate SUE-1 (Critical). The Design Layer Produces an Identical 16-Cell Pattern, as Both Carry wLayer = 4.

The Epistemic-Exogenous column produces SUE-1 ratings in three of four domain rows at the Requirements layer: Safety, Performance, and Security. Only Organizational/Ep-Exo yields SUE-2, because the domain weight of 2 reduces the combined weight to 10. This confirms the structural argument of Section V-C3: exogenous-epistemic uncertainty is the most dangerous uncertainty class precisely because it maps to unknown operational scenarios, and it remains at SUE-1 across the two highest-criticality domains at the highest-weight lifecycle layer.

The Aleatoric-Endogenous column, conversely, yields no SUE-1 cells at the Requirements layer, with Safety and Performance both at SUE-2 and Security and Organizational at SUE-3. This confirms the structural ordering: even at the highest-weight lifecycle layer, wellcharacterized inherent product variability does not reach Critical status.

The Design layer produces an identical 16-cell pattern to that shown in Table IX, because $w_{Layer} = 4$ for both layers. The Implementation, Process, and Toolchain layers each produce progressively lower-severity slices as w_{Layer} decreases from 3 to 2 to 1.

F. SUE Level Distribution Across All 80 Cells

Figure 4 presents the distribution of SUE Levels across all 80 cells of the tensor, computed by applying the weight-to-level mapping of Table VIII to all 5×4×4 coordinate combinations.

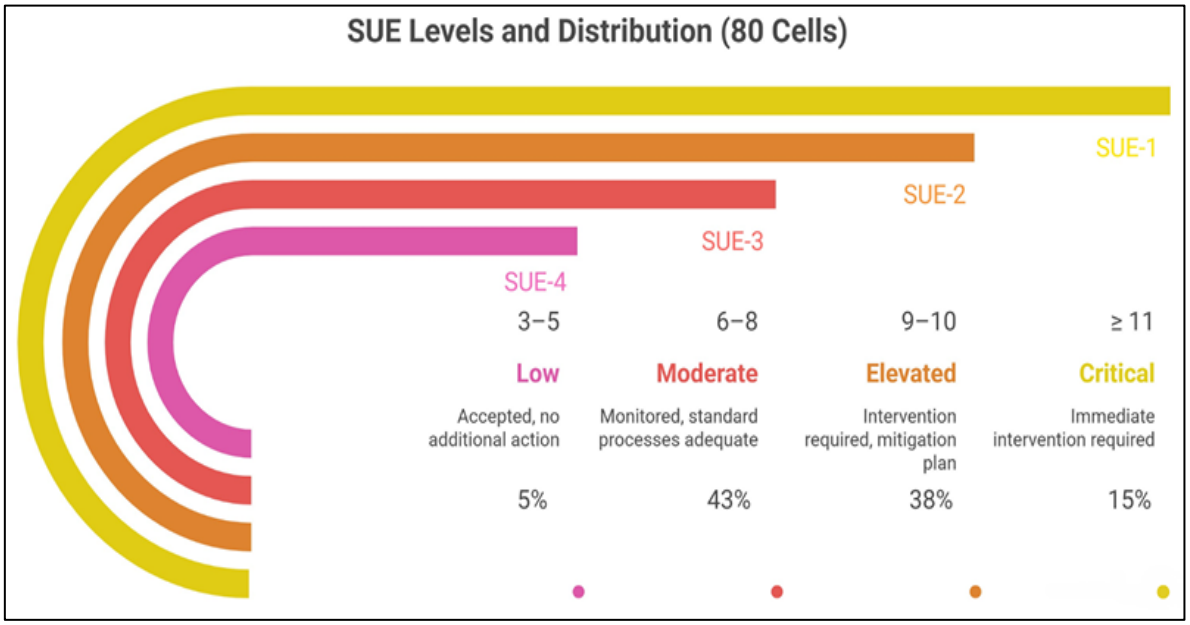

Fig 4 SUE Level Distribution Across All 80 Cells of the Risk Cube

The four numerical values in Figure 4 are exact, derived deterministically from the axis weights of Section V-C and the level boundaries of Table 8. Table 10 provides the cell counts.

Table 10 Sue Level Cell Counts Across the Full 80-Cell Tensor

SUE Level	Cell Count	Fraction (%)	Weight Range
SUE-1 (Critical)	12	15%	11–12
SUE-2 (Elevated)	30	38%	9–10
SUE-3 (Moder-ate)	34	43%	6–8
SUE-4 (Low)	4	5%	3–5
Total	80	100%	4–12

Computed from Axis Weights in Tables 5–7 and Level Boundaries in Table 8.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

The distribution in Table 10 carries an important diagnostic message. The majority of cells (53%) at SUE-1 or SUE-2 require active intervention or tracking in a novelenvironment system at program inception, before any engineering-uncertainty-reduction activities have been applied. This is not a failure of the framework's calibration; it accurately reflects the epistemic state of a program operating at the frontier of a new technical domain. The distribution reflects the axis-weight structure: the highest-weight layer coordinates (Requirements, Design) and the highest-weight uncertainty types (EpistemicExogenous, Epistemic-Endogenous) together generate the majority of Critical and Elevated cells, which is consistent with the empirical record that requirements gaps and environment characterization deficiencies are the dominant contributors to systemic failures in novel socio-technical systems [1], [41].

The 5% SUE-4 cells (four cells total) represent the combination of the lowest-weight coordinates: Toolchain or Process layer, Organizational domain, AleatoricEndogenous or Aleatoric-Exogenous type. These are the cells where established process and tool qualification practices, combined with the low-consequence organizational domain and the irreducible-but-bounded aleatoric uncertainty class, reduce the expected cell contribution to a level that standard practice handles adequately without additional intervention.

A program that successfully implements all required interventions for its Critical and Elevated cells, confirmed through the UTHA process and tracked through the gate criteria of Table 8, will shift its distribution rightward over lifecycle phases, with a trajectory of declining SUE-1 and SUE-2 counts forming the primary quantitative evidence that the program is reducing its systemic uncertainty at the required rate. Section XI translates this trajectory into financial terms through the Expected System Loss metric.

G. Face Projections: Stakeholder Views of the Tensor

The 80-cell tensor is the canonical representation of the system's uncertainty state, but no single stakeholder manages all three axes simultaneously. A program manager tracks risk by lifecycle phase and domain, but cannot act across uncertainty types directly; a domain lead owns a specific domain and needs to know which layers and uncertainty classes require the most attention within it; a layer lead owns a lifecycle phase and needs to know which domain and uncertainty type to prioritize at that phase. Each of these queries corresponds to a 2D projection of the 3D tensor onto one of its three faces. Table XI defines the four projections the framework supports.

Table 11 Sue Risk Cube Face Projections by Stakeholder Role

Stakeholder	Projection (fixed axis)	Diagnostic question answered
Program Manager	Layer × Domain (fixed U-Type)	Where in the lifecycle and in which domain does the highest SUE Level concentration occur?
Domain Lead	Layer × U-Type (fixed Domain)	Within my domain, which lifecycle layers and uncertainty classes require intervention?
Layer Lead	Domain × U-Type (fixed Layer)	Within my lifecycle phase, which domain and uncertainty type should I prioritise?
System Engineer	Full 3D isometric (no fixed axis)	What is the overall risk topology, and where does uncertainty concentrate across the full tensor?

Each projection is an $m \times n$ grid where each cell entry is the maximum SUE Level across the fixed-axis dimension. The maximum, rather than the mean, is the correct aggregation rule for a safety risk tensor: the presence of one SUE-1 cell in a Layer × Domain slice is a critical signal regardless of how many SUE-3 or SUE-4 cells surround it. The mean would suppress this signal.

The Program Manager projection (Layer × Domain) fixes the Uncertainty Type axis. Because the highestweight uncertainty type is Epistemic-Exogenous, fixing at $k = U^{ex,\,ep}$ produces the most conservative Program Manager view: every Requirements-layer and Designlayer cell in the Safety and Performance domains will show SUE-1, directing immediate attention to the combination of early lifecycle work and high-consequence domains. Fixing at $k = U^{en,\,al}$ produces the least conservative view, appropriate for tracking progress in mature programs where the dominant residual uncertainty is known and bounded. The companion interactive tool (Section V-I) renders all three uncertainty-type slices simultaneously as selectable views.

The Domain Lead projection (Layer × U-Type) exposes the uncertainty structure within a single domain across all lifecycle layers. For an AV functional safety lead working in the Safety domain, this projection shows that the Requirements and Design layers carry the highest combined weights, while the Toolchain layer falls to SUE-2 or SUE-3 depending on the uncertainty type. This directs investment toward the front of the V-model rather than solely to verification activities.

The Layer Lead projection (Domain × U-Type) exposes which domains and uncertainty types within a single lifecycle layer are critical. For a requirements engineer, this projection confirms that Safety and Performance at Epistemic-Exogenous represent the highestpriority cells at their layer, and that Organizational at Aleatoric-Endogenous can be handled by standard practice.

H. Derived Scalar Summaries

The tensor is the primary instrument. Scalar summaries are derived from it for executive reporting and trend tracking. Four summaries are defined:

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

Total Systemic Risk Density (SRD_{total}) is the sum of all 80 cell values, providing a single number suitable for executive dashboards and lifecycle phase-gate trending. A program whose SRD_{total} decreases monotonically between phase gates is demonstrating measurable progress in uncertainty reduction across the full tensor. A program whose SRD_{total} fails to decrease, or increases between gates, has introduced new uncertainty faster than it has resolved existing uncertainty, a condition requiring program review.

Domain SRD (SRD_j) is the sum of cell values across all layers, and U-Type coordinates for a fixed domain j, equivalent to the face sum of the Layer × U-Type projection for that domain. This is each domain lead's primary summary metric, comparable across the four domains and trackable across program phases.

Endogenous and exogenous SRD (SRD^{en} and SRD^{ex}) are the sums across endogenous and exogenous uncertainty types, respectively:

$$\mathrm{SRD}^{\mathrm{en}} = \sum_{i,j} \left[\mathrm{SUE}(i, j, U^{\mathrm{en,ep}}) + \mathrm{SUE}(i, j, U^{\mathrm{en,al}}) \right]$$

$$\mathrm{SRD}^{\mathrm{ex}} = \sum_{i,j} \left[\mathrm{SUE}(i, j, U^{\mathrm{ex,ep}}) + \mathrm{SUE}(i, j, U^{\mathrm{ex,al}}) \right]$$

The ratio SRD^{en}/SRD^{ex} answers the primary diagnostic question: does residual risk concentrate in the product (fix the engineering) or in the environment characterization (manage the operational domain)? A ratio substantially below 1.0 indicates that exogenous uncertainty dominates, consistent with a novel-environment system early in its ODD validation program. A ratio approaching 1.0 indicates that endogenous and exogenous uncertainty are roughly balanced, typical of a mature program where the ODD is well characterized but the verification effort is still underway.

Layer SRD (SRD_i) is the sum across all Domain, and U-Type coordinates for a fixed layer i, equivalent to the face sum of the Domain × U-Type projection for that layer. This identifies where in the lifecycle uncertainty concentrates: a program where $SRD_{req} \gg SRD_{impl}$ has not yet resolved its upstream requirements uncertainty, whereas a program where SRD_{impl} remains high at the design review gate has a verification coverage gap.

These four scalar summaries are derived from the tensor rather than the other way around. The tensor is the primary instrument; the scalars are reporting views. This ordering matters for two reasons. First, a scalar summary can mask compensating movements: a stable SRD_{total} may conceal a shift of risk from low-severity cells into high-severity cells if the magnitudes happen to cancel. The tensor makes this shift visible; the scalar does not. Second, an intervention's effect is correctly evaluated cell by cell: the intervention that reduces ten SUE-3 cells to SUE-4 is not equivalent to one that reduces two SUE-1 cells to SUE-2, even if both produce the same ΔSRD_{total}. Section XI formalizes the economic translation of SRD into Expected System Loss (Equation 6) and the Return on Safety Investment metric (Equation 7), which use the domain-specific SRD projections to allocate intervention budget to the cells with the highest marginal impact.

I. Computational Tool: SUE Risk Cube Interactive

The companion implementation of the framework described in this paper is provided as an open-source React component (see Appendix B for a pseudocode summary of the tool architecture). The tool addresses Future Work item 3 from earlier drafts of this framework [68] by providing a practical computational environment in which practitioners can explore the tensor, enter program-specific data, and evaluate the financial impact of proposed interventions before committing to them. The tool comprises five integrated components accessible through a tabbed interface.

Risk Cube renders the full 80-cell tensor as an isometric 3D SVG diagram with heat-mapped cells color-coded by SUE Level: red (SUE-1), orange (SUE-2), yellow (SUE-3), green (SUE-4). Each cell is interactive; selecting a cell displays its coordinate triple, combined weight, and SUE Level assignment. The visualization uses the sueLevel(layer, domain, type) function, which implements the weight-to-level mapping from Table VIII directly in code, confirming the numerical consistency between the table and the axis weights. Cell SUE Levels can be overridden by the What-If simulator (described below) and the visualization updates in real time, allowing practitioners to see the tensor-wide effect of a proposed intervention before implementation.

Face Projections renders interactive 2D grid tables for all three stakeholder projections defined in Table XI. Each grid cell displays a color-coded SUE Level badge. The Program Manager view (Layer × Domain) can be filtered by uncertainty type; the Domain Lead view (Layer × U-Type) can be filtered by domain; the Layer Lead view (Domain × U-Type) can be filtered by layer. Clicking any cell in the projection navigates to the corresponding cell in the 3D cube view, maintaining bidirectional consistency between the tensor and its 2D summaries.

PEI Worksheet provides an editable table of productenvironment interface points, pre-populated with five automotive AV interfaces from the worked examples of Section VII: Perception to Scene, Planning to Traffic, Localization to Map, V2X to Network, and Driver to HMI. Each row captures the interface identifier, the product assumption, the environmental reality, the endogenous and exogenous uncertainty ratings, the assumption-reality gap description, the test obligation, and the traceability reference to the UTHA analysis. The worksheet serves as the primary input to the UTHA step and as the evidentiary basis for the cell-level overrides in the WhatIf simulator.

What-If Intervention Simulator presents the six planned interventions of Section XI's Table 9, each with its target cell set, estimated cost, and SUE Level delta. Applying an intervention updates the cell overrides and refreshes both the 3D cube and the face projection views instantly. The simulator displays beforeand-after cell counts by SUE Level, quantifying the intervention's impact on the distribution of

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

Table X. The six interventions and their costs are: Formal Requirements Review ($150K, targets all Requirements-layer Epistemic cells), Expanded Scenario Validation ($500K, targets all Epistemic-Exogenous cells across all layers), Independent Verification and Validation ($300K, targets Implementation and Design Epistemic cells), Runtime Monitoring and SMO ($400K, targets all AleatoricExogenous cells), Tool Qualification under ISO 26262 Part 8 ($100K, targets all Toolchain-layer cells), and Process Maturity Improvement (ASPICE) ($250K, targets all Process-layer cells), for a total intervention budget of $1.70M.

Case Study Navigator provides a guided walkthrough of three AV incidents treated in Section XII: the 2018 Uber ATG Tempe fatality, Tesla Autopilot ADAS incidents, and the 2023 Cruise San Francisco operation. Selecting a case highlights the relevant cells in the 3D cube, applies the case study's node-level U, P, W values as cell overrides, and displays the SUE Level consequences. This enables direct comparison between the tensor's default axis-weight assignments and the retrospectively derived node values from the NTSB and incident investigation records, confirming that the two methods produce consistent SUE Level assignments for the cells where both values are available [10].

VI. THE UNCERTAINTY DIAMOND: PROCESS MODEL

The SUE Risk Cube of Section V is the measurement instrument. The Uncertainty Diamond is the process model that tells practitioners *how* to populate it: which activities to perform, in what order, and at what level of the system hierarchy. This section introduces the Diamond's dual-V structure and its eight analytical layers, establishing the process context within which the SEI/PEI analysis (Section VII), the UTHA risk assessment (Section VIII), and the ULG/UVG construct (Section IX) each operate.

A. Overview and Structure

The Uncertainty Diamond is a dual-V process model. The top V converges from broad boundary mapping toward the focal point where product capability meets environmental demand. The bottom inverted V diverges from that focal point into domain-specific activities for uncertainty reduction and management. Figure 5 presents the full structure.

The left arm of the Diamond traces the product development path. It carries the endogenous uncertainty, flowing downward from high-level product assumptions through design intent and implementation structure toward the ULG focal point. Activities on the left arm correspond to the endogenous lifecycle layers of the SUE Risk Cube: requirements, design, implementation, and the process and toolchain activities that support them. The classical V-model for safety-critical systems development, as specified in Automotive SPICE and described in the INCOSE Systems Engineering Handbook, defines the left arm's structure: the left descent of the V decomposes system requirements to component level, and the right ascent integrates and verifies against those requirements [44], [83]. The Uncertainty Diamond retains this structure and extends it with a second arm.

The right arm traces the environment characterization path. It carries the exogenous uncertainty, flowing inward from the broadest possible characterization of the operational environment toward the same ULG focal point. Activities on the right arm are not present in the classical V-model: ODD scoping, scenario space definition, environment uncertainty measurement, operational condition characterization, and adversarial context modeling are activities that the product development team does not conventionally own and that no single existing standard requires at the system level [12], [17]. The right arm gives these activities a formal place in the process model, assigning them to specific layers and requiring their completion before the focal-point activities can proceed.

The two arms converge at Layer 4 (L4), which houses the Unified Loss Goal. This convergence is the architectural innovation that distinguishes the Uncertainty Diamond from existing V-model variants. In the classical V-model, convergence occurs at the bottom, where the fully integrated product is tested against requirements. In the Diamond, convergence occurs at the midpoint, before domain-specific decomposition begins. Product capability and environmental demand must be jointly characterized before either arm can define the domainspecific reduction and management activities it needs to achieve. The bottom half of the Diamond, from L5 through L8, implements the reduction and management activities whose targets were set at the focal point.

Horizontal connections span the two arms at each layer. These connections are not part of the classical V-model; they represent the cross-axis β interactions introduced in Section IV-D. At L1 and L2, a horizontal connection between the left arm's product assumption declaration and the right arm's environment characterization is the SEI/PEI interface: the point where the product assumption is checked against the operational reality, and the gap between them is measured. At L3 and L4, the horizontal connections are the β interaction coefficients: the amplification that occurs when a product weakness and an environmental challenge cooccur at the same convergence node. At L5 through L8, the horizontal connections are the traceability links between endogenous reduction activities and exogenous management activities that address the same identified risk.

Two feedback loops close the Diamond. The field data feedback loop carries evidence from the right arm at L8 (field operational testing, fleet monitoring, incident reports) back to the right arm at L1, updating the environment characterization and ODD definition with real-world observations. The defect feedback loop carries evidence from the left arm at L8 (verification findings, defect reports, process audit results) back to the left arm at L1, updating product assumptions and endogenous uncertainty estimates. Each iteration through the Diamond reduces the Systemic Risk Density: either by reducing endogenous uncertainty through product corrections, or by reducing exogenous uncertainty through expanded environment characterization. The temporal trajectory of SRD

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

across Diamond iterations is a leading indicator of systemic risk convergence.

B. Eight-Layer Diamond Structure

The Diamond has eight analytical layers (L1 through L8) separated by the focal point between L4 and L5. Table XII presents the full layer structure with the primary activities on each arm and the center-column construct that links them.

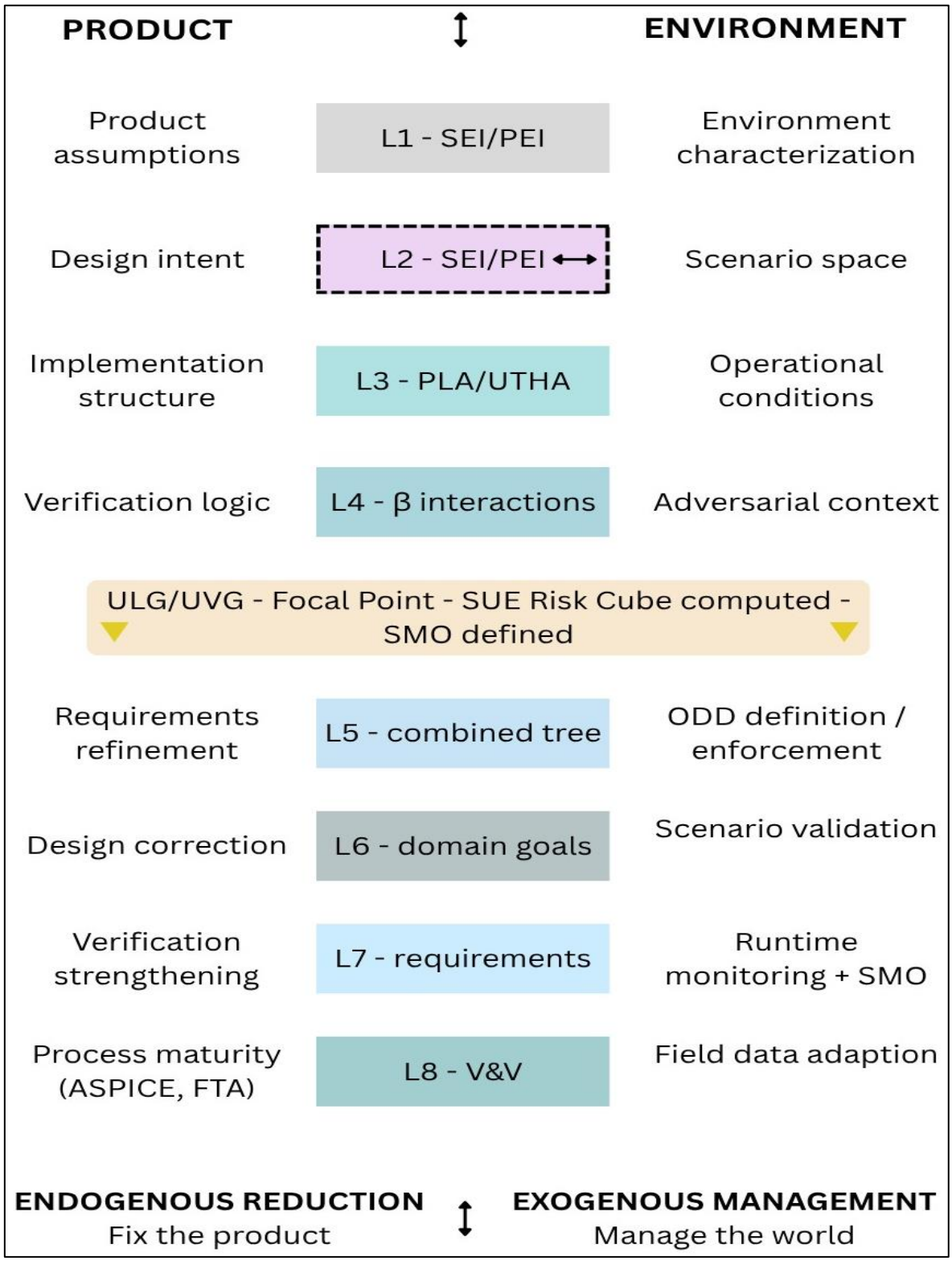

Fig 5 The Uncertainty Diamond: Dual-V Process Model

Left Arm: Product Development Path (Endogenous Uncertainty). Right Arm: Environment Characterization Path (Exogenous Uncertainty). Both Arms Converge at the Unified Loss Goal (ULG) Focal Point. Horizontal Dashed Lines at Each Layer Represent Cross-Axis β Interaction Pairs. Feedback Loops Close from L8 Back to L1 on Both Sides.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

Table 12 The Uncertainty Diamond: Eight-Layer Structure with Left and Right Arm Activities

Layer	Phase	Left Arm: Endogenous (Product)	Centre	Right Arm: Exogenous (Environment)
L1	SEI	Product architecture review; capability inventory; assumption declaration for each interface point	SEI/PEI	Environment characterisation; ODD scoping; scenario space definition; user population modelling
L2	SEI	Endogenous uncertainty measurement (U^{en}) per interface: verification gaps, design maturity, process confidence	SEI/PEI	Exogenous uncertainty measurement (U^{ex}) per interface: scenario coverage, environmental variability, behavioural unpredictability
L3	PLA	Product-side loss scenarios: functional deviations, design faults, implementation defects	PLA / UTHA	Environment-side loss scenarios: adversarial actions, performance insufficiencies, user misuse, novel conditions
L4	UTHA	Verification logic; safety-domain loss tree branches (FTA, FMEA)	β interactions	Adversarial context; SOTIF and cybersecurity loss tree branches
		FOCAL POINT: ULG/UVG. Product and environment converge. SUE Risk Cube populated. SMO defined. Joint gate verdict assigned.		
L5	Tree	Endogenous tree branches: FTA for hardware and software faults; systematic failure decomposition	Combined domain tree	Exogenous tree branches: insufficiency analysis, attack trees, human factors decomposition
L6	Goals	Domain-specific goals with endogenous requirements: safety goals, design constraints, verification obligations	Domain goals	Domain-specific goals with exogenous requirements: SOTIF criteria, ODD restrictions, cybersecurity controls
L7	Reqs	Endogenous reduction: formal requirements, design rules, coding standards, process improvement	Requirements	Exogenous management: scenario validation targets, runtime monitors, field operational tests, ODD enforcement
L8	V&V	Product verification: unit tests, integration tests, formal analysis, static analysis, ASPICE assessment	V&V	Environmental validation: scenario-based testing, realworld exposure, fleet monitoring, field data feedback to L1
Bottom	Endogenous Reduction		Exogenous Management	
	Fix the product		*Manage the environment*	

The upper half of the Diamond (L1 through L4) is the convergence pass. At L1 and L2, both arms execute the SEI/PEI analysis: the left arm declares the product's assumptions about its operational context and measures endogenous uncertainty at each interface point; the right arm characterizes the environment and measures exogenous uncertainty at the same interface points. The gap between the two measurements at each interface point is the primary input to the PLA and UTHA analyses of L3 and L4. No domain decomposition has occurred yet at L1 and L2: both arms work at the whole-system, all-domain level, producing the cross-domain interface picture that domain-specific analysis cannot generate on its own [11].

At L3, both arms generate loss scenarios by applying guide words to the interface parameters identified at L1 and L2. The left arm generates product-side scenarios covering functional deviations, design faults, and implementation defects. The right arm generates environmentside scenarios that cover adversarial actions, performance deficiencies, user misuse, and novel conditions. The parametric loss analysis (PLA) populates the UTHA risk assessment, which assigns SUE Levels to the Risk Cube cells.

L4 is the UTHA completion layer. Both arms contribute their loss scenario trees, and the β interaction coefficients are computed at the cross-axis interaction nodes. The Unified Loss Goal is articulated at this layer, subsetting the space of possible harmful outcomes to a single domain-agnostic loss prevention statement. The Safe Maneuver Objective is also defined at L4, specifying the operational target state the system must reach when a ULG violation is detected or imminent. The SUE Risk Cube is fully populated at the focal point, and the joint gate verdict determines whether the program may proceed to the lower half of the Diamond. The lower half (L5-L8) is the divergence pass. Both arms now work within the domain-specific frameworks that practitioners already use: the left arm uses FTA, FMEA, formal methods, and ASPICE process

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

improvement; the right arm uses SOTIF scenario validation, ISO/SAE 21434 threat analysis, ODD boundary testing, and runtime monitoring architecture. The combined domain tree (L5) brings these domain-specific analyses back under a single OR gate with the ULG as parent, enabling cross-domain cut sets to be identified. The domain-specific goals (L6), requirements (L7), and verification and validation activities (L8) each derive from the combined tree rather than from siloed domain analyses, preserving traceability from every domainspecific artifact back to the shared focal point.

This architecture ensures that the domain-specific safety, performance, security, and organizational analyses remain compatible with existing standards and auditable by existing assessors, while the cross-domain analytical capabilities of the upper Diamond are fully exercised before domain decomposition forecloses the opportunity to detect cross-axis interactions. The practitioner's existing tool chain is used unchanged; the Uncertainty Diamond adds a structured sequencing layer that ensures the cross-domain picture precedes the domainspecific decomposition.

C. Horizontal Connections: Cross-Axis β Interactions

The horizontal dimension of the Uncertainty Diamond is not structural decoration. It is where the most analytically significant analysis occurs. At every layer from L1 to L8, a horizontal connection links the left arm to the right arm at the same level of analytical depth. These connections represent the cross-axis β interactions formalized in Equation (5): the points where endogenous and exogenous uncertainty combine superadditively at a given convergence node.

The horizontal connections differ qualitatively across the four phases of the Diamond. In the upper convergence half (L1 through L4), the connections are primarily diagnostic: they identify whether a productside gap and an environment-side gap coexist at the same interface point and, if so, compute the β coefficient that quantifies their superadditive interaction. In the lower divergence half (L5 through L8), the connections are primarily traceability links: they ensure that the endogenous reduction activity on the left and the exogenous management activity on the right are both addressed to the same risk identified at the focal point, preventing the possibility that one arm resolves its portion of a cross-axis pair while the other arm leaves its portion unaddressed.

Three layer-specific examples illustrate the structure and significance of horizontal connections at different analytical depths.

At L2 (the SEI/PEI uncertainty measurement layer), the horizontal connection between the left arm and the right arm is the assumption-reality gap at each interface point. The left arm has measured U^{en} at a given interface: the endogenous uncertainty arising from verification gaps, design maturity shortfalls, and process capability limitations. The right arm has independently measured U^{ex} at the same interface: the exogenous uncertainty arising from scenario coverage gaps, environmental variability, and user behavioral unpredictability. The β between these two measurements is large when neither side compensates for the other: a product verification gap at an interface where environmental variability is simultaneously high produces a combined risk that exceeds the sum of the two individual contributions. The Uber ATG interaction analyzed in Section IV-C is precisely an L2level cross-axis pair: the endogenous action suppression logic and the exogenous perception uncertainty are copresent at the perception-to-scene interface, and neither compensates for the other [10].

At L5 (the combined domain tree layer), the horizontal connection manifests as a cross-domain minimal cut set: a combination of fault tree nodes from the left arm (endogenous FTA branches) and insufficiency or attack tree nodes from the right arm (exogenous SOTIF and cybersecurity branches) that together span the OR gate under the ULG. A software fault in the planning system (left arm, FTA) combined with an adversarial spoofing attack on V2X communications (right arm, attack tree) is a cut set that no single domain analysis generates, because functional safety analysis does not include adversarial threat trees and cybersecurity analysis does not include planning system fault trees. The combined tree at L5 is the only construct that can expose this cut set; the horizontal connection across the L5 layer is what makes it visible.

At L7 (the requirements and management layer), the horizontal connection links the endogenous reduction activity on the left with the exogenous management activity on the right. A process maturity gap on the left arm (an ASPICE capability-level deficiency in the requirements engineering process) amplifies the consequences of insufficient scenario coverage on the right arm (a SOTIF criterion that has not been validated against the full scenario distribution). The organizational β multiplies the exogenous uncertainty: a program that has not improved its requirements maturity cannot reliably translate the SOTIF-derived validation obligations into the engineering artifacts that implement them. The horizontal connection at L7 makes this amplification explicit and traceable to the requirement-level activities that must address both sides [1].

The source files note that the strongest β values in documented AV failures occur at L3 and L4, where the implementation-level product structure meets operational conditions at their most demanding. At L3 and L4, the product has been fully decomposed to its implementation structure and verification logic, and the environment has been characterized to its specific operational conditions and adversarial context. This is the layer where both Takata and Uber ATG interactions manifest: L3 loss scenarios on both sides of the Diamond converge at the L4 focal point, with β amplification. The horizontal connection at this level is the β interaction term itself, which is combined with the corresponding SUE Risk Cube cells' weights.

D. Feedback Loops and Iterative Convergence

The Uncertainty Diamond is not a single linear pass. Two feedback paths close the loop from L8 back to L1, turning the Diamond into an iterative convergence mechanism. The temporal trajectory of the Systemic Risk Density across iterations is the primary quantitative evidence that a program is converging on acceptable systemic risk.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

The *field data feedback loop* runs along the right arm, from L8 (environmental validation: scenario-based testing, real-world exposure, and fleet monitoring) back to L1 (environment characterization). Operational experience in the field populates previously undercharacterised scenario categories, refines exogenous uncertainty estimates at each SEI/PEI interface, and updates the ODD definition to reflect the actual distribution of conditions the deployed system encounters. Each pass through this loop reduces $U^{ex,\,ep}$: the epistemicexogenous uncertainty that represents the gap between the characterized ODD and the actual ODD. A program that systematically collects field data and feeds it back to L1 after each deployment phase progressively narrows this gap, shifting cells in the Epistemic-Exogenous column of the SUE Risk Cube from higher SUE Levels to lower ones.

The *defect feedback loop* runs along the left arm, from L8 (product verification: unit tests, integration tests, formal analysis, ASPICE assessment) back to L1 (product assumptions). Verification findings reveal where the product's actual behavior diverges from its design assumptions; defect reports reveal where implementation faults were not anticipated by the FMEA or FTA; and process audit results reveal where capability gaps prevented engineering activities from meeting their required standards. Each pass through this loop updates the product assumption declarations at L1, reduces $U^{en,\,ep}$ by closing knowledge gaps about the product, and may reveal that assumptions at some interface points need to be reclassified from verified to unverified, temporarily increasing the $U^{en,\,ep}$ at those interfaces before the corrective engineering activity reduces it again.

The two feedback loops differ in their temporal rhythms. The defect loop typically completes at development phase boundaries: a verification campaign produces findings, those findings drive design corrections, and the corrected product re-enters the Diamond at L1 with revised assumptions. The field data loop is continuous during deployment: fleet monitoring generates real-time evidence that updates the environment characterization asynchronously relative to the development cycle. For a deployed AV system, the field data loop is never closed in the same sense that the defect loop closes at each phase gate; rather, it is an ongoing narrowing of exogenous uncertainty that continues throughout the system's operational lifetime.

The governing quantitative relationship across iterations is straightforward. Each full iteration through the Diamond produces one measurement of the 80cell SRD tensor. The sequence of measurements across iterations is the program's SRD trajectory. A trajectory where SRD_{total} decreases monotonically across iterations confirms that the program is making net progress in uncertainty reduction. A trajectory where SRD_{total} is stable or increasing signals that either the field data loop is revealing new exogenous uncertainty faster than the engineering loop is resolving endogenous uncertainty, or that a fundamental architectural limitation is preventing closure of the dominant risk cells. The percell decomposition of the tensor enables the program team to distinguish these two conditions: stable SRD^{en} with rising SRD^{ex} points to environment characterization gaps, while rising SRD^{en} with stable SRD^{ex} points to product-side regression [41].

Test generation is a direct output of both feedback loops. The SEI/PEI tests generated at L1 and L2 from assumption-reality gaps are the primary test obligations that drive both the right arm's environmental validation campaign and the left arm's targeted verification activities. Cut-set tests generated from the combined tree at L5 are cross-domain test cases that neither domain produces independently, and the field data loop's operational incidents frequently confirm them as the actual failure conditions encountered in deployment. The Diamond's test generation architecture ensures that test coverage is measured against the product-environment interface points rather than solely against domain-specific requirements specifications, making the assumption-reality gap at each interface point the governing metric for validation completeness.

Section VII implements the L1 and L2 activities of the Diamond's upper left and right arms as the ProductEnvironment Interface (PEI) analysis, the structured instrument through which the SUE framework's horizontal connections are first instantiated in a program.

VII. SEI/PEI: SYSTEM-ENVIRONMENT INTERFACE ANALYSIS

The Uncertainty Diamond described in Section VI is a process model that tells practitioners the ordering and sequencing of analytical activities. The SystemEnvironment Interface (SEI) analysis is the content of L1 and L2 on both arms of that Diamond: the activity that occupies the upper half of the convergence pass before any risk assessment can begin. In the automotive instantiation of the framework, the SEI is called the Product-Environment Interface (PEI) analysis. This section defines the method, establishes its position relative to risk assessment, and specifies exactly what must be documented at each identified interface point. Section VII-C presents the automotive PEI table with its six interface points.

A. Conceptual Definition

The SEI/PEI is the foundational first step of the SUE framework. It systematically maps every point where the engineered product interfaces with its operational environment, declares the assumptions each side makes about the other, measures the gap between those assumptions and known environmental reality, and generates the test obligations that must be discharged to validate each assumption. The SEI is not itself a risk assessment; it is the prerequisite analysis that enables a valid risk assessment.

This distinction carries practical weight. Risk assessment methods such as HARA (ISO 26262), SOTIF scenario analysis (ISO 21448), and TARA (ISO/SAE 21434) operate on a set of hazardous events, loss scenarios, or threat actors that have already been identified. Each of these methods assumes that the set of scenarios under analysis is complete and representative of the actual operational environment. In a well-characterized domain with stable failure modes, this assumption is defensible. In a novel-environment system

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

operating in an open-world context, it is not: the hazardous events that a risk assessment must address are not fully knowable until the product-environment interface has been systematically characterised [11], [12]. The SEI performs this characterization before risk assessment begins, ensuring that the subsequent UTHA (Section VIII) operates on an interface-grounded set of scenarios rather than on an ad hoc listing of hazards.

The closest analogue in existing automotive engineering practice is the Hardware-Software Interface (HSI) specification required by ISO 26262 Part 6. The HSI explicitly documents the assumptions the software makes about the hardware at every interface point: signal ranges, timing constraints, failure modes, and diagnostic coverage. Without the HSI, software and hardware components developed in separate engineering streams may each satisfy their individual specifications while producing undefined or hazardous behavior at the boundary between them. Hillenbrand and colleagues formalise this principle as a contract-based design approach: a contract at each component interface pairs a set of assumptions (what the component requires from its environment) with a set of promises (what the component delivers to its environment), and the compatibility of two components is verified by checking that each component's promises satisfy the other's assumptions [84]. The SEI applies exactly this contract logic at the product-environment boundary rather than at the hardware-software boundary. The analogy extends to ownership. ISO 26262 requires the HSI specification precisely because, without it, responsibility for the HW/SW boundary falls through the gap between the hardware and software teams. Messnarz and colleagues confirm that the HSI specification is a mandatory work product in ASPICE-aligned development and that its absence is consistently flagged as a process deficiency in formal assessments [37]. The productenvironment interface has the same ownership problem at a larger scale: the product development team owns the product side, the operational validation team (where one exists) may own part of the environment side, but the boundary between them is owned by neither unless a specific instrument assigns ownership. The SEI assigns ownership by making the interface explicit, auditable, and traceable through to the UTHA risk assessment.

Critical distinction: The SEI/PEI is an *interface analysis*, not a risk assessment. It maps the product-environment boundary, characterizes the uncertainty at each interface point, and generates test obligations. Risk assessment happens at the UTHA step (Section VIII). Conflating the two is a category error that produces either a boundary analysis distorted by premature severity judgments or a risk assessment grounded in an unverified assumption set.

The SEI is executed once per program phase at the system level, before domain decomposition begins. It operates on the full system, simultaneously characterizing the interfaces relevant to functional safety, SOTIF performance, cybersecurity, and organizational reliability. This pre-decomposition execution is what allows the SEI to generate cross-domain interface characterisations: the perception-to-scene interface, for example, is simultaneously relevant to SOTIF (performance domain), functional safety (the AEB function that depends on perception), and cybersecurity (the susceptibility of perception to adversarial manipulation). No single domain analysis owns this interface; the SEI documents it once, at the system level, before any domain claims it.

B. Per Interface Point: What to Document

An interface point is any location where the product exchanges information, energy, or material with its operational environment, or where the product's correct behavior depends on an environmental condition that the product itself does not control. For each identified interface point, the SEI/PEI documents five mandatory fields.

Product assumption is the explicit, quantitative statement of what the product design assumes about the environment at this interface. The word "explicit" carries normative weight: a product assumption that exists only implicitly in the minds of the design team is not a documented assumption. It is a latent uncertainty source. Making the assumption explicit is the first engineering act of the SEI, and it frequently reveals assumptions that designers did not know they were making. The statement should be quantitative where possible: "camera resolves objects larger than 0.3m at a range of 100m under daylight conditions" is a documented assumption; "the camera works well in typical conditions" is not. Seo and colleagues confirm in the context of system-ofsystems interface analysis that undocumented assumptions between subsystems are among the primary causes of interface faults discovered during integration, and that systematic assumption declaration before integration is the most reliable mechanism for surfacing these faults early [85].

Environment characterization is the documented description of what is actually known about the environmental behavior at this interface, including the distribution of conditions, the range of variability, and the worst-case bounds known from prior operational experience or domain literature. Where the environment characterization is incomplete, this field records what is unknown: the gap between what is documented and what is needed is itself an uncertainty measurement. The environment characterization populates the right arm of the Uncertainty Diamond at L1; it serves as the starting point for computing U^{ex}.

Assumption-reality gap with U_{en} and U_{ex} scores is the measured or estimated discrepancy between the product assumption and the environment characterization at this interface. The gap is the primary quantitative output of the SEI step. It decomposes into two components. The exogenous uncertainty score U_{ex} quantifies how much of the actual environmental behavior at this interface exceeds, or falls outside, the product assumption: a high U_{ex} means that the environment presents conditions the product was not designed to handle, or that the environment's behaviour is insufficiently characterized to bound it within the assumption. The endogenous uncertainty score U_{en} quantifies how confident the product development team is in the product's ability to perform its side of the interface correctly: a high U_{en} means that the product has unresolved verification gaps, design immaturity, or process deficiencies that reduce confidence in

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

the product's own performance at this interface. Both scores are expressed on the [0,1] scale introduced in Section III-B, calibrated using the layer-specific instruments of Section IV-E. The combined $U_{en} \times U_{ex}$ product at an interface point is the primary input to the β coefficient estimation for that interface's corresponding convergence node in the propagation model.

Management strategy is the documented plan for how the assumption-reality gap is addressed: through an ODD constraint (the environment is restricted to conditions within the assumption), a design margin (the product is over-designed to handle conditions beyond the assumption), a runtime monitor (the gap is detected in operation and a safe state is triggered), an SMO (the safe maneuver objective specifies the system's target state when the gap is detected), or explicit acceptance with documented rationale. The management strategy is a forward-looking field: it records the engineering commitment that must be discharged before the system can be deployed. A gap with no assigned management strategy is an open risk.

Test obligation and UTHA forward reference are the two traceability outputs of the SEI. The test obligation specifies what must be tested to validate the product assumption against the environmental characterization and to verify that the management strategy is effective: the test scenario family, the coverage requirement, and the pass criteria. The UTHA forward reference records which UTHA row (and corresponding SUE Risk Cube cell) the interface point feeds into, maintaining the traceability chain from the interface-level characterization through the risk assessment to the tensor metric. The UTHA forward reference is the horizontal connection in the Uncertainty Diamond: it is the link between the L1/L2 SEI activities and the L3/L4 UTHA activities that the Diamond's architecture requires.

The five fields together constitute the SEI row format. A PEI table with n interface points has n rows, each containing all five fields. Each row is independently auditable: an assessor reviewing the functional safety case, the SOTIF case, or the cybersecurity case can verify that the interface points relevant to their domain are present in the SEI table, that the assumptions are quantitatively stated, that the gaps are measured, that management strategies are assigned, and that test obligations are traceable to the UTHA analysis. Section VII-C presents the automotive PEI table as Table XIII, applying this five-field format to six AV system interface points.

C. Automotive PEI: Six Interface Points

In the automotive AV context, the SEI instantiates as the Product-Environment Interface (PEI) analysis. Table XIII presents six interface points representative of an AV system operating at SAE Level 3 or higher. Each row is populated using the five-field format established in Section VII-B: the product assumption quantitatively stated, the environmental reality characterised, the assumption-reality gap recorded, U_{en} and U_{ex} scores assigned, a test obligation derived, and a forward reference to the corresponding UTHA scenario given.

Five of the six interfaces merit extended commentary to establish the analytical foundation for the UTHA scenarios of Section VIII.

PEI-001 (Perception to Scene). Perception performance under adverse conditions is the most extensively documented interface failure mode in the AV literature. The product assumption of camera-based object resolution at 100m under daylight conditions is operationally necessary but systematically violated by rain, fog, glare, and objects whose visual characteristics fall outside the training distribution [12]. The endogenous uncertainty at this interface is Moderate rather than Low because the perception algorithm's performance envelope is not fully characterized at the boundaries of its training distribution, leaving a gap between claimed and verified capability. This is the interface stressed by the 2018 Uber ATG incident, where perception cycling across classification categories was the exogenous contribution to the cross-axis cut set [10].

PEI-002 (Planning to Traffic). The planning system's assumption that other road users obey traffic rules is among the most fundamentally challenged assumptions in urban AV deployment. The assumption must hold for the planning algorithm's trajectory generation to be complete: a planner that does not account for ruleviolating road users cannot generate valid responses to their behavior. The U_{ex} High rating reflects not unusual edge cases but routine urban driving conditions [11]. The U_{en} Low rating reflects that the deficiency is in the product's operational design domain specification rather than in the quality of the planning algorithm itself: the algorithm correctly solves the problem it was given; the problem specification excludes the cases that matter most.

PEI-004 (V2X to Network). The V2X interface introduces a cybersecurity dimension that the perception and planning interfaces do not have. Network congestion and equipment failure are environmental (aleatoricexogenous) threats; spoofing and jamming are adversarial (epistemic-exogenous). Both classes are captured under the same interface point because they produce the same effect at the product boundary: the latency and integrity assumptions fail. ISO/SAE 21434 provides the threat analysis and risk assessment method for the adversarial class [17], but the PEI documents both classes together before domain decomposition assigns them to separate analytical frameworks.

PEI-005 (Driver to HMI). The 4-second takeover assumption is the most consequential human-reliability assumption in Level 3 AV design and the one with the weakest empirical support. SAE J3016 defines Level 3 as requiring the human driver to respond to a takeover request when issued; it does not specify a required response time, and field studies document a wide distribution of actual response times with a long right tail [86]. The U_{en} Moderate rating reflects that the human-machine interface design's ability to reliably trigger and monitor a driver response is itself uncertain, not just the driver's performance. Both sides of the interface carry non-trivial uncertainty, a pattern seen only at this interface and PEI001 in the six-row table.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

PEI-006 (Actuator to Road Surface). The road surface interface is classified U_{ex} Moderate rather than High because surface-induced braking degradation is well-characterized in the road safety literature and the range of degradation factors is bounded, unlike the unbounded scenario space of perception or planning. The management strategy is primarily ODD enforcement: restricting deployment to conditions where the dry-road physics model's predictions remain within acceptable margins, supplemented by runtime friction estimation where available.

D. Key Patterns from PEI Data

Three structural patterns emerge from Table XIII. Each pattern constitutes a testable claim about the nature of uncertainty in novel-environment socio-technical systems, and each is confirmed by the data in the table.

➢ *Pattern 1: Exogenous Uncertainty Dominates.*

In four of the six interface points (PEI-001, PEI-002, PEI-004, PEI-005), the U_{ex} rating is High while U_{en} is Low or Moderate. The environment presents conditions the product was not designed to handle, or conditions whose distribution is insufficiently characterized to bound them within the product assumption. This pattern directly confirms the framework's core thesis: in novel-environment systems, the primary source of uncertainty is the productenvironment interface, not the product itself. The product's internal quality, measured by U_{en}, is generally lower than the environmental uncertainty at the same interface. A program that allocates its entire uncertaintyreduction budget to product verification and process maturity activities without addressing the exogenous side will not reduce its dominant risk.

Table 13 PEI Analysis: Automotive Av System (Six Interface Points)

PEI ID	Interface	Product Assumption	Environment Reality	U_{en}	U_{ex}	Assumption-Reality Gap	Test Obligation	UTHA Ref
PEI-001	Perception ↔ Scene	Camera resolves objects >0.3 m at 100 m range in daylight conditions	Glare, rain, fog, low light, novel object classes, partial occlusion	Mod	**High**	Significant gap in adverse visual conditions; environment regularly violates daylight assumption	Low-light, misaligned, occlusion test suite across full ODD envelope	UTHA-001
PEI-002	Planning ↔ Traffic	Other vehicles and road users obey applicable traffic rules	Erratic drivers, jaywalkers, cyclists, rule-violating road users	Low	**High**	Fundamental assumption violated routinely in urban deployment environments	Adversarial traffic scenario suite including illegal manoeuvres and unpredictable pedestrians	UTHA-002
PEI-003	Localisation ↔ Map	HD map accurate to ± 0.1 m; current within operational zone	Construction zones, road surface changes, map staleness	Low	Mod	Temporal decay of map accuracy in dynamic areas; construction not reflected in map	Stale map and active construction zone scenarios; map-reality divergence tests	UTHA-003
PEI-004	V2X ↔ Network	Communication latency <100 ms; message integrity assured	Network congestion, jamming, spoofing, infrastructure failure	Low	**High**	Adversarial and environmental disruption both present; integrity not guaranteed	Congestion, spoofing, and jamming test scenarios; authentication failure cases	UTHA-004
PEI-005	Driver ↔ HMI	Driver responds to takeover request within 4 s	Distraction, overtrust, fatigue, automation complacency	Mod	**High**	Human reliability assumption unsupported by field data at Level 3; response times highly variable	Non-responsive and distracted-driver test scenarios; graduated distraction protocol	UTHA-005
PEI-006	Actuator ↔ Road Surface	Braking performance per dry-road physics model	Wet, icy, gravel, compacted snow, seasonal surface variation	Low	Mod	Surface variability affects stopping distance in ways the dry-road model does not bound	Adverse surface and ice scenario tests; stopping-distance verification across surface types	UTHA-006

SUE_Risk_Cube_Interactive.jsx PEI data (5 rows); XLSX: PEI Analysis sheet. U_{en} and U_{ex} ratings: Low ≈ $[0.0, 0.3]$; Mod(erate) ≈ $[0.3, 0.6]$; High ≈ $[0.6, 1.0]$ on the $[0, 1]$ scale. Rows PEI-007 through PEI-016 in the companion workbook contain additional interface points.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

The contrast with traditional manufactured products is instructive. In a conventional automotive safety context, the product (braking system, airbag inflator) is the dominant uncertainty source, whereas the environment (road surface, crash dynamics) is well characterized by decades of field data and standardized test protocols. The four High-U_{ex} interfaces in Table XIII are interfaces where no equivalent field data exists, and no standardized test protocol captures the full operational distribution. The PEI makes this contrast explicit and quantified.

- *Pattern 2: Two Interfaces Carry Simultaneous Product and Environment Uncertainty.*

PEI-001 (Perception to Scene) and PEI-005 (Driver to HMI) each carry U_{en} Moderate alongside U_{ex} High. These are the two interfaces where the product's own performance at the boundary is itself uncertain, not just the environment it faces. At PEI-001, the perception algorithm's characterized performance envelope does not extend to the full distribution of adverse visual conditions, so both the product's capability claim and the environment's actual distribution are uncertain simultaneously. At PEI005, both the reliability of the HMI's takeover alert and the driver's response are uncertain. These dualuncertainty interfaces exhibit the highest β interaction coefficients in the propagation model: the mutual noncompensation condition is most severe when neither side of the interface reliably performs its role.

- *Pattern 3: Two Interfaces are Moderate-Exogenous Rather than High-Exogenous.*

PEI-003 (Localisation to Map) and PEI-006 (Actuator to Road Surface) carry U_{ex} Moderate. In both cases, the environmental variability is real and relevant, but its distribution is better characterized than for the High interfaces, and the gap between the product assumption and the known environmental behavior is smaller. For PEI-003, HD map accuracy and the rate of environmental change are measurable and bounded; for PEI-006, road surface physics is well-studied, and the range of friction coefficients is documented. These interfaces still require test obligations and UTHA analysis, but their SUE Levels in Section VIII are SUE-2 rather than SUE-1, reflecting the lower combined weight of the Moderate U_{ex} rating.

The three patterns together constitute an empirical argument for the SEI/PEI as a necessary precursor to risk assessment. A HARA or SOTIF scenario analysis conducted without a prior PEI would need to independently discover each of the six interface-level gaps documented in Table XIII. In practice, domain-specific risk assessments often assume the product's operating conditions and test against those assumed conditions rather than against the actual environmental distribution. The PEI forces this assumption to the surface before any domain analysis begins, creating the condition for risk assessment to be grounded in the actual product environment relationship rather than in the product team's prior beliefs about what the environment will present.

The companion workbook contains PEI rows PEI007 through PEI-016, providing capacity for ten additional interface points beyond those shown in Table XIII. Programs with more complex AV architectures, additional sensor modalities, extended ODD definitions, or regulatory interface requirements can populate these rows using the five-field format of Section VII-B. Each populated row becomes a new row in the UTHA analysis and, through the UTHA-to-UVG traceability chain, a contribution to the combined domain tree and the SUE Risk Cube.

VIII. PLA/UTHA: UNIFIED THREAT AND HAZARD ANALYSIS

The SEI/PEI analysis of Section VII produces a characterized set of interface points, each with measured uncertainty and a documented assumption-reality gap. The Parametric Loss Analysis (PLA) operates on these outputs to generate the loss scenarios that populate the SUE Risk Cube. In the automotive instantiation, PLA is called the Unified Threat and Hazard Analysis (UTHA). This section defines PLA, presents the sixscenario UTHA risk assessment for the interface points in Table XIII, and identifies patterns in the resulting SUE Level assignments.

A. What PLA is

The Parametric Loss Analysis is the risk-assessment step in the SUE framework. It is executed after the SEI/PEI has characterised the product-environment interface and before the Unified Loss Goal is articulated. PLA sits at L3 and L4 of the Uncertainty Diamond: the parametric analysis layer and the β interaction layer, converging at the focal point where the Risk Cube is populated.

PLA's methodology borrows the guide word technique from Hazard and Operability (HAZOP) studies, which apply structured deviations to process parameters to systematically enumerate failure conditions [87]. In PLA, the parameters are the interface variables identified by the SEI: product assumptions, environmental conditions, and measured uncertainty values at each interface point. The guide words are deviations from normal interface behaviour: *too high*, *too low*, *none*, *reversed*, *other than*, *before*, *after*. Applying each guide word to each interface parameter generates a candidate loss scenario. The candidate is retained if it represents a credible hazardous state; it is dismissed otherwise, with a documented rationale.

PLA's decisive structural advantage over the singledomain risk assessment methods it replaces is that it operates on all interface parameters simultaneously, before domain decomposition assigns each parameter to a specific standard. The parameters identified at the Perception-to-Scene interface (PEI-001) are relevant to all three domains: functional safety (the AEB function), SOTIF performance (the misclassification scenario), and cybersecurity (the adversarial spoofing case). A HARA conducted under ISO 26262 alone would analyze the AEB functional failure but would not capture the SOTIF misclassification; a SOTIF scenario analysis alone would capture misclassification but would not capture the spoofing case [12], [13], [17]. The PLA generates all three from the same interface parameter in a single analytical pass, preserving the cross-domain information that

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

domain-specific methods discard at the point of decomposition.

In the automotive instantiation, PLA is called the Unified Threat and Hazard Analysis (UTHA). UTHA subsumes three domain-specific methods: the Hazard Analysis and Risk Assessment (HARA) of ISO 26262, the Threat Analysis and Risk Assessment (TARA) of ISO/SAE 21434, and the scenario-based performance insufficiency analysis of ISO 21448 (SOTIF). Each of these methods generates its own risk assessment artifacts; UTHA does not replace those artifacts but supersedes them at the level above, ensuring that the scenario set feeding each domain's analysis was generated from the same cross-domain interface characterization.

The output of each UTHA scenario is a triplet: the System Layer, Safety Domain, and Uncertainty Type, which locates the scenario in the SUE Risk Cube. The combined weight of those three coordinates determines the scenario's SUE Level. The SUE Level is the primary output of the UTHA step and the primary input to the Risk Cube population. Section IX traces each UTHA scenario forward to its Unified Vehicle Goal; the present section documents the scenario identification and SUE Level assignment.

B. UTHA Risk Assessment: Six Scenarios

Table 14 presents the six UTHA scenarios derived from the six PEI interface points of Table XIII. Each scenario is generated by applying PLA guide words to the interface parameter at its source PEI, identifying the dominant system layer and domain, assigning the uncertainty type, and computing the combined weight that determines the SUE Level.

Three observations from Table 14 warrant explanation before the summary statistics.

UTHA-002 and UTHA-005 reach the maximum combined weight of 12. Both scenarios achieve $w_{\text{Layer}} = 4$ (Design and Requirements respectively), $w_{\text{Domain}} = 4$ (Safety), and $w_{\text{UType}} = 4$ (Epistemic-Exogenous). These are the two scenarios in which the source interface makes a fundamental assumption about the human or social environment (other road users obeying rules; driver responding within 4 seconds) that is simultaneously unsupported and irreducible through product engineering alone. Neither scenario can be resolved by improving the product's internal quality; both require either ODD restriction, environmental management, or architectural changes to the product's dependence on the uncontrollable environmental variable.

UTHA-004 spans functional safety and cybersecurity domains. The V2X spoofing scenario is generated at the Security domain, not the Safety domain, yet it produces a SUE-1 rating because the Design-layer weight (4) and the Epistemic-Exogenous type weight (4) together compensate for the Security domain's lower weight (3) relative to Safety. This illustrates the tensor's cross-domain resolving power: a cybersecurity scenario that affects the vehicle's physical control system is as risk-significant as a functional safety scenario, a property that siloed analysis obscures by treating security and safety risks on separate scales.

The Aleatoric-Exogenous type distinguishes the two SUE-2 scenarios from the four SUE-1 scenarios. UTHA-003 and UTHA-006 both involve inherent environmental variability (map staleness as a form of temporal aleatoric uncertainty; road surface friction as physical aleatoric uncertainty) rather than unknown-unknown scenario gaps. This classification correctly reduces their severity: the variability is real and consequential, but it is bounded, characterised, and manageable through ODD constraints and runtime monitoring. The SUE-2 designation does not mean these scenarios are unimportant; it means the intervention pathway is different from the SUE-1 scenarios, where the dominant uncertainty is epistemic and addressable through expanded validation and ODD characterisation.

C. UTHA Summary

The six-scenario UTHA produces the following distribution of SUE Levels:

- SUE-1 (Critical): UTHA-001, UTHA-002, UTHA004, UTHA-005 (4 of 6 scenarios)
- SUE-2 (Elevated): UTHA-003, UTHA-006 (2 of 6 scenarios)
- SUE-3 or SUE-4: None at this interface population

The 4/6 SUE-1 distribution is consistent with the axisweight structure of the SUE Risk Cube. Four of the six interface points carry High U_{ex}; when those interfaces generate loss scenarios, the Epistemic-Exogenous uncertainty type (weight 4) combines with Design- or Requirements-layer positions (weight 4) and Safety or Performance domains (weight 4) to produce combined weights of 11 or 12. The only way a PEI-derived UTHA scenario avoids SUE-1 is if either the uncertainty type is Aleatoric (reducing the type weight) or the system layer is Implementation or below (reducing the layer weight). UTHA-003 and UTHA-006 satisfy these conditions; the other four do not.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

Table 14 Utha Risk Assessment: Automotive AV (Six Scenarios)

UTHA ID	Source PEI	Loss Scenario	System Layer	Safety Domain	U-Type	UVG Ref	SUE Level	Rationale (Combined Weight)
UTHA-001	PEI-001	Perception misclassification in adverse lighting causes AV failure to detect VRU or obstacle	Implementation (wt: 3)	Performance (wt: 4)	Ep-Exogenous (wt: 4)	UVG-001	**SUE-1** (11)	High-weight layer × high domain × highest U-type; combined weight 11
UTHA-002	PEI-002	Planning collision with rule-violating road user not modelled in trajectory generation	Design (wt: 4)	Safety (wt: 4)	Ep-Exogenous (wt: 4)	UVG-001	**SUE-1** (12)	Maximum combined weight; Design × Safety × Ep-Exo = 4 + 4 + 4 = 12
UTHA-003	PEI-003	Localization error from stale map induces unintended lane departure	Implementation (wt: 3)	Performance (wt: 4)	Al-Exogenous (wt: 2)	UVG-002	SUE-2 (9)	Moderate layer weight reduces total below SUE-1 threshold; 3 + 4 + 2 = 9
UTHA-004	PEI-004	Spoofed V2X message causes AV to execute unsafe manoeuvre from corrupted command	Design (wt: 4)	Security (wt: 3)	Ep-Exogenous (wt: 4)	UVG-003	**SUE-1** (11)	Adversarial cross-domain scenario; Design × Security × Ep-Exo = 4 + 3 + 4 = 11
UTHA-005	PEI-005	Driver fails to respond to takeover request; AV exceeds operational boundary	Requirements (wt: 4)	Safety (wt: 4)	Ep-Exogenous (wt: 4)	UVG-001	**SUE-1** (12)	Maximum combined weight; Requirements × Safety × Ep-Exo = 4 + 4 + 4 = 12
UTHA-006	PEI-006	Braking performance degraded on adverse surface; stopping distance exceeds available clearance	Design (wt: 4)	Safety (wt: 4)	Al-Exogenous (wt: 2)	UVG-004	SUE-2 (10)	Aleatoric type reduces total; Design × Safety × Al-Exo = 4+4+2 = 10

Combined weight = $w_{Layer} + w_{Domain} + w_{UType}$. SUE Level boundaries per Table VIII: weight 11–12 = SUE-1; weight 9–10 = SUE-2. UVG-004 is currently empty in the companion workbook (see Section VIII-C).

An important open discrepancy requires flagging here. UTHA-006 references UVG-004 as its parent Unified Vehicle Goal. At the time of this analysis, UVG-004 is empty in the companion workbook: no UVG statement has been recorded for the braking-on-adverse-surface scenario. This means the traceability chain from UTHA006 to the combined domain tree is broken, and the safety case for this scenario is incomplete. The scenario's SUE-2 designation and its test obligation (adverse surface and ice scenario tests, from Table 13 PEI-006) are established; what is missing is the authoritative loss prevention statement that parents the domain-specific goals for this scenario. Section IX documents UVG-001 through UVG-003 as active goals and flags UVG-004 as an open item requiring resolution before the full UTHAto-UVG traceability chain can be closed.

The UTHA output flows in two directions. Upward, each UTHA scenario contributes to the SUE Risk Cube tensor: the triplet (System Layer, Safety Domain, Uncertainty Type) places the scenario at a specific coordinate, and the SUE Level populates that cell. Downward, each UTHA scenario generates a UVG or feeds an existing

UVG from a previous scenario (UTHA-001, -002, and -005 all feed UVG-001, reflecting that three different interface failures can each independently violate the same loss prevention goal). Section IX develops the UVG register, the SMO architecture, and the combined domain tree that give each UTHA scenario its full downstream traceability.

IX. ULG/UVG, SMO, AND COMBINED DOMAIN TREES

The UTHA scenarios of Section VIII each identify a loss event, assign it a SUE Level, and require a forward reference to a parent loss prevention statement. That statement is the Unified Loss Goal (ULG): a construct that exists above the domain-specific goal hierarchy and is stated in terms of what must not happen, rather than what caused it. This section develops the ULG, its automotive instantiation as the Unified Vehicle Goal (UVG), the Safe Maneuver Objective (SMO) that each UVG specifies, and the combined domain tree under which domain-specific analysis is subordinated.

A. Unified Loss Goal and Unified Vehicle Goal

Each PLA/UTHA cell produces a Unified Loss Goal: a domain-agnostic loss prevention statement that parents all domain-specific goals derived from that loss scenario.

The ULG is stated in terms of the loss to be prevented, not the mechanism that would cause it. A ULG does not say "the AEB shall not fail due to a sensor fault"; it says "the

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

vehicle shall not strike a vulnerable road user." This level of abstraction is deliberate: it allows the same goal to be the parent of a safety goal (preventing AEB failure due to hardware fault), a cybersecurity goal (preventing AEB suppression due to spoofing), and a SOTIF criterion (preventing AEB non-activation due to perception insufficiency) without requiring any of the three domain analyses to acknowledge the others' existence. Each domain analysis remains structurally unchanged; the ULG adds a traceability layer above without modifying anything below [11].

In the automotive AV context, the ULG instantiates as the Unified Vehicle Goal (UVG). Each UVG decomposes into three classes of domain-specific child goals: a Safety Goal with an Automotive Safety Integrity Level (ASIL), governed by ISO 26262; a Cybersecurity Goal with a Cybersecurity Assurance Level (CAL), governed by ISO/SAE 21434; and a SOTIF Criterion (SC) specifying the validation target for the functional insufficiency class, governed by ISO 21448 [12], [13], [17]. Each child goal then flows downward through its domain's own requirements decomposition: the safety goal into functional safety requirements and technical safety requirements; the cybersecurity goal into cybersecurity requirements and security controls; the SOTIF criterion into validation targets and ODD restrictions.

Table 15 presents the UVG register for the automotive AV case, derived from the UTHA scenarios of Table 14.

Two properties of Table XV warrant emphasis. First, three UTHA scenarios (UTHA-001, UTHA-002, and UTHA-005, derived from PEI-001, PEI-002, and PEI005 respectively) all feed UVG-001. This convergence reflects the architecture of the combined domain tree: multiple distinct failure pathways across multiple domains and lifecycle layers can independently violate the same loss prevention statement. The ULG construct makes this convergence explicit; without it, the three scenarios would be documented in three separate domain analyses, with no shared parent and no mechanism to identify that each is a sufficient condition for the same loss outcome.

Second, UVG-004 is currently empty. The UTHA-006 scenario (braking degradation on adverse surface, SUE-2) has been identified and assigned a SUE Level, but no authoritative UVG statement has been recorded to parent it. This means the combined domain tree for this scenario cannot be constructed and the child goals for the adverse surface braking class cannot be traced. Programs using this framework should treat UVG-004 as an open action item: the UVG statement must be written, an SMO specified, and domain-specific child goals derived before the safety case for this interface can be considered complete.

Table 15 UVG Register: Automotive AV (Three Active Goals; UVG-004 Open)

UVG ID	UVG Statement	Source PEI	Safety Goal	Cyber Goal	SOTIF Criterion	SMO	SUE Level	Cross-Domain Cut Set
UVG-001	Vehicle shall not strike a VRU due to perception, planning, or supervision failure	PEI-001, PEI-002, PEI-005	SG-001 (ASIL D)	CG-001 (CAL 2)	SC-001: Classify VRU correctly across full ODD envelope	Controlled deceleration to safe refuge; perception-degraded speed constraint	SUE-1	{Perception cycling} AND {AEB suppression logic} AND {driver inattention}
UVG-002	Vehicle shall not depart lane due to localisation or mapping failure	PEI-003	SG-002 (ASIL C)	n/a	SC-002: Maintain lane position within tolerance across map-validity conditions	Reduce speed; widen lane-keeping tolerance; fallback to camera-only localisation	SUE-2	{Map staleness} AND {localisation drift}
UVG-003	Vehicle shall not execute an unsafe manoeuvre from compromised communications	PEI-004	SG-003 (ASIL B)	CG-002 (CAL 3)	n/a	Fallback to local-only planning; V2X message quarantine	SUE-1	{Spoofed V2X command} AND {planning system trust assumption}
UVG-004	*Open: no UVG statement recorded*	PEI-006	(open)	(open)	(open)	OPEN	(open; links to UTHA-006)	

ASIL levels per ISO 26262 decomposition from UTHA. CAL levels per ISO/SAE 21434 TARA. UVG-005 and UVG-006 are planned goals under development; UVG-004 requires a UVG statement before the UTHA-006 traceability chain is closed. Note: UTHA-001, UTHA-002, and UTHA-005 each feed UVG-001 independently, reflecting three distinct failure pathways to the same loss event.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

B. Safe Maneuver Objective

Each UVG specifies a Safe Maneuver Objective: the operational target state the vehicle must reach when a UVG violation is detected or is imminent. The SMO is the unified safe state for the vehicle as a whole, transcending the domain-specific safe state concepts of individual standards.

The SMO has three defining properties. First, it is *domain-agnostic*: the SMO specifies what physical state the vehicle must achieve, not which domain's intervention achieves it. The SMO for UVG-001 is "controlled deceleration to safe refuge with a perception-degraded speed constraint." This target is the same whether the UVG violation trigger is a functional safety fault, a cybersecurity event, or a SOTIF perception insufficiency. Domain responses may differ in their internal mechanics, but they all serve the same terminal objective.

Second, the SMO is *continuously evaluated*: the vehicle maintains a precomputed contextual SMO based on its current operating conditions, which is updated in real time. A vehicle on a motorway with clear sight lines and adequate clearance has a different contextually available SMO than the same vehicle in dense urban traffic at night. Pre-computing the SMO ensures that when a violation trigger arrives, the system does not need to plan a safe state under the degraded conditions that the trigger has created.

Third, and most importantly for the framework's analytical value, the SMO is *conflict-resolving*. This property addresses a well-documented challenge in automotive safety engineering: the Minimal Risk Condition (MRC) specified by ISO 26262 for functional safety failures (for example, bring the vehicle to a controlled stop at the roadside) can directly conflict with the SOTIF-appropriate response to a perception insufficiency (for example, maintain speed and trajectory to avoid creating a rear-collision hazard). Without a unified safe state arbitrating between these domain-specific responses, a system that simultaneously enters a safetytriggered MRC and an SOTIF-triggered minimum-risk maneuver has no coherent behaviour [12], [13]. The SMO provides the arbitration: it specifies the physical target state that both domain responses must serve, and domain-specific responses provide constraints on how that target is reached rather than independently defining the target itself [11].

The SMOs in Table XV illustrate this. UVG-001's SMO (controlled deceleration to safe refuge) can be achieved through a functional safety MRC, a SOTIFtriggered speed reduction when perception confidence falls below the threshold, or a cybersecurity-triggered isolation of V2X commands. All three triggers lead to the same physical terminal state; the domain differences lie in the trigger conditions and the path constraints, not in the destination.

C. Combined Domain Tree

Each UVG is decomposed using a combined domain tree: a single fault-and-threat tree with a single OR gate, where each branch uses the domain-native analysis method for its domain.

The OR gate expresses the logical structure of the UVG: the loss event occurs if any branch is realized. Each branch uses the domain-native method so that domain analysts work in the language of their standard: safety analysts conduct FTA as specified by ISO 26262, cybersecurity analysts build attack trees as specified by ISO/SAE 21434, performance analysts build insufficiency trees as required by ISO 21448. The combined tree does not require domain analysts to learn each other's methods; it requires only that the outputs of all four methods be brought under the same OR gate at the UVG level [12], [13], [17]. The tree structure is:

```
UVG (loss prevention statement)
    |
    OR gate (UVG violation)
    |
    +-- Fault Tree branch (Safety domain;
        FTA per ISO 26262)
    +-- Attack Tree branch (Security
        domain; TARA per ISO/SAE 21434)
    +-- Insufficiency Tree branch (
        Performance domain; SOTIF/AI)
    +-- Organisational branch (
        Organisational domain; HF/ASPICE)
```

The unique analytical output of the combined tree is the *cross-domain minimal cut set*: a combination of leaf nodes from two or more branches that together satisfy the OR gate, where no single branch's leaf nodes alone are sufficient. Leveson demonstrates that the most dangerous failure modes in complex sociotechnical systems arise from exactly this class of multibranch interactions, where no single domain's analysis can identify the combination because each domain owns only its own branch [1]. The combined tree makes these cross-domain combinations structurally visible.

Three categories of cut sets emerge from the combined tree analysis:

Single-domain cut sets are conventional within-branch fault or attack trees where a sequence of events within one domain independently satisfies the OR gate. A hardware failure causing AEB non-activation is a singledomain (Safety) cut set.

Cross-domain cut sets span two or more branches. The example for UVG-001 illustrates the structure: {perception cycling across object classification categories, exogenous SOTIF branch} AND {AEB action suppression logic, endogenous Safety branch} AND {single-driver supervision policy, endogenous Organizational branch}. No single branch generates this cut set: the SOTIF insufficiency tree sees only the perception cycling; the FTA sees only the suppression logic; the organizational analysis sees only the supervision policy. The combined tree under UVG-001 sees all three simultaneously, and their AND conjunction is a cross-domain

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

minimal cut set. This is the Uber ATG Tempe fatality cut set, documented retrospectively in Section XII [10].

Cross-axis cut sets are a subclass of cross-domain cut sets where the spanning combination includes at least one endogenous branch node and at least one exogenous branch node. These are the highest-β interactions in the propagation model of Section IV: a product weakness (endogenous) meeting an environmental challenge (exogenous) at the same OR gate. The Uber ATG cut set above is cross-axis: the suppression logic and supervision policy are endogenous (product and organizational design decisions); the perception cycling is exogenous (SOTIF, driven by the environment's visual characteristics). Cross-axis cut sets are the structural manifestation of the β interaction coefficients at the analytical level of the combined tree.

Each cross-domain or cross-axis cut set identified at L5 in the Uncertainty Diamond generates a test case that no single domain would produce independently: a test that simultaneously instantiates the endogenous condition (the suppression logic is active) and the exogenous condition (the perception system encounters a challenging object class under adverse lighting), and the organizational condition (the driver monitoring is absent). These test cases serve as the primary validation evidence for cross-domain cut sets and as the primary input to the field data feedback loop of Section VI-D.

X. AUTOMOTIVE INSTANTIATION

Sections VII through IX developed the SUE methodology for the automotive AV domain, using the domainspecific names PEI, UTHA, and UVG throughout. This section makes explicit the systematic correspondence between the framework's generic constructs and their automotive counterparts, traces the full seven-step chain from product-environment boundary to operational deployment, and establishes how the framework relates to and complements the existing standards that govern automotive safety engineering.

A. Generic-to-Automotive Mapping

Each construct in the generic SUE framework maps to a specific automotive artifact, method, or standard. Table XVI presents the full mapping. The fourth column identifies the primary source of each mapping: where an existing standard already defines the relevant artifact, the framework instantiates as that standard's artifact; where no existing standard covers the construct, the automotive name is new and fills a gap across all existing standards. The PEI is the only entry that has no existing standard to which it can be instantiated. All other rows map to an existing standard or set of standards that already define the corresponding activity for their domain. The PEI fills a structural gap: the product-environment boundary analysis that makes the product assumption explicit and measures the assumption-reality gap that exists in no existing automotive standard. ISO 26262 defines the HSI for the HW/SW boundary but not for the product-environment boundary. ISO 21448 acknowledges the existence of performance limitations at the product-environment boundary but provides no systematic method for enumerating and measuring the gaps at individual interface points before scenario analysis begins. ISO/SAE 21434 defines the system boundary for cybersecurity purposes but does so in terms of threat entry points, not in terms of the full landscape of assumptions the product makes about the environment it operates in [12], [13], [17]. The PEI addresses this gap by treating the product-environment boundary as a first-class engineering artifact, analogous to the HSI, with its own structured documentation format and its own test obligation output.

The domain-to-standard mapping in Table XVI reflects an important design principle: each domain's weight in the SUE Risk Cube is calibrated to the hazard and loss potential of that domain in the automotive AV context, not to the maturity or coverage of the standard governing it. The Safety and Performance domains carry equal weight 4 because both can produce direct physical harm through independent pathways, even though the ISO 26262 standard is considerably more mature than ISO 21448. The Security domain carries weight 3, not because cybersecurity is less important, but because a successful cyber attack requires an additional step beyond the security event itself to cause physical harm. The Organizational domain carries a weight of 2 because process maturity failures amplify other domains' risks rather than generating independent harm outcomes.

B. Full Traceability Chain

Figure 6 presents the complete traceability chain from the product-environment boundary through to operational deployment and back. The chain has six forward steps and one feedback return.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

Table 16 Generic Framework to Automotive Instantiation

Framework (Generic)		Automotive Instantiation	Description	Standard / Note
SEI (System-Environment Interface analysis)		PEI (Product-Environment Interface)	Systematic mapping of product assumptions against environmental reality; generates U_{en}/U_{ex} gap measurements and test obligations	New; fills gap in all existing standards
PLA (Parametric Analysis)	Loss	UTHA (Unified Threat and Hazard Analysis)	Risk assessment step; generates loss scenarios across all domains simultaneously; assigns SUE Levels	Subsumes HARA (ISO 26262), TARA (ISO/SAE 21434), SOTIF analysis (ISO 21448)
ULG (Unified Goal)	Loss	UVG (Unified Vehicle Goal)	Domain-agnostic loss prevention statement; parents Safety Goal, Cybersecurity Goal, and SOTIF Criterion; specifies SMO	Parents SG, CG, SOTIF Criterion
Safety domain		Functional Safety	Endogenous-epistemic and endogenous-aleatoric hazard management; hardware and software fault coverage	ISO 26262
Performance domain		Safety of the Intended Functionality (SOTIF)	Exogenous-epistemic insufficiency management; scenario coverage; ODD validation	ISO 21448
Security domain		Cybersecurity Engineering	Adversarial exogenous threat management; TARA; security controls; CAL assignment	ISO/SAE 21434
Organisational domain		Process Maturity	Process-layer uncertainty management; capability assessment; supplier quality	Automotive SPICE

The six-step chain in Figure 6 provides a property that none of the individual standards currently achieves: complete bidirectional traceability from every derived domain requirement back to the specific productenvironment interface assumption from which it originated. A functional safety requirement that traces to a Safety Goal, which traces to a UVG, which traces to a UTHA scenario, which traces to a PEI interface point, gives the safety assessor a direct answer to the question "why does this requirement exist?" The answer is "because the product makes this assumption about the environment at this interface, the environment's actual behavior departs from that assumption in these conditions, and the UTHA analysis determined that the departure produces a Safety-domain loss scenario at this SUE Level."

This bidirectional traceability serves several practical functions beyond the safety case. The impact analysis for a design change becomes tractable: changing the product assumption at a PEI interface point immediately identifies all UTHA scenarios, UVGs, combined tree branches, child goals, and requirements that derive from that assumption, bounding the change's downstream scope. The validation coverage metric becomes interface-grounded: test completeness is measured against the test obligations at each PEI interface point rather than solely against domain-specific requirements lists, ensuring that the assumption-reality gap at each interface is closed before the system is released. The feedback loop at Step 6 closes the chain by returning operational evidence to the PEI characterization, updating U_{ex} estimates and triggering re-analysis where new data reveals

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

that the deployment environment departs from the characterized ODD.

C. Integration with Existing Standards

The SUE framework is complementary to, not a replacement for, the existing automotive safety and quality standards. This distinction is structural and deliberate.

The existing standards define *what activities to perform*: ISO 26262 specifies the HARA, safety goal derivation, functional and technical safety requirements, hardware architectural metrics, and the verification and validation activities required for each ASIL level. ISO 21448 specifies the scenario analysis, the ODD definition, and the performance validation activities required for SOTIF coverage. ISO/SAE 21434 specifies the TARA, cybersecurity requirements, and security controls. Automotive SPICE specifies the process capability requirements across all engineering processes [12], [13], [17], [44]. Each standard performs its defined role well within its domain.

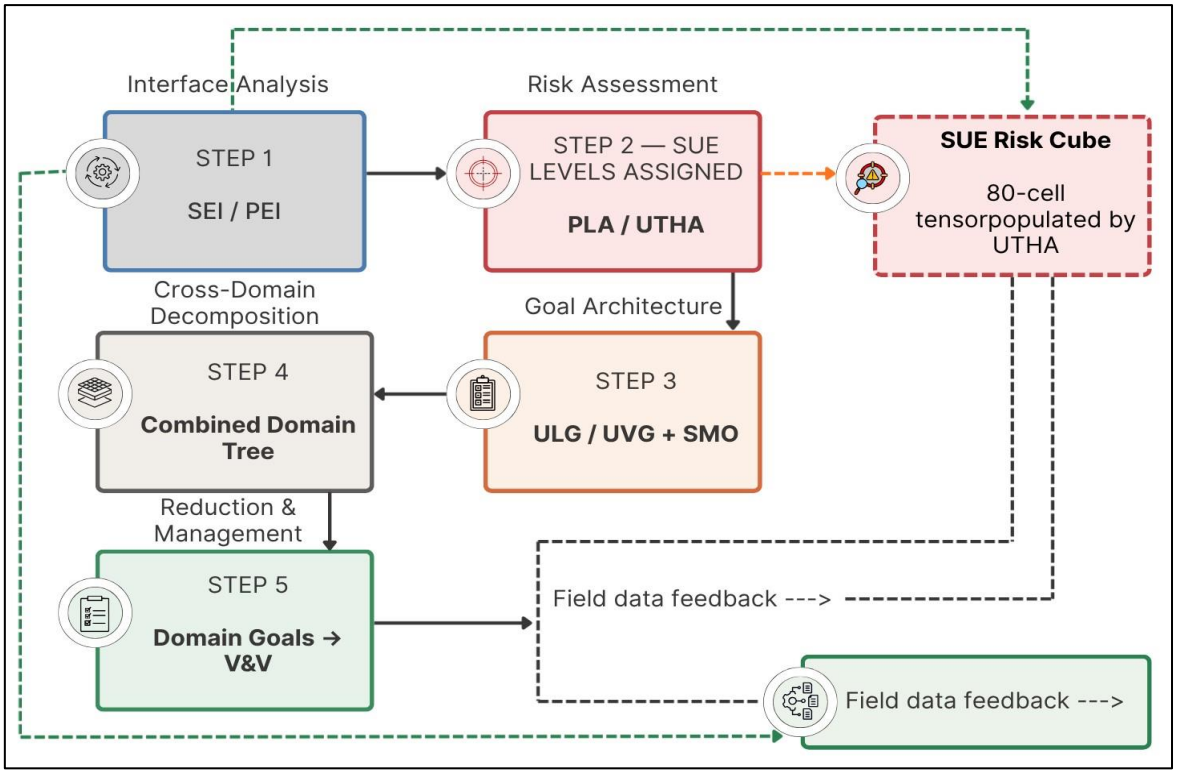

Fig 6 SUE Framework: Full Traceability Chain (Steps 1–6)

- *Step 1: SEI/PEI (interface analysis: maps boundary, measures U_{en}/U_{ex} gap, generates test obligations).*
- *Step 2: PLA/UTHA (risk assessment: applies guide words, generates loss scenarios, assigns SUE Levels, populates Risk Cube).*
- *Step 3: ULG/UVG (loss prevention statement: articulates unified goal, specifies SMO, defines joint gate verdict).*
- *Step 4: Combined Domain Tree (FTA + attack tree + insufficiency tree under single OR gate; cross-domain cut sets). Step 5: Domain-specific child goals (SG/ASIL, CG/CAL, SOTIF Criterion; flows into domain requirements).*
- *Step 6: Implementation, V&V, operational deployment (field data feeds back to Step 1 via right-arm L8 feedback loop).*

The SUE framework addresses what the standards do not: it quantifies *how well those activities have reduced the system's uncertainty*. A program that has completed all required ISO 26262 activities may still carry high $U^{en,ep}$ at its requirements layer if those activities were executed with low process maturity or produced requirement artifacts with high ambiguity density. A program that has completed all required ISO 21448 scenario coverage activities may still carry a high $U^{ex,ep}$ if its ODD characterization is narrower than the deployment environment. The standards confirm compliance; the SUE Risk Cube quantifies residual uncertainty. The relationship is precisely analogous to the relationship between a manufacturing process standard (ISO 9001) and Six Sigma: the process standard defines what to do; the quantitative framework measures how well it is working.

Each standard's artifacts are preserved intact within the framework's structure. Auditors reviewing ISO 26262 compliance see the same safety goals, ASIL assignments, and hardware architectural metrics they expect; those artifacts are simply child goals under the UVG rather than top-level goals, which adds traceability above without modifying the artifacts themselves. Auditors reviewing ISO 21448 coverage see the same scenario partitioning, ODD definition, and performance validation evidence they expect; those are now the right-arm activities of the Uncertainty Diamond at L6 through L8, positioned in the process model but unchanged in their content. Auditors reviewing ASPICE capability levels see the same process ratings they expect; those ratings now feed

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

directly into the process-layer $U(v_i)$ values of the SUE propagation model, quantifying the relationship between process capability and residual uncertainty in a way that ASPICE alone cannot.

This non-disruptive integration is the framework's advantage in practical adoption. An organization adopting the SUE framework does not need to discard its existing safety processes, retrain its domain experts, or restructure its audit artifacts. It adds: the PEI as the pre-decomposition interface analysis that domain analyses have been missing; the UTHA as the crossdomain scenario synthesis layer that unifies what HARA, TARA, and SOTIF analysis currently do in parallel; the UVG as the above-domain traceability anchor that makes the connections between domain artifacts explicit; and the SUE Risk Cube as the quantitative instrument that measures residual uncertainty and supports investment prioritization. Each addition fills a structural gap without removing anything that existing standards require.

The complementarity is summarized by a single organizing principle: *standards define activities; the Uncertainty Diamond quantifies how well those activities reduce uncertainty*. This principle places the two in a well-defined relationship in which the standards provide the activity definitions, and the framework provides the measurement theory. A program that operates both achieves what neither can achieve alone: compliance with domain-specific activity requirements and a quantitative measure of the residual systemic uncertainty left behind by those activities.

XI. ECONOMIC INTERPRETATION

The SUE Risk Cube provides a quantitative tensor of residual uncertainty across 80 cells. To translate this engineering metric into the resource-allocation language of program management, two additional constructs are required: a financial translation that converts Systemic Risk Density into an expected monetary loss, and a return metric that ranks competing interventions by their costeffectiveness. This section develops both, then applies them to the six planned interventions for the automotive AV program analyzed in Sections VII through X.

A. Expected System Loss

The Expected System Loss (ESL) translates the domain-specific SRD projections of Section V-H into financial terms. The domain differentiation is essential because different uncertainty domains carry structurally different cost profiles in automotive applications: a functional safety failure producing a fatality or serious injury carries direct tort liability, recall cost, and regulatory penalty; a SOTIF performance failure producing a loss of function or near-miss carries a distinct cost structure dominated by reputational exposure and regulatory scrutiny; a cybersecurity breach carries investigation, remediation, and potential fine costs; an organizational maturity failure carries program delay and rework costs. Aggregating all four into a single scalar before translation loses the domain-specific cost information that is needed to determine where, within the risk landscape, intervention is most valuable. The ESL is therefore defined at the domain level and summed:

$$\mathrm{ESL} = \sum_{d \in \mathrm{domains}} \mathrm{SRD}_d \times C_{e,d} \times E_d \tag{9}$$

Where SRD_d is the Systemic Risk Density for domain d, derived as the face-sum projection of the Risk Cube over the Layer and Uncertainty Type axes at fixed domain d (Section V-H); $C_{e,d}$ is the expected cost per realized loss event in domain d, encompassing direct costs, recall costs, litigation, regulatory penalties, and reputational impact; and E_d is the operational exposure in domain d, expressed as fleet size times operational hours for automotive deployment, or an equivalent exposure metric.

The product $\mathrm{SRD}_d \times C_{e,d}$ is the loss density per unit exposure for domain d: the financial consequence of the current uncertainty state, per unit of deployment time. The product $\mathrm{SRD}_d \times C_{e,d} \times E_d$ is the total expected loss attributable to domain d over the planned deployment exposure. Summing across domains gives the program's total expected system loss under the current uncertainty state, expressed in monetary units.

Reniers and Sörensen formally demonstrate that optimal prevention investment decisions require a structured cost-benefit framework in which avoided accident costs are explicitly calculated as the difference between expected losses before and after each intervention [31]. The ESL formulation provides exactly this structure for the SUE framework: the ESL before an intervention is the baseline from which $\Delta\mathrm{ESL}$ is measured, and $\Delta\mathrm{ESL}$ is the financial benefit against which the intervention cost I is compared. Calibrating the $C_{e,d}$ values requires domain-specific actuarial data; for automotive AV applications, the US Department of Transportation's value of a statistical life provides one anchor for the Safety domain cost, and regulatory fine schedules provide anchors for the Security domain [88].

B. Return on Safety Investment

The Return on Safety Investment (ROSI) expresses the financial benefit of an intervention relative to its cost:

$$\mathrm{ROSI} = \frac{\Delta\mathrm{ESL} - I}{I} \tag{10}$$

Where $\Delta\mathrm{ESL}$ is the reduction in expected system loss produced by the intervention, and I is the intervention cost. An intervention with $\mathrm{ROSI} > 0$ produces more financial benefit than it costs; an intervention with

$\mathrm{ROSI} < 0$ costs more than the loss it avoids and should not be undertaken unless non-financial obligations (regulatory compliance, contractual commitments) require it. Ranking interventions by ROSI provides the optimal allocation sequence: within a fixed budget, apply interventions in descending ROSI order until the budget is exhausted.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

Collier and colleagues confirm in the context of security investment that ratio-based metrics of the ROSI form are the most reliable guide to investment prioritization when the metric is grounded in an expected-benefitof-investment calculation rather than in ordinal risk scores [30]. The SUE ROSI satisfies this requirement: ΔESL is derived from the tensor-level SRD change produced by the intervention, translated through the ESL formula, and therefore grounded in the same quantitative measurement framework as the risk assessment itself rather than in a separate ordinal scoring exercise.

This framing converts the organizational conversation about safety investment from a compliance question to a resource-allocation question. Compliance asks: Have all required activities been completed? Resource allocation asks: which uncertainty reductions produce the most value per unit cost? The ROSI provides a quantitative answer to the second question. The transformation is structurally parallel to what Six Sigma achieved for quality investment in manufacturing: DPMO and process capability metrics converted quality improvement from an activity requirement into an investment with measurable financial returns, enabling objective comparison of competing improvement initiatives [14].

C. Intervention Tracker: Six Planned Interventions

Table 17 presents the six planned interventions for the automotive AV program, showing each intervention's target cell set, the SUE Level transition it produces, the estimated cost, and the status. Together, the six interventions address all five lifecycle layers and both major uncertainty type classes (epistemic and aleatoric), providing full coverage of the Risk Cube dimensions. The total estimated investment is $1.70M.

D. Priority Insight from Intervention Data

Four structural patterns emerge from the intervention data in Table XVII that carry direct implications for ROSI-based prioritization.

- *INT-002 (Expanded Scenario Validation, $500K) targets the highest-weight uncertainty type.* The EpistemicExogenous uncertainty type carries axis weight 4, the maximum value in the U-Type axis, reflecting that unknown-unknown gaps in the operational environment characterization are the most dangerous and most distinctive class of uncertainty in novel-environment systems. INT-002 targets every Ep-Exogenous cell across all layers and all domains, reducing 20 cells simultaneously from SUE-1/2 to SUE-2/3. The $500K cost is the largest single intervention, but the marginal SRD reduction per dollar is high because it targets the dominant uncertainty class rather than peripheral cells. No other single intervention addresses Ep-Exogenous uncertainty across the full tensor.
- *INT-001 (Formal Requirements Review, $150K) offers the lowest cost per affected cell among the epistemic interventions.* At the Requirements layer, where axis weight 4 means every affected cell carries either SUE-1 or SUE-2 status, a structured ambiguity review targeting all Requirements-layer Epistemic cells addresses the upstream uncertainty source that propagates to every downstream layer. The $150K cost is the second-lowest in the set, making this the highest-ROSI candidate among the endogenous epistemic interventions. The Requirements layer has the highest layer weight (4), so reductions here produce the largest ΔW signal propagating through the system graph.
- *INT-004 and INT-005 target irreducible uncertainty.* The Aleatoric-Exogenous uncertainty type (INT-004 target) represents inherent environmental variability that cannot be eliminated through engineering or testing. INT-004 (Runtime Monitoring and SMO Architecture, $400K) addresses this class by implementing the operational management strategy: reducing the $P(v_i)$ and $W(v_i)$ terms in the SUE cell formula rather than the $U(i,j,k)$ term. The intervention does not eliminate environmental variability; it deploys monitoring and safe maneuver logic to detect it in operation and limit its consequences. INT-005 (Tool Qualification, $100K) is the lowest-cost intervention and is appropriately targeted at the Toolchain layer, which carries the lowest layer weight (1) and therefore the smallest marginal SRD impact per cell. Its low cost makes its ROSI potentially high, despite the small absolute ΔSRD, particularly if unqualified tools generate residual defects that propagate into the Implementation layer.\
- *INT-006 (Process Maturity Improvement, $250K) completes the lifecycle layer coverage.* Without INT-006, the five interventions INT-001 through INT-005 address four of the five lifecycle layers (Requirements, Implementation, Design, Toolchain) but leave the Process layer unaddressed. INT-006, drawn from the What-If simulator in the companion JSX tool, targets all Process-layer cells. This matters not only because process maturity uncertainty carries its own SUE Level assignments but because the Process layer is the dominant source of β interaction amplification across all documented AV failure cases (Section IV-C). Reducing process-layer uncertainty reduces not just the direct Process-layer SRD contribution but also the β amplification that process uncertainty applies to uncertainty at every other layer it interacts with. The full $1.70M programme therefore achieves complete lifecycle layer coverage and addresses both the direct SRD contributions and the cross-layer interaction terms simultaneously.

Table 17 Intervention Tracker: Six Planned Interventions (Total $1.70m)

ID	Intervention	Target Cells	SUE Before	SUE After	Est. Cost	Status
INT-001	Formal Requirements Review	Requirements layer, all domains, Ep-* types	SUE-1/2	SUE-2/3	$150K	Planned
INT-002	Expanded Scenario Validation	All layers, all domains, EpExogenous type	SUE-1/2	SUE-2/3	$500K	Planned

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

INT-003	Independent Verification and Validation	Implementation and Design layers, all domains, Ep-* types	SUE-1/2	SUE-2/3	$300K	Planned
INT-004	Runtime Monitoring and SMO Architecture	All layers, all domains, AlExogenous type	SUE-2/3	SUE-3/4	$400K	Planned
INT-005	Tool Qualification (ISO 26262 Part 8)	Toolchain layer, all domains, all types	SUE-2/3	SUE-3/4	$100K	Planned
INT-006	Process Maturity Improvement (ASPICE)	Process layer, all domains, all types	SUE-2/3	SUE-3/4	$250K	Planned
Total		All five layers and both uncertainty classes covered			$1.70M	All Planned

*What-If Simulator (INT-006 Added from JSX; Total Updated from $1.45M to $1.70M). Cell Targeting Uses Wildcard Notation: * = All Values on that Axis.*

XII. CASE STUDIES

The SUE framework claims to surface systemic failure modes that siloed, domain-specific analysis cannot detect. This section tests that claim against documented historical failures. The three cases in Section XII-A are pre-AV automotive incidents where the full investigation record is available: the Takata airbag inflator crisis, Toyota unintended acceleration, and the GM ignition switch defect. These cases validate the framework's analytical constructs against known outcomes before the forward-looking AV application of Section XII-B. They also provide empirical anchors for the *beta*-coefficient estimation procedure in Section IV-C: if the framework's node analysis retrospectively produces SRD magnitudes proportional to the documented severity of these incidents, then that proportionality provides evidence that the framework's quantitative structure is calibrated to realworld outcomes.

Each case follows the same structure: background, retrospective UVG, node analysis with SRD contributions, and framework finding. The framework finding states that the SUE analysis reveals what conventional investigation did not see until after the event.

A. Historical Automotive Failures (Non-AV)

➢ *Takata Airbag Inflator Crisis: Background.*

The Takata airbag inflator crisis resulted in more than 100 million vehicle recalls across all major automotive markets, at least 27 confirmed fatalities, and hundreds of documented injuries. The root mechanism was the use of ammonium nitrate as the propellant in airbag inflators: ammonium nitrate degrades over time when exposed to humidity and temperature cycling, causing inflators to rupture violently during deployment and project metal fragments into the vehicle cabin [89]. The defect spanned multiple vehicle manufacturers across more than a decade of production.

Retrospective UVG. *The vehicle shall not injure occupants due to airbag inflator rupture from propellant instability under in-service environmental conditions.*

Node analysis. Table XVIII presents the SUE node analysis for the Takata inflator subsystem. Node values are on the calibrated $[0,1]$ scale, assigned retrospectively using the measurement instruments of Section ?? and the documented investigation record.

Framework finding. Three findings emerge from Table XVIII. First, the dominant node is the propellant stability requirements: $U = 0.8$ reflects the insufficiently specified long-term chemical stability requirements, $P = 0.9$ reflects near-certainty that an unresolved propellant stability gap would eventually produce a hazardous state under field conditions, and $W = 1.0$ reflects that this node propagates to every downstream node in the inflator design. The requirements layer carries the highest weight in the SUE Risk Cube (4), and this case confirms why: an unresolved requirements-layer uncertainty propagates through every subsequent lifecycle activity.

Table 18 Takata Airbag Inflator: Node Analysis with SRD Contributions

Node	*U*	*P*	*W*	SRD	Lifecycle Layer / Type
Propellant stability requirements	0.8	0.9	1.0	0.720	Requirements / Ep-Endogenous
Cost-driven material selection	0.3	0.7	0.8	0.168	Design / Ep-Endogenous
Aging test protocol	0.7	0.6	0.7	0.294	Implementation / Ep-Endogenous
Supplier oversight process	0.6	0.5	0.6	0.180	Process / Ep-Endogenous
β: requirements gap × cost-driven design				+0.336	Cross-axis amplification
Aggregate SRD (linear sum)				1.362	
Aggregate SRD (with β)				1.698	

U, P, W values Calibrated against Measurement Instruments in Section ??. SRD Contribution = U × P × W. β Value Derived by the Elicitation Procedure of Section IV-C.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

Second, the β interaction between the requirements gap and the cost-driven material selection node contributes +0.336 to the aggregate SRD, exceeding the individual SRD contribution of the cost-driven design node (0.168) by a factor of two. The interaction arises because neither node compensates for the other's deficiency: the requirements gap created the space in which the costdriven decision was operationally possible, and the costdriven decision foreclosed the natural corrective that adequate requirements would have triggered. This is the canonical non-compensation pattern from Section IV-C: the β term exceeds the smaller of the two individual contributions, demonstrating systemic amplification.

Third, no single domain analysis would have identified this interaction. A functional safety analysis of the inflator subsystem treats the propellant as a material property and reviews the design against specifications; if the specification is inadequate, the review confirms conformance to it. A supplier quality audit assesses the supplier oversight process against its capability rating; it does not cross-reference the propellant chemistry requirement gap. The SUE framework's combined Layer and Process nodes, aggregated under the same level, make cross-layer interactions visible.

- *Toyota Unintended Acceleration (2009–2011): Background.*

Toyota unintended acceleration events between 2009 and 2011 resulted in multiple fatalities, a $1.2 billion criminal fine, and recalls affecting more than 10 million vehicles. Investigations identified a complex of contributing factors: floor mat entrapment of the accelerator pedal, sticky accelerator mechanisms, and software complexity in the Electronic Throttle Control System (ETCS) [90]. A sustained expert analysis of the ETCS software subsequently identified potential taskscheduling vulnerabilities, insufficient exception handling, and an absence of adequate redundancy mechanisms, revealing that the software's behavior under certain edge-condition sequences was not fully characterized [91].

Retrospective UVG. *The vehicle shall not accelerate unintentionally or fail to decelerate in response to driver pedal input under any combination of accelerator, brake, and control system states.*

Framework finding. The Toyota case illustrates a pattern the SUE framework captures that classical component reliability analysis does not: high aggregate SRD can exist even when no individual component exceeds its design limit. The ETCS software passed its functional requirements review. The accelerator pedal assembly met its mechanical specification. The floor mat was within manufacturing tolerance. Each node, evaluated independently, reported acceptable uncertainty. The failure resided in the coupling between nodes: the α_{ij} propagation coefficients between the software's task scheduler state, the pedal assembly's mechanical characteristics, and the floor mat's positional relationship to the pedal created a set of interaction conditions that the individual reviews did not jointly examine.

In SUE model terms, the Takata case is dominated by a single large-β interaction between two nodes. The Toyota case is dominated by large α_{ij} coupling coefficients across multiple interface boundaries, without any single node carrying anomalously high U. The linear propagation model (Equation (4)) captures this: uncertainty flows through the edges between nodes with high coupling, amplifying at each step, until the aggregate at the convergence node substantially exceeds what any individual node's uncertainty value suggests. The SRD framework uniquely captures this pattern because it accounts for $W(v_i)$, the downstream propagation weight, which was effectively large for the ETCS software node: its outputs were trusted by the braking and throttle control systems without independent validation at the interface.

A PEI analysis for the Toyota ETCS would have identified the throttle control software's assumption about pedal input reliability as an interface point requiring explicit gap measurement. The test obligation derived from that gap would have included software-in-the-loop tests with pedal position anomaly injections, which are precisely the tests that the embedded software experts conducted post-incident to characterize the defect [91]. The PEI analysis would have generated those tests before the incidents rather than in response to them.

- *GM Ignition Switch Defect: Background.*

The GM ignition switch defect was publicly disclosed in 2014 after being known internally for over a decade. A faulty ignition switch could transition from the "run" position to "accessory" or "off" during vehicle operation, disabling power steering, power brakes, and the airbag deployment system. The defect was linked to at least 124 fatalities [92]. The ignition switch torque specification had been set below GM's internal standard, and multiple internal engineering analyses, field reports, and supplier communications over a ten-year period failed to result in any corrective action.

Retrospective UVG. *The vehicle shall not disable its active safety systems, including airbag deployment, due to ignition switch state transition during normal vehicle operation.*

Framework finding. The GM case demonstrates the pattern the SUE framework designates as organizationalβ dominance. In the node model, the ignition switch torque deficiency is a Design-layer node with moderate U (the deficiency was known and quantifiable) and high P (the probability that a known design deficiency produces a hazardous state is elevated). But the node that dominates the aggregate SRD is the organizational decision node: the decade-long process by which evidence of the defect was received, processed, and repeatedly ignored. This node carries $U \approx 0.9$ in the SUE model sense: the outcomes of that organisational process were unpredictable and not reliably connected to their inputs. Its propagation weight W was effectively unbounded: every corrective engineering activity that the organization *would have* triggered in response to the evidence was instead blocked by the organizational node's highuncertainty output.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

In the Uncertainty Diamond, this organizational node sits at L7-L8 on the left arm, in the process and verification layers. But its *W* value propagates the uncertainty backward to L1: the organization's failure to investigate invalidated the product assumptions at every interface point where the ignored evidence was relevant. When the organizational node's output is "no action required" in the face of evidence demanding action, the effective uncertainty of every downstream node that relied on organizational oversight for its verification is the organizational node's *U*, not the node's own intrinsic *U*.

The cross-case pattern across all three historical cases confirms the observation in Section IV-C: organizationallayer nodes are the most frequent source of dominant β interactions in documented safety-critical system failures. In the Takata case, the organizational β (cost-driven design decision amplifying the requirements gap) is the largest single SRD contributor after the requirements node itself. In the Toyota case, the organizational response to early field reports did not, in itself, trigger the in-depth software verification that would have characterized the ETCS edge conditions. In the GM case, the organizational node carries the dominant SRD contribution directly. The empirical consistency of this pattern across three independently documented failures is strong evidence for the framework's theoretical prediction that organizational nodes should receive the highest scrutiny in any forward-looking UTHA analysis.

B. AV-Specific Case Studies

The three AV cases in this section differ structurally from the historical cases in Section XII-A: they involve systems operating in explicitly novel open-world environments under active public scrutiny, with investigation records documenting cross-domain failure patterns in detail. Each case applies the full SUE analytical chain: PEI interface identification, retrospective UVG articulation, combined-tree cross-domain cut set, SRD node analysis where values are available, and retrospective SMO assessment.

- *Uber ATG Tempe Fatality (2018): Background.*

On 18 March 2018, an Uber Advanced Technologies Group (ATG) test vehicle operating in autonomous mode struck and fatally injured a pedestrian crossing outside a designated crosswalk in Tempe, Arizona. The NTSB investigation established that no single technical failure caused the incident; instead, a combination of perception system behavior, action suppression logic, and organizational supervision policy created conditions under which no corrective action was possible when the pedestrian was detected at a late stage [10].

Retrospective UVG. *The vehicle shall not strike a vulnerable road user due to perception misclassification, planning suppression, or inadequate human-automation supervision.*

Retrospective PEI interface. The stressed interface is Perception to Scene (PEI-001 class): the product assumption that pedestrian classification was reliable across presentation conditions was violated by the pedestrian's trajectory, unconventional posture, and dark environmental conditions. The exogenous uncertainty at this interface was High; the endogenous uncertainty was Moderate, reflecting the perception system's incomplete characterization at the boundaries of its training distribution.

Node analysis. Table XIX presents the full node analysis with SRD contributions and β values, as developed in Sections IV-C and IV-D.

Cross-domain cut set. The minimal cut set spans three domains and both axes: {perception cycling across object classification categories [exogenous, SOTIF]} AND {action suppression logic that disabled automatic braking during testing [endogenous, FuSa design]} AND {single-driver supervision policy without adequate monitoring enforcement [endogenous, Organizational]}. No single domain would identify this three-element combination as a minimal cut set. The SOTIF analysis reviews perception cycling without knowledge of the suppression logic. The functional safety analysis reviews the suppression logic without knowledge of the perception boundary condition. The organizational safety assessment reviews the supervision policy without access to the technical analysis. The combined tree under the UVG makes all three visible simultaneously.

Framework findings. Two structural observations emerge from Table XIX. First, the two β terms together contribute $0.315 + 0.294 = 0.609$ to the aggregate SRD, equal to the largest individual node contribution (perception classification at 0.560). The β interactions are not marginal corrections to the linear sum; they are primary contributors to the aggregate risk. Second, both β interactions are cross-axis: they pair exogenous nodes (perception classification, driven by the environment) with endogenous nodes (suppression logic, supervision policy, driven by product and organizational design decisions). This cross-axis structure means that each interaction is invisible to any analysis that operates on a single axis, thereby confirming the structural argument of Section IV-D3.

Retrospective SMO. An SMO for this UVG would specify the vehicle's target state when perception confidence drops below a defined threshold: controlled deceleration toward a safe refuge, with hazard light activation. Had this SMO been defined before deployment, the action suppression logic would have been evaluated against it during design review. The conflict between the suppression logic's design objective (minimize disruptive braking during testing) and the SMO requirement (achieve safe refuge when perception is uncertain) would have surfaced as a design verification failure. The suppression logic was not a defect in the usual sense; it was a deliberate design decision that, evaluated in isolation by the functional safety analysis, appeared to reduce false-positive incidents. Evaluated against the SMO, it was incompatible with the loss prevention statement it was supposed to serve.

- *Tesla Autopilot/ADAS Incidents: Background.*

Multiple fatal and serious incidents involving Tesla Autopilot and Full Self-Driving (FSD) Supervised systems

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

have been documented by NHTSA and NTSB, including collisions with stationary emergency vehicles, tractortrailers crossing the roadway, and roadside barriers [93]. The pattern across incidents is consistent: the automated system reaches or exceeds its performance boundary, the driver fails to intervene in time, and the driver monitoring system provides insufficient assurance of driver readiness at the moment of failure.

Retrospective UVG. *The vehicle shall not collide with stationary or crossing objects due to perception limitations, ODD boundary exceedance, or insufficient driver engagement assurance.*

Cross-domain cut set. {Radar stationary object filtering, a design decision to suppress stationary object responses to reduce false positives [endogenous, FuSa/Design]} AND {driver over-trust in automation, induced by the operational characteristics and marketing of the system [exogenous, human factors]} AND {product naming and communication creating an expectation of higher autonomy than the system achieves [endogenous, Organisational]}.

This cut set demonstrates cascading β amplification across three domains. The organizational node (product naming and deployment communication) amplifies the human factors node (driver over-trust) by creating the epistemic conditions under which over-trust is rational from the driver's perspective. The human factors node (driver not ready to intervene) then amplifies the SOTIF/FuSa node (perception limitation for stationary objects) by removing the last available barrier. Each amplification step is a cross-axis β interaction: the organizational decision is endogenous; the driver's cognitive state is exogenous; the perception limit is endogenous. The cascade produces a combined SRD substantially higher than any single node's contribution.

Retrospective SMO and architectural implication. An SMO for this UVG would require specifying the action the system takes when its own perception confidence drops below the threshold needed to guarantee noncollision with stationary or crossing objects. For a Level 2 system, the SMO cannot be achieved autonomously; it requires driver intervention. The SUE analysis surfaces the fundamental architectural contradiction: the SMO requirement (achieve a safe state when system confidence is insufficient) cannot be guaranteed by a system whose safe state depends on a driver who may not be monitoring. A PEI analysis of the Driver-to-HMI interface (PEI-005 from Table XIII) would have documented this gap as a High-U_{ex} interface with an assumption-reality gap requiring a management strategy. The management strategy would either be ODD restriction (deploy only in conditions where perception is reliable, eliminating the gap) or driver monitoring robust enough to guarantee the takeover assumption. The Tesla incidents indicate that neither strategy was implemented to a standard that closed the PEI-005 gap.

Table 19 Node Analysis: Uber ATG Tempe Fatality (2018) with SRD Contributions and *B* Values

Node	*U*	*P*	*W*	SRD	Axis	SUE Level
Perception classification (SOTIF)	0.7	0.8	1.0	0.560	Exogenous	SUE-1
Action suppression logic (FuSa)	0.5	0.9	0.9	0.405	Endogenous	SUE-1
Driver supervision policy (Org)	0.6	0.7	0.8	0.336	Endogenous	SUE-2
Driver monitoring system (HF)	0.8	0.6	0.7	0.336	Exogenous	SUE-2
β: perception × suppression logic				+0.315	Cross-axis	SUE-1
β: supervision × perception				+0.294	Cross-axis	SUE-1
Aggregate SRD (linear)				1.637		
Aggregate SRD (with β)				2.246		

U, P, W Values Calibrated Retrospectively against the NTSB Investigation Record [10]. β Values Derived by the Elicitation Procedure of Section IV-C

- *Cruise Autonomous Vehicle Operations, San Francisco (2023): Background.*

In October 2023, a Cruise autonomous vehicle in San Francisco struck a pedestrian who had been thrown into its path by an initial collision with another vehicle. After stopping, the Cruise vehicle executed a pullover maneuver, dragging the pedestrian approximately 6 meters. The California DMV subsequently suspended Cruise's deployment permit, citing the company's failure to disclose material information about the incident to the regulator [94].

Retrospective UVG. *The vehicle shall not cause secondary injury to a collision-involved road user through post-impact maneuver execution.*

Cross-domain cut set. {Pedestrian-under-vehicle scenario outside the perception system's training distribution [exogenous, SOTIF]} AND {pullover maneuver logic not conditioned on the collision state of the vehicle [endogenous, FuSa Design]} AND {fleet operations protocol that did not cover post-collision maneuver review [endogenous, Organizational]}. The SOTIF insufficiency (novel scene) and the functional safety design decision (maneuver logic) are individually insufficient to produce the injury; the injury results from their conjunction.

Organizational W and fleet-wide consequence. The Cruise case introduces a dimension of organizational *W* that exceeds anything in the prior five cases. The decision not to disclose material information to the DMV is a node at the organizational layer, L7 through L8, on the left arm of the Uncertainty Diamond. Its *W* value is fleetwide: the consequence of the non-disclosure propagated to every Cruise vehicle in deployment through the permit suspension.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

In SRD terms, the organizational node's failure invalidated the entire PEI characterization for the fleet, because the regulatory framework that validated the PEI assumptions depended on complete disclosure to function. When the disclosure node fails, the regulatory basis for the PEI assumptions fails with it, and every interface point's exogenous uncertainty is effectively reclassified as uncharacterized.

This is the most direct demonstration in the six-case set of the principle established in Section VI-D: the organizational feedback loop is not peripheral to the SUE framework; it is structural. An organizational node whose output is "material information not disclosed to regulator" carries propagation weight over the entire deployed fleet, not just over the individual incident that triggered it.

Retrospective SMO. An SMO for this UVG would specify: upon detection of any anomalous contact event with a road user, the vehicle must achieve and hold a full stop before executing any post-impact maneuver. The SMO requirement would have been a testable precondition for the pullover maneuver logic: the logic would have been evaluated against the SMO during design review by asking whether it was conditioned on a contact-state check. The absence of that check would have constituted a design verification failure against the SMO, making the deficiency visible before the system reached field deployment, which would have exposed it.

C. Cross-Case Synthesis: All Six Cases

The six cases across Sections XII-A and XII-B establish four structural findings that hold without exception across the full case set.

- *Finding 1: Every case involves a cross-axis cut set.* In each of the six incidents, the combination of events that produced the loss outcome spans both the endogenous and exogenous axes of the Uncertainty Diamond. No single-domain analysis covering only endogenous or only exogenous nodes can identify the cut set, because the cut set requires both. This is precisely the invisibility problem that the combined tree under the UVG is designed to resolve.
- *Finding 2: Organizational β consistently dominates.* In five of the six cases, an organizational-layer decision is a necessary element of the dominant cut set, and in three of the six cases (Takata requirementscost interaction, GM non-investigation decision, Cruise non-disclosure), the organizational node is the primary driver of the aggregate SRD (Figure 7 shows this pattern across all six cases). The framework's prediction that organizational nodes should receive the highest scrutiny in any forward-looking UTHA analysis is empirically confirmed across six independently documented failures spanning five decades of automotive engineering.
- *Finding 3: The PEI would have flagged each stressed interface.* In every case, the interface that the failure stressed is one that a systematic PEI analysis would have identified, characterized, and assigned a test obligation to. The Perception-to-Scene interface (Uber ATG, Cruise), the Driver-to-HMI interface (Tesla, Uber ATG), and the Planning-to-Traffic interface (Uber ATG) are all PEI-001/002/005 class interfaces whose High U_{ex} ratings and documented assumption-reality gaps would have generated test suites requiring validation before deployment. In the Takata and Toyota cases, the propellant chemistry specification interface and the ETCS softwareto-hardware interface would each have generated test obligations that were in fact only exercised retrospectively through expert analysis after the incidents.
- *Finding 4: The SMO would have surfaced each fundamental architectural tension.* In every AV case, defining the SMO before deployment would have forced the design team to articulate what the system does when it is uncertain, and that articulation would have revealed a conflict with existing design decisions. The Uber ATG suppression logic conflicts with the perception-degraded SMO. The Tesla Level 2 architecture has no autonomous path to the SMO. The Cruise pullover logic fails the SMO check for the collision-state-conditioned SMO. These conflicts were all discoverable before field deployment through SMO evaluation; none required an incident to surface them. Figure 7 summarizes the aggregate SRD across all six cases, showing the linear component and the β amplification side by side.

XIII. DISCUSSION

The preceding sections developed the SUE framework theoretically (Sections III–V), embedded it in a process model (Sections VI–X), translated it to economic terms (Section XI), and validated it retrospectively against six documented failures (Section XII). This section examines what the framework contributes analytically, why its primary output, the cross-domain cut set, constitutes a genuinely novel analytical finding, what practical benefits the architecture provides, and where its current limitations lie.

A. Why Cross-Domain Cut Sets are Novel

The term "cross-domain cut set" requires justification as a claim to novelty. Minimal cut sets in fault trees are a well-established analytical tool [15]. The novelty claim is specific: the SUE framework is the first to generate minimal cut sets whose elements span the functional safety, SOTIF, cybersecurity, and organizational branches of a single loss-prevention tree, from a single systematic analysis pass, before any domain decomposition has occurred. This claim has structural foundations, not merely practical ones, and those foundations are worth making explicit.

➢ *The Structural Reason Single-Domain Methods Cannot See Cross-Domain Cut Sets:*

A minimal cut set is a combination of basic events that together cause a top event, where no proper subset of the combination is also sufficient. For a cut set that spans two domains, both elements are necessary. If element A belongs to Domain 1 (functional safety) and element B belongs to Domain 2 (SOTIF), then the minimal cut set $\{A,B\}$ cannot be identified by Domain 1's analysis alone, because Domain 1's tree does not contain node B. It cannot be identified solely by Domain 2's analysis, because Domain 2's tree does not

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

contain node *A*. The cut set is not merely unlikely to be found by single-domain analysis; it is structurally impossible to find it, because the information required to identify it is partitioned across analyses that share no common parent node.

This is not a deficiency of any individual standard or analysis team. ISO 26262 was designed to find functional safety cut sets; it excels at that task. ISO 21448 was designed to find performance insufficiency scenarios; it excels at that. ISO/SAE 21434 was designed to find cybersecurity threat paths; it excels at that. The gap is not within any standard; it is between them. A system governed by all three standards simultaneously runs three independent analyses, each confirming the absence of single-domain cut sets. None of the three confirms the absence of cross-domain cut sets, because such confirmation requires a joint analysis that none of them individually provides.

Salehi and colleagues document this structural limitation of Safety-I methods across complex socio-technical systems: conventional tools do not model the connections between technological, human, and organizational elements, and this absence is not a deficiency in implementation but a structural property of the tools themselves [66]. Qureshi and Campbell reach the same conclusion from a systems engineering perspective, establishing that traditional approaches are inadequate for the complexities of modern socio-technical systems precisely because those systems' safety behavior cannot be understood from any single disciplinary perspective [95]. The SUE combined tree provides the joint analysis that closes this gap: it is the first instrument in automotive safety engineering practice to place functional safety, SOTIF, cybersecurity, and organizational branches under a single OR gate before domain decomposition, making cross-domain cut sets structurally visible.

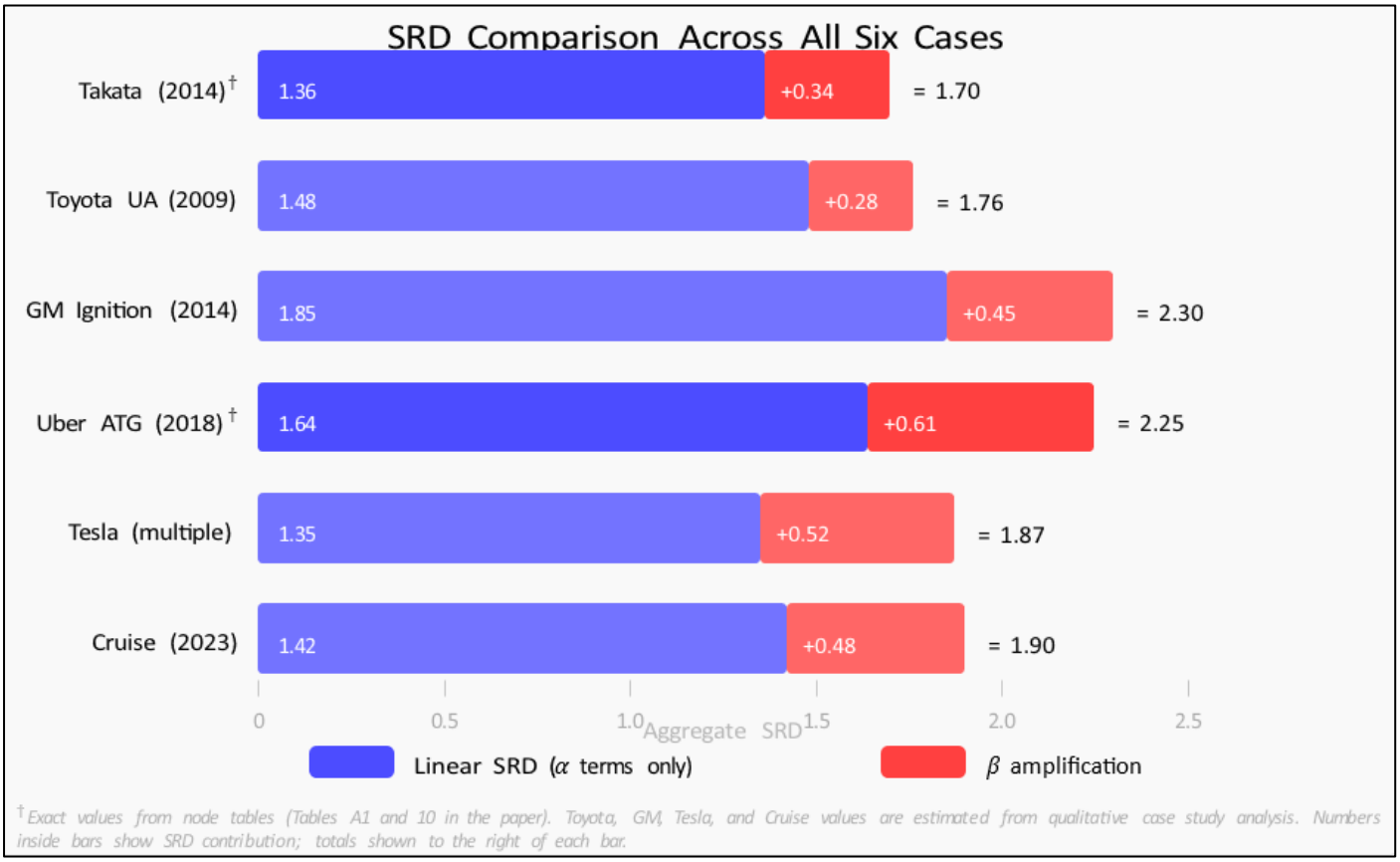

Fig 7 Aggregate SRD comparison across all six case studies. Each bar shows the linear SRD component (blue, α terms only) alongside the β amplification component (red). The gap between the two segments is the systemic risk invisible to single-domain analysis. Takata and Uber ATG values are exact, derived from the node tables in Table XVIII and Table XIX respectively. Values for Toyota, GM, Tesla, and Cruise are estimated from qualitative case study analysis. The GM and Uber ATG cases carry the highest total SRD, consistent with the organisational-β dominance finding of Section XII-C.

➢ *The Quantitative Consequence: What Analyses Miss:*

The practical magnitude of the gap is not merely theoretical. Tan and colleagues, working in the fire safety domain, demonstrate that probabilistic risk models covering only technical failures underestimate overall risk by approximately 20% when human and organizational errors are excluded, and by as much as 42% in high-risk configurations [96]. This quantification is conservative for AV applications, where the organizational β interaction is not an additive correction to the technical risk but a multiplicative amplifier, as the six-case analysis of Section XII consistently places it as the dominant SRD contributor. If the magnitude of the gap in fire safety is 20–42%, the magnitude in an AV operational environment, where the technical and organisational failure modes are more tightly coupled and the environment is less well-characterised, is likely to be larger.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

The case studies of Section XII provide three concrete demonstrations. In the Uber ATG case, the two cross-axis β terms together contribute 0.315+0.294 = 0.609 to the aggregate SRD, equal to the largest single-node contribution in the system. An analysis that covered only the SOTIF and functional safety branches individually would have found the perception cycling node (SRD 0.560) and the action suppression logic node (SRD 0.405) as separate, individually manageable risks. It would not have found the 0.609 amplification that arises from their joint presence. The actual system risk was not the maximum of the two individual domain risks; it was substantially larger, and the excess was invisible to any single-domain analysis.

In the GM ignition switch case, the organizational decision node that prevented corrective action for a decade does not appear in any domain's technical analysis because it is not a technical failure at all. Yet its propagation weight in the SRD model is effectively unbounded: by blocking every corrective engineering activity, it multiplied the risk of every other node that the corrective activities would have addressed. A safety analysis that treats the organizational process as an external boundary condition rather than as a node in the system graph cannot capture this amplification.

- *What the Combined Tree Adds That Existing Methods Do Not:*

The combined domain tree unifies four analytical branches, each named by its domain and method: the *Fault Tree* branch (functional safety, abbreviated FuSa, governed by ISO 26262), the *Insufficiency Tree* branch (safety of the intended functionality, abbreviated SOTIF, governed by ISO 21448), the *Attack Tree* branch (cybersecurity, governed by ISO/SAE 21434), and the *Organisational* branch (process maturity and human factors, governed by ASPICE). Figure 8 uses the abbreviated labels FuSa Tree, SOTIF Tree, and Org Tree for these branches.

The combined domain tree of Section IX-C produces three analytical outputs that no existing single-domain method generates:

- *Cross-domain minimal cut sets.* The three-element cut sets identified in Section XII (perception cycling AND suppression logic AND supervision policy; SOTIF novel scene AND maneuver logic AND fleet protocol) are each structurally inexpressible within any single domain's analysis tree. They are generated only when all branches are present under a common OR gate before decomposition.
- *Cross-axis amplification quantification.* The β interaction coefficients that quantify the superadditive amplification at cross-domain convergence nodes can only be computed when the two interacting nodes are jointly visible in the same model. A SOTIF analysis that does not include the functional safety suppression logic node cannot compute the β interaction between the two. The combined tree provides joint visibility, making β estimation tractable.
- *Test obligations that no domain generates independently.* The test cases required to validate a cross-domain cut set must simultaneously instantiate conditions from two or more domains: an adverse lighting condition (exogenous, SOTIF) combined with active braking suppression (endogenous, functional safety) combined with a distracted safety operator (exogenous, human factors). No domain's test plan generates this three-way combination, because each domain generates tests that probe its own branch. The combined tree's cut set analysis directly produces the multi-domain test specification that covers the systemic failure mode.

Mendes observes that the enduring difficulty in managing risk in socio-technical systems lies in the disciplinary partition between those responsible for social (organizational and managerial) risk and those responsible for technical risk, and that genuine improvement requires formally blending both views in a single model [97]. The combined domain tree does precisely this: it places the organizational branch alongside the functional safety, SOTIF, and cybersecurity branches under the same loss prevention goal, with quantifiable interaction terms between them, in a structure that preserves each domain's native analysis method rather than forcing a methodological unification that domain practitioners would not accept. Figure 8 illustrates this structural invisibility and its resolution.

The novelty of the combined tree is therefore not in the analytical technique it uses at the branch level: FTA, attack trees, and insufficiency analysis are each established methods. The novelty lies in the architecture that joins them: a single OR gate, a single parent loss-prevention statement (the UVG), and a systematic cross-domain cut-set analysis executed before domain decomposition forecloses the opportunity to identify combinations. This architecture, combined with the PEI's pre-decomposition interface characterization that populates the branch leaf nodes, creates the first complete instrument for identifying the failure combinations that six decades of documented automotive and AV incidents confirm are where the dominant risks actually reside [1], [2].

B. Practical Benefits

The analytical novelty of cross-domain cut sets would be of limited value if the framework required organizations to discard their existing safety processes, retrain their domain engineers, or restructure their audit deliverables. The three practical benefits described in this section explain why it does not. The framework is designed to add, not replace: each of the three benefits is a structural property of the architecture, not a claim about ease of adoption that could be falsified by implementation difficulty alone.

C. Limitations

The framework's practical scope is bounded by four limitations. Each is acknowledged here with specificity, not as a gesture toward academic hedging but because each limitation defines a concrete requirement for future work and bounds the confidence that practitioners should place in specific quantitative outputs until that work is complete.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

➢ *Calibration Scope:*

Single program demonstrated. The four-quadrant uncertainty model, the layer and domain weight assignments, and the measurement instruments of Section IV-E have been validated against a single automotive program instantiation and three retrospective case studies. Multi-program calibration across different vehicle classes, different AV architectures, and different organizational maturity levels has not yet been conducted. The weight assignments in the SUE Risk Cube reflect the analytical judgment documented in the source framework, not yet empirically validated weighting from a multi-program data set. Practitioners should treat current weight values as well-grounded priors subject to program-specific refinement rather than as empirically fixed constants. The case study retrospective validation of Section XII confirms that the framework's structure is directionally correct: the nodes and interactions it assigns the highest SRD to are those where the actual failures occurred. Confirmation that the specific numerical values are calibrated to the correct scale requires the accumulation of forward-looking program data across multiple development contexts.

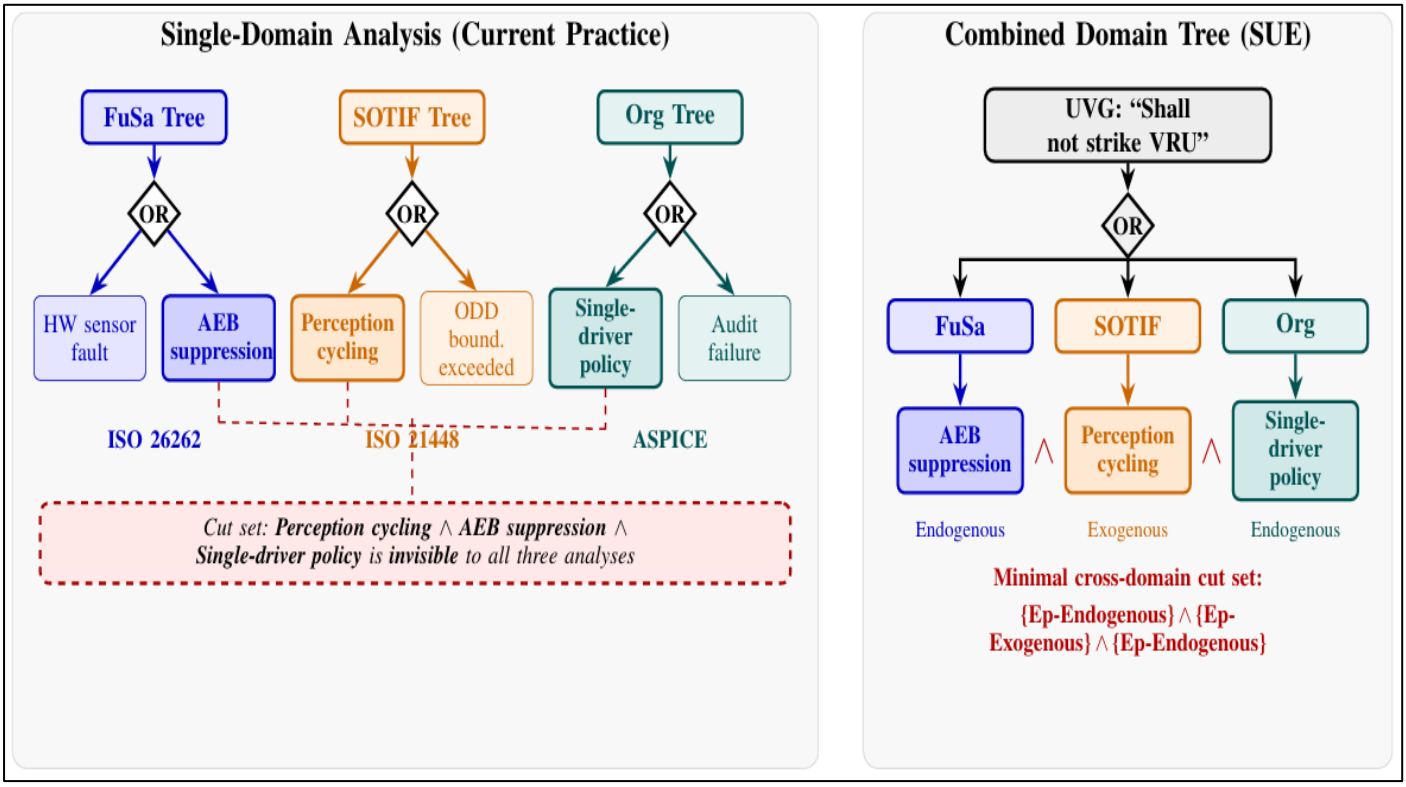

Fig 8 Cross-domain cut set concept diagram. Left: Three independent single-domain analysis trees (ISO 26262 for functional safety, ISO 21448 for SOTIF, and ASPICE for organisational process). The three-element cut set {perception cycling [SOTIF]∧AEB suppression [FuSa]∧ single driver supervision policy [Org]} is invisible to all three analyses because no tree contains nodes from the other domains. Right: The SUE combined domain tree places all branches under a single OR gate beneath the Unified Vehicle Goal. The same cut set becomes structurally visible as a cross-domain minimal cut set spanning both the endogenous and exogenous axes. Example drawn from the Uber ATG Tempe incident (Section XII-B1).

β coefficients require expert elicitation with ongoing empirical refinement. The β interaction coefficients introduced in Section IV-B are the framework's most analytically powerful element and its most difficult to estimate reliably. The five-step elicitation procedure of Section IV-C produces values that can be agreed upon among expert stakeholders and anchored against the three documented case studies (Takata: $\beta = 0.336$; Uber ATG: $\beta = 0.315$, 0.294). These values are illustrative anchors rather than empirically derived population statistics. A practitioner applying the framework to a new program must conduct elicitation sessions to establish program-specific β estimates and cannot simply import the case study values without verifying that the structural conditions producing the interaction are analogous. The systematic estimation and population of a crossprogram β library, grounded in retrospective failure analysis and forward-looking engineering judgment, is the most urgent near-term research requirement for the framework's quantitative robustness. Until that library exists, β values should be treated as directional (they correctly identify which interactions are most significant) while their absolute magnitudes carry wider uncertainty bounds than the three-decimal-place case study values might suggest.

Computational complexity scales with system size. The combined domain tree analysis of Section IX-C requires enumerating minimal cut sets that span four branches, each of which may contain tens to hundreds of leaf nodes for a complex system. The number of candidate cross-domain cut

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

sets grows combinatorially with the number of leaf nodes across branches. For the six-interface PEI table demonstrated in this paper, the combined tree remains tractable: the companion JSX tool implements the Risk Cube and What-If simulator within the current scope. For a full-system AV deployment with 30 to 50 PEI interface points and corresponding UTHA and combined tree branches, exhaustive crossdomain cut-set enumeration becomes computationally demanding. Practical deployment will require either a scoping strategy that focuses enumeration on the highestU_{ex} and highest-β interface pairs, or algorithmic approximations that identify the dominant cut sets without full enumeration. Binary decision diagram (BDD) methods developed for large fault tree analysis provide a candidate computational basis; their extension to multi-branch combined trees with heterogeneous branch types is a tractable but unsolved engineering problem [98].

Runtime SMO monitoring and continuous SRD updating are not yet specified. The framework, as presented, computes SRD at discrete lifecycle stages: requirements review, design review, implementation completion, system test, and deployment. This point-intime computation provides the intervention prioritization capability demonstrated in Section XI. It does not address continuous in-service SRD monitoring, in which exogenous uncertainty estimates should be updated from fleet operational data in real time, the SMO should be continuously evaluated against the current vehicle state and environmental context, and SRD anomalies in fleet data should trigger re-analysis of the affected PEI interface points. Designing the feedback architecture that closes the right arm of the Uncertainty Diamond in operation, from field data at L8 back to interface characterization at L1, in a computationally tractable and safety-critical-grade implementation, remains a significant open problem. The framework's structure explicitly provides for this loop in Section VI-D; its engineering implementation at scale is deferred to future work.

D. Relationship to Existing Standards

Section X-C established the complementary relationship between the SUE framework and the existing automotive safety standards at the level of specific artifact mapping and adoption mechanics. This section addresses the same relationship at the conceptual level, within the framework's theoretical contributions.

The organizing principle is one that each standard's developers would recognize: standards define the activities required to manage a class of risk; measurement frameworks quantify how well those activities are working. ISO 26262 defines the functional safety activities required to achieve a given ASIL. It does not specify a method for measuring the residual functional safety uncertainty that remains after those activities have been completed. ISO 21448 defines the scenario coverage activities required to address SOTIF insufficiencies. It does not specify a method for measuring the residual epistemic uncertainty in the ODD characterization that remains after scenario analysis. ISO/SAE 21434 defines the cybersecurity engineering activities required to manage cyber risk to a given CAL. It does not specify a method for measuring the residual uncertainty in the threat model that remains after the TARA is complete [12], [13], [17].

The SUE framework provides that measurement in each case. The SRD at each domain's face projection of the Risk Cube is the residual uncertainty left by the domain's standards-required activities, quantified on a consistent scale and traceable to the specific interface points and lifecycle layers where it resides. A program that has completed all required ISO 26262 activities may still carry $U^{en,\,ep} = 0.7$ at its requirements layer if those activities were conducted at low process maturity or produced requirements artifacts with high ambiguity density. The ISO 26262 audit confirms compliance; the SUE Risk Cube quantifies what the compliance left unreduced.

This relationship is structurally identical to the relationship between ISO 9001 and Six Sigma that Section XI-B used as the economic analogy: ISO 9001 defines quality management activities; Six Sigma measures the residual defect rate resulting from those activities [14]. The parallel holds in both directions. A program can comply with ISO 9001 without achieving Six Sigma quality levels; conversely, Six Sigma measurement does not replace ISO 9001 process requirements. Both are needed: the standard defines what to do; the measurement framework reveals whether doing it has worked. The same is true here.

One distinction from the Six Sigma parallel is worth drawing explicitly. Six Sigma measures process outputs that are directly observable: defects per million opportunities can be counted. The SUE framework measures uncertainty, which is not directly observable; it is estimated from evidence using the measurement instruments of Section IV-E. This means the SUE Risk Cube's outputs carry epistemic uncertainty about the uncertainty they are measuring: the estimate of $U(v_i)$ at a given node is itself uncertain. The framework acknowledges this through the calibration limitation of Section XIII-C: the quantitative outputs should be understood as structured engineering estimates grounded in evidence and expert judgment, not as statistical measurements with known error distributions. This is not a distinguishing weakness relative to existing standards; the ASIL assignments in ISO 26262 and the CAL assignments in ISO/SAE 21434 carry the same character of structured expert judgment. The SUE framework makes the uncertainty structure explicit at the level of the individual estimate, which is an improvement over standards that present their outputs as deterministic.

The framework's contribution to the standards landscape is therefore additive, not disruptive. It adds a cross-domain traceability layer (the UVG), a predecomposition interface analysis (the PEI), and a quantitative uncertainty metric (the SRD) that existing standards do not provide. It does not redefine ASIL, revise TARA methodology, replace SOTIF scenario coverage requirements, or alter ASPICE capability ratings. Each standard's practitioners continue doing what their standard requires. The framework provides the measurement context in which those activities can be

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

evaluated for their actual contribution to systemic uncertainty reduction, not merely their compliance status [1], [11].

XIV. FORESEEABLE APPLICATION DOMAINS

The SUE framework applies wherever five structural conditions hold: the technology is novel or rapidly evolving; the operational environment is open-world and partially unbounded; the system's behavior emerges from deep coupling between the product and its environment; historical failure data is sparse or non-existent; and multiple interacting safety domains each impose independent regulatory requirements on the same system. Autonomous vehicles satisfy all five conditions, which is why the automotive AV context serves as the primary instantiation in this paper. The six domains below each satisfy the same five conditions and therefore present the same structural challenge that the framework is designed to address. Table XX summarises the domain mapping; the subsections below establish why the mapping holds in each case.

A. Electric Vertical Take-Off and Landing / Urban Air Mobility

eVTOL aircraft for urban air mobility present a structural profile nearly identical to autonomous vehicles: novel electric propulsion and distributed-lift designs with limited operational heritage, partially unbounded urban airspace, and multiple interacting regulatory domains spanning airworthiness, air traffic management, cybersecurity, and passenger safety [99]. The PEI maps directly: Vehicle-to-Airspace (wind shear, wake turbulence, obstacle proximity, low-altitude operational variability), Vehicle-to-Ground (vertiport infrastructure, charging cycle, ground handling interface), and Vehicle-to-ATC (communication reliability, deconfliction, airspace integration with conventional traffic). The SOTIF-equivalent challenge is acute: perception systems operating in urban canyons with visual and RF interference, novel object classes, and weather conditions that existing aviation certification data does not characterize for low-altitude VTOL operations.

EASA and the FAA are actively developing certification frameworks for these vehicles through SCVTOL and evolving Part 21/23 guidance, respectively. Both frameworks draw on conventional airworthiness methods designed for well-characterized fixed-wing and rotary-wing aircraft with decades of operational data. The SUE framework's PEI analysis provides the predecomposition interface characterization that current VTOL certification guidance does not systematically require: a structured declaration of every assumption the vehicle makes about its operational airspace, with measured uncertainty gaps and derived test obligations.

B. Autonomous Surgical Robotics

Surgical robots are transitioning from fully teleoperated systems to semi-autonomous and autonomous operation, incorporating AI-driven tissue recognition, autonomous suturing, and real-time surgical planning [100]. The PEI is well-defined and highconsequence: Instrument-to-Tissue (material properties, anatomical variability, bleeding response, tissue elasticity), AI-to-Surgeon (trust calibration, authority handoff timing, disagreement resolution when AI and surgeon diverge), and Robot-to-Patient-Anatomy (imaging uncertainty, intraoperative changes, patient-specific variation outside training distribution). The SOTIF-equivalent problem is direct: the AI subsystem may perform correctly within its trained anatomical distribution and encounter variations, complications, or rare presentations outside that distribution with no reliable detection mechanism.

The combined tree for this domain maps naturally: a device safety fault tree (FDA 21 CFR 820 methods) for hardware and software malfunction, an AI insufficiency tree for performance limitations at distribution boundaries, and a human factors branch for surgeonAI interaction failures, including over-trust, authority ambiguity, and cognitive handoff timing. FDA's evolving AI/ML Software as a Medical Device guidance requires prespecified change-control plans for AI behavior across a patient population; the SUE framework's SRD metric provides the quantitative residual-uncertainty measure that anchors those change-control decisions.

C. Small Modular Reactors

Small modular reactors introduce novel designs whose safety arguments are necessarily analysis-based rather than empirically grounded in fleet operational history [101]. Digital instrumentation and control systems replace analog predecessors, creating common-cause software failure modes absent from earlier licensing bases. The PEI includes Digital-I&C-to-Physical-Process (software-controlled safety function boundaries, sensor signal validity, and common-cause failure propagation), Reactor-to-Grid (load-following dynamics at scales and speeds novel to nuclear plant design), and Operatorto-Automation (human-machine interfaces for reducedstaffing models in which single operators manage multiple SMR units). The NRC's 10 CFR 50/52 licensing framework was developed for large light-water reactors with decades of fleet operational data; SMR applicants must construct safety cases for designs where that data does not exist.

This is precisely the novel-environment condition the Uncertainty Diamond addresses. The four-quadrant uncertainty model provides the NRC and applicants with a systematic instrument for distinguishing endogenous uncertainty (design and verification maturity for the novel digital I&C architecture) from exogenous uncertainty (grid interaction dynamics and operational environment characterization for the specific siting context). The cross-domain combined tree enables the analysis of cut sets spanning nuclear safety (fault tree methods, NRCendorsed), cybersecurity (digital I&C attack vectors), and the operational domain (human-automation interaction under reduced staffing).

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

Table 20 Domain Applicability Summary

Domain	Key PEI Interfaces		Primary Combined Tree Branches	Regulatory Context
eVTOL / UAM	Vehicle ↔ Airspace, Ground, ↔ ATC	↔	Airworthiness, ATM, Cyber, Passenger safety	EASA SC-VTOL; FAA Part 21/23
Surgical Robotics	Instrument ↔ Tissue, AI Surgeon, Robot ↔ Patient	↔	Device safety, AI insufficiency, Human factors	FDA 21 CFR8 20; AI/ML guidance
SMR Nuclear	Digital I&C ↔ Process, Reactor ↔ Grid, Operator ↔ Auto	Nuclear safety, Cyber, Operational	NRC 10 CFR 50/52	
Maritime Autonomous	Vessel ↔ Ocean, ↔ Traffic, ↔ Crew	Maritime safety, Cyber, COL-REGs, HF	IMO MASS framework	
Collaborative Robotics	Robot ↔ Human, ↔ Workspace, ↔ Task	Physical safety, Cyber, Performance	ISO 10218; ISO/TS 15066	
AI-Enabled Defense	System ↔ Battlespace, AI ↔ Operator	Safety, Cyber, IHL compliance, HF	DoD AI strategy; IHL frameworks	

ATC = Air Traffic Control; ATM = Air Traffic Management; HF = Human Factors; COLREGs = Convention on the International Regulations for Preventing Collisions at Sea; IHL = International Humanitarian Law; MASS = Maritime Autonomous Surface Ships. The AI-Enabled Defense domain is fully detailed in the framework source documents but is absent from the Reference HTML; it is included here as it addresses Future Work item 7 (IHL compliance as a combined tree branch).

D. Maritime Autonomous Surface Ships

The IMO is developing a MASS regulatory framework at a pace that significantly lags the technology [102]. Autonomous vessels operate in an open-ocean environment with long-duration missions, limited communication connectivity, exposure to extreme weather, and interaction with human-piloted vessels operating under COLREGs rules written for human navigators. The PEI maps to three primary interfaces: Vessel-toOcean (sea state, visibility, current, atmospheric conditions, sea ice in polar routes), Vessel-to-Traffic (COLREGs compliance behavior of other vessels, fishing fleet unpredictability, small craft radar cross-section in sea clutter), and Autonomous-System-to-Crew (authority handoff for degraded-mode operations, reduced-manning bridge watchkeeping, fatigue considerations under long passages).

The SOTIF-equivalent challenge is the perception system's ability to maintain reliable situational awareness through fog, rain, sea clutter, and at night, under conditions that exceed the characterized performance envelope of camera-radar-lidar fusion systems developed for coastal or inland waterway deployments. The combined tree for this domain adds the COLREGs compliance branch as a distinct analytical element: a vessel that correctly detects another vessel but applies an incorrect COLREGs interpretation of the encounter geometry creates a cross-domain cut set between the perception/SOTIF branch and the planning/rules-compliance branch that neither branch independently identifies.

E. Collaborative Robotics in Unstructured Environments

Current collaborative robot safety standards (ISO 10218, ISO/TS 15066) were developed for structured manufacturing environments with defined workspaces, predictable human behavior, and speed-and-force-limited operation as the primary risk control. The next generation of collaborative robotics moves into unstructured environments: construction sites, agricultural settings, healthcare, and disaster response, where the workspace is dynamic, human behavior is unpredictable, and operating at full capability is operationally necessary [103]. The PEI interfaces are Robot-to-Human-Worker (proximity, intent prediction, shared workspace negotiation, physical contact force distribution), Robot-to-UnstructuredEnvironment (terrain variability, obstacle unpredictability, weather effects on perception), and Robot-to-Task (force requirements versus safety constraints, tool interaction under material variability).

ISO 10218 and ISO/TS 15066 provide the safety branch methods for the combined tree, but neither standard covers the SOTIF-equivalent perception insufficiency branch for unstructured environments (what happens when the robot misclassifies a worker's body part as an obstacle or vice versa), nor the cybersecurity branch for compromised collaborative robot behavior near human workers. The combined tree enables cross-domain analysis of cut sets spanning these three branches, which are the combinations that result in the serious injuries documented in collaborative robot incident databases.

F. AI-Enabled Defense Systems

Autonomous and AI-enabled defense systems operate in adversarial environments by definition, with extreme consequences for misclassification, engagement decision errors, and human-AI teaming failures at decision speeds that exceed human cognition [104]. The PEI includes System-to-Battlespace (target identification under contested, degraded, and operationally stressful conditions; civilian presence uncertainty; rules-of-engagement boundary characterization), AI-to-Operator (trust calibration under time pressure, decision authority allocation, cognitive handoff at tempo

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

exceeding human response), and System-to-Adversary (deception, electronic warfare, adversarial AI inputs designed to exploit perception boundaries).

The AI-Enabled Defense domain introduces a combined tree branch with no civilian equivalent: International Humanitarian Law (IHL) compliance, specifically the principles of proportionality and distinction, becomes an engineering constraint that must be represented as a branch in the tree alongside safety and cybersecurity. A ULG for a defensive system must encompass not only prevention of unintended physical harm but also compliance with the proportionality and distinction requirements of the laws of armed conflict. The crossdomain cut set between technical capability limitations (AI perception boundary), human-AI authority allocation (operator decision under time pressure), and IHL compliance (distinction between combatants and civilians in contested environments) is the most critical analytical output the combined tree produces for this domain and one that no existing single-domain analysis in defense systems engineering currently generates.

G. Common Pattern Across Domains

Each of the six domains exhibits the same structural property that motivates the SUE framework's design: the product-environment interface is the primary locus of uncertainty; that interface is shared across multiple regulatory domains that each own only a slice of it; and the most dangerous failure combinations are cross-domain cut sets that no single domain's analysis can identify. In every domain, the framework's core constructs apply without modification. Only the instantiation changes: the interface points at which the PEI analysis is applied, the analysis methods that populate each branch of the combined tree, the regulatory standards that govern each domain's child goals, and the operational architecture through which the SMO is executed. The automotive AV application demonstrated in Sections VII through XII is one instantiation of a general method, and Table XX maps the dimensions of that instantiation to five additional domains where the same method applies directly.

XV. CONCLUSION

The defining characteristic of novel socio-technical systems is that their most consequential failures do not originate inside the product. They originate at the boundary between what the product was designed to do and what the operational environment actually presents. The Takata airbag propellant degraded in an environment its specification did not adequately characterise. The Uber ATG perception system encountered an environment whose visual complexity exceeded its training distribution. The GM ignition switch defect persisted in an organizational environment that did not translate engineering evidence into corrective action. In each case, the failure lived at an interface that no single domain's analysis fully owned. This paper has proposed a framework for engineering that interfaces systematically.

➢ *Seven Contributions*

The Systemic Uncertainty Engineering (SUE) framework makes seven specific contributions to the safety engineering of novel socio-technical systems.

- Four-quadrant uncertainty model. Decomposing uncertainty along two independent axes, epistemic versus aleatoric and endogenous versus exogenous, provides the measurement structure that distinguishes reducible from irreducible uncertainty and product-side from environment-side uncertainty. This decomposition determines which interventions address which uncertainty class and prevents the conflation of fundamentally different risk types under a single scalar [105], [106].
- The Uncertainty Diamond. A dual-V process model that positions product development activities on the left arm and environment characterization activities on the right, converging at the Unified Loss Goal focal point, with horizontal connections encoding the β interaction terms that capture crossaxis amplification. The Diamond provides the analytical activities of the framework with temporal ordering and structural relationships to existing Vmodel development processes [12], [13].
- System-Environment Interface (SEI) / ProductEnvironment Interface (PEI) analysis. The foundational first step: systematic mapping of every assumption the product makes about its operational environment, with quantified gap measurement (U_{en}, U_{ex}), management strategies, and test obligations derived before domain decomposition. The PEI is the instrument that fills the gap none of the existing automotive safety standards addresses: a structured, auditable characterization of the product-environment boundary as a first-class engineering artifact, analogous to the hardwaresoftware interface specification of ISO 26262 Part 6 but applied at the product-environment boundary.
- Parametric Loss Analysis (PLA) and Unified Threat and Hazard Analysis (UTHA). A guideword-based risk assessment method that generates loss scenarios across all domains simultaneously from a single pass over the PEI interface parameters, before domain decomposition assigns each scenario to a single standard. The automotive instantiation, UTHA, subsumes HARA, TARA, and SOTIF scenario analysis, preserving each standard's artifacts as children of the UTHA output rather than replacing them [87].
- Unified Loss Goal (ULG) / Unified Vehicle Goal (UVG), Safe Maneuver Objective (SMO), and combined domain trees. The goal architecture that parents domain-specific safety goals, cybersecurity goals, and SOTIF criteria under a single loss prevention statement, specifies the unified operational safe state through the SMO, and enables crossdomain cut set analysis by placing all four domain analysis trees under a single OR gate. The combined tree is the only instrument in the automotive safety engineering tool set that, structurally, produces cross-domain minimal cut sets: failure combinations spanning two or more domains and therefore invisible to single-domain analysis methods.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

- SUE Risk Cube: 80-cell tensor metric with economic translation. A three-axis tensor over five System Layers, four Safety Domains, and four Uncertainty Types producing 80 cells, each assigned a SUE Level from 1 (Critical) to 4 (Low) by combined-weight threshold. The tensor provides simultaneous cross-domain quantification of residual uncertainty, face-projection views for each stakeholder role, and scalar summaries including Systemic Risk Density, Expected System Loss (ESL), and Return on Safety Investment (ROSI). The economic translation converts systemic uncertainty from a compliance metric into a resource-allocation instrument [30], [31].
- Automotive instantiation and demonstrated generalisability. A complete automotive AV instantiation through PEI (six interface points, Table XIII), UTHA (six scenarios, Table XIV), UVG (Table XV), combined domain trees, SUE Risk Cube, and six planned interventions ($1.70M, Table XVII). Retrospective validation against six documented failures confirms the framework's directional accuracy. The domain applicability analysis in Section XIV maps the automotive instantiation to five additional domains, each of which satisfies the structural conditions required by the framework.

➢ *The Six Sigma Parallel*

The Six Sigma analogy that has run through this paper is worth stating precisely at its close. Six Sigma did not discover that manufacturing processes produce defects; engineers had known this for generations. Its contribution was to convert defect management from an art practiced by quality specialists into a quantitative discipline practiced by engineers across the organization: defects per million opportunities could be measured, tracked, decomposed by source, and reduced through targeted investment with measurable return. Safety leadership changed when quality became quantitative, because quantitative metrics give management the instrument to allocate resources, compare initiatives, and hold programs accountable to outcomes rather than to activity completion [14].

The SUE framework proposes the same move for systemic uncertainty. Safety engineers have long understood that complex systems fail at interfaces, that organizational decisions amplify technical risks, and that crossdomain failure combinations are the most dangerous failure modes. The framework converts these qualitative insights into measurable quantities: $U(v_i)$ at each node, β at each cross-domain interaction, SRD at each cell of the tensor, ΔESL, and ROSI for each intervention. These quantities can be tracked across program phases, compared across programs, and used to allocate the safety investment portfolio to the interventions that produce the greatest uncertainty reduction per unit cost.

The shift the framework enables is from compliance to control: from demonstrating that required activities have been completed to measuring the amount of systemic uncertainty that remains after they have been completed, and directing the remaining investment to reduce it further. Six Sigma transformed manufacturing quality into a quantitative discipline. The Uncertainty Diamond proposes achieving the same for systemic risk at the product-environment boundary: quantitative control of socio-technical uncertainty in domains where traditional statistical risk management fails.

FUTURE WORK

The framework, as presented in this paper, is complete in its theoretical structure and validated against six documented failure cases. Five research directions remain open. Each is defined by a specific gap between what the current paper provides and what full operational deployment of the framework requires.

Empirical calibration across multiple programs. The weight assignments in the SUE Risk Cube, the layer- and domain-specific U measurement instruments, and the combined-weight thresholds that define the SUE Level boundaries (Section V-D) have been validated against a single automotive program instantiation and three retrospective failure analyses. The case study retrospective confirms that the framework assigns the highest SRD to the nodes and interfaces where documented failures occurred, providing directional validation. What is missing is forward-looking, multi-program calibration: the collection of $U(v_i)$, $P(v_i)$, and $W(v_i)$ values at program milestones across multiple AV development programs at different organizations and maturity levels, followed by correlation of those values with program outcomes. This calibration effort would answer the question that the current paper cannot: not only does high SRD predict where failures occur, but does a specific SRD threshold reliably distinguish programs that experienced safety incidents from those that did not? Establishing that correlation across a statistically meaningful program population would transform the framework's current role as a structured engineering estimate into an empirically grounded prognostic instrument.

Integration with MBSE and ALM platforms. The PEI, UTHA, and UVG constructs are defined as document-based artifacts in this paper. Their full analytical value is realized only when they are embedded in the model-based and requirements-management toolchains that govern modern automotive development: ModelBased Systems Engineering (MBSE) environments such as Cameo Systems Modeler, requirements management platforms such as Polarion and JAMA, and functional safety toolchains integrated with those platforms. A PEI interface point in a spreadsheet can be referenced by a UTHA analysis but cannot be automatically propagated to affected safety goals, test cases, and verification activities when the product assumption changes. An MBSE integration would make each PEI interface point a model element with live traceability: a change to the product assumption at PEI-001 would automatically propagate to the UTHA-001 scenario, the UVG-001 loss prevention statement, the SG-001 safety goal, and the test obligation for the low-light occlusion test suite. Realising this integration requires defining the SUE constructs as stereotypes within a SysML or UML profile, implementing the UTHA guide-word application as a plugin within the target MBSE environment,

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

and validating the end-to-end traceability against an automotive functional safety case.

Runtime SMO evaluation and continuous SRD monitoring. The framework, as presented, computes SRD at discrete program milestones. The Uncertainty Diamond's architecture explicitly provides for a continuous feedback loop in which field operational data at L8 on the right arm updates the environment characterization at L1 (Section VI-D), but this loop is not implemented in the current paper. Closing it at scale requires three components that each present independent engineering challenges. First, a continuous U^{ex} update mechanism: a statistical pipeline that ingests fleet operational telemetry, detects shifts in the environmental distribution at each PEI interface relative to the characterized ODD, and updates the U^{ex} estimate at the corresponding PEI row. Second, a contextual SMO evaluator: a real-time module that maintains a pre-computed SMO conditioned on the vehicle's current environmental state (weather, traffic density, map quality, driver readiness) and updates that SMO as conditions change, ensuring that when a UVG violation trigger arrives, the system does not need to plan a safe state under degraded conditions. Third, a fleet-level SRD anomaly detector: a monitoring system that aggregates updated U^{ex} estimates across the fleet, identifies PEI interface points whose updated SRD has crossed a threshold requiring re-analysis, and initiates the corresponding UTHA and UVG review. This three-component architecture defines a concrete engineering specification for what continuous SRD monitoring requires; implementing it at automotive scale is a significant systems engineering undertaking that this paper scopes but does not execute.

Pilot applications to eVTOL certification and autonomous surgical robotics. Section XIV established that eVTOL and surgical robotics satisfy the structural conditions for SUE framework applicability and described the domain-specific PEI interface points, combined tree branches, and regulatory contexts for each. What is missing is a forward-looking program application: a full PEI analysis, UTHA, and UVG derivation conducted for an actual eVTOL or surgical robot development program, producing artifacts that can be submitted to the relevant regulatory authority (EASA, FAA, or FDA) and evaluated for acceptability within the existing certification framework. Such a pilot would answer a question that the automotive case study cannot: does the framework's cross-domain cut-set analysis identify failure modes that existing domain-specific certification methods would not? The eVTOL domain is particularly time-sensitive because EASA SC-VTOL and FAA AC 21-17G are both under active development for aircraft types without extensive operational history; a SUE-based PEI analysis submitted during the certification basis development phase could influence the interface analysis requirements embedded in those standards before they are finalised [99].

Formal comparison with GSN and CAE safety case structures. The ULG/UVG architecture of Section IX provides a traceability structure above the domainspecific goal hierarchy. Goal Structuring Notation (GSN) and Claims, Arguments, and Evidence (CAE) are the two dominant safety case argumentation frameworks in automotive and defense safety engineering [107], [108]. Both provide structured argument templates for demonstrating that a system is acceptably safe; neither provides a quantitative metric for the residual uncertainty that the safety case leaves unaddressed. The relationship between the UVG hierarchy and a GSN or CAE safety case is an open question: can a UVG-based traceability chain be represented as a GSN goal decomposition without loss of the cross-domain visibility that the UVG provides? Does the combined tree's OR-gate structure correspond to a defensible GSN argument pattern, or does it require an extension to GSN that does not yet exist? Answering these questions formally would determine whether the framework's output is directly integrable into existing safety case practice or requires a new safety case pattern to express the cross-domain analytical layer that the ULG provides above standard domain goals.

REFERENCES

[1]. N. G. Leveson, *Engineering a Safer World: Systems Thinking Applied to Safety*. Cambridge, MA: MIT Press, 2011.

[2]. C. Perrow, *Normal Accidents: Living with High-Risk Technologies*, 2nd ed. Princeton, NJ: Princeton University Press, 1999.

[3]. V. Venkatasubramanian and Z. Zhang, "TeCSMART: A hierarchical framework for modeling and analyzing systemic risk in sociotechnical systems," *AIChE Journal*, vol. 62, no. 9, pp. 3065–3084, 2016.

[4]. V. Venkatasubramanian, "Systemic failures: Challenges and opportunities in risk management in complex systems," *AIChE Journal*, vol. 57, no. 1, pp. 2–9, 2010.

[5]. T. Pawlicki, A. Samost, D. W. Brown, R. P. Manger, G. Kim, and N. G. Leveson, "Application of systems and control theorybased hazard analysis to radiation oncology," *Medical Physics*, vol. 43, no. 3, pp. 1514–1530, 2016.

[6]. J. Betz, T. Betz, F. Fent, M. Geisslinger, A. Heilmeier, L. Hermansdorfer, T. Herrmann, S. Huch, P. Karle, M. Lienkamp, B. Lohmann, F. Nobis, L. Ögretmen, M. Rowold, F. Sauerbeck, T. Stahl, R. Trauth, F. Werner, and A. Wischnewski, "TUM autonomous motorsport: An autonomous racing software for the Indy Autonomous Challenge," *Journal of Field Robotics*, vol. 40, no. 4, pp. 783–809, 2023.

[7]. A. Bansal, H. Kim, S. Yu, B. Li, N. Hovakimyan, M. Caccamo, and L. Sha, "Perception simplex: Verifiable collision avoidance in autonomous vehicles amidst obstacle detection faults," *Software Testing, Verification and Reliability*, vol. 34, no. 6, 2024.

[8]. C. Ryan, F. Murphy, and M. Mullins, "Semiautonomous vehicle risk analysis: A telematics-based anomaly detection approach," *Risk Analysis*, vol. 39, no. 5, pp. 1125–1140, 2018.

[9]. C. Deng, Y. Li, Q. Liu, X. Zheng, and K. Sun, "Quantitative risk assessment for autonomous vehicles: Integrating functional resonance analysis method and Bayesian network," *Quality and*

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

Reliability Engineering International, vol. 41, no. 3, pp. 970– 991, 2024.

[10]. National Transportation Safety Board, "Collision between vehicle controlled by developmental automated driving system and pedestrian, tempe, arizona, march 18, 2018," National Transportation Safety Board, Washington, DC, Tech. Rep. NTSB/HAR-19/03, 2019.

[11]. P. Koopman and M. Wagner, "Autonomous vehicle safety: An interdisciplinary challenge," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 90–96, 2017.

[12]. International Organization for Standardization, "Road vehicles — safety of the intended functionality," ISO, Geneva, Switzerland, International Standard ISO 21448:2022, 2022.

[13]. "Road vehicles — functional safety," ISO, Geneva, Switzerland, International Standard ISO 26262:2018, 2018.

[14]. M. J. Harry and R. Schroeder, *Six Sigma: The Breakthrough Management Strategy Revolutionizing the World's Top Corporations*. New York, NY: Doubleday, 2000.

[15]. W. E. Vesely, F. F. Goldberg, N. H. Roberts, and D. F. Haasl, "Fault tree handbook," U.S. Nuclear Regulatory Commission, Washington, DC, Tech. Rep. NUREG-0492, 1981.

[16]. J. Famfulik, M. Richtar, R. Rehak, J. Smiraus, P. Dresler, M. Fusek, and J. Mikova, "Application of hardware reliability calculation procedures according to ISO 26262 standard," *Quality and Reliability Engineering International*, vol. 36, no. 6, pp. 1822–1836, 2020.

[17]. International Organization for Standardization and SAE International, "Road vehicles — cybersecurity engineering," ISO, Geneva, Switzerland, International Standard ISO/SAE 21434:2021, 2021.

[18]. J. Dobaj, G. Macher, D. Ekert, A. Riel, and R. Messnarz, "Towards a security-driven automotive development lifecycle," *Journal of Software: Evolution and Process*, vol. 35, no. 8, 2021.

[19]. J. Yu, F. Luo, and S. Abdelwahed, "A systematic approach for cybersecurity design of in-vehicle network systems with tradeoff considerations," *Security and Communication Networks*, vol. 2020, no. 1, 2020.

[20]. A. Yousefi, M. R. Hernandez, and V. L. P. na, "Systemic accident analysis models: A comparison study between AcciMap, FRAM, and STAMP," *Process Safety Progress*, vol. 38, no. 2, 2018.

[21]. D. C. Montgomery and W. H. Woodall, "An overview of Six Sigma," *International Statistical Review*, vol. 76, no. 3, pp. 329– 346, 2008.

[22]. M. S. Shaikh and B. Moiz, "Analytical performance evaluation of a high-volume hematology laboratory utilizing sigma metrics as standard of excellence," *International Journal of Laboratory Hematology*, vol. 38, no. 2, pp. 193–197, 2016.

[23]. J. K. Visich, A. M. Wicks, and F. Zalila, "Practitioner perceptions of the A3 method for process improvement in health care," *Decision Sciences Journal of Innovative Education*, vol. 8, no. 1, pp. 191–213, 2010.

[24]. E. V. Gijo, J. Scaria, and J. Antony, "Application of Six Sigma methodology to reduce defects of a grinding process," *Quality and Reliability Engineering International*, vol. 27, no. 8, pp. 1221–1234, 2011.

[25]. B. W. Oppenheim, E. M. Murman, and D. A. Secor, "Lean enablers for systems engineering," *Systems Engineering*, vol. 14, no. 1, pp. 29–55, 2011.

[26]. P. Baraldi and E. Zio, "A combined Monte Carlo and possibilistic approach to uncertainty propagation in event tree analysis," *Risk Analysis*, vol. 28, no. 5, pp. 1309–1326, 2008.

[27]. P. Hester and T. Dohi, "Epistemic uncertainty analysis: An approach using expert judgment and evidential credibility," *Journal of Quality and Reliability Engineering*, vol. 2012, 2012.

[28]. C. Lijie, L. Zhenzhou, L. Guijie, and Y. N. Sotskov, "Reliability analysis in presence of random variables and fuzzy variables," *Journal of Applied Mathematics*, vol. 2015, 2015.

[29]. C. Ponsard, J. Grandclaudon, and P. Massonet, "A goal-driven approach for the joint deployment of safety and security standards for operators of essential services," *Journal of Software: Evolution and Process*, vol. 33, no. 9, 2021.

[30]. Z. A. Collier, B. Briglia, T. Finkelston, M. C. Manasco, D. L. Slutzky, and J. H. Lambert, "On metrics and prioritization of investments in hardware security," *Systems Engineering*, vol. 26, no. 4, pp. 425–437, 2023.

[31]. G. L. Reniers and K. Sörensen, "An approach for optimal allocation of safety resources: Using the knapsack problem to take aggregated cost-efficient preventive measures," *Risk Analysis*, vol. 33, no. 11, pp. 2056–2067, 2013.

[32]. S. Ray, P. Das, B. K. Bhattacharyay, and J. Antony, "Measuring Six Sigma project effectiveness using fuzzy approach," *Quality and Reliability Engineering International*, vol. 29, no. 3, pp. 417–430, 2012.

[33]. R. K. Sharma and R. G. Sharma, "Integrating Six Sigma culture and TPM framework to improve manufacturing performance in SMEs," *Quality and Reliability Engineering International*, vol. 30, no. 5, pp. 745–765, 2013.

[34]. H. Yu, F. Khan, and B. Veitch, "A flexible hierarchical Bayesian modeling technique for risk analysis of major accidents," *Risk Analysis*, vol. 37, no. 9, pp. 1668–1682, 2017.

[35]. M. Kaushik and M. Kumar, "An application of fault tree analysis for computing the bounds on system failure probability through qualitative data in intuitionistic fuzzy environment," *Quality and Reliability Engineering International*, vol. 38, no. 5, pp. 2420– 2444, 2022.

[36]. A. Bouafia, M. Bougofa, W. Benhamlaoui, and M. Rouainia, "Integrating functional resonance and Bayesian networks for quantitative risk assessment: Application to Adrar's refinery pre-fractionation,"

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

Quality and Reliability Engineering International, vol. 42, no. 1, pp. 503–523, 2025.

[37]. R. Messnarz, C. Kreiner, G. Macher, and A. Walker, "Extending Automotive SPICE 3.0 for the use in ADAS and future selfdriving service architectures," *Journal of Software: Evolution and Process*, vol. 30, no. 5, 2018.

[38]. A. Plioutsias, N. Karanikas, and M. M. Chatzimihailidou, "Hazard analysis and safety requirements for small drone operations: To what extent do popular drones embed safety?" *Risk Analysis*, vol. 38, no. 3, pp. 562–584, 2017.

[39]. M. M. Chatzimichailidou, J. Ward, T. Horberry, and J. P. Clarkson, "A comparison of the bow-tie and STAMP approaches to reduce the risk of surgical instrument retention," *Risk Analysis*, vol. 38, no. 5, pp. 978–990, 2017.

[40]. A. Mashkoor, A. Egyed, R. Wille, and S. Stock, "Modeldriven engineering of safety and security software systems: A systematic mapping study and future research directions," *Journal of Software: Evolution and Process*, vol. 35, no. 7, 2022.

[41]. Y. Y. Haimes, "Systems-based guiding principles for risk modeling, planning, assessment, management, and communication," *Risk Analysis*, vol. 32, no. 9, pp. 1451–1467, 2012.

[42]. J. Pence and Z. Mohaghegh, "A discourse on the incorporation of organizational factors into probabilistic risk assessment: Key questions and categorical review," *Risk Analysis*, vol. 40, no. 6, pp. 1183–1211, 2020.

[43]. D. A. Broniatowski and J. Moses, "Measuring flexibility, descriptive complexity, and rework potential in generic system architectures," *Systems Engineering*, vol. 19, no. 3, pp. 207– 221, 2016.

[44]. Automotive SIG, "Automotive SPICE process assessment / reference model," Automotive SIG / VDA QMC, Tech. Rep., 2017.

[45]. M. Ashrafi, "Forward and backward risk assessment throughout a system life cycle using dynamic Bayesian networks: A case in a petroleum refinery," *Quality and Reliability Engineering International*, vol. 37, no. 1, pp. 309–334, 2020.

[46]. E. Borgonovo, "Epistemic uncertainty in the ranking and categorization of probabilistic safety assessment model elements: Issues and findings," *Risk Analysis*, vol. 28, no. 4, pp. 983–1001, 2008.

[47]. C. Xie, G. Li, and E. R. Vaidogas, "Quantification of margins and uncertainties approach for structure analysis based on evidence theory," *Mathematical Problems in Engineering*, vol. 2016, 2016.

[48]. M. Vogel, P. Knapik, M. Cohrs, B. Szyperrek, W. Pueschel, H. Etzel, D. Fiebig, A. Rausch, and M. Kuhrmann, "Metrics in automotive software development: A systematic literature review," *Journal of Software: Evolution and Process*, vol. 33, no. 2, 2020.

[49]. G. Carrozza, R. Pietrantuono, and S. Russo, "Defect analysis in mission-critical software systems: A detailed investigation," *Journal of Software: Evolution and Process*, vol. 27, no. 1, pp. 22–49, 2014.

[50]. M. Azzeh, Y. Alqasrawi, and Y. Elsheikh, "A soft computing approach for software defect density prediction," *Journal of Software: Evolution and Process*, vol. 36, no. 4, 2023.

[51]. J. E. Hale and D. P. Hale, "Evaluating testing effectiveness during software evolution: A time-series cross-section approach," *Journal of Software: Evolution and Process*, vol. 24, no. 1, pp. 35–49, 2011.

[52]. D. R. Karanki, H. S. Kushwaha, A. K. Verma, and S. Ajit, "Uncertainty analysis based on probability bounds (P-box) approach in probabilistic safety assessment," *Risk Analysis*, vol. 29, no. 5, pp. 662–675, 2009.

[53]. P. Uday and K. Marais, "Designing resilient systems-of-systems: A survey of metrics, methods, and challenges," *Systems Engineering*, vol. 18, no. 5, pp. 491–510, 2015.

[54]. M. C. Dietze, "Prediction in ecology: A first-principles framework," *Ecological Applications*, vol. 27, no. 7, pp. 2048–2060, 2017.

[55]. P. Winter, J. Downer, J. Wilson, D. B. Abeywickrama, S. Lee, S. Hauert, and S. Windsor, "Applying the "SOTEC" framework of sociotechnical risk analysis to the development of an autonomous robot swarm for a public cloakroom," *Risk Analysis*, vol. 45, no. 4, pp. 878–895, 2024.

[56]. W. N. Caballero, D. R. Insua, and R. Naveiro, "Some statistical challenges in automated driving systems," *Applied Stochastic Models in Business and Industry*, vol. 39, no. 5, pp. 629–652, 2023.

[57]. A. D. Adesiji, S. E. Ibitoye, R. M. Mahamood, O. A. Olayemi, P. O. Omoniyi, T. Jen, and E. T. Akinlabi, "Safety considerations in deployment of robotic systems: A systematic review," *Journal of Field Robotics*, vol. 43, no. 1, pp. 5–33, 2025.

[58]. A. U. Kulkarni, A. Salado, and C. Wernz, "Optimal verification strategies in multi-firm projects," *Systems Engineering*, vol. 25, no. 3, pp. 254–270, 2022.

[59]. A. U. Kulkarni, A. Salado, P. Xu, and C. Wernz, "An evaluation of the optimality of frequent verification for vertically integrated systems," *Systems Engineering*, vol. 24, no. 1, pp. 17–33, 2020.

[60]. N. K. Singh, M. Lawford, T. S. E. Maibaum, and A. Wassyng, "A formal approach to rigorous development of critical systems," *Journal of Software: Evolution and Process*, vol. 33, no. 4, 2021.

[61]. A. Ruiz-Tagle, E. L. Droguett, and K. M. Groth, "Exploiting the capabilities of Bayesian networks for engineering risk assessment: Causal reasoning through interventions," *Risk Analysis*, vol. 42, no. 6, pp. 1306–1324, 2021.

[62]. C. D. Persis, J. L. Bosque, I. Huertas, M. R. Sillero-Denamiel, and S. P. Wilson, "Quantitative system risk assessment from incomplete data with belief networks and pairwise comparison elicitation," *Risk Analysis*, vol. 45, no. 11, pp. 4014–4038, 2025.

[63]. S. N. Hall, M. A. Gallagher, and D. S. Fenn, "Risk framework for an organizational system with major components," *Risk Analysis*, vol. 40, no. 12, pp. 2509–2523, 2020.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

[64]. A. Sarwar, F. Khan, M. Abimbola, and L. James, "Resilience analysis of a remote offshore oil and gas facility for a potential hydrocarbon release," *Risk Analysis*, vol. 38, no. 8, pp. 1601– 1617, 2018.

[65]. C. Bao, M. Cai, J. Li, Q. Zheng, D. Wu, and Q. Meng, "Risk aggregation considering probabilistic and consequential interactions: A general formulation with computational cost handling," *Risk Analysis*, vol. 44, no. 6, pp. 1440–1459, 2023.

[66]. V. Salehi, B. Veitch, and D. Smith, "Modeling complex sociotechnical systems using the FRAM: A literature review," *Human Factors and Ergonomics in Manufacturing & Service Industries*, vol. 31, no. 1, pp. 118–142, 2020.

[67]. O. Štumbauer and A. Lališ, "Progressing the aerospace performance factor toward nonlinear interactions," *Risk Analysis*, vol. 42, no. 10, pp. 2243–2252, 2022.

[68]. S. F. D. Team, "Quantitative framework for systemic risk reduction, version 2," Technical Framework Document, 2024, unpublished; provided as project source material. Section III.D establishes the calibration requirement: explicit, consistent, auditable.

[69]. M. J. Barons, S. Mascaro, and A. M. Hanea, "Balancing the elicitation burden and the richness of expert input when quantifying discrete Bayesian networks," *Risk Analysis*, vol. 42, no. 6, pp. 1196–1234, 2021.

[70]. P. Baybutt, "The validity of engineering judgment and expert opinion in hazard and risk analysis: The influence of cognitive biases," *Process Safety Progress*, vol. 37, no. 2, pp. 205–210, 2017.

[71]. C. Wiecher, C. Mandel, M. Günther, J. Fischbach, J. Greenyer, M. Greinert, C. Wolff, R. Dumitrescu, D. Mendez, and A. Albers, "Model-based analysis and specification of functional requirements and tests for complex automotive systems," *Systems Engineering*, vol. 27, no. 4, pp. 728–744, 2024.

[72]. C. Macrae, "Managing risk and resilience in autonomous and intelligent systems: Exploring safety in the development, deployment, and use of artificial intelligence in healthcare," *Risk Analysis*, vol. 45, no. 4, pp. 910–927, 2024.

[73]. X. Wang, C. Li, and L. Zhao, "Requirement specification extraction and analysis based on propositional projection temporal logic," *Journal of Software: Evolution and Process*, vol. 36, no. 4, 2023.

[74]. F. Talha, T. Tahir, and T. Nadeem, "A semiautomated approach for detecting ambiguities in software requirements using SpanBERT and named entity recognition," *Journal of Software: Evolution and Process*, vol. 37, no. 8, 2025.

[75]. R. Dreves, F. Hällmayer, L. Haunert, B. Sechser, and A. Rieß, "A method to realize traceability in development processes," *Journal of Software: Evolution and Process*, vol. 28, no. 11, pp. 1011–1019, 2016.

[76]. R. Messnarz, T. Wegner, D. Ekert, B. Steger, R. Mayer, R. Dreves, B. Sechser, C. Schlager, and C. Karner, "Process improvement guidance for successful automotive SPI implementation," *Journal of Software: Evolution and Process*, vol. 35, no. 8, 2021.

[77]. T. Varkoi, T. Mäkinen, F. Cameron, and R. Nevalainen, "Validating effectiveness of safety requirements compliance evaluation in process assessments," *Journal of Software: Evolution and Process*, vol. 32, no. 3, 2019.

[78]. V. Anes, E. Henriques, M. Freitas, and L. Reis, "A new risk prioritization model for failure mode and effects analysis," *Quality and Reliability Engineering International*, vol. 34, no. 4, pp. 516–528, 2018.

[79]. J. R. Bradley and H. H. Guerrero, "An alternative FMEA method for simple and accurate ranking of failure modes," *Decision Sciences*, vol. 42, no. 3, pp. 743–771, 2011.

[80]. Z. Wang, Y. Ran, H. Yu, C. Jin, and G. Zhang, "Failure mode and effects analysis using function-motion-action decomposition method and integrated risk priority number for mechatronic products," *Quality and Reliability Engineering International*, vol. 37, no. 6, pp. 2875–2899, 2021.

[81]. Y. Li and L. Zhu, "Risk analysis of human error in interaction design by using a hybrid approach based on FMEA, SHERPA, and fuzzy TOPSIS," *Quality and Reliability Engineering International*, vol. 36, no. 5, pp. 1657–1677, 2020.

[82]. J. A. Moseman, "Retrospective on the risk matrix, part II: Mathematics," *Process Safety Progress*, vol. 43, no. 3, pp. 455– 468, 2024.

[83]. P. Younse, J. Cameron, and T. H. Bradley, "Comparative analysis of model-based and traditional systems engineering approaches for simulating a robotic space system architecture through automatic knowledge processing," *Systems Engineering*, vol. 25, no. 4, pp. 360–386, 2022.

[84]. M. Hillenbrand, M. Heinz, J. Matheis, and K. D. Müller-Glaser, "Development of electric/electronic architectures for safetyrelated vehicle functions," *Software: Practice and Experience*, vol. 42, no. 7, pp. 817–851, 2012.

[85]. K. Seo, K. Park, and Z. Gao, "Interface data modeling to detect and diagnose intersystem faults for designing and integrating system of systems," *Complexity*, vol. 2018, p. 7081501, 2018.

[86]. SAE International, "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," SAE International, Warrendale, PA, SAE Standard SAE J3016:2021, 2021.

[87]. T. Kletz, *HAZOP and HAZAN: Identifying and Assessing Process Industry Hazards*, 4th ed. Rugby, UK: Institution of Chemical Engineers, 1999.

[88]. G. C. Waycaster, T. Matsumura, V. Bilotkach, R. T. Haftka, and N. H. Kim, "Review of regulatory emphasis on transportation safety in the United States, 2002–2009: Public versus private modes," *Risk Analysis*, vol. 38, no. 5, pp. 1085–1101, 2017.

[89]. National Highway Traffic Safety Administration, "Takata airbag inflator recall: Coordinated remedy programme," NHTSA, Washington, DC, Tech. Rep., 2016, covers over 100 million inflators under consent

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

order; root cause: ammonium nitrate propellant degradation.
[90]. "Technical assessment of Toyota electronic throttle control (ETC) systems," NHTSA and NASA, Washington, DC, Tech. Rep., 2011.
[91]. M. Barr, "Software analysis finds that unintended acceleration fault could cause a crash in Toyota vehicles," Expert Witness Report, Bookout v. Toyota Motor Corp., 2013, expert analysis of ETCS software identifying task-scheduling vulnerabilities and insufficient exception handling.
[92]. A. R. Valukas, "Report to board of directors of General Motors Company regarding ignition switch recalls," Jenner and Block LLP, commissioned by GM, 2014, independent investigation; documents decade-long organisational failure to act on known defect evidence; linked to at least 124 fatalities.
[93]. National Highway Traffic Safety Administration, "Investigation PE22-002: Tesla motors, inc. autopilot advanced driver assistance system," NHTSA, Washington, DC, Tech. Rep., 2022, multi-incident investigation covering collisions with emergency vehicles and other stationary objects.
[94]. California Department of Motor Vehicles, "Order of suspension of autonomous vehicle deployment permit: Cruise LLC," California DMV, Sacramento, CA, 2023, suspension citing material non-disclosure of post-collision dragging incident; October 2023.
[95]. Z. H. Qureshi and A. Campbell, "Systemic safety and accident modelling of complex socio-technical systems," in *INCOSE International Symposium*, vol. 19, no. 1, 2009, pp. 21–35.
[96]. S. Tan, D. Weinert, P. Joseph, and K. A. Moinuddin, "Incorporation of technical, human and organizational risks in a dynamic probabilistic fire risk model for high-rise residential buildings," *Fire and Materials*, vol. 45, no. 6, pp. 779–810, 2020.
[97]. J. P. Mendes, "Model-based risk analysis for system design," *Systems Engineering*, vol. 27, no. 1, pp. 5–20, 2023.
[98]. A. Rauzy, "New algorithms for fault trees analysis," *Reliability Engineering & System Safety*, vol. 40, no. 3, pp. 203–211, 1993.
[99]. European Union Aviation Safety Agency, "Special condition for small-category VTOL aircraft," EASA, Cologne, Germany, Tech. Rep. SC-VTOL-01, 2022.
[100]. A. J. Hung, A. Goh *et al.*, "Artificial intelligence in urologic robotic surgery," *Urology Practice*, vol. 8, no. 1, pp. 10–18, 2021.
[101]. D. T. Ingersoll, Z. J. Houghton, R. Bromm, and C. Desportes, "Nuscale small modular reactor for co-generation of electricity and water," *Desalination*, vol. 340, pp. 84–93, 2014.
[102]. T. Porathe, "Remote control of unmanned vessels in the MUNIN project: A challenge for human factors research," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 58, no. 1, pp. 1467–1471, 2014.
[103]. V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on humanrobot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248–266, 2018.
[104]. M. L. Cummings, "Artificial intelligence and the future of warfare," *International Security*, vol. 40, no. 1, pp. 31–47, 2017.
[105]. W. E. Walker, P. Harremoës, J. Rotmans, J. P. van der Sluijs, M. B. A. van Asselt, P. Janssen, and M. P. K. von Krauss, "Defining uncertainty: A conceptual basis for uncertainty management in model-based decision support," *Integrated Assessment*, vol. 4, no. 1, pp. 5–17, 2003.
[106]. T. Aven, "On the need for restricting the probabilistic analysis in risk assessments to variability," *Risk Analysis*, vol. 30, no. 3, pp. 354–360, 2010.
[107]. T. P. Kelly, "Arguing safety — a systematic approach to managing safety cases," in *Proceedings of the 17th International System Safety Conference*, Orlando, FL, 1999, foundational paper introducing Goal Structuring Notation (GSN).
[108]. P. Bishop and R. Bloomfield, "A methodology for safety case development," in *Safety-Critical Systems: The Convergence of Art and Science*. London: Springer, 2000, pp. 194–203, claims, Arguments, and Evidence (CAE) safety case methodology.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

APPENDIX A
GLOSSARY OF FRAMEWORK TERMS

The following terms are defined in the order they appear in the analytical chain of the SUE framework: from the discipline name through the process model, the five analytical steps, the primary metric, the derived scalars, and the four uncertainty type components.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

APPENDIX B
SUE RISK CUBE COMPANIONTOOL: PSEUDOCODE SUMMARY

The companion implementation of the SUE framework is a browser-based interactive tool built in React withD3 visualisation. The full source file (SUE Risk Cube Interactive.jsx) contains approximately 510 lines of production code. The pseudocode below summarises the five functional modules, the core SUE Level computation, and the six intervention definitions that correspond to Table 17.

```
//         AXIS DEFINITIONS (3 axes    80 cells total)

LAYERS  = [Requirements, Design, Implementation, Process, Toolchain]
DOMAINS = [Safety, Performance, Security, Organizational]
TYPES   = [Ep-Exogenous, Ep-Endogenous, Al-Exogenous, Al-Endogenous]

//         AXIS WEIGHTS (reflect harm potential per axis value)
```

```
LAYER_W  = { Requirements: 4, Design: 4, Implementation: 3,
             Process: 2, Toolchain: 1 }
DOMAIN_W = { Safety: 4, Performance: 4, Security: 3,
             Organizational: 2 }
TYPE_W   = { Ep-Exogenous: 4, Ep-Endogenous: 3,
             Al-Exogenous: 2, Al-Endogenous: 1 }

//         SUE LEVEL BOUNDARIES

//   Combined weight = LAYER_W + DOMAIN_W + TYPE_W  (range: 4 to 12)
//   SUE-1 (Critical)  : combined >= 11
//   SUE-2 (Elevated)  : combined >= 9
//   SUE-3 (Moderate)  : combined >= 6
//   SUE-4 (Low)       : combined <  6
```

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

➢ *Listing1. Module1–Axis Definitions and Axis Weight Tables*

```
// Input : layer (l), domain (d), uncertainty type (t)
// Output: integer 1..4 representing the SUE Level for that cell

function sueLevel(l, d, t):
    combined = LAYER_W[l] + DOMAIN_W[d] + TYPE_W[t]
    if   combined >= 11  return SUE-1   // Critical
    elif combined >= 9   return SUE-2   // Elevated
    elif combined >= 6   return SUE-3   // Moderate
    else                 return SUE-4   // Low

//          OVERRIDE MECHANISM

// Each cell key has the form "Layer|Domain|Type".
// overrides[key] = null  ->  baseline sueLevel() is used.
// overrides[key] = n     ->  user-supplied value n overrides baseline.
// The What-If module (Module 5) writes override values when an
// intervention is applied.
```

➢ *Listing 2. Module2–Core SUE Level Computation (SUE Level)*

```
// TAB 1: Risk Cube
//   Renders an isometric 3-D projection of the 80-cell tensor.
//   Each cell is a coloured square; colour maps to SUE Level.
//   Axes labelled: Layer (x), Domain (y), Uncertainty Type (z).
//   User clicks a cell to inspect its Layer | Domain | Type triplet
//   and the combined weight that determines its SUE Level.

// TAB 2: Face Projections
//   Three 2-D heat-map slices of the tensor, one per axis pair:
//      Domain x Type  (collapsed over Layer)
//      Layer  x Type  (collapsed over Domain)
//      Layer  x Domain (collapsed over Type)
```

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

Term	Category	Definition
Systemic Uncertainty Engineering (SUE)	Discipline	The overall quantitative discipline proposed in this paper. SUE treats uncertainty as a measurable, propagating system property and expresses residual systemic risk as expected financial loss.
Uncertainty Diamond	Process model	A dual-V process model with product development activities on the left arm and environment characterisation activities on the right arm, converging at the Unified Loss Goal focal point. Horizontal connections encode the β interaction terms that capture cross-axis amplification.
System-Environment Interface (SEI) / Product-Environment Interface (PEI)	Step 1	The foundational first step: systematic mapping of every assumption the product makes about its operational environment, with measured uncertainty gap (U_{en}, U_{ex}), management strategy, and test obligations derived before domain decomposition. In the automotive AV application the SEI instantiates as the PEI. The SEI/PEI is an interface analysis, not a risk assessment; risk assessment happens at the UTHA step.
Parametric Loss Analysis (PLA) / Unified Threat and Hazard Analysis (UTHA)	Step 2	The risk assessment step. PLA applies guide words to SEI interface parameters and generates loss scenarios across all domains simultaneously before domain decomposition. In automotive application it instantiates as UTHA, which subsumes HARA (ISO 26262), TARA (ISO/SAE 21434), and SOTIF scenario analysis (ISO 21448). SUE Levels are assigned here.
Unified Loss Goal (ULG) / Unified Vehicle Goal (UVG)	Step 3	A domain-agnostic loss prevention statement that parents all domain-specific goals derived from a given loss scenario. Stated in terms of the loss to be prevented, not the mechanism. In automotive applications, the ULG instantiates as the UVG, which decomposes into a Safety Goal (ASIL, ISO 26262), a Cybersecurity Goal (CAL, ISO/SAE 21434), and a SOTIF Criterion (ISO 21448).
Safe Maneuver Objective (SMO)	Step 4	The operational target state the vehicle must reach when a ULG violation is detected or imminent. Domain-agnostic, continuously evaluated against current operating conditions, and conflict-resolving: it specifies the unified physical target that all domain responses must serve, resolving conflicts between domain-specific safe state definitions.
Combined Domain Tree	Step 5	A single fault and threat tree with one OR gate, whose branches use each domain's native analysis method (FTA for safety, attack trees for cybersecurity, insufficiency trees for SOTIF/AI, an organizational branch for human factors and process). The unique output is the cross-domain minimal cut set: a combination of leaf nodes from two or more branches that is invisible to any single-domain analysis.
SUE Risk Cube	Primary metric	An 80-cell tensor over three axes: five System Layers (Requirements, Design, Implementation, Process, Toolchain), Four Safety Domains (Safety, Performance, Security, Organizational, and four Uncertainty Types (Ep-Exogenous, Ep-Endogenous, Al-Exogenous, Al-Endogenous). Each cell is assigned an SUE Level by combined weight threshold. The tensor is populated by the UTHA step.
SUE Level (1–4)	Cell rating	Severity classification assigned to each SUE Risk Cube cell based on the combined weight of its System Layer, Safety Domain, and Uncertainty Type axis values. SUE-1 (Critical, combined weight ≥ 11) through SUE-4 (Low, combined weight ≤ 5).

```
//   Projection value = minimum SUE Level in the collapsed dimension.
//   Supports role-based views (Domain lead, Layer lead, System Eng).

// TAB 3: PEI Worksheet
//   Editable table pre-loaded with 5 PEI rows from the paper:
//     PEI-001 Perception <-> Scene       | U_en: Moderate | U_ex: High
//     PEI-002 Planning <-> Traffic       | U_en: Low      | U_ex: High
//     PEI-003 Localisation <-> Map       | U_en: Low      | U_ex: Moderate
//     PEI-004 V2X <-> Network            | U_en: Low      | U_ex: High
//     PEI-005 Driver <-> HMI             | U_en: Moderate | U_ex: High
//   Fields per row: ID, Interface, Product Assumption, Environment Reality, U_en, U_ex, Gap, Test Obligation, UTHA Reference.
//   "+ Add Interface Point" button appends an empty row (PEI-006...).

// TAB 4: What-If Simulator  (see Module 5)

// TAB 5: Case Study Viewer  (see Module 4)
```

Listing 3. Module 3 – Five interactive tabs and their functions

```
// Three case studies pre-loaded; user selects via button:

CASE_1: Uber ATG Tempe (2018)
  UVG: "Vehicle shall not strike VRU due to perception / planning / supervision failure"
```

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

Term	Category	Definition
Systemic Risk Density (SRD)	Scalar metric	The aggregated quantitative score derived from the SUE Risk Cube. At the cell level, $\mathrm{SRD}(i,j,k) = U(i,j,k) \times P(i,j,k) \times W(i,j,k)$. Domain-level SRD is the face-sum projection over Layer and Uncertainty Type axes. Total SRD is the sum across all 80 cells. Used as the executive dashboard metric and as the input to ESL computation.
Expected System Loss (ESL)	Economic metric	Financial translation of the domain-level SRD: $\mathrm{ESL} = \sum_d \mathrm{SRD}_d \times C_{e,d} \times E_d$, where $C_{e,d}$ is the expected cost per loss event in domain d and E_d is the operational exposure. ESL converts residual systemic uncertainty into a monetary figure comparable across interventions and programmes.
Return on Safety Investment (ROSI)	Economic metric	$\mathrm{ROSI} = (\Delta\mathrm{ESL} - I)\,/\,I$, where ΔESL is the reduction in expected system loss produced by an intervention and I is the intervention cost. ROSI enables objective prioritization of competing safety investments: within a fixed budget, interventions should be applied in descending ROSI order.
β Coefficient	Propagation term	An interaction coefficient in the nonlinear uncertainty propagation model that captures superadditive amplification when uncertainties from two different nodes co-occur. A positive β indicates that joint risk exceeds the sum of individual contributions, reflecting the non-compensation condition. Cross-axis β values (pairing an endogenous node with an exogenous node) are consistently the largest SRD contributors across documented failure cases.
Epistemic Uncertainty	Uncertainty type	Reducible uncertainty arising from incomplete knowledge, insufficient data, or unvalidated assumptions. Epistemic uncertainty can in principle, be reduced by further testing, analysis, or ODD characterization. Carries axis weight 3 (Ep-Endogenous) or weight 4 (Ep-Exogenous) in the Risk Cube.
Aleatoric Uncertainty	Uncertainty type	Irreducible uncertainty arising from inherent variability in the system, environment, or human behaviour. Aleatoric uncertainty cannot be eliminated by further analysis; it must be managed through ODD constraints, runtime monitoring, or SMO architecture. Carries axis weight 1 (Al-Endogenous) or weight 2 (Al-Exogenous) in the Risk Cube.
Endogenous Uncertainty	Uncertainty type	Uncertainty on the product side of the Uncertainty Diamond, arising from the product's design, requirements, implementation, process maturity, or toolchain. Addressed through engineering improvements and verification activities (left arm of the Diamond).
Exogenous Uncertainty	Uncertainty type	Uncertainty on the environment side of the Uncertainty Diamond, arising from the product's operational environment including physical conditions, human behavior, adversarial agents, and ODD variability. Addressed through the environment characterization, ODD management, and runtime monitoring (right arm of the Diamond).

```
  Critical cells (SUE-1):
    Implementation | Performance |
        Ep-Exogenous
      -> Perception cycling across object
          classes
    Design         | Safety      |
        Ep-Endogenous
      -> Action suppression logic prevented
          AEB
    Process        | Organizational |
        Ep-Endogenous
      -> Single-driver supervision policy
  Cross-domain cut set displayed:
    {Ep-Exogenous} AND {Ep-Endogenous} AND
        {Ep-Endogenous}

CASE_2: Tesla Autopilot Incidents
  UVG: "Vehicle shall not collide due to
      perception /
        ODD / driver engagement failure"
  Critical cells (SUE-1):
    Implementation | Performance   |
        Ep-Exogenous
    Requirements   | Safety        |
        Ep-Exogenous
    Process        | Organizational |
        Ep-Endogenous

CASE_3: Cruise SF Operations (2023)
  UVG: "Vehicle shall not cause secondary
      injury through
        post-impact manoeuvre execution"
  Critical cells (SUE-1):
    Implementation | Performance   |
        Ep-Exogenous
    Design         | Safety        |
        Ep-Endogenous
    Process        | Organizational |
        Ep-Endogenous

// Clicking a highlighted cell updates the
    Risk Cube view on Tab 1.
```

Listing 4. Module 4 – Case study viewer (three AV incidents)

```
// Each intervention defines: name, affected
    cell set, delta, cost.
// Applying an intervention raises each
    affected cell by one SUE Level
// (e.g. SUE-1 -> SUE-2), bounded at SUE-4.
// The simulator shows before / after cell
    counts for SUE-1..4.
```

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may044

```
INTERVENTIONS = [

  INT-001: Formal Requirements Review  cost: $150K
    target: ALL cells where Layer = Requirements
    affected: 4 Domains x 4 Types = 16 cells

  INT-002: Expanded Scenario Validation  cost: $500K
    target: ALL cells where Type = Ep-Exogenous
    affected: 5 Layers x 4 Domains = 20 cells
    // Highest single-intervention SRD reduction

  INT-003: Independent Verification & Validation  cost: $300K
    target: cells where Layer in {Implementation, Design}
        AND Type in {Ep-Endogenous, Ep-Exogenous}
    affected: 2 Layers x 4 Domains x 2 Types = 16 cells

  INT-004: Runtime Monitoring + SMO  cost: $400K
    target: ALL cells where Type = AI-Exogenous
    affected: 5 Layers x 4 Domains = 20 cells
    // Manages irreducible aleatoric uncertainty

  INT-005: Tool Qualification (ISO 26262 Pt 8)  cost: $100K
    target: ALL cells where Layer = Toolchain
    affected: 4 Domains x 4 Types = 16 cells
    // Lowest-cost intervention

  INT-006: Process Maturity Improvement (ASPICE)  cost: $250K
    target: ALL cells where Layer = Process
    affected: 4 Domains x 4 Types = 16 cells
    // Completes lifecycle layer coverage; reduces beta amplification
]

TOTAL BUDGET: $1.70M  (all six interventions, all planned)

// reset() restores all cells to sueLevel() baseline values.
```

➢ *Listing 5. Module 5– What-If Simulator: six Planned Interventions (Corresponds to Table 9 in the Paper, Total $1.70M)*