Da IA generativa à engenharia auditável: um framework de assurance human-in-the-loop para artefatos de engenharia no Modelo em V - case de payload inteligente em UAV

Ali Kamel Issmael Junior e José Vicente Calvano

Crivum DA88 · DOI 10.67665/crivum.donc0e5o · publicada em 12/09/2026

DA IA GENERATIVA À ENGENHARIA AUDITÁVEL: UM FRAMEWORK DE ASSURANCE HUMAN-IN-THE-LOOP PARA ARTEFATOS DE ENGENHARIA NO MODELO EM V - CASE DE PAYLOAD INTELIGENTE EM UAV Ali Kamel Issmael Junior[1]; José Vicente Calvano[2] [1] Pesquisador Independente | [2] Universidade Federal do Rio de Janeiro - COPPE/UFRJ

Resumo

A inteligência artificial (IA) generativa já é capaz de apoiar várias tarefas de engenharia. Isso cria, porém, um problema de assurance, isto é, de demonstração documentada de que um artefato pode ser aceito com confiança: versões tecnicamente relevantes podem receber contribuição de modelos probabilísticos sem que sua proveniência, materialidade, rastreabilidade, revisão e evidência de aceitação permaneçam suficientemente claras. Este case propõe um framework human-in-the-loop, no qual a decisão final continua humana, para governar Artefatos Assistidos por IA (AAIA) no Modelo em V. A materialidade da contribuição é classificada pelo AI Assistance Level (AAL, Nível de Assistência da IA), em escala ordinal A0-A4 e separada do tipo de intervenção. A Unidade Semântica Consequencial (USC) sustenta a medida prospectiva S; criticidade, força empírica de detecção, evidências e vetos regulam o gate, isto é, o ponto formal de decisão. Além da demonstração em payload de veículo aéreo não tripulado (UAV), realiza-se triangulação empírica externa com dois datasets públicos. No conjunto Zenodo (2026) de requisitos gerados por grandes modelos de linguagem (Large Language Models - LLMs), a amostra de 50 requisitos avaliada por cinco especialistas apresentou baixa concordância semântica (- de Krippendorff entre 0,150 e 0,394); a extensão textual correlacionou-se positivamente com não ambiguidade (-=0,687) e verificabilidade (-=0,635), mas negativamente com singularidade (-=-0,697), mostrando que quantidade textual não substitui atomicidade semântica. Na telemetria Microsoft (2026) de 21 programadores, 35,2% de 1.096 decisões aceitaram sugestões Copilot e 41,2% das 386 sugestões aceitas sofreram edição posterior; em modelo de Equações de Estimação Generalizadas (GEE) agrupado por usuário, maior extensão da sugestão aumentou as chances de edição posterior, enquanto maior confiança do modelo as reduziu. Esses resultados não validam integralmente o framework, mas fornecem evidência externa de viabilidade e validade discriminante para USC/S e para o custo de supervisão. Nove hipóteses causais permanecem prospectivas.

Palavras-chave: IA generativa; assurance; Engenharia de Sistemas; Modelo em V; human-in-the-loop; qualificação de ferramentas; rastreabilidade; materialidade; UAV/VANT.

1. Introdução

Ferramentas de IA generativa passaram a apoiar ideação, análise textual, geração de código, documentação e preparação de testes. Estudos empíricos com assistentes de programação mostram potencial de aceleração, mas também deixam claro que supervisão, interpretação e decisão humanas continuam necessárias (VAITHILINGAM; ZHANG; GLASSMAN, 2022; BARKE; JAMES; POLIKARPOVA, 2023). Em engenharia, portanto, uma resposta plausível da IA não é, por si só, evidência suficiente: antes de ser incorporada, deve ser possível identificar sua origem, vinculá-la aos requisitos, revisá-la, controlar sua configuração e demonstrar sua verificação e validação (V&V).

Esse problema já aparece em referenciais de governança e risco. O Artificial Intelligence Risk Management Framework (AI RMF), do National Institute of Standards and Technology (NIST), organiza a gestão de riscos nas funções GOVERN, MAP, MEASURE e MANAGE (NIST, 2023). O perfil NIST AI 600-1 adapta essa lógica à IA generativa e inclui confabulation, isto é, conteúdo apresentado de forma convincente embora falso ou incorreto, entre os riscos relevantes (AUTIO et al., 2024). As normas conjuntas da International Organization for Standardization e da International Electrotechnical Commission (ISO/IEC) 42001:2023 e 23894:2023 tratam, respectivamente, sistemas de gestão de IA e

gerenciamento de riscos; e o Regulamento (UE) 2024/1689 estabelece obrigações de risco, registro, transparência e supervisão humana para situações abrangidas (ISO/IEC, 2023a; 2023b; EUROPEAN UNION, 2024).

A literatura de assurance oferece outro conjunto de instrumentos. Assurance cases e safety cases organizam argumentos e evidências de confiança; Model-Based Systems Engineering (MBSE) integra modelos ao processo de engenharia; e Assurance of Machine Learning for use in Autonomous Systems (AMLAS) estrutura evidências para componentes de Machine Learning (ML) em sistemas autônomos (HAWKINS et al., 2021; WEI et al., 2024; AHLBRECHT; SPROCKHOFF; DURAK, 2025). Existe ainda um vizinho conceitual importante: a qualificação de ferramentas de desenvolvimento. DO-330, ISO 26262-8 e IEC 61508-3 tratam situações em que uma ferramenta pode produzir, transformar ou verificar artefatos e, assim, introduzir um erro ou deixar de detectá-lo. Esse paralelo precisa ser considerado antes de reivindicar uma lacuna.

A lacuna investigada é, portanto, específica. Este estudo não pretende substituir a governança de IA, o assurance de sistemas nem a qualificação formal de ferramentas. A proposta é complementar essas abordagens governando a versão concreta de um artefato de Engenharia de Sistemas que recebeu contribuição de GenAI e será submetida a um gate. Para cada versão, registram-se a materialidade da contribuição, o tipo de intervenção, a criticidade de uma falsa aceitação, a revisão realizada, as evidências disponíveis e a decisão final.

A aplicação utiliza um payload modular de veículo aéreo não tripulado (VANT; UAV, Unmanned Aerial Vehicle) oriundo de uma linha de pesquisa aceita no Encontro Nacional de Engenharia de Produção (ENEGEP) 2026 e no Simpósio de Engenharia de Produção (SIMPEP) 2026 (ISSMAEL JUNIOR; CALVANO, 2026a; 2026b). Esses trabalhos fornecem documentação do demonstrador, mas não são usados como autoridade para afirmar originalidade.

Para reduzir a dependência exclusiva de argumentação conceitual, o estudo acrescenta uma análise secundária de dois conjuntos públicos independentes. O primeiro reúne 900 requisitos gerados por Large Language Models (LLMs, grandes modelos de linguagem), com uma subamostra de 50 requisitos avaliada por cinco especialistas. O segundo contém a telemetria de 21 programadores usando GitHub Copilot. O objetivo não é declarar uma validação completa do framework, mas verificar três pontos mais modestos: se julgamentos semânticos podem ser operacionalizados, se quantidade ou aceitação de uma sugestão representam adequadamente sua contribuição persistente e se há custos observáveis de supervisão.

O Modelo em V é usado como mapa do processo de engenharia, e não como um procedimento imposto pela ISO/IEC/IEEE 15288. Os processos técnicos da ISO/IEC/IEEE 15288:2023, desenvolvida conjuntamente com o Institute of Electrical and Electronics Engineers (IEEE), - necessidades e requisitos, requisitos de sistema, arquitetura, design, implementação, integração, verificação e validação - são posicionados nas duas pernas do V. O Systems Engineering Handbook do International Council on Systems Engineering (INCOSE, 2023) fornece a representação Vee e a relação entre definição, realização, revisões técnicas e decision gates. Essa distinção evita atribuir à 15288 uma arquitetura de ciclo de vida que a norma não prescreve.

2. Revisão Bibliográfica e Mapeamento Comparativo

2.1 Estratégia de mapeamento O mapeamento comparativo adota uma abordagem estruturada e não exaustiva, com cobertura parcial bibliográfica e normativa disponível até 6 de setembro de 2026. A pesquisa reúne literatura acadêmica, documentos normativos e fontes institucionais, incluindo ACM Digital Library, IEEE/ACM, periódicos indexados acessíveis por SpringerLink e ScienceDirect, catálogos oficiais do NIST, ISO, IEC e RTCA, o Zenodo (2026) e repositórios institucionais. As estratégias de busca combinam os termos 'generative AI' OR 'large language model' com 'systems engineering', requirements, verification, validation, assurance case, traceability, 'tool qualification', 'AI-assisted programming', Copilot e provenance. Referenciais normativos e fundacionais não recebem corte temporal; para aplicações de GenAI/LLM, a ênfase recai sobre publicações de 2022 a 2026. Os critérios de inclusão abrangem trabalhos sobre geração ou revisão de artefatos, assurance, qualificação de ferramentas, rastreabilidade e supervisão humana; usos de IA sem relação com produção, revisão ou aceitação de artefatos técnicos ficam fora do escopo. O objetivo é tornar o mapeamento auditável e suficiente para delimitar a contribuição, sem caracterizá-lo como revisão sistemática segundo o Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA)

nem como prova de inexistência de trabalhos próximos.

2.2 Governança de IA, assurance e rastreabilidade Os referenciais de governança convergem em um ponto: usar IA de forma responsável exige identificar riscos, documentar decisões e manter supervisão humana. Isso aparece no NIST AI RMF, no NIST AI 600-1, nas normas ISO/IEC 42001 e ISO/IEC 23894 e no Artificial Intelligence Act (AI Act) da União Europeia, embora cada documento tenha objeto e finalidade próprios (NIST, 2023; AUTIO et al., 2024; ISO/IEC, 2023a; 2023b; EUROPEAN UNION, 2024).

No campo de assurance, o Assurance Case Centric Engineering of Safety-critical Systems (ACCESS) integra um assurance case evolutivo e métodos formais a sistemas críticos, enquanto AMLAS relaciona requisitos, desenvolvimento, verificação e evidências ao assurance de ML (HAWKINS et al., 2021; WEI et al., 2024). Ahlbrecht, Sprockhoff e Durak (2025) integram MBSE, System-Theoretic Process Analysis (STPA) e Goal Structuring Notation (GSN), reforçando que assurance orientado a modelos e artefatos já constitui um campo estabelecido.

A rastreabilidade conecta necessidades e requisitos às decisões de arquitetura, implementação, testes e evidências. Cleland-Huang et al. (2014) e Guo, Cheng e Cleland-Huang (2017) mostram sua relevância e sua automatização parcial; a National Aeronautics and Space Administration (NASA) também exige rastreabilidade bidirecional entre requisitos e elementos como design, código, verificações e não conformidades (NASA, 2022). Em termos práticos, rastrear significa conseguir responder de onde veio uma decisão técnica e quais evidências demonstram que ela foi atendida.

Dois trabalhos são particularmente próximos porque usam automação ou LLMs dentro da própria engenharia de assurance. Muram e Javed (2023) apresentam o ATTEST para automatizar revisão e atualização de argumentos de assurance. Odu et al. (2025), por sua vez, formalizam padrões GSN e usam LLMs para instanciar assurance cases, concluindo que a abordagem semiautomática continua preferível diante de limitações interpretativas. A sobreposição com este estudo é real: nos três casos, a IA modifica artefatos de assurance. A diferença reivindicada é de escopo e de unidade de análise: ATTEST e Odu et al. focalizam assurance cases e argumentos; aqui, o objeto inclui qualquer versão de artefato de Engenharia de Sistemas assistida por GenAI e submetida a gate, incluindo requisitos, arquitetura, código e evidência de V&V.

2.3 Automação, viés de supervisão e risco específico de GenAI A supervisão de automação não é neutra. Parasuraman e Manzey (2010) descrevem complacência e automation bias - tendência a confiar excessivamente na automação - como mecanismos capazes de gerar erros de omissão e comissão. Goddard, Roudsari e Wyatt (2012) mostram ainda que carga de trabalho, complexidade, confiança e forma de apresentação da recomendação podem aumentar ou reduzir essa sobreconfiança.

A GenAI acrescenta, aparentemente, uma dificuldade particular: confabulation. O NIST AI 600-1 usa esse conceito para tratar respostas apresentadas com confiança, mas que podem ser falsas ou errôneas (AUTIO et al., 2024). Isso permite formular uma previsão mais específica do que a regra genérica de 'mais exposição, mais risco': um erro fluentemente escrito e tecnicamente plausível, porém pouco sustentado por evidências, pode ser mais difícil de perceber na primeira leitura. O presente trabalho operacionaliza essa situação por meio do Gap de Plausibilidade-Evidência (GPE).

2.4 Escalas de automação/autonomia e classificações de criticidade As escalas clássicas de automação respondem a uma pergunta diferente da nossa. Sheridan e Verplank (1978) e Parasuraman, Sheridan e Wickens (2000) classificam como o controle ou as funções são distribuídos entre pessoa e máquina; a SAE J3016 classifica níveis de automação da tarefa dinâmica de direção. O AAL, por outro lado, não mede autoridade operacional: mede a materialidade da contribuição da GenAI na versão de um artefato. Da mesma forma, as classes C1-C3 não substituem níveis normativos como Safety Integrity Level (SIL) ou Automotive Safety Integrity Level (ASIL); IEC 61508 e ISO 26262 continuam sendo restrições externas quando aplicáveis.

Para explicitar essa diferença, a Tabela 1 compara o AAL com escalas de automação/autonomia e com classificações normativas de criticidade, mostrando que cada uma responde a uma pergunta distinta.

Tabela 1 - Fronteiras entre AAL, autonomia e criticidade normativa

Referencial	Objeto	Eixo	Diferença para o framework
Sheridan & Verplank / Parasuraman et al.	Alocação humano-automação	Autoridade/função automatizada	AAL mede efeito da contribuição de GenAI na versão do artefato.
SAE J3016	Dynamic driving task	Automação operacional 0-5	Não classifica processo de autoria técnica.
IEC 61508 / ISO 26262	Função/item relacionado à segurança	SIL/ASIL e risco funcional	C1-C3 mede consequência de falsa aceitação do AAIA; não há conversão.

Fonte: elaborado pelos autores com base em Sheridan e Verplank (1978), Parasuraman, Sheridan e Wickens (2000), SAE International (2021), IEC (2010a) e ISO (2018a; 2018b).

2.5 Disclosure, Model Cards e proveniência Em resumo, políticas editoriais atuais já diferenciam formas de uso de IA: A Elsevier distingue verificações básicas de linguagem de alterações substantivas que exigem disclosure, isto é, declaração transparente do uso. O International Committee of Medical Journal Editors (ICMJE) também exige transparência sobre tecnologias assistidas por IA. Model Cards documentam finalidade, desempenho e limitações de modelos, enquanto a Coalition for Content Provenance and Authenticity (C2PA) registra proveniência e ações sobre ativos digitais (ELSEVIER, 2026; ICMJE, 2026; MITCHELL et al., 2019; C2PA, 2024).

A Tabela 2 sintetiza essa distinção entre transparência, proveniência e materialidade, evidenciando por que disclosure, Model Cards e C2PA não substituem uma classificação de AAL no nível do artefato.

Tabela 2 - AAL frente a disclosure e proveniência

Referencial	Objeto	O que registra	Escala de materialidade no gate?	Diferença para AAL
Elsevier / ICMJE	Uso de IA em manuscrito	Disclosure, autoria e supervisão	Não	Transparência editorial, não assurance de artefato técnico.
Model Cards	Modelo de ML	Uso pretendido, desempenho e limitações	Não	Documenta o modelo, não sua contribuição a uma versão técnica.
C2PA	Ativo digital	Proveniência e ações verificáveis	Não	Registra origem/ações, evitando juízo de qualidade.
AAL	Versão de AAIA	Materialidade da contribuição no artefato final	Sim	Graduado para definir intensidade de assurance.

Fonte: elaborado pelos autores com base em Elsevier (2026), ICMJE (2026), Mitchell et al. (2019) e C2PA (2024).

USC/S também não equivale às unidades tradicionais de inspeção de software nem às métricas usuais de assistência à programação. Defect density por linhas de código, função ou módulo normaliza defeitos pelo tamanho do produto; acceptance rate, lines accepted, tokens gerados ou AI-generated content ratio medem volume ou interação com a ferramenta. S mede outra propriedade: a fração das unidades semanticamente consequenciais da versão final cuja semântica técnica permanece atribuível à GenAI. Por isso, uma sugestão extensa pode ser aceita e depois amplamente reescrita, enquanto poucas linhas podem constituir uma USC nuclear. A reanálise da telemetria Microsoft (2026), apresentada na Seção 4.5, fornece evidência discriminante para essa separação: 41,2% das sugestões aceitas sofreram edição posterior.

2.6 Qualificação de ferramentas: o análogo técnico mais próximo A qualificação de ferramentas é o análogo mais próximo do problema estudado porque considera a possibilidade de uma ferramenta introduzir um erro no produto ou deixar de detectar um erro existente. O DO-330, documento

complementar ao conjunto DO-178C/DO-278A, estrutura objetivos de qualificação de software tools conforme o impacto e a confiança requerida no uso. Na ISO 26262-8:2018, Cláusula 11, a confiança em software tools considera Tool Impact (TI, impacto da ferramenta), Tool Error Detection (TD, capacidade de detectar erro da ferramenta) e o correspondente Tool Confidence Level (TCL, nível de confiança na ferramenta). No DO-330, a lógica equivalente conduz a Tool Qualification Levels (TQL, níveis de qualificação da ferramenta). Na IEC 61508-3:2010, Cláusula 7.4.4, seleção e uso de ferramentas são tratados no desenvolvimento de software relacionado à segurança (RTCA, 2011; ISO, 2018c; IEC, 2010b).

O framework proposto não substitui essas normas e não pretende qualificar um LLM. A diferença está na unidade observada. A qualificação de ferramentas busca estabelecer alguma confiança no uso de uma ferramenta dentro de um processo regulado; aqui, observa-se uma versão concreta de artefato e a contribuição efetivamente realizada por uma GenAI, que pode variar entre interações, modelos e prompts. Em aplicações reguladas, os requisitos de qualificação continuam válidos; o framework atua apenas como camada complementar de rastreabilidade e controle do uso.

A comparação direta com qualificação de ferramentas é apresentada na Tabela 3, que delimita o que o framework aproveita conceitualmente desses referenciais e o que ele não pretende substituir.

Tabela 3 - Framework proposto frente à qualificação de ferramentas

Referencial	Unidade	Pergunta central	Tratamento de detecção	Diferença/integração
DO-330	Software tool em contexto aeronáutico	Qual confiança/qualificação é necessária ao uso da ferramenta?	Considera impacto do tool e verificação/detecção downstream conforme contexto	Framework não atribui TQL; registra efeito real da GenAI em cada versão de artefato.
ISO 26262-8, cl. 11	Software tool no ciclo automotivo	Qual confiança é necessária no tool?	TI + TD conduzem ao TCL	C1/D são analogias conceituais, sem conversão para TI/TD/TCL.
IEC 61508-3, cl. 7.4.4	Software tools e linguagens no desenvolvimento safety-related	Como selecionar/usar tools de modo compatível com integridade?	Rigor cresce quando o tool pode criar ou ocultar erro	Framework acrescenta proveniência, materialidade e gate por versão; não substitui conformidade.
Framework proposto	Versão de AAIA submetida a gate	Quanto a GenAI influenciou a versão e que assurance é exigido?	D é calibrado empiricamente para o protocolo de revisão	Pode complementar tool qualification em contextos regulados.

Fonte: elaborado pelos autores com base em RTCA (2011), ISO 26262-8:2018, Cláusula 11, e IEC 61508-3:2010, Cláusula 7.4.4. Não se propõe equivalência entre AAL/C/D e TQL, TCL ou classes IEC.

2.7 Síntese da lacuna O confronto com tool qualification reduz a amplitude da novidade reivindicada, mas torna a contribuição mais precisa. A literatura já sabe tratar ferramentas capazes de inserir ou ocultar erro. A lacuna residual está em combinar, para cada versão de artefato assistida por GenAI, seis elementos: materialidade semanticamente medida, tipo de intervenção, criticidade da falsa aceitação, proveniência da interação, revisão independente e gate não compensatório. O framework, portanto, complementa tanto a governança de IA quanto a qualificação de ferramentas.

3. Metodologia

A pesquisa adota Design Science Research (DSR) para desenvolver o artefato metodológico. Conforme Peffers et al. (2007), a DSR compreende seis atividades: identificação do problema, definição dos objetivos da solução, design e desenvolvimento, demonstração, avaliação e comunicação. Este trabalho cobre as quatro primeiras e define um protocolo prospectivo para a avaliação. Portanto, a avaliação experimental não é apresentada como concluída.

3.1 Definições operacionais e segmentação de USC Para evitar ambiguidades na aplicação do método, a Tabela 4 apresenta as definições operacionais que sustentam a unidade de análise do estudo.

Tabela 4 - Definições operacionais

Termo	Definição
Artefato de Engenharia	Representação técnica controlável do ciclo de vida: requisito, modelo, decisão arquitetural, código, caso de teste, relatório de V&V, registro de configuração ou equivalente.
AAIA	Versão de Artefato de Engenharia que recebeu intervenção atribuível a GenAI em A1-A4 ou em um tipo de intervenção IT diferente de IT0.
USC	Unidade Semântica Consequencial: menor unidade cujo conteúdo pode alterar significado técnico, comportamento, decisão ou evidência.
Superfície semântica exposta S	Proporção de USC consequenciais da versão final que foram geradas ou substantivamente transformadas por GenAI. S - [0,1].
Gate	Ponto formal em que responsável autorizado aprova, rejeita ou condiciona incorporação da versão à baseline/evidência.

Fonte: elaborado pelos autores. AAIA, USC e S são definições operacionais propostas neste estudo.

Como a USC depende do tipo de artefato analisado, a Tabela 5 detalha regras de segmentação para requisitos, arquitetura, código, testes, V&V e decisões de engenharia.

Tabela 5 - Regras de segmentação de USC por tipo de artefato

Tipo de artefato	USC primária	Regra de segmentação
Requisitos	Declaração de requisito ou critério de aceitação	Uma obrigação/testável por USC; critérios independentes contam separadamente.
Arquitetura/design	Decisão, elemento funcional/interface ou relação consequencial	Separar apenas quando a alteração puder mudar alocação, interface, restrição ou comportamento.
Código	Função/método ou bloco lógico consequencial	Usar função/método como padrão; dividir bloco apenas quando houver decisões independentes de comportamento.
Teste	Caso de teste	Precondição+ação+resultado esperado formam uma USC; casos independentes contam separadamente.
V&V / assurance	Claim ou conclusão ligada a evidência	Cada conclusão de conformidade, não conformidade ou aceitação é uma USC.
Decisão de engenharia	Decision record	Uma decisão com alternativas/racional próprios constitui uma USC.

Fonte: elaborado pelos autores. O protocolo prioriza unidade funcional/decisória, não contagem mecânica de palavras ou linhas.

Em termos simples, a metodologia segue uma sequência: primeiro divide-se o artefato em USC; depois identifica-se quanto da semântica dessas unidades veio da GenAI (AAL e S); em seguida avalia-se a consequência de uma aceitação incorreta (C1-C3); por fim, aplicam-se revisão, evidências e vetos antes da decisão de gate. As seções seguintes formalizam cada uma dessas etapas.

Depois dessa sequência geral, o Quadro 1 explicita as rubricas usadas para reconhecer uma USC nuclear e para atribuir plausibilidade aparente antes da checagem de evidências.

Quadro 1 - Rubricas operacionais para USC nuclear e plausibilidade aparente

Construto	Critério/valor	Âncora operacional	Saída
USC nuclear	N1 - comportamento	A alteração da USC muda comportamento funcional, lógica de controle, interface, alocação ou restrição técnica central.	Sim/Não

Construto	Critério/valor	Âncora operacional	Saída
USC nuclear	N2 - verificação	A alteração muda critério de aceitação, resultado esperado de teste ou interpretação de V&V.	Sim/Não
USC nuclear	N3 - decisão/baseline	A alteração muda uma decisão de engenharia, conclusão técnica ou conteúdo que será baselined.	Sim/Não
USC nuclear	Regra de classificação	Nuclear se ao menos um N1-N3 = Sim e a consequência não for meramente editorial/auxiliar. Dois avaliadores; divergência adjudicada.	Binária
Plausibilidade Puv	0,00	Defeito aparente/contradição técnica reconhecível sem consultar evidência externa.	Âncora
Plausibilidade Puv	0,25	Baixa plausibilidade; inconsistências internas ou incompatibilidade técnica são perceptíveis.	Âncora
Plausibilidade Puv	0,50	Plausível, mas contém salto lógico, premissa não sustentada ou sinal técnico de cautela.	Âncora
Plausibilidade Puv	0,75	Coerente e tecnicamente verossímil na leitura inicial; nenhuma inconsistência interna evidente.	Âncora
Plausibilidade Puv	1,00	Altamente convincente na forma e coerência interna antes de qualquer checagem externa.	Âncora

Fonte: elaborado pelos autores. Puv mede plausibilidade aparente antes da inspeção de evidências, não correção factual. Em C2/C3, Puv é avaliado cegamente por dois revisores; a concordância é reportada por ICC e divergências relevantes são adjudicadas.

A segmentação é prospectiva: deve ser feita durante a coleta futura, quando a história de edição ainda pode ser observada. Para versões C2/C3, dois avaliadores independentes segmentam e classificam o artefato; divergências são resolvidas por um terceiro avaliador. A concordância do AAL será reportada por kappa ponderado e a concordância de S por coeficiente de correlação intraclasse (ICC). Quando a documentação histórica não permite reconstruir S com segurança, o valor recebe NR (não recuperável) e não entra nos testes causais baseados em S.

3.2 Materialidade AAL e tipo de intervenção: dimensões separadas AAL e tipo de intervenção respondem a perguntas diferentes. O AAL indica quanto da semântica técnica do artefato final permanece atribuível à GenAI; por isso, a escala contém apenas A0-A4 e preserva uma ordem crescente de materialidade. Já o tipo de intervenção descreve o que a IA fez - edição, análise, geração ou recomendação de decisão. Assim, uma recomendação de aprovação pode ocorrer com baixa ou alta materialidade, mas não 'supera' artificialmente uma arquitetura substantivamente gerada.

A operacionalização dessa materialidade é apresentada na Tabela 6, que organiza o AAL em níveis A0-A4 e indica a regra de classificação aplicável a cada caso.

Tabela 6 - Escala ordinal de materialidade AAL

AAL	Critério	S	Regra
A0 - Ausente	Nenhuma intervenção de GenAI.	S=0	Sem uso identificado.
A1 - Não semântica	Apenas ortografia, gramática, formatação ou clareza sem alteração de USC.	S=0	Qualquer alteração de significado exclui A1.
A2 - Influência sem incorporação	GenAI influencia análise/seleção, mas nenhuma USC de sua saída é incorporada sem reconstrução humana independente.	S=0	Influência deve ser registrada; se houver USC incorporada, A3/A4.
A3 - Incorporação parcial	USC de GenAI são incorporadas e representam menos de 50% da superfície consequencial; nenhuma USC nuclear.	0<S<0,50	Maior nível aplicável sem satisfazer A4.

AAL	Critério	S	Regra
A4 - Incorporação substantiva	GenAI responde por ao menos 50% das USC consequenciais OU produz USC nuclear que define comportamento, arquitetura, lógica de teste ou conclusão técnica central.	S>=0,50 ou USC nuclear	Critério nuclear prevalece sobre proporção.

Fonte: elaborado pelos autores. O limiar de 50% e a regra de USC nuclear são hipóteses de design a serem calibradas.

Como a materialidade não descreve sozinha o que a IA fez, a Tabela 7 complementa a escala AAL com os tipos de intervenção da GenAI.

Tabela 7 - Tipo de intervenção da GenAI (IT)

Código	Tipo	Descrição
IT0	Nenhuma	Sem intervenção.
ITE	Edição	Transformação linguística/formal.
ITA	Análise	Síntese, comparação, detecção de lacunas, explicação.
ITG	Geração	Produção de conteúdo técnico candidato.
ITD	Recomendação de decisão	Recomendação explícita de aprovar, rejeitar, liberar, declarar conformidade ou fechar gate.

Fonte: elaborado pelos autores. IT é ortogonal ao AAL; ITD nunca possui autoridade para fechar gate.

A fronteira A2/A3 é determinada pelo que permanece na versão final, e não pela intenção do usuário. Em A2, a GenAI pode influenciar a análise ou a seleção de alternativas, mas nenhuma USC da saída permanece semanticamente incorporada: a reconstrução humana deve ser independente e completa. Se essa reconstrução for apenas parcial e qualquer conteúdo semanticamente atribuível à saída de GenAI sobreviver na versão submetida ao gate, então S>0 e a classificação mínima é A3. Essa regra elimina a zona cinzenta entre 'influência' e 'incorporação'.

3.3 Criticidade, revisão e evidência Além da criticidade, o framework distingue o grau de independência da revisão: IR0 indica revisão pelo próprio responsável pelo artefato; IR1 exige um segundo profissional qualificado; e IR2 exige um revisor independente da produção. Quanto maior a consequência de uma falsa aceitação, maior o grau mínimo de independência requerido. A relação entre consequência da falsa aceitação e independência da revisão é consolidada na Tabela 8.

Tabela 8 - Criticidade e revisão mínima

Classe	Consequência de falsa aceitação	Exemplo	Revisão mínima
C1	Permanece auxiliar; não altera baseline, liberação, segurança ou V&V.	Rascunho não decisório.	IR0: autor/responsável.
C2	Pode propagar erro para especificação, design, implementação ou preparação de teste, mas há verificação posterior.	Código preliminar, alternativa arquitetural, caso candidato.	IR1: segundo profissional qualificado.
C3	Pode autorizar/sustentar baseline, liberação, conformidade, V&V ou aceitação.	Requisito baselined, código liberado, evidência de V&V, decisão final.	IR2: revisor independente da produção.

Fonte: elaborado pelos autores. C1-C3 é criticidade de assurance e não substitui classificação de segurança setorial.

Além da revisão, cada decisão exige evidências mínimas. A Tabela 9 organiza essas evidências em quatro classes, cobrindo proveniência, rastreabilidade, verificação e decisão.

Tabela 9 - Classes de evidência

Classe	Evidência mínima
E1 - Proveniência	Ferramenta/modelo, data, versão quando disponível, instrução/prompt relevante, saída e versão do artefato.
E2 - Rastreabilidade	Vínculo com necessidade, requisito, risco, interface, item de configuração ou caso de teste.
E3 - Verificação	Inspeção, análise, demonstração, teste ou revisão independente contra critério de aceitação.
E4 - Decisão	Aprovação/rejeição, responsável, justificativa, versão e risco residual.

Fonte: elaborado pelos autores, informado por NIST (2023), NASA (2022), Hawkins et al. (2021) e Wei et al. (2024).

3.4 Mecanismo causal: variável dependente, - e GPE A lógica do mecanismo pode ser entendida sem a equação: cada defeito existente em uma USC pode ser interceptado antes do gate ou escapar para a versão aceita. O modelo procura explicar como materialidade da contribuição da IA, criticidade, capacidade de detecção e plausibilidade sem evidência alteram essa chance de escape.

O mecanismo causal acompanha cada USC defeituosa até o gate. Para a USC u da versão v, define-se Yuv=1 quando o defeito atravessa o gate sem ser detectado e Yuv=0 quando é interceptado antes do fechamento. Assim, a variável dependente Pr(Yuv=1) tem interpretação direta: é a probabilidade de escape de um defeito presente em uma USC. O modelo não atribui unidade física à palavra “risco” nem multiplica uma classe ordinal por uma quantidade contínua. A criticidade entra no modelo como variável categórica, por indicadores I(C2) e I(C3), mantendo C1 como referência; C1-C3 não são tratados como intervalares. A força de detecção é - - [0,1], estimada para um protocolo de revisão por -- = defeitos semeados corretamente detectados / defeitos semeados. O símbolo - é deliberadamente distinto do controle D do catálogo, que significa decisão registrada.

O GPE é calculado no mesmo nível da variável dependente. Para uma USC defeituosa, G_uv=P_uvx(1-E_uv). P_uv-[0,1] representa a plausibilidade aparente, avaliada cegamente antes de consultar as evidências; E_uv-[0,1] representa a cobertura de evidência independente para aquela mesma USC. E_uv é a razão entre o número de verificações ou evidências pré-especificadas satisfeitas e o total aplicável à USC. Se esse denominador não tiver sido definido previamente, o GPE não é calculado. Dessa forma, plausibilidade e evidência se referem à mesma unidade semântica.

O modelo confirmatório é logístico e hierárquico por versão/equipe: logit[Pr(Yuv=1)] = -0 + -1S_v + -2I(C2) + -3I(C3) + -4- + -5G_uv + -6(S_vx-) + -7(G_uvxEF), em que EF indica a condição de revisão evidence-first. Os coeficientes, e não relações de proporcionalidade impostas, determinam direção e magnitude. A extensão específica de GenAI é refutada se -5 não for positivo com precisão suficiente e/ou se -7 não indicar atenuação do efeito de GPE pela revisão baseada em evidência. A expectativa teórica é -4<0, porém o ajuste não impõe o sinal do coeficiente.

3.5 Gates não compensatórios e papel do IGAIA-C Na prática, um gate funciona como uma lista de condições obrigatórias. Um indicador agregado pode mostrar que quase todos os controles foram atendidos, mas não pode compensar a falta de um controle considerado veto. Por isso, um único veto não satisfeito mantém a versão em HOLD (pendente) ou FAIL (reprovado) até que a evidência faltante seja resolvida.

A decisão de gate vem antes de qualquer indicador agregado. O catálogo utiliza oito controles: P - proveniência; R - rastreabilidade; O - responsável humano; I - independência de revisão; A - critério de aceitação prévio; V - verificação objetiva; C - configuração; e D - decisão registrada. Para C3, todos funcionam como vetos. Para C2, P, O, A, V, C e D são vetos, e R também se torna veto quando existe dependência upstream/downstream. Para C1, P, O e D são obrigatórios. Como AAIA pressupõe intervenção de IA, proveniência não é opcional em nenhuma classe.

O IGAIA-C é o indicador de cobertura dos controles de governança aplicáveis e é calculado por IGAIA-C = 100 x Nc/Napp. Ele serve apenas para mostrar quão completo está o conjunto de controles; não decide o gate. Os estados PASS (aprovado), FAIL (reprovado) e HOLD (pendente) são determinados pelos vetos. Assim, a falha de qualquer veto requerido impede o fechamento do gate

mesmo que o IGAIA-C seja alto, preservando a lógica não compensatória. A Figura 1 reúne visualmente esses elementos, posicionando a assistência da IA, a governança human-in-the-loop e os controles de gate ao longo do Modelo em V.

Figura 1 - Integração da governança Human-in-the-Loop ao Modelo em V

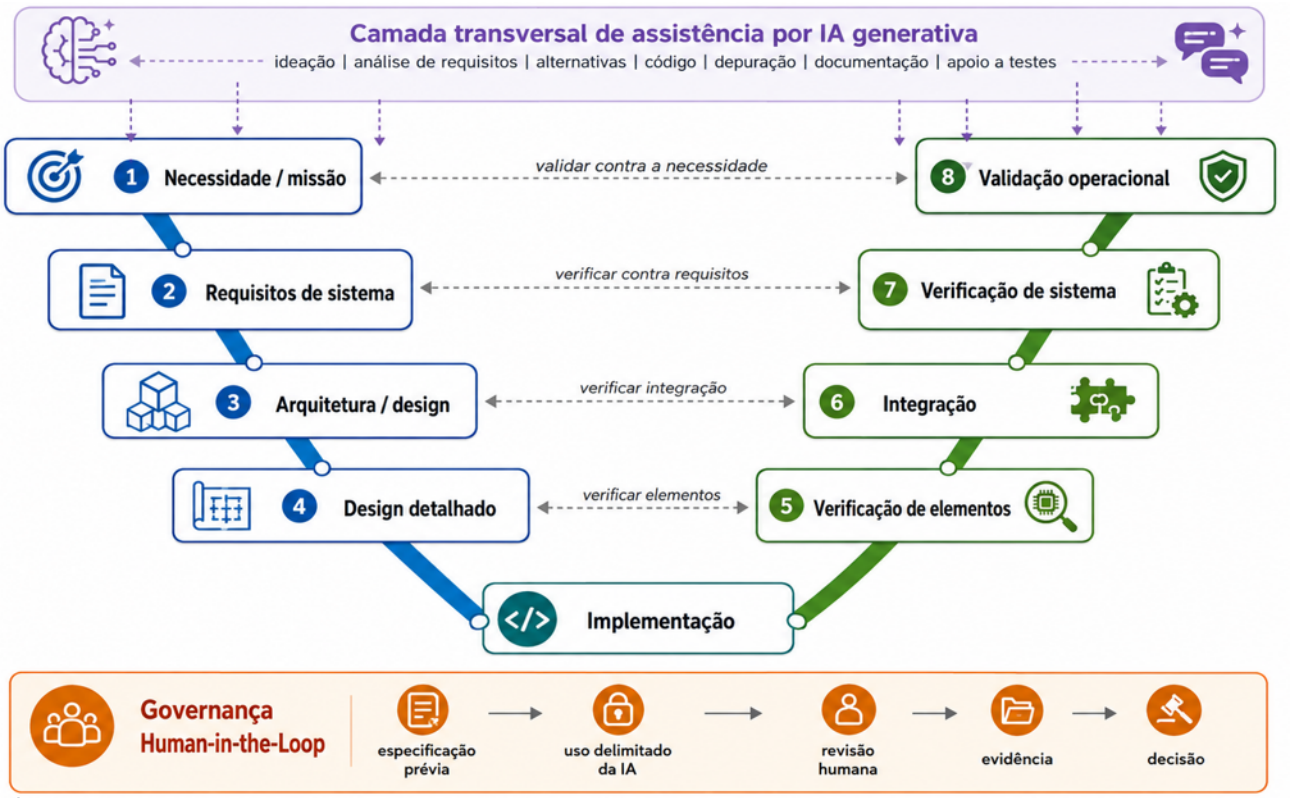


Figura 1

ideação | análise de requisitos | alternativas | código | depuração | documentação | apoio a testes

1 Necessidade / missão 2 Requisitos de sistema 3 Arquitetura / design 4 Design detalhado

Implementação

5 Verificação de elementos 6 Integração 7 Verificação de sistema 8 Validação operacional

validar contra a necessidade verificar contra requisitos verificar integração verificar elementos

Governança Human-in-the-Loop

especificação prévia uso delimitado da IA revisão humana evidência decisão Fonte: elaborado pelos autores para este case com apoio do ChatGPT.

3.6 Hipóteses, instrumentos e desenho experimental Cada hipótese é ligada a um instrumento de medição e a um smallest effect size of interest (SESOI, menor efeito de interesse) provisório. Esses limiares são alvos definidos antes da coleta, e não efeitos já observados. Um estudo piloto servirá para estimar a variabilidade dos dados e, a partir dela, dimensionar a etapa confirmatória. A Tabela 10 consolida as hipóteses prospectivas, os instrumentos previstos, os efeitos mínimos de interesse e as respectivas condições de refutação.

Tabela 10 - Hipóteses testáveis, instrumentos e refutação

H	Predição	Instrumento	SESOI provisório	Refutação
H1	Framework aumenta cobertura de rastreabilidade.	Matriz E2 por versão.	+15 pp	Diferença <15 pp.

H	Predição	Instrumento	SESOI provisório	Refutação
H2	Gates aumentam interceptação pré-baseline.	Log de correções/rejeições.	+10 pp	Diferença <10 pp.
H3	Critérios prévios reduzem retrabalho pós-gate.	Horas/ciclos normalizados.	-15%	Redução <15%.
H4	Catálogo aumenta completude documental.	Checklist P-R-O-I-A-V-C-D.	+15 pp	Diferença <15 pp.
H5	C3 exige mais intervenção que C1/C2.	Intervenções/rejeições por classe.	+20 pp	Diferença <20 pp.
H6	Maior S aumenta densidade de correção a K constante.	Correções por 100 USC.	A4>A3 >=15%	Sem tendência/diferença <15%.
H7	A4/C3 aumenta esforço pré-gate sob o framework.	Minutos por 100 USC.	-Tpré >=+15%	-Tpré<=0 sem maior interceptação.
H8	Maior - atenua efeito de S sobre defeitos escapados.	Defeitos pós-gate e interação Sx-.	Interação negativa	Interação nula/positiva com precisão suficiente.
H9 - GenAI	Entre saídas GenAI defeituosas, GPE alto reduz detecção em primeira leitura; revisão evidence-first atenua a diferença.	Taxa de detecção first-pass, G e condição de revisão.	>=10 pp de menor detecção no alto G; redução >=50% do gap com evidence-first	Gap ausente/invertido ou não reduzido pela revisão baseada em evidência.

Fonte: elaborado pelos autores. H9 é a predição específica de GenAI derivada do risco de confabulação/plausibilidade sem evidência.

O desenho confirmatório proposto é um crossover pareado por equipe: as mesmas equipes executam tarefas equivalentes em condições diferentes, com a ordem contrabalançada para reduzir efeito de aprendizagem. Quando viável, a revisão é cega. Para H9, defeitos técnicos são inseridos em saídas GenAI e estratificados prospectivamente pelo GPE; a primeira leitura ocorre antes da apresentação de evidências externas e é seguida de uma condição evidence-first, na qual o revisor examina primeiro as evidências. Tipo de artefato, experiência, AAL, S e criticidade são tratados como covariáveis. O piloto não será usado para confirmar hipóteses, apenas para calibrar -, GPE, variância e tamanho amostral.

3.7 Triangulação empírica externa: dados e procedimentos A triangulação empírica externa utiliza dois datasets públicos independentes. O primeiro é o LLM-Generated Software Requirements from GitHub Issues, Zenodo (2026) v2.1.1 (DOI 10.5281/zenodo.19520570), com 900 requisitos gerados por dois LLMs e três estratégias de prompting. Na subamostra de validação humana, os modelos identificados são deepseek-r1-distill-llama-70b e o3-mini; como o arquivo público não informa snapshot/build mais granular para o3-mini, nenhuma versão adicional é inferida. A validação humana compreende 50 requisitos, cinco avaliadores independentes e notas de 1 a 5 para Unambiguity, Verifiability e Singularity. A análise reproduz os coeficientes de Krippendorff e Spearman fornecidos pelo pacote e acrescenta uma medida simples de extensão textual, definida pela contagem de tokens separados por espaço na variável Requirement. Essa extensão é correlacionada, por Spearman, com a média das avaliações humanas; a comparação entre julgamento automático e humano emprega o teste pareado de Wilcoxon.

Para controlar o aumento da probabilidade de falsos positivos decorrente de múltiplas comparações, aplica-se a correção de Holm à família das três correlações entre extensão e julgamento. A robustez dos resultados também é examinada por uma análise de sensibilidade aos confundidores disponíveis: os postos da extensão textual e do escore humano são residualizados por efeitos fixos de LLM, Prompt Style e Repository, e a associação é estimada sobre esses resíduos. Como o arquivo de validação contém Issue Title, mas não uma variável estruturada de tipo de issue, nenhuma classificação retrospectiva desse tipo é introduzida. Essas análises têm função de sensibilidade e não são

interpretadas como identificação causal.

O segundo conjunto reúne a telemetria pública de Mozannar et al. (2024), distribuída pelo repositório Microsoft (2026) coderec_programming_states. O arquivo data_labeled_study.pkl contém 21 sessões de programadores e 3.490 eventos. A taxa de aceitação considera apenas eventos Accepted e Rejected. Entre as sugestões aceitas, 'edição posterior' corresponde à ocorrência de valor maior que zero em pelo menos uma das cinco medidas de distância relativa de EditPercentage. As variáveis numTokens e confidence são obtidas do campo Measurements. A associação entre características da sugestão e edição posterior é estimada por Equações de Estimação Generalizadas (GEE), com família binomial, variância robusta e agrupamento por UserId; o modelo conjunto inclui log(1+tokens) e confiança. O custo de supervisão é representado pela fração de tempo, por usuário, dedicada a quatro estados da taxonomia CodeRec User Programming States (CUPS): Thinking/Verifying Suggestion, Editing Last Suggestion, Prompt Crafting e Waiting For Suggestion.

Essas reanálises constituem triangulação empírica externa parcial de construtos, e não validação do framework. Como nenhum dos datasets foi concebido para testar AAL, USC/S, C1-C3 ou a eficácia de gates, seus resultados não confirmam H1-H9. A função das análises é mais delimitada: verificar validade discriminante e plausibilidade empírica de premissas específicas, mantendo as hipóteses causais como prospectivas.

3.8 Auditabilidade e pacote de reprodutibilidade A auditabilidade das reanálises é apoiada por um pacote suplementar de reprodutibilidade (Rev02), disponível publicamente no Zenodo, registro https://zenodo.org/records/22542208, Digital Object Identifier (DOI) 10.5281/zenodo.22541347 (ISSMAEL JUNIOR; CALVANO, 2026c). O depósito reúne scripts independentes para as análises Zenodo e Microsoft, README com regras operacionais, especificação de ambiente e versões, semente de projeto 20260906 e saídas tabulares/model summaries. Os datasets originais não são redistribuídos; os scripts recebem os arquivos públicos em formato ZIP como entrada. A licença de reutilização é a indicada nos metadados do registro. Para verificação de integridade, o arquivo originalmente depositado possui hash SHA-256 (Secure Hash Algorithm de 256 bits) 5e43dc6e61af258e557ab22ee2abfe5b0ecab3f70b4befff7cd0594b37b9f397.

Na telemetria Microsoft, 'edição posterior' adota o critério estrito de que a sugestão não permaneceu inalterada: basta que ao menos uma das cinco distâncias relativas em EditPercentage seja maior que zero. A análise de sensibilidade mostra que a conclusão permanece qualitativamente estável sob limiares mais exigentes: 41,2% das sugestões aceitas apresentam qualquer edição, 39,9% excedem 5% e 33,9% excedem 10% de distância máxima. Os quatro estados CUPS usados para estimar o tempo específico de interação com Copilot representam atividades diretamente ligadas à assistência - elicitar, aguardar, verificar ou editar uma sugestão - e são definidos a priori. Atividades gerais de escrita, documentação e debugging não integram esse cálculo.

O tempo de supervisão é resumido pela média das 21 proporções individuais e por seu intervalo de confiança de 95% (IC95%), calculado com intervalo t bilateral (média=46,77%; desvio-padrão, DP=18,19%; graus de liberdade, gl=20), resultando em 38,49%-55,05%. No modelo GEE, as 386 sugestões aceitas possuem numTokens, mas 11 não apresentam confidence no campo Measurements; por isso, a análise conjunta considera 375 casos completos, sem imputação. A especificação utiliza família binomial, correlação de trabalho independente, covariância robusta e agrupamento por UserId.

4. Resultados e discussão

O framework resultante separa três dimensões que poderiam ser confundidas. A primeira é a materialidade AAL (A0-A4): quanto da semântica do artefato final permanece atribuível à GenAI. A segunda é o tipo de intervenção (IT): o que a IA fez. A terceira é a criticidade C: qual a consequência de aceitar incorretamente aquela versão. Uma recomendação de decisão ITD, por exemplo, pode coexistir com A2 ou A4, mas nunca possui autoridade para fechar o gate.

A força de detecção - torna-se observável e calibrável, eliminando a indeterminação do mecanismo anterior. A USC recebe regras específicas por tipo de artefato, e S é tratada como variável prospectiva. Se a trilha de edição não permitir recuperar S com segurança, o dado recebe NR e não é imputado. Essa escolha reduz a quantidade de dados retrospectivos utilizáveis, mas preserva a validade da medição.

O confronto com DO-330, ISO 26262-8 e IEC 61508-3 mostra que controlar uma ferramenta capaz de

introduzir ou ocultar erro é uma prática anterior e consolidada. A diferença operacional do framework é observar a contribuição concreta da GenAI em cada versão do artefato, em vez de inferir confiança global na ferramenta. Em contexto regulado, as duas camadas devem coexistir.

4.1 Demonstração no payload: cadeia real e gate retrospectivo A documentação do demonstrador ENEGEP permite reconstruir uma cadeia técnica real, embora incompleta: necessidade de comunicação local independente de infraestrutura externa -> requisito de comunicação -> exploração assistida por IA de alternativas e caminhos de implementação -> arquitetura baseada em módulo de baixo custo -> código preliminar e depuração assistidos por IA -> revisão humana, integração e ensaios de comunicação. A publicação também registra que outputs assistidos deveriam ser ligados a requisito, tarefa, verificação, decisão de implementação e item de especificação (ISSMAEL JUNIOR; CALVANO, 2026a).

A cadeia reconstruída do item de comunicação do payload é apresentada na Tabela 11, distinguindo o conteúdo documentado, a evidência disponível e a leitura produzida pelo framework.

Tabela 11 - Cadeia documental reconstruída do item de comunicação do payload

Elo	Conteúdo documentado	Evidência disponível	Leitura pelo framework
Necessidade	Experimentação com troca de dados/monitoramento em UAV de baixo custo.	Descrição de missão do demonstrador.	Origem humana identificável.
Requisito	Comunicação local sem dependência de infraestrutura externa.	Tabela de requisitos/baseline.	Rastreabilidade requisito->arquitetura preservada.
Alternativa/ / implementação	IA apoiou exploração de alternativas, código preliminar e depuração.	Descrição do ciclo de desenvolvimento.	Intervenção ITA/ITG documentada; S não recuperável.
Decisão técnica	Equipe humana selecionou alternativas admissíveis e integrou solução.	Método e aplicação narrados.	Owner coletivo conhecido; registro individual de gate ausente.
Verificação	Ensaios laboratoriais/campo contra comportamento esperado.	Relato de communication trials/test evidence.	E3 parcial; logs brutos não estão no documento disponível.

Fonte: reconstrução pelos autores a partir de ISSMAEL JUNIOR; CALVANO (2026a). Nenhum campo ausente foi preenchido por inferência.

Com base nessa cadeia, a Tabela 12 apresenta o registro retrospectivo de gate para o artefato AAIA-UAV-COM-01.

Tabela 12 - Registro retrospectivo de gate: AAIA-UAV-COM-01

Campo	Registro	Status	Justificativa
AAL / IT	AAL não atribuível prospectivamente porque S é NR; ITA/ITG documentados.	NR/PARCIAL	Intervenções conhecidas, trilha semântica insuficiente.
Criticidade	C2, por se tratar de proof of concept sujeito a integração/testes e não liberação operacional.	PASS	Falsa aceitação ainda encontraria verificação subsequente.
P	Uso de IA é conhecido, mas modelo/prompt/data/versão por item não foram preservados.	FAIL	E1 incompleta.
R	Vínculo comunicação->arquitetura->teste é identificável.	PASS	Cadeia documental reconstruível.
O/I	Responsabilidade humana e regra de segundo revisor estão descritas, mas nomes por versão não constam.	PARCIAL	Processo documentado; log individual ausente.

Campo	Registro	Status	Justificativa
A/V	Critério funcional e ensaios são descritos; logs brutos não disponíveis.	PARCIAL	Há evidência narrativa, não pacote completo.
C	Versão/configuração específica do código não identificada.	FAIL	Não há baseline versionada no documento acessível.
D	Incorporação ao protótipo é narrada, mas não há decisão individualizada datada.	FAIL	Falta E4 completa.
Gate	HOLD/FAIL até completar P, C e D e formalizar O/I.	FAIL	Aplicação das regras C2.

Fonte: aplicação retrospectiva do framework à documentação de ISSMAEL JUNIOR; CALVANO (2026a). O identificador é retrospectivo.

A demonstração produz dois aprendizados. Primeiro, o mecanismo baseado em S não pode ser testado quando a documentação retrospectiva é insuficiente; nesse caso, S deve permanecer como não recuperável, e não ser estimado artificialmente. Segundo, a cadeia técnica existente permite justificar C2 e identificar quais evidências faltam para fechar um gate auditável. A utilidade do framework, portanto, não está simplesmente em 'descobrir que faltam documentos', mas em distinguir entre informação suficiente para demonstrar que um protótipo funciona e informação suficiente para demonstrar, de forma reproduzível, que uma versão pode ser aceita.

4.2 Exemplo discriminante: decisão produzida pelo framework Considere uma versão de requisito de interface produzida com GenAI que altera o tempo máximo de resposta (timeout) admissível de 100 ms para 500 ms com base em uma premissa tecnicamente plausível, porém sem evidência independente. Suponha, para isolar os mecanismos, que o uso da ferramenta esteja declarado e que todas as obrigações externas aplicáveis de qualificação da ferramenta tenham sido satisfeitas. Essas condições garantem transparência e confiança no uso da ferramenta, mas não respondem à pergunta central do gate: esta versão específica pode entrar na baseline?

O Quadro 2 organiza esse exemplo contrafactual e mostra em que ponto o framework produz uma decisão no nível da versão do artefato que não decorre apenas de disclosure ou tool qualification.

Quadro 2 - Exemplo contrafactual de decisão discriminante

Camada	Condição no exemplo	Resultado	Consequência
Disclosure	Uso de GenAI, ferramenta e finalidade declarados.	Atendido	Não decide correção técnica da versão.
Tool qualification	Hipoteticamente satisfeita para o contexto de uso.	Atendido	Estabelece confiança/controle do tool, não rastreabilidade factual desta mudança.
Framework: AAL/IT	A4-ITG: USC nuclear modifica requisito de interface que será baselined.	Alta materialidade	Eleva intensidade de assurance da versão.
Framework: criticidade	C3: falsa aceitação altera baseline e critérios downstream de V&V.	C3	Todos os vetos aplicáveis tornam-se obrigatórios.
Framework: R/A/V	Premissa de 500 ms não rastreada a necessidade/requisito superior; evidência independente ausente.	FAIL	Falham rastreabilidade/critério/evidência.
Decisão do gate	Não incorporar a versão até restaurar rastreabilidade e verificação independente.	HOLD/FAIL	Decisão no nível do artefato, não produzida por disclosure ou qualificação do tool isoladamente.

Fonte: exemplo contrafactual elaborado pelos autores. Não constitui evidência empírica. Seu objetivo é mostrar a diferença entre controlar ou declarar o uso da ferramenta e decidir sobre uma versão concreta do artefato. Se a premissa de 500 ms for posteriormente rastreada e verificada, o mesmo framework pode permitir o fechamento do gate. Portanto, a decisão depende das evidências da versão, e não de uma proibição abstrata ao uso de GenAI.

4.3 Predição específica de GenAI: plausibilidade sem evidência H9 é a hipótese que diferencia o mecanismo de uma teoria genérica de defeitos. O NIST GenAI Profile reconhece confabulation como risco de conteúdo apresentado com confiança apesar de falso ou errôneo (AUTIO et al., 2024). O framework prevê que, entre saídas GenAI tecnicamente defeituosas e de severidade semelhante, maior GPE reduzirá a probabilidade de detecção na primeira leitura. A hipótese também prevê que uma revisão evidence-first, na qual as evidências são examinadas antes da decisão, deve reduzir esse efeito. Se GPE não se relacionar à detecção ou se a revisão baseada em evidência não alterar o gap, a extensão específica de GenAI deverá ser rejeitada ou reformulada.

4.4 Condições de transferência e limitações A transferência do framework para outros contextos exige cinco condições mínimas: versões identificáveis, regras de USC aplicáveis, rastreabilidade, autoridade humana de gate e critérios de aceitação. Em contextos sujeitos a DO-330, ISO 26262, IEC 61508 ou outros regimes, o framework não reduz obrigações de tool qualification, functional safety ou certification; ele apenas acrescenta uma camada de controle sobre a contribuição concreta da GenAI ao artefato.

O estudo possui limitações importantes. O mapeamento não é uma revisão sistemática exaustiva. - depende de calibração local e pode variar entre equipes. GPE e USC nuclear continuam envolvendo julgamento humano, embora agora usem rubricas, dupla avaliação e medidas de concordância. S é essencialmente prospectivo. A triangulação externa sustenta apenas viabilidade e validade discriminante de construtos selecionados, não eficácia causal dos gates. O Quadro 2 é contrafactual e demonstra uma distinção decisória, não utilidade observada por praticantes. Ainda não foi executado um piloto prospectivo comparando interceptação pré-gate com e sem o framework, nem foi observado em campo um caso real em que a decisão de gate divergiu daquela produzida apenas por disclosure e tool qualification. Esses dois testes constituem a próxima etapa empírica. Limiares AAL, SESOI, regras de veto e utilidade do artefato ainda requerem avaliação prospectiva própria.

4.5 Resultados da triangulação empírica externa No dataset Zenodo (2026), os cinco avaliadores apresentaram baixa concordância nos três critérios: -=0,287 para não ambiguidade, -=0,150 para verificabilidade e -=0,394 para singularidade. A avaliação automática, entretanto, manteve associação positiva com a média humana (-=0,326; 0,424; 0,488, respectivamente), sem ser equivalente a ela. O avaliador LLM atribuiu, em média, +0,356 ponto em não ambiguidade em relação à média humana, diferença significativa no teste de Wilcoxon (p=0,010). As diferenças em verificabilidade e singularidade não atingiram significância nessa amostra.

Os resultados consolidados dessa reanálise aparecem na Tabela 13, que relaciona concordância humana, associação entre avaliador automático e humanos, extensão textual e viés médio.

Tabela 13 - Evidência externa do dataset Zenodo (2026) para julgamentos semânticos

Critério	Krippendorff -	LLM x humano -	Palavras x humano -	Viés médio LLM-humano
Não ambiguidade	0,287	0,326 (p=0,0209)	+0,687 (p<0,000001)	+0,356; Wilcoxon p=0,010
Verificabilidade	0,150	0,424 (p=0,0022)	+0,635 (p<0,000001)	+0,156; p=0,177
Singularidade	0,394	0,488 (p=0,00032)	-0,697 (p<0,000001)	+0,204; p=0,209

Fonte: reanálise dos arquivos brutos do Zenodo (2026) v2.1.1 pelos autores. n=50 requisitos; cinco avaliadores humanos. O resultado central é o comportamento diferente da singularidade: requisitos mais longos foram avaliados como mais claros e verificáveis, porém menos singulares. Portanto, volume textual não constitui um proxy consistente de atomicidade semântica e não substitui a USC por simples contagem de palavras, tokens ou linhas.

A correção de Holm preservou significância nas três associações bivariadas entre extensão textual e avaliação humana; os valores p ajustados por essa correção são indicados por p_Holm (p_Holm<0,000001 nos três casos). Quando foram controlados LLM, Prompt Style e Repository na análise parcial em postos, a associação permaneceu negativa e forte para singularidade (r=-0,698; p_Holm<0,000001), tornou-se mais modesta para não ambiguidade (r=0,321; p_Holm=0,046) e deixou de ser estatisticamente distinguível de zero para verificabilidade (r=0,201; p_Holm=0,162). Assim, a conclusão mais robusta é a dissociação entre extensão textual e atomicidade/singularidade; as demais associações devem ser interpretadas com maior cautela.

A Figura 2 complementa essa leitura ao mostrar graficamente que a extensão textual se comporta de modo diferente para não ambiguidade, verificabilidade e singularidade.

Figura 2 - Correlação entre extensão textual e avaliação humana dos requisitos

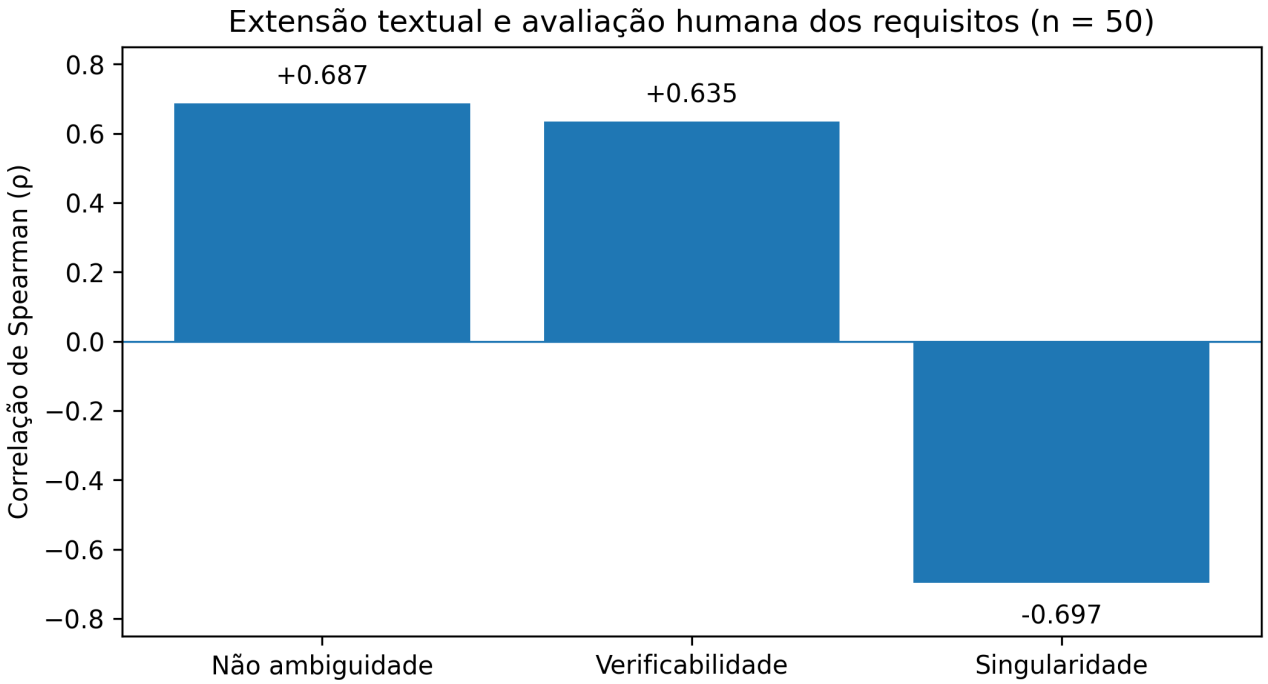


Figura 2

Eixo Y: Correlação de Spearman (-)

Valores do eixo Y:

0.8 0.6 0.4 0.2 0.0 -0.2 -0.4 -0.6 -0.8

Categorias do eixo X e valores anotados:

Não ambiguidade: +0.687 Verificabilidade: +0.635 Singularidade: -0.697 Fonte: elaborado pelos autores a partir da reanálise do Zenodo (2026) v2.1.1.

Na telemetria Microsoft (2026), foram observadas 1.096 decisões finais sobre sugestões: 386 Accepted e 710 Rejected, o que corresponde a uma taxa de aceitação de 35,2%. Entre as 386 sugestões aceitas, 159 (41,2%) apresentaram EditPercentage não nulo em pelo menos um dos cinco eventos still-in-code subsequentes. Em termos simples, aceitar uma sugestão na interface não significa que ela permaneça inalterada na versão de trabalho.

A síntese da reanálise da telemetria Microsoft é apresentada na Tabela 14, destacando aceitação, edição posterior, tamanho das sugestões, confiança do modelo e tempo de supervisão.

Tabela 14 - Evidência externa da telemetria Microsoft (2026) sobre aceitação, edição e supervisão

Indicador	N	Resultado	Interpretação restrita
Decisões Accepted/Rejected	1.096	386 aceitas; 710 rejeitadas	Taxa de aceitação = 35,2%
Sugestões aceitas com edição posterior	386	159	41,2% das aceitas
Tokens médios: sem edição vs. com edição	386	8,17 vs. 26,14	Sugestões posteriormente editadas eram maiores
Confiança média: sem edição vs. com edição	375 válidas	0,721 vs. 0,547	Editadas tinham menor confiança média

Indicador	N	Resultado	Interpretação restrita
GEE conjunto: log(1+tokens)	375 válidas	OR=2,115; p=0,00175	Maior extensão associada a maior chance de edição
GEE conjunto: confiança +0,1	375 válidas	OR=0,751; p=0,00185	Maior confiança associada a menor chance de edição
Tempo em atividades Copilot-específicas	21 usuários	média por usuário = 46,77%	IC95% t: 38,49%-55,05%

Fonte: reanálise pelos autores de data_labeled_study.pkl, Microsoft (2026) Research. ORs (odds ratios, razões de chances) do GEE conjunto com variância robusta agrupada por UserId. n=375 porque 11 das 386 sugestões aceitas não possuem confidence; não houve imputação. O tempo Copilot-específico é a média das proporções individuais dos quatro estados de interação direta, e seu IC95% é um intervalo t bilateral entre usuários (gl=20). Esses resultados não medem USC/S; mostram que aceitação, tamanho, confiança e persistência após revisão são fenômenos distintos.

A Figura 3 resume visualmente o achado central sobre persistência: aceitar uma sugestão Copilot não significa que ela permaneça inalterada na versão de trabalho.

Figura 3 - Persistência de sugestões Copilot aceitas

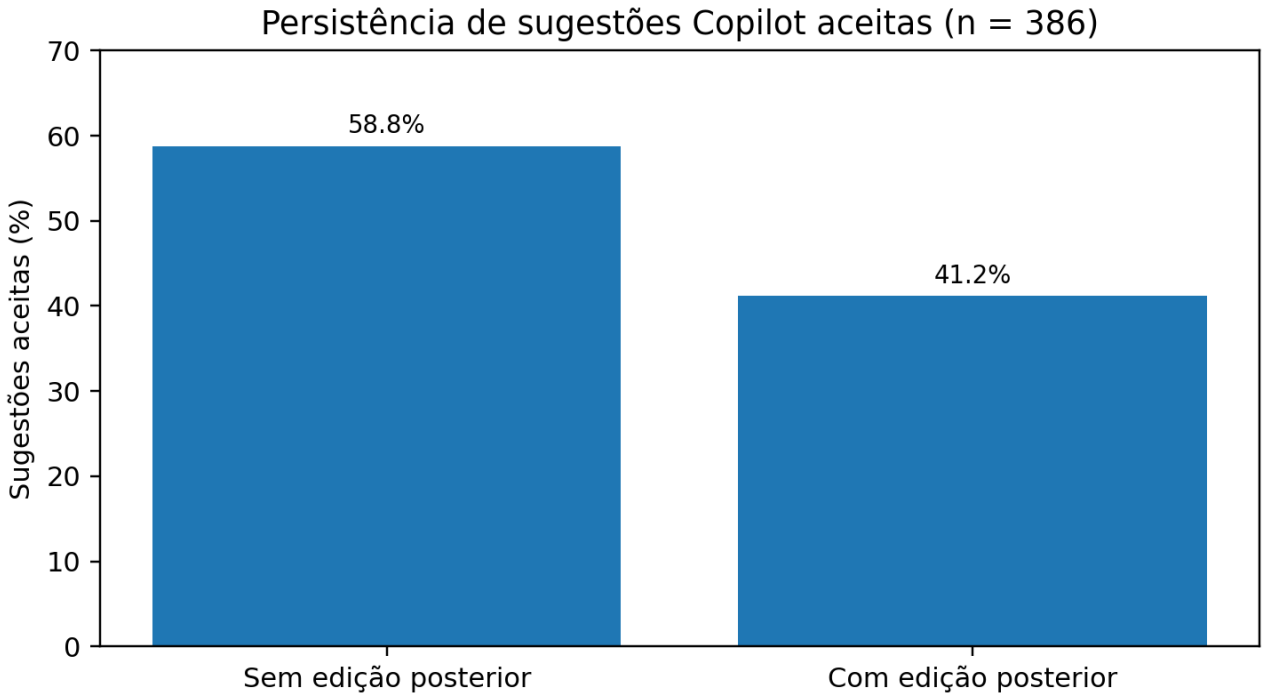


Figura 3

Sugestões aceitas (%)

Sem edição posterior — 58.8% Com edição posterior — 41.2% Fonte: elaborado pelos autores a partir da reanálise da telemetria pública de Mozannar et al. (2024).

Em conjunto, os dois datasets sustentam uma conclusão mais estreita do que “validação do framework”. O Zenodo (2026) mostra que propriedades semanticamente consequenciais exigem um protocolo de julgamento e que extensão textual não representa atomicidade de forma monotônica; a análise ajustada reforça sobretudo a associação negativa com singularidade. A Microsoft (2026) mostra que aceitar uma sugestão na interface não significa incorporá-la de forma inalterada e que verificação e edição consomem tempo observável. O GEE é estritamente associativo e não é usado como evidência confirmatória de H7. Esses padrões fornecem validade discriminante inicial para distinguir S/USC de contagens sintáticas e taxas de aceitação, mas não permitem estimar eficácia do gate, - causal, GPE ou defeitos escapados. H1-H9 permanecem sem confirmação causal.

5. Conclusão

O trabalho propõe um framework de assurance para versões de artefatos de Engenharia de Sistemas assistidas por GenAI. A contribuição é incremental e deliberadamente delimitada: combina fundamento normativo, confronto com tool qualification, formalização mensurável e triangulação empírica externa. No nível de cada versão submetida a gate, o framework reúne materialidade da contribuição, tipo de intervenção, criticidade de falsa aceitação, proveniência, revisão e evidências.

A separação entre AAL com níveis A0-A4 e IT preserva a ordinalidade da escala; - possui domínio e procedimento de calibração; USC recebe protocolo de segmentação; e S permanece uma variável prospectiva. Os dados Zenodo (2026) indicam que quantidade textual não equivale a atomicidade semântica, enquanto a telemetria Microsoft (2026) mostra que aceitação não equivale a incorporação inalterada: 41,2% das sugestões aceitas sofreram edição posterior. Esses resultados fortalecem a justificativa para medir a contribuição que persiste no artefato, em vez de depender apenas de volume ou métricas de interface.

A evidência secundária não confirma as hipóteses H1-H9. O mecanismo específico de GenAI formulado pelo GPE, a eficácia dos gates e a taxa de defeitos escapados ainda exigem avaliação prospectiva própria. A contribuição empírica desta etapa é mais limitada: demonstrar viabilidade de julgamento semântico, validade discriminante inicial de USC/S e custo observável de supervisão. As reanálises e suas decisões operacionais estão publicamente auditáveis no pacote de reprodutibilidade depositado no Zenodo (DOI 10.5281/zenodo.22541347), sem apresentar esses resultados como validação causal.

A presente versão também explicita uma decisão discriminante no nível do artefato: mesmo quando disclosure e requisitos externos de tool qualification estão satisfeitos, uma versão pode permanecer em HOLD/FAIL se a contribuição da GenAI for material e faltarem rastreabilidade ou evidência para o conteúdo proposto. A validação prospectiva de S, a avaliação por praticantes e a eficácia causal dos gates permanecem como agenda de pesquisa futura e não são reivindicadas como resultados deste case.

6. Política de Uso de Inteligência Artificial

Foi usada, parcialmente, a ferramenta ChatGPT, da OpenAI, como apoio à revisão linguística, à organização de alternativas de redação, à conferência bibliográfica e à elaboração gráfica conceitual da Figura 1. O uso ocorreu sob supervisão humana. Os autores permaneceram responsáveis pela seleção e interpretação das fontes, formulação do problema, definição metodológica, decisões técnicas, análise e conclusões. A ferramenta não é autora e não assume responsabilidade científica pelo conteúdo.

7. Referências

AHLBRECHT, A.; SPROCKHOFF, J.; DURAK, U. A system-theoretic assurance framework for safety-driven systems engineering. Software and Systems Modeling, v. 24, p. 253-270, 2025. DOI: https://doi.org/10.1007/s10270-024-01209-6.

AUTIO, C.; SCHWARTZ, R.; DUNIETZ, J.; JAIN, S.; STANLEY, M.; TABASSI, E.; HALL, P.; ROBERTS, K. Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile. NIST AI 600-1. Gaithersburg, MD: NIST, 2024. DOI: https://doi.org/10.6028/NIST.AI.600-1.

BARKE, S.; JAMES, M. B.; POLIKARPOVA, N. Grounded Copilot: How Programmers Interact with Code-Generating Models. Proceedings of the ACM on Programming Languages, v. 7, OOPSLA1, p. 85-111, 2023. DOI: https://doi.org/10.1145/3586030.

BORG, M. et al. Ergo, SMIRK is safe: a safety case for a machine learning component in a pedestrian automatic emergency brake system. Software Quality Journal, v. 31, p. 335-403, 2023. DOI: https://doi.org/10.1007/s11219-022-09613-1.

C2PA. C2PA Technical Specification, version 2.1. Coalition for Content Provenance and Authenticity, 2024. Seções pertinentes: technical overview, assertions, claims, manifests e provenance. URL: https://spec.c2pa.org/specifications/specifications/2.1/specs/_attachments/C2PA_Specification.pdf.

CLELAND-HUANG, J.; GOTEL, O. C. Z.; HAYES, J. H.; MÄDER, P.; ZISMAN, A. Software traceability: trends and future directions. In: Future of Software Engineering (FOSE), 2014. p. 55-69. DOI: https://doi.org/10.1145/2593882.2593891.

ELSEVIER. Generative AI policies for journals. Updated June 2026. URL: https://www.elsevier.com/about/policies-and-standards/generative-ai-policies-for-journals. Acesso em: 5 set. 2026.

EUROPEAN UNION. Regulation (EU) 2024/1689 (Artificial Intelligence Act). Official Journal of the European Union, 2024. Artigos pertinentes: 9, 12, 13 e 14. URL: https://eur-lex.europa.eu/eli/reg/2024/1689/oj.

GODDARD, K.; ROUDSARI, A.; WYATT, J. C. Automation bias: a systematic review of frequency, effect mediators, and mitigators. Journal of the American Medical Informatics Association, v. 19, n. 1, p. 121-127, 2012. DOI: https://doi.org/10.1136/amiajnl-2011-000089.

GUO, J.; CHENG, J.; CLELAND-HUANG, J. Semantically enhanced software traceability using deep learning techniques. ICSE, 2017, p. 3-14. DOI: https://doi.org/10.1109/ICSE.2017.9.

HAWKINS, R. D.; PATERSON, C.; PICARDI, C.; JIA, Y.; CALINESCU, R.; HABLI, I. Guidance on the Assurance of Machine Learning in Autonomous Systems (AMLAS). University of York, 2021. URL: https://www.york.ac.uk/assuring-autonomy/guidance/amlas/.

ICMJE. Recommendations for the Conduct, Reporting, Editing, and Publication of Scholarly Work in Medical Journals. Updated January 2026. Section V.A: Use of AI by Authors. URL: https://www.icmje.org/recommendations/browse/artificial-intelligence/ai-use-by-authors.html. Acesso em: 5 set. 2026.

IEC. IEC 61508-5:2010. Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 5: Examples of methods for determination of safety integrity levels. 2. ed. Geneva: IEC, 2010a. URL: https://webstore.iec.ch/en/publication/5519.

IEC. IEC 61508-3:2010. Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 3: Software requirements. 2. ed. Geneva: IEC, 2010b. Seção pertinente: 7.4.4, support tools and programming languages. URL de catálogo: https://webstore.iec.ch/en/publication/5517.

INCOSE. Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. 5. ed. Hoboken: Wiley, 2023. URL: https://www.incose.org/resource/incose-systems-engineering-handbook-a-guide-for-system-life-cycle-processes-and

ISO. ISO 26262-3:2018. Road vehicles - Functional safety - Part 3: Concept phase. 2. ed. Geneva: ISO, 2018a. URL: https://www.iso.org/standard/68385.html.

ISO. ISO 26262-9:2018. Road vehicles - Functional safety - Part 9: ASIL-oriented and safety-oriented analyses. 2. ed. Geneva: ISO, 2018b. URL: https://www.iso.org/standard/68391.html.

ISO. ISO 26262-8:2018. Road vehicles - Functional safety - Part 8: Supporting processes. 2. ed. Geneva: ISO, 2018c. Seção pertinente: Clause 11, Confidence in the use of software tools. URL: https://www.iso.org/standard/68390.html.

ISO/IEC. ISO/IEC 42001:2023. Information technology - Artificial intelligence - Management system. 1. ed. Geneva: ISO/IEC, 2023a. URL: https://www.iso.org/standard/42001.

ISO/IEC. ISO/IEC 23894:2023. Information technology - Artificial intelligence - Guidance on risk management. 1. ed. Geneva: ISO/IEC, 2023b. URL: https://www.iso.org/standard/77304.html.

ISO/IEC/IEEE. ISO/IEC/IEEE 15288:2023. Systems and software engineering - System life cycle processes. 2. ed. Geneva: ISO/IEC/IEEE, 2023. URL: https://www.iso.org/standard/81702.html.

ISSMAEL JUNIOR, A. K.; CALVANO, J. V. An Experimental Human-AI Co-Design Method for Low-Cost UAV Payload Development in Higher Education. XLVI ENEGEP, São Paulo, 2026. No prelo.

ISSMAEL JUNIOR, A. K.; CALVANO, J. V. Gestão Tecnológica de Projetos de 'Payloads' Inteligentes em UAVs com Apoio de IA Generativa: um Modelo de Maturidade. XXXIII SIMPEP, Bauru, 2026. No prelo.

ISSMAEL JUNIOR, A. K.; CALVANO, J. V. Reproducibility Package Rev02 for “From Generative AI to Auditable Engineering: A Human-in-the-Loop Assurance Framework for Engineering Artifacts in the V-Model”. Zenodo, 2026. URL: https://zenodo.org/records/22542208, DOI: https://doi.org/10.5281/zenodo.22541347

MITCHELL, M. et al. Model Cards for Model Reporting. Proceedings of FAT* 2019, p. 220-229. DOI: https://doi.org/10.1145/3287560.3287596.

MURAM, F. U.; JAVED, M. A. ATTEST: Automating the review and update of assurance case arguments. Journal of Systems Architecture, v. 134, art. 102781, 2023. DOI: https://doi.org/10.1016/j.sysarc.2022.102781.

NASA. SWE-052 - Bidirectional Traceability. NASA Software Engineering Handbook, Version D, 2022. URL: https://swehb.nasa.gov/spaces/SWEHBVD/pages/102695427/SWE-052+-+Bidirectional+Traceability.

NIST. Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1. Version 1.0. Gaithersburg, MD: NIST, 2023. DOI: https://doi.org/10.6028/NIST.AI.100-1.

ODU, O.; BELLE, A. B.; WANG, S.; KPODJEDO, S.; LETHBRIDGE, T. C.; HEMMATI, H. Automatic instantiation of assurance cases from patterns using large language models. Journal of Systems and Software, v. 222, art. 112353, 2025. DOI: https://doi.org/10.1016/j.jss.2025.112353.

PARASURAMAN, R.; MANZEY, D. H. Complacency and Bias in Human Use of Automation: An Attentional Integration. Human Factors, v. 52, n. 3, p. 381-410, 2010. DOI: https://doi.org/10.1177/0018720810376055.

PARASURAMAN, R.; SHERIDAN, T. B.; WICKENS, C. D. A model for types and levels of human interaction with automation. IEEE Transactions on Systems, Man, and Cybernetics - Part A, v. 30, n. 3, p. 286-297, 2000. DOI: https://doi.org/10.1109/3468.844354.

PEFFERS, K.; TUUNANEN, T.; ROTHENBERGER, M. A.; CHATTERJEE, S. A design science research methodology for information systems research. Journal of Management Information Systems, v. 24, n. 3, p. 45-77, 2007. DOI: https://doi.org/10.2753/MIS0742-1222240302.

RTCA. DO-330: Software Tool Qualification Considerations. Washington, DC: RTCA, 2011. Documento complementar à família DO-178C/DO-278A. Catálogo RTCA: https://www.rtca.org/?s=DO-330.

SAE INTERNATIONAL. SAE J3016_202104. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. Revised April 2021. DOI: https://doi.org/10.4271/J3016_202104.

SHERIDAN, T. B.; VERPLANK, W. L. Human and Computer Control of Undersea Teleoperators. MIT Man-Machine Systems Laboratory, 1978. DOI: https://doi.org/10.21236/ADA057655.

VAITHILINGAM, P.; ZHANG, T.; GLASSMAN, E. L. Expectation vs. Experience: Evaluating the Usability of Code Generation Tools Powered by Large Language Models. CHI Extended Abstracts, 2022. DOI: https://doi.org/10.1145/3491101.3519665.

WEI, R. et al. ACCESS: Assurance Case Centric Engineering of Safety-critical Systems. Journal of Systems and Software, v. 213, art. 112034, 2024. DOI: https://doi.org/10.1016/j.jss.2024.112034.

MICROSOFT. coderec_programming_states: Code and Data for Reading Between the Lines and When to Show a Suggestion. GitHub repository. URL: https://github.com/microsoft/coderec_programming_states. Acesso em: 6 set. 2026.

MOZANNAR, H.; BANSAL, G.; FOURNEY, A.; HORVITZ, E. Reading Between the Lines: Modeling User Behavior and Costs in AI-Assisted Programming. Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, 2024. DOI: https://doi.org/10.1145/3613904.3641936.

ZENODO. LLM-Generated Software Requirements from GitHub Issues. Version 2.1.1. Dataset, 2026. DOI: https://doi.org/10.5281/zenodo.19520570. Acesso em: 6 set. 2026.