VIII ENCONTRO INTERNACIONAL DE GESTÃO, DESENVOLVIMENTO E INOVAÇÃO
NOVOS CONTEXTOS PARA O AGRONEGÓCIO
ISSN: 2594-8083
VIII EIGEDIN
25 a 27 de Maio de 2026
Totalmente Online e Gratuito

( ) Graduação (X) Pós-Graduação
(X) Artigo completo ( ) Relato de prática ( ) Resumo expandido

USO DA INTELIGÊNCIA ARTIFICIAL NA OTIMIZAÇÃO DO GERENCIAMENTO DO *PRODUCT BACKLOG* EM PROJETOS ÁGEIS

Sidney Comandulli
Universidade Tecnológica Federal do Paraná
sidney.comandulli@alunos.utfpr.edu.br

Andressa Araújo
Universidade Tecnológica Federal do Paraná
andressa.araujo@alunos.utfpr.edu.br

Kauterine de Lima Dallanol
Universidade Tecnológica Federal do Paraná
kauterinelimadallanol@alunos.utfpr.edu.br

Orlando Oliveira Rodrigues
Universidade Tecnológica Federal do Paraná
orlandorodrigues@alunos.utfpr.edu.br

Sérgio Eduardo Gouvêa da Costa
Universidade Tecnológica Federal do Paraná
gouvea@utfpr.edu.br

RESUMO

Este artigo propõe e avalia uma metodologia baseada em modelos de linguagem de larga escala (LLMs), com foco no aprimoramento da qualidade dos itens do *Product Backlog* em ambientes ágeis em escala. Diante dos desafios recorrentes na engenharia de requisitos ágil, como baixa padronização, ambiguidade e ausência de critérios objetivos, desenvolveu-se um *framework* técnico utilizando o modelo GPT-4o da *OpenAI*. A solução automatizada foi aplicada em uma indústria de óleo e gás, envolvendo times ágeis reais, com a finalidade de diagnosticar e melhorar, por meio de Inteligência Artificial generativa, a clareza, completude e testabilidade de artefatos como *User Stories*, Critérios de Aceitação, *Definition of Ready e Definition of Done*. A metodologia seguiu abordagem exploratória, combinando técnicas quantitativas e qualitativas, estruturadas em cinco etapas: levantamento bibliográfico, aplicação de questionário, coleta de dados, desenvolvimento da solução e análise de resultados. Os resultados demonstraram que o modelo foi capaz de oferecer diagnósticos objetivos e recomendações contextualizadas, contribuindo para o aumento da maturidade ágil e promovendo ganhos em padronização, alinhamento entre equipes e previsibilidade das entregas. Conclui-se que a integração de LLMs à governança ágil representa uma abordagem viável, escalável e com alto potencial de impacto organizacional.

Palavras-chave: *Product Backlog*. Inteligência Artificial. Métodos Ágeis.

ISSN: 2594-8083

1 INTRODUÇÃO

O desenvolvimento ágil de *software* consolidou-se como abordagem predominante na indústria, graças à sua flexibilidade, capacidade de resposta rápida às mudanças e promoção de colaboração entre equipes (ALMEIDA & CARNEIRO, 2023). O *Scrum*, em especial, destaca-se como o framework mais adotado, de acordo com o 16º *State of Agile Report* (DIGITAL.AI, 2021), o que evidencia seu papel central na entrega contínua de valor e no alinhamento com as necessidades dos usuários através de feedbacks iterativos (MEILIANA et al., 2023).

Paralelamente a essa evolução, a complexidade dos produtos digitais tem aumentado exponencialmente, especialmente em contextos como manufatura, logística e *supply chain*, que exigem mecanismos mais sofisticados para planejamento, comunicação entre equipes, gerenciamento de requisitos e gestão de *backlogs* (JAYARAMAN et al., 2024). Assim, a digitalização tem impactado profundamente os processos de negócio, demandando novas abordagens para lidar com grandes volumes de dados e relações complexas (DORNBERGER et al., 2018). Nesse cenário, em ambientes que adotam métodos ágeis em escala, como *SAFe*, *LeSS* e *Nexus*, a gestão eficiente do *Product Backlog*, definido como uma lista ordenada de tudo o que é conhecido como necessário no produto (SEDANO et al., 2019), torna-se essencial para garantir que múltiplas equipes trabalhem de forma coordenada e alinhada com a visão do produto (ALSALEMI, 2017).

Neste contexto de crescente complexidade e volume de informações, a Inteligência Artificial (IA), e em especial os *Large Language Models* (LLMs), emergem como uma tecnologia disruptiva com potencial para transformar profundamente a engenharia de requisitos e a gestão de *backlog*, uma vez que modelos baseados em *Machine Learning* (ML) e *Natural Language Processing* (NLP) já demonstram capacidade de capturar relações não triviais entre elementos de software, além de automatizar tarefas cognitivamente intensivas, como o agrupamento de requisitos e a extração de padrões a partir de dados não estruturados (JAYARAMAN et al., 2024; MEILIANA et al., 2023; NASIRI & LAHMER, 2024). Desse modo, a IA generativa, em particular, permite que *Product Owners* (POs) deleguem à esses modelos de LLMs parte das atividades de documentação e detalhamento de requisitos, como a formulação de histórias de usuário, critérios de aceitação e wireframes iniciais (DIEBOLD, 2024). Esse 3 deslocamento do foco operacional para o estratégico proporciona maior capacidade de tomada de decisão em níveis táticos e arquiteturais (DIEBOLD, 2024). Além

ISSN: 2594-8083

disso, LLMs podem ser treinados para sugerir melhorias nos itens do *backlog* com base em métricas de valor, histórico de *sprint*, dados de incidentes e *feedbacks* de *stakeholders*, contribuindo para uma gestão mais informada e contínua (NASIRI & LAHMER, 2024).

Apesar da crescente adoção de métodos ágeis e do potencial evidente da IA, há uma carência notável de abordagens sistemáticas que integrem a IA para suportar a gestão dinâmica e estratégica do *backlog* (DILORENZO et al., 2020). Tradicionalmente, a engenharia de requisitos no *Scrum* é caracterizada por sua informalidade e forte dependência do conhecimento e das habilidades individuais da equipe, o que frequentemente resulta em histórias de usuário ambíguas, incompletas ou redundantes (ALHAZMI & HUANG, 2020). Assim, a elevada volatilidade dos requisitos, uma característica inerente aos ambientes ágeis, representa um fator de risco significativo, com impacto direto no cumprimento dos prazos e no controle dos custos do projeto (ALHAZMI & HUANG, 2020; SEDANO et al., 2019). Soma-se a isso a dificuldade recorrente em se alcançar uma compreensão plena e compartilhada das reais necessidades dos *stakeholders*, sobretudo quando se depende exclusivamente de histórias de usuário informais como principal instrumento de comunicação e documentação dos requisitos (SENFT et al., 2018).

A priorização do *Product Backlog*, responsabilidade central do PO na busca pela maximização do valor entregue pela equipe, frequentemente revela-se como uma tarefa complexa e limitada, muitas vezes baseada exclusivamente no valor de negócio, desconsiderando fatores críticos como dependências técnicas, riscos e restrições operacionais (SACHDEVA et al., 2019). Simultaneamente, a gestão de mudanças mostra-se fragilizada, uma vez que alterações nos requisitos são frequentemente registradas como itens novos, sem histórico de modificações, prejudicando a rastreabilidade (ALSALEMI, 2017). Além disso, a detecção de histórias de usuário duplicadas ou semanticamente semelhantes demanda tempo e esforço consideráveis, reduzindo a eficiência do *backlog* (NASIRI & LAHMER, 2024). Outro ponto crítico diz respeito aos atributos de qualidade (QAs), como desempenho, segurança e usabilidade, que embora essenciais para a arquitetura do sistema, são comumente implícitos nas histórias de usuário, dificultando seu 4 tratamento explícito e o devido planejamento (GILSON et al., 2019). Essa ausência de formalização gera um conflito constante entre a velocidade na entrega de funcionalidades e a construção de uma base técnica sustentável e de qualidade a longo prazo (SEDANO et al., 2019).

A estimação de esforço em *Scrum* também representa um gargalo com práticas pouco padronizadas e suscetíveis a vieses cognitivos, como ancoragem e excesso de confiança,

ISSN: 2594-8083

resultando em subestimação recorrentes (MEILIANA et al., 2023). Adicionalmente, a natureza da documentação mínima no *Scrum* tende a limitar a rastreabilidade dos requisitos ao longo do ciclo de desenvolvimento, evidenciando a ausência de abordagens metodológicas específicas que atendam às particularidades dessa metodologia (ALSALEMI, 2017). Dado esse contexto, as abordagens atuais para agrupamento automático de requisitos, como *clustering*, ainda apresentam baixa precisão e pouca automação, dificultando sua aplicação prática no contexto ágil (NASIRI & LAHMER, 2024). O *Product Backlog*, embora central ao *Scrum*, é um artefato informal e não representa uma especificação de requisitos estruturada, o que intensifica a ambiguidade e a subjetividade nas decisões (SEDANO et al., 2019). Com isso, a priorização e o refinamento permanecem altamente dependentes da interpretação manual dos *stakeholders*, muitas vezes guiados por critérios não sistematizados (GILSON et al., 2019). Ademais, a adoção prática de novas ferramentas baseadas em IA/ML/NLP também enfrentam barreiras, como a complexidade percebida e a necessidade de dados balanceados para o treinamento de modelos (ALHAZMI & HUANG, 2020).

Diante dos desafios enfrentados na gestão do *Product Backlog* em métodos ágeis em escala, especialmente no que tange à identificação, estruturação e refinamento de itens do backlog, este artigo tem como objetivo investigar o potencial uso das *Large Language Models* (LLMs) para aprimorar esses processos, por meio do desenvolvimento e da avaliação de um *framework* baseado no uso da Inteligência Artificial generativa aplicado ao contexto ágil. A proposta visa oferecer suporte à tomada de decisão de *Product Owners* e *Scrum Masters*, contribuindo para a avaliação da qualidade dos itens do *backlog* com base em critérios específicos, boas práticas de mercado e princípios dos manifestos ágeis. Espera-se, com isso, promover maior assertividade na análise comparativa de entregas entre equipes, identificar oportunidades de melhoria no planejamento e reduzir retrabalhos. Nesse sentido, a eficácia do método será inicialmente validada 5 com o desenvolvimento de um Produto Mínimo Viável (MVP), que permitirá mensurar a qualidade do backlog avaliado, além de registrar lições aprendidas e indicar caminhos para o aperfeiçoamento futuro do modelo.

2 REFERENCIAL TEÓRICO

2.1 METODOLOGIAS ÁGEIS E O PAPEL DO *PRODUCT BACKLOG*

As metodologias ágeis se consolidaram como uma abordagem fundamental para o desenvolvimento de *software* e gestão de projetos, especialmente por sua capacidade de promover flexibilidade, entregas contínuas e foco na satisfação do cliente (HIGHSMITH,

2002). Diferentemente dos modelos tradicionais, que seguem etapas rígidas e sequenciais, *frameworks* ágeis valorizam a adaptação rápida às mudanças e a colaboração constante entre equipes e *stakeholders*, promovendo maior inovação e agilidade no desenvolvimento (BECK et al., 2001).

Entre as diversas estratégias ágeis, o *Scrum* se destaca por ser um modelo estruturado que organiza o trabalho em ciclos curtos, denominados *sprints*, com entregas frequentes e funcionais do produto (SCHWABER & SUTHERLAND, 2017). O *Product Backlog*, artefato central do *Scrum*, é um elemento fundamental na gestão de projetos ágeis de *software*. Trata-se de uma lista dinâmica de itens de trabalho que pode incluir *user stories*, *bugs*, *chores*, entre outros, e é utilizada pelas equipes para coordenar e organizar as atividades a serem realizadas (SUTHERLAND & SCHWABER, 2013). Conforme destacado por (ALHAZMI & HUANG, 2020), o *Product Backlog* funciona como a fonte principal dos requisitos do usuário, sendo priorizado pelo *Product Owner* com base em critérios como prioridade do cliente, valor de negócio, dependências, custo, tempo e complexidade.

Uma vez concluído o processo de *Design Thinking*, os requisitos finais são transformados em user stories ou itens, que são então organizados no *backlog* conforme sua prioridade. O *Design Thinking*, ao integrar o foco na empatia e nas necessidades dos usuários, complementa a abordagem ágil, proporcionando uma base mais sólida para a criação de soluções inovadoras (BROWN, 2009).

Dessa forma, o *Product Owner* (PO) é o responsável pela gestão do *Product Backlog*, com o objetivo principal de maximizar o valor entregue pela equipe a cada *Sprint* (PICHLER, 2010). Para isso, é fundamental que o *backlog* esteja sempre bem refinado e atualizado, pois a falta de cuidado nesse aspecto pode dificultar o planejamento de *sprints*, a estimativa de esforços e a entrega consistente de valor. (NASIRI & LAHMER, 2024) reforçam que o refinamento contínuo do *backlog* é essencial para priorizar o trabalho, identificar e resolver problemas rapidamente, além de alinhar os esforços da equipe aos objetivos do projeto.

Em síntese, processos ágeis proporcionam uma abordagem dinâmica e centrada na entrega de valor contínuo, sendo o *Product Backlog* um elemento-chave nesse processo. Sua correta utilização garante que as necessidades do cliente sejam atendidas de forma interativa e colaborativa, promovendo maior alinhamento entre a equipe e os objetivos do projeto. O papel do *Product Owner*, ao manter a lista de desenvolvimentos bem priorizada e refinada, torna-se decisivo para o sucesso das entregas em ambientes complexos e sujeitos a constantes mudanças, reforçando a importância de uma gestão estratégica dos requisitos no contexto ágil

ISSN: 2594-8083

(SUTHERLAND & SCHWABER, 2017).

2.2 HISTÓRIAS DE USUÁRIO E ENGENHARIA DE REQUISITOS ÁGIL

A Engenharia de Requisitos (RE) é uma disciplina essencial e vital no desenvolvimento de *software*. De modo geral, o processo de engenharia de requisitos é composto por atividades como elicitação, análise, triagem, especificação e verificação (SENFT, FISCHER, OBERTHUR & PATKAR, 2018). Essas atividades são fundamentais para garantir que o produto final atenda às expectativas do cliente e às necessidades do mercado (HIGHTOWER et al., 2019).

Nesse cenário de desenvolvimento ágil, (ALHAZMI & HUANG, 2020) afirmam que a engenharia de requisitos é realizada de forma iterativa ao longo do processo de desenvolvimento, em vez de ser uma fase fechada no início. Essa perspectiva é baseada nos estudos de (SCHÖN, THOMASCHEWSKI & ESCALONA, 2017), que ressaltam a flexibilidade das metodologias ágeis na adaptação dos requisitos ao longo do projeto. É nesse contexto que as histórias de usuário (user stories) surgem como uma forma prática e simples de registrar esses requisitos. Segundo (CHANG et al., 2016), as user stories têm se mostrado uma ferramenta eficaz para documentar de maneira enxuta os requisitos do usuário.

De acordo com (COHN, 2004), uma *user story* é uma caracterização de requisito centrada no cliente, contendo apenas as informações necessárias para que os desenvolvedores do projeto compreendam claramente o que precisa ser implementado. Elas servem como o principal ativo para armazenar o conhecimento do produto em projetos de Desenvolvimento de *Software* Ágil. *User stories* bem definidas são uma das chaves para o sucesso na entrega ágil de *software* (GROSS et al., 2017).

Antigamente, esses requisitos eram documentados de maneira extensa e detalhada, muitas vezes em formatos complexos que dificultavam a compreensão pelos não especialistas. (GILSON & IRWIN, 2018) afirmam que *user stories* mal definidas podem acarretar um aumento significativo do trabalho devido à documentação incompleta ou incorreta, evidenciando a importância de uma definição clara desde o início. Nesse sentido, requisitos incorretos ou incompletos também podem gerar retrabalho custoso, conforme destacado por (SENFT, FISCHER, OBERTHUR & PATKAR, 2018), que alertam sobre os custos adicionais gerados por requisitos imprecisos e mal comunicados.

Para mitigar esses problemas, (ALHAZMI & HUANG, 2020) sugerem a integração do *Design Thinking* no *Scrum*, visando melhorar a gestão da engenharia de requisitos ao focar em

ISSN: 2594-8083

uma compreensão abrangente das necessidades do cliente. A combinação dessas abordagens tem se mostrado eficaz para alinhar as necessidades do cliente com as capacidades da equipe, o que, segundo (KIM et al. 2019), favorece a criação de soluções mais inovadoras e direcionadas ao real valor do cliente.

Normalmente, as *user stories* seguem um formato estruturado que enfatiza "quem, o quê e por quê", conforme o template clássico: "Como um, eu quero, para que" (COHN, 2004). Esse tipo de descrição ajuda a equipe a entender claramente o que o sistema precisa fazer e o valor daquela funcionalidade para o usuário. De acordo com (LUCAS et al., 2018), esse formato proporciona clareza e foco, permitindo que a equipe entenda rapidamente os objetivos e os benefícios das funcionalidades a serem implementadas.

Em resumo, as user stories facilitam a comunicação e o alinhamento entre a equipe e o cliente, garantindo que os requisitos sejam claros e focados no valor entregue. Combinadas a práticas como o *Design Thinking*, elas tornam a engenharia de requisitos mais eficiente, promovendo entregas ágeis, redução de retrabalho e maior satisfação do cliente.

2.3 INTELIGÊNCIA ARTIFICIAL APLICADA A DESENVOLVIMENTO DE *SOFTWARE*

A Inteligência Artificial (IA) tem ganhado destaque no desenvolvimento de *software*, trazendo uma verdadeira revolução na maneira como as equipes operam. Ferramentas baseadas em IA têm o potencial de automatizar tarefas repetitivas, como a detecção de *bugs*, análise de código e testes de *software*. Isso permite que os desenvolvedores se concentrem em atividades de maior valor estratégico, como a definição de requisitos e a criação de novas funcionalidades. De acordo com (SANTOS, 2023), a IA aplicada a ferramentas de teste pode antecipar falhas no sistema antes da implementação de novas funcionalidades, reduzindo significativamente os custos de manutenção e retrabalho.

Uma das maiores vantagens da IA no desenvolvimento de *software* é sua capacidade de otimizar a qualidade do código e reduzir erros. Ferramentas de análise automatizada podem identificar falhas de segurança e inconsistências no código de maneira mais eficiente do que abordagens tradicionais. (LÓPEZ et al. , 2022) ressaltam que, ao aplicar *machine learning*, essas ferramentas aprendem com erros passados, melhorando constantemente a detecção de problemas e aumentando a cobertura dos testes. Ademais, a IA pode sugerir otimizações no código, garantindo maior robustez e segurança ao produto final (SCHWAB et al., 2018).

No desenvolvimento ágil, a IA também desempenha um papel crucial na gestão e

priorização das atividades. Com algoritmos preditivos, é possível analisar o inventário de requisitos e identificar itens de maior valor estratégico. Segundo (NASIRI E LAHMER, 2024), a IA ajuda a otimizar o processo de refinamento do *backlog*, priorizando as funcionalidades mais importantes. Isso não só melhora a alocação de recursos, mas também permite que as equipes ajustem suas prioridades conforme o andamento do projeto, promovendo maior transparência e redução da dependência de decisões manuais (GROSSO et al., 2021).

Além da melhoria na gestão de tarefas, a IA também tem sido utilizada para personalizar a experiência do usuário. Modelos como *SBERT* e *Word2Vec*, empregados em Processamento de Linguagem Natural (PLN), analisam dados de interação do usuário e ajustam as funcionalidades do sistema com base nas preferências individuais. Isso resulta em interfaces mais adaptativas, elevando a satisfação do usuário final. (GILSON & IRWIN, 2018) destacam que essas técnicas de PLN ajudam as equipes a entender melhor os requisitos do usuário e antecipar suas necessidades, permitindo que o *software* se ajuste dinamicamente ao comportamento do usuário (ALMEIDA & CARNEIRO, 2023).

Por fim, a integração da IA no desenvolvimento de *software* não apenas melhora a qualidade e a gestão do projeto, mas também facilita a inovação em novas funcionalidades. A automação de tarefas como a geração de documentação e o acompanhamento de progresso têm sido aprimoradas por IA, proporcionando mais eficiência. De acordo com (DI LORENZO et al., 2020), o Modelo de Aceitação da Tecnologia (TAM) é frequentemente utilizado para avaliar a aceitação dessas ferramentas de IA, garantindo que elas agreguem valor real às equipes de desenvolvimento. As inovações tecnológicas, portanto, não só otimizam processos, mas também se ajustam às dinâmicas de trabalho, garantindo que as soluções sejam eficazes e bem aceitas.

2.4 MÉTRICAS DE AVALIAÇÃO EM PROJETOS ÁGEIS E VALIDAÇÃO DE SOLUÇÕES COM IA

As métricas são instrumentos necessários para medir o desempenho e a qualidade dos processos utilizados no desenvolvimento de *software* e na gestão de projetos. Elas auxiliam as equipes a compreender se os objetivos estão sendo alcançados e onde há oportunidades de melhoria. No contexto da indústria de *software*, as métricas permitem planejar e monitorar projetos, avaliar a qualidade dos produtos entregues e aprimorar a comunicação e o uso de ferramentas no desenvolvimento (STARON et al., 2016; FOWLER, 2019).

Com o avanço da Inteligência Artificial (IA), o uso de indicadores se estende também à

ISSN: 2594-8083

avaliação do desempenho de modelos computacionais. Indicadores como acurácia, que mede a porcentagem de acertos, e F1-*score*, que combina precisão e sensibilidade, são amplamente utilizados para verificar a eficiência dos modelos. Outro aspecto importante é o tempo de execução das tarefas, que impacta diretamente nos custos operacionais. Nesse sentido, estudos apontam que é possível reduzir esses custos em até 98% com perdas mínimas de acurácia (JAYARAMAN, AZAR & MAALOUF, 2024; WANG et al., 2021).

Além do desempenho dos sistemas, outras métricas são essenciais na gestão da qualidade do *software*, como a confiabilidade, que avalia o quanto o sistema opera sem apresentar falhas, e a facilidade de manutenção (LÓPEZ et al., 10 2022; SILVA et al., 2020). A experiência da equipe também exerce influência na valorização dessas métricas, especialmente em aspectos como o desempenho da equipe, o valor do trabalho a ser realizado e a automação de testes (ALMEIDA & CARNEIRO, 2023).

Entre as métricas de maior interesse nas práticas ágeis está a porcentagem de automação de testes, visto que a automação contribui para ampliar a cobertura de testes, garantir consistência nas execuções, reduzir erros manuais e fornecer *feedbacks* mais rápidos e confiáveis. A ausência desse recurso, por outro lado, pode comprometer a qualidade do software ao longo do tempo (ALMEIDA & CARNEIRO, 2023; RIBEIRO et al., 2021).

Além disso, metodologias ágeis como o *Scrum* utilizam parâmetros específicos para apoiar a gestão e a tomada de decisões. O *Lead Time* mede o tempo total para concluir uma tarefa, o *Cycle Time* registra o tempo dedicado a cada atividade e a *Velocity* indica a média de trabalho entregue por *sprint*. Já gráficos como o *Burndown*, *Burnup* e o *Cumulative Flow Diagram* ajudam a visualizar o progresso das entregas e a identificar atrasos ou gargalos no processo (SCHWAB et al., 2018).

Para que essas soluções baseadas em IA sejam efetivamente incorporadas às rotinas de trabalho, é fundamental validar sua usabilidade e o valor percebido pelas equipes. O Modelo de Aceitação da Tecnologia (TAM) é uma das abordagens que permite avaliar a aceitação e utilidade percebida dessas ferramentas (DILORENZO et al., 2020; VANDERMEER et al., 2022).

Em síntese, o uso integrado de métricas e Inteligência Artificial tem transformado significativamente a forma como os projetos de *software* são planejados, executados e validados, promovendo maior eficiência, adaptabilidade e alinhamento com as necessidades reais dos clientes (SANTOS, 2023).

ISSN: 2594-8083

3. METODOLOGIA

3.1 CARACTERIZAÇÃO DA PESQUISA

Os procedimentos metodológicos empregados neste estudo colaboraram diretamente para o alcance dos objetivos propostos, em busca de diagnosticar de maneira automatizada e estruturada a qualidade de histórias de usuário em backlogs de equipes ágeis. Do ponto de vista da natureza, a pesquisa pode ser classificada como aplicada, uma vez que a pesquisa aplicada foi dedicada à geração de conhecimento para solução de problemas específicos, foi dirigida à busca da verdade para determinada aplicação prática em busca de apresentar soluções para determinadas questões organizacionais (NASCIMENTO, 2023).

Sob a perspectiva dos métodos ou abordagens metodológicas, ainda de acordo com Nascimento (2023) a pesquisa obedeceu a dois métodos, ou à conjugação de ambos: quantitativo e qualitativo. Tais abordagens e métodos têm características diferentes, mas carregam caráter complementar, não excludente. Creswell e Creswell (2021) citam que o método quantitativo envolve o processo de coleta, análise, interpretação e redação dos resultados de um estudo enquanto a abordagem do método qualitativo é uma pesquisa por estratégia de investigação e sua análise é interpretativa, amparada pela vivência com seus participantes.

Ao considerar-se o objetivo geral como medida, a presente pesquisa configurou-se como exploratória. A pesquisa de caráter exploratório teve como objetivo proporcionar maior familiaridade com o problema, com vistas a torná-lo mais explícito ou a construir hipóteses, a grande maioria dessas pesquisas envolve levantamento bibliográfico, entrevistas com pessoas que tiveram experiências práticas com o problema pesquisado ou análise de exemplos que estimulem a compreensão (GIL, 2022).

3.2 DELIMITAÇÃO DA PESQUISA

A pesquisa foi aplicada a uma área específica de uma indústria de óleo e gás, tendo como público-alvo o time de desenvolvimento, incluindo profissionais como *Agile Managers* e *Product Owners*. Essa delimitação visou garantir que a coleta de dados fosse realizada em um contexto real e relevante para a aplicação de metodologias ágeis em escala, onde a gestão do *Product Backlog* torna-se essencial para garantir a coordenação e alinhamento das equipes com a visão do produto. A escolha de uma indústria de óleo e gás permitiu explorar os desafios em um ambiente corporativo com múltiplos times e demandas voláteis, que ampliam a complexidade na gestão do *Product Backlog*.

ISSN: 2594-8083

A proposta baseou-se na aplicação de modelos de linguagem de grande escala (LLMs), mais especificamente o *GPT-4o* da *OpenAI*, combinando técnicas avançadas de processamento de linguagem natural com boas práticas da engenharia de requisitos ágeis, tais como *Definition of Ready (DoR), Definition of Done (DoD)* e Critérios de Aceitação (CA). A abordagem busca suprir lacunas recorrentes na documentação, clareza e padronização das histórias de usuário, aspectos frequentemente negligenciados e que impactam diretamente a qualidade das entregas em ambientes *Scrum, Kanban* ou *SAFel*.

3.3 ETAPAS DA PESQUISA

A figura 1 abaixo ilustra as etapas da pesquisa e descreve os passos necessários para o alcance do objetivo da pesquisa, em busca de diagnosticar de maneira automatizada e estruturada a qualidade de histórias de usuário em backlogs de equipes ágeis, aliada à Inteligência Artificial generativa. Nos tópicos a seguir, as etapas para a realização da pesquisa são descritas:

Figura 1: Framework metodológico da pesquisa

Fonte: Elaborado pelos autores (2025).

3.3.1 ETAPA 1 - LEVANTAMENTO BIBLIOGRÁFICO

Considerando as palavras-chave concernentes ao estudo, realizou-se buscas na base de dados Scopus, de estudos aplicados relacionados à linha de pesquisa, que permitiram maior aprofundamento quanto ao conhecimento sobre os conceitos relativos ao tema proposto. Inicialmente, foram definidos os critérios de busca utilizando "TITLE-ABS-Key: *Product Backlog*" para abranger artigos relevantes sobre o tema. Após a aplicação dos critérios, os dados da pesquisa foram exportados em formato CSV.

Nesta etapa, os 186 artigos inicialmente encontrados foram processados. Primeiramente, todos os registros foram gravados em um banco de dados. Em seguida, foram removidos os artigos com DOI duplicados, resultando em 156 artigos. Os artigos foram então filtrados para o período de 2018 a 2025, resultando em 113 artigos. As palavras-chave dos artigos foram analisadas e agrupadas por frequência para identificar os principais eixos. Estes eixos foram categorizados como Inteligência Artificial (IA), Qualidade, Produtividade e Outros. Em seguida, os artigos classificados na categoria "Outros" foram removidos, resultando em 66 artigos. Para realizar uma análise mais aprofundada, uma planilha foi estruturada com as

ISSN: 2594-8083

informações e categorias dos artigos selecionados.

Dessa forma, a análise dos artigos seguiu critérios específicos, como alinhamento com os objetivos do estudo, conceitos fundamentais de metodologias ágeis e gerenciamento de product backlog, desafios e oportunidades no gerenciamento de *product backlog*, e o uso de IA na otimização de processos e gerenciamento de *product backlog*.

Como resultado, 32 artigos foram delimitados por categoria. As informações foram analisadas com base nos critérios estabelecidos, examinando título, palavras-chave e resumo, o que reduziu o número para 24 artigos. Os artigos selecionados foram então reunidos. Posteriormente, artigos com PDF indisponível foram removidos, resultando em 19 artigos, que compuseram a base final de artigos para a análise completa.

3.3.2 ETAPA 2 - APLICAÇÃO DE QUESTIONÁRIO COM PROFISSIONAIS

A partir do levantamento bibliográfico realizado, esta pesquisa incorporou uma abordagem investigativa com objetivo de compreender as percepções, desafios e expectativas de profissionais atuantes em ambientes ágeis em relação ao uso da Inteligência Artificial na gestão do *backlog*. Foi conduzido um estudo com *Agile Masters* e *Product Owners* da área de uma área de negócio da empresa avaliada, onde aplicou-se um conjunto de 12 (doze) questões organizadas em 04 (quatro) categorias temáticas: engenharia de requisitos, gestão e refinamento do backlog, estimativas e planejamento, e adoção de tecnologias de automação.

A análise desses dados, de natureza qualitativa e quantitativa, buscou validar a aderência da solução proposta às necessidades reais dos times, além de fornecer subsídios para o aprimoramento do sistema. Foram identificadas preocupações comuns quanto à dificuldade em manter critérios de aceitação objetivos, à falta de rastreabilidade nas histórias e à escassez de tempo durante o refinamento. As respostas evidenciaram uma abertura à utilização de ferramentas de apoio baseadas em IA, desde que estas fossem integradas de forma natural ao fluxo de trabalho e oferecessem sugestões contextualizadas e acionáveis.

Para dar sustentação ao processo de diagnóstico, foi desenvolvida uma escala de pontuação de 0 a 5, que orienta o modelo de linguagem na classificação dos elementos analisados com base em níveis crescentes de maturidade e aderência às boas práticas ágeis. A seguir, a tabela apresenta os significados de cada pontuação, utilizada de maneira uniforme para todos os critérios avaliados.

ISSN: 2594-8083

Tabela 1: Descrição da pontuação utilizada

Nota	Significado
0	Inexistente ou irrelevante
1	Muito superficial (cita apenas um aspecto, de forma vaga)
2	Poucos aspectos mencionados, sem objetividade
3	Alguns aspectos abordados, linguagem clara, mas sem verificação técnica
4	Abrangente e claro, com evidência parcial de verificação técnica
5	Abrangente, claro, mensurável e tecnicamente verificável

Fonte: Elaborado pelos autores (2025).

A aplicação da escala se deu sobre quatro categorias distintas de análise por item de backlog: a descrição da história de usuário (*User Story*), os critérios de aceitação (CA), a *Definition of Ready (DoR)* e a *Definition of Done (DoD)*. Para cada uma delas, foram definidos critérios específicos que guiaram o modelo de linguagem na atribuição da nota. Esses critérios foram estruturados com base na literatura especializada (COHN, 2004; SCHWABER & SUTHERLAND, 2020) e ajustados ao longo da experimentação inicial com o GPT-4o. A seguir, na tabela X são detalhados os critérios considerados para cada categoria de avaliação.

Tabela 2: Critérios de avaliação para cada elemento do *backlog*

Elemento Avaliado	Critérios Considerados
User Story	Clareza da redação; uso do formato padrão (“Como [persona], quero [ação], para [benefício]”); foco no valor ao usuário; tamanho adequado para uma sprint.
Critérios de Aceitação	Alinhamento com a User Story; clareza e compreensibilidade; testabilidade (preferência por formato “Dado / Quando / Então”); abrangência de casos positivos e negativos.
Definition of Ready (DoR)	Existência de critérios que garantam que o item está pronto para desenvolvimento; clareza nos requisitos; estimativas; dependências identificadas; alinhamento

	com o PO.
Definition of Done (DoD)	Presença de elementos como testes, revisão de código, documentação, critérios de qualidade; clareza sobre o que configura uma entrega completa e verificável.

Fonte: Elaborado pelos autores (2025)

Durante o processamento, o modelo era instruído a atribuir notas com base nesses critérios e, adicionalmente, a justificar suas decisões por meio de comentários explicativos e sugestões de melhoria. Cada resposta continha também exemplos ideais para fins comparativos, o que reforçou a natureza educativa da abordagem. O uso combinado da pontuação e das justificativas permitiu que a análise fosse ao mesmo tempo objetiva e interpretável. A triangulação entre os resultados automatizados e os depoimentos dos especialistas gerou uma visão abrangente e fundamentada, revelando não apenas a viabilidade técnica da ferramenta, mas também sua compatibilidade com a realidade das equipes analisadas. Essa análise revelou que, para além do diagnóstico numérico, os times valorizavam sobretudo a explicação oferecida pela IA, pois ela se apresentava como base concreta para a reescrita ou melhoria das histórias analisadas.

Essa abordagem validou o potencial da metodologia como mecanismo de apoio à maturidade ágil, ao mesmo tempo em que forneceu subsídios relevantes para o aprimoramento contínuo dos *prompts* e da lógica de avaliação. A partir dos dados coletados, foram sugeridas adaptações futuras, como a contextualização por tipo de time, fase do produto e histórico de avaliações anteriores, de forma a personalizar ainda mais as sugestões fornecidas pela IA.

3.3.3 ETAPA 3 - LEVANTAMENTO DE DADOS

A base para a avaliação automatizada da qualidade das histórias de usuário consiste na leitura de arquivos estruturados (nos formatos .csv ou .xlsx) contendo user *stories* reais, extraídos de diferentes squads, compostas por campos como descrição da história (*StoryDesc*), Critérios de Aceitação, *DoR* e *DoD*.

Adicionalmente, o modelo de aplicação da pesquisa prevê a utilização de transcrições de reuniões de alinhamento como entrada para o processamento de dados não estruturados.

3.3.4 ETAPA 4 - DESENVOLVIMENTO DA SOLUÇÃO AUTOMATIZADA

A solução automatizada foi desenvolvida com base em princípios de LLMOps,

ISSN: 2594-8083

adotando uma estrutura modular em linguagem *Python* e organizando o sistema em etapas bem definidas que vão desde a ingestão dos dados até a interpretação e visualização dos resultados. A Figura 2 a seguir ilustra, de forma geral e estrutural, a organização modular do sistema proposto.

Figura 2: Visão da estrutura de módulos do sistema

```
BacklogAnalyzer/
├── data/                    # Contém os arquivos CSV com as histórias de backlog
├── outputs/                 # Resultados gerados em JSON para consumo em BI
├── src/                     # Códigos-fonte do projeto
│   ├── main.py              # Script principal de execução do pipeline
│   ├── backlog_processor.py # Lógica de leitura e preparação do CSV
│   ├── prompt_builder.py    # Construtor do prompt dinâmico para a LLM
│   ├── openai_client.py     # Módulo de comunicação com a API OpenAI
│   └── analyzer.py          # Interpretação e estruturação da resposta em JSON
├── requirements.txt         # Lista de dependências
```

Fonte: Elaborado pelos autores (2025).

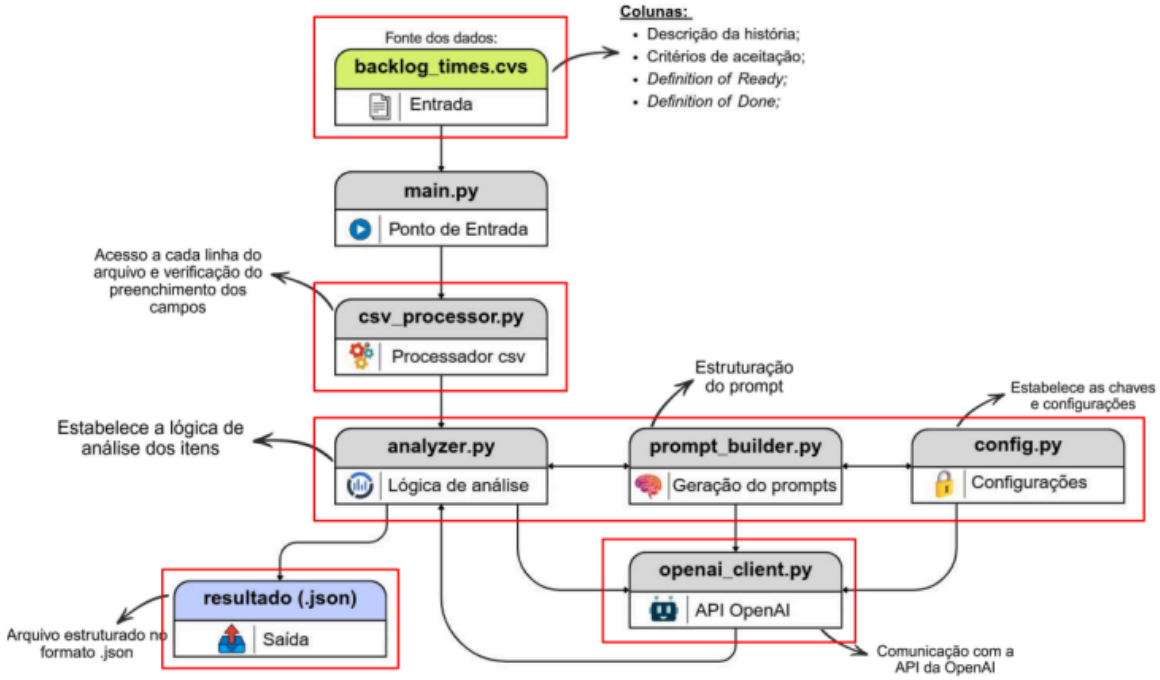
Esses dados passam por uma etapa de pré-processamento, onde são tratados valores ausentes, padronizados os nomes das colunas e normalizadas as informações textuais. A seguir, o sistema inicia o processo de engenharia de prompt, no qual um módulo específico é responsável por construir uma solicitação textual orientada ao modelo de linguagem. Esse prompt explicita o papel da IA como avaliadora especialista em práticas ágeis e contém instruções claras para que o modelo avalie criticamente quatro dimensões da história: a qualidade da descrição da *user story*, a testabilidade dos critérios de aceitação, a completude da definição de pronto e a robustez da definição de finalizado.

As instruções fornecidas ao modelo incluem critérios bem definidos de avaliação, uma escala de pontuação que varia de zero (ausência total do critério) até cinco (critérios claros, mensuráveis e tecnicamente verificáveis), além da exigência de que a resposta seja estruturada em JSON. Cada resposta retornada pelo modelo contém, para cada item avaliado, uma análise textual, uma recomendação prática de melhoria, um exemplo idealizado e a nota atribuída.

Essa interação com o modelo GPT-4o é realizada por meio de uma API segura, com tratamento de exceções, limites de tokens e reenvio automático de requisições incompletas. A resposta recebida é interpretada por outro módulo, que transforma o conteúdo JSON em uma estrutura de dicionário *Python*, atribuindo os dados à entrada correspondente. Caso algum critério esteja ausente ou não tenha sido avaliado corretamente, o sistema atribui uma resposta padrão, garantindo a completude dos registros.

Ao final do processamento, os resultados são exportados em arquivos JSON com carimbo de data e hora, sendo estes consumidos por painéis interativos desenvolvidos em ferramentas de *Business Intelligence* (como *Power BI* ou *Streamlit*). Esses painéis permitem que os times visualizem a qualidade das suas histórias, comparem desempenhos ao longo das sprints, identifiquem padrões de falhas recorrentes e implementem melhorias com base em evidências objetivas. Um resumo geral das etapas funcionais do modelo é apresentado na Figura 3 a seguir.

Figura 3: Visão das etapas funcionais do modelo



Fonte: Elaborado pelos autores (2025).

3.3.5 ETAPA 5 - VALIDAÇÃO E ANÁLISE DOS RESULTADOS

Ao final do processo, a análise dos resultados envolveu a validação da aderência da ferramenta às expectativas dos usuários, o refinamento dos *prompts* utilizados, a identificação de oportunidades de melhoria e o suporte à adoção mais consciente e informada de soluções baseadas em IA no ciclo de vida ágil. Dessa forma, o projeto se propõe a ir além da automação do diagnóstico, promovendo um verdadeiro diálogo entre tecnologia e prática profissional, e contribuindo para a construção de um backlog mais sólido, transparente e estratégico.

A metodologia proposta representa uma inovação significativa no campo da engenharia de requisitos ágil, ao transformar o *backlog*, historicamente tratado como artefato textual estático, em objeto de análise dinâmica, contínua e auditável. Por meio da Inteligência Artificial generativa, foi possível oferecer não apenas diagnósticos estruturados e imparciais, mas

ISSN: 2594-8083

também recomendações precisas que elevam o padrão de escrita e documentação das histórias.

Ao incorporar um mecanismo de *feedback* automatizado, a solução permite ganhos expressivos em tempo, padronização e qualidade. Equipes que tradicionalmente gastam horas discutindo subjetivamente a clareza de uma *user story* passam a contar com uma referência técnica embasada, promovendo maior alinhamento interno e aumentando a previsibilidade das entregas. A integração com dashboards de BI amplia ainda mais esse impacto, ao oferecer visibilidade gerencial sobre a maturidade do *backlog* em tempo real. Assim, mais do que automatizar uma tarefa, a proposta aqui descrita inaugura um novo paradigma na governança ágil, em que a Inteligência Artificial atua como agente orientador e parceiro na tomada de decisões.

4. APRESENTAÇÃO DOS RESULTADOS E DISCUSSÕES

A interpretação dos resultados obtidos com a aplicação da metodologia automatizada revelou padrões significativos de qualidade no *backlog* das equipes avaliadas, ao mesmo tempo em que destacou lacunas recorrentes na estruturação dos artefatos ágeis. A análise das pontuações atribuídas pelo modelo GPT-4o, associada às justificativas qualitativas fornecidas em cada resposta, permitiu uma leitura refinada sobre o grau de maturidade de cada squad no uso de boas práticas na construção de histórias de usuário.

Um primeiro aspecto observado refere-se à grande variabilidade nas notas atribuídas aos diferentes critérios, mesmo dentro de uma mesma equipe. Em muitos casos, a história de usuário apresentou uma descrição clara e bem estruturada, enquanto os critérios de aceitação estavam ausentes ou redigidos de maneira genérica. Isso sugere que, embora haja uma conscientização crescente sobre a importância da clareza na formulação da *user story*, os critérios que sustentam sua testabilidade e completude ainda não estão plenamente incorporados à rotina das equipes. Assim, a Figura 4 a seguir apresenta um resumo consolidado das notas atribuídas às equipes ágeis após a avaliação automatizada.

ISSN: 2594-8083

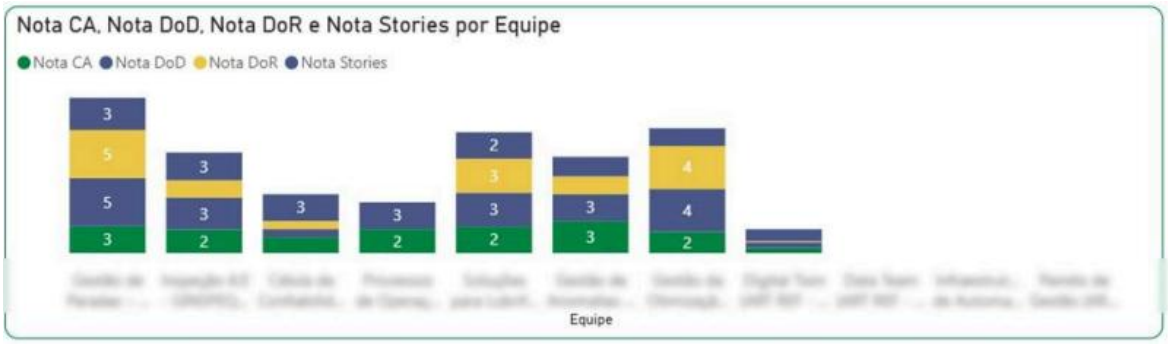
Figura 4: Notas obtidas pelos times ágeis

Equipe	Qtde Stories	Nota Geral	Nota Stories	Nota CA	Nota DoD	Nota DoR
[redacted]	2	14,50	3,00	2,50	4,50	4,50
[redacted]	3	11,67	1,67	2,00	4,00	4,00
[redacted]	17	11,29	2,47	2,47	3,18	3,18
[redacted]	5	9,40	2,60	2,20	3,00	1,60
[redacted]	6	9,00	1,83	3,00	2,50	1,67
[redacted]	4	5,50	2,50	1,50	0,75	0,75
[redacted]	4	4,75	2,50	2,25	0,00	0,00
[redacted]	34	2,24	1,12	0,50	0,53	0,09
[redacted]	1	0,00	0,00	0,00	0,00	0,00
[redacted]	9	0,00	0,00	0,00	0,00	0,00
[redacted]	2	0,00	0,00	0,00	0,00	0,00
Total	**87**	**5,45**	**1,55**	**1,31**	**1,45**	**1,14**

Fonte: Elaborado pelos autores (2025).

Complementarmente, a Figura 5 apresenta as notas organizadas graficamente, permitindo uma visualização comparativa do desempenho dos times em cada dimensão.

Figura 5: Gráfico com as notas



Fonte: Elaborado pelos autores (2025).

As notas atribuídas à *Definition of Ready* revelaram uma tendência à ausência de critérios objetivos sobre o estado de preparação do item para desenvolvimento. Itens muitas vezes seguiam diretamente para planejamento mesmo sem validação prévia de dependências, estimativas ou alinhamento com o *Product Owner*. Essa constatação reforça a importância de mecanismos que reforcem a cultura de preparação e análise crítica antes do início da sprint.

Já no tocante à *Definition of Done*, houve um padrão semelhante de inconsistência. Enquanto algumas equipes apresentavam critérios bem definidos que cobriam aspectos como testes, revisão de código e documentação, outras limitavam-se a afirmar que o item estaria "pronto quando concluído", sem qualquer evidência verificável. O modelo, ao apontar essas falhas com justificativas detalhadas, ofereceu um ponto de partida objetivo para que os times pudessem revisar seus acordos internos sobre o que caracteriza a entrega de valor.

ISSN: 2594-8083

A análise longitudinal dos resultados demonstrou ainda que as equipes com maiores pontuações médias nas quatro dimensões avaliadas tendiam a apresentar maior coesão interna entre os critérios, evidenciando uma cultura mais madura de documentação, refinamento e entrega. Nessas equipes, os critérios de aceitação estavam diretamente conectados à história, o *DoR* estabelecia claramente as condições de entrada e o *DoD* servia como verificação concreta da saída. Isso mostra que, mais do que o uso isolado de boas práticas, o alinhamento entre os artefatos é um indicativo de qualidade sistêmica.

Outro ponto de destaque foi a aceitação prática das recomendações geradas pela IA. Durante os *feedbacks* qualitativos colhidos com os *Product Owners* e *Agile Masters*, muitos relataram que as sugestões apresentadas eram, em sua maioria, acionáveis e compatíveis com o nível de entendimento da equipe. Isso evidencia o valor pedagógico da ferramenta, que atua não apenas como avaliadora, mas como instrumento de aprendizagem contínua. Para ilustrar de forma concreta os resultados obtidos, a Figura 6 apresenta a avaliação de três histórias distintas, com pontuações representativas de diferentes níveis de qualidade.

Figura 6: Demonstração de 3 histórias com avaliações distintas com nota 4,2 e 0

Story	descricao.Story	avaliacao.Story	recomendacoes.Story	exemplos.Story	Nota Stories
STS0076100-US128: Alterar componente de filtro de unidade gestora para permitir a múltipla seleção na tela de Registro de Não Conformidade	COMO usuário do SIGA OS EU QUERO que o componente de filtro de unidade gestora na tela de Registro de Não Conformidade permita múltipla seleção PARA facilitar o registro de ocorrências que envolvam mais de uma unidade gestora.	A história está bem estruturada no formato padrão, mas poderia ser mais clara quanto ao valor de negócio. Está pequena o suficiente para um sprint.	Esclarecer o valor de negócio e garantir que a história não inclua soluções técnicas.	COMO usuário do SIGA OS, EU QUERO que o componente de filtro de unidade gestora na tela de Registro de Não Conformidade permita múltipla seleção PARA facilitar o registro de ocorrências que envolvam mais de uma unidade gestora.	4,00
STS0077947-US134: [Sustentação] Tratar problema de reportado no incidente (INC3101044)	Após análise do incidente, identificamos que a lista de ações na aba Implementação não possui paginação. Devido a falta de paginação, o usuário não consegue visualizar todas as ações registradas do RTA.	A história não está no formato padrão de user story e não foca no que o usuário deseja alcançar, mas sim em uma solução técnica. Não há clareza quanto ao valor de negócio ou resultado esperado.	Reescrever a história no formato padrão de user story, focando no que o usuário deseja alcançar e o valor de negócio.	Como usuário do RTA, quero visualizar todas as ações registradas na aba Implementação para que eu possa gerenciar eficientemente minhas tarefas.	2,00
STS0056156-US80: Atualizar manual referente a tela Melhorias no registro do RTA		Sem conteúdo para análise.	Preencher este campo conforme boas práticas ágeis.	Exemplo de preenchimento conforme metodologia ágil.	0,00

Fonte: Elaborado pelos autores (2025).

Conforme mostrado, o modelo entrega a avaliação, a recomendação de como deve ser tratado e complementa com um exemplo. Este mesmo formato também foi implementado para o critério de aceitação, o *Definition of Done* e o *Definition of Ready*. Isso traz uma forma padronizada de ao devido tempo criar o conceito para que todas as equipes atinjam o seu melhor nível de planejamento, alinhados com as práticas descritas nas metodologias.

Por fim, observou-se que a presença de uma estrutura padronizada de avaliação, apoiada por uma IA explicável, contribuiu para a construção de um vocabulário comum entre os membros dos times. Termos como “critérios testáveis”, “verificável”, “valor de negócio” e

"fluxos alternativos" passaram a fazer parte das discussões cotidianas, fortalecendo a capacidade dos times de revisar criticamente suas histórias, com base em parâmetros técnicos claros.

Em síntese, os resultados reforçam que a aplicação de LLMs para avaliação semântica de itens do *backlog* não apenas é viável, como pode atuar como um catalisador de maturidade organizacional. A análise detalhada de cada artefato, combinada à geração de insights contextualizados, representa um avanço importante no uso da Inteligência Artificial como aliada na governança de produtos ágeis.

4.1. LIÇÕES APRENDIDAS

Durante a execução deste projeto, diversas lições foram consolidadas, tanto no plano técnico quanto organizacional. Esses aprendizados refletem os desafios enfrentados na integração entre tecnologias emergentes de linguagem natural e os processos ágeis em escala, bem como as estratégias adotadas para garantir a viabilidade e a escalabilidade da solução proposta.

Uma das primeiras lições aprendidas foi a compreensão prática sobre a utilização da API do ChatGPT. O time enfrentou inicialmente dificuldades operacionais para processar grandes volumes de informações textuais, especialmente histórias de usuário estruturadas em planilhas eletrônicas. O desafio impulsionou o domínio do uso combinado de *Python*, bibliotecas específicas para manipulação de dados e conectividade segura com a API da *OpenAI*. Esse processo tornou evidente que, embora existam custos associados ao consumo da API, a viabilidade da solução se justifica pelo grau de precisão, velocidade e escala que seria inatingível por meios manuais.

Outro aprendizado relevante emergiu do processo iterativo de construção dos prompts. Inicialmente, foram testadas instruções simples, com baixa densidade semântica e poucos critérios, o que resultou em respostas genéricas e pouco acionáveis. Ao evoluir para prompts mais robustos, com instruções explícitas, critérios de avaliação claramente definidos e escalas de pontuação graduadas de 0 a 5, foi possível obter respostas mais consistentes, coerentes com os objetivos do projeto. Essa experiência reforça que a clareza na formulação do prompt é essencial para assegurar a aderência do modelo às expectativas de análise.

A colaboração com stakeholders internos, especialmente *Product Owners* e *Agile Masters*, revelou-se indispensável. A validação das respostas geradas pela IA junto a essas *personas* permitiu ajustar o processo de avaliação, garantir a contextualização adequada dos

critérios e ampliar a aceitação da ferramenta. Uma crítica lição aprendida foi o reconhecimento da importância de envolver as equipes desde a concepção do modelo de análise até a interpretação dos resultados.

Em paralelo, a aplicação de um questionário diagnóstico junto aos times permitiu aferir a maturidade e a disposição das equipes em adotar ferramentas baseadas em Inteligência Artificial. O retorno foi positivo: identificou-se uma abertura significativa para adoção dessa abordagem, o que reforçou a relevância da solução e justificou o investimento em sua implementação. Essa etapa revelou-se estratégica para calibrar expectativas e ajustar a solução à realidade organizacional.

No decorrer do projeto, também ficou evidente a necessidade de modularizar as entregas. A proposta inicial previa uma arquitetura mais sofisticada, com múltiplos agentes automatizados. No entanto, a experiência mostrou que iniciar com uma solução mais simples e controlável, focada na avaliação automatizada de qualidade do *backlog*, foi decisivo para o sucesso da primeira fase. Essa decisão permitiu validar a abordagem, reduzir riscos e acumular conhecimento incremental para expansões futuras.

Outro aspecto significativo refere-se à função educativa do sistema. Ao retornar não apenas uma nota para cada critério avaliado, mas também uma justificativa textual, uma sugestão de melhoria e um exemplo idealizado, a ferramenta passou a operar como mecanismo de treinamento contínuo. Assim, além de mensurar a qualidade das histórias, o sistema contribui para a disseminação de boas práticas e para o desenvolvimento das competências dos times ágeis.

A análise das respostas geradas pela LLM também demonstrou alinhamento com percepções qualitativas previamente levantadas. Times que, em avaliações humanas, já se destacavam pela qualidade na escrita de histórias de usuário, também obtiveram melhores resultados na análise automatizada, o que valida a coerência da abordagem. Por outro lado, equipes com baixa aderência aos critérios esperados foram rapidamente identificadas, reforçando o valor do sistema como instrumento de diagnóstico e melhoria contínua.

Por fim, uma das lições mais relevantes é que o *pipeline* desenvolvido não se limita à avaliação do *backlog*. A lógica da solução é totalmente escalável para outros contextos e processos organizacionais. Alterando-se o conteúdo do *prompt*, os critérios de avaliação e os dados de entrada, a metodologia pode ser replicada para múltiplas finalidades, mantendo a estrutura técnica e operacional já validada. Esse aspecto confere à solução um caráter transversal e adaptável, com alto potencial de reaproveitamento em outras iniciativas

ISSN: 2594-8083

estratégicas da organização.

5. CONSIDERAÇÕES FINAIS

Este estudo teve como objetivo investigar o uso de modelos de linguagem de larga escala (LLMs), como ferramenta de apoio à gestão da qualidade de itens do *Product Backlog* em ambientes ágeis. A partir da análise da literatura, foi possível identificar lacunas recorrentes na engenharia de requisitos ágil, tais como baixa padronização, falta de critérios objetivos e elevada ambiguidade, o que compromete a eficácia do *backlog* como instrumento de comunicação, planejamento e entrega de valor. Com base nesse diagnóstico, foi proposto e implementado um *framework* técnico baseado em LLMs, cuja eficácia foi validada por meio do desenvolvimento de um Produto Mínimo Viável aplicado a backlogs reais de diferentes equipes.

Os resultados obtidos demonstraram a viabilidade prática da proposta e seu alinhamento com as necessidades identificadas, onde o modelo de linguagem mostrou-se capaz de avaliar, de forma padronizada e coerente, diferentes dimensões da qualidade das histórias de usuário, incluindo clareza descritiva, testabilidade dos Critérios de Aceitação, completude da *Definition of Ready* e robustez da *Definition of Done*. Além da pontuação atribuída, a entrega de justificativas, sugestões de melhoria e exemplos ideais transformou o modelo em uma ferramenta de apoio à aprendizagem contínua das equipes, promovendo a difusão de boas práticas e o fortalecimento da cultura de qualidade nos processos ágeis. Um resultado pertinente observado foi a correlação entre maturidade técnica das equipes e consistência interna nos critérios avaliados, revelando que squads com maior aderência às boas práticas apresentaram não apenas melhores notas, mas também maior alinhamento entre os artefatos ágeis. Outro ponto de destaque foi a boa receptividade por parte de *Product Owners* e *Agile Masters*, que identificaram valor nas recomendações geradas pela IA, especialmente por sua natureza contextualizada e acionável.

Entretanto, a pesquisa apresenta algumas limitações. Primeiramente, o uso do modelo GPT-4o depende de conectividade com a API da *OpenAI* e está sujeito a custos operacionais e limites técnicos (como tamanho de token e latência de resposta). Além disso, embora a análise automatizada tenha se mostrado coerente, os resultados ainda demandam interpretação humana, especialmente em contextos com alta ambiguidade ou requisitos altamente específicos. A avaliação também foi conduzida com um número limitado de squads e em um cenário controlado, o que pode restringir a generalização dos resultados.

ISSN: 2594-8083

As lições aprendidas durante a implementação reforçam a importância da modularização da solução, da construção cuidadosa de prompts, e do envolvimento contínuo dos *stakeholders* para garantir aderência contextual e validação prática. Um aprendizado essencial foi que a estrutura técnica desenvolvida é escalável e pode ser adaptada para outros contextos organizacionais, como avaliação de épicos, testes de aceitação ou até mesmo revisão de incidentes, apenas ajustando os critérios de avaliação e o conteúdo dos *prompts*.

Como direções futuras, propõe-se expandir o uso da ferramenta para ambientes mais heterogêneos e equipes com diferentes níveis de maturidade ágil, ampliando a base de validação empírica do modelo. Além disso, uma linha promissora de evolução consiste na integração da IA ao ciclo de vida dos requisitos desde sua origem. Especificamente, sugere-se explorar o uso de LLMs para estruturar histórias de usuário a partir de transcrições de reuniões de planejamento ou refinamento, considerando o contexto específico de cada projeto e utilizando dados complementares, como documentos de visão, *roadmap*, regras de negócio e restrições operacionais. Com isso, seria possível gerar automaticamente histórias completas, claras e alinhadas aos padrões organizacionais, exportando-as diretamente para planilhas ou sistemas de gestão como *ServiceNow*, com validadores embutidos. A adoção desse modelo exigiria, naturalmente, a adaptação do formato das reuniões, com a condução orientada por roteiros predefinidos que maximizem a extração de informações relevantes pelo modelo de linguagem. Adicionalmente, a integração com plataformas de reuniões online e ferramentas de transcrição automatizada poderia facilitar esse fluxo, viabilizando uma coleta de dados mais fluida e precisa.

Como próximas etapas, também prevê-se a integração do sistema com ferramentas de backlog em tempo real (como *Jira* e *Azure DevOps*), o desenvolvimento de assistentes interativos para apoiar a escrita de histórias durante o refinamento e a aplicação de técnicas de similaridade semântica para estimativas por analogia. Ao estabelecer uma ponte entre tecnologia de ponta e práticas de gestão ágil, esta metodologia contribui não apenas para a eficiência operacional, mas para a consolidação de uma cultura de qualidade, transparência e aprendizado contínuo nos times de desenvolvimento.

Em síntese, os resultados alcançados demonstram que o uso de LLMs na avaliação e suporte à construção de itens do backlog representam uma abordagem promissora, viável e alinhada aos desafios da engenharia de requisitos ágil. A proposta apresentada contribui tanto para o aumento da qualidade técnica das histórias de usuário quanto para a evolução da maturidade organizacional no uso de práticas ágeis orientadas por dados e Inteligência

Artificial. Espera-se que os achados deste trabalho sirvam como base para novos estudos e inovações voltadas à integração da IA nos processos de desenvolvimento ágil em larga escala.

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) e da Universidade Tecnológica Federal do Paraná.

REFERÊNCIAS

ALHAZMI, Alhejab; HUANG, Shihong. Integrating Design Thinking into Scrum Framework in the Context of Requirements Engineering Management. In: **Proceedings of the 2020 3rd International Conference on Computer Science and Software Engineering (CSSE'20)**, Beijing, China, 2020. p. 1–13. ACM. DOI: 10.1145/3403746.3403902.

ALHAZMI, O. A.; HUANG, Y. Agile methodologies for software development: practices and challenges. **International Journal of Computer Science and Information Technology**, v. 10, n. 3, p. 68-75, 2020.

ALMEIDA, Fernando; CARNEIRO, Pedro. Perceived importance of metrics for agile Scrum environments. **Information**, v. 14, n. 6, p. 327, 2023. DOI: 10.3390/info14060327.

BECK, K.; CUNNINGHAM, W.; MARTIN, R. Agile Software Development: Principles, Patterns, and Practices. **Prentice Hall**, 2001.

BROWN, T. Change by Design: How Design Thinking Creates New Alternatives for Business and Society. **Harper Business**, 2009.

CHANG, W.; LEE, T.; YANG, Y. A study on user stories in agile software development. **Journal of Software Engineering**, v. 34, n. 2, p. 113-128, 2016.

CHESNEY, R. Artificial Intelligence and Software Development: A New Era. **MIT Press**, 2023.

COHN, M. User Stories Applied: For Agile Software Development. Boston: **Addison-Wesley**, 2004.

CRESWELL, J. W.; CRESWELL J. D. Projeto de pesquisa: métodos qualitativo, quantitativo e misto. 5. ed. São Paulo: **Penso**, 2021.

DIEBOLD, A. Generative AI in Agile Development. **Wiley**, 2024.

DIEBOLD, Philipp. From backlogs to bots: Generative AI's impact on Agile role evolution. **Journal of Software: Evolution and Process**, v. 37, e2740, 2025. DOI: 10.1002/smr.2740.

DIGITAL.AI. 16th State of Agile Report. **Digital.AI** 2021. Disponível em: https://www.digital.ai/resource-center/analyst-reports/state-of-agile-report. Acesso em: 15 jul. 2025.

DILORENZO, Ednaldo et al. Enabling the reuse of software development assets through a taxonomy for user stories. **IEEE Access**, v. 8, p. 107285–107298, 2020. DOI: 10.1109/ACCESS.2020.2996951.

DILORENZO, L.; PIERCE, R.; SMITH, J. A study on technology acceptance and AI in software

projects. **Journal of Software Engineering**, v. 26, p. 45-61, 2020.

FOWLER, M. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. **Addison-Wesley**, 2019.

GIL, A. C. Como elaborar projetos de pesquisa. 7. ed. São Paulo: **Atlas**, 2022.

GILSON, Fabian; GALSTER, Matthias; GEORIS, François. Extracting quality attributes from user stories for early architecture decision making. In: 2019 **IEEE International Conference on Software Architecture Companion (ICSA-C)**. IEEE, 2019. p. 129–136. DOI: 10.1109/ICSA-C.2019.00031.

GILSON, Fabian; IRWIN, Calum. From user stories to use case scenarios: towards a generative approach. In: 2018 25th **Australasian Software Engineering Conference (ASWEC)**. IEEE, 2018. p. 61–65. DOI: 10.1109/ASWEC.2018.00016.

GILSON, M.; IRWIN, R. Automating user story management using AI. **Agile Software Engineering Journal**, v. 19, p. 121-135, 2018.

GILSON, M.; IRWIN, R. Managing agile software development projects: using user stories to prevent common mistakes. **Journal of Agile Project Management**, v. 14, p. 45-58, 2018.

GRIFFITH, P. Artificial Intelligence and Product Management: A Framework for the Future. **Springer**, 2020.

GROSS, M. D.; CAI, L.; LEE, B. Challenges in defining and managing user stories in agile projects. **Proceedings of the International Conference on Software Engineering**, p. 1125-1136, 2017.

GROSS, M.; CAI, L.; LEE, B. AI tools for backlog refinement in agile projects. **Journal of Agile Development**, v. 25, n. 3, p. 85-99, 2021.

HIGHTOWER, L.; DOUGLAS, A.; WILSON, M. The role of engineering in requirements management. **IEEE Transactions on Software Engineering**, v. 45, n. 6, p. 514-528, 2019.

HIGHSMITH, J. Agile Project Management: Creating Innovative Products. **Addison-Wesley**, 2002.

KIM, W.; PARK, J.; SHIN, J. Integrating Design Thinking with Agile methodologies: a survey. **Journal of Agile Development**, v. 23, p. 84-96, 2019.

LI, Z.; JAYARAMAN, V.; AZAR, M.; MAALOUF, S. Reducing operational costs through AI without compromising accuracy. **Journal of Artificial Intelligence and Software Engineering**, v. 34, n. 1, p. 56-72, 2024.

LÓPEZ, M.; RIVERA, A.; TORO, V. Software quality metrics for agile software development. **Software Quality Journal**, v. 30, p. 27-42, 2022.

LUCAS, R.; MORAES, T.; PEREIRA, R. User stories and agile methodologies: A comprehensive approach. **Agile Development Review**, v. 29, n. 2, p. 98-105, 2018.

MEILIANA, Meiliana et al. Agile software development effort estimation based on product backlog items. In: 8th International Conference on Computer Science and Computational Intelligence (ICCSCI 2023). **Procedia Computer Science**, v. 227, p. 186–193, 2023. DOI: 10.1016/j.procs.2023.10.516.

NASCIMENTO, F. P.; SOUSA, F. L. L. Metodologia da pesquisa científica: teoria e prática – como elaborar TCC. 3. ed. Brasília: **Thesaurus**, 2023.

NASIRI, M.; LAHMER, S. Automated Tools for UML Generation from User Stories: Improving Agile Practices. **Journal of Software Engineering and Applications**, v. 17, p. 57-68, 2024.

NASIRI, M.; LAHMER, S. Automating user story identification and management with AI. **International Journal of Agile Software Development**, v. 18, p. 102-115, 2024.

NASIRI, M.; LAHMER, S. Role of continuous refinement in agile project management. **Journal of Software Engineering and Applications**, v. 17, p. 35-47, 2024.

NASIRI, Samia; LAHMER, Mohammed. A smart AI framework for backlog refinement and UML diagram generation. **International Journal of Advanced Computer Science and Applications**, v. 15, n. 4, p. 722–736, 2024. DOI: 10.14569/IJACSA.2024.0150489.

PICHLER, R. Agile Product Management with Scrum: Creating Products that Customers Love. **Addison-Wesley**, 2010.

RIBEIRO, J.; SILVA, P.; COSTA, F. The importance of test automation in agile software development. **Journal of Test Engineering**, v. 15, p. 47-58, 2021.

ROLA, Paweł; KUCHTA, Dorota. Application of fuzzy sets to the expert estimation of Scrum-based projects. **Symmetry**, v. 11, n. 8, p. 1032, 2019. DOI: 10.3390/sym11081032.

SACHDEVA, Samridhi et al. Prioritizing user requirements for agile software development. In: 2018 **International Conference on Advances in Communication and Computing Technology (ICACCT)**, Sangamner, India. IEEE, 2018. p. 495–498. DOI: 10.1109/ICACCT.2018.8529592.

SANTOS, P. AI and Metrics in Software Development: Transforming Project Management. **Springer**, 2023.

SCHÖN, D.; THOMASCHEWSKI, J.; ESCALONA, M. Agile requirements engineering and Design Thinking: A comparison and proposal. **International Journal of Software Engineering and Knowledge Engineering**, v. 27, n. 4, p. 45-61, 2017.

SCHÖN, EVA-MARIA; THOMASCHEWSKI, JÖRG; ESCALONA, MARÍA JOSÉ. Agile Requirements Engineering: A Systematic Literature Review. **Computer Standards & Interfaces**, 2017.

SCHWAB, J.; BROWN, P.; HICKS, R. Key Scrum metrics and their impact on software projects. **Journal of Agile Practices**, v. 24, p. 89-103, 2018.

SCHWABER, K.; SUTHERLAND, J. The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. **Scrum.org**, 2017.

SEDANO, Todd; RALPH, Paul; PÉRAIRE, Cécile. The product backlog. In: Proceedings of the 2019 **IEEE/ACM 41st International Conference on Software Engineering (ICSE)**, Montreal, Canada. IEEE, 2019. p. 200–211. DOI: 10.1109/ICSE.2019.00036.

SENFT, B.; FISCHER, H.; OBERTHÜR, S.; PATKAR, N. Assist users to straightaway suggest and describe experienced problems. In: MARCUS, A.; WANG, W. (Eds.). DUXU 2018. Cham: **Springer International Publishing**, 2018. v. 10918, p. 758–770.

SENFT, D.; FISCHER, M.; OBERTHUR, T.; PATKAR, V. Agile Methods in Software Engineering. **Springer**, 2018.

ISSN: 2594-8083

SILVA, M.; PEREIRA, R.; COSTA, G. The relationship between software maintainability and quality metrics. **Software Engineering Journal**, v. 33, p. 19-30, 2020.

SOMMER, T. AI-Driven Agile Development: Enhancing Product Owner's Role. **Elsevier**, 2022.

STARON, M.; BOSCH, J.; JÄRVELIN, A. Software metrics and their importance in agile environments. **Software Process Improvement Journal**, v. 11, p. 78-90, 2016.

SUTHERLAND, J.; SCHWABER, K. Software in 30 Days: How Agile Managers Beat the Odds, Delight Their Customers, and Leave Competitors in the Dust. **Crown Business**, 2013.

VANDERMEER, S.; FISCHER, T.; NOBLE, S. Evaluating user acceptance of AI-based tools for agile teams. **Journal of Technology Acceptance Studies**, v. 17, n. 4, p. 98-112, 2022.

WAGENAAR, M.; KALLIO, K.; HARRINGTON, J. The Role of Software Product Management in Agile Methodologies. **Journal of Product Development**, v. 22, p. 34-48, 2017.

WANG, Q.; LIU, Z.; WANG, R. Cost reduction with AI: Efficiency versus accuracy in software testing. **International Journal of Software Testing**, v. 21, n. 2, p. 112-125, 2021.