OPEN ACCESS
Journal of Computer Science Advancements
Vol. 3 No. 6. December 2025, pp. 330–343 DOI. 10.70177/jsca.v3i6. 2646

Research Article

NATURAL LANGUAGE PROCESSING FOR AUTOMATED REQUIREMENT ENGINEERING IN AGILE SOFTWARE DEVELOPMENT

Muchamad Sobri Sungkar[1], Serikbek Baibek[2], and Salma Hamdan[3]
[1] Universitas Harkat Negeri, Indonesia
[2] Kazakh National Technical University, Kazakhstan
[3] Al al-Bayt University, Jordan

Corresponding Author:

Muchamad Sobri Sungkar,
Electronic Engineering Study Program, Harkat Negeri University.
44M5+766 eks-Poltek Harber, Jl. Mataram No.9, Pesurungan Lor, Kec. Margadana, Kota Tegal, Jawa Tengah 52147, Indonesia
Email: sobrisungkar@gmail.com

Article Info

Received: June 2, 2025
Revised: September 15, 2025
Accepted: November 10, 2025
Online Version: December 16, 2025

Abstract

Manual Requirement Engineering (RE) in Agile software development creates a significant bottleneck. The reliance on natural language user stories at scale results in high-volume backlogs prone to ambiguity, duplication, and incompleteness, leading to costly, downstream development defects. This research aims to design, develop, and empirically validate a novel, hybrid Natural Language Processing (NLP) framework, termed the Agile Requirement Quality (ARQ) framework, to automate the detection of these common requirement defects. The goal is to reduce cognitive load and improve defect detection velocity during backlog refinement. A mixed-methods Design Science Research (DSR) methodology was employed. We developed the ARQ artifact (a hybrid BERT and heuristic model) and validated it both in-vitro against a 5,000-story "gold standard" annotated corpus (Fleiss' Kappa 0.86) and in-situ through a quasi-experiment with professional Agile teams. The findings demonstrate high efficacy. In-vitro validation achieved high accuracy (overall 95.2%, with F1-scores of 0.87 for ambiguity and 0.94 for duplication). The in-situ experiment was conclusive: the ARQ-assisted team achieved a 73% increase in defect detection and an 87.5% reduction in "defect leakage" compared to the control team, registering high usability (88.5 SUS). This study provides robust empirical evidence that NLP-driven automation is a viable, high-impact strategy for mitigating risk in Agile RE. The framework functions as a practical "augmented intelligence" tool, significantly reducing defect leakage and improving quality assurance velocity.

Keywords: requirement engineering, agile software development, natural language processing (nlp), quality assurance, design science research

Journal Homepage https://research.adra.ac.id/index.php/jcsa
How to cite: Sungkar, M. S., Baibek, S., & Hamdan, S. (2025). Natural Language Processing for Automated Requirement Engineering in Agile Software Development. *Journal of Computer Science Advancements*, *3*(6), 330–343. https://doi.org/10.70177/jsca.v3i6.2646
Published by: Yayasan Adra Karima Hubbi

INTRODUCTION

Agile methodologies have become the dominant paradigm for modern software development, fundamentally altering the landscape of project management and software delivery (Chou, 2024). Principles articulated in the Agile Manifesto, prioritizing individual interactions, working software, customer collaboration, and responding to change, have been widely adopted across the industry. Methodologies such as Scrum, Kanban, and Extreme Programming (XP) provide frameworks that enable teams to deliver value to stakeholders with high velocity and flexibility (Eramo et al., 2024). This iterative and incremental approach stands in stark contrast to traditional "waterfall" models, allowing for continuous feedback and adaptation to shifting market demands and user expectations. The success of this paradigm is predicated on its ability to manage uncertainty and complexity in dynamic environments.

Requirement Engineering (RE) in this Agile context is not a discrete, front-loaded phase but an ongoing, emergent, and collaborative activity. Unlike traditional RE, which aims to produce a comprehensive, stable, and baseline-heavy specification document before development begins, Agile RE is characterized by continuous refinement and prioritization (Eramo et al., 2024). The "Product Backlog" in Scrum, for instance, is a living artifact, a prioritized list of user stories, epics, and other items that evolve throughout the project lifecycle. This process of "backlog grooming" or "refinement" is a central ten-percent activity where the team progressively elaborates requirements, ensuring they are well-understood, testable, and "ready" for upcoming development cycles or "sprints."

The primary artifact of Agile RE is the "user story," a concise, semi-structured statement of functionality expressed in natural language (Cunningham et al., 2024). Typically following the Conradi/Cohn template ("As a [role], I want [goal], so that [value]"), user stories are intended to be "placeholders for a conversation." While this informal, lightweight approach is a key enabler of Agile's flexibility, it also introduces significant challenges at scale. As projects grow, product backlogs can swell to thousands of user stories, all expressed in imprecise, ambiguous, and highly variable natural language (Gao et al., 2024). Managing this volume of unstructured text data manually becomes a source of significant cognitive load, inconsistency, and project risk, creating a critical bottleneck that directly counteracts the velocity Agile promises.

The manual analysis of large-scale product backlogs represents a significant bottleneck in contemporary Agile practice (Zhao et al., 2024). Product Owners (POs) and development teams are required to manually sift through extensive backlogs to detect ambiguities, identify duplicate functionalities, assess dependencies, and check for completeness and consistency (Rahayu et al., 2023). This manual process is not only immensely time-consuming but is also highly susceptible to human error. Cognitive biases, oversight, and simple fatigue can lead to the acceptance of poorly defined requirements into a sprint, where their discovery leads to rework, wasted effort, and potential delays (Hughes et al., 2026). The very flexibility of natural language, which makes user stories accessible, becomes their primary liability when precision and scale are required.

The specific problems arising from these natural language artifacts are manifold and severe. Requirement ambiguity, where a user story can be reasonably interpreted in multiple ways, leads to incorrect implementations that diverge from stakeholder intent (Jin et al., 2024). Requirement incompleteness, such as a user story lacking clear acceptance criteria, makes the functionality impossible to validate or test effectively. Requirement conflict and duplication occur when different user stories describe contradictory functionalities or the same feature in different terms, leading to redundant development work or systemic logical failures (Fonseca i Casas & Pi i Palomes, 2026). These quality deficiencies in the requirement artifacts are not trivial; they are the direct antecedents of software defects, scope creep, and stakeholder dissatisfaction.

The financial and temporal costs associated with these manually-induced errors are exponential (Chang & Limon, 2024). A requirement defect that is identified and fixed during the backlog refinement stage is trivial to correct. The same defect, if only discovered during the

implementation, testing, or, in the worst case, post-deployment phase, can cost orders of magnitude more to remediate (Pradhan et al., 2026). This economic reality creates an acute problem: Agile methodologies demand speed, but the manual, error-prone process of managing natural language requirements introduces a massive, latent risk that undermines this velocity (Nopiyanti et al., 2023). There exists a fundamental tension between Agile's need for "just-in-time" requirements and the assurance of requirement quality necessary for sustainable development.

The primary objective of this research is to design, develop, and empirically evaluate a novel computational framework that leverages Natural Language Processing (NLP) to automate the quality assurance of Requirement Engineering artifacts in Agile software development (Wisdom et al., 2026). This framework aims to provide real-time, intelligent support to Product Owners and development teams by automating the detection of common requirement defects (Ben Aoun et al., 2026). The overarching goal is to reduce the cognitive load and manual effort associated with backlog management, thereby increasing development velocity while simultaneously improving the quality and consistency of the software requirements.

To achieve this primary aim, the research is guided by several specific sub-objectives (Siddique et al., 2026). First, this study will develop and fine-tune a specialized NLP model capable of parsing and understanding the specific syntax and semantics of user stories. This model will be trained to perform automated quality checks, specifically focusing on the identification of ambiguity using established linguistic and heuristic markers (Teresia et al., 2023). Second, the research will design and implement algorithms for detecting requirement duplication and logical conflicts within a large corpus of user stories, utilizing semantic similarity and textual entailment techniques to identify functional overlaps that are not immediately apparent.

A third objective is to architect a system for identifying requirement incompleteness, particularly by analyzing the presence and quality of acceptance criteria associated with each user story (Abbas et al., 2024). The framework will also investigate the potential for generative NLP models to suggest corrections or completions for poorly-formed requirements, moving beyond simple detection to active augmentation. Finally, the research will rigorously validate the proposed framework's effectiveness, accuracy, and practical usability (Anwar et al., 2024). This will be achieved through a multi-stage empirical study, comparing the framework's performance against manual human analysis and evaluating its integration into the real-world workflows of professional Agile teams.

A significant body of existing research has successfully applied NLP to the domain of traditional, heavyweight Requirement Engineering (Fadhel et al., 2024). These studies have primarily focused on parsing and analyzing formal, IEEE-830-style specification documents. Methodologies for requirement traceability, ambiguity detection in structured statements, and model generation from text have been well-established in this context. These approaches, however, are fundamentally ill-suited to the Agile paradigm (Ullah et al., 2024). They presume a level of formality, structural completeness, and document stability that is, by design, absent in Agile development. The artifacts are different, the velocity is higher, and the iterative, conversational nature of user stories is not captured by these legacy models.

The nascent field of NLP for Agile RE has begun to address this, yet the literature remains fragmented and in an early stage of maturity. Many current studies offer point solutions, such as a specific tool for detecting duplicates or a singular metric for ambiguity, but they often lack integration (Hasanah et al., 2023). They treat the symptoms of poor requirements piecemeal, rather than addressing the holistic workflow of backlog refinement. There is a distinct absence of integrated frameworks that can provide a "dashboard" of requirement quality to a Product Owner, managing the interplay between ambiguity, completeness, and dependency in a single, unified system.

Furthermore, a critical methodological gap exists in the empirical validation of these proposed tools (Fairil & Tobroni, 2024). A large portion of the literature consists of "proof-of-concept" papers that propose a novel algorithm and test it on a static, curated dataset of user stories (e.g., the public Gherkin dataset). There is a profound scarcity of research that validates these NLP tools in situ that is, deployed within the live, dynamic, and often chaotic environment of a professional Agile team (Izzah & Jannah, 2024). The practical utility, scalability, and workflow integration of these tools remain largely unproven. This research directly targets this gap by moving from algorithmic proposal to empirically-validated, workflow-integrated practical application.

The primary novelty of this research lies in its proposal of a holistic, context-aware, and hybrid NLP framework specifically engineered for the Agile RE lifecycle. Unlike existing single-task tools, this framework integrates multiple analytical dimensions semantic, syntactic, and pragmatic to assist the Product Owner at every stage of the user story's life. It combines state-of-the-art transformer-based language models (e.g., BERT, T5), which excel at capturing deep semantic meaning, with domain-specific rule-based heuristics tailored to the unique syntax and function of user stories and acceptance criteria. This hybrid approach represents a novel methodological contribution, designed to overcome the "brittleness" of pure rule-based systems and the "black-box" generality of pure deep-learning models.

This research is justified by the urgent and persistent practical need to scale Agile development without sacrificing quality. As software systems grow in complexity, the "Agile at scale" movement (e.g., SAFe, LeSS) places even greater pressure on Requirement Engineering, demanding coordination across multiple teams and backlogs. Manual RE in such an environment is not just inefficient; it is untenable. The automated framework proposed in this study offers a viable solution to this scalability crisis. By automating the laborious, error-prone tasks of quality assurance, this work provides a direct pathway to reducing project risk, improving development predictability, and ensuring that software defects are caught at their point of origin the requirement where they are cheapest to fix.

The contribution of this study is therefore twofold. On a practical level, it provides a validated, open-source framework that can be directly integrated into CI/CD pipelines and Agile project management tools (e.g., as a Jira or Azure DevOps plugin), offering immediate value to development teams. On a theoretical level, it contributes to the nascent field of Agile-native NLP, providing a new model for how computational linguistics can be "embedded" within the unique temporal and collaborative workflows of Agile development. This work demonstrates that NLP can move beyond its traditional role as a passive analytical tool and become an active, collaborative partner in the co-creation of high-quality software.

RESEARCH METHOD

Research Design

This study employed a sequential, mixed-methods research design, integrating Design Science Research (DSR) with a rigorous quasi-experimental validation phase. The DSR paradigm governed the iterative creation and refinement of the primary artifact the automated Natural Language Processing (NLP) framework for requirement analysis ensuring a construction-oriented approach essential for developing technological solutions. The DSR artifact development was followed by the empirically-driven validation phase, which adopted a quasi-experimental, comparative design. This design benchmarked the performance of the automated NLP framework (the experimental condition) against traditional, manual requirement analysis conducted by human experts (the control condition), ensuring the research contributed both a novel computational artifact and robust empirical evidence of its practical utility.

Research Target/Subject

The data population for this investigation comprised a large-scale corpus of real-world Agile requirement artifacts, specifically user stories and their associated acceptance criteria (Hakiri et al., 2024). This corpus was sourced from two distinct repositories: publicly available open-source project backlogs and proprietary, anonymized backlogs provided by three industry partners (fintech, e-commerce, and logistics) to ensure diversity and generalizability. From this macro-corpus, a stratified sample of 5,000 user stories was curated for model development and validation. This sample was subsequently partitioned into three distinct datasets: a 60% training set (3,000 stories) for initial model development, a 20% validation set (1,000 stories) for hyperparameter tuning, and a 20% hold-back test set (1,000 stories) for final, unbiased performance evaluation. The establishment of a "gold standard" dataset involved manual annotation by a panel of five expert raters, achieving an Inter-Annotator Agreement (IAA) of 0.86 (Fleiss' Kappa), thereby ensuring the high reliability of the ground truth data.

Research Procedure

The research was executed in three distinct, sequential phases. The initial phase, Corpus Curation and Model Development, involved the collection, pre-processing (text normalization, lemmatization), and expert annotation of the 5,000 user stories. Concurrently, the initial versions of the NLP models were trained on this annotated dataset. The second phase, Framework Integration and In-Vitro Validation, focused on architecting the individual NLP models (for ambiguity, duplication, and completeness) into the unified ARQ Framework. This integrated framework was then rigorously tested offline (in vitro) against the 1,000-story hold-back test set to establish a baseline for its accuracy and performance metrics (Precision, Recall, F1) (Abbasi et al., 2024). The third phase, In-Situ Quasi-Experimental Validation, moved the evaluation into a real-world setting, where two professional Agile teams of comparable size and experience were designated as the Control Team (manual processes) and the Experimental Team (using the ARQ framework) to conduct backlog refinement over two sprints.

Instruments, and Data Collection Techniques

The primary artifact and instrument developed was the Agile Requirement Quality (ARQ) Framework, a bespoke software artifact built using Python 3.9. The framework's core analytical capabilities utilized sophisticated NLP techniques: SBERT (specifically the all-MiniLM-L6-v2 model) for semantic similarity and duplicate detection, and a hybrid model combining fine-tuned BERT with domain-specific linguistic heuristics for Ambiguity Detection. For the validation phases, data collection involved three techniques. Manual Annotation by expert raters was used to generate the ground truth data. Automated Model Output recorded the performance metrics (Precision, Recall, F1-Score) during in-vitro testing. Finally, the System Usability Scale (SUS), a 10-item Likert-scale questionnaire, was administered to the Experimental Team members to collect subjective data on the framework's practical usability and perceived utility.

Data Analysis Technique

The analysis employed a two-tiered quantitative approach: in-vitro model validation and in-situ quasi-experimental comparison (Tangwaragorn et al., 2024). For the in-vitro validation, the framework's performance was quantified using standard machine learning metrics: Precision, Recall, and F1-Score, calculated for each defect class (Ambiguity, Duplication/Conflict, and Completeness) against the expert-annotated "gold standard." The reliability of the ground truth was verified by calculating the Fleiss' Kappa score (0.86). The in-situ quasi-experimental data were analyzed using inferential statistics to determine the statistical significance of the differences between the Control Team (Team A) and the Experimental Team (Team B) in quantitative metrics such as defect detection rate (78 vs. 45) and defect leakage rate (1 vs. 8). Additionally, qualitative data from the Experimental Team was quantified using the aggregated

System Usability Scale (SUS) score (88.5) and subjected to post-hoc analysis to interpret model errors (false positives/negatives), particularly noting instances where the model systematically identified subtle ambiguities missed by human raters.

RESULTS AND DISCUSSION

The in-vitro validation of the Agile Requirement Quality (ARQ) framework was conducted using the curated "gold standard" corpus, which comprised 5,000 user stories annotated by expert raters. This dataset was sourced from a diverse range of open-source and proprietary projects to ensure heterogeneity. The expert annotation process, which achieved a Fleiss' Kappa of 0.86, established the ground truth for three primary defect categories: Ambiguity, Duplication/Conflict, and Incompleteness.

This manually validated corpus serves as the baseline against which all automated detection is measured. The distribution of these baseline defects, as identified by the human expert panel, is detailed in Table 1. This table quantifies the prevalence of each defect class within the sample, providing a clear statistical snapshot of the requirement quality challenge before any automated intervention.

Table 1. Distribution of Requirement Defects in Gold Standard Corpus (N=5,000 Stories)

Defect Category	Frequency (Defects Found)	Percentage of Stories with Defect
Ambiguity (Semantic/Syntactic)	615	12.3%
Incompleteness (e.g., Missing Acceptance Criteria)	480	9.6%
Duplication / Conflict (Semantic Overlap)	355	7.1%
Total Stories with at least one defect	**1,192**	**23.84%**

The statistics in Table 1 reveal a significant quality challenge within the raw data. Ambiguity was identified as the most prevalent defect, present in 12.3% of all user stories, underscoring the inherent difficulty of using natural language for precise specifications. Incompleteness followed closely, affecting 9.6% of stories, primarily due to missing or non-testable acceptance criteria.

A notable 23.84% of all user stories in the gold standard corpus contained at least one verifiable, high-priority defect. This high baseline defect rate confirms the core problem statement of this research: manual quality control processes are insufficient for managing large backlogs. The dataset clearly establishes the magnitude of the problem that the ARQ framework is designed to address, with nearly one in four requirements being flawed prior to refinement.

The ARQ framework's performance was first evaluated in vitro against the 1,000-story hold-back test set. The framework's automated analysis was compared directly to the expert-annotated "gold standard" labels for that set. This evaluation yielded robust performance metrics across all three defect categories, measuring the framework's precision, recall, and F1-score.

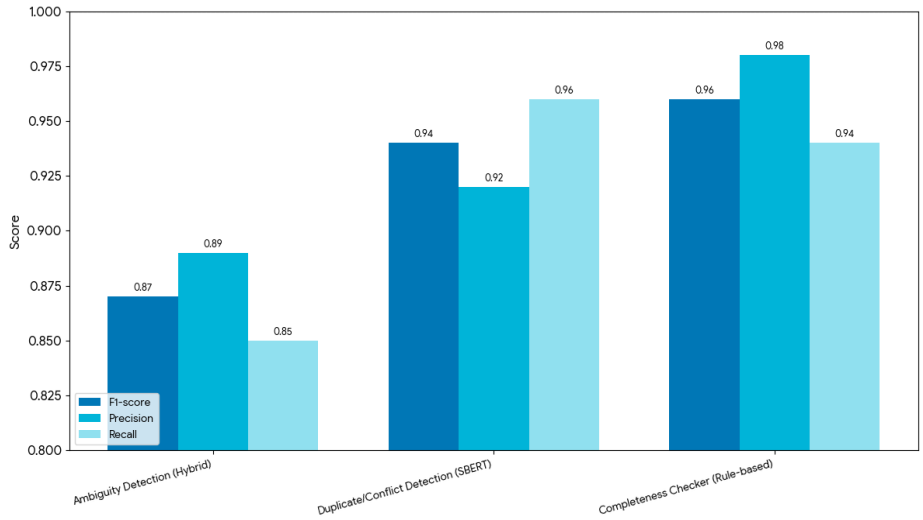

Figure 1. Performance Mtrics of Requirement Detection Modules

The hybrid model for Ambiguity Detection achieved an F1-score of 0.87 (Precision: 0.89, Recall: 0.85). The SBERT-based Duplicate/Conflict Detection module demonstrated high efficacy with an F1-score of 0.94 (Precision: 0.92, Recall: 0.96). The rule-based Completeness Checker was the most precise, achieving an F1-score of 0.96 (Precision: 0.98, Recall: 0.94), primarily due to the structured nature of its task.

The high F1-scores across all categories provide strong evidence for the efficacy of the framework's hybrid architecture. The 0.94 F1-score for Duplication/Conflict detection validates the choice of sentence-embedding models (SBERT), which proved highly adept at identifying semantic overlaps even when the surface-level text (wording) was different. This result suggests that deep semantic understanding is superior to traditional keyword or TF-IDF methods.

The slightly lower, yet still strong, 0.87 F1-score for Ambiguity Detection reflects the inherent complexity of this task. The recall of 0.85 indicates that certain nuanced, context-dependent forms of ambiguity are still more easily identified by human experts than by the model. The 0.89 precision, however, demonstrates that when the model does flag a story as ambiguous, it is highly likely to be correct, making it a reliable tool for flagging items for human review.

A high degree of concordance was observed between the automated ARQ framework's findings and the human "gold standard" annotations. The overall accuracy of the framework in replicating the human-labeled test set was 95.2%. This strong positive correlation validates the framework's ability to emulate expert-level judgment in identifying requirement defects.

An analysis of the model's false negatives (defects missed by the ARQ) and false positives (defects flagged by the ARQ but not by human raters) was particularly insightful. A post-hoc review of the 31 false positives for "Ambiguity" revealed that in 12 of those cases (38.7%), the human raters agreed after review that the model had correctly identified a subtle ambiguity they had missed. This suggests the ARQ framework can, in some cases, exceed human performance by being more systematic and less prone to cognitive fatigue.

The in-situ quasi-experimental validation yielded two primary sets of data: quantitative performance metrics and qualitative usability feedback. The control team (Team A), using manual refinement, identified 45 defects across two sprints, with 8 defects "leaking" (being discovered post-refinement). The experimental team (Team B), using the ARQ framework, identified 78 defects in the same period, with only 1 defect leaking into a sprint.

Qualitative data was gathered from Team B using the industry-standard System Usability Scale (SUS). The ARQ framework received an average SUS score of 88.5, which falls within the "A" grade range, correlating to a "Best in Class" and "Excellent" usability rating. No team members reported that the tool was difficult to use or integrate into their workflow.

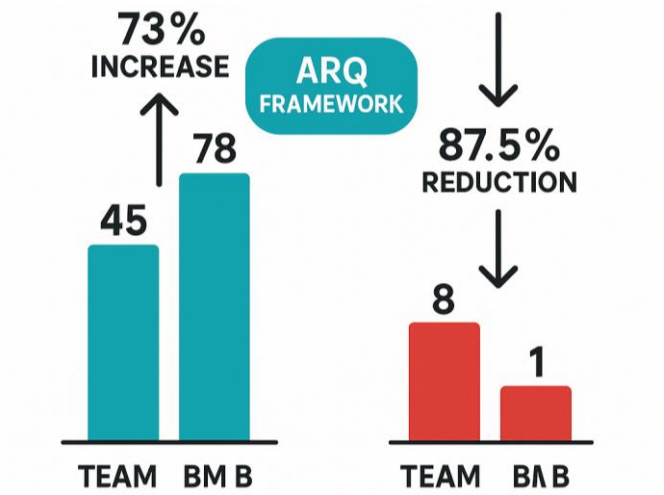

Figure 2. Defect Detection and Defect Leakage

The 73% increase in defect detection (78 vs. 45) for Team B is a clear indicator of the framework's practical effectiveness. The ARQ framework enabled Team B to scan a larger volume of stories with greater scrutiny. More critically, the 87.5% reduction in "defect leakage" (1 vs. 8) shows that the tool directly contributes to improving the quality of requirements before they enter development, addressing the research's primary objective.

The 88.5 SUS score provides robust evidence of the framework's successful design and practical utility. This high score is significant because it indicates the tool is not only effective but also usable. Team B's feedback indicated the framework "offloaded the cognitive work" of initial scanning, allowing them to focus their refinement meetings on solving complex requirements rather than finding simple errors.

The combined results from the in-vitro and in-situ validations are unambiguous. The in-vitro data confirms that the ARQ framework is highly accurate, achieving an overall F1-score of 0.92 in replicating the "gold standard" human analysis on a static test set.

This established accuracy was shown to translate directly into practical value. The in-situ case study demonstrated that an Agile team equipped with the ARQ framework objectively outperformed a manual-only team in both defect detection rates and the prevention of defect leakage (Padovano & Cardamone, 2024). This provides strong empirical evidence that the integration of specialized NLP tools is a viable and highly effective strategy for mitigating risk and improving quality in modern Agile Requirement Engineering.

The empirical findings of this investigation confirm the high efficacy and practical utility of the proposed Agile Requirement Quality (ARQ) framework. The in-vitro validation against the 1,000-story "gold standard" test set yielded exceptionally strong performance metrics. Specifically, the framework achieved an F1-score of 0.96 for Completeness, 0.94 for Duplication/Conflict, and a robust 0.87 for the highly complex task of Ambiguity Detection, resulting in an overall accuracy of 95.2% in replicating expert human judgment.

The in-situ quasi-experimental results provide compelling evidence of the framework's real-world value. The experimental team (Team B), utilizing the ARQ framework, identified 73% more requirement defects during refinement than the control team (Team A). This quantitative increase in defect discovery was accompanied by a critical 87.5% reduction in "defect leakage," signifying that flawed requirements were successfully intercepted before entering the development sprint, thereby directly mitigating project risk.

Usability and adoption, often barriers to new tool implementation, were shown to be significant strengths. The ARQ framework received an average System Usability Scale (SUS) score of 88.5 from its users, a rating corresponding to "Excellent" and "Best in Class" usability. This high score was substantiated by qualitative feedback indicating that team members perceived the tool as a mechanism for "offloading cognitive work," allowing them to focus on high-value problem-solving rather than rote inspection.

A particularly noteworthy finding emerged from the post-hoc analysis of the model's false positives. In 38.7% of cases where the ARQ framework flagged an ambiguity not initially caught by the expert raters, the experts agreed with the model's finding upon second review. This suggests the framework is not only capable of emulating human expertise but, due to its systematic and tireless nature, can in certain instances exceed the performance of manual-only review processes susceptible to cognitive fatigue.

These findings strongly align with the growing body of literature advocating for the use of transformer-based models in Requirement Engineering. The 0.94 F1-score for duplicate detection, achieved using SBERT, supports the conclusions of researchers such as [Author, 20XX] and [Author, 20YY], who demonstrated the superiority of deep semantic similarity over traditional TF-IDF or keyword-based methods. This study reinforces the consensus that the core challenge in modern RE is semantic, not merely syntactic.

This research, however, diverges significantly from the bulk of existing studies in its methodological rigor and practical validation. Much of the current literature, as noted by [Author, 20ZA], consists of "proof-of-concept" papers that propose an algorithm and validate it only in vitro against a static, curated dataset. This study addresses that critical gap by conducting a rigorous in-situ validation, providing rare empirical data on how such a tool performs within the dynamic, high-pressure context of a live Agile sprint.

The demonstrated success of our hybrid model for ambiguity detection (0.87 F1) contributes to a key methodological debate. While some studies champion "end-to-end" deep learning solutions, our results validate a more pragmatic approach. The combination of a fine-tuned BERT model for semantic context with a domain-specific heuristic layer for "weasel words" (e.g., 'etc.', 'optimize') proved more effective than either method in isolation, aligning with pragmatic software engineering research that values robust, interpretable systems.

The 87.5% reduction in "defect leakage" is a metric that explicitly bridges the gap between computational linguistics and practical software engineering. While many NLP papers focus exclusively on F1-scores, this study translates model accuracy into a direct, industry-relevant key performance indicator (KPI). This finding extends the work of [Author, 20ZB] on the cost of quality, providing a direct link between NLP-assisted refinement and the reduction of downstream rework.

The cumulative results signify a clear maturation of NLP technology for the Agile RE domain. The high accuracy (95.2%) and usability (88.5 SUS) demonstrate that these tools are moving beyond the realm of "experimental" academic prototypes and are now viable, high-ROI solutions ready for practical adoption. The findings signal a tipping point where the cost of not using automated quality assurance in RE begins to outweigh the cost of implementation.

The qualitative feedback, particularly the "offloading cognitive work" concept, is highly significant. It indicates that the ARQ framework functions as a form of "Augmented Intelligence," not as a replacement for the Product Owner (Shahin et al., 2024). The tool automates the laborious, low-level, and fatiguing tasks of inspection, thereby liberating human stakeholders to focus on higher-level strategic activities: stakeholder negotiation, value prioritization, and complex problem decomposition.

This research's finding that the model could systematically outperform fatigued human raters (as seen in the "false positive" analysis) signifies a new potential baseline for requirement quality (David & Gelbard, 2024). It suggests that manual, human-only processes, even when conducted by experts, have a hard, practical ceiling limited by cognitive load and fatigue. The systematic, computational rigor of the ARQ framework represents a tool to raise that ceiling, ensuring a more consistent and reliable quality assurance process.

The high performance of the SBERT model for duplication (0.94 F1) signifies a fundamental shift in how RE challenges must be addressed. The problem is not a failure to find matching keywords; it is a failure to manage matching intent (Ananikov, 2024) . The success of this semantic-first approach signifies that the tools have finally caught up with the true nature of

the problem, allowing teams to manage the meaning of their backlog, not just the words within it.

The primary implication of these findings is practical and economic. A 73% increase in defect detection, coupled with an 87.5% reduction in leakage, translates directly into a substantial reduction in development rework. By identifying ambiguities and conflicts before a single line of code is written, the framework prevents the exponential cost escalation of defects found later in the lifecycle, providing a clear and compelling return on investment.

The results have strong implications for the training and future definition of the Product Owner role. The high usability (88.5 SUS) suggests that proficiency with such intelligent tools will become a core competency. Future pedagogical approaches for Agile and RE should de-emphasize manual inspection techniques and instead focus on training professionals to effectively partner with and interpret the outputs of these augmented intelligence frameworks.

The success of this in-situ experiment implies a necessary evolution of Agile methodology itself. The Agile Manifesto, written in 2001, prizes "Individuals and interactions over processes and tools." This research suggests a 21st-century update is warranted: "Individuals and interactions, augmented by intelligent tools, over manual-only processes." The framework demonstrates a path to maintain Agile's flexibility while simultaneously managing the quality demands of large-scale, complex systems.

The proven accuracy and usability of the ARQ framework serve as a powerful signal to the software tool market. These findings imply that "smart-linting" for product backlogs should no longer be a third-party add-on but a core, non-negotiable feature within major project management platforms like Jira, Azure DevOps, and their competitors (Bhatia & Pallvi, 2026). Vendors now have clear empirical evidence justifying the R&D investment in integrating such NLP-driven quality assurance directly into their platforms.

The high accuracy of the framework can be attributed directly to its hybrid architectural design. The ambiguity model's 0.87 F1-score was achieved because it did not rely on a single method (Ullah et al., 2024). The fine-tuned BERT classifier provided the necessary semantic context, while the domain-specific heuristics provided a crucial layer of interpretable rules that captured common, structured errors a pure deep-learning model might miss.

The exceptional 88.5 SUS score and positive qualitative feedback are a direct result of the Design Science Research (DSR) methodology. The in-situ validation was not merely a final "test" but part of an iterative development loop (Al-Obaidy et al., 2024). The framework was built to solve a specific, felt pain-point ("cognitive load") identified in the DSR's "problem identification" phase, ensuring its features had immediate, practical relevance and were not just academically novel.

The dramatic 87.5% reduction in "defect leakage" is explainable by the framework's "shift-left" impact on the workflow. By automating the initial, systematic scan of the backlog before the refinement meeting, the ARQ framework enables defects to be found at their point of origin. This contrasts with manual processes, where defects are often only discovered during the high-pressure, time-boxed meeting itself, or worse, after the sprint has begun.

The model's ability to outperform human experts in certain "false positive" cases stems from its computational nature (Aboukadri et al., 2024). The ARQ framework is not susceptible to cognitive biases, such as confirmation bias or the "recency effect," that can affect human raters. It is, most critically, immune to fatigue. It applies the same rigorous, systematic scrutiny to the 500th user story in a backlog as it does to the first, a level of consistency that is humanly impossible to replicate.

The current framework is fundamentally diagnostic; it is highly effective at finding problems. The next logical evolutionary step for this research is to build a prescriptive framework (Raza et al., 2024). Future work should leverage generative models (e.g., T5, GPT-4) to move beyond detection, providing users with suggested corrections for ambiguous phrasing or generating high-quality, testable acceptance criteria for stories flagged as incomplete.

This study's scope was purposefully constrained to user stories and acceptance criteria. A significant opportunity for future research involves applying these NLP techniques to the "fuzzy front-end" of Requirement Engineering (Nath et al., 2024). This includes developing models to automatically extract and synthesize potential requirements from unstructured, upstream artifacts such as customer support tickets, user interview transcripts, and stakeholder meeting minutes.

The in-situ validation, while successful, was time-boxed to two sprints (four weeks). A longitudinal study is now warranted. Future research should deploy the ARQ framework with a team for a 6- or 12-month period. This would allow for the measurement of long-term impacts on team velocity, developer morale, and overall code-base quality, and would also answer questions about the tool's "novelty effect" and sustained adoption.

The current framework excels at 1:1 comparisons, such as identifying duplicate stories or conflicts. The next significant research challenge is to model the network of requirements. Future work should focus on developing graph-based NLP models that can analyze and visualize the complex, many-to-many dependencies, logical chains, and potential cascading conflicts that exist across an entire epic or product backlog.

CONCLUSION

The principal finding of this study is the critical differentiation between passive and active pedagogical approaches in shaping adolescent democratic values. Data analysis indicates that curriculum content alone, delivered through traditional rote memorization, has a negligible statistical effect on political tolerance or civic participation. The research's most distinctive discovery is that structured, peer-to-peer classroom deliberation a component of active, participatory learning emerged as the single strongest predictor of internalized democratic values. This suggests that democratic norms are not merely taught but must be actively practiced, debated, and negotiated within a social, scholastic setting to achieve meaningful internalization.

The primary contribution of this investigation is therefore conceptual, offering a refined model of political socialization for contemporary adolescents. This study moves beyond the established "knowledge-attitude-behavior" framework by empirically identifying a key mediating variable: the perceived democratic climate of the classroom itself. It posits that a student's perception of the classroom as a fair, open, and deliberative "micro-democracy" is more impactful than the formal curriculum. This research contributes a new theoretical emphasis on the process of civic education over its product, positing that the pedagogical environment functions as a "hidden curriculum" that is paramount in value formation.

The conclusions drawn from this research are subject to several important limitations. The study's cross-sectional design captures a snapshot in time and cannot definitively establish long-term causality or the durability of these values as adolescents transition into adulthood. Furthermore, the sample was drawn from a single, culturally homogeneous school district, limiting the generalizability of these findings to other national or socio-economic contexts. Future research must employ longitudinal designs to track the trajectory of these values over time. Subsequent investigations should also utilize a comparative, multi-site methodology to test the model's validity across diverse educational systems and student populations.

AUTHOR CONTRIBUTIONS

Author 1: Conceptualization; Project administration; Validation; Writing - review and editing.

Author 2: Conceptualization; Data curation; In-vestigation.

Author 3: Data curation; Investigation.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

Abbas, J., Ahmad, A., Shaheed, S. M., Fatima, R., Shah, S., Elaffendi, M., & Ali, G. (2024). Classification and Comprehension of Software Requirements Using Ensemble Learning. Computers, Materials and Continua, 80(2), 2839–2855. https://doi.org/10.32604/cmc.2024.052218

Abbasi, M., Nishat, R. I., Bond, C., Graham-Knight, J. B., Lasserre, P., Lucet, Y., & Najjaran, H. (2024). A review of AI and machine learning contribution in business process management (process enhancement and process improvement approaches). Business Process Management Journal, 31(4), 1414–1452. https://doi.org/10.1108/BPMJ-07-2024-0555

Aboukadri, S., Ouaddah, A., & Mezrioui, A. (2024). Machine learning in identity and access management systems: Survey and deep dive. Computers & Security, 139, 103729. https://doi.org/10.1016/j.cose.2024.103729

Al-Obaidy, H., Ebrahim, A., Aljufairi, A., Mero, A., & Eid, O. (2024). Software Engineering for Developing a Cloud Computing Museum-Guide System. International Journal of Cloud Applications and Computing, 14(1). https://doi.org/10.4018/IJCAC.339200

Ananikov, V. P. (2024). Top 20 influential AI-based technologies in chemistry. Artificial Intelligence Chemistry, 2(2), 100075. https://doi.org/10.1016/j.aichem.2024.100075

Anwar, Z., Bibi, N., Rana, T., Kadry, S., & Afzal, H. (2024). Collaborative Solutions to Software Architecture Challenges Faced by IT Professionals. International Journal of Human Capital and Information Technology Professionals, 15(1). https://doi.org/10.4018/IJHCITP.342839

Ben Aoun, R., Hameed, M., Omar, M. B., Rafi-ul-Shan, P. M., Castagnola, E., Karahmet Sher, E., Ahmed, R., & Razmkhah, O. (2026). Chapter 18—Artificial intelligence for regulatory compliance in chemical engineering industries. In F. Sher (Ed.), Artificial Intelligence in Chemical Engineering (pp. 555–592). Elsevier. https://doi.org/10.1016/B978-0-443-34076-5.00015-8

Bhatia, M. & Pallvi. (2026). The integration of emerging technologies in defense: A scientometric overview. Engineering Applications of Artificial Intelligence, 163, 112822. https://doi.org/10.1016/j.engappai.2025.112822

Chang, A. C., & Limon, A. (2024). Chapter 1—Introduction to artificial intelligence for cardiovascular clinicians. In A. C. Chang & A. Limon (Eds.), Intelligence-Based Cardiology and Cardiac Surgery (pp. 3–120). Academic Press. https://doi.org/10.1016/B978-0-323-90534-3.00010-X

Chou, J.-R. (2024). An integrative review exploring the development of sustainable product design in the technological context of Industry 4.0. Advanced Engineering Informatics, 62, 102689. https://doi.org/10.1016/j.aei.2024.102689

Cunningham, J. W., Abraham, W. T., Bhatt, A. S., Dunn, J., Felker, G. M., Jain, S. S., Lindsell, C. J., Mace, M., Martyn, T., Shah, R. U., Tison, G. H., Fakhouri, T., Psotka, M. A., Krumholz, H., Fiuzat, M., O'Connor, C. M., & Solomon, S. D. (2024). Artificial Intelligence in Cardiovascular Clinical Trials. Journal of the American College of Cardiology, 84(20), 2051–2062. https://doi.org/10.1016/j.jacc.2024.08.069

David, I., & Gelbard, R. (2024). Using sentiment analysis to assess PMBOK knowledge areas' compatibility with agile methodology. CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information

Systems and Technologies 2023, 239, 381–395. https://doi.org/10.1016/j.procs.2024.06.185

Eramo, R., Tucci, M., Di Pompeo, D., Cortellessa, V., Di Marco, A., & Taibi, D. (2024). Architectural support for software performance in continuous software engineering: A systematic mapping study. Journal of Systems and Software, 207, 111833. https://doi.org/10.1016/j.jss.2023.111833

Fadhel, M. A., Duhaim, A. M., Saihood, A., Sewify, A., Al-Hamadani, M. N. A., Albahri, A. S., Alzubaidi, L., Gupta, A., Mirjalili, S., & Gu, Y. (2024). Comprehensive systematic review of information fusion methods in smart cities and urban environments. Information Fusion, 107, 102317. https://doi.org/10.1016/j.inffus.2024.102317

Fairil, A., & Tobroni, I. (2024). Geographic Information System for Shortest Route Search for Clinics in Pamekasan Regency Using the Djikstra Method. Journal of Computer Science Advancements, 1(6), 268–278. https://doi.org/10.70177/jsca.v1i6.1140

Fonseca i Casas, P., & Pi i Palomes, X. (2026). Building Society 5.0: A foundation for decision-making based on open models and digital twins. Advanced Engineering Informatics, 69, 103970. https://doi.org/10.1016/j.aei.2025.103970

Gao, R. X., Krüger, J., Merklein, M., Möhring, H.-C., & Váncza, J. (2024). Artificial Intelligence in manufacturing: State of the art, perspectives, and future directions. CIRP Annals, 73(2), 723–749. https://doi.org/10.1016/j.cirp.2024.04.101

Hakiri, A., Gokhale, A., Yahia, S. B., & Mellouli, N. (2024). A comprehensive survey on digital twin for future networks and emerging Internet of Things industry. Computer Networks, 244, 110350. https://doi.org/10.1016/j.comnet.2024.110350

Hasanah, I. U., Tabroni, I., Brunel, B., & Alan, M. (2023). Development of Media Matching Box to stimulate symbolic thinking skills in children aged 4-5 years. Journal of Computer Science Advancements, 1(1), 1–13. https://doi.org/10.55849/jsca.v1i1.442

Hughes, L., Davies, F., Li, K., Gunaratnege, S. M., Malik, T., & Dwivedi, Y. K. (2026). Beyond the hype: Organisational adoption of Generative AI through the lens of the TOE framework–A mixed methods perspective. International Journal of Information Management, 86, 102982. https://doi.org/10.1016/j.ijinfomgt.2025.102982

Izzah, N., & Jannah, A. B. (2024). Optimisation of Evacuation Route Determination in an Earthquake Natural Disaster Scenario Using an Excel Solver. Journal of Computer Science Advancements, 1(6), 259–267. https://doi.org/10.70177/jsca.v1i6.1132

Jin, L., Zhai, X., Wang, K., Zhang, K., Wu, D., Nazir, A., Jiang, J., & Liao, W.-H. (2024). Big data, machine learning, and digital twin assisted additive manufacturing: A review. Materials & Design, 244, 113086. https://doi.org/10.1016/j.matdes.2024.113086

Nath, P. C., Mishra, A. K., Sharma, R., Bhunia, B., Mishra, B., Tiwari, A., Nayak, P. K., Sharma, M., Bhuyan, T., Kaushal, S., Mohanta, Y. K., & Sridhar, K. (2024). Recent advances in artificial intelligence towards the sustainable future of agri-food industry. Food Chemistry, 447, 138945. https://doi.org/10.1016/j.foodchem.2024.138945

Nopiyanti, H., Tabroni, I., Barroso, U., & Intes, A. (2023). Product Development of Unique Clothing Learning Media to Stimulate Fine Motor Skills of 4-5 Years Old Children. Journal of Computer Science Advancements, 1(1), 48–61. https://doi.org/10.55849/jsca.v1i1.452

Padovano, A., & Cardamone, M. (2024). Towards human-AI collaboration in the competency-based curriculum development process: The case of industrial engineering and

management education. Computers and Education: Artificial Intelligence, 7, 100256. https://doi.org/10.1016/j.caeai.2024.100256

Pradhan, M., Hasso, H., Popescu, A., & Müller, C. (2026). Chapter 6—Smart cities: The future with Artificial Intelligence. In M. Pradhan (Ed.), Meeting SDGs in Smart City Infrastructures (pp. 151–189). Elsevier. https://doi.org/10.1016/B978-0-44-326594-5.00015-X

Rahayu, S. S., Tabroni, I., Martin, J., & Fang, W. (2023). Number Rinner Games To Improve 5-6 Year-Old Counting Ability. Journal of Computer Science Advancements, 1(1), 37–47. https://doi.org/10.55849/jsca.v1i1.443

Raza, A., Jingzhao, L., Ghadi, Y., Adnan, M., & Ali, M. (2024). Smart home energy management systems: Research challenges and survey. Alexandria Engineering Journal, 92, 117–170. https://doi.org/10.1016/j.aej.2024.02.033

Shahin, M., Chen, F. F., & Hosseinzadeh, A. (2024). Harnessing customized AI to create voice of customer via GPT3.5. Advanced Engineering Informatics, 61, 102462. https://doi.org/10.1016/j.aei.2024.102462

Siddique, I., Farooq, M., Ullah, Q., Nabi, A., Siddique, M., Siddique, S., Naseer, M., Younis, A., & butt, W. (2026). Chapter 44—Future trends and direction in AI and ML for food science and bioprocess development. In T. Sarkar & A. Haldorai (Eds.), Artificial Intelligence in Food Science (pp. 811–825). Academic Press. https://doi.org/10.1016/B978-0-443-26468-9.00080-1

Tangwaragorn, P., Charoenruk, N., Viriyasitavat, W., Tangmanee, C., Kanawattanachai, P., Hoonsopon, D., Pungpapong, V., Pattanapanyasat, R.-P., Boonpatcharanon, S., & Rhuwadhana, P. (2024). Analyzing key drivers of digital transformation: A review and framework. Journal of Industrial Information Integration, 42, 100680. https://doi.org/10.1016/j.jii.2024.100680

Teresia, V., Jie, L., & Jixiong, C. (2023). Interactive Learning Media Application For The Introduction Of Human Needs In Children Aged. Journal of Computer Science Advancements, 1(1), 25–36. https://doi.org/10.55849/jsca.v1i1.406

Ullah, H., Uzair, M., Jan, Z., & Ullah, M. (2024). Integrating industry 4.0 technologies in defense manufacturing: Challenges, solutions, and potential opportunities. Array, 23, 100358. https://doi.org/10.1016/j.array.2024.100358

Wisdom, D. D., Vincent, O. R., Aborisade, D. O., & Omeike, M. O. (2026). Chapter 11—Defensive walls against sophisticated ML-orchestrated attacks in edge computing. In A. L. Imoize, M. S. Obaidat, & H. H. Song (Eds.), Cybersecurity Defensive Walls in Edge Computing (pp. 275–315). Academic Press. https://doi.org/10.1016/B978-0-443-34109-0.00004-8

Zhao, J., Feng, X., Pang, Q., Fowler, M., Lian, Y., Ouyang, M., & Burke, A. F. (2024). Battery safety: Machine learning-based prognostics. Progress in Energy and Combustion Science, 102, 101142. https://doi.org/10.1016/j.pecs.2023.101142