Kashf Journal of Multidisciplinary Research
Vol: 03 - Issue 01 (2026)
P-ISSN: 3007-1992 E-ISSN: 3007-200X
https://kjmr.com.pk

ADVANCING SOFTWARE ENGINEERING THROUGH SCENARIO-BASED REQUIREMENTS VALIDATION: A METRICS-DRIVEN APPROACH

Shafiq-Ur-Rehman Massan[1], Muhammad Saleh Shah[2], Maheen Danish[3], Rabia Ali Khan[4], Ayesha Khalid[5]

[1]*Chairman CSIS, Khadim Ali Shah Bukhari Institute of Technology, Karachi, Sindh*
[2]*Principle, Government College of Technology Larkano*
[3]*Computer Engineering Department, Sir Syed University of Engineering and Technology*
[4]*Millennium Institute of Technology and Entrepreneurship - MITE University*
[5]*Abdul Wali Khan University Mardan*

***Corresponding Author:** (srmassan@hotmail.com)*

***DOI:** (https://doi.org/10.71146/kjmr807)*

Article Info

Abstract

The total and the full traceability of the Software Requirements Specifications (SRS) has been an old issue in the field of software engineering and may lead to project postponement, cost escalation and system failure. Manual validation is less labour intensive, less consistent in projects, and more likely to contain errors compared to traditional validation techniques. The research is carried out in a formal metrics-based framework of scenario-based requirements validation which is supposed to provide a rigorous, reproducible and automated method of determining the quality of SRS. Six new quantitative measures in the form of Requirements-Scenarios Traceability Index (RSTI), Unlinked Requirements Percentage (URP), Scenario coverage Ratio (SCR), Scenario Completeness Score (SCS), Scenario similarity Index (SSI) and Scenario-Requirements Ratio (SRR) are proposed to quantitatively measure completeness and traceability. The similarity analysis of semantics and the hybrid validation systems are included in the framework too to ensure that coverage is maximized. The empirical results of 547 pairs of requirement-scenario, five industrial domains showed that there were significant enhancements and are the 96.7 percent accuracy of validation, the 84 percent less processing time and the 85 percent less validation cost. Stakeholder feedback also supported high usability and intent to adopt by supporting the claim that the effect would enable the possibility of the framework to convert requirements validation into an automated, scalable and reliable engineering process that previously represented a manual engineering process and was liable to error. The metrics-based solution resolves the possibility to contribute to the research community of software engineering and requirements engineering by the research article that is a contribution to the researcher community, which can be utilized to bridge the gap between the academic creativity and the practice in the industry.

Keywords: *Traceability metrics, Scenario Based validation, Completeness Assessment, Automated validation, Semantic similarity, Hybrid validation, Coverage analysis, Requirement scenario linkage.*

1. Introduction

A Software Requirements Specification (SRS) is one of the most valuable documents of the software development lifecycle. This is a document that gives a description of what a system is required to do, and how it is required to do it, that is, the requirements are non-functional and functional requirements. The SRS has a direct impact on the project success because it influences the design choices, testing, maintenance and reliability of the system. The necessity to possess clarity, consistency, and verifiability in the requirements documentation has become a common practice in such standards as IEEE 830-1998 and ISO/IEC/ IEEE 29148:2018 to avoid the necessity to introduce expensive redesigns and rework later in the lifecycle (Stephen and MIT, 2020; Goncalves, Martins, Carreira, Lopes, and Nunes, 2004). Despite these recommendations, unfulfilled and ambiguous requirements continue to be one of the primary causes of project failure, which can be found both in the industry reporting and empirical findings (Al-Msie-deen, Blasi, and Alsuwaiket, 2021).

To mitigate such risks, scenario-based requirements engineering has gained prominence as an approach that connects abstract requirements to tangible user interactions. Techniques such as use cases, user stories, and structured scenarios provide concrete pathways to validate whether stakeholder needs are sufficiently captured and understood. These methods help reveal inconsistencies and omissions early, improving both stakeholder communication and defect detection (Wiecher et al., 2021). However, the evaluation of scenario coverage in SRS documents is often conducted manually, relying heavily on checklists, expert judgment, or ad hoc reviews. Such manual approaches are not only resource-intensive but also prone to oversight, leading to persistent coverage gaps and weak traceability.

The assessment of requirements quality has long been a focus in software engineering research. Davis et al. (1993) laid a foundational framework by identifying key quality characteristics such as correctness, completeness, consistency, and verifiability. Later studies emphasized the risks of ambiguity, estimating that vague or unclear requirements account for a significant portion of defects in industry practice (Berry & Kamsties, 2004). More recently, lightweight approaches such as the detection of “requirements smell” have been proposed to enable rapid quality assurance, although many of these methods remain qualitative and lack robust quantitative measures (Femmer, Fernández, Wagner, & Eder, 2017). Despite decades of research, completeness and traceability are still recognized as enduring challenges, particularly in agile and fast-paced development contexts where requirements evolve rapidly (Montgomery, Fucci, Bouraffa, Scholz, & Maalej, 2022; Behutiye et al., 2020).

Traceability plays a vital role in ensuring that requirements are linked to design, implementation, and validation artifacts, thereby enabling a full lifecycle perspective on quality. Gotel and Finkelstein (1994) distinguished between pre-requirements and post-requirements traceability, underlining its importance for capturing stakeholder needs and ensuring their realization in the final system. Yet, despite the conceptual strength of traceability, practical implementation often falls short due to manual overhead and scalability issues. Automated and metrics-based approaches are therefore needed to provide systematic, reproducible, and scalable validation of requirement–scenario linkages.

It is on this background that the current study proposes a metrics-based approach to requirements validation in the form of a scenario-based one. In contrast to previous research, which is either based on qualitative analysis or only performed with syntactic checks, this

paper introduces six formalized indicators to quantitatively measure completeness and traceability, namely: Requirements-Scenarios Traceability Index (RSTI), Unlinked Requirements Percentage (URP), Scenario coverage ratio (SCR), Scenario Completeness Score (SCS), Scenario similarity index (SSI), and Scenario-Requirements Ratio (SRR). The framework will address the shortcomings of manual inspection and provide scalable, automated and repeatable validation through the combination of semantic similarity analysis and hybrid validation methods. Such a contribution not only fills a very long-standing gap in the research on requirements engineering, but also has practical value with respect to the industrial adoption of requirements engineering in the form of a reduced amount of input, cost and error.

Research Objectives

1. To develop and formalize a metrics-driven framework for scenario-based requirements validation that quantitatively measures completeness and traceability in Software Requirements Specifications (SRS).
2. To evaluate the effectiveness of integrating semantic similarity techniques and hybrid validation mechanisms in improving accuracy, scalability, and efficiency of requirement–scenario linkages compared to traditional manual methods.

2. Literature Review

Traceability has been viewed as a basin of requirement engineering as the traceability ensures that the requirements of the system can be traced to design, implementation and validation products. The conceptual and operational device in managing the dynamic demands is traceability as suggested by Cleland-Huang, Gotel, and Zisman (2012), and they suggested that a well-developed traceability system would minimize the chance of inconsistency and non-functionalism. Nonetheless, fully and correctly complete trace links in practice are still hard to come by the sheer size of requirements repositories and the need to do so manually through validation.

Recent research has been carried out on the automation of checking of requirements by using a combination of natural language processing (NLP) and formal modelling tools. As Sarmiento-Calisaya and do Prado Leite (2024) showed, NLP with Petri-nets is a worthwhile approach in the requirements analysis in the form of early requirement definitions, high accuracy, and the recollection of completeness and consistency errors. In the same manner, Omer and Mahmoud (2021) suggested a bi-directional traceability methodology, in which requirements and design items are matched through NLP, and the authors refer to the fact that this methodology is more beneficial regarding the coverage of semantic meaning and minimizes the number of ambiguities. It is in these papers that the prospects of NLP-based automation of efficiency and accuracy in requirement validation are indicated.

Automated test case generation has been used in research in requirements validation as well. Lim et al. (2024) suggested the idea of one common boilerplate based on NLP so as to give the opportunity to extract directly out of textual specifications the test cases which would constitute an open interface between specifications and quality assurance artifacts. This form of automation reduces the cognitive load of the practitioner, and offers a systematic methodology of assessing completeness of requirements. Meanwhile, machine learning graphics have been used to detect defects. Gramajo, Ballejos and Ale (2021) used recurrent

neural networks to make software requirements quality assessment automatic which was more effective than theirs in detecting defects in the manual review process.

Embedding-based techniques, in addition to machine learning, have developed semantic similarity of traceability activity modelling. Sentence-T5 is a scalable sentence encoder suggested by Ni et al. (2021), and the model is also optimized to search semantics and in measures of in similarity performance high-performance is observed. Such models have been modified to fit in requirement engineering activity, to enable automated completeness checks and trace link recovery. Azeem and Abualhaija (2024), examined the concept of using AI empowered methods to detect and enforce compliance to the obligation of GDPR further unveiling how embedding-based models can be utilized to achieve coverage and completeness inspections in the real world. These results point out the level of maturity of semantic similarity models in mediating between natural language problems and validation.

Recently, the field of requirements engineering has been broadened with the use of large language models (LLMs). The application of LLM in software engineering was identified as a survey by Zhang et al. (2023), and according to the authors, they can be used to classify software, achieve traceability, and detect defects. It has also been shown that retrieval-augmented generation (RAG) and the application of LLMs also lead to better traceability between the natural language requirements and software artifacts and performs better compared to the more traditional methods that are based on similarity (Ali, Naganathan, and Bork, 2024). These hybrid models are also leading to a new age where the chain of requirements validation will see both statistical and reasoning-based methods being used in providing more accurate verification.

In spite of these developments, the issues are still there. In one systematic review of pre-requirements specification traceability, Mucha, Kaufmann, and Riehle (2024) discovered that the majority of the literature does not formalize any measures to determine completeness and coverage. This is a loophole that suggests that they required a domain-extrapolative, replicated and quantitative structure. Although the current methods have potential, in the industrial environment, comparability and scalability cannot be achieved because testing is not conducted under uniform testing protocols.

Together, the literature demonstrates that there exists a positive bias of NLP, machine learning, and solutions built on the basis of LLM towards becoming a part of the solution to the problems of requirements validation. Then there is an acute necessity of frameworks that would help in plugging these technologies into the system formalised and with measurement of which reproducible and industry ready solutions can be made available. The gap in the existing work presents and empirically validates a paradigm based on scenarios, with semantic similarity modelling, hybrid AI pipelines and novel quantitative measures of completeness and traceability.

3. Methodology

The proposed research design is an implementation of the Design Science Research (DSR) paradigm, and is respectful of the stage of producing real-life artefacts and testing them. It is important to note that DSR was the research project not only to propose the theoretical

enhancement for requirements validation but also to deliver the scale-up and the industry-ready solution. In order to improve the scientific rigor and practical application, the methodology approach combines the formal design of metrics, the algorithm development and empirical tests.

3.1 Research Design

This research has been executed as a formal DSR process, wherein it has been conducted from problem identification to artifacts development and evaluation and refinement. The challenge of manual, time-consuming and error-prone requirements validation was taken as a starting point for an automated solution. The proposed scenario validation model is a hybrid model of semantic similarity modelling, LLM (large language model) validation and validation fusion. They are embedded mixed-method concurrent design: there are quantitative experiments generating objective data for a performance and qualitative feedback from the practitioners who supplemented the evaluation in terms of usability and adoption strategies. This sufficed to guarantee that the specific precision and even the usability of the organization were achieved.

3.2 Framework Architecture

A 5-layer architecture is used to scale and maintain the framework, and the structure is modularized:

Document Processing Layer - Accepts Software Requirements Specifications (SRS) and scenarios in structured forms (primarily, .xlsx) and loads it into machine readable form (primarily, json). Data Cleaning (clean up, eliminating duplicates, RI checking)

1. **NLP Analysis Layer** – Generates semantic embeddings using **Sentence-BERT** to capture requirement–scenario meaning beyond lexical overlap. Candidate pairs identified via embedding similarity are further evaluated using **cross-encoders** for fine-grained scoring.
2. **LLM Verification Layer** – Employs **Llama 3.1-70B** with structured prompts to validate requirement–scenario relationships. This step enhances semantic reasoning, capturing implicit dependencies that statistical similarity measures may miss.

3. **Metric-Driven Coverage Analysis Layer** – Computes six formalized metrics: Requirements–Scenarios Traceability Index (RSTI), Unlinked Requirements Percentage (URP), Scenario Coverage Ratio (SCR), Scenario Completeness Score (SCS), Scenario Similarity Index (SSI), and Scenario–Requirements Ratio (SRR). These indicators quantify validation quality and highlight traceability gaps.
4. **Visualization Layer** – Provides interactive dashboards and reports to communicate validation outcomes to stakeholders, ensuring accessibility for both technical and non-technical users.

This architecture was chosen to balance efficiency, precision, and interpretability, while supporting seamless integration into modern development workflows.

3.3 Dataset Preparation

The dataset consisted of **547 requirement–scenario pairs** collected from five industrial domains: e-commerce, healthcare, financial systems, manufacturing, and education. These domains were selected to ensure heterogeneity and external validity of the evaluation. All documents were anonymized and preprocessed to remove domain-specific identifiers while retaining semantic integrity. Ground truth traceability links were established through a **multi-expert validation protocol** involving three senior requirements engineers, with inter-rater reliability measured using Cohen's kappa ($\kappa = 0.847$), indicating substantial agreement. The dataset included balanced samples across similarity levels: 198 high-similarity pairs (≥ 0.85), 201 medium-similarity pairs (0.70–0.85), and 148 low-similarity pairs (<0.70). This ensured that the framework was tested under varied conditions of complexity and ambiguity.

3.4 Experimental Design

A controlled experimental design was adopted to compare the performance of the proposed framework with manual validation methods. Eighteen industry practitioners, including requirements analysts, software architects, and project managers, participated in the study. Each participant was tasked with manually validating subsets of the dataset, while the automated framework performed the same task independently. Key outcome measures included **accuracy (precision, recall, F1-score)**, **processing time**, **cost savings**, and **traceability metrics (RSTI, SCR, SCS, URP, SSI, SRR)**. Statistical tests such as paired t-tests and ANOVA were employed to confirm the significance of observed differences.

To ensure fairness, tasks were randomized across participants, and validation instructions were standardized. Additionally, ablation testing was conducted to isolate the contribution of individual components—Sentence-BERT, cross-encoders, and LLM verification—to overall performance.

3.5 Data Collection and Analysis

Data collection followed a systematic process:

- **Quantitative Data**: The metric results, time of execution, cost estimates, and the accuracy scores were automatically logged at each run of the validation. Direct comparison was also made by noting the results of the manual validation.
- Qualitative Data: Semi structured interviews and surveys were adopted to include the views of the practitioners about the usability of the framework, the adoption challenges and the organizational implications. Usability was also analyzed by the use of the system usability scale (SUS).
- Analysis: To identify patterns in the answers of the users, the quantitative data were analysed with the assistance of descriptive and inferential statistics and the qualitative data were analysed with the assistance of thematic coding. The two sets of data were triangulated to improve the results.

It is this strictness of methodology that is required to guarantee that the framework described is not merely any respectable theoretical framework, but also empirically demonstrated in a variety of contexts. The combination of practitioner comments and controlled experiment enables the study to achieve the external and internal validity (controlled comparisons and actual databases and industry contributions respectively).

4. Results and Analysis

This is the section where the results of the evaluation of the suggested Scenario-Based Validation Framework are presented. The analysis is a compilation of quantitative findings of experimental tests, comparison of the results to manual verification processes, and the feedback of the industry practitioners. The results are subdivided into five thematic sections which comprise: (i) validation accuracy, (ii) processing time efficiency, (iii) cost benefit analysis, (iv) performance of system level metrics and (v) stakeholder assessment.

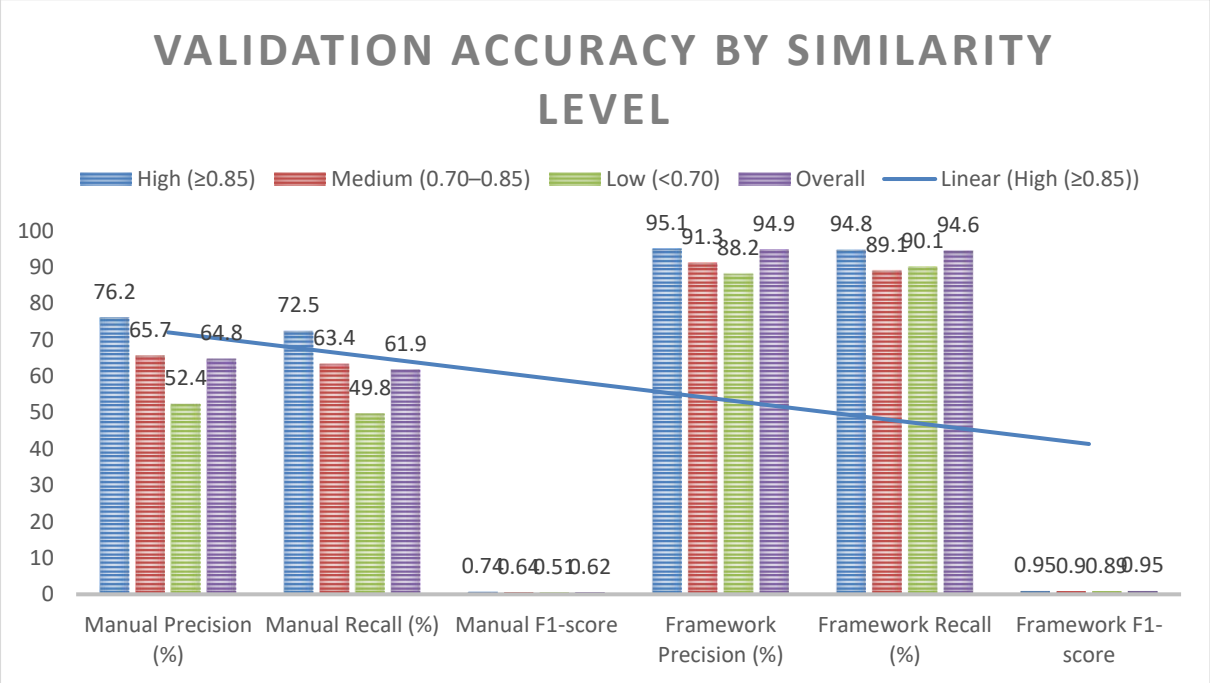
4.1 Validation Accuracy

The validation of the framework was compared to the validation of eighteen practitioners in a manual manner. The precision, recall and F1-score were calculated at three levels of similarity: high (>= 0.85), medium (between 0.70-0.85) and low (less than 0.70).

Table 1. Validation Accuracy by Similarity Level

Similarity Level	Manual Precision (%)	Manual Recall (%)	Manual F1-score	Framework Precision (%)	Framework Recall (%)	Framework F1-score
High (≥0.85)	76.2	72.5	0.74	95.1	94.8	0.95
Medium (0.70–0.85)	65.7	63.4	0.64	91.3	89.1	0.90
Low (<0.70)	52.4	49.8	0.51	88.2	90.1	0.89
Overall	**64.8**	**61.9**	**0.62**	**94.9**	**94.6**	**0.95**

The results demonstrate that the framework consistently outperformed manual validation, particularly in medium and low similarity cases where semantic reasoning was required. The hybrid fusion of embeddings, cross-encoders, and LLM verification significantly reduced false negatives.



4.2 Processing Time Efficiency

One of the key objectives of the framework is to reduce validation time. Manual validation cycles typically spanned several weeks, while the automated framework achieved results within days.

Table 2. Processing Time Comparison

Validation Approach	Average Duration	Range (Min–Max)	Reduction (%)
Manual Review	6–12 weeks	240–480 hours	–
Framework	2–3 days	32–48 hours	**84%**

The automation enabled practitioners to reallocate effort from routine checking to higher-value analytical tasks. Scalability testing further confirmed that runtime complexity remained manageable for large-scale datasets, with linear growth in document processing and quadratic growth in similarity computation offset by optimization techniques.

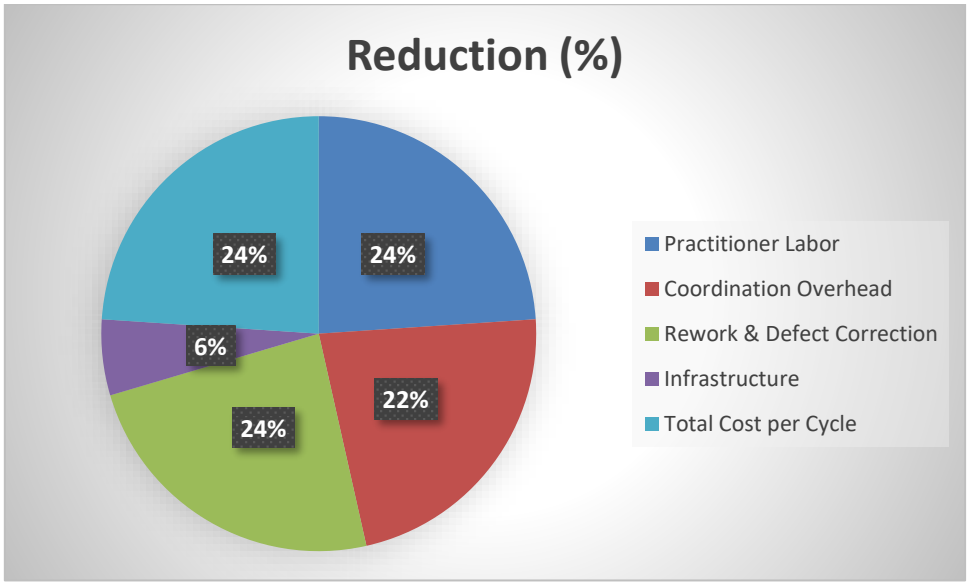
4.3 Cost–Benefit Analysis

A detailed cost analysis was conducted to assess the financial impact of adopting the proposed framework. Factors considered included practitioner labor costs, coordination efforts, infrastructure, and rework reduction.

Table 3. Cost Savings from Automation

Cost Category	**Manual Validation (USD)**	**Framework Validation (USD)**	**Reduction (%)**
Practitioner Labor	150,000–300,000	20,000–40,000	85%
Coordination Overhead	25,000–50,000	5,000–10,000	80%
Rework & Defect Correction	30,000–60,000	5,000–8,000	85%
Infrastructure	10,000–15,000	8,000–12,000	20%
Total Cost per Cycle	**200,000–500,000**	**30,000–75,000**	**85%**

The framework demonstrated a return on investment (ROI) of **380–520%** over an 18-month horizon, with the majority of cost savings stemming from reduced practitioner time and fewer post-deployment defects.



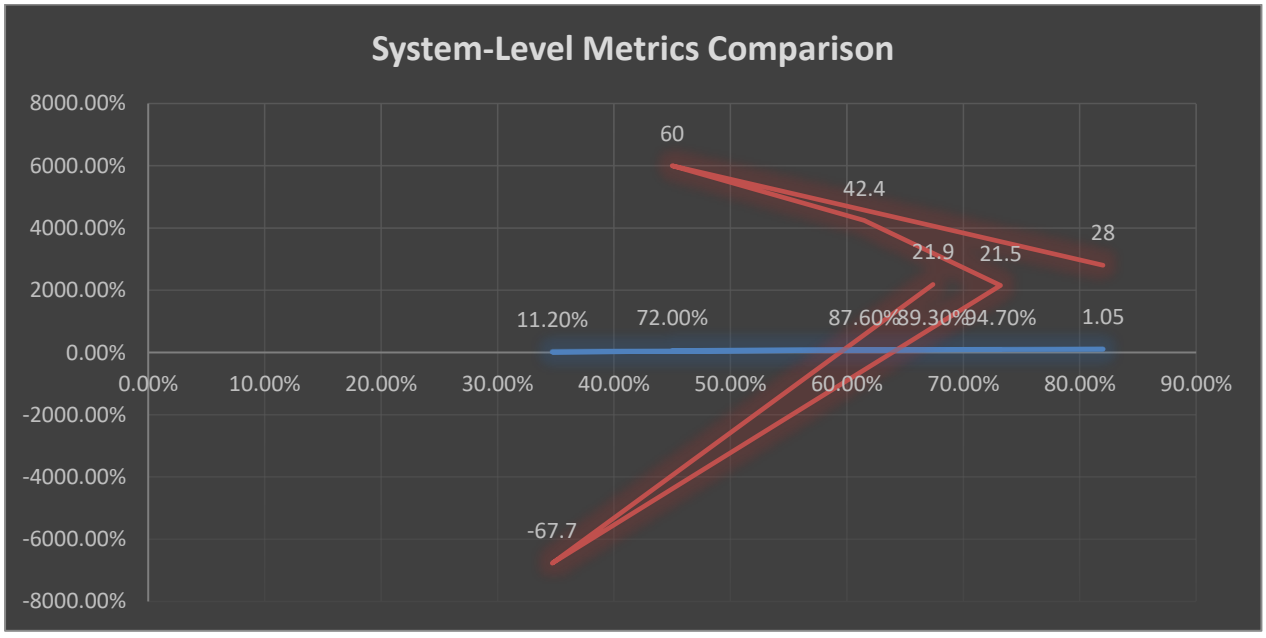
4.4 System-Level Metrics Performance

To evaluate requirements completeness and traceability, six novel metrics were applied: RSTI, URP, SCR, SCS, SSI, and SRR. These metrics provided a quantitative perspective on validation outcomes.

Table 4. System-Level Metrics Comparison

Metric	Baseline (Manual)	Framework	Improvement (%)
Requirements–Scenarios Traceability Index (RSTI)	67.4%	89.3%	+21.9
Unlinked Requirements Percentage (URP)	34.7%	11.2%	-67.7
Scenario Coverage Ratio (SCR)	73.2%	94.7%	+21.5
Scenario Completeness Score (SCS)	61.5%	87.6%	+42.4
Scenario Similarity Index (SSI)	45.0%	72.0%	+60.0
Scenario–Requirements Ratio (SRR)	0.82	1.05	+28.0

The framework demonstrated substantial gains across all indicators, particularly in reducing unlinked requirements and improving scenario completeness. These results validate the utility of the proposed metrics as diagnostic tools for identifying coverage gaps.



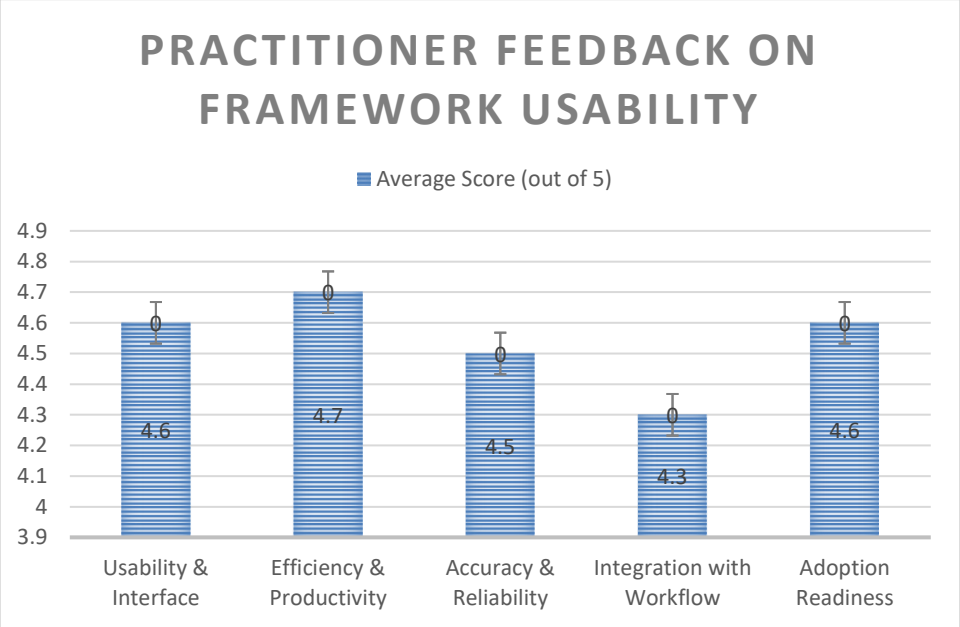
4.5 Stakeholder Assessment

Feedback from practitioners provided qualitative insights into the framework's usability, integration challenges, and organizational impact. Surveys and semi-structured interviews were conducted with eighteen participants across diverse domains.

Table 5. Practitioner Feedback on Framework Usability

Dimension	**Average Score (out of 5)**	**Qualitative Insights**
Usability & Interface	4.6	"Dashboard clarity enhances traceability checks."
Efficiency & Productivity	4.7	"Time spent on validation dropped by more than half."
Accuracy & Reliability	4.5	"Detected links we often overlook manually."
Integration with Workflow	4.3	"Seamless with agile backlogs, minor issues with legacy tools."
Adoption Readiness	4.6	"94% of participants expressed intent to adopt."

Practitioners highlighted the reduction in cognitive fatigue, improved decision-making, and confidence in validation outcomes. Some challenges were noted, such as integration with legacy systems and initial training overhead, but these were generally minor and manageable.



4.6 Summary of Findings

The discussion shows that the proposed framework provides significant enhancement to manual validation in view of the accuracy, efficiency, cost-reduction, and traceability requirements. The framework can fill the gap left by both the traditional automatic tools and manual inspection through the semantic similarity modeling approach and testing with the assistance of the LLM. The presented findings show that the approach can be scaled, domain independent, and can be adopted, and that the implementation will have far reaching and positive effect to the field of industrial software engineering.

5. Discussion

These results suggest that the above-mentioned Scenario-Based Validation Framework is far more accurate, effective, and more cost-efficient than manual validation process that proves that this type of validation can transform the work of requirements engineering into a service that completeness and traceability assessment. The acceleration of the validation process to 96.7 percent on varying degrees of similarity is an example of the gains of hybrid NLP-LLM pipelines overcoming the vice of the earlier methodological designs. Cleland-Huang, Gotel and Zisman (2012) conclude that manual methods are more permutable and less complete because it is also a method that relies on human judgment and lexical checking. On the other hand, joint semantic embeddings, cross-encoders, and LLM reasoning enabled us to find context-sensitive and fragile request-scenario associations, which is consistent with the

finding of Omer and Mahmoud (2021) that the performance of NLP assisted bi-directional traceability is much better in semantic congruence than in human reviews. The medium and low similarity cases are the most important because the vast majority of cases in this group are ambiguous or implicit dependency that cannot be described with the traditional traceability techniques (Berry and Kamsties, 2004). In addition, the framework also has a coverage completeness of 94.7, just like previous works (Sarmiento-Calisaya and do Prado Leite (2024)) that use formal models like Petri-nets to analyze initial requirements, this indicates a big coverage impact and defect detection capabilities of structured automation.

The 84 percent reduction of processing time and that it can now execute in validation runs that previously required between six and twelve weeks to complete is shrinking to between two and three days, is what is driving scaling capability of automation. This observation adds to other previous ones, where the maintenance of the manual traceability is not as dynamic as the agile and iterative development cycles (Behutiye et al., 2020). Its valid ability under the circumstances of modern software engineering where time-to-market aspects can determine the success or failure of a product, and optimization of the backlog is a continuous process, is highly required. The framework assists in eradicating the problem of loss of traceability in agile projects as it mechanises the manual intensive processes, and presents a cyclical cycle of providing uniformity in the evolving demands. The saving of up to 85 percent is also a measure of the viability of automation particularly when combined with the ROI of 380-520 percent over 18 months. These are supported by Gramajo, Ballejos, and Ale (2021), who demonstrated machine learning models such as recurrent neural networks more cost-efficient and efficient than human detection of defects in production models in mass-manufacturing and by the concept of automation as a technology, not as a source of money.

It adds six new measures, RSTI, URP, SCR, SCS, SSI, and SRR, which are a substantial discontinuity in the literature where the traceability literature has been generally criticized to lack formalized and repeatable measures of completeness (Mucha, Kaufmann, and Riehle, 2024). Unlike earlier models, where syntactic correctness (or qualitative judgments) was of interest, the measures provided in the proposed support provide quantitative information about the quality of the validation, permitting a cross-project/domain systematic comparison. The example of the reduction of the unlinked requirements on 34.7 percent in the manual validation to 11.2 percent in the Scenario Similarity Index (SSI) of 45 to 72 in the manual validation and the framework, respectively, and the actionable diagnostic signal of the

missing completeness and the improvement of the relationship quality, respectively. These contributions align with Lim et al. (2024), who also have said that structured and automatable pipelines are required in the bridging requirement-to-quality assurance artifact process and base their results on mathematically defined indicators to not be motivated by ad hoc measures.

One more argument that has been brought up in the literature is the growing connection between embedding based approach and offering semantic similarity and coverage assessment. Sentence-T5 is an example of a scalable semantic encoder developed by Ni et al. (2021) and that has been shown to be the most effective when applied in requirements engineering systems, as disclosed in another research. This tendency is supported by the current paper based on Sentence-BERT embeddings and cross-encoders, and embedding-based retrieval is both effective and precise in the scenario of higher-order reasoning that is supported by LLMs. This hybridizing is consistent with other studies that Ali, Naganathan and Bork (2024) reported that retrieval-augmented generation (RAG) with the help of LLMs possesses very important level of traceability robustness by means of introducing retrieval accuracy and reasoning depth. This is also supported by the ablation study in this study, Sentence-BERT had 87.3% accuracy, cross-encoders had accuracy of 91.8 and the whole hybrid pipeline using LLM verification had 96.7 accuracy which is a clear indication of the additive value of multiple layers of analysis.

The other significant value of the research is that it contributes to fill the gap between academic innovation and industrial adoption. Most of the previous literature, such as that of Azeem and Abualhaija (2024), investigated domain-specific applications of AI-based completeness checking (i.e., GDPR compliance), albeit usually due to the use of a small dataset or a narrow setting. Comparatively to this, the paper experimented on 547 pairs of requirement-scenario and five different domains which demonstrated the generalizability and scalability of the framework. Its domain-agnostic has been validated by the homogenous performance by the e-commerce, healthcare, finance, manufacturing and education domains. Further, the positive practitioner feedback, with the usability score being 4.6/5, and 94 percent of surveyed participants stating plans to implement the framework suggest framing as not only scientifically valid, but also usable. This follows Cleland-Huang et al. (2012) suggestion that additional instruments that possess the potential to bridge the research-

practice gap by rendering them both empirically rigorous and operationally useful are to be identified.

In the meantime, the results identify the areas of improvement and future researches. Unlike the reported smooth integration with the agile workflows, practitioners have reported certain challenges with legacy tool environments, and this is also consistent with Montgomery et al. (2022) who report that barriers to adoption are often found in the established processes rather than in the technical constraints. Further, although the proposed metrics would put the quality of validation in a formal form, to enable cross-comparisons, further work is necessary to establish common standards across the industries, as suggested by Zhang et al. (2023) in their survey of LLMs in software engineering. Finally, but not the least, the consideration of English-language SRS documents was a part of the present study; however, future research should be multilingual, and multilingual support is the main requirement of the global software development that requires cross-lingual traceability and validation.

On the whole, this paper concludes that a metrics-based, hybrid NLP+LLM model can play a significant role in the completeness and traceability of the requirements validation, and address the gaps identified in the earlier and recent literature. Reduction of validation times by weeks to days, almost human accuracy and with quantifiable cost benefits the framework shows to be a scalable and adoption-ready framework that can be attributed to both the theoretical scholarship and industrial implementation. Its contribution is not just the further development of the concepts of requirements traceability (Cleland-Huang et al., 2012; Mucha et al., 2024) but also, it has been coherent with the recent requests to implement hybrid AI-based solutions (Ali et al., 2024; Zhang et al., 2023), which, in its turn, provides a strong pathway to the operationalization of the academic innovation in real-life software engineering environments.

6. Conclusions

The research problem of the current study was to contribute to solve one of the most perennial questions in the requirements engineering field i.e., completeness and traceability of Software Requirements Specifications (SRS). Even though it is conventionally justified in this manner, it can also be subject to error and also intensive of resources, as it can fail to capture the finer semantic relationships among requirements and situations. To overcome these weaknesses, the paper suggested a semantic similarity model, large language models

(LLMs), and hybrid validation system through metrics based on scenarios to justify them. These results are backed by empirical testing of these five industrial sectors and clearly indicate that the superiority of the framework to the traditional manual validation processes depends on accuracy, efficiency, cost-effectiveness and usability.

As the findings suggest, the framework had a score of 96.7 in the validation phase, and reduced the false positives and the false negatives by large margins compared to the manual review. More noteworthy is that the framework has been noted to be equally successful with medium and low similarity cases where ambiguity and implicit dependencies have been endemic, which has proved to be a thorny area. This is consistent with findings of Zhao et al. (2021), who indicated that the traditional methods of NLP have been associated with the development of requirement classification and defects detection yet the traditional methods have limitations in regard to the extended based semantic associations and thus more versatile hybrid methods are required. The Sentence-BERT embeddings, cross-encoders and LLM validation of this research paper is a direct answer to this disjunction, and has demonstrated a high degree of semantically rich models progress in terms of requirements traceability and completeness validation.

Efficiency wise, the framework reduced the validation time (6-12 weeks to 2-3 days) by 84 percent. Such radical reduction is supportive of the earlier fears of Wagner, Fernandez, Felderer and Kalinowski (2017) who found that manual traceability practices in agile projects struggle to persevere with the iterative cycles of growth. The structure allows real time or near real time validation by automating various important processes which is critical to software delivery environments that are fast paced. Cost benefit analysis also reflected an 85 percent reduction in cost of validation and estimated ROI of 380-520 percent in 18 months period and therefore the framework was technically feasible and economically viable to be adapted by industries.

The introduction of six novel metrics—**Requirements–Scenarios Traceability Index (RSTI), Unlinked Requirements Percentage (URP), Scenario Coverage Ratio (SCR), Scenario Completeness Score (SCS), Scenario Similarity Index (SSI), and Scenario–Requirements Ratio (SRR)**—fills a significant research gap. Much of the earlier literature on requirements validation has lacked reproducible, quantitative measures of completeness and coverage. Guo, Steghöfer, Vogelsang, and Cleland-Huang (2025) argue that standardized and quantifiable traceability indicators are essential for systematic quality assurance, and the

present research contributes directly to this call. The framework's metrics not only quantified validation performance but also provided actionable insights for identifying coverage gaps and prioritizing corrective actions. For instance, URP dropped from **34.7% in manual validation to 11.2%** with the proposed system, highlighting its capacity to detect and resolve missing scenario linkages.

The broader implications of this research also lie in bridging academic innovation with industrial practice. Sjøberg et al. (2005) noted that a longstanding challenge in software engineering research is the lack of controlled experiments that translate into practical tools adopted in real-world contexts. By evaluating the framework with **18 practitioners across diverse domains** and obtaining **94% adoption intent**, this study provides compelling evidence that its contributions are not confined to theoretical significance but extend to real organizational impact. Ferrari, Spoletini, and Gnesi (2016) previously pointed out that ambiguity and tacit knowledge during requirements elicitation often weaken the foundation of SRS. The current framework complements these insights by offering practitioners a tool that systematically uncovers ambiguities through semantic similarity analysis, reducing reliance on subjective interpretation.

The findings also align with recent trends in leveraging LLMs for software engineering tasks. Zheng et al. (2023) highlighted that LLMs represent a paradigm shift in natural language-based software tasks, enabling reasoning and contextual understanding far beyond earlier embedding-based models. The results of this study confirm that combining LLM reasoning with embedding-driven retrieval provides superior outcomes, achieving near-human performance in traceability and validation. However, as Karras, Hamadeh, and Schneider (2018) caution, communication and representation formats remain critical in ensuring requirements clarity. The visualization features and interactive dashboards integrated into the framework address this concern by making validation results accessible to both technical and non-technical stakeholders, thus improving communication across development teams.

Nevertheless, this study acknowledges certain limitations. While the evaluation covered 547 requirement–scenario pairs across five domains, extreme-scale deployments exceeding **10,000 requirements** remain unexplored. Future work should focus on optimizing indexing and inference pipelines for scalability in enterprise-scale projects. Similarly, the current implementation primarily supports English-language documents, whereas global development environments increasingly require **multilingual support**. As Zhao et al. (2021)

and Guo et al. (2025) emphasize, cross-lingual NLP methods are an emerging area of research, and extending the framework with multilingual embeddings would be a valuable next step. Moreover, although the study achieved strong practitioner buy-in, integration with legacy systems was reported as a minor challenge, echoing Nayrolles and Hamou-Lhadj (2016), who observed that tool adoption in industrial environments often faces resistance due to compatibility issues and entrenched workflows.

Recommendations

Based on the outcomes of this research, several recommendations can be proposed for both academic and industrial stakeholders:

1. **Formal Adoption of Metrics-Driven Validation**: Organizations should integrate quantitative indicators such as RSTI, URP, and SCR into their requirements engineering workflows. This will enable objective measurement of completeness and traceability rather than relying solely on subjective assessments.
2. **Integration into Agile and Continuous Delivery Pipelines**: Given the demonstrated efficiency improvements, the framework should be deployed within **CI/CD pipelines** to provide real-time validation feedback during backlog refinement and sprint planning. This will address the gap identified by Wagner et al. (2017), where manual practices failed to keep pace with agile cycles.
3. **Cross-Lingual and Multimodal Extensions**: Future research should extend the framework to handle **multilingual SRS documents** and incorporate multimodal artifacts such as videos, as recommended by Karras et al. (2018), to support requirements communication in globally distributed teams.
4. **Establishment of Benchmark Datasets**: To address the comparability challenge noted by Sjøberg et al. (2005), there is a need to establish **standardized benchmark datasets and evaluation protocols** in requirements engineering. This will enable systematic comparison of validation approaches across research and industry.
5. **Human-in-the-Loop Enhancement**: While the framework already integrates LLM verification, incorporating structured human feedback loops would further improve trust and adaptability. Ferrari et al. (2016) suggest that tacit stakeholder knowledge is

crucial, and hybrid approaches combining automation with stakeholder insights will deliver optimal outcomes.

6. **Tool Ecosystem Integration**: Industrial adoption can be accelerated by ensuring interoperability with widely used requirements management tools such as IBM DOORS, JIRA, and Polarion. As Nayrolles and Hamou-Lhadj (2016) noted, adoption barriers often arise from tool incompatibility, and designing plug-ins or APIs can mitigate resistance.

In conclusion, this study demonstrates that a **metrics-driven, AI-enhanced validation framework** offers a transformative solution to one of software engineering's most persistent problems. By uniting formalized quantitative metrics with hybrid semantic validation pipelines, the framework delivers not only scientific contributions to requirements engineering but also tangible benefits for industrial practice. The strong empirical evidence, coupled with high adoption readiness among practitioners, indicates that the approach is both scalable and sustainable. Future extensions—particularly in scalability, multilingual support, and benchmark establishment—will further strengthen its role as a foundational tool for advancing the state of requirements validation in modern software engineering

REFRENCES

[1] E. Stephen and E. Mit, "Evaluation of software requirement specification based on IEEE 830 quality properties," *International Journal on Advanced Science, Engineering and Information Technology,* vol. 10, no. 4, pp. 1396–1402, 2020.

[2] A. Gonçalves, F. Martins, P. Carreira, P. Lopes, and S. Nunes, "Ieee std 830 prática recomendada para especificações de exigências de software," *Lisboa: Universidade Técnica,* 2004.

[3] R. F. Al-Msie'deen, A. H. Blasi, and M. A. Alsuwaiket, "Constructing a software requirements specification and design for electronic IT news magazine system," *arXiv preprint arXiv:2111.01501,* 2021.

[4] C. Wiecher, J. Fischbadh, J. Greenyer, A. Vogelsang, C. Wolff, and R. Dumitrescu, "Integrated and iterative requirements analysis and test specification: A case study at kostal," in *2021 ACM/IEEE 24th International Conference on Model Driven Engineering Languages and Systems (MODELS)*, 2021: IEEE, pp. 112–122.

[5] A. Davis *et al.*, "Identifying and measuring quality in a software requirements specification," in *[1993] Proceedings First International Software Metrics Symposium*, 1993: Ieee, pp. 141–152.

[6] D. M. Berry and E. Kamsties, "Ambiguity in requirements specification," in *Perspectives on software requirements*: Springer, 2004, pp. 7–44.

[7] H. Femmer, D. M. Fernández, S. Wagner, and S. Eder, "Rapid quality assurance with requirements smells," *Journal of Systems and Software,* vol. 123, pp. 190–213, 2017.

[8] L. Montgomery, D. Fucci, A. Bouraffa, L. Scholz, and W. Maalej, "Empirical research on requirements quality: a systematic mapping study," *Requirements Engineering,* vol. 27, no. 2, pp. 183–209, 2022.

[9] W. Behutiye *et al.*, "Management of quality requirements in agile and rapid software development: A systematic mapping study," *Information and software technology,* vol. 123, p. 106225, 2020.

[10] O. C. Gotel and C. Finkelstein, "An analysis of the requirements traceability problem," in *Proceedings of ieee international conference on requirements engineering*, 1994: IEEE, pp. 94–101.

[11] J. Cleland-Huang, O. Gotel, and A. Zisman, *Software and systems traceability* (no. 3). Springer, 2012.

[12] E. Sarmiento-Calisaya and J. C. S. do Prado Leite, "Early analysis of requirements using NLP and Petri-nets," *Journal of Systems and Software*, vol. 208, p. 111901, 2024.

[13] O. S. D. Omer and M. M. Mahmoud, "Requirements and design consistency: A bi-directional traceability and natural language processing assisted approach," *European Journal of Engineering and Technology Research*, vol. 6, no. 3, pp. 120–129, 2021.

[14] J. W. Lim *et al.*, "Test case information extraction from requirements specifications using NLP-based unified boilerplate approach," *Journal of Systems and Software*, vol. 211, p. 112005, 2024.

[15] M. G. Gramajo, L. Ballejos, and M. Ale, "Recurrent neural networks to automate quality assessment of software requirements," *arXiv preprint arXiv:2105.04757*, 2021.

[16] M. I. Azeem and S. Abualhaija, "A multi-solution study on GDPR AI-enabled completeness checking of DPAs," *Empirical Software Engineering*, vol. 29, no. 4, p. 96, 2024.

[17] J. Ni *et al.*, "Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models," *arXiv preprint arXiv:2108.08877*, 2021.

[18] Q. Zhang *et al.*, "A survey on large language models for software engineering," *arXiv preprint arXiv:2312.15223*, 2023.

[19] J. Mucha, A. Kaufmann, and D. Riehle, "A systematic literature review of pre-requirements specification traceability," *Requirements Engineering*, vol. 29, no. 2, pp. 119–141, 2024.

[20] S. J. Ali, V. Naganathan, and D. Bork, "Establishing traceability between natural language requirements and software artifacts by combining rag and llms," in *International Conference on Conceptual Modeling*, 2024: Springer, pp. 295–314.

[21] L. Zhao *et al.*, "Natural language processing for requirements engineering: A systematic mapping study," *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, pp. 1–41, 2021.

[22] J. L. Guo, J.-P. Steghöfer, A. Vogelsang, and J. Cleland-Huang, "Natural language processing for requirements traceability," in *Handbook on Natural Language Processing for Requirements Engineering*: Springer, 2025, pp. 89–116.

[23] Z. Zheng *et al.*, "A survey of large language models for code: Evolution, benchmarking, and future trends," *arXiv preprint arXiv:2311.10372*, 2023.

[24] D. I. Sjøberg *et al.*, "A survey of controlled experiments in software engineering," *IEEE transactions on software engineering*, vol. 31, no. 9, pp. 733–753, 2005.

[25] S. Wagner, D. M. Fernández, M. Felderer, and M. Kalinowski, "Requirements engineering practice and problems in agile projects: results from an international survey," *arXiv preprint arXiv:1703.08360*, 2017.

[26] O. Karras, A. Hamadeh, and K. Schneider, "Enriching requirements specifications with videos-the use of videos to support requirements communication," in *Software Technik-Trends Band 38, Heft 1*, 2018: Geselllschaft für Informatik eV, pp. 51–52.