Empirical Studies on M-X Algorithm Implementation in Tester Eligibility Classification for Crowdsourced Based User Acceptance Testing

Glorious Satria Dhamang Aji
Software Engineering, School of Computing
Telkom University
Bandung, Indonesia
glorioussatria@student.telkomuniversity.ac.id

Dana Sulistiyo Kusumo
Software Engineering, School of Computing
Telkom University
Bandung, Indonesia
danakusumo@telkomuniversity.ac.id

***Abstract*—Crowdsourced User Acceptance Testing (UAT) presents unique challenges due to tester variability, which can impact the consistency and reliability of test outcomes. This study evaluates the M-X algorithm as a probabilistic quality control mechanism for identifying qualified testers without relying on predefined correct answers. A simulation was conducted through a client-server system built with GraphQL and MongoDB, involving 24 participants. Evaluation using Confusion Matrix metrics—mapped directly to research questions—shows that the M-X algorithm achieves 79% accuracy, 82% precision, and 75% recall. These results demonstrate its effectiveness in detecting consistent testers and reinforce its applicability to subjective and explorative UAT scenarios. The findings establish a foundation for response-based tester filtering in crowdsourced environments and point to future opportunities for enhancing algorithm robustness in dynamic user acceptance testing contexts.**

***Index Terms*—crowdsourcing, user acceptance testing, quality control, M-X algorithm, online assessment, client-server**

I. Introduction

Crowdsourced User Acceptance Testing (UAT) offers cost-efficient software evaluation by engaging a diverse pool of testers [1]. This approach improves testing coverage across varied usage scenarios, but also introduces challenges due to inconsistencies in tester quality [2]. In UAT practices, tasks are often subjective and context-dependent, making it difficult to assess tester reliability using conventional correctness-based metrics [7]. The lack of predefined answers in exploratory or usability-focused UAT scenarios further complicates quality control [4].

The M-X algorithm [8] addresses this challenge by inferring worker quality through inter-tester agreement, rather than comparing responses to known verified outcomes. It offers a generalizable solution by evaluating answer consistency across multiple-choice tasks, enabling probabilistic classification of tester eligibility without requiring predefined correct answers. While the algorithm has demonstrated theoretical effectiveness and computational scalability via MapReduce, prior evaluations have been limited to synthetic crowdsourcing contexts with uniform task distributions. [8]

A. Research Gap and Motivation

Previous studies on crowdsourced worker quality control primarily focus on task performance in objective contexts or rely on extensive historical and demographic data. For instance, Wang et al. [12] employed multi-objective optimization based on prior bug detection history, while Hong et al. [13] proposed a Worker Search Model utilizing wide learning, which raises privacy concerns due to its dependence on personal attributes. Yao et al. [14] examined learning curves in testers but did not address initial quality or consistency filtering. These methods either assume the availability of verified outcomes, require significant manual supervision, or overlook the issue of tester variability in subjective testing tasks.

To date, no study has explicitly validated the M-X algorithm in realistic UAT scenarios that:

1) Lack predefined answers and involve subjective judgments,
2) Exhibit heterogeneous tester backgrounds with varying domain knowledge and prior experience,
3) Require automated quality classification mechanisms with minimal human intervention.

Moreover, while Dang et al. introduced M-X as a theoretical solution to such challenges [8], they did not empirically evaluate its robustness in crowdsourced UAT settings, particularly in terms of handling inter-tester variability and assessing how prior experience influences classification accuracy.

B. Research Objectives

This study aims to fill these gaps by empirically validating the M-X algorithm in a simulated UAT environment. Key contributions include:

1) Evaluating M-X's classification accuracy using real tester responses in subjective tasks,
2) Measuring the impact of tester variability, particularly prior experience, on algorithm performance,
3) Using Confusion Matrix metrics—Accuracy, Precision, Recall, F1-Score, and TNR—to assess classification quality across varying crowd sizes and task batches.

C. Research Questions

The study is guided by the following research questions:

1) **RQ1:** How accurately does the M-X algorithm classify testers in crowdsourced UAT environments?
2) **RQ2:** To what extent can the M-X algorithm reduce tester variability and ensure consistent testing quality, particularly in relation to prior experience?

Through these objectives and questions, this research aims to provide a comprehensive understanding of M-X's effectiveness as a quality control mechanism in subjective, real-world crowdsourced UAT environments.

II. Theoretical Background

A. User Acceptance Testing and Crowdsourcing Integration

User Acceptance Testing (UAT) represents the final validation phase where actual users verify system compliance with business requirements and readiness for deployment [3], [5]. UAT serves as a critical bridge between development teams and end-users, ensuring delivered systems meet user expectations through evaluation of completeness, usability, and business alignment [6].

Crowdsourced UAT leverages distributed user populations to provide cost-effective, scalable testing with diverse perspectives [1]. This approach combines crowdsourcing accessibility with UAT's user-centric validation objectives, engaging external participants who represent authentic end-users [18], [19].

However, crowdsourced UAT introduces significant challenges including participant heterogeneity, varying domain expertise, and absence of controlled environments [2]. Unlike conventional testing scenarios, UAT often lacks predefined correct answers and involves highly contextual, subjective evaluations [7]. These characteristics necessitate quality control mechanisms that can assess tester reliability without ground truth dependencies [4], making the M-X algorithm's peer-consistency approach particularly suitable for such environments.

B. M-X Algorithm

The M-X algorithm, introduced by Dang et al. [8], is a probabilistic approach for evaluating worker quality in crowdsourcing environments without relying on predefined correct answers. The algorithm works by analyzing the consistency of responses across multiple workers and inferring worker quality based on inter-worker agreement patterns.

1) Mathematical Foundation of M-X Algorithm: The M-X algorithm is designed to evaluate worker quality in scenarios where correct answers are unknown. It begins with a mathematical model for multiple-choice problems, where the key principle is analyzing agreement between workers. For a single choice problem with M options, where M represents the total number of available options, the algorithm defines:

- A_i represents the accuracy rate (or quality) of worker w_i, meaning the probability they provide the correct answer
- Worker set is defined as $\{w_i | 1 \leq i \leq K\}$, where K is the total number of workers
- Problem set is represented as $\{p_u | 1 \leq u \leq N\}$, where N is the total number of problems

For any two workers w_i and w_j answering the same problems, the algorithm tracks their agreement through random variable X_{ij}^u, which equals 1 if they agree on problem p_u and 0 otherwise. The total number of times workers w_i and w_j agree is denoted as:

$$T_{ij} = \sum_{1 \leq u \leq N} X_{ij}^u \tag{1}$$

The key insight of the algorithm is that the probability of agreement between two workers (Q_{ij}) can be expressed as:

$$Q_{ij} = A_i \cdot A_j + \frac{(1 - A_i)(1 - A_j)}{M - 1} \tag{2}$$

This equation captures two ways workers can agree:

- $A_i \cdot A_j$: Both workers select the correct option
- $\frac{(1-A_i)(1-A_j)}{M-1}$: Both workers select the same incorrect option

The second term can be explained as follows: $(1 - A_i)$ is the probability that worker w_i chooses an incorrect option, $(1 - A_j)$ is the probability that worker w_j chooses an incorrect option, and $\frac{1}{M-1}$ is the probability that they both choose the same incorrect option from the $(M - 1)$ available incorrect options.

For three workers, by calculating the agreement probabilities Q_{12}, Q_{13}, and Q_{23}, the algorithm can solve for each worker's accuracy. For example, worker w_1's accuracy is derived as:

$$A_1 = \frac{1}{M} + \frac{M-1}{M} \cdot \sqrt{\frac{(M \cdot Q_{12} - 1) \cdot (M \cdot Q_{13} - 1)}{M \cdot Q_{23} - 1}} \tag{3}$$

For scenarios with more than three workers, the algorithm employs a sliding window approach where:

- Workers are arranged in a circular pattern
- Groups of three workers are evaluated using the base algorithm
- Each worker's final accuracy is the average of multiple evaluations

For multiple-choice problems, the M-X algorithm first divides each problem with M options into M separate single-choice (yes/no) problems for each option. It then calculates worker accuracy on each option dimension and combines these scores to determine the comprehensive worker quality.

C. Algorithm Selection Rationale

Existing worker quality control approaches face significant limitations when applied to UAT environments. Traditional methods such as MOCOM [12] require extensive historical performance data, while demographic-based approaches like WSM [13] raise privacy concerns and potential bias issues. Learning curve methodologies [14] focus on long-term skill development rather than immediate quality assessment, making them unsuitable for real-time worker selection.

The M-X algorithm was selected for this study due to its unique compatibility with UAT characteristics. Unlike conventional approaches, M-X operates through inter-worker agreement analysis without requiring predefined correct answers or historical data, making it immediately deployable with new worker populations. This peer-consistency foundation enables effective evaluation in subjective UAT scenarios where ground truth is often unavailable or contextual.

Additionally, M-X preserves worker privacy by evaluating response patterns rather than collecting personal attributes, while maintaining computational efficiency compared to multi-objective optimization approaches. These characteristics make M-X particularly suitable for crowdsourced UAT environments where immediate deployment, privacy preservation, and subjective evaluation capabilities are essential requirements.

D. Prior-Experience Based Validation

Prior-Experience Based Validation establishes tester eligibility through historical performance and domain expertise assessment, incorporating factors such as previous testing outcomes, domain knowledge, and demonstrated competency [12]. Research indicates that testers with documented experience show superior effectiveness in identifying relevant issues and providing consistent feedback [17].

In this study, prior experience serves as the ground truth for evaluating M-X algorithm classifications. Participants were categorized as eligible or non-eligible based on their UAT experience, creating a practical benchmark against which algorithmic determinations could be assessed. This dual-validation approach enables objective performance measurement while highlighting areas where consistency-based evaluation may diverge from experience-based assessments, thereby strengthening the practical applicability of our findings across varying crowdsourced UAT contexts.

E. Confusion Matrix

Confusion Matrix is a methodology used to assess the performance of classification models, particularly in scenarios where direct true labels may be unavailable or imprecise. When applied to tester eligibility classification, Confusion Matrix provides a framework for evaluating worker quality based on observed behaviors and response patterns, rather than direct comparison with known correct answers [15].

1) Confusion Matrix Metrics: To evaluate the performance of the M-X algorithm in classifying testers, several standard metrics are employed based on the Confusion Matrix:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{6}$$

$$\text{F1 Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

$$\text{TNR (Specificity)} = \frac{TN}{TN + FP} \tag{8}$$

Where:

- TP (True Positive): Number of eligible testers correctly classified as eligible
- TN (True Negative): Number of non-eligible testers correctly classified as non-eligible
- FP (False Positive): Number of non-eligible testers incorrectly classified as eligible
- FN (False Negative): Number of eligible testers incorrectly classified as non-eligible

These metrics provide a comprehensive evaluation of the algorithm's classification performance, balancing between correctly identifying eligible testers and minimizing the inclusion of non-eligible testers. High precision indicates that the algorithm minimizes false positives, which is crucial in UAT scenarios where including unqualified testers could compromise testing quality. High recall demonstrates the algorithm's ability to identify all potentially qualified testers, ensuring comprehensive coverage of the available talent pool.

III. Research Methodology

A. Research Design

This study adopts a quantitative approach through a simulation-based implementation of the M-X algorithm, utilizing participant composition, task structure, simulation execution conditions, and evaluation metrics to systematically assess its performance in classifying tester quality within a crowdsourced User Acceptance Testing (UAT) framework. The simulation is facilitated by a custom-built web-based system developed to support the operationalization of the algorithm.

Simulation data is numerical and analyzed using metrics within Confusion Matrix. Each research question is directly mapped to corresponding evaluation metrics to ensure a systematic performance assessment of the algorithm.

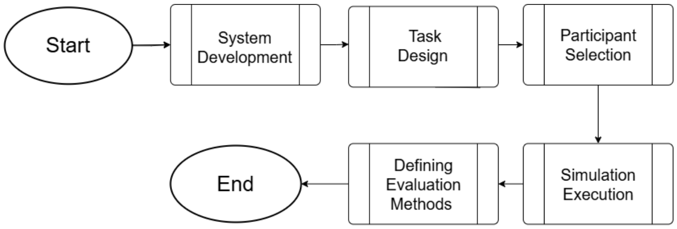

Fig. 1. Research Methodology Flowchart

The research methodology follows a structured approach as illustrated in Fig. 1. The process begins with simulation planning, followed by execution, algorithm validation and evaluation, and concludes with analysis and reporting.

B. System Development

To facilitate the simulation of tester quality classification in a crowdsourced UAT setting, a scalable client-server system was developed [9]. The system integrates the M-X algorithm within a quality control service to classify testers based on the consistency of their responses. GraphQL was used as the API gateway for flexible client-server communication [10], while MongoDB handled complex data storage operations [11].

The system consists of four main modules: Worker Management for tester registration and profiling, Auth Service for authentication, Task Management for test assignment and validation, and Quality Control Service for computing accuracy scores using the M-X algorithm.

Fig. 2 illustrates the system architecture, including the flow of task assignment, worker responses, and accuracy computation based on the M-1 probabilistic foundation.

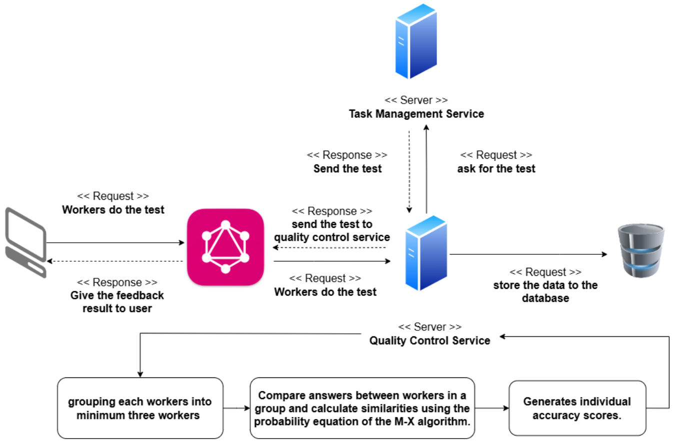

Fig. 2. System Architecture for Crowdsourced UAT Simulation with M-X Algorithm Integration

C. Task Design for M-X Algorithm Compatibility

Designing UAT test scenarios required alignment with both user validation principles and the M-X algorithm's mathematical constraints. Gherkin syntax was chosen for its clarity, structured format, and relevance to behavior-driven development practices [16].

To ensure compatibility with the M-X algorithm [8], each task adhered to the following criteria:

- Tasks used a multiple-choice format with M independent options.
- Each option was decomposable into a binary (yes/no) sub-question.
- Options were logically independent, avoiding overlap or exclusivity.
- No predefined correct answers was provided; quality was inferred via inter-tester agreement.
- The algorithm assumed uniform distribution when incorrect answers were chosen.

The task design process included:

1) **User Journey Mapping**: Defining real user flows to determine relevant testing scenarios.
2) **Scenario Structuring**: Representing each flow as a scenario.
3) **Gherkin Formatting**: Using the "Given-When-Then" model:
    - **Given**: Initial state or context
    - **When**: Triggering action
    - **Then**: Expected outcome
4) **M-X Integration**: Transforming "Then" into a set of multiple, binary-evaluable, independent options.

During the simulation, testers were presented with full Gherkin scenarios and asked to choose from multiple ”Then” options. Responses were stored as binary selections, and at least three testers per scenario ensured the algorithm's peer-consistency validation could operate effectively.

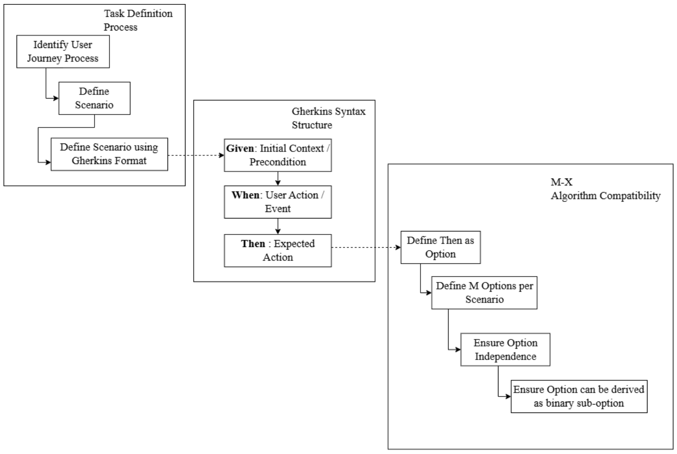
Task Definition Process
Identify User Journey Process
Define Scenario
Define Scenario using Gherkins Format
Gherkins Syntax Structure
Given: Initial Context / Precondition
When: User Action / Event
Then : Expected Action
M-X Algorithm Compatibility
Define Then as Option
Define M Options per Scenario
Ensure Option Independence
Ensure Option can be derived as binary sub-option

Fig. 3. Task Scenario Construction and M-X Compatibility Process

D. Participant Selection and Characteristics

The simulation involved 24 participants, primarily students from Telkom University aged 20-24 years. Of these participants, 12 had prior experience in User Acceptance Testing, while the other 12 had no such experience. The majority (17 participants) came from IT-related study programs like Software Engineering, Informatics, and Information Systems, with 7 specifically from Software Engineering backgrounds.

This balanced ratio of experienced to inexperienced participants (12:12) was deliberately designed to facilitate a balanced evaluation of the M-X algorithm's ability to classify testers. Prior-Experience Based Validation was used to establish the actual eligibility status data for participant eligibility, creating a reference point against which the algorithm's classifications could be evaluated.

E. Simulation Execution

The simulation was conducted over three days in a hybrid format to facilitate accessibility. Participants were presented with a series of UAT scenarios using the Gherkin syntax format and were asked to complete tasks through a web-based platform developed specifically for this research. The platform implemented the M-X algorithm to analyze participant responses and classify them based on response consistency.

F. Defining Evaluation Methods

The evaluation was conducted using Confusion Matrix with a direct connection between research questions and performance metrics.

1) Research Question to Metrics Mapping Framework: To ensure a systematic evaluation approach, we established a direct mapping between research questions and specific measurement metrics:

TABLE I
RESEARCH QUESTIONS TO METRICS MAPPING

Research Question	Evaluation Metrics
RQ1: How accurately does the M-X algorithm classify testers in crowdsourced UAT environments?	Accuracy, Precision, Recall, F1 Score
RQ2: To what extent can the M-X algorithm reduce tester characteristic variability?	TNR, FP Rate

The mapping framework established clear success criteria for each research question while maintaining methodological rigor. By directly connecting each research question to specific metrics, we ensured that all aspects of the algorithm's performance were systematically evaluated.

IV. RESULTS AND DISCUSSION

A. Comprehensive Evaluation Results

This study involved a comprehensive evaluation of the M-X algorithm's performance in classifying testers based on response consistency. All 24 participants were evaluated simultaneously within the same experimental context, providing a robust dataset for analysis. Based on the consistency of their responses as analyzed by the M-X algorithm, the classification results were obtained by comparing the algorithm's eligibility determination against the prior experience-based validation data.

From the evaluation, the following confusion metrics were calculated:

TABLE II
CLASSIFICATION RESULTS SUMMARY

Classification Category	Count
True Positives (TP)	9
True Negatives (TN)	10
False Positives (FP)	2
False Negatives (FN)	3
Total Participants	24

These results yielded the performance metrics shown in Table III:

TABLE III
PERFORMANCE METRICS OF M-X ALGORITHM

Metric	Value
Accuracy	79%
Precision	82%
Recall	75%
F1 Score	78%
TNR (Specificity)	83%

B. Detailed Analysis

The high precision value (82%) demonstrates that the M-X algorithm is particularly effective at minimizing false positives, which is crucial in crowdsourced UAT contexts where ensuring that only qualified testers are included is paramount. This suggests that the algorithm's probabilistic approach to evaluating inter-tester agreement successfully identifies consistent response patterns indicative of eligible testers.

The slightly lower recall value (75%) indicates a moderate tendency of the algorithm to produce false negatives, potentially excluding some eligible testers. This conservative tendency may be attributed to the algorithm's reliance on consistency patterns that might not fully capture all dimensions of tester expertise or capability.

Analysis of misclassifications shows specific patterns in the data. Among the false negatives were experienced 3 software engineering students who were considered eligible based on prior experience but were not identified by the algorithm. The false positives included 2 participants without formal UAT experience who were classified as eligible by the algorithm. These misclassifications suggest that the algorithm's consistency-based approach may sometimes diverge from experience-based eligibility determinations, highlighting the potential complementary nature of these assessment methods. Overall, the M-X algorithm demonstrated robust performance across all key metrics. The balanced F1 score (78%) further confirms the algorithm's effectiveness as a mechanism for filtering testers based on response consistency in crowdsourced UAT environments.

C. Response to Research Questions

Based on the direct research question to metrics mapping and the comprehensive evaluation results, we can address the key research questions:

RQ1: How accurately does the M-X algorithm classify testers in crowdsourced UAT environments? The M-X algorithm demonstrated an overall accuracy of 79%, with a precision of 82% and recall of 75%. These metrics indicate that the algorithm is generally effective in classifying testers, with particular strength in correctly identifying eligible testers (high precision) while maintaining reasonable effectiveness in capturing all eligible testers (moderate recall). The F1 score of 78% further validates the algorithm's balanced performance in classification tasks. These performance metrics suggest the algorithm is a viable tool for quality control in crowdsourced UAT environments where identifying consistent testers is crucial.

RQ2: To what extent can the M-X algorithm reduce tester characteristic variability and ensure consistent testing quality? The algorithm successfully identified and filtered out 10 of 12 non-eligible testers, achieving a True Negative Rate (TNR) of 83%. This indicates the algorithm's effectiveness in reducing tester characteristic variability by establishing a probabilistic threshold for consistency that maintains testing quality. The high precision (82%) further confirms that the algorithm is effective at ensuring most admitted testers are indeed capable of providing consistent, high-quality testing responses. The algorithm's ability to identify testers with inconsistent response patterns, regardless of their formal qualifications, demonstrates its potential to standardize the quality of testing outcomes by creating a more homogeneous tester pool.

D. Limitations

Several limitations must be acknowledged. The sample size of 24 participants limits generalizability to larger crowdsourced environments. The participant pool consisted primarily of university students aged 20-24, which may not represent the diversity of real-world crowdsourcing platforms. The simulation environment lacks the complexity of actual production deployments, and the multiple-choice task format required for M-X compatibility may not capture the full spectrum of UAT activities. Future research should address these limitations through larger-scale studies and real-world deployment validation.

V. Conclusion

This study evaluated the M-X algorithm for tester classification in crowdsourced User Acceptance Testing, achieving 79% accuracy, 82% precision, and 75% recall across 24 participants. While high precision demonstrates effective quality control with minimal false positives, the moderate recall reveals a critical limitation: the algorithm excluded 3 experienced software engineering students deemed eligible by prior experience. This conservative tendency suggests consistency-based assessment may discard valuable contributors whose response patterns deviate from algorithmic consensus, potentially reducing tester pool diversity and quality in complex UAT scenarios requiring varied perspectives.

The algorithm's effectiveness likely varies across domains and contexts. While suitable for filtering casual participants in generic applications, enterprise software or specialized systems may require domain knowledge that transcends response consistency. The reliance on inter-tester agreement assumes consensus indicates quality, but in innovative domains, expert minority opinions might be more valuable than majority agreement. Future implementations should consider domain-specific calibration and hybrid approaches combining consistency metrics with expertise indicators.

Compared to Dang et al.'s theoretical framework [8], this study provides empirical validation in realistic UAT settings while revealing practical limitations. The findings support the algorithm's use where predefined answers are unavailable but highlight the need for complementary assessment mechanisms. Future research should explore adaptive thresholds and multi-criteria frameworks balancing consistency with domain expertise. The algorithm serves as a valuable foundation for crowdsourced UAT quality control, but requires careful domain-specific calibration and supplementary qualification mechanisms.

References

[1] Leicht, N., Knop, N., Blohm, I., Müller-Bloch, C., & Leimeister, J. M. (2016). When is crowdsourcing advantageous? The case of crowdsourced software testing. In *Proceedings of the European Conference on Information Systems (ECIS 2016)*. Istanbul, Turkey.

[2] Nasir, M., Ikram, N., & Jalil, Z. (2022). Usability inspection: Novice crowd inspectors versus expert. *Journal of Systems and Software*, 183, 111122.

[3] H. K. N. & W. P. W. L. Leung, "A study of user acceptance tests," *Software Quality Journal,* pp. 137-149, 1997.

[4] Hossfeld, T., Seufert, M., Zinner, T., & Tran-Gia, P. (2021). On inter-rater reliability for crowdsourced QoE. *Quality and User Experience*, 6(1), 1–16.

[5] I. Sommerville, *Software Engineering 9th*, Boston: Addison-Wesley, 2011.

[6] L. Bormane, D. Gržibovska, S. Bērziša, and E. Grabis, "Impact of Requirements Elicitation Processes on Success of Information System Development Projects," *Information Technology and Management Science*, vol. 19, no. 1, pp. 57-64, 2016.

[7] S. Loss, R. F. Ciriello, and J. Cito, "Revealing the Vicious Circle of Disengaged User Acceptance: A SaaS Provider's Perspective," in *Proceedings of the 40th International Conference on Information Systems (ICIS)*, Munich, Germany, 2019.

[8] D. Dang, Y. Liu, X. Zhang and S. Huang, "A Crowdsourcing Worker Quality Evaluation Algorithm on MapReduce for Big Data Applications," in *IEEE Transactions on Parallel and Distributed Systems*, 2015.

[9] G. Blinowski, A. Ojdowska and A. Przybyłek, "Monolithic vs. Microservice Architecture: A Performance and Scalability Evaluation," in *IEEE Access*, 2022.

[10] P. Margański and B. Pańczyk, "REST and GraphQL comparative analysis," *Journal of Computer Sciences Institute,* vol. 19, pp. 89-94, 2021.

[11] N. Chaudhary and N. Mittal, "Leveraging Mongo DB for Efficient Data storage in MERN," in *International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)*, Noida, India, 2024.

[12] S. W. J. C. T. M. Q. C. M. X. Q. W. Junjie Wang, "Characterizing Crowds to Better Optimize Worker Recommendation in Crowdsourced Testing," *IEEE Transactions on Software Engineering,* 2019.

[13] Y.-Y. L. J.-H. K. U.-M. K. Jeon-Pyo Hong, "Crowd Worker Selection with Wide Learning and Narrow Evaluation," in *15th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, Seoul, Korea, 2021.

[14] S. H. C. Z. E. L. a. N. C. Yongming Yao, "A Study on TestersĹearning Curve in Crowdsourced Software Testing," *IEEE Access,* vol. 9, pp. 77127-77137, 2021.

[15] C. Spampinato, S. Palazzo, and D. Giordano, "Evaluation of tracking algorithm performance without ground-truth data," *Proc. - Int. Conf. Image Process. ICIP*, pp. 1345–1348, 2012, doi: 10.1109/ICIP.2012.6467117.

[16] T. R. Silva, "Towards a Domain-Specific Language for Behaviour-Driven Development," in *2023 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, Odense, Denmark, 2023.

[17] Q. Cui, S. Wang, J. Wang, Y. Hu, Q. Wang, and M. Li, "Multi-Objective Crowd Worker Selection in Crowdsourced Testing," in *Proceedings of the 29th International Conference on Software Engineering and Knowledge Engineering (SEKE)*, July 2017, doi: 10.18293/SEKE2017-102.

[18] M. & A. S. Alsayyari, "Supporting Coordination in Crowdsourced Software Testing Services," in *2018 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, Bamberg, Germany, 2018.

[19] S. Alyahya, "Crowdsourced Software Testing: A Systematic Literature Review," *Information and Software Technology*, 2020.