NJMSR

*Nepalese Journal of Management Science and Research*
*2026, Vol. 9, No. 1, pp. 77-97*

***Original Article***

# AI Driven Quality Assurance: A Study using Factor Analysis from the Nepalese IT Industry

**Rusha Thapa**

## Abstract

*This study investigates the influence of Artificial Intelligence (AI) on software Quality Assurance (QA) processes within the Nepalese IT sector. A quantitative research design was employed, and data were collected using a structured questionnaire comprising multiple-choice and five-point Likert-scale questions. The sample included 230 QA professionals selected through a purposive sampling method from various IT companies in Nepal. The study was guided by the AI-Driven Quality Assurance Effectiveness Model (AI-QAEM), which integrates the Technology Acceptance Model (TAM) and Quality Management Theory (QMT) to explain how AI adoption impacts QA outcomes. Data analysis using correlation and regression revealed a strong model significance ($R^2 = 0.774$, $F = 66.752$, $p < 0.001$) and high reliability (Cronbach's Alpha = 0.906). The KMO value of 0.880 confirms sampling adequacy, while reliability tests show strong internal consistency. Findings indicate that Testing Speed ($\beta = 0.480$) and Testing Techniques ($\beta = 0.234$) are the most influential factors in improving software quality. Overall, effective AI implementation enhances testing accuracy, efficiency, and product reliability across Nepal's IT sector.*

**Keywords:** Artificial Intelligence, IT Sector, Quality Assurance, Software Quality, Testing Techniques

---

**Rusha Thapa is a Senior Quality Assurance Engineer at Proshore and an IMBA graduate of Herald College Kathmandu, with research interests in software quality, and Artificial Intelligence.*

*Author Email: rushathapa888@gmail.com

ISSN (Print) :2467-9356, ISSN (Electronic) : 2795 - 1545




https://doi.org/10.53056/njmsr-2026.9.1.004

*https://www.nepjol.info/index.php/njmsr*

## 1. Background of the Study

In the rapidly evolving landscape of software development, Quality Assurance (QA) plays a critical role in ensuring product reliability, functionality, and user satisfaction. Traditional QA processes, which often depend on manual and rule-based automation testing, are increasingly challenged by the need for speed, scalability, and accuracy in agile and continuous delivery environments (Thant & Tin, 2023). Artificial Intelligence (AI) has emerged as a transformative force in this context, offering the potential to automate defect detection, optimize test coverage, and predict failures through data-driven insights (Omri & Sinz, 2021). AI-driven QA leverages techniques such as machine learning and natural language processing to enhance efficiency and reduce human errors, allowing teams to focus on complex problem-solving and innovation (Thomas, 2025). Despite these advancements, the adoption of AI in QA remains uneven globally, influenced by factors such as organizational readiness, cost, and skill availability.

From an international perspective, Artificial Intelligence has become increasingly embedded in quality assurance practices as software systems grow more complex, distributed, and data-driven. Organizations across North America, Europe, and parts of Asia are integrating AI to support activities such as anomaly detection, intelligent defect prediction, automated test case generation, and risk-based prioritization of tests. These AI-enabled capabilities are particularly valuable in agile and DevOps environments, where frequent releases require faster and more reliable validation processes (Garousi, Felderer, & Mäntylä, 2016). As a result, AI is gradually shifting QA from a primarily reactive activity toward a more predictive and proactive function.

Global studies indicate that AI-driven QA contributes to improved software quality by reducing human bias, enhancing traceability, and enabling deeper analysis of large testing datasets. Machine learning models trained on historical testing records, production logs, and bug repositories can identify hidden patterns and suggest areas most likely to fail, thereby optimizing testing resources and reducing overall costs (Davenport & Kalakota, 2019). Moreover, natural language processing tools are increasingly used to analyze requirements, detect inconsis-

tencies, and automatically derive test cases, helping to bridge communication gaps between developers, testers, and stakeholders.

However, international experience also shows that AI adoption in QA is not uniform. Technology-intensive sectors such as fintech, healthcare, telecommunications, and e-commerce have progressed faster because they operate with large datasets, strong infrastructure, and higher quality demands. Meanwhile, many organizations, especially small and medium-sized enterprises, remain cautious due to concerns related to tool maturity, integration with legacy systems, ethical risks, and lack of specialized expertise (Jobin, Ienca, & Vayena, 2019). These disparities suggest that AI-driven QA is not simply a technological upgrade but a strategic transformation requiring investment in people, processes, and governance.
Furthermore, researchers emphasize that AI cannot entirely replace human testers but rather complements their expertise. While AI excels at pattern recognition, repetitive validation, and predictive analytics, human judgment remains essential for exploratory testing, usability evaluation, and interpretation of complex or ambiguous results (Garousi, Felderer, & Mäntylä, 2016). International debates, therefore, focus on designing hybrid QA approaches where AI supports decision-making while testers retain oversight and accountability.

Overall, the global landscape demonstrates that AI-driven Quality Assurance is evolving from experimental adoption toward more structured, mainstream use. Yet, success largely depends on organizational readiness, continuous learning, ethical safeguards, and clear alignment between AI tools and QA objectives. These global experiences provide important lessons for developing contexts, offering insights into both opportunities and challenges when transitioning from traditional to AI-enabled QA environments.

In the context of Nepal, the IT industry has experienced significant growth over the past decade, with companies increasingly engaging in software exports and digital transformation projects. However, QA practices in Nepalese IT firms largely remain conventional, with limited integration of AI-driven tools and automation frameworks (Broadway Infosys, 2025). This gap presents both a challenge and an


https://doi.org/10.53056/njmsr-2026.9.1.004 https://www.nepjol.info/index.php/njmsr

opportunity. Nepalese IT companies could benefit from AI-enhanced QA to improve software quality and competitiveness in the global market. Yet, challenges such as a lack of technical expertise, insufficient investment in innovation, and limited awareness of AI applications hinder widespread adoption (Khanal, 2024). Therefore, this study seeks to explore how AI-driven QA is being adopted within the Nepalese IT industry, its impact on software quality, and the barriers faced by organizations and professionals in leveraging AI for quality improvement.

The main objective of this study is to explore the adoption, impact, and challenges of AI-driven Quality Assurance (QA) in the Nepalese IT industry. It aims to understand how AI enhances software quality and transforms traditional QA practices. Accordingly, the study addresses the following key research questions: (1) How is AI being adopted and utilized in Quality Assurance within the Nepalese IT industry, and what impact does it have on software quality?

## 2. Literature Review

The integration of Artificial Intelligence (AI) into quality assurance (QA) processes has brought transformative changes in how software is tested, defects are identified, and quality is ensured. This review explores the methodological approaches to studying AI-driven QA, focusing on key variables such as the implementation of AI tools, testing techniques, data quality, automated test scripts, and testing speed. The aim is to assess how these factors impact software quality as the ultimate dependent variable.

Research on AI implementation in QA has predominantly focused on case studies and experimental designs to evaluate the effectiveness of AI tools. For instance, early studies employed machine learning algorithms to automate defect detection, using historical defect data as training sets. Tools such as Eggplant AI, Applitools, Kusho, Groq, ChatGPT, and Gemini have been analyzed for their ability to simulate user interactions and optimize testing workflows. Experimental methodologies often compare traditional QA processes with AI-enhanced ones, highlighting measurable improvements in defect detection rates and testing efficiency (Ramadan, Yasin, & Pektas, 2024). These studies underscore that successful

implementation depends on factors such as organizational readiness, technical infrastructure, and workforce adaptability.

Methodological research on testing techniques often combines comparative analysis and simulation models. Traditional approaches, such as black-box and white-box testing, are juxtaposed with AI-driven methods like predictive analytics and model-based testing. For instance, a study by Khan et al. (2024) employed simulation tools to evaluate AI's ability to identify edge cases and optimize test coverage. Findings indicate that AI-driven techniques are more effective in detecting defects that manual methods or traditional automation might overlook. Researchers often use controlled experiments to measure the precision and recall of defect detection under AI-enhanced and traditional methodologies.

Data quality is a crucial variable in AI-driven QA, and its role is frequently studied using empirical and experimental designs. High-quality datasets ensure the accuracy of AI predictions and the reliability of testing outcomes. Researchers like Omri & Sinz (2021) conducted longitudinal studies to observe how variations in data accuracy and completeness affect AI performance in defect detection. Methodologies often involve data augmentation techniques and error analysis to understand the impact of poor-quality data on software quality. Results consistently demonstrate that ensuring data relevance and accuracy is fundamental to achieving reliable AI-driven QA processes.

The design and optimization of automated test scripts are commonly studied using design science research methodologies. These studies aim to create self-adapting scripts that evolve with software changes, improving efficiency over time. For example, Ramchand et al. (2021) developed a framework for AI-enabled automated testing and validated its performance through iterative testing cycles. Such research often employs iterative development and validation approaches, focusing on how automated scripts reduce redundancy, enhance coverage, and streamline defect detection. Performance metrics such as defect detection rate, execution time, and error rate are used to evaluate the effectiveness of these scripts.

Improvements in testing speed are typically examined through time-motion studies and comparative analysis. Researchers analyze how AI-driven automation re-


https://doi.org/10.53056/njmsr-2026.9.1.004 https://www.nepjol.info/index.php/njmsr

duces the time required for testing cycles while maintaining or enhancing accuracy. For instance, studies by Islam et al. (2023) utilized real-world project data to compare the time efficiency of AI-enhanced QA versus traditional testing. Methodologies often include statistical analysis to determine the correlation between testing speed and defect detection accuracy. Findings suggest that AI significantly reduces testing time without compromising quality, making it a valuable asset in agile and DevOps environments.

The ultimate goal of all QA methodologies is to ensure high software quality, defined by functionality, reliability, performance, and user satisfaction. Methodological research on software quality often employs mixed methods, combining quantitative metrics such as defect density and customer satisfaction scores with qualitative insights from stakeholder interviews. For instance, Ramchand et al. (2021) used a combination of surveys and performance metrics to evaluate how AI-driven QA processes improved software deliverables. Results indicate that AI integration not only enhances defect detection and testing efficiency but also contributes to higher user satisfaction and reduced post-release defects.

While AI-driven quality assurance offers significant benefits, the implementation of AI tools comes with several challenges. One of the primary obstacles is the initial cost and resource allocation required for integrating AI systems into existing QA infrastructures. Research by Khaliq et al. (2022) found that many organizations, particularly smaller ones, struggle with the financial investment needed to implement advanced AI tools. Furthermore, a study by Ramchand et al. (2021) highlighted that organizations often face difficulties in training their staff to adapt to AI-based systems, as a lack of expertise can lead to underutilization of AI tools. These challenges necessitate a careful consideration of organizational readiness and a gradual implementation strategy that balances the costs with the expected returns in terms of software quality and efficiency.

Another key aspect of AI-driven QA is its alignment with modern software development practices, such as continuous integration (CI) and agile methodologies. AI tools are especially valuable in agile environments, where the need for rapid testing and continuous feedback is paramount. Studies by Kwasek et al.

(2024) explored how AI technologies, such as machine learning and natural language processing, facilitate real-time defect detection and code quality assessment during frequent software iterations. These tools help ensure that quality is maintained throughout the development lifecycle, reducing the time between iterations and speeding up product releases without sacrificing the quality of the software. As a result, AI's role in agile environments is pivotal, as it enables testers to identify issues earlier in the development process, which can ultimately improve the overall user experience.

AI's ability to optimize test coverage is another area where significant advancements have been made. In traditional manual testing, ensuring complete test coverage is a complex and time-consuming task. However, AI-driven testing tools can analyze vast amounts of code and data to identify the most critical areas to test based on risk factors, usage patterns, and historical defect data. This shift towards risk-based testing is becoming increasingly important as software systems grow in complexity. A study by Job (2021) demonstrated how AI can prioritize test cases based on the likelihood of defects in different modules, allowing for more focused testing efforts. This results in more efficient use of resources, as AI tools automatically adjust test suites to maximize the detection of high-risk issues while reducing redundant testing efforts.

Predictive quality assurance is an emerging trend driven by AI's ability to predict software quality and potential defects before they occur. Using historical data, machine learning models can forecast areas of the software most likely to encounter issues, allowing QA teams to proactively address these areas before they become significant problems. Studies have shown that AI-based predictive models can significantly reduce post-release defects, as they enable early intervention and timely resolution of issues. For instance, research by Ouaarous et al. (2024) highlighted how predictive analytics could estimate defect densities based on historical project data, leading to more targeted testing efforts. This shift from reactive to proactive QA is one of the most promising aspects of AI-driven software quality assurance, as it allows organizations to not only detect but also prevent defects from occurring.


https://doi.org/10.53056/njmsr-2026.9.1.004 https://www.nepjol.info/index.php/njmsr

In summary, the international perspective on AI-driven QA highlights the increasing shift toward automation, predictive analytics, and intelligent test optimization. Developed economies have successfully integrated AI-driven QA into their workflows, while emerging markets face challenges related to infrastructure, expertise, and cost. However, the growing accessibility of AI technologies and global research collaborations indicate that AI-driven QA is on the path to becoming a standardized practice in software testing across different regions. The insights gained from global implementations can serve as a guiding framework for developing nations like Nepal to harness AI-driven QA effectively, improving software quality and advancing their IT industry.

## 3. Theoretical Framework

This study is grounded in the Technology Acceptance Model (TAM) and Quality Management Theory (QMT) to explain the adoption and effectiveness of Artificial Intelligence (AI) in Quality Assurance (QA) processes within the Nepalese IT industry. TAM posits that the acceptance and use of new technologies are primarily determined by perceived usefulness and perceived ease of use (Venkatesh & Davis, 2000). In QA contexts, professionals are more likely to adopt AI-driven tools when these technologies demonstrably enhance testing efficiency, accuracy, and decision-making while remaining compatible with existing workflows (Fan, Xie, Zheng, Liang, & Di, 2023). Prior research indicates that AI-based testing tools capable of automating test generation, improving defect detection, and accelerating AI testing cycles positively influence users' perceptions of usefulness, thereby increasing adoption intention and actual usage (Ramadan, Yasin, & Pektas, 2024).

Complementing TAM, Quality Management Theory emphasizes continuous improvement, process optimization, defect prevention, and data-driven decision-making as core principles for achieving high-quality outcomes (Davenport & Kalakota, 2019). In software development, quality is achieved not only through final inspection but through systematic improvement of testing processes, reliable data management, and continuous feedback across the development lifecycle (Garousi, Felderer, & Mäntylä, 2016). AI-driven QA aligns closely with these

principles by enabling predictive defect analysis, intelligent test prioritization, improved test coverage, and faster execution cycles, thereby enhancing consistency and reliability in QA processes (Omri & Sinz, 2021).

By integrating TAM and Quality Management Theory, this study captures both the human acceptance dimension of AI adoption and the process-oriented mechanisms through which AI enhances software quality. This integrated theoretical foundation provides a robust basis for examining how AI implementation factors such as testing techniques, testing speed, data quality, automation, and AI tool implementation contribute to improved software quality outcomes in the Nepalese IT industry.

The AI-Driven Quality Assurance Effectiveness Model (AI-QAEM) is developed by integrating the Technology Acceptance Model (TAM) and Quality Management Theory to explain how AI adoption influences software quality outcomes. Consistent with TAM, the model assumes that QA professionals are more likely to adopt AI tools when they perceive them as useful in enhancing testing accuracy, efficiency, and defect detection, and when the tools are easy to use within existing QA workflows. These perceptions play a critical role in determining the successful implementation of AI-driven QA practices.

From a Quality Management Theory perspective, the model posits that effective AI implementation improves QA processes through enhanced testing techniques, increased testing speed, improved data quality, and optimized automation of test scripts. These process improvements collectively support continuous quality enhancement by reducing errors, improving test coverage, and enabling data-driven decision-making. As a result, optimized QA processes lead to higher software quality in terms of reliability, performance, and consistency.

Overall, the AI-QAEM suggests that when AI tools are effectively adopted and integrated into QA processes, they facilitate continuous improvement and superior quality outcomes. The model provides a structured framework for examining how AI-driven QA components interact to influence software quality, offering empirical insights into the mechanisms through which AI enhances Quality Assurance in the Nepalese IT industry.


https://doi.org/10.53056/njmsr-2026.9.1.004 https://www.nepjol.info/index.php/njmsr

## 4. Methodology

This study adopted a descriptive research design to explore the adoption, impact, and challenges of AI-driven Quality Assurance (QA) in the Nepalese IT industry. The population included QA professionals working in Nepalese IT companies, while a sample of approximately 230 respondents was selected using purposive sampling to ensure relevant expertise. The study utilized both primary and secondary data. Primary data were collected through online questionnaires, while secondary data were obtained from journals, reports, and industry publications related to AI in QA. The data collection procedure involved distributing Google Forms via email or social media. Collected data were analyzed using descriptive statistics for quantitative responses. This quantitative approach ensures a comprehensive understanding of AI's role in transforming QA practices in Nepal's IT sector.

The reliability analysis using Cronbach's Alpha shows an overall value of 0.906, indicating excellent internal consistency. All variables demonstrate acceptable reliability, with alpha values ranging from 0.719 to 0.886. Software Quality (0.886), Testing Techniques (0.838), and Testing Speed (0.831) exhibit strong reliability, while others also meet the acceptable threshold. These results confirm that the study's measurement scales are reliable for further analysis.

## 5. Data Analysis

To analyze the relationship between the independent and dependent variables, a regression model is employed. The regression model used in this study is as follows:

$$SQ = \beta_0 + \beta_1 IAT + \beta_2 TT + \beta_3 DQ + \beta_4 ATS + \beta_5 TS + \in$$

Dependent Variable SQ = Software Quality and Independent Variables IAT = Implementation of AI Tools, TT = Testing Techniques, DQ = Data Quality, ATS = Automated Test Scripts, TS = Testing Speed, $\in$ = Error, $\beta_1$, $\beta_2$, $\beta_3$, $\beta_4$, $\beta_5$ are

 https://doi.org/10.53056/njmsr-2026.9.1.004

regression coefficients that refer to the amount with which a dependent variable increases when one of these independent variables increases while others remain constant.

## 6. Results and Discussions

The data and information gathered from respondents were systematically presented, interpreted, and analyzed. The questionnaire was distributed using a purposive sampling method to collect relevant responses. The collected data were then entered into SPSS for detailed statistical analysis. Various analytical techniques, including frequency distribution, cross-tabulation, correlation, and reliability analysis, were performed and interpreted to derive meaningful insights.

### a. Descriptive Analysis

The demographic profile of the respondents shows that nearly half (49.1%) fall within the 20–25 age group, followed by 39.6% aged 26–30, indicating that the QA workforce is predominantly young. In terms of gender, 60% of respondents are male and 40% are female.

Table 1: Demographic information of respondents

| Variables | Category | Percent |
|---|---|---|
| Age of respondents | 20-25<br>26-30<br>31-40<br>41-45<br>46 and above | 49.1<br>39.6<br>10.4<br>0<br>0.9 |
| Gender of respondents | Male<br>Female | 60<br>40 |
| Experience as a QA of respondents | < 1 year<br>1-3 years<br>3-5 years<br>5 years and above | 38.3<br>40.9<br>9.6<br>11.3 |


https://doi.org/10.53056/njmsr-2026.9.1.004 https://www.nepjol.info/index.php/njmsr

| | | |
|---|---|---|
| Professional Level of respondents | Intern | 16.5 |
| | Junior | 26.5 |
| | Mid | 30 |
| | Senior | 17 |
| | Lead | 6.1 |
| | Manager | 3.9 |

Source: SPSS (2025)

Regarding professional experience, 40.9% have 1–3 years of experience and 38.3% have less than one year, suggesting that many participants are early in their QA careers. The professional level distribution further supports this, with the majority being mid-level (30%), followed by junior (26.5%) and intern positions (16.5%). Senior-level respondents account for 17%, while lead and managerial positions represent 6.1% and 3.9% respectively. Overall, the sample reflects a youthful QA population with a strong concentration in early- to mid-career stages.

**b. Inferential Analysis**

In the inferential analysis, the study utilized several statistical tests, including the Kaiser-Meyer-Olkin (KMO) measure, communalities assessment, eigenvalue analysis, one-sample testing, and regression analysis.

Table 2: Analysis of KMO and Bartlett's Test

| **KMO and Bartlett's Test** | | |
|---|---|---|
| Kaiser-Meyer-Olkin Measure of Sampling Adequacy. | | .880 |
| Bartlett's Test of Sphericity | Approx. Chi-Square | 626.525 |
| | df | 15 |
| | Sig. | <.001 |

Source: SPSS (2025)

The results of the Kaiser-Meyer-Olkin (KMO) and Bartlett's Test confirm the suitability of the dataset for factor analysis. The KMO Measure of Sampling Adequacy is 0.880, which, according to Kaiser (1974), is considered *meritorious*, indicating that the sample size is adequate and the variables share common factors. Furthermore, Bartlett's Test of Sphericity shows a Chi-square value of 626.525 with 15 degrees of freedom and a significance level of less than 0.001, confirming that the correlation matrix is not an identity matrix. This means the variables are

sufficiently correlated and appropriate for factor extraction. Overall, these results validate that the data meet the statistical assumptions necessary for factor analysis.

Table 3: Analysis of Communalities

| **Communalities** | | |
|---|---|---|
| | Initial | Extraction |
| IAT | 1.000 | .613 |
| DQ | 1.000 | .404 |
| ATS | 1.000 | .462 |
| TT | 1.000 | .725 |
| TS | 1.000 | .717 |
| SQ | 1.000 | .712 |
| Extraction Method: Principal Component Analysis. | | |

Source: SPSS (2025)

The communalities table shows how much of each variable's variance is explained by the extracted factors. Initially, all variables have a value of 1.000, indicating that each variable's total variance is considered before extraction. After extraction using Principal Component Analysis (PCA), the communalities range from 0.404 to 0.725, meaning that between 40.4% and 72.5% of the variance in each variable is explained by the factors retained in the analysis. Specifically, **TT (0.725), TS (0.717),** and **SQ (0.712)** have the highest communalities, indicating they are well represented by the extracted components. In contrast**, DQ (0.404)** has the lowest value, suggesting it is less strongly associated with the common factors. Overall, the results indicate that most variables are adequately explained by the factor model, supporting the suitability of PCA for the dataset.

Table 4: Analysis of Total Variance

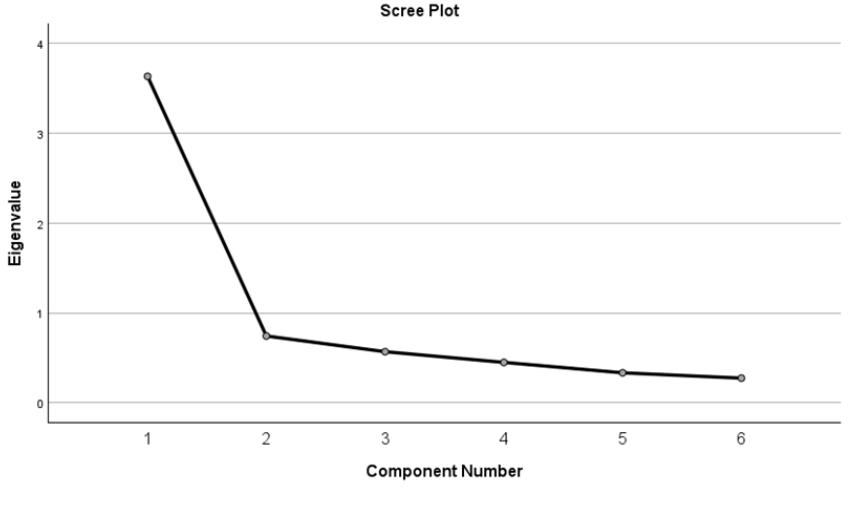
| **Total Variance Explained** | | | | | | |
|---|---|---|---|---|---|---|
| Component | Initial Eigenvalues | | | Extraction Sums of Squared Loadings | | |
| | Total | % Of Variance | Cumulative % | Total | % Of Variance | Cumulative % |
| SQ | 3.634 | 60.570 | 60.570 | 3.634 | 60.570 | 60.570 |


https://doi.org/10.53056/njmsr-2026.9.1.004 https://www.nepjol.info/index.php/njmsr

| | | | | | | |
|---|---|---|---|---|---|---|
| IAT | .743 | 12.376 | 72.946 | | | |
| DQ | .568 | 9.467 | 82.413 | | | |
| ATS | .449 | 7.478 | 89.891 | | | |
| TT | .333 | 5.549 | 95.440 | | | |
| TS | .274 | 4.560 | 100.000 | | | |
| Extraction Method: Principal Component Analysis. | | | | | | |

Source: SPSS (2025)

The **Total Variance Explained** table summarizes the amount of variance accounted for by each principal component. The results show that only the **first component** has an eigenvalue greater than 1 (3.634), explaining **60.57% of the total variance**. This indicates that a single dominant factor accounts for the majority of the shared variance among the variables. The remaining components have eigenvalues less than 1 and contribute comparatively little to explaining the total variance, with the second component explaining only 12.38% and subsequent components contributing even less. According to the **Kaiser criterion** (eigenvalues > 1) and the **cumulative variance rule,** only the first component should be retained for further analysis. Therefore, the extraction results suggest that one principal factor sufficiently represents the underlying structure of the dataset, explaining a substantial portion of the total variance.




Source: SPSS (2025)

Figure 1: Scree Plot

The Scree Plot shows a sharp decline after the first component, followed by a gradual flattening of the curve, indicating a clear "elbow" point. This suggests that only one component should be retained, as it explains most of the total variance while the remaining components contribute minimally. This finding supports the **Total Variance Explained** result, where the first component accounted for **60.57%** of the total variance, confirming that a single factor sufficiently represents the dataset.

## Correlation

The Pearson correlation results in Table 15 indicate a statistically significant positive relationships between Software Quality (SQ) and all independent variables:

Table 5: Correlation Analysis

*(Table 5 shows the relation between variables. Where Software Quality (SQ) is dependent variable and Implementation of AI Tools (IAT), Data Quality (DQ), Automated Test Scripts (ATS), Testing Techniques (TT), Testing Speed (TS) are independent variables. The result is based on sample size 230.)*

| **Correlations** | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | SQ | IAT | DQ | ATS | TT | TS |
| SQ | Pearson Correlation | 1 | | | | | |
| | Sig. (2-tailed) | | | | | | |
| IAT | Pearson Correlation | .563** | 1 | | | | |
| | Sig. (2-tailed) | <.001 | | | | | |
| DQ | Pearson Correlation | .479** | .420** | 1 | | | |
| | Sig. (2-tailed) | <.001 | <.001 | | | | |
| ATS | Pearson Correlation | .449** | .509** | .272** | 1 | | |
| | Sig. (2-tailed) | <.001 | <.001 | <.001 | | | |


https://doi.org/10.53056/njmsr-2026.9.1.004 https://www.nepjol.info/index.php/njmsr

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| TT | Pearson Correlation | .661** | .589** | .452** | .519** | 1 | |
| | Sig. (2-tailed) | <.001 | <.001 | <.001 | <.001 | | |
| TS | Pearson Correlation | .719** | .560** | .438** | .471** | .689** | 1 |
| | Sig. (2-tailed) | <.001 | <.001 | <.001 | <.001 | <.001 | |
| **. Correlation is significant at the 0.01 level (2-tailed). | | | | | | | |

Source: SPSS (2025)

Implementation of AI Tools (IAT), Data Quality (DQ), Automated Test Scripts (ATS), Testing Techniques (TT), and Testing Speed (TS), all show statistically significant positive relationships with Software Quality at the 0.01 significance level. Among these, Testing Speed (r = .719) shows the strongest correlation with software quality, suggesting that faster testing enabled by AI tools significantly enhances software performance and reliability. Testing Techniques (r = .661) and Implementation of AI Tools (r = .563) also demonstrate strong associations, implying that effective AI implementation and advanced testing methodologies contribute notably to improving quality outcomes. Meanwhile, Data Quality (r = .479) and Automated Test Scripts (r = .449) show moderate yet meaningful relationships, indicating their supportive roles in maintaining consistency and accuracy in testing. Overall, the findings suggest that AI-driven components collectively and positively influence software quality in the Nepalese IT industry.

**Regression**

Table 6: Regression Analysis

*(Table 6 represents the result based on the 230 samples. By using regression model: $SQ = \beta 0 + \beta 1IAT + \beta 2TT + \beta 3DQ + \beta 4ATS + \beta 5TS + \in$, where Software Quality (SQ) is dependent variable and Implementation of AI Tools (IAT), Data Quality (DQ), Automated Test Scripts (ATS), Testing Techniques (TT), Testing Speed (TS) are independent variables.)*

| **Coefficients** | | | | | | | |
|---|---|---|---|---|---|---|---|
| Model | B | t | Sig. | F | Sig | R. square | Durbin Watson |
| (Constant) | -.239 | -1.081 | .281 | 66.752 | 0.00 | 0.774 | 1.917 |
| IAT | .136 | 2.149 | .033 | | | | |
| DQ | .147 | 2.714 | .007 | | | | |
| ATS | .041 | .742 | .459 | | | | |
| TT | .234 | 3.338 | <.001 | | | | |
| TS | .480 | 6.892 | <.001 | | | | |
| a. Dependent Variable: SQ | | | | | | | |

Source: SPSS (2025)

The regression results indicate that the model explaining Software Quality (SQ) is statistically significant, with an F-value of 66.752 and a p-value of 0.000, confirming that the independent variables collectively have a strong effect on software quality. The $R^2$ value of 0.774 shows that approximately 77.4% of the variation in software quality can be explained by the predictors: Implementation of AI Tools (IAT), Data Quality (DQ), Automated Test Scripts (ATS), Testing Techniques (TT), and Testing Speed (TS). The Durbin-Watson statistic of 1.917 suggests that there is no serious autocorrelation problem, supporting the reliability of the model. Among the variables, Testing Speed ($\beta = 0.480$, $p < 0.001$) and Testing Techniques ($\beta = 0.234$, $p < 0.001$) have the strongest and most significant positive impacts on software quality. Data Quality ($\beta = 0.147$, $p = 0.007$) and Implementation of AI Tools ($\beta = 0.136$, $p = 0.033$) also show significant positive effects. However, Automated Test Scripts ($\beta = 0.041$, $p = 0.459$) is statistically insignificant, indicating it does not substantially influence software quality in this model. Overall, the findings highlight that faster testing, effective testing techniques, and reliable data, supported by AI tools, play crucial roles in enhancing software quality within AI-driven QA environments.


https://doi.org/10.53056/njmsr-2026.9.1.004 https://www.nepjol.info/index.php/njmsr

## Hypothesis Testing

The hypothesis testing results show how different factors influence Software Quality. For H1, the relationship between Implementation of AI and Software Quality is significant, as the obtained p-value (0.033) is below the 0.05 threshold, leading to acceptance of the hypothesis. Similarly, H2 is accepted because Data Quality shows a strong significant relationship with Software Quality (p = 0.007).

Table 7: Hypothesis Testing

| Code | Hypothesis | Obtained p-Value | Threshold p-Value | Result |
|---|---|---|---|---|
| H1 | There is a significant relationship between Implementation of AI and Software Quality. | 0.033 | <0.05 | Accepted |
| H2 | There is a significant relationship between Data Quality and Software Quality. | 0.007 | <0.05 | Accepted |
| H3 | There is a significant relationship between Automated Test Scripts and Software Quality. | 0.459 | > 0.05 | Rejected |
| H4 | There is a significant relationship between Testing Techniques and Software Quality. | <0.01 | <0.05 | Accepted |
| H5 | There is a significant relationship between Testing Speed and Software Quality. | <0.01 | <0.05 | Accepted |

However, H3 is rejected because the p-value for Automated Test Scripts (0.459) is greater than 0.05, indicating no significant relationship with Software Quality. In contrast, both H4 and H5 are accepted, as Testing Techniques and Testing Speed have p-values less than 0.01, demonstrating strong significant relationships with Software Quality. Overall, these findings indicate that most tested variables significantly contribute to software quality, except Automated Test Scripts.

## 7. Limitations of the study

Despite its contributions, this study has several limitations that should be acknowledged. First, the use of purposive sampling limits the generalizability of the

findings beyond the sampled QA professionals and IT firms in Nepal. Although this approach ensured respondents had relevant QA experience, the results may not fully represent all segments of the Nepalese IT industry, particularly smaller firms or organizations with minimal exposure to AI technologies. Second, the study relies on self-reported perceptions, which may be subject to response bias and individual interpretation of AI usage and effectiveness. Third, the level of AI adoption in Nepal is still emerging, meaning that some respondents may have limited hands-on experience with advanced AI-driven QA tools. As a result, the findings primarily reflect early-stage adoption rather than mature AI integration. Finally, the cross-sectional nature of the study captures relationships at a single point in time and does not account for how AI-driven QA practices and their impact on software quality may evolve. Future research could address these limitations by using longitudinal designs, probabilistic sampling, and comparative studies across different countries or maturity levels of AI adoption.

## 8. Conclusion

This study concludes that AI-driven Quality Assurance plays a significant and positive role in enhancing software quality within the Nepalese IT industry. The strong KMO value of 0.880 and significant Bartlett's Test confirm that the dataset was highly suitable for factor analysis, ensuring the validity of the statistical findings. The regression model further demonstrated that 77.4% of the variation in software quality is explained by the selected predictors, highlighting the robustness of the analytical framework. Testing Speed and Testing Techniques emerged as the most influential factors, followed by Data Quality and Implementation of AI Tools, all showing significant positive relationships with software quality. Although Automated Test Scripts showed no meaningful impact, the overall results emphasize that integrating AI, improving data integrity, and optimizing testing processes substantially improve QA outcomes. Thus, AI adoption presents a valuable opportunity for Nepalese IT firms to enhance reliability, efficiency, and competitiveness in software development.


https://doi.org/10.53056/njmsr-2026.9.1.004 https://www.nepjol.info/index.php/njmsr

## References

Broadway Infosys. (2025). The Role of AI and Automation in Nepal's IT Job Market: AI and Automation in Nepal's IT Jobs.

Bussa, S. (2023). Artificial Intelligence in Quality Assurance for Software Systems.

Davenport, T., & Kalakota, R. (2019). The potential for artificial intelligence in healthcare. *Future Healthcare Journal*.

Fan, G., Xie, X., Zheng, X., Liang, Y., & Di, P. (2023). Static Code Analysis in the AI Era: An In-depth Exploration of the Concept, Function, and Potential of Intelligent Code Analysis Agents.

Garousi, V., Felderer, M., & Mäntylä, M. (2016). The need for multivocal literature reviews in software engineering: Complementing systematic literature reviews with grey literature. *Information and Software Technology*.

Islam, M., Khan, F., Alam, S., & Hasan, M. (2023). Artificial Intelligence in Software Testing: A Systematic Review.

Job, M. A. (2021). Automating and Optimizing Software Testing using Artificial Intelligence Techniques.

Jobin, A., Ienca, M., & Vayena, E. (2019). The global landscape of AI ethics guidelines. *Nature Machine Intelligence*.

Khaliq, Z., Farooq, S. U., & Khan, D. (2022). Artificial Intelligence in Software Testing : Impact, Problems, Challenges and Prospect.

Khan, M. I., Mahmud, F. U., Hossen , A., & Masum, A. M. (2024). A NEW APPROACH OF SOFTWARE TEST AUTOMATION USING AI. *21*(1), 559-570.

Khan, S. A. (n.d.). AI - Based Software Testing.

Khanal, B. P. (2024). Information Technology in Nepal: History and Current Status.

Kwasek, A., Kocot, M., Kocot, D., & Maciaszczyk, M. (2024). The Role of Artificial Intelligence in Agile Organization Management. *XXVII*(2), 118-130.

LambdaTest. (2023). Future of Quality Assurance.

Omri, S., & Sinz, C. (2021). Machine Learning Techniques for Software Quality Assurance: A Survey.

Ouaarous, R., Hilal, I., & Mezrioui, A. (2024). On Using Artificial Intelligence in Software Quality Assurance: A State of the Art.

Ramadan, A., Yasin, H. N., & Pektas, B. (2024). The Role of Artificial Intelligence and Machine Learning in Software Testing.

Ramchand, S., Shaikh, S., & Alam, I. (2021). Role of Artificial Intelligence in Software Quality Assurance. *2*.

Sinha, K., & Jana, D. S. (2025). AI-Powered Software Testing: Transforming Quality Assurance through Artificial Intelligence.

Thant, K. S., & Tin, H. H. (2023). The impact of manual and automation testing on software testing efficiency and effectiveness.

Thomas, A. T. (2025). Review of AI-Driven Approaches for Automated Defect Detection and Classification in Software Testing.

Venkatesh, V., & Davis, F. D. (2000). A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies.



https://doi.org/10.53056/njmsr-2026.9.1.004 https://www.nepjol.info/index.php/njmsr