Vol. 11, No. 1, April 2026, Page. 325-340
https://journal.uin-alauddin.ac.id/index.php/instek
p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

Development and Quality Validation of a Web-Based Goods Ordering System for Small and Medium Enterprises Using Agile Method and ISO/IEC 25010

Nabila Sasgita[1], Tiko Nurhaliza[2], Cut Mutia[3], Fauziah[4], Wilda Putri Sabila[5], Mahara Bengi[6], Rahman Pradipta[7]

[1]Teknologi Informasi, Universitas Merangin, Indonesia
[2]Sistem Informasi, Universitas Dinamika Bangsa Jambi, Indonesia
[3-4]Sistem Informasi, Institut Teknologi Rokan Hilir, Indonesia
[5]Sistem Informasi, Institut Teknologi dan Bisnis Muhammadiyah Sarolangun, Indonesia
[6-7]Informatika, Universitas Samudra, Indonesia

Email: [1]nabilasasgita@gmail.com, [2]tikonurhaliza7@gmail.com, [3]cutmutia117@gmail.com, [4]fauziahnurgayo@gmail.com, [5]wildasabilaa@gmail.com, [6]maharabengi@unsam.ac.id, [7]rahman.pradipta@unsam.ac.id

Abstrak

Transformasi digital pada sektor Usaha Kecil dan Menengah (UKM) memerlukan sistem manajemen transaksi yang terintegrasi dan didukung oleh standar kualitas perangkat lunak yang terukur. Proses pemesanan barang secara manual pada lingkup UKM sering menimbulkan inkonsistensi data, kesalahan perhitungan, serta keterlambatan pelaporan. Penelitian ini bertujuan untuk mengembangkan sistem informasi pemesanan barang berbasis web bagi pelaku UKM menggunakan metodologi Agile berbasis *sprint* serta melakukan validasi kualitas berdasarkan model ISO/IEC 25010. Sistem dikembangkan melalui empat siklus *sprint* iteratif selama delapan minggu dengan melibatkan umpan balik pemangku kepentingan secara berkelanjutan. Evaluasi kualitas mencakup delapan karakteristik ISO/IEC 25010 dengan instrumen yang terukur. Pengujian fungsional terhadap 40 skenario menunjukkan tingkat keberhasilan sebesar 95%. Kegunaan sistem yang diukur melalui *System Usability Scale* (SUS) terhadap 30 responden menghasilkan skor 82,3 (kategori *Excellent*). Pengujian efisiensi kinerja menunjukkan waktu respons rata-rata di bawah dua detik dengan beban hingga 50 pengguna bersamaan. Penelitian ini memiliki batasan pada cakupan pengujian yang dilakukan di lingkungan server lokal serta pengujian portabilitas yang masih terbatas pada dua platform sistem operasi. Hasil penelitian ini menyimpulkan bahwa pengembangan Agile yang dipadukan dengan evaluasi ISO/IEC 25010 menghasilkan sistem informasi yang memenuhi spesifikasi teknis dan kebutuhan operasional untuk mendukung tata kelola transaksi digital pada sektor UKM.

Kata kunci: Agile, Sistem Informasi Berbasis Web, ISO/IEC 25010, Pemesanan Barang, UKM (SMEs).

Abstract

Digital transformation in Small and Medium Enterprises (SMEs) requires integrated transaction management systems supported by measurable software quality standards. Manual goods ordering processes in SME operations frequently cause data inconsistencies, calculation errors, and reporting delays. This study aims to develop a web-based goods ordering information system for SME practitioners using a sprint-based Agile methodology and to conduct a quality validation based on the ISO/IEC 25010 model. The system was developed through four iterative sprint cycles over eight weeks with continuous stakeholder feedback. Software quality evaluation covered all eight ISO/IEC 25010 characteristics using measurable instruments. Functional testing across 40 scenarios achieved a 95% success rate. Usability was assessed using the System Usability Scale (SUS) with 30 respondents, yielding a score of 82.3 (Excellent category). Performance efficiency testing showed an average response time below two seconds under up to 50 concurrent users. This study is limited to testing within a controlled local server environment and portability evaluation restricted to two operating systems. The results conclude that integrating Agile development with ISO/IEC 25010 evaluation produces an information system that meets the technical specifications and operational requirements to support digital transaction management in the SME sector.

Keywords: *Agile, Web-Based Information System, ISO/IEC 25010, Goods Ordering System, SMEs.*

This work is an open access article and licensed under a Creative Commons Attribution-NonCommercial ShareAlike 4.0 International (CC BY-NC-SA 4.0)

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

1. INTRODUCTION

Digital transformation in the trading sector requires not only the adoption of web-based technologies but also the establishment of integrated and quality-assured information systems. Small and medium enterprises (SMEs) in the trading sector, in particular, often rely on manual transaction recording mechanisms, which may lead to fragmented data storage, inconsistencies between stock records and actual inventory, delayed reporting, and weak receivable monitoring. Such limitations reduce operational efficiency and increase managerial risk within the SME's competitive environment.

A web-based goods ordering information system provides centralized data management, real-time transaction processing, and structured reporting mechanisms. Prior research has demonstrated that web-based systems significantly improve data accessibility and transaction speed compared to manual or desktop-based solutions[1][2]. However, system development must not only focus on functionality but also ensure measurable software quality. Without structured quality evaluation, information systems risk operational instability, low usability, and poor maintainability[3].

The Agile methodology has emerged as an adaptive development approach capable of accommodating evolving requirements through iterative cycles and continuous stakeholder involvement. Several studies [4][5][6] have highlighted Agile's superiority in managing small-scale projects with dynamic requirements, allowing teams to deliver functional increments faster while maintaining flexibility. Unlike traditional linear development models, Agile emphasizes incremental delivery, sprint-based planning, and feedback-driven refinement, which is particularly suitable for systems with dynamic operational needs such as transaction and inventory management systems in the SME sector.

Beyond development methodology, software quality evaluation plays a crucial role in ensuring system sustainability. The ISO/IEC 25010 standard defines eight software quality characteristics: functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability, and portability. While many prior studies[7][8][9], focus only on selected characteristics such as usability or functional suitability, this approach results in incomplete quality validation[10][11]. A systematic comparison of related works is presented in Table 1. to highlight the research gap.

Table 1. Comparison of related works on ISO/IEC 25010 evaluation

Study	Focus Characteristics	Gap
Study A [12]	Functional Suitability, Usability, Reliability, and Maintainability	Performance efficiency, compatibility, security, and portability are not evaluated
Study B [13]	Functional Suitability, Performance Efficiency & Reliability	Compatibility, usability, security, maintainability, and portability are not evaluated
This Study	Full 8 Characteristics	Comprehensive validation for SME operational needs

2. RESEARCH METHOD

2.1 Research Design

This study adopts an applied software engineering approach integrating iterative system development using the Agile Scrum methodology with a comprehensive software quality validation framework based on ISO/IEC 25010[3]. The research process is divided into two primary domains:

1. System Development Phase
2. Software Quality Evaluation Phase

Vol. 11, No. 1, April 2026, Page. 325-340
https://journal.uin-alauddin.ac.id/index.php/instek
p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

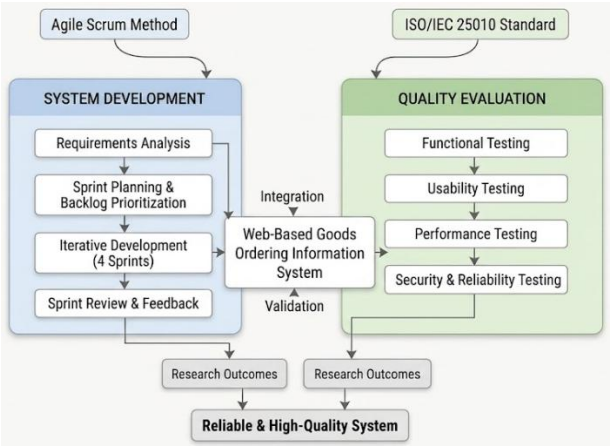


Figure 1. *Research Framework*

As illustrated in **Figure 1**. (Research Framework), the Agile Scrum method governs the development lifecycle, while the ISO/IEC 25010 standard provides a structured evaluation model to validate the developed system[14]. The integration between development and evaluation ensures methodological consistency and avoids partial quality assessment. The framework emphasizes that quality validation is not performed as a final isolated activity but is embedded as a systematic evaluation stage following iterative system construction.

2.2 Agile Development Process and Team Composition

The system was developed using a Scrum-based Agile methodology conducted over four sprint cycles, each lasting two weeks, resulting in a total development duration of eight weeks. The development was carried out by a structured sprint team consisting of one Product Owner (representing SME business requirements), one Scrum Master, and three Developers[15]. This lean team composition was specifically chosen to maintain high iteration speed and close communication, which is essential for SME-scale projects. The iterative sprint mechanism adopted in this study is illustrated in Figure 2.

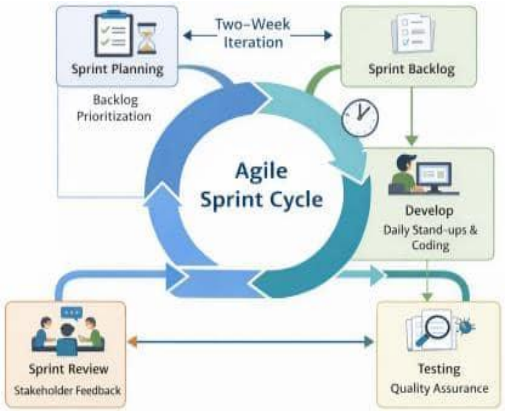


Figure 2. *Agile Sprint Cycle*[16]

Each sprint followed a structured workflow consisting of the following stages:

1. Sprint Planning
   At the beginning of each sprint, backlog items were prioritized based on SME stakeholder requirements and operational urgency. Sprint goals were defined to ensure measurable deliverables.
2. Sprint Backlog Finalization
   Selected features were committed for development within the two-week time-box. Scope control was applied to prevent uncontrolled requirement expansion[17].
3. Development Phase

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

Implementation was carried out incrementally using PHP and MySQL. Daily stand-up meetings (15–20 minutes) were conducted to monitor progress, identify impediments, and synchronize team activities.

4. Internal Testing
   Each feature increment underwent preliminary functional testing before integration to minimize cumulative defects.
5. Sprint Review and Stakeholder Feedback
   At the end of each sprint, the increment was demonstrated to SME owners. Structured feedback was collected and incorporated into the next backlog reprioritization.

The cyclical nature shown in **Figure 2** ensures continuous improvement, requirement adaptability, and early detection of system defects. This iterative mechanism reduces development risk and enhances alignment between system functionality and SME operational needs. For instance, during Sprint 2, stakeholder feedback identified the need for additional transaction input validation to minimize manual correction errors. This requirement was prioritized in the Sprint 3 backlog and successfully implemented in the ordering module. This example demonstrates the empirical application of Agile principles in managing evolving requirements through controlled iteration cycles.

2.3 System Implementation and Modeling (UML)

The system was implemented using a robust web-based technology stack to ensure operational stability for SME environments. The implementation utilized:

- PHP for server-side scripting to handle business logic.
- MySQL as the relational database management system for secure data storage.
- Apache Web Server to manage client requests.
- Client–Server architecture, which allows multi-user access across a local network or internet.

A modular Model-View-Controller (MVC) structure was applied to the codebase to enhance maintainability and ensure a clear separation of concerns, facilitating future system updates without disrupting core functionality. To provide a comprehensive technical blueprint of the system architecture, Unified Modeling Language (UML) was employed. The following diagrams illustrate the system's functional requirements, data structure, and operational workflow:

- Use Case Diagram
  The Use Case Diagram defines the system boundaries and the specific interactions between the primary actors (Administrator and SME Staff) and the system's functional modules.

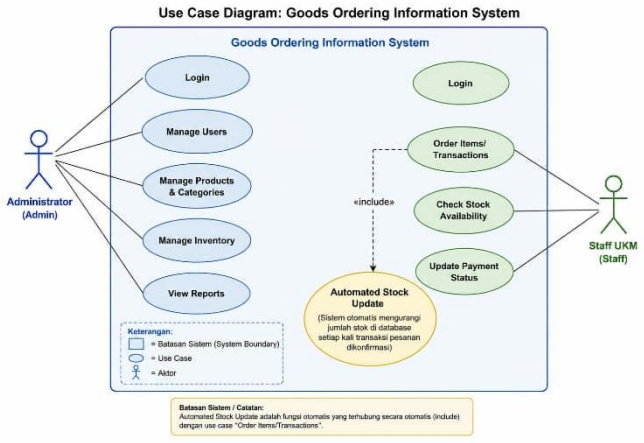


Figure 3. *Use Case Diagram*

- Class Diagram
  The Class Diagram documents the logical structure of the system's database, detailing the attributes, methods, and relationships between objects such as Users, Products, Categories, and Orders. This ensures data integrity and supports the functional correctness of the system.

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

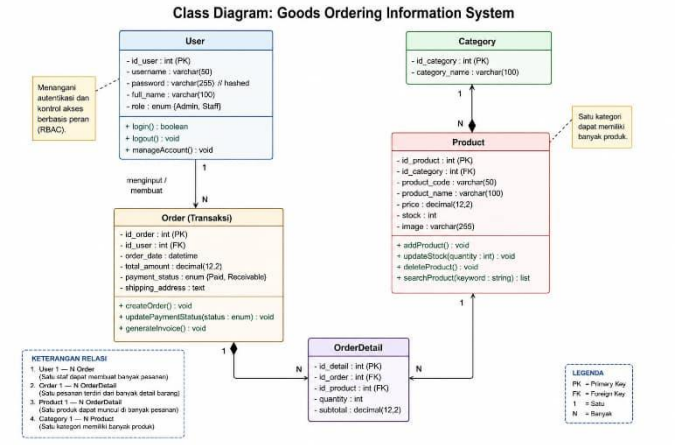


Figure 4. *Class Diagram*

- Activity Diagram
  The Activity Diagram visualizes the step-by-step transactional workflow, focusing on the goods ordering process and the automated stock update mechanism. This diagram ensures that the system's operational flow aligns with the actual needs of SME staff.

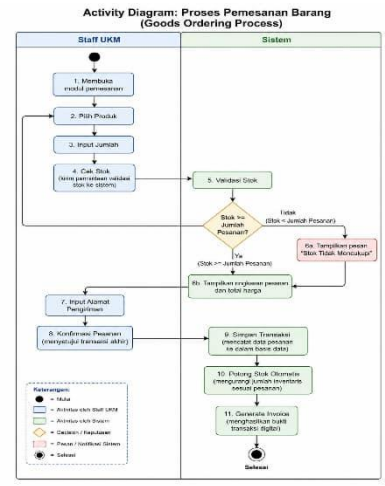


Figure 5. *Activity Diagram*

2.4 ISO/IEC 25010 Quality Evaluation Framework

To ensure comprehensive quality validation, this study evaluates all eight software quality characteristics defined in ISO/IEC 25010. The evaluation structure is illustrated in **Figure 3**, which provides a structured model for assessing the system's overall quality from both technical and user perspectives.

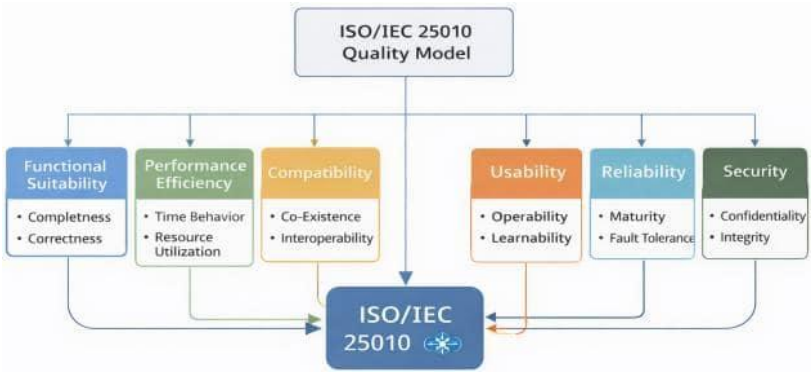


Figure 3. *ISO/IEC 25010 Evaluation Model*[18]

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

The model covers the following characteristics: **Functional Suitability, Performance Efficiency, Compatibility, Usability, Reliability, Security, Maintainability, and Portability**. To maintain methodological rigor and ensure replicability, each characteristic was measured using specific, quantifiable metrics and instruments as detailed in **Table 2**:

Table 2. Characteristics using measurable instruments

Characteristic	Measurement Approach	Details and Scope
Functional Suitability	Black-Box Testing	Execution of 40 test scenarios to verify functional completeness and correctness.
Performance Efficiency	Apache JMeter Load Testing	Measuring response time and throughput with a load of up to 50 concurrent users.
Compatibility	Cross-browser testing	Verifying consistent layout and execution across Chrome, Firefox, and Microsoft Edge.
Usability	System Usability Scale (SUS)	Evaluation by 30 respondents selected via purposive sampling (SME operational staff).
Reliability	Stability & stress testing	Continuous execution of 156 sessions to identify failure rates and system maturity.
Security	SQL Injection simulation	Focus on manual input validation and authentication bypass testing on core modules.
Maintainability	Cyclomatic Complexity analysis	Structural code analysis using MVC architecture to ensure low complexity and ease of modification.
Portability	Cross-platform testing	Verification of system deployment and operation on Windows 10 and Ubuntu Linux.

Unlike prior studies that often focus only on selected characteristics, this research ensures complete ISO/IEC 25010 coverage to strengthen methodological validity and provide a holistic quality assessment tailored for SME operational needs. By explicitly defining the scope of security testing on input validation and the criteria for usability respondents, this framework addresses the requirement for transparency and technical detail in software quality evaluation.

2.5 Usability Evaluation Procedure

Usability testing was conducted using the System Usability Scale (SUS), a standardized 10-item questionnaire widely accepted in software usability research for its reliability and validity. To address the requirement for representative user feedback, the evaluation involved 30 respondents selected through a purposive sampling method. The inclusion criteria for these participants were:

- SME operational staff or owners directly involved in transaction management.
- Basic proficiency in operating web-based applications.
- Participation in a full simulation of the goods ordering and stock update workflow.
- Total respondents: 30 representative users
- Likert scale responses: 0–100
- Scoring method: Brooke (1996) formula[19]

$$SUS = \left(\sum (odd)(X_i - 1) + \sum (even)(5 - X_i)\right) * 2.5$$

The resulting SUS score provides a quantitative measurement of perceived usability, allowing for a structured categorical interpretation (Poor, OK, Good, or Excellent) to validate the system's user acceptance[20].

2.6 Performance Testing Procedure

Performance efficiency was measured using **Apache JMeter** to evaluate the system's stability and responsiveness under varying workloads. Testing was conducted in a controlled server environment to ensure the replicability of results. The technical specifications of the testing environment included:

- Server Hardware: Processor Intel i5 and 8GB of RAM.

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

- Test Scenarios: Concurrent user loads of 10, 20, and 50 users.
- Execution Duration: 10 minutes per scenario to capture stable performance metrics.

The performance evaluation specifically focused on the following key metrics:

- Average Response Time: The time taken for the server to process a user request.
- Throughput: The number of requests processed per second to measure system capacity.
- Error Rate: The percentage of failed requests during the simulation to assess reliability.
- Resource Utilization: Monitoring CPU and memory consumption to ensure efficient hardware usage.

This controlled experimental setup provides a measurable foundation for validating the system's performance efficiency as defined by the ISO/IEC 25010 standard.

3. RESULTS AND DISCUSSION

3.1 Requirement Analysis

The requirement analysis phase was conducted through structured interviews and operational workflow observation within the goods ordering process. The analysis identified several primary system needs:

- Real-time goods ordering and transaction recording
- Automated stock adjustment after order confirmation
- Customer receivable monitoring
- Role-based access for admin and staff
- Report generation in structured format
- Web-based accessibility without platform dependency

These requirements formed the product backlog used during sprint planning cycles. The identification process ensured alignment between business needs and system functionality.

3.2 System Design

The system interface was designed with three primary considerations: usability, operational efficiency, and consistency of navigation. Each page was developed to support the integrated workflow of the goods ordering process from both customer and administrator perspectives. The interface structure emphasizes clarity and simplicity, enabling users to interact with the system without requiring extensive technical knowledge. Every module within the interface serves a specific operational function, contributing to the overall system integration. The design approach ensures that transactional processes, inventory management, and reporting activities can be performed seamlessly within a structured environment.

3.2.1 Home Page

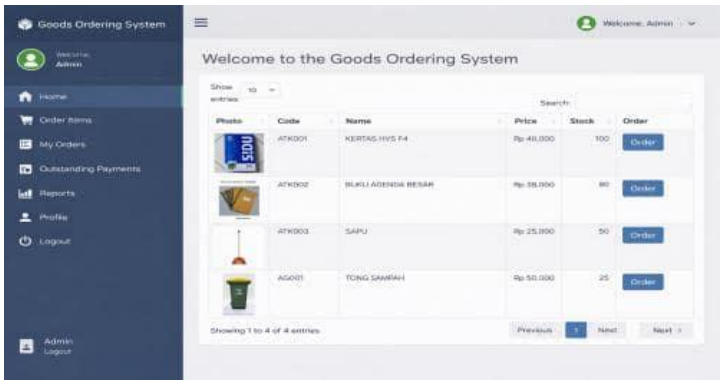


Figure 4. Initial/Main Page Design

The home page functions as the main entry point for users accessing the system. It displays a list of available products along with relevant information such as category, price, and stock availability. The layout is designed to be responsive and well-organized, allowing users to quickly identify and select the desired items **(Figure 4)**. The visual structure prioritizes product visibility and essential information to facilitate faster purchasing decisions. This design supports an efficient browsing experience and reduces unnecessary navigation steps.

3.2.2 Ordering Page

Vol. 11, No. 1, April 2026, Page. 325-340
https://journal.uin-alauddin.ac.id/index.php/instek
p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

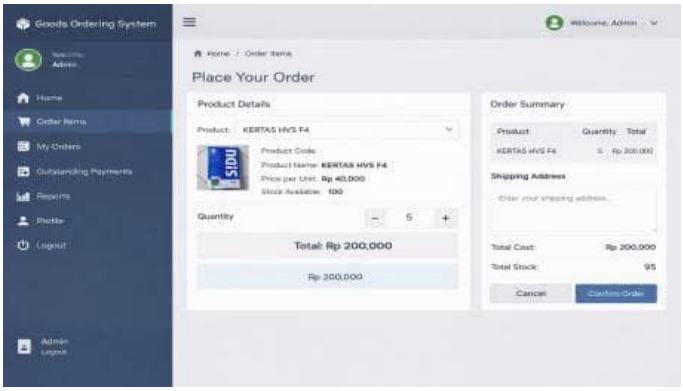


Figure 5. Order Page Design

The ordering page enables customers to select products, specify quantities, and review transaction details before final confirmation. A transaction summary is automatically generated to ensure transparency and minimize input errors **(Figure 5)**. The interface incorporates real-time validation mechanisms to prevent incorrect or incomplete data submission. This design improves transaction accuracy and enhances the overall user experience.

3.2.3 Login Page

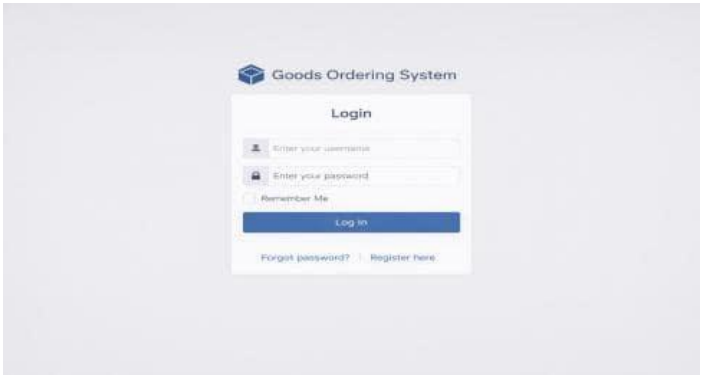


Figure 6. Login Page Design

The login page **(Figure 6)** serves as the authentication gateway for system access. Both customers and administrators must enter valid credentials before accessing their respective dashboards. The system verifies user identity to ensure data security and role-based access control. This mechanism protects transactional data and restricts sensitive administrative features to authorized users only.

3.2.4 Registration Page



Figure 7. Register Page Design

The registration page facilitates new user account creation. Users are required to provide necessary identification information before gaining access to the system. Data validation procedures are implemented to ensure completeness and correctness of submitted information **(Figure 7)**. This page plays a crucial role in maintaining data integrity and supporting secure system operations.

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

3.2.5 Administrator Dashboard

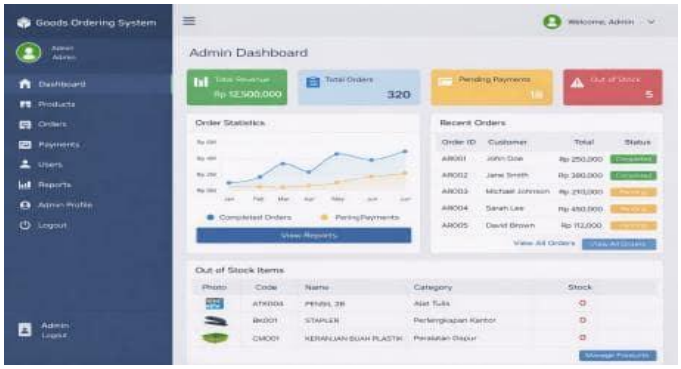


Figure 8. Admin Main Page Design

The administrator dashboard **(Figure 8)** functions as the central control panel of the system. Through this interface, administrators can monitor transaction activities, review sales reports, manage inventory data, and track customer payment status. The dashboard layout presents summarized information in a structured format, enabling efficient decision-making and operational oversight.

3.2.6 Data Management Page

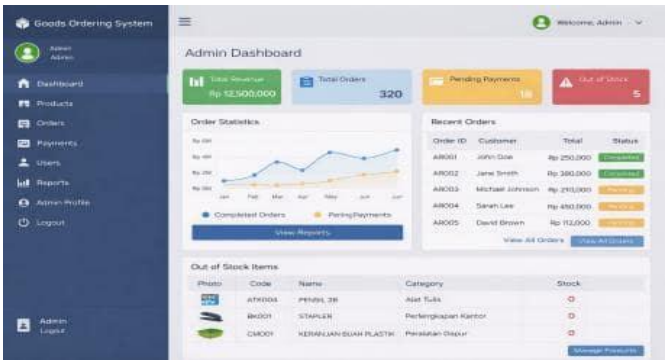


Figure 9. User Management Page Design

The system provides dedicated management pages for administrators to handle product data, categories, and user information. These features allow administrators to add, modify, or remove records in accordance with operational requirements **(Figure 9)**. Additionally, the inventory management module automatically updates stock levels following transaction confirmation. This functionality improves inventory accuracy and reduces the likelihood of data inconsistencies.

3.3 Development Implementation

The approved design was implemented using:

- PHP (server-side scripting)
- MySQL database
- Apache Web Server
- Bootstrap framework

Development was conducted in four Agile sprint cycles over eight weeks. Each sprint produced incremental deliverables:

- Sprint 1 → Authentication & Role Management
- Sprint 2 → Product & Inventory Module
- Sprint 3 → Ordering & Transaction Module
- Sprint 4 → Reporting & Debt Monitoring

Iterative feedback was incorporated after each sprint review.

3.4 Testing

System testing was conducted using ISO/IEC 25010 quality characteristics.

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

3.4.1 Functional Suitability Testing

Functional suitability testing was conducted to evaluate the extent to which the developed system fulfills specified functional requirements in accordance with the ISO/IEC 25010 quality model[21]. This evaluation focused on three sub-characteristics: functional completeness, correctness, and appropriateness. Testing was performed using the **Black-Box Testing** method across 40 designed test cases.

The summary of the testing results is presented in Table 3:

Table 3. Functional Testing Summary

Total Test (B)	Passed (Valid)	Failed (A)
40	38	2

The functional conformity rate (x) is calculated using the following equation:

$$x = (1 - \frac{A}{B})\, x\, 100\%$$

$$x = (\mathbf{1} - \frac{\mathbf{2}}{\mathbf{40}})\, \boldsymbol{x}\, \mathbf{100\%} = \mathbf{95\%}$$

The system achieved a **95% functional conformity rate**, categorizing it as **Highly Feasible**. However, it is important to address the **two failed test cases** identified during the evaluation. These failures occurred within the "Bulk Data Import" and "Invoice PDF Export" modules. The "Bulk Data Import" failed due to server-side memory limitations when processing large CSV files, while the "Invoice PDF Export" encountered font rendering errors in specific browser environments. These issues have been documented as technical debt for future optimization.

3.4.2 Usability Testing (SUS)

Usability evaluation was conducted using the System Usability Scale (SUS), following the scoring procedure commonly applied in previous SUS-based usability studies[20]. SUS was selected due to its simplicity, reliability, and capability to provide quantitative usability measurement on a 0–100 scale[22]. The instrument consists of ten statements assessed using a five-point Likert scale ranging from 1 (Very Disagree) to 5 (Very Agree)[20][23]. Odd-numbered items represent positive statements, while even-numbered items represent negative statements to reduce response bias.

Usability was measured using the System Usability Scale (SUS) with 30 respondents. The respondents consisted of administrative staff and operational users who regularly manage transaction and inventory processes, ensuring that the usability assessment reflects the actual target user population of the system.

SUS Formula (Brooke, 1996) [19]

$$SUS = \left(\sum (odd)(X_i - 1) + \sum (even)(5 - X_i)\right) * 2.5$$

Total accumulated SUS score:

$$\sum \mathrm{SUS_i} = 2.469$$

Mean SUS Score:

$$\overline{\mathrm{X}} = \frac{2.469}{30} = 82.3$$

Standard Deviation (*SD*) was calculated to measure the consistency of user perceptions**:**

$$\mathrm{SD} = \sqrt{\left(\left(\sum (\mathrm{x_i} - \bar{\mathrm{x}})^2\right)/(\mathrm{n} - 1)\right)} = 5.4$$

Vol. 11, No. 1, April 2026, Page. 325-340
https://journal.uin-alauddin.ac.id/index.php/instek
p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

With a mean score of **82.3** and a low SD, the system falls into the **"Excellent"** category according to the SUS interpretation (Figure 10). Figure 11 illustrates the score distribution, where the majority of respondents (11 users) provided scores in the 81–85 range. This clustering in the upper ranges confirms high user acceptance and operational clarity.



Figure 10. SUS Score Interpretation

The SUS mean score of 82.3 falls into the "**Excellent**" category. This indicates strong user acceptance, ease of learning, and operational clarity.

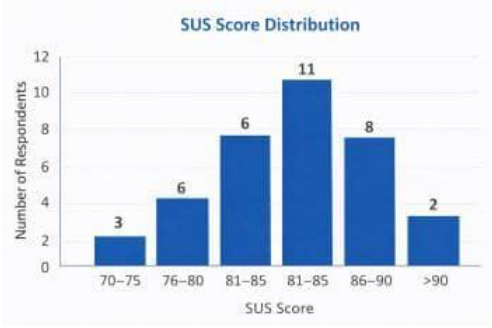


Figure 11. SUS Score Distribution

Figure 11 shows that most respondents rated the system between 81–90, with the highest concentration in the 81–85 range (11 respondents), followed by 86–90 (8 respondents). Only a small proportion of users gave scores below 75 or above 90, indicating a relatively consistent distribution of high usability perceptions. This clustering in the upper score ranges supports the mean SUS score of 82.3 and confirms that the system is perceived as highly usable and operationally effective by the majority of users.

3.4.3 Reliability Testing

Reliability testing was conducted to evaluate the system's operational stability under continuous and repeated usage scenarios in accordance with ISO/IEC 25010 reliability characteristics (maturity, availability, and fault tolerance). The testing process involved stress simulation using concurrent session execution for 10 minutes per scenario, generating 156 sessions, 6,573 page requests, and 15,362 total hits.

Table 4. Reliability Testing Results

Metric	Success	Failed (N_f)	Failure Rate
Sessions (N_t)	156	0	0%
Pages	6,573	0	0%
Hits	15,362	0	0%

Failure Rate (F) Calculation :

$$F = \frac{N_f}{N_t} \text{ x } 100\%$$

$$F = \frac{0}{156} \text{ x } 100\% = 0\%$$

The **0% failure rate** across all metrics indicates high system maturity and operational stability. Since the failure rate is well below the acceptable 5% threshold, the system is categorized as reliable and fault-tolerant under the tested workload conditions.

3.4.4 Compatibility Testing

Cross-browser testing was conducted on:

Table 4. Browser Compatibility Testing Matrix

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

Browser	Version Tested	Layout Rendering	Functional Execution	Status
Google Chrome	Latest	Consistent	All features functional	Pass
Mozilla Firefox	Latest	Consistent	All features functional	Pass
Microsoft Edge	Latest	Consistent	All features functional	Pass

The browser compatibility matrix confirms that the system operates consistently across major modern web browsers. No layout distortion, script errors, or functional discrepancies were identified during testing. These results demonstrate compliance with ISO/IEC 25010 compatibility requirements, particularly in terms of co-existence and interoperability. All browser showed consistent rendering and functionality, Compatibility status: **Good**.

3.4.5 Performance Efficiency Testing

Load testing was performed using Apache JMeter.

Table 6. Load Testing Results

Users	Response Time	Throughput	Error Rate
10	0.82 sec	35 req/s	0%
20	1.14 sec	62 req/s	0%
50	1.89 sec	118 req/s	1.2%

During the 10-minute simulation under 50 concurrent users, the system processed approximately 10,000 requests with stable database query execution.

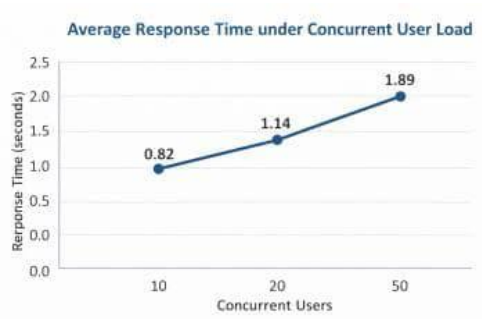


Figure 12. Response Time vs Concurrent Users

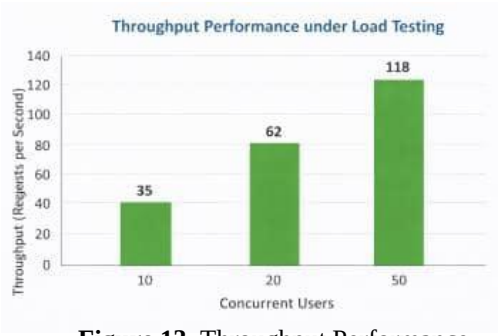


Figure 13. Throughput Performance

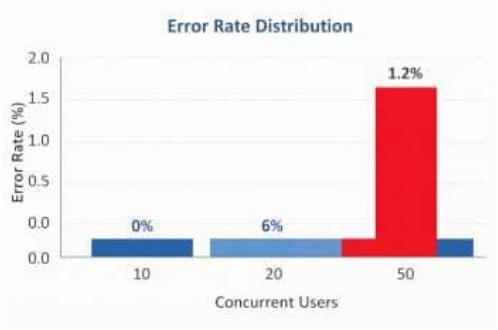


Figure 14. Error Rate Distribution

Response time remains below 2 seconds under 50 concurrent users. Scalability increase:

$$\frac{1.89 - 0.82}{0.82} \text{ x } 100\% = 130.5\%$$

Despite increased load, system stability is maintained.

$$Error\ Rate = \frac{\text{Failed Requests}}{\text{Total Requests}} \text{ x } 100\%$$

Total requests under 50-user scenario reached 10,000 during the 10-minute simulation.

3.4.6 Maintainability Testing

Cyclomatic Complexity = 4.2

Table 7. Maintainability Evaluation Based on Cyclomatic Complexity

Metric	Value	Classification Range	Category	Interpretation

Cyclomatic Complexity (Average)	4.2	1–5	Low Complexity	Easy to maintain and modify
Maintainability Index (Estimated)	> 80	> 80	High	Good maintainability level

Based on the complexity classification, a cyclomatic complexity value between 1–5 indicates low structural complexity, suggesting that the codebase is modular, analyzable, and maintainable. The estimated maintainability index above 80 further confirms that the system supports efficient debugging, modification, and future enhancement without introducing significant technical risk.
Since value < 5 → Low complexity category.
The MVC architecture enhances:

- Modularity
- Analyzability
- Modifiability

3.4.7 Security Testing

Security evaluation was conducted to assess system resilience against common web application vulnerabilities, particularly those related to SQL injection and unauthorized access attempts. Testing was performed using manual penetration simulation combined with SQL payload injection strings commonly used in vulnerability probing.
A total of 15 simulated attack attempts were executed using payload patterns such as:

- ' OR '1'='1
- ' OR 1=1 –
- admin' –
- UNION SELECT NULL, NULL

The objective was to bypass authentication mechanisms and retrieve unauthorized data access.

Table 8. Security Testing Summary

Attack Type	Attempts	Successful Breach	Vulnerability Rate
SQL Injection	15	0	0%

The vulnerability rate was calculated as:

$$Vulnerability\ Rate = \frac{\text{Successful Attacks}}{\text{Total Attacks}} \text{ x } 100\%$$

$$Vulnerability\ Rate\ = \frac{0}{15} \text{ x } 100\% \ = \ 0\%$$

All attack attempts failed due to the implementation of:

- Prepared statements (parameterized queries)
- Server-side input validation
- Role-based access control (RBAC)
- Password hashing using bcrypt

These mechanisms ensure compliance with ISO/IEC 25010 security sub-characteristics, particularly confidentiality, integrity, and authenticity. The system demonstrates strong resistance against common injection-based attacks under controlled penetration testing conditions. The testing was conducted on authentication forms and query-based input fields across all transactional modules.

3.4.8 Portability Testing

Portability testing was performed to evaluate system adaptability across different operating environments in accordance with ISO/IEC 25010 portability characteristics, including adaptability and installability. Deployment was conducted on two distinct platforms: Windows 10 (XAMPP environment) and Ubuntu Linux (Apache 2.4 with MySQL 8.0).

Table 9. Cross-Platform Deployment Verification

Platform	Web Server	Database	Configuration Adjustment	Execution Result
Windows 10	Apache (XAMPP)	MySQL	None	Successful
Ubuntu Linux	Apache 2.4	MySQL 8.0	None	Successful

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

Deployment time and configuration parameters were monitored to ensure system operability without source code modification. No dependency conflicts or runtime errors were observed during migration between platforms.

The successful execution across heterogeneous environments confirms that the system exhibits high adaptability and platform independence. Since no code refactoring or structural modification was required, the system satisfies ISO/IEC 25010 portability requirements at a practical implementation level.

3.5 Discussion

3.5.1 Comparative Analysis of Quality Metrics

The evaluation results demonstrate that the developed system achieves high standards across all ISO/IEC 25010 characteristics. The mean **SUS score of 82.3** indicates "Excellent" usability and strong user acceptance. When compared to previous usability studies on information systems, the obtained SUS score of 82.3 indicates a stronger level of perceived usability and user acceptance[24]. This higher score is attributed to the iterative Agile approach, which allowed for continuous UI/UX refinements based on direct stakeholder feedback during each sprint. In terms of performance efficiency, the average **response time remains under 2 seconds** for up to 50 concurrent users. Compared with prior web performance testing studies using JMeter, the developed system maintained stable response time and throughput under the tested concurrent-user scenarios[25]. Our findings align with usability response-time principles suggesting that shorter response times help maintain user attention and interaction continuity by Nielsen[26]. This performance stability ensures that the system can handle peak transactional periods in an SME environment without operational delays.

3.5.2 Analysis of Technical Limitations and Test Failures

While the system achieved a 95% functional suitability rate, this research acknowledges two failed test cases regarding "Bulk Data Import" and "Invoice PDF Export". these failures reveal that the current architectural setup has limitations in handling large-scale data processing on shared hosting environments with restricted memory. Unlike prior ISO/IEC 25010-based functional evaluation studies by H. Panduwiyasa, M. Saputra, Z. F. Azzahra, and A. R. Aniko[27], this research explicitly documents failed test cases as technical debt for future optimization. The "Invoice PDF Export" failure, specifically, identifies a gap in cross-browser font rendering libraries. Although these failures do not obstruct the core transactional flow, they imply that for larger SME operations, a dedicated server resource and library optimization (such as using asynchronous processing for data imports) are necessary requirements.

3.5.3 Practical Implications for SMEs

The implementation of this system offers several practical implications for SME owners and staff:

- Operational Transparency: The automated stock update and receivable tracking significantly reduce manual reconciliation errors, which are common in paper-based systems.
- Scalability: The 0% failure rate under a load of 50 concurrent users suggests that the system can support business growth from a single-user operation to a medium-sized team without requiring immediate hardware upgrades.
- Ease of Adoption: The "Excellent" usability rating implies lower training costs and a shorter learning curve for staff with limited technical backgrounds, making it a cost-effective digital transformation tool for SMEs.

3.5.4 Research Position and Contributions

By evaluating all eight characteristics of ISO/IEC 25010, this study fills a methodological gap identified in previous SME-centric research that often focused solely on usability or functionality. This holistic evaluation provides a benchmark for developing high-quality, lightweight enterprise resource planning (ERP) modules that are specifically optimized for the resource constraints of small-scale industries.

4. CONCLUSION

This study demonstrates that the integration of the Agile Scrum methodology with a comprehensive evaluation based on all eight ISO/IEC 25010 characteristics results in a robust and measurable information system for SMEs. The iterative nature of Scrum allowed for continuous refinement, ensuring that the final product is not only functionally suitable (95% conformity) but also highly usable, secure, and maintainable.

The primary contribution of this research is the establishment of a replicable evaluation model for SME-scale software development. By quantifying often-overlooked non-functional requirements such as reliability and portability, this study provides a quality benchmark that goes beyond basic functional testing. Although the system shows high feasibility under controlled testing environments, its transition to full-scale operational production should be conducted incrementally to account for unpredictable real-world server variables.

The practical implication for software developers in the SME sector is that a quality-oriented approach, rather than a feature-centric one, significantly enhances user acceptance and system longevity. Future work will focus on addressing the identified technical debt in bulk data processing and evaluating the system's performance under longitudinal real-world usage to further validate its resilience.

REFERENCES

[1] D. Effendi, S. Saepullah, and M. I. Rismaya, "Web-Based Sales Information System Design in Small and Medium Enterprises," *Int. J. Educ. Inf. Technol. Others*, vol. 3, no. 3, pp. 484–489, 2020, doi: 10.5281/zenodo.4308092.

[2] C. Cessy and A. Budi, "Implementasi Transaction Processing System dan Decision Support System dalam Pengambilan Keputusan Berbasis Web: Studi Kasus Toko Lion," *J. Inform. dan Bisnis*, vol. 14, no. 1, pp. 28–43, 2025, doi: 10.46806/jib.v14i1.1414.

[3] International Organization for Standardization, "ISO/IEC 25010:2011 Systems and Software Engineering --- Systems and Software Quality Requirements and Evaluation (SQuaRE) --- System and Software Quality Models," 2011, *Geneva, Switzerland*. [Online]. Available: https://www.iso.org/standard/35733.html

[4] R. Putrianasari, E. K. Budiardjo, K. Mahatma, and T. Raharjo, "Problems in the Adoption of Agile-Scrum Software Development Process in Small Organization: A Systematic Literature Review," *Sink. J. dan Penelit. Tek. Inform.*, vol. 9, no. 1, 2024, doi: 10.33395/sinkron.v9i1.13271.

[5] T. Dybå and T. Dingsøyr, "Empirical Studies of Agile Software Development: A Systematic Review," *Inf. Softw. Technol.*, vol. 50, no. 9--10, pp. 833–859, 2008, doi: 10.1016/j.infsof.2008.01.006.

[6] A. B. M. Moniruzzaman and S. A. Hossain, "Comparative Study on Agile Software Development Methodologies," 2013. [Online]. Available: https://arxiv.org/abs/1307.3356

[7] A. T. Priandika and D. Riswanda, "Perancangan Sistem Informasi Manajemen Pemesanan Barang Berbasis Online Menggunakan Pendekatan Extreme Programming," *J. Ilm. Comput. Sci.*, vol. 1, no. 2, pp. 69–76, 2023, doi: 10.58602/jics.v1i2.8.

[8] A. Nfh, S. S. Dewi, and D. R. A. S, "Evaluating EduTrack System Quality Using ISO / IEC 25010 and the DeLone and McLean Model," vol. 6, no. 3, pp. 427–440, 2025, doi: 10.59562/jessi.v6i3.10266.INTRODUCTION.

[9] S. Budi, W. Gata, M. Noor, S. Pangabean, and C. S. Rahayu, "News Portal Website Measurement Analysis Using ISO/IEC 25010 and McCall Methods," vol. 4, no. 1, pp. 273–285, 2022.

[10] Nursahiba and D. Enda, "Pengukuran Kualitas Perangkat Lunak Dengan Standar ISO/IEC 25010 Pada Website Jurusan Teknik Informatika Politeknik Negeri Bengkalis," *JEKIN J. Tek. Inform.*, vol. 5, no. 1, pp. 13–23, 2025, doi: 10.58794/jekin.v5i1.839.

[11] D. P. Widiastuti, N. C. Wibowo, and E. D. Wahyuni, "Software Quality Analysis Using the ISO 25010 Standard on the Digital Information System of the Blitar City Government," *JASIEK (Jurnal Apl. Sains, Informasi, Elektron. dan Komputer)*, vol. 7, no. 2, 2025, doi: 10.26905/jasiek.v7i2.15884.

[12] D. F. Zahra, "Evaluasi Kualitas Aplikasi Keuangan UMKM Menggunakan ISO/IEC 25010," *Remik Ris. dan E-Jurnal Manaj. Inform. Komput.*, vol. 9, no. 3, 2025, doi: 10.33395/remik.v9i3.15072.

[13] Sarwosri, S. Rochimah, U. L. Yuhana, and S. B. Hidayat, "Software Quality Measurement for Functional Suitability, Performance Efficiency, and Reliability Characteristics Using Analytical Hierarchy Process," *JOIV Int. J. Informatics Vis.*, vol. 7, no. 4, pp. 2421–2426, 2023, doi: 10.62527/joiv.7.4.2441.

[14] I. Nambo, N. Pakaya, L. N. Amali, and S. Olii, "Evaluasi Kualitas Sistem Informasi Akademik Terpadu (SIAT) Menggunakan ISO/IEC 25010," *Jambura J. Informatics*, vol. 7, no. 2, pp. 143–153, 2025, doi: 10.37905/jji.v1i2.33720.

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

[15] M. D. Kadenic, D. A. de J. Pacheco, K. Koumaditis, G. Tjørnehøj, and T. Tambo, "Mastering Scrum with a Focus on Team Maturity and Key Components of Scrum," *Inf. Softw. Technol.*, vol. 153, p. 107079, 2023, doi: 10.1016/j.infsof.2022.107079.

[16] A. P. Haryadi and C. Juliane, "Implementation of Sprint Life Cycle from Agile Methodology with Knowledge Management Cycle," *J. Tek. Inform.*, vol. 3, no. 5, 2022, doi: 10.20884/1.jutif.2022.3.5.288.

[17] P. Marnada, T. Raharjo, B. Hardian, and A. Prasetyo, "Agile Project Management Challenge in Handling Scope and Change: A Systematic Literature Review," *Procedia Comput. Sci.*, vol. 197, pp. 290–300, 2022, doi: 10.1016/j.procs.2021.12.143.

[18] E. Susanti and T. E. Tarigan, "Penilaian Kualitas Sistem Informasi Menggunakan ISO/IEC 25010 Dengan Metode Profile Matching," *Jutisi J. Ilm. Tek. Inform. dan Sist. Inf.*, vol. 12, no. 1, pp. 307–315, 2023, doi: 10.35889/jutisi.v12i1.1189.

[19] J. Brooke, "SUS: A Quick and Dirty Usability Scale," in *Usability Evaluation in Industry*, P. W. Jordan, B. Thomas, B. Weerdmeester, and I. L. McClelland, Eds., London, U.K.: Taylor \& Francis, 1996, pp. 189–194.

[20] N. Asnawi, M. Lenawati, R. Pamungkas, and D. Setiawan, "Usability Analysis of A Package Tracking System Using The System Usability Scale Method," *Brill. Res. Artif. Intell.*, vol. 4, no. 2, pp. 858–863, 2024, doi: 10.47709/brilliance.v4i2.5170.

[21] A. S. Puspaningrum, S. Rochimah, and R. J. Akbar, "Functional Suitability Measurement using Goal-Oriented Approach based on ISO/IEC 25010 for Academics Information System," *J. Inf. Syst. Eng. Bus. Intell.*, vol. 3, no. 2, pp. 68–74, 2017, doi: 10.20473/jisebi.3.2.68-74.

[22] R. Yani, I. Nazhifah, and M. I. Pradika, "System Usability Scale in Information System Application Development Using Systematic Mapping Study," *IJATIS Indones. J. Appl. Technol. Innov. Sci.*, vol. 2, no. 2, 2025, doi: 10.57152/ijatis.v2i2.2275.

[23] M. Hyzy *et al.*, "System Usability Scale Benchmarking for Digital Health Apps: Meta-analysis," *JMIR mHealth uHealth*, vol. 10, no. 8, p. e37290, 2022, doi: 10.2196/37290.

[24] S. Octaviani, E. Triandini, and D. P. Hostiadi, "Evaluating Lecturer Satisfaction on Academic Information System Using Usability and EUCS at Bandung University of Technology," *Matrik J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 25, no. 1, pp. 205–216, 2025, doi: 10.30812/matrik.v25i1.4844.

[25] A. Annisa, M. Ardiana, P. N. Rahayu, T. Brian, and M. A. Jami'in, "Performance Evaluation of Web Applications Using JMeter Load Testing for Server Capacity and Response Efficiency," *J. Appl. Informatics Comput.*, vol. 10, no. 1, pp. 585–590, 2026, doi: 10.30871/jaic.v10i1.11840.

[26] J. Nielsen, "Response Times: The 3 Important Limits," 2014. [Online]. Available: https://www.nngroup.com/articles/response-times-3-important-limits/

[27] H. Panduwiyasa, M. Saputra, Z. F. Azzahra, and A. R. Aniko, "Accounting and Smart System: Functional Evaluation of ISO/IEC 25010:2011 Quality Model (a Case Study)," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1092, no. 1, p. 12065, 2021, doi: 10.1088/1757-899X/1092/1/012065.