Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 2, April 2026
DOI : https://doi.org/10.33395/sinkron.v10i2.15947
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Developing an Integrated Capital Assistance and Community Training System Using Agile Scrum

Ahmad Muzaki Zuhdi[1)], Erba Lutfina[2)], Galuh Wilujeng Saraswati[3)]
[1,2,3)]Universitas Dian Nuswantoro, Indonesia
[1)]ahmadmuzaki360@gmail.com, [2)]erba.lutfina@dsn.dinus.ac.id, [3)]galuhwilujengs@dsn.dinus.ac.id

Submitted :Mar 2, 2026 | **Accepted** : Mar 13, 2026 | **Published** : April 2, 2026

Abstract: Local governments increasingly require Cross-Agency Integration platforms to deliver transparent, auditable public services, yet capital assistance and community training programs are often managed through fragmented applications and manual workflows, leading to duplicated data, slow verification, and limited status traceability. This study develops an integrated capital assistance and community training system for local government using Agile Scrum, and evaluates its functional acceptance, usability, and security readiness to support Public Service Digitalization. Requirements were elicited through observation and interviews across three service-managing municipal agencies, while system governance and evaluation also involved the Communication and Informatics Office. The system was implemented as a web application with iterative sprints and backlog prioritization. Evaluation employed a User Acceptance Test (Likert 1–5, 10 items), System Usability Scale, and penetration testing using OWASP ZAP focusing on session management and HTTP security headers. Fifteen agency users participated in the evaluation. The system achieved 93% functional acceptance and a System Usability Scale score of 82.3, indicating excellent perceived usability. Security scanning found no high-risk issues, while medium- and low-risk findings were dominated by missing headers (Content Security Policy and X-Frame-Options) and incomplete cookie flags, which can be mitigated through standard hardening. The proposed platform improves cross-agency coordination and citizen-facing transparency while meeting usability expectations. Agile Scrum enabled rapid alignment with stakeholders and incremental quality improvements. Future work includes analytics, financial-system integration, and continuous security monitoring.

Keywords: Agile Scrum, Cross-Agency Integration, Local Government, OWASP ZAP, Public Service Digitalization, System Usability Scale, User Acceptance Test

INTRODUCTION

Electronic-based government initiatives increasingly emphasize integrated, citizen-centric services that remain traceable across agencies. In Indonesia, the Electronic-Based Government System (SPBE) mandates stronger governance and public-service delivery through digital systems (Presidential Regulation No. 132/2022). In practice, however, many municipal programs still operate with fragmented applications and manual workflows. Capital assistance and community training services often require applicants to resubmit similar identity data and supporting documents to different offices. Verification and scheduling are frequently coordinated through physical files, informal messaging channels, or spreadsheets. Such fragmentation amplifies administrative workload, slows verification cycles, and weakens transparency because applicants cannot reliably monitor service status through a single, consistent channel.

This operational reality aligns with recent interoperability findings in the context of One Data and SPBE, where cross-agency data exchange remains constrained by inconsistent metadata, uneven governance readiness, and limited institutional alignment at the local level (Ramadhan et al., 2025). Evidence from SPBE-based service integration in licensing also suggests that efficiency and transparency improve when agencies share real-time data through an integrated platform (Sumawijaya & Sartika, 2025). Nevertheless, municipal integration initiatives continue to face practical constraints, including evolving rules, fluctuating service volumes, and multi-stakeholder approval chains that complicate standardization and sustained adoption.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 2, April 2026
DOI : https://doi.org/10.33395/sinkron.v10i2.15947
e-ISSN : 2541-2019
p-ISSN : 2541-044X

From a software engineering perspective, local-government systems must accommodate policy changes and operational adjustments without compromising service continuity. Agile methods have therefore gained attention in the public sector as a development approach that supports short iterations, continuous stakeholder feedback, and adaptive prioritization. Agile government has been conceptualized as the capacity to respond to changing public needs, drawing on agile software development principles (Mergel et al., 2021). Empirical accounts further indicate that agile practices can be embedded in public-sector IT management to improve collaboration and responsiveness (Ylinen, 2021), although meaningful adaptation is required to align iterative delivery with bureaucratic procedures and accountability mechanisms (Wijaya et al., 2024).

Beyond implementation, municipal systems are expected to demonstrate production readiness through multi-dimensional quality evidence. Functional fit remains essential, but usability and security increasingly determine whether a platform can be adopted at scale. The System Usability Scale is widely used as a lightweight benchmark for perceived usability (Brooke, 1996) and has been applied in Indonesian information-system evaluations (Mahendra & Asmarajaya, 2022). In parallel, web security guidance highlights recurring risks in session handling and HTTP security configurations, applied studies recommend systematic verification using tools such as OWASP ZAP and report header misconfigurations and session-related issues as common findings (Umar et al., 2024; Putri et al., 2025).

The research gap arises because prior work typically examines public-service integration or agile development practices in isolation, while evaluation is often limited to functional testing without combining usability and security evidence within a single end-to-end municipal case. Addressing this gap, this study develops a unified platform that integrates submission, verification, monitoring, and cross-agency reporting for municipal capital assistance and community training programs. The resulting system is evaluated using a multi-dimensional testing package, combining functional acceptance, perceived usability, and security scanning evidence to support practical production deployment.

LITERATURE REVIEW

Research on SPBE and local-government digital transformation consistently highlights interoperability, enterprise architecture, and governance readiness as key determinants of successful integration. Under the One Data and SPBE context, persistent barriers include inconsistent standards, fragmented sectoral applications, and weak alignment in shared data models, which collectively hinder end-to-end integrated services (Ramadhan et al., 2025). Complementing this perspective, internal policy evaluation of SPBE architecture in local government indicates that stronger alignment between architecture standards and implementation practices is necessary to reduce duplication and strengthen accountability mechanisms across agencies (Fitri et al., 2025). Evidence from licensing-sector integration further suggests that real-time cross-agency data exchange can improve service efficiency and transparency by enabling faster verification and clearer process traceability (Sumawijaya & Sartika, 2025). Taken together, these studies position integration not merely as a technical challenge, but as a governance and operational-readiness problem that directly shapes service quality outcomes.

From a software development standpoint, agile methods are increasingly adopted to support rapid digital transformation in local-government environments characterized by evolving requirements and multi-stakeholder decision making. Recent work reports that agile approaches can help government teams deliver incremental value through iterative planning and continuous feedback loops (Andini et al., 2025). In implementation-oriented studies, Scrum applied to Laravel-based web information systems demonstrates practical benefits in backlog prioritization, sprint-based delivery, and adaptability when requirements change during development (Sumaryanto et al., 2024; Wandri et al., 2025). This line of evidence supports the suitability of Scrum for municipal systems that must frequently adjust to policy refinements and operational constraints without sacrificing delivery continuity.

Beyond development and integration, empirical evaluations increasingly emphasize that municipal platforms should be validated through complementary quality dimensions. Functional testing remains essential, and combining black-box testing with perceived-usability measurement using the System Usability Scale (SUS) has been shown to provide a practical assessment package for public-facing applications (Mahendra & Asmarajaya, 2022). In parallel, security-focused studies applying OWASP ZAP commonly report recurring issues related to HTTP security header hardening and cookie attribute configuration, suggesting that systematic scanning can reveal actionable weaknesses even in otherwise stable deployments (Umar et al., 2024; Putri et al., 2025). Collectively, these findings motivate an integrated municipal platform developed through iterative stakeholder engagement and validated using convergent evidence from functional acceptance, usability, and security testing.

Building on the reviewed literature, this study formalizes the proposed platform as a Cross-Agency Service Orchestration Framework for municipal capital assistance and community training services. The framework consists of four aligned layers, a citizen access layer for registration and status tracking, a workflow orchestration layer for verification and routing, an inter-agency governance layer for authority boundaries and shared coordination, and an assurance layer combining functional acceptance, usability, and security evidence. The contribution of this framework lies in showing that municipal service integration should be understood not only as

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 2, April 2026
DOI : https://doi.org/10.33395/sinkron.v10i2.15947
e-ISSN : 2541-2019
p-ISSN : 2541-044X

technical system integration, but also as coordinated workflow governance supported by iterative Agile Scrum delivery.

METHOD

The research methodology explains the systematic stages used to achieve the research objectives. This study falls within the category of software engineering research, which focuses on the design, development, and evaluation of technology-based information systems in the context of local government.

The research approach uses the Agile Scrum method, a framework within the Agile Development paradigm. This method was chosen because it is iterative, adaptive to changing requirements, and actively involves users in each development cycle. With Scrum, the system can be developed incrementally through short cycles (sprints) that produce continuous functional improvements.

Overview of the Research Flow

In general, the research flow begins with the problem identification and requirements analysis stage, followed by system design and development using the Agile Scrum method. Each sprint produces a version of the system that can be tested and improved based on user feedback. After the system reaches a stable condition, comprehensive testing is conducted using UAT, SUS, and Penetration Testing to ensure the system's functionality, ease of use, and security. The overall research flow can be illustrated as follows:

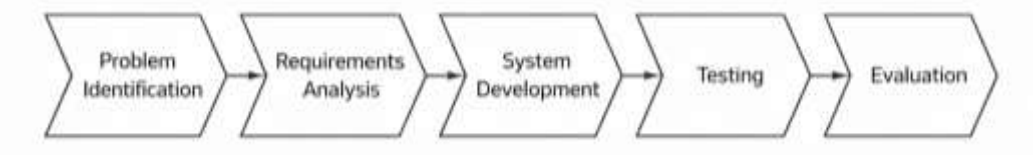

Fig. 1 Research Flow

Based on Figure 1, it can be explained that the research flow starts from the stages of problem identification, requirements analysis, system development, testing, and evaluation. The contribution of this study lies not only in applying multi-dimensional evaluation, but also in formalizing a cross-agency municipal service orchestration model implemented through iterative Agile Scrum development.

Requirements Analysis

The data used in this study consist of primary and secondary data. Primary data were obtained through direct observation and interviews with parties involved in the delivery of public services, particularly within local government organizations (OPD) that manage capital assistance and training programs. Observations were conducted to understand the existing service workflow, while interviews were carried out to identify the main problems and user requirements for the system to be developed.

Secondary data were obtained from policy documents, standard operating procedures (SOP), and related literature on the development of the Electronic-Based Government System (SPBE). In addition, technical data such as data structures, business process diagrams, and system requirements specifications were also used as references in preparing the product backlog. All collected data were then analyzed to ensure that the developed system is truly able to address the needs and problems encountered in the field.

System Development Method

The system development method used is Agile Scrum. Scrum is a software development framework that emphasizes team collaboration, flexibility toward change, and the delivery of valuable products in a short time. In general, the main stages in the Scrum method include Product Backlog creation, Sprint Planning, Sprint Execution, Daily Scrum, Sprint Review, and Sprint Retrospective.

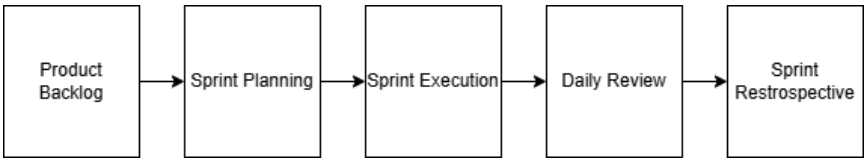

Fig. 2 Agile Scrum Process

The first stage is Product Backlog Creation, which is the process of identifying the functional and non-functional requirements of the system and translating them into a prioritized list of features. This list is prepared based on the results of user requirements analysis and the applicable public service policies. The next stage is

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 2, April 2026
DOI : https://doi.org/10.33395/sinkron.v10i2.15947
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Sprint Planning, where the development team determines which features will be worked on in one sprint cycle, including the estimated time and resources required.

Next is Sprint Execution, which is the implementation stage where developers begin writing code, designing the interface, and integrating system components. During its execution, the team conducts regular coordination through Daily Scrum Meetings to monitor progress, identify obstacles, and align work among team members. After the sprint is completed, a Sprint Review is conducted, in which the development results are presented to stakeholders to obtain direct feedback. The final stage is Sprint Retrospective, which serves to evaluate the team's work process during the sprint, identify constraints, and establish improvements for the next cycle.

Through these stages, system development can be carried out adaptively and continuously evolve toward a stable, efficient system that meets user needs. The implementation of this method also supports the principles of transparency and collaboration in the development of digital government systems.

System Testing Techniques

To ensure that the developed system meets expected quality standards, three types of evaluation were conducted: User Acceptance Test (UAT), System Usability Scale (SUS), and penetration testing. UAT and SUS were administered to the same group of respondents to maintain consistency between functional acceptance and perceived usability outcomes. The respondents were purposively selected from four municipal agencies involved in the service and governance of the platform: the Communication and Informatics Office (Dinas Komunikasi dan Informatika), the Trade and Industry Office (Dinas Perdagangan dan Perindustrian), the Cooperatives, Micro Enterprises, and Manpower Office (Dinas Koperasi Usaha Mikro dan Tenaga Kerja), and the Food Security and Agriculture Office (Dinas Ketahanan Pangan dan Pertanian). A total of 15 users participated, representing the main operational roles that interact with the system, namely verification officers, operators, and coordinators. The distribution was arranged to keep cross-agency representation balanced while reflecting operational availability, 4 respondents from Communication and Informatics Office, 4 from Trade and Industry, 4 from Cooperatives and Micro Enterprises and Manpower, and 3 from Food Security and Agriculture.

Given the study's objective, this evaluation was positioned as an exploratory pilot implementation study intended to assess practical feasibility in the main operational environment rather than to produce statistical generalization. Therefore, a purposive sample of 15 respondents was considered adequate because it covered the principal operational roles and all agencies directly involved in cross-agency verification, coordination, and monitoring.

The UAT questionnaire used a 1–5 Likert scale consisting of 10 statements to assess whether the system functions and workflows match user needs, with acceptance defined as a score of at least 80% of the maximum. The SUS instrument used 10 standard items rated on a 1–5 scale and was calculated using Brooke's method to obtain a score from 0–100. A SUS score above 68 indicates usability above the general benchmark.

Penetration testing was performed using OWASP ZAP, focusing on session management and HTTP security configuration to identify deployability risks and hardening priorities.

Evaluation

The evaluation stage was conducted to assess the extent to which the system meets the research objectives and user needs. The evaluation was performed by comparing the testing results (UAT, SUS, and PenTest) against the functional targets that had been defined in the product backlog.

In addition, an analysis of the effectiveness of the Agile Scrum method was also carried out by reviewing the efficiency of development time, the level of team collaboration, and the improvement of user satisfaction in each sprint. The results of this evaluation serve as the basis for recommendations for further system development in the next stage.

RESULT

This section explains the results of system development based on the stages of the Agile Scrum method that have been applied, as well as the results of the tests conducted on the system. The entire development process was carried out iteratively and adaptively to ensure that the system built is able to address user needs within the Kediri City Government environment.

Requirements Analysis

Based on the identification results, a system requirements analysis was conducted to ensure that the developed system is able to resolve the main problems. The analysis was carried out through observation of work processes in each relevant agency and interviews with prospective system users.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 2, April 2026
DOI : https://doi.org/10.33395/sinkron.v10i2.15947
e-ISSN : 2541-2019
p-ISSN : 2541-044X

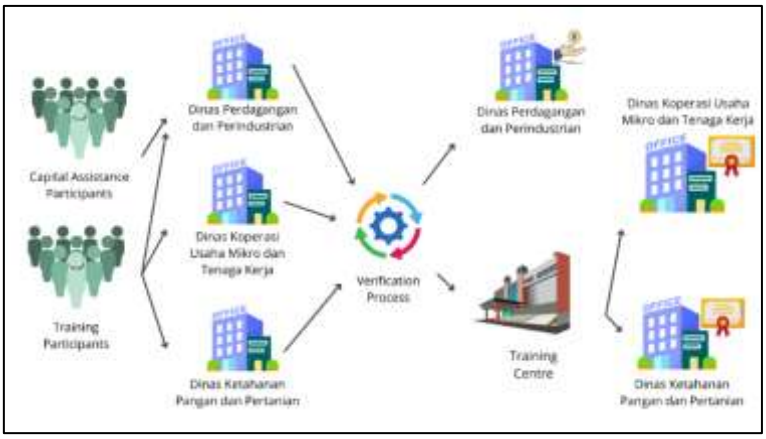

Fig. 3 Current Business Process (As-Is)

Based on observations and interviews with the Trade and Industry Office, the Cooperatives, Micro Enterprises, and Manpower Office, as well as the Food Security and Agriculture Office of Kediri City, it was found that the service process for capital assistance and training is still carried out manually and separately across agencies.

People who wish to participate in assistance or training programs must visit each relevant agency office to obtain registration forms and complete administrative documents in physical form. After the documents are received, officers perform manual verification without an integrated digital system. Participant data are stored in Excel documents or paper archives, so the processes of re-checking, removing duplicate data, and compiling reports require a long time.

In addition, there is no online mechanism for tracking application status, so citizens must ask agency officers directly to find out the selection results. Limitations in the reporting system also cause difficulties in conducting cross-agency evaluation, because training and assistance data cannot be accessed centrally. This has an impact on low efficiency, the potential duplication of assistance recipients, and minimal transparency in program implementation.

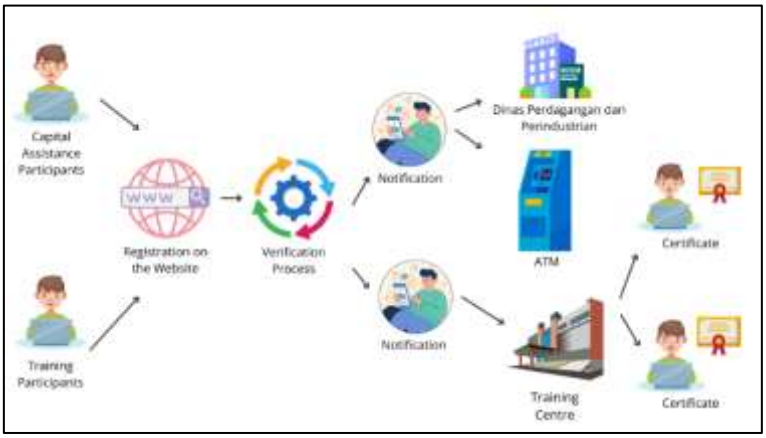

Fig. 4 Proposed Business Process (To-Be)

The Integrated Capital Assistance and Training System is designed to become a digital platform that connects the community, OPD administrators, and system managers online through a single centralized portal. Through the proposed system, citizens can register independently through a web page without having to come to the agency office. The process of uploading documents and filling in participant data is carried out digitally and is directly stored in a centralized database. Each application will be verified online by the administrator of the relevant agency through a verification panel connected to the system.

The verification and data validation process is conducted across OPDs using a role-based access control mechanism so that each agency can only access data according to its authority. After the verification process is completed, the system automatically provides application status notifications to citizens. In addition, the system provides reporting features and a monitoring dashboard that can be accessed by government officials to view assistance recipient data and training participant data in real time. With this model, inter-agency coordination

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

becomes more efficient, the risk of data duplication can be minimized, and citizens obtain transparency and convenience in accessing public services.

Table 1 Functional Requirements

No	Requirement	Description
1.	Online registration for capital assistance and training programs	Citizens can register through the system without coming to the office
2.	Data and document verification by cross-agency administrators	Each OPD admin can verify participant documents digitally
3.	Participant and training management based on program categories	The system groups participants according to the type of program
4.	Integrated reporting and monitoring of assistance recipient data	Displays real-time inter-agency reports
5.	Application status notifications to citizens	The system informs application status changes through the dashboard

Based on Table 1, functional requirements include the main features that must be available so that the system can carry out its functions optimally. These features include online registration for capital assistance and training programs, data verification by cross-agency administrators, participant management based on program categories, integrated data reporting, and application status notifications to citizens. Each of these functions is designed to improve administrative efficiency and minimize manual processes across agencies.

Table 2 Non-Functional Requirements

No	Requirement	Description
1.	Responsive web-based access	The system is developed using Laravel and MySQL with a responsive design
2.	Multi-role user authentication	Supports login for OPD admins, verifiers, and participants
3.	System security	Equipped with data encryption mechanisms and input validation
4.	Simple and easy-to-use interface	The UI design is user-friendly and follows usability principles

Based on Table 2, non-functional requirements cover technical and quality aspects of the system that ensure the application runs stably and is easy to use. These requirements include a responsive web interface, multi-role authentication so that each user has different access rights, data security using encrypted protocols, and a simple and intuitive interface.

Implementation of Scrum Stages

System development was carried out iteratively using the six main Scrum stages (1) Product Backlog, (2) Sprint Planning, (3) Sprint Execution, (4) Daily Scrum, (5) Sprint Review, and (6) Sprint Retrospective.

Product Backlog

The Product Backlog contains 12 main items derived from the requirements analysis. Examples include user registration, document verification, program management, reporting, and the admin dashboard. The backlog was arranged by prioritizing the features most needed by the OPD.

Table 3 Product Backlog

No	Backlog Item	Description	Priority	Estimate
1.	User Account Registration	Citizens can create an account, complete personal data, and activate it to access the system.	High	5 Days
2.	Authentication and Role Management	The system supports multi-role login (OPD Admin, Main Admin, Citizens) with token-based security.	High	5 Days
3.			High	5 Days

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 2, April 2026
DOI : https://doi.org/10.33395/sinkron.v10i2.15947
e-ISSN : 2541-2019
p-ISSN : 2541-044X

No	Backlog Item	Description	Priority	Estimate
4.	Program Application Form	Users can fill in an online application form for capital assistance or training, including document uploads.	High	5 Days
5.	Document and Data Verification	OPD admins verify document completeness and eligibility status of assistance recipients.	High	5 Days
6.	Participant and Program Data Management	Admins can manage participant lists, training types, and each participant's participation status.	Medium	5 Days
7.	Cross-Agency Reporting Dashboard	Displays registration, training, and capital assistance summaries in the form of graphs and dynamic tables.	Medium	5 Days
8.	Application Status Notifications	The system sends automatic notifications to users regarding verification results and application status.	Medium	5 Days
9.	Participant Search by NIK	Admins can search participant data by NIK to prevent duplicate assistance recipients.	Medium	5 Days
10.	Inter-OPD Data Integration	Provides an internal API for automatic synchronization of assistance and training data across agencies.	Low	5 Days
11.	Audit Trail and User Activity	The system records all user activities (login, data edits, verification, reports) as a form of accountability.	High	5 Days
12.	Application Security and Data Encryption	Implements security based on HTTPS, CSRF Protection, and password hashing using the bcrypt algorithm.	Medium	5 Days
	Automated Reporting (Data Export)	The main admin can export cross-OPD activity reports in Excel and PDF formats.		

Based on Table 3, all backlog items above were compiled from the system requirements analysis and discussions with OPD representatives. Five main needs (online registration, verification, management, reporting, and notifications) were then broken down into twelve more measurable and operational backlog items. High-priority items focus on core features directly used by citizens and OPD admins (No. 1–5 and 11). Medium-priority items are supporting features for integration and data transparency (No. 6–9 and 12). Low-priority items include additional functions such as audit logs (No. 10) which are important for accountability but not critical to the main service. This backlog became the main reference in Sprint Planning, where high-priority features were completed earlier in Sprint 1 and 2, while additional features were developed in Sprint 3 after the system was stable.

Sprint Planning

The Sprint Planning stage was conducted to determine which Product Backlog items would be worked on in each sprint. Sprint planning considered feature priority, work complexity, and the need for the system to be tested incrementally. Each sprint lasted two weeks and produced a system increment that could be used for early trials by users. Sprint allocation was designed based on the principle of "core functions first," so that high-priority features were developed in Sprint 1 and Sprint 2, while supporting features and refinements were placed in Sprint 3.

Sprint 1 Development of Basic System Features (Core System Initialization). The focus was on building the system foundation so that users could register, log in, and submit applications.

Table 4. Sprint 1

No	Backlog Item	Reason for Selection	Sprint Output
1.	User Account Registration	Main feature to open system access for citizens	Registration form active and stored in the database
2.	Authentication and Role Management	Needed for user access division (citizens & OPD admin)	Login & role-based access function properly
3.	Program Application Form	Assistance/training submission must be available from the start	Application form and document upload function properly
4.	Basic Application Security	Basic security must be applied from the start (HTTPS, CSRF)	System is secure for initial testing

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 2, April 2026
DOI : https://doi.org/10.33395/sinkron.v10i2.15947
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Sprint 2 Development of Service and OPD Management Features. The focus was on providing main functions used by OPD admins to verify and manage registrant data.

Table 5. Sprint 2

No	Backlog Item	Reason for Selection	Sprint Output
1.	Document and Data Verification	Core process for OPD to determine participant eligibility	Cross-OPD verification module functions properly
2.	Participant and Program Data Management	Needed for internal management	OPD administration pages ready to use
3.	Cross-Agency Reporting Dashboard	Enables OPD to monitor real-time progress	Initial dashboard and basic data recap available
4.	Participant Search by NIK	Prevents duplicate assistance recipients	Search and filter features function properly

Sprint 3 Cross-Agency Integration, Notifications, and Final Reporting. The focus was on feature refinement, inter-department data integration, advanced security, and structured reporting.

Table 6 Sprint 3

No	Backlog Item	Reason for Selection	Sprint Output
1.	Application Status Notifications	Improves transparency and user experience	System sends automatic email/in-app notifications
2.	Inter-OPD Data Integration	Ensures real-time data synchronization among 3 departments	Internal API & synchronization function properly
3.	Audit Trail and User Activity	Required for government accountability	User activity logs are active
4.	Automated Reporting (Data Export)	Needed for cross-OPD activity reporting	PDF/Excel export functions properly

Sprint Execution and Daily Scrum

Sprint Execution is the implementation process of the previously planned work. In practice, many priority changes, feature adjustments, and fixes occurred based on trial results. During development, each sprint produced an increment that was directly tested by OPD within the Kediri City Government environment.

Table 7 Sprint Execution

Sprint	Results	Constraints	Output
Sprint 1	Registration form works; PDF file validation; Login	Upload size error	Registration module completed; Authentication stable
Sprint 2	Verification rules for 3 OPDs; Verifier notes; Fast queries	-	Cross-OPD verification module; Fast dashboard
Sprint 3	Email notifications; Audit log; Security enhanced	-	System stable; Ready for UAT/SUS/PenTest

Sprint Review

Sprint Review was conducted at the end of each sprint cycle to assess the feasibility and quality of the product increment that had been developed. This session was attended by representatives from relevant OPDs, namely the Trade and Industry Office, the Cooperatives, Micro Enterprises, and Manpower Office, and the Food Security and Agriculture Office of Kediri City, and also involved community representatives as early users.

In Sprint 1, the review focus was ensuring smooth user authentication and interface consistency. Stakeholders assessed that the system navigation structure was easy to understand, but requested adjustments to the user-role categories to better match the organizational structure of the OPD. This input was then included in the Product Backlog as an improvement item for Sprint 2.

In Sprint 2, the demo focused on the document verification module and the reporting dashboard. From the evaluation session, a data-loading delay was found in verification data due to unoptimized queries. After improvements, data loading performance increased and was considered more responsive by verification officers. Stakeholders also proposed a document completeness indicator feature to facilitate the selection process.

*name of corresponding author

In Sprint 3, the system ran more stably with inter-OPD modules already connected. Stakeholders assessed that inter-department integration met service needs, including application status tracking by citizens. In the review, users provided feedback to add automatic status-based notifications, which then became an additional backlog item for the next iteration.

Overall, Sprint Review results showed improved system quality in each iteration. Stakeholder input during reviews proved to accelerate feature refinement and increase system alignment with service needs in the field. These findings and improvements also became the basis for evaluation in Sprint Retrospective.

Sprint Retrospective

Sprint Retrospective was conducted at the end of each sprint to evaluate the development process, identify emerging problems, and formulate improvement steps for the next sprint. The evaluation was conducted with the entire development team consisting of a backend developer, frontend developer, and tester.

In Sprint 1, the team noted that development was still hindered by differences in understanding of the business flow across OPDs. This caused several interface components to need revision to match the needs of each department. As a follow-up, the team decided to align understanding through daily clarification sessions with stakeholders in the early phase of Sprint 2.

In Sprint 2, the main constraint was data consistency from verification results. Some incoming data were not immediately synchronized across modules, causing the system to temporarily display inaccurate application status. To address this issue, the team improved the database structure, optimized queries, and added automatic data-check logs. These improvements proved effective in Sprint 3, where all modules could communicate more stably.

In Sprint 3, the team's biggest challenge was completing additional feature requests that emerged during Sprint Review, such as application status notifications and document completeness indicators. The team realized that sudden change requests could affect the work duration if not managed properly. Therefore, it was decided that every new request must go through an impact assessment process before being included in the next sprint backlog.

Overall, Sprint Retrospective showed that the Scrum method enabled the team to adapt quickly to changing requirements while improving system quality incrementally. Regular evaluations also helped the team maintain performance consistency, minimize recurring errors, and ensure that the developed system remained relevant to user and OPD needs within the Kediri City Government environment.

System Testing Results

User Acceptance Test (UAT) Testing

The User Acceptance Test (UAT) was conducted using a 10-item questionnaire on a 1–5 Likert scale to evaluate the alignment between system functions and user needs. The acceptance index was computed as the ratio of the obtained total score to the maximum possible score, expressed as a percentage. With 15 respondents and 10 items, the maximum score is 750 (15 × 10 × 5). The system is considered accepted when the acceptance index is ≥ 80%.

Results show that the aggregated acceptance index exceeded the acceptance threshold, indicating overall functional fit for daily operations. The aggregated acceptance index reached 93%, which corresponds to an average rating of approximately 4.65 out of 5 across the UAT statements. This indicates that respondents generally selected "Agree" to "Strongly Agree," with remaining gaps mainly reflecting minor improvement areas rather than core functional mismatches.

Table 8. UAT Testing Results

No	Tested Aspect	Score	Result
1.	User registration	95	Functions well
2.	Document verification	92	Functions well
3.	Monitoring	94	Functions well
4.	Cross-OPD reporting	91	Functions well

System Usability Scale (SUS) Results

The SUS score was calculated following Brooke's method by converting item scores (odd items: score − 1; even items: 5 − score), summing all converted values, and multiplying by 2.5 to obtain a 0–100 score. The SUS test was administered to the same 15 respondents as UAT to ensure comparable evaluation conditions. The system achieved a SUS score of 82.3, which is substantially above the usability benchmark of 68. This indicates that users perceived the interface and workflow as easy to learn, consistent to navigate, and efficient for completing verification and monitoring tasks. Although the usability perception is strong, respondents' qualitative feedback suggests that performance-related aspects (e.g., page loading speed under certain conditions) remain a practical improvement target.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 2, April 2026
DOI : https://doi.org/10.33395/sinkron.v10i2.15947
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Table 9 SUS Testing Results

No	Assessment Component	Average Score
1.	Ease of use	4.6
2.	Display clarity	4.6
3.	Time efficiency	4.6
4.	User satisfaction	4.6
	SUS Score	82.3

Penetration Testing Results

System security testing was carried out using OWASP Zed Attack Proxy (ZAP) to identify potential vulnerabilities in the web application. The testing focused on server-side security aspects, HTTP header configuration, session management, and potential weaknesses that could be exploited by unauthorized parties. The scan results indicated several findings with different risk levels, as summarized in the ZAP report.

Overall, OWASP ZAP detected two findings with a medium risk level, five findings with a Low risk level, and several Informational findings. These findings are described as follows

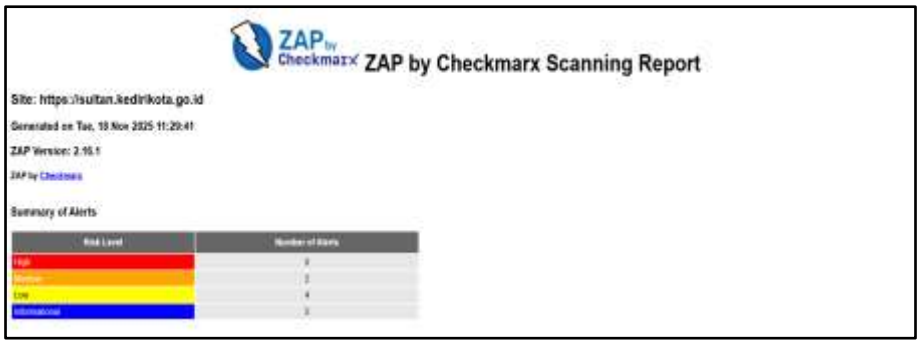

Fig. 5 Summary of OWASP ZAP Scan Results

Table 10. Summary of Security Findings

No	Finding Type	Risk Level	Score	Result
1.	CSP Header Missing	Medium	1	Implement CSP
2.	X-Frame-Options Missing	Medium	1	Anti-clickjacking
3.	Cookie Flag Missing	Low	4	HttpOnly & Secure
4.	Informational	Info	5	Server hardening

The reported ZAP findings represent the pre-deployment baseline. At the time of writing, the identified header and cookie hardening actions had been documented as mandatory wrap-up tasks prior to broader production release. Therefore, the findings are treated as deployment prerequisites rather than ignored residual risks.

Evaluation

The system evaluation covered functionality, usability, and security. UAT results indicated that core features met cross-OPD needs with a 93% success rate, particularly for registration, verification, monitoring, and reporting. SUS scored 82.3 (“Excellent”), suggesting the interface and workflow are easy to understand and comfortable for both citizens and OPD administrators.

Security testing using OWASP ZAP reported no high-risk findings, but improvements are needed mainly for security headers (Content-Security-Policy, X-Frame-Options) and cookie attributes (HttpOnly). These findings indicate that the main remaining risks are hardening-related rather than critical business-logic flaws. Overall, Agile Scrum supported incremental quality improvements and rapid stakeholder feedback handling, indicating the system meets technical and operational needs.

DISCUSSIONS

This study was motivated by three practical and research gaps in municipal public-service digitalization. The first gap is operational fragmentation, assistance and training services are often handled through separated workflows and non-integrated data storage, which creates duplicated submissions, slow verification cycles, and weak traceability for citizens. The second gap is methodological, prior studies frequently discuss service integration or agile development separately, without showing how iterative agile delivery can directly resolve cross-agency alignment issues in an end-to-end municipal case. The third gap is evaluation-related, many

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

implementations report functional testing only, while usability and security readiness two factors that strongly influence adoption and production deployment are not consistently evidenced together.

The developed platform addresses the first gap by consolidating submission, verification, monitoring, and reporting into a single portal with centralized storage and role-based verification. The UAT acceptance index of 93% demonstrates that the implemented workflows matched day-to-day operational needs across agencies. Importantly, the value of 93% is not merely a "high number," but indicates that the average response per acceptance statement was approximately 4.65/5, meaning users largely agreed that core tasks (registration, verification, monitoring, and cross-agency reporting) can be completed as intended. Since the acceptance threshold was set at 80%, the 93% result provides a clear margin that the system meets functional expectations rather than only minimally passing.

From a digital-governance perspective, the findings support the view that municipal service integration is not merely a matter of connecting databases, but also of aligning roles, data interpretation, and operational authority across agencies. This study therefore extends prior e-government integration research by operationalizing interoperability at the service-process level, rather than discussing integration only as a technical or architectural issue. It also complements prior agile public-sector studies by showing that sprint reviews and backlog reprioritization can function as practical mechanisms for reconciling cross-agency workflow differences in a municipal setting.

Compared with prior implementation-oriented studies that primarily emphasize adaptability in software delivery, this study contributes a more integrated municipal case by examining service integration, agile delivery, and deployment-oriented evaluation together. Compared with studies that report only functional testing or only usability outcomes, the inclusion of OWASP ZAP broadens the evidence from user acceptance toward early deployment readiness. In this sense, the study offers both a practical and conceptual contribution to public-service digitalization research.

Regarding the second gap, Agile Scrum contributed by enabling iterative clarification of cross-agency business rules through sprint reviews and continuous reprioritization of backlog items. This was particularly relevant in local-government settings where workflow details and verification rules evolve during development. The sprint-based approach allowed the system to converge toward a stable, agreed process while keeping stakeholders involved in incremental validation. The practical implication is that agility here is not a general claim, but is reflected in the ability to align verification procedures and reporting needs across different agencies without waiting for a single "final" delivery.

For the third gap, the evaluation combines usability and security readiness evidence with functional acceptance. The SUS score of 82.3 is well above the benchmark value of 68, meaning user-perceived usability is above average and strongly supportive for adoption. In practical terms, a score in the low-80s indicates that most users can learn and operate the system with minimal assistance, navigation is perceived as consistent, and the interaction flow is comfortable for routine work. This usability result complements the UAT finding, the system is not only functionally acceptable, but also usable enough to support repeated operational use. Meanwhile, OWASP ZAP results found no high-risk issues, suggesting early deployability, but identified medium and low findings dominated by missing security headers and incomplete cookie flags. This means the system is suitable for municipal use provided that standard hardening (e.g., Content Security Policy, anti-clickjacking header, and cookie attributes) is applied before broader deployment.

From a security-governance perspective, the dominant findings are configuration-oriented rather than logic-oriented vulnerabilities. Missing Content-Security-Policy and X-Frame-Options headers indicate hardening gaps associated with security misconfiguration and clickjacking prevention, while incomplete cookie flags indicate session-protection weaknesses. In relation to the OWASP Top 10 perspective, these issues are most closely associated with security misconfiguration and inadequate protective controls at the deployment layer. Therefore, the security results should be interpreted as early deployment-readiness evidence, the application showed no high-risk findings, but standard hardening remains necessary before broader production exposure.

Overall, the combined evidence indicates that the study contributes an end-to-end municipal case showing how cross-agency integration can be delivered through Agile Scrum and validated using convergent measures, functional acceptance (UAT), perceived usability (SUS), and security scanning (OWASP ZAP). However, the evaluation is limited by a single-city context and a respondent group focused on agency users, broader citizen-facing trials and performance testing under higher load are recommended to strengthen generalizability and operational scalability.

This study has several limitations. First, the evaluation involved 15 purposively selected users from a single municipal context, so the findings should be interpreted as exploratory evidence of feasibility rather than broad statistical generalization. Second, the respondent group was dominated by agency-side operational users, meaning that broader citizen-facing usability perceptions are not yet fully represented. Third, the archived evaluation evidence reported in this study is maintained in aggregated form, therefore, advanced item-level reliability analysis and correlation testing were not emphasized in the present manuscript and are recommended for future research.

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

CONCLUSION

This study developed and evaluated an integrated capital assistance and community training system for a municipal environment to improve cross-agency traceability, reduce duplicated submissions, and streamline verification and reporting. Beyond system implementation, the study contributes a Cross-Agency Service Orchestration Framework showing how citizen access, verification workflows, governance boundaries, and quality assurance can be integrated within one municipal service platform.

The platform was delivered using Agile Scrum through iterative sprints, enabling continuous alignment of business rules and operational requirements across agencies. The results provide convergent evidence of readiness, functional acceptance reached 93% (exceeding the 80% acceptance threshold), usability achieved a SUS score of 82.3 (well above the benchmark of 68), and OWASP ZAP security scanning reported no high-risk findings. These outcomes indicate that the system is not only functionally fit but also sufficiently usable for routine operations, while requiring targeted hardening focused on HTTP security headers and cookie configurations prior to broader production deployment.

By combining cross-agency integration, agile delivery, and multi-dimensional evaluation in a single municipal case, the study extends prior work that reports only implementation or functional testing. Future work should expand trials to larger and more diverse user groups, add performance and load monitoring, integrate analytics dashboards and municipal financial systems, and implement continuous vulnerability scanning to strengthen operational scalability and long-term security posture.

REFERENCES

Andini, D. Y. A., Rizki, F., & Yulia, A. F. (2025). Agile digital transformation in local government: An extreme programming approach to public service mall applications. *Journal of Information Systems and Informatics, 7*(2), 1264–1282. https://doi.org/10.51519/journalisi.v7i2.1083

Brooke, J. (1996). SUS: A quick and dirty usability scale. In P. W. Jordan, B. Thomas, I. L. McClelland, & B. Weerdmeester (Eds.), *Usability evaluation in industry* (pp. 189–194). Taylor & Francis. https://doi.org/10.1201/9781498710411-35

Fitri, U., Dahlan, D., Ihsan, Z., Amalia, R. S., & Yulia, R. (2025). Evaluation of internal policy of electronic-based government system (SPBE) in Pariaman City. *Journal of Election and Leadership (JOELS), 6*(2), 88–100. https://journal.unilak.ac.id/index.php/joels/article/view/23698

Mahendra, G. S., & Asmarajaya, I. K. A. (2022). Evaluation using black box testing and system usability scale in the Kidung Sekar Madya application. *Sinkron: Jurnal dan Penelitian Teknik Informatika, 6*(4), 2292–2302. https://doi.org/10.33395/sinkron.v7i4.11755

Mergel, I., Ganapati, S., & Whitford, A. B. (2021). Agile: A new way of governing. *Public Administration Review, 81*(1), 161–165. https://doi.org/10.1111/puar.13202

OWASP Foundation. (2021). *OWASP Top 10: 2021*. https://owasp.org/Top10/

Putri, V. R., Sobandi, A., & Santoso, B. (2025). Analysis of information system security using OWASP ZAP on a web-based electronic archiving system. *Telematika: Jurnal Telematika dan Teknologi Informasi, 22*(3), 28–42. https://doi.org/10.31315/telematika.v22i3.14241

Ramadhan, A., Suhendra, A., & Yohanitas, W. A. (2024). One data Indonesia: A retrospective analysis of data interoperability in declaring regional planning and development. In *The 5th International Conference on Governance, Public Administration, and Social Science (ICoGPASS 2024)*. *KnE Social Sciences, 10*(16), 152–171. https://doi.org/10.18502/kss.v10i16.19169

Republic of Indonesia. (2022). *Peraturan Presiden Republik Indonesia Nomor 132 Tahun 2022 tentang Arsitektur Sistem Pemerintahan Berbasis Elektronik*. https://peraturan.bpk.go.id/

Sumaryanto, M. A., & Sinnun, A. (2024). Implementation of Scrum and Laravel in the new student admission information system at STMIK Mercusuar. *Jurnal Multidisiplin Sahombu, 4*(1), 300–317. https://doi.org/10.58471/jms.v4i01

Sumawijaya, H., & Sartika, I. (2025). Integration of SPBE-based public services in the licensing sector of the Perkimta Office of South Tangerang City. *Jurnal Studi Ilmu Sosial dan Politik, 5*(2), 181–194. https://doi.org/10.35912/jasispol.v5i2.5376

Umar, R., Riadi, I., & Wicaksono, S. A. (2024). Application of OWASP ZAP framework for security analysis of LMS using pentest method. *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer), 10*(2), 224–230. https://doi.org/10.33480/jitk.v10i2.5534

Wandri, R., Fadhilah, M. R., Setiawan, P. R., & Fadhilla, M. (2024). Agile Scrum as a development approach: A case study of web-based school information system design. *Sistemasi: Jurnal Sistem Informasi, 14*(4), 1722–1735. https://doi.org/10.32520/stmsi.v14i4.5273

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

Wijaya, R., Kumorotomo, W., Ratminto, & Djunaedi, A. (2024). Government organization adaptation to implement agile practices: A study of provincial smart city agencies in Indonesia. *Eduvest – Journal of Universal Studies, 4*(9), 8024–8045. https://doi.org/10.59188/eduvest.v4i9.30330

Ylinen, M. (2021). Incorporating agile practices in public sector IT management: A nudge toward adaptive governance. *Information Polity, 26*(3), 251–271. https://doi.org/10.3233/IP-200269

*name of corresponding author