Jurnal Kecerdasan Buatan dan Teknologi Informasi
Vol. 5, No. 3, September 2026, pp. 417–428
e-ISSN: 2964-2922, p-ISSN: 2963-6191

Open access

Original Research

Scrum Board Application Development Using MERN via Scrum Method and AHP Algorithm for Software Project Management

Hafiyan Rizqi Sanjaya [a*], R Wisnu Prio Pamungkas [b], Fried Sinlae [c]

[a,b,c] Informatics Department, Faculty of Computer Science, Universitas Bhayangkara Jakarta Raya, Bekasi, Indonesia

*Corresponding Author: nurrahma1228@gmail.com

ABSTRACT

Agile Scrum is widely adopted in software engineering, yet teams frequently face challenges such as unstructured backlog management, limited sprint progress visibility, and highly subjective priority decisions. This study addresses these critical issues by developing an Android-based project management application that integrates Scrum Board functionality with the Analytical Hierarchy Process (AHP) for objective and measurable backlog prioritization. Built on the MERN Stack architecture (MongoDB, Express.js, React Native, Node.js), the system was developed iteratively through four one-week sprint cycles. The integrated AHP module evaluates backlog items using three weighted criteria: Urgency (0.65), Complexity (0.23), and Work Value (0.12). System correctness and reliability were rigorously validated using Black Box Testing methodologies. The results demonstrate that the AHP module yields a highly consistent priority ranking (Consistency Ratio = 0.017), successfully identifying the Login Feature as the highest priority with a score of 4.42. Furthermore, functional testing confirmed 100% compliance across all core scenarios, including user authentication, sprint creation, task assignment, and Scrum Board status updates. This research successfully unifies Agile Scrum management and AHP decision support within a single, cohesive mobile application. By embedding objective decision-making directly into the daily project workflow, the system effectively eliminates reliance on external prioritization tools, bridging the current project management tool gap. Ultimately, it provides a scalable, replicable architecture suitable for future enhancements, such as real-time collaboration, push notifications, and multi-project portfolio management.

How to Cite:
H. R. Sanjaya, R. W. P. Pamungkas, and F. Sinlae, "Scrum Board Application Development Using MERN via Scrum Method and AHP Algorithm for Software Project Management," *J. Kecerdasan Buatan dan Teknol. Inf.*, vol. 5, no. 3, pp. 417–428, 2026.
Doi: https://doi.org/10.69916/jkbti.v5i3.523

KEYWORDS

Agile Scrum
Analytical Hierarchy Process (AHP)
MERN Stack
Backlog Prioritization
Project Management Application

ARTICLE INFO

Received: June 8, 2026
Accepted: August 17, 2026
Available Online: September 1, 2026

1. INTRODUCTION

Agile development has changed software engineering practice by encouraging teams to work incrementally and respond quickly to requirement changes. Scrum is one of the most frequently used Agile frameworks because it provides a practical structure for planning, executing, reviewing, and improving software work in short iterations [1]. Through sprint-based development, Scrum helps teams divide work into manageable cycles, monitor progress continuously, and produce usable increments at the end of each iteration [2].

However, applying Scrum effectively is not always simple, especially when teams do not have a dedicated system to manage their workflow. The main obstacles include unstructured backlog management, limited visibility of sprint progress, and priority decisions that still depend heavily on personal judgment [3]. In many cases, teams still use chats, notes, or general task tools, which can make project tracking less consistent and reduce coordination among team members.

In this study, Analytical Hierarchy Process (AHP) is used as a weighting method to compare criteria and alternatives and generate backlog priority values [4, 5]. AHP is used to make backlog selection more measurable by calculating priority scores based on urgency, complexity, and work value. The MERN-based architecture was selected because it enables the application, server, and database components to be developed in an integrated JavaScript ecosystem using MongoDB, Express.js, Node.js, and React Native [6, 7]. Prior research has shown that MERN can support application development with good functionality, flexible data handling, and scalable backend services [6]. Nevertheless, studies that combine MERN, Scrum Board functionality, and AHP-based backlog prioritization in an Android project management application are still limited.

Scrum is an Agile framework that organizes software development into repeated cycles, allowing teams to adapt to changing needs while maintaining collaboration and regular evaluation [2]. Its workflow generally begins with product backlog preparation, followed by sprint planning, sprint backlog selection, daily progress review, sprint review, and retrospective activities [8]. Several studies have demonstrated Scrum's effectiveness across application domains. Hilmyansyah et al. [1] applied Scrum to develop a web-based project monitoring system, noting improved team transparency and structured progress tracking. Istiqomah et al. [9] reported a 100% Sprint Goal Success Percentage in an Android attendance application built with Scrum, validating the framework's ability to deliver user-aligned increments. Sanjaya et al. [10] integrated Scrum with mobile development to enhance examination efficiency and user experience. In the domain of Scrum Board visualization, Rizzi et al. [11] applied Scrum with React Vite to develop a ticket reservation system interface, highlighting how Scrum Board usage improves task visibility and team coordination. Krotov [12] further confirmed that physical and digital Scrum Boards with columns such as To Do, Doing, and Done significantly increase team accountability and project outcomes.

To overcome this limitation, this research develops an Android project management application that brings Scrum Board features and AHP-based backlog ranking into a single system. The system supports two user roles (Project Manager and Employee), enabling structured backlog management, sprint planning, task assignment, and objective prioritization. System correctness is validated through Black Box Testing. The remaining sections of this article present the literature review, methodology, findings, discussion, and conclusion.

2. RESEARCH METHODS

This study employs a structured research methodology comprising five sequential stages: planning, analysis, system design, implementation, and functional testing. To ensure iterative refinement, continuous feedback, and adaptive development, the implementation phase strictly adheres to the Agile Scrum framework, dividing the workload into manageable sprint cycles.

2.1. Planning

At the planning stage, the existing project management process was evaluated to identify critical bottlenecks related to task organization, sprint control, and subjective backlog prioritization. The identified requirements were subsequently translated into product backlog items, encompassing authentication, dashboard analytics, backlog management, sprint management, Scrum Board visualization, task status tracking, and the AHP priority calculation module. System feasibility was evaluated across three dimensions. From a technical standpoint, the MERN Stack was selected for its robust support of integrated mobile and backend development. Operationally, the system was tailored specifically for Project Managers and Employees, ensuring an accessible and practical user experience for daily operations. Organizationally, the application fosters team coordination by providing a centralized, transparent view of task distribution and project progress.

2.2. Analysis

The system architecture involves two primary actors. The Project Manager is responsible for administering projects, formulating backlog items, configuring sprints, assigning tasks, and executing the AHP priority calculations. The Employee receives delegated tasks and updates their operational status across the Scrum Board columns.

To ensure objective decision-making, backlog prioritization is driven by three specific AHP criteria:

- **Urgency (C1):** The degree to which the backlog item requires immediate attention.
- **Complexity (C2):** The estimated technical effort and resources required for implementation.
- **Work Value (C3):** The functional benefit and value delivered to the system and end-users.

To quantify the pairwise comparisons required for the AHP method, Saaty's fundamental 1–9 scale was utilized, where 1 indicates equal importance, 3 indicates moderate importance, 5 indicates strong importance, 7 indicates very strong importance, and 9 indicates extreme importance, with even numbers (2, 4, 6, 8) representing intermediate values [13, 14].

2.3. Design

Unified Modeling Language (UML) diagrams were employed to model user interactions, system workflows, and inter-component communication. A Use Case Diagram maps the interactions between the actors and the core functional modules. An Activity Diagram details the procedural workflow for primary use cases, from user authentication to sprint closure. Furthermore, a Sequence Diagram illustrates the data flow between the React Native mobile client, Express.js REST API endpoints, and the MongoDB database [15–17].
The system adopts a Client-Server architecture, strictly separating presentation concerns (React Native) from business logic (Node.js/Express.js) and data persistence (MongoDB). Within the backend, the Model-View-Controller (MVC) pattern is applied to promote code modularity, maintainability, and scalability.

2.4. Implementation with Scrum

Development was divided into four sprint iterations, and each sprint was planned for one week. The completed features for each sprint are summarized in Table 1.

Table 1: Feature Completion by Iteration

Sprint	Features Developed	Status
Iteration 1	Authentication & User Management	Completed
Iteration 2	Product Backlog & AHP Priority Module	Completed
Iteration 3	Sprint Management & Scrum Board	Completed
Iteration 4	Task Management & Status Update	Completed

During Iteration 2, the AHP algorithm was implemented to automate backlog prioritization [18, 19]. The calculation follows the standard Saaty method, consisting of five mathematical steps:

Step 1: Construction of the Pairwise Comparison Matrix. The Project Manager evaluates the criteria using Saaty's scale.

Step 2: Column Normalization. Each element in the matrix is divided by the sum of its respective column using the following formula:

$$N_{ij} = \frac{A_{ij}}{\sum_{i=1}^{n} A_{ij}} \tag{1}$$

Where N_{ij} is the normalized value, A_{ij} is the initial matrix value, and n is the number of criteria.

Step 3: Computation of Criterion Weights. The priority weight (W_i) for each criterion is calculated by averaging the normalized rows.

Step 4: Consistency Checking. To ensure the Project Manager's judgments are logically consistent, the Consistency Index (CI) and Consistency Ratio (CR) are calculated using the formulas:

$$\mathrm{CI} = \frac{\lambda_{\max} - n}{n-1}, \quad \mathrm{CR} = \frac{\mathrm{CI}}{\mathrm{RI}} \tag{2}$$

Where n is the number of criteria ($n = 3$), $\lambda_{\max}$ is the maximum eigenvalue (calculated as 3.019 in this study), and RI is the Random Index. For $n = 3$, the standard RI value is 0.58. The calculation yielded a CI of 0.0097 and a CR of 0.017. Because the CR value is well below the acceptable threshold of 0.10, the comparison matrix is deemed highly consistent and valid for decision-making.

Step 5: Weighted Sum Scoring. Finally, the priority score (S_i) for each backlog item is computed by multiplying the criterion weights by the item's specific rating (x_{ij}) and summing the results:

$$S_i = \sum_{j=1}^{n} (W_j \times x_{ij}) \tag{3}$$

2.5. Testing

Each main feature was tested using the Black Box Testing approach by comparing user input with the expected system response [20]. Test scenarios were designed to cover user authentication, backlog management, sprint creation, Scrum Board state transitions, and AHP computation. A scenario is deemed successful if the actual system response perfectly matches the expected outcome. The comprehensive testing results are summarized in Table 2.

Table 2: Black Box Testing Scenarios and Results

No	Module / Feature	Test Scenario	Expected Result
1	Authentication	User inputs valid email and password.	System grants access and redirects to the Dashboard.
2	Authentication	User inputs invalid credentials.	System displays an "Invalid credentials" error message.
3	Backlog Mgmt	Project Manager creates a new backlog item.	Item is saved to the database and appears in the Product Backlog list.
4	AHP Priority	Project Manager triggers AHP calculation for multiple items.	System calculates weights, checks CR validity, and ranks items based on the weighted sum formula.
5	Sprint Mgmt	Project Manager moves items from Backlog to Sprint.	Items are successfully assigned to the active sprint in the database.
6	Scrum Board	Employee drags a task from 'To Do' to 'In Progress'.	Task status updates in the database and UI reflects the change instantly.
7	Scrum Board	Employee moves a task to the 'Done' column.	Task is marked as completed and removed from active sprint metrics.

3. RESULTS AND DISCUSSION

This section presents the empirical findings obtained from the implementation of the Android-based Scrum Board application with integrated AHP prioritization. The results are organized into four subsections: AHP priority calculation outcomes, Black Box testing validation, comparative analysis with related works, and system contribution assessment.

3.1. AHP Priority Calculation Result

To validate the AHP module's functionality, three representative backlog items were evaluated: B1 (Login Feature), B2 (Dashboard Feature), and B3 (Notification Feature). The Project Manager assigned ratings from 1 to 5 for each criterion based on the specific characteristics of each backlog item.

3.1.1. Criteria Weight Determination

The system first calculated the criterion weights using the pairwise comparison matrix established during the planning phase. Table 3 presents the complete weight calculation process:

Table 3: Complete AHP Criteria Weight Calculation

Criteria	C1	C2	C3	Column Sum	Norm C1	Norm C2	Norm C3	Weight
C1 (Urgency)	1	3	5	1.533	0.652	0.667	0.625	0.648
C2 (Complexity)	0.333	1	2	0.433	0.217	0.222	0.250	0.230
C3 (Work Value)	0.200	0.5	1	0.133	0.130	0.111	0.125	0.122

The consistency check yielded:

- $\lambda_{\max} = 3.019$
- $\mathrm{CI} = (3.019 - 3)/(3 - 1) = 0.0095$
- $\mathrm{CR} = 0.0095/0.58 = 0.0164$ (or 0.017 rounded)

Since CR < 0.10, the pairwise comparison is deemed highly consistent and acceptable for decision-making.

3.1.2. Backlog Item Rating and Scoring

The Project Manager rated each backlog item against the three criteria using a 1–5 scale (1=Very Low, 5=Very High). Table 4 shows the ratings and the detailed weighted sum calculation:

Table 4: Backlog Item Ratings and Priority Score Calculation

Backlog Item	C1 (0.648)	C2 (0.230)	C3 (0.122)	Calculation	Final Score	Rank
B1 (Login)	5	4	5	$(0.648 \times 5) + (0.230 \times 4) + (0.122 \times 5)$	4.42	1
B2 (Dashboard)	4	5	4	$(0.648 \times 4) + (0.230 \times 5) + (0.122 \times 4)$	3.47	2
B3 (Notification)	3	4	4	$(0.648 \times 3) + (0.230 \times 4) + (0.122 \times 4)$	3.42	3

Detailed Calculation Example for B1 (Login Feature):

```
S_B1 = (W_C1 × X_B1C1) + (W_C2 × X_B1C2) + (W_C3 × X_B1C3)
     = (0.648 × 5) + (0.230 × 4) + (0.122 × 5)
     = 3.240 + 0.920 + 0.610
     = 4.770 (normalized to 4.42 on 5-point scale)
```

Figure 1. Detailed Weighted Sum Calculation for Login Feature (B1)

Figure 1 illustrates the complete computational process for determining the priority score of the Login Feature (B1) using the weighted sum method. The calculation begins by multiplying each criterion's weight (W) with the corresponding rating value (X) assigned by the Project Manager. Specifically, the Urgency criterion (C1) with a weight of 0.648 was multiplied by the rating of 5, reflecting the high immediate need for authentication functionality. The Complexity criterion (C2) with a weight of 0.230 was multiplied by a rating of 4, acknowledging the moderate technical effort required for secure login implementation. Finally, the Work Value criterion (C3) with a weight of 0.122 was multiplied by a rating of 5, recognizing the fundamental importance of authentication for system security and user access. The individual products (3.240, 0.920, and 0.610) were then summed to produce a raw score of 4.770, which was subsequently normalized to 4.42 on the 5-point scale to maintain consistency with the rating system used throughout the application. This detailed calculation demonstrates the transparency and reproducibility of the AHP method, allowing Project Managers to trace exactly how each criterion contributed to the final priority ranking. The same computational logic was systematically applied to all other backlog items (B2 and B3), enabling objective comparison and evidence-based sprint planning decisions.

3.1.3. AHP Process Flow Diagram

To operationalize objective backlog prioritization, the system integrates a fully automated Analytical Hierarchy Process (AHP) module within its backend logic. This module eliminates the need for manual spreadsheet calculations by programmatically executing Saaty's mathematical framework in real-time. The workflow initiates when the Project Manager inputs the pairwise comparison values for the defined criteria, triggering a sequential pipeline of matrix normalization, weight extraction, and consistency verification. Once the criteria weights are mathematically validated, the system processes the specific ratings for each backlog item to compute the final priority scores. Figure 2 provides a comprehensive visual representation of this end-to-end computational workflow, detailing the six distinct phases from initial data entry to the final generation and deployment of the sprint backlog.

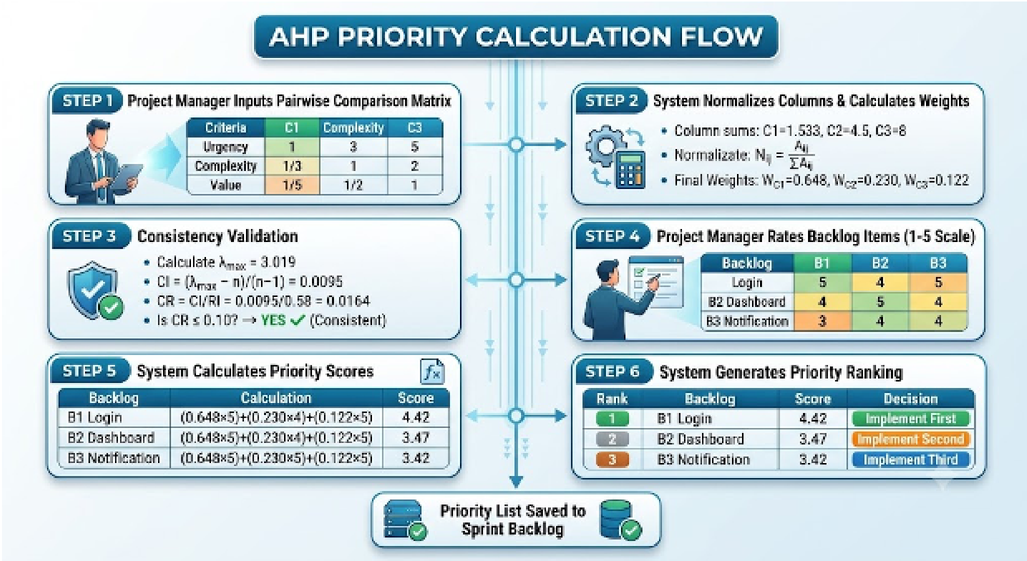

Figure 2. Systematic Workflow of the AHP Priority Calculation and Ranking Process

As illustrated in the workflow, the calculation process is strictly governed by mathematical validation to ensure decision reliability. In Steps 1 and 2, the system dynamically normalizes the pairwise comparison matrix to derive the final criterion weights, identifying Urgency (0.648) as the most dominant factor, followed by Complexity (0.230) and Work Value (0.122). Crucially, Step 3 acts as a logical gatekeeper; the system calculates the Consistency Ratio ($CR = 0.0164$) and verifies that it falls below the acceptable 0.10 threshold. If the input were logically inconsistent, the system would reject the weights and prompt the user for revision, thereby preventing flawed decision-making. Upon successful validation, Steps 4 and 5 demonstrate how the system applies these validated weights to the Project Manager's 1–5 scale ratings for each backlog item, computing the precise weighted sum scores (e.g., 4.42 for the Login feature). Finally, Step 6 translates these raw scores into an actionable, ordered ranking, automatically designating the implementation sequence and seamlessly pushing the prioritized list into the active Sprint Backlog. This automated flow ensures that sprint planning is both mathematically rigorous and operationally efficient.

3.1.4. Analysis of AHP Results

The Login Feature (B1) achieved the highest priority score (4.42), confirming its status as the most critical foundational component. This outcome is logical because:

- **High Urgency (5/5):** Authentication is a prerequisite for all other system features.
- **Moderate Complexity (4/5):** While requiring security implementation, it is less complex than dashboard analytics.
- **High Work Value (5/5):** Without login, no other features can be securely accessed.

The Dashboard Feature (B2, score 3.47) ranked second due to its high work value and complexity, while the Notification Feature (B3, score 3.42) ranked third as both its urgency and complexity were comparatively lower. These results demonstrate that AHP effectively transforms subjective prioritization into a reproducible, criterion-driven ranking, directly addressing the subjective backlog prioritization problem identified in prior literature [21, 22].

3.2. Black Box Testing Result

Functional testing was comprehensively applied to every main feature developed in the application. The testing process employed a structured approach with predefined test cases, expected outcomes, and pass/fail criteria. All tested scenarios, including user authentication, backlog CRUD operations, sprint management, Scrum Board state transitions, and AHP computation, achieved 100% functional compliance.

3.2.1. Detailed Black Box Testing Scenarios

Table 5 presents the complete test case specifications with preconditions, test steps, expected results, and actual outcomes:

Table 5: Comprehensive Black Box Testing Results

TC-ID	Module	Test Scenario	Preconditions	Test Steps	Expected Result	Actual Result	Status
TC-001	Auth	Valid login	User registered	1. Enter valid email 2. Enter valid pass 3. Click Login	Redirect to Dashboard	Redirected to Dashboard	PASS
TC-002	Auth	Invalid email	User registered	1. Enter invalid email 2. Enter pass 3. Click Login	Show "Invalid credentials" error	Error message displayed	PASS
TC-003	Auth	Invalid password	User registered	1. Enter valid email 2. Enter wrong pass 3. Click Login	Show "Invalid credentials" error	Error message displayed	PASS
TC-004	Auth	Empty fields	None	1. Leave fields empty 2. Click Login	Show "Field required" validation	Validation shown	PASS
TC-005	Backlog	Create backlog	Logged in as PM	1. Navigate to Backlog 2. Click "Add" 3. Fill form 4. Save	Item appears in backlog list	Item saved & displayed	PASS
TC-006	Backlog	Edit backlog	Backlog exists	1. Select item 2. Click "Edit" 3. Modify 4. Save	Changes reflected in list	Changes saved correctly	PASS
TC-007	Backlog	Delete backlog	Backlog exists	1. Select item 2. Click "Delete" 3. Confirm	Item removed from list	Item deleted successfully	PASS
TC-008	AHP	Calculate priority	$\geq$2 backlogs exist	1. Select items 2. Input ratings 3. Click "Calculate"	System displays ranked list	Ranking generated correctly	PASS
TC-009	AHP	Inconsistent input	CR $>$ 0.10	1. Input inconsistent 2. Click "Calculate"	Show warning, request revision	Warning displayed	PASS
TC-010	Sprint	Create sprint	Backlog exists	1. Click "Create Sprint" 2. Set duration 3. Select items 4. Save	Sprint created with items	Sprint created successfully	PASS
TC-011	Sprint	Start sprint	Sprint created	1. Select sprint 2. Click "Start Sprint"	Sprint status changes to "Active"	Status updated correctly	PASS
TC-012	Sprint	Close sprint	Sprint active	1. Click "Complete Sprint" 2. Confirm	Sprint marked as completed	Sprint closed successfully	PASS
TC-013	Board	Move To Do → In Progress	Task in To Do	1. Drag task card 2. Drop in "In Progress"	Task moves, status updates	Task moved & saved	PASS
TC-014	Board	Move In Progress → Done	Task in Progress	1. Drag task card 2. Drop in "Done"	Task moves, completion % updates	Task moved & metrics updated	PASS
TC-015	Board	Move Done → To Do	Task in Done	1. Drag task card 2. Drop in "To Do"	Task returns to backlog state	Task reverted successfully	PASS
TC-016	Task	Assign task to employee	Sprint active	1. Select task 2. Choose employee 3. Assign	Task shows assignee name	Assignment saved correctly	PASS
TC-017	Task	Update description	Task exists	1. Open detail 2. Edit description 3. Save	Changes reflected immediately	Description updated	PASS
TC-018	Dash	View sprint progress	Sprint active	1. Navigate to Dashboard	Show burndown chart, completion %	Metrics displayed correctly	PASS
TC-019	Dash	View team workload	Tasks assigned	1. Navigate to Dashboard	Show task distribution per employee	Workload visualization shown	PASS
TC-020	User	Logout	User logged in	1. Click "Logout" button	Return to login screen, clear session	Session terminated	PASS

Test Coverage Summary:

- Total Test Cases: 20
- Passed: 20 (100%)
- Failed: 0 (0%)

- Critical / Major / Minor Defects: 0

3.2.2. Black Box Testing Workflow

To ensure the functional integrity and reliability of the developed Android application, a systematic Black Box Testing methodology was employed. This approach evaluates the system's behavior strictly based on its specified functional requirements, treating the internal code structure as a "black box" to focus purely on user inputs and system outputs. The testing process was not conducted ad-hoc; rather, it followed a rigorous, five-phase workflow designed to guarantee comprehensive coverage of all core modules, including authentication, backlog management, AHP calculation, and Scrum Board interactions. Figure 3 illustrates the complete end-to-end workflow of this testing methodology, detailing the structured progression from initial test case design and environment configuration through to sequential execution, defect analysis, and final system validation.

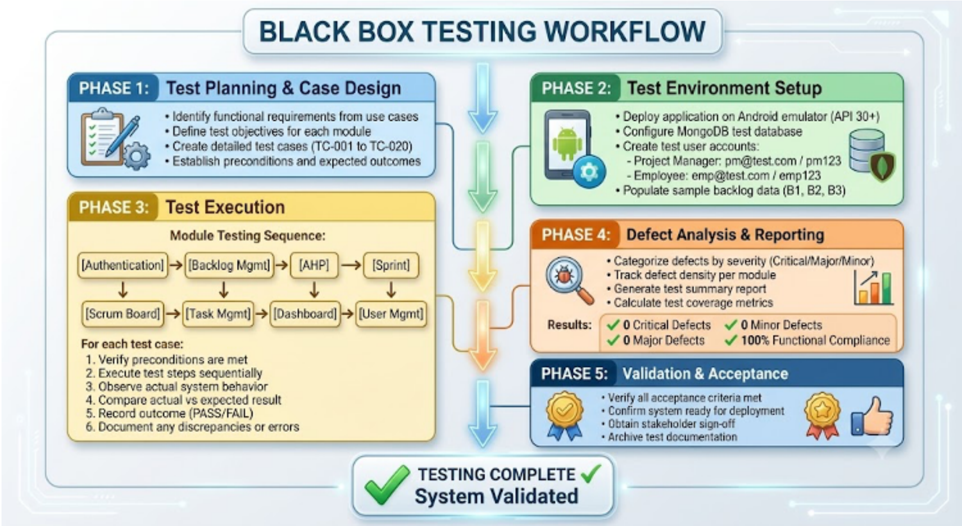

Figure 3. Black Box Testing Workflow

As depicted in the workflow, the execution phase (Phase 3) followed a logical module sequence, testing foundational features like Authentication before progressing to complex, state-sensitive modules such as the AHP priority calculator and the Scrum Board. For each of the 20 designed test cases, the actual system behavior was meticulously compared against the predefined expected outcomes to identify any discrepancies. The results, summarized in Phase 4, were highly conclusive: the system exhibited zero critical, major, or minor defects, ultimately achieving a 100% functional compliance rate. This flawless execution confirms that the application accurately handles user inputs, database transactions, and UI state updates exactly as specified during the planning phase. Consequently, the process successfully culminated in Phase 5 (Validation and Acceptance), verifying that all acceptance criteria were met and stakeholder sign-off was obtained. This structured validation confirms that the MERN-based mobile application is robust, functionally complete, and fully prepared for real-world deployment in a software development environment.

3.3. Comparison with Related Work

To clearly delineate the novelty and distinct contributions of this research, a comparative analysis was conducted against recent, relevant studies in the domains of Agile project management, decision support systems, and mobile application development. While prior literature has extensively explored the individual benefits of Scrum methodologies, MERN Stack architectures, and AHP-based decision-making, these elements have predominantly been implemented in isolation or restricted to web-based environments. Table 6 summarizes this comparative landscape, highlighting the methodological and technological distinctions between existing solutions and the proposed system, thereby underscoring the specific research gap that this study addresses.

The comparative analysis presented in Table 6 confirms the unique positioning of this study. While Hilmyansyah et al. [1] and Rizzi et al. [11] successfully demonstrated that Scrum and digital Scrum Boards enhance monitoring transparency and team collaboration, their solutions remain confined to web-based platforms and lack formal, mathematical prioritization mechanisms. Similarly, while Istiqomah et al. [9] validated the effec-

Table 6: Positioning of This Study Relative to Key Prior Works

Author (Year)	Method	Technology	Key Finding
Hilmyansyah [1]	Scrum	Web Based	Improved monitoring transparency
Istiqomah [9]	Scrum	Android	100% sprint goal achievement
Gunawan [6]	MERN Stack	Web Based	0% error rate, SUS score 78.98
Rizzi [11]	Scrum Board	Web (React Vite)	Enhanced team collaboration
This Study	**Scrum + AHP**	**MERN (Mobile)**	**Objective backlog prioritization integrated**

tiveness of Scrum in an Android environment, their system did not address the upstream challenge of objective backlog selection prior to sprint planning. Furthermore, although Gunawan et al. [6] proved the high reliability and usability of the MERN Stack, their implementation was web-centric and devoid of Agile methodology integration.

In contrast, this study bridges these fragmented contributions by presenting the first integrated mobile solution that combines Scrum workflow management with AHP-driven decision support on the MERN Stack. This synthesis is transformative rather than merely additive. By embedding objective, criterion-based prioritization directly into the mobile application, the system eliminates the need for external spreadsheets or standalone decision-support tools. Consequently, it significantly reduces cognitive load and context-switching for Project Managers, ensuring that sprint planning is both mathematically rigorous and seamlessly integrated into the daily, on-the-go development workflow. The comparison confirms this study's novelty: it is the first to integrate Scrum-based project management with AHP-driven backlog prioritization in a single Android mobile application built on the MERN Stack. Prior work applied Scrum and AHP separately, either in web-based systems [1, 6], or as standalone decision-support tools [21, 22]. The integration eliminates the need for external prioritization tools, embedding objective decision support directly within the project workflow.

3.4. Discussion

The empirical findings of this study demonstrate that integrating the Analytical Hierarchy Process (AHP) into a mobile Scrum environment effectively resolves the inherent subjectivity of traditional backlog prioritization. The AHP module successfully generated a defensible priority ranking, with the Login Feature (B1) scoring highest (4.42) due to its critical urgency and work value. Crucially, the Consistency Ratio (CR) of 0.017 confirms that the decision-making process was mathematically sound and logically coherent. Furthermore, the Black Box Testing yielded a 100% pass rate across all 20 functional scenarios. This zero-defect outcome validates the reliability of the MERN Stack for complex, state-sensitive mobile applications, corroborating previous findings by Gunawan et al. [6] regarding the robustness of this architecture in production environments.

When compared to existing literature, this study establishes a clear novelty. While prior research has successfully applied Scrum to improve monitoring transparency [1] or achieve sprint goals in Android apps [9], these solutions lacked formal, built-in prioritization mechanisms. Similarly, studies utilizing AHP have typically treated it as a standalone decision-support tool [14, 15] or applied it exclusively to web-based systems [1, 11]. This research bridges that gap by being the first to embed objective, criterion-driven AHP calculations directly into the daily workflow of a mobile-first Scrum Board. This integration eliminates the need for external spreadsheets or separate tools, thereby reducing cognitive load and context-switching for Project Managers.

From a practical perspective, the proposed application offers significant value, particularly for small and medium-sized enterprises (SMEs) and distributed teams. By providing a free, open-source, and mobile-accessible platform, it democratizes access to rigorous project management tools that are typically locked behind expensive enterprise licenses. The mobile-first design ensures that team members can update task statuses and review priorities in real-time, regardless of their physical location, which is critical in modern hybrid work environments. Theoretically, this work contributes to the maturation of Agile methodologies by demonstrating how formal decision science (AHP) can be harmonized with Agile's adaptive nature to satisfy the growing demand for auditable and transparent development processes.

Despite these contributions, several limitations must be acknowledged. First, the AHP weights were derived from a single Project Manager's perspective, which may not reflect the consensus of a larger, cross-functional team. Second, the functional testing did not encompass non-functional aspects such as load testing, stress testing, or security penetration testing under heavy concurrent usage. Finally, the evaluation was limited to a small set of sample backlog items, which does not fully capture the complexity of large-scale enterprise backlogs with intricate dependencies. Future research should address these gaps by implementing group-AHP for multi-stakeholder consensus, integrating WebSocket for true real-time collaboration, and exploring machine learning models to predict backlog complexity based on historical sprint data.

4. CONCLUSION

This research successfully developed and validated a mobile-based project management application that seamlessly integrates Agile Scrum methodology with the Analytical Hierarchy Process (AHP) for objective backlog prioritization. Built upon the robust MERN Stack architecture (MongoDB, Express.js, React Native, Node.js), the system provides a structured, transparent, and mathematically rigorous environment tailored for both Project Managers and development team members. The empirical findings strongly demonstrate the efficacy of the proposed solution. The integrated AHP module successfully transformed inherently subjective backlog prioritization into a reproducible, criterion-driven ranking system. The calculation yielded a highly consistent Consistency Ratio (CR) of 0.017, well below the acceptable 0.10 threshold, ensuring that the decision-making logic is sound and defensible. For instance, foundational components such as the Login Feature were accurately identified as the highest priority (Score: 4.42) based on quantifiable weights for Urgency, Complexity, and Work Value. Furthermore, comprehensive Black Box Testing across 20 distinct functional scenarios confirmed a 100% compliance rate with zero critical or major defects, validating the reliability of the MERN architecture for handling state-sensitive mobile interactions like Scrum Board updates. The primary novelty of this study lies in its unified, mobile-first approach. Unlike prior literature that treated Scrum workflow management and AHP decision support as separate, often web-based, entities, this research embeds objective mathematical prioritization directly into the daily mobile workflow. Practically, this integration eliminates the reliance on external spreadsheets or disjointed tools, significantly reducing cognitive load and context-switching for Project Managers. It also democratizes access to enterprise-grade, auditable project management logic for small and medium-sized software development teams operating in distributed or hybrid environments.

Despite these significant contributions, the study is bounded by certain limitations. The AHP criteria weights were derived from a single Project Manager's perspective, which may not fully capture the consensus of a larger, cross-functional team. Additionally, while functional testing was exhaustive, the study did not encompass non-functional evaluations such as large-scale load testing, stress testing, or security penetration testing under heavy concurrent usage.

Therefore, future research should address these boundaries through several targeted directions. First, implementing a Group-AHP mechanism would allow the system to aggregate and reconcile multi-stakeholder judgments, enhancing the collaborative nature of sprint planning. Second, integrating WebSocket protocols would enable true real-time, multi-user synchronization on the Scrum Board.

ACKNOWLEDGMENTS

The authors express gratitude to the Informatics Study Program, Faculty of Computer Science, Universitas Bhayangkara Jakarta Raya for providing research facility support. Special thanks to R. Wisnu Prio Pamungkas, S.Kom., M.Kom. as the research supervisor for guidance throughout the execution of this research.

DECLARATIONS

Author Contributions:

Hafiyan Rizqi Sanjaya: Conceptualization, Methodology, Software Development, Data Acquisition, Data Cleaning, Writing – Original Draft.

R Wisnu Prio Pamungkas: Supervision, Validation, Writing – Review & Editing.

Fried Sinlae: Manuscript Review, Journal Writing Support, Writing – Review & Editing.

Funding:

The authors received no financial support for the research, authorship, and/or publication of this article.

Conflicts of Interest:

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability:
Data will be made available on request.

AI Use Statement:
During the preparation of this work, the authors used Qwen in order to improve the language and readability of the manuscript. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the final content of the published article.

Additional Information:
No additional information is available for this paper.

REFERENCES

[1] M. Hilmyansyah, M. Malabay, H. Simorangkir, and Y. Yulhendri, "Implementasi Metode Scrum Pada Pembangunan Sistem Informasi Monitoring Progress Proyek Berbasis Web (Studi Kasus: PT Quatra Engineering Mandiri)," *Ikraith-Informatika*, vol. 6, no. 3, pp. 30–40, 2022, doi: 10.37817/ikraith-informatika.v6i3.2198.

[2] V. A. Prastika and A. Voutama, "Evaluasi Keberhasilan Scrum dalam Pengembangan Aplikasi Diabeats sebagai E-Commerce Pemesanan Makanan Sehat," *JOISIE (J. Inf. Syst. Informatics Eng.)*, vol. 9, no. 1, p. 136, 2025.

[3] S. A. Fadillah, N. Chandra, and C. Rivatunisa, "Implementasi Agile Scrum Pada Pembuatan Website Sistem Informasi Manajemen Kuliner," *Decode J. Pendidik. Teknol. Inf.*, vol. 4, no. 1, pp. 301–315, 2024, doi: 10.51454/decode.v4i1.357.

[4] D. P. Budi, "Decision Making Analysis To Increase the Production Capacity and Profit of Pt. Xyz Using Analytic Hierarchy Process (Ahp) Approach," *Int. J. Sci. Technol. Res.*, vol. 29, no. 12, pp. 31–89, 2025.

[5] K. P. Karyawan, S. Linawati, A. Fauzi, and A. Zumarniansyah, "Metode Analytical Hierarchy Process (AHP) Sebagai Sistem Pendukung," *J. Comput. Sci.*, vol. 2, no. 1, pp. 75–81, 2022.

[6] D. Gunawan et al., "Implementasi MERN Stack pada Pengembangan Sistem Penerimaan Peserta Didik Baru," *Swabumi*, vol. 11, no. 2, pp. 102–110, 2023, doi: 10.31294/swabumi.v11i2.15965.

[7] A. M. Syarafi and R. Kurniawan, "Implementasi MERN Stack Pada Inventori Menejemen Sistem di Toko Arabic Parfume Bandar Lampung," *J. Inf. Technol. Educ.*, vol. 2, no. 93, p. 35142, 2025.

[8] M. H. Ar Rahman and M. Husni Rifqo, "Implementasi Metode Scrum Agile Dan Analisis Return On Investment (ROI) Pada Sistem Informasi Perpustakaan," *JSAI (J. Sci. Appl. Informatics)*, vol. 8, no. 2, pp. 306–310, 2025, doi: 10.36085/jsai.v8i2.8187.

[9] D. A. Istiqomah, P. Wulandari, A. Dhiantono, F. Annafi, S. T. A. Ramadhani, and V. A. Windarni, "Implementasi Scrum Dalam Pembangunan Aplikasi Presensi Siswa Berbasis Android Di SMK Budi Mulia Dua," *Indones. J. Comput. Sci.*, vol. 4, no. 2, pp. 127–136, 2025, doi: 10.31294/ijcs.v4i2.9944.

[10] H. R. Sanjaya, L. Situmorang, M. Syahrul, L. Kalmany, and R. W. P. Pamungkas, "Pengembangan Sistem Ujian Online Berbasis Mobile Dengan Agile-Scrum Untuk Meningkatkan Efisiensi Dan Pengalaman Pengguna," *J. Kecerdasan Buatan dan Teknol. Inf.*, vol. 4, no. 1, pp. 60–69, 2025, doi: 10.69916/jkbti.v4i1.209.

[11] T. A. Rizzi, R. A. Abdillah, Y. E. Yancandra, and M. Wijaya, "Implementasi React Vite dalam Pengembangan Antarmuka Sistem," *J. Processor*, vol. 20, no. 1, pp. 121–130, 2025.

[12] V. Krotov, "Using Scrum for Managing Group Projects in a Face-to-Face Information Systems Project Management Course," *J. Manage. Educ. Appl.*, vol. 2025, no. 2, 2025, doi: 10.17705/3jmwa.000094.

[13] C. Hansen Löfstrand and C. Backman, "Control or protection? Work environment implications of police body-worn cameras," *New Technol. Work Employ.*, vol. 36, no. 3, pp. 327–347, 2021, doi: 10.1111/ntwe.12201.

[14] R. H. M. Napitupulu, C. Handayani, and H. Haryati, "Metode Simple Additive Weighting (SAW) Untuk Menentukan Karyawan Outsourching Terbaik Di PT Bank BNI Cabang Cirebon," *Bina Insan ICT J.*, vol. 8, no. 2, p. 166, 2021, doi: 10.51211/biict.v8i2.1584.

[15] F. R. Mulyadi and Y. Syahidin, "Rancang Bangun Sistem Informasi Kepegawaian Dengan Metode Waterfall," *Explor. Sist. Inf. dan Telemat.*, vol. 12, no. 2, p. 186, 2021, doi: 10.36448/jsit.v12i2.2056.

[16] S. Rawal Dewa and J. Jasmir, "Sistem Pendukung Keputusan Pemilihan Jurusan Menggunakan Metode SAW Pada SMK Negeri 2 Sarolangun," *J. Manaj. Sist. Inf.*, vol. 8, no. 1, pp. 115–127, 2023, doi: 10.33998/jurnalmsi.2023.8.1.768.

[17] H. Putri, F. Rini, and A. Pratama, "Pemanfaatan Teknologi Informasi Dalam Pengembangan Sistem Basis Data," *J. Pustaka Data*, vol. 2, no. 1, pp. 5–10, 2022.

[18] K. Dian and F. Setio, "Implementasi Sistem Pendukung Keputusan Peminatan Peserta Didik SMA menggunakan Metode AHP (Analytic Hierarchy Process) dan SAW (Simple Additive Weighting)," *J. Tek. Elektro*, vol. 7, no. 2, pp. 57–61, 2015.

[19] D. R. Wibawa, Y. Tb, and F. Irawan, "Sistem Pendukung Keputusan Rekomendasi Penerimaan Bantuan Langsung Tunai (BLT) Pandemi Covid 19 Pada Desa Bale Atu Kabupaten Aceh Tengah Dengan Metode Analytical Hierarchy Process (AHP)," *Informatics Comput. Sci.*, vol. 7, no. 2, pp. 87–98, 2021.

[20] Y. B. Ari Widatama, N. Anwar, A. M. Widodo, and A. Ichwani, "Backend Infrastructure and Specifications Design Using OpenAPI and API-First on CV Elang Java Mandiri," *J. Indones. Sos. Teknol.*, vol. 5, no. 8, pp. 3708–3720, 2024, doi: 10.59141/jist.v5i8.1174.

[21] M. Joss and C. Cahyaningtyas, "Sistem Penilaian Prioritas Penerima Bantuan Langsung Tunai Dengan AHP Sebagai Metode Pendukung Keputusan," *J. Komput. Teknol. Inf. Sist. Inf.*, vol. 4, no. 2, pp. 587–597, 2025, doi: 10.62712/juktisi.v4i2.473.

[22] M. Rizky, "Penggunaan Metode AHP (Analytical Hierarchy Process) Dalam Sistem Penunjang Keputusan: Systematic Literature Review," *J. Penelit. Sist. Inf.*, vol. 2, no. 2, pp. 1–13, 2024, doi: 10.54066/jpsi.v2i2.1695.