

Development and Evaluation of an Adaptive Scrum Model for Wedding Organizer Systems Using UML and ISO 25010

Haisyam Maulana^{1,*}, Rian Dwicahya Supriatman², Tuti Rohayati³, Nana Yudi Permana⁴

Faculty of Engineering, Galuh University
Jl. R. E. Martadinata No.150, Ciamis, West Java, Indonesia

¹haisyam.maulana22@unigal.ac.id

²riandwicahyasupriatman@unigal.ac.id

³tutirohayati@unigal.ac.id

⁴nana_yudi_permana@unigal.ac.id

Received: 29-04-2026; Revised: 16-05-2026; Accepted: 10-06-2026.

Abstract

This study proposes an adaptive Scrum model for developing a web-based Wedding Organizer system by integrating structured modeling and iterative development practices. Unlike conventional Scrum implementations, the proposed model incorporates Unified Modeling Language use case diagrams for defining product backlog, working hour-based effort estimation for sprint planning, and a client validation loop to enhance continuous user involvement. The system was developed through iterative sprints aligned with wedding preparation phases, ensuring better synchronization between development activities and real-world workflows. An analyst role within the development team was introduced to maintain consistency between system modeling and implementation. System evaluation was conducted based on ISO/IEC 25010, focusing on functional suitability and usability aspects. The results indicate that all system features operate as expected, and usability testing using the System Usability Scale achieved a score of 82.5, categorized as Excellent. The findings demonstrate that the proposed model improves requirement clarity, planning accuracy, and user involvement, leading to enhanced system usability and development efficiency. This study contributes a structured and practical approach to adapting Scrum for event-based application domains, particularly in Wedding Organizer systems.

Keywords: *adaptive scrum, iso 25010, UML, usability, wedding organizer*

This is an open access article under the CC BY-SA license.



Corresponding Author:

*Haisyam Maulana

Faculty of Engineering, Galuh University

Jl. R. E. Martadinata No.150, Ciamis, West Java, Indonesia

Email: haisyam.maulana22@unigal.ac.id

I. INTRODUCTION

The service sector, particularly in the Wedding Organizer (WO) industry, has shown rapid growth driven by increasing public demand for professional and integrated event management services. Wedding Organizers are responsible not only for executing wedding events but also for managing a wide range of

activities, including package reservations, vendor coordination, concept design development, and scheduling of event timelines. Despite this growth, many Wedding Organizer businesses still depend on manual or partially digital systems, which often result in operational inefficiencies such as communication gaps with clients, challenges in handling design revisions, and limited transparency in monitoring project progress.

These limitations can reduce service quality and customer satisfaction, potentially affecting business sustainability. To overcome these challenges, the adoption of web-based information systems has become an effective approach to streamline operational processes and enhance service performance [1, 2, 3]. In addition, recent developments in software engineering highlight the increasing adoption of Agile methodologies, which are known for their flexibility in responding to evolving user requirements [4, 5].

Among Agile frameworks, Scrum is widely recognized for its iterative development cycles and emphasis on collaboration between development teams and stakeholders [6]. Prior research indicates that Scrum can significantly improve productivity and software quality, particularly in projects characterized by dynamic and frequently changing requirements [7, 8]. However, its standard implementation is generally designed to be domain-independent, which makes it less optimal for specific application areas such as Wedding Organizer systems. These systems require timeline-oriented workflows and continuous client involvement throughout the development and execution phases [9, 10].

To address this gap, this study proposes an adaptive Scrum model specifically tailored to the operational characteristics of Wedding Organizer systems. The proposed model integrates iterative client validation and aligns development activities with event-based timelines. System representation in this research is carried out using Unified Modeling Language, a widely accepted standard for modeling both the structural and behavioral aspects of software systems [11]. UML facilitates a clear and systematic depiction of interactions among system components and stakeholders.

Furthermore, system quality is evaluated using the ISO/IEC 25010 standard, which provides a comprehensive framework for assessing software performance and quality attributes [12]. The evaluation focuses on functional suitability and usability to ensure that the system performs its intended functions correctly while also being easy to use. This approach is supported by previous studies emphasizing that software quality plays a crucial role in the successful implementation of information systems [10, 13].

Based on these considerations, this research aims to develop and evaluate an adaptive Scrum model for a web-based Wedding Organizer system. The primary contribution of this study lies in proposing a domain-specific Scrum adaptation and validating its effectiveness through system modeling and quality evaluation.

II. RESEARCH METHOD

A. Research Approach

This study employs an Agile-based software development approach using the Scrum framework to develop a web-based Wedding Organizer system. Scrum is selected due to its iterative and incremental nature, which enables continuous refinement of system features and effective collaboration between developers and stakeholders [6, 4].

However, the standard implementation of Scrum is generally domain-independent and does not fully accommodate the specific characteristics of Wedding Organizer systems, which involve timeline-based workflows and intensive client interaction. Therefore, this study proposes an adaptive Scrum model tailored to the operational needs of Wedding Organizer services [9, 10].

B. Research Stages

The research is conducted through the following sequential stages:

1. **Requirement Analysis:** This stage focuses on identifying system requirements based on Wedding Organizer business processes, including booking management, vendor coordination, design revision, and scheduling.
2. **System Design:** The system is designed using Unified Modeling Language to represent both structural and behavioral aspects of the system [11]. The modeling includes use case diagrams, activity diagrams, class diagrams, and sequence diagrams.
3. **System Development:** The system is developed as a web-based application by implementing the proposed adaptive Scrum model.
4. **System Evaluation:** The developed system is evaluated to ensure its quality and performance based on defined criteria.

C. Adaptive Scrum Model

This research proposes an adaptation of the Scrum framework tailored to the workflow of Wedding Organizer systems (see Figure 1). The model emphasizes alignment with event timelines and active client involvement. The stages of the proposed model are as follows:

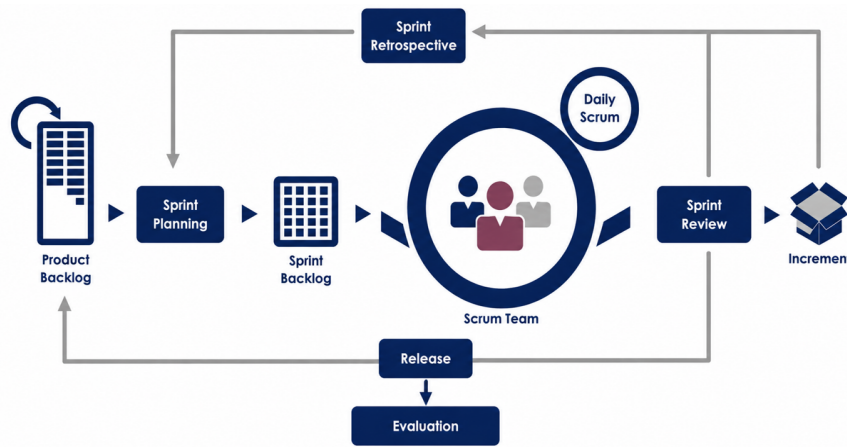


Fig. 1. Adaptive Scrum Framework Integrated with UML Modeling and Client Validation.

1) Product Backlog

In Scrum, the product backlog represents a prioritized list of features, functionalities, and development tasks required to achieve the project objectives [6, 5]. Rather than functioning as a software design artifact, the product backlog serves as a collection of work items that guide iterative development throughout sprint cycles.

In this study, Unified Modeling Language (UML) use case diagrams are utilized to support requirement identification and backlog definition. The use case diagram helps identify interactions between stakeholders and the system, enabling clearer feature identification relevant to Wedding Organizer operational processes. Therefore, the use case model does not replace the product backlog but acts as a supporting artifact to systematically define backlog items.

Each identified use case is translated into one or more backlog items representing implementation tasks to be completed during sprint execution. For example, use cases such as Login, Booking, Notification, and Vendor Management are transformed into actionable backlog entries for sprint planning.

In addition, each backlog item is assigned an effort estimation in terms of working hours to improve planning clarity and workload distribution. Although Scrum commonly employs relative estimation techniques such as story points, this study adopts working-hour estimation as a practical approach suited to small-scale and domain-specific development environments.

2) Sprint Planning

Sprint planning is conducted by selecting backlog items based on event timelines and effort estimation. Each sprint represents a phase of wedding preparation, ensuring alignment between development activities and real-world workflows.

3) Sprint Execution (Development) / Client Validation Loop

During the sprint execution phase, development activities are carried out iteratively by the development team.

In this study, the development process includes a specialized role within the team responsible for system analysis and modeling activities. An analyst, as part of the development team, is responsible for refining system requirements and producing detailed system models using Unified Modeling Language, including activity diagrams, sequence diagrams, and class diagrams.

These models are continuously updated based on feedback obtained from the client validation loop. Although Scrum traditionally emphasizes cross-functional teams without rigid role separation, this study

introduces an Analyst as a distinct functional entity due to the complexity of requirement clarification and UML-based system modeling in Wedding Organizer systems.

The analyst is responsible for maintaining consistency between evolving client requirements, UML artifacts, and implementation outputs.

This role becomes particularly relevant in event-based domains where client preferences frequently change and require structured documentation to minimize requirement ambiguity. This approach ensures that system design remains aligned with evolving user requirements while maintaining consistency between system modeling and implementation.

4) *Sprint Review and Sprint Retrospective*

At the end of each sprint, completed features are reviewed, and the development process is evaluated to improve future performance.

5) *Release*

The system is prepared for deployment after completing several sprint cycles.

D. *System Evaluation*

System evaluation is performed based on the ISO/IEC 25010 standard to ensure software quality in terms of functional suitability and usability [12]. These evaluation aspects are selected to verify whether the developed Wedding Organizer system performs its intended functions correctly and provides a satisfactory user experience.

1) *Functional Suitability*

Functional suitability evaluation is conducted using the Black Box testing method. Black Box testing is a software testing approach that evaluates system functionality based on input-output behavior without considering the internal structure or implementation of the program [14]. This method is commonly used to validate whether software features operate according to predefined requirements and expected outcomes [15].

In this study, functional testing is applied to verify core system features, including authentication, booking management, and notification services. Each feature is tested based on predefined scenarios, expected outputs, and actual system responses.

The functional suitability score is calculated using the following formula:

$$\text{Functional Suitability} = \frac{\text{Number of Valid Functions}}{\text{Total Tested Functions}} \times 100\% \quad (1)$$

where:

Number of Valid Functions represents the number of system functions that successfully meet expected outcomes, while Total Tested Functions represents the total number of evaluated functions. The functional suitability percentage is then interpreted to determine whether the system successfully meets operational requirements.

2) *Usability*

Usability evaluation is conducted using the System Usability Scale (SUS), a standardized and widely accepted instrument for measuring system usability and user satisfaction [16]. SUS consists of 10 questionnaire items designed to evaluate system ease of use, learnability, consistency, and overall user experience using a five-point Likert scale ranging from strongly disagree (1) to strongly agree (5).

The usability testing involved 25 respondents representing the main stakeholders of the Wedding Organizer system, consisting of Wedding Organizer staff, clients (brides/grooms), and vendor representatives who interacted directly with system functionalities. This respondent composition was selected to ensure usability assessment from multiple perspectives relevant to operational system usage.

The SUS questionnaire items used in this study are presented in Table I. The SUS score is calculated using the following formula:

$$SUS = \left(\sum_{i=1}^{10} \text{Score}_i \right) \times 2.5 \quad (2)$$

For odd-numbered items (Q1, Q3, Q5, Q7, and Q9), the contribution score is calculated by subtracting 1 from the respondent's rating. For even-numbered items (Q2, Q4, Q6, Q8, and Q10), the contribution score is calculated by subtracting the respondent's rating from 5. The total score is then multiplied by 2.5 to obtain a final usability score ranging from 0 to 100 [16].

TABLE I. SYSTEM USABILITY SCALE (SUS) QUESTIONNAIRE.

No	SUS Statement
Q1	I think that I would like to use this system frequently
Q2	I found the system unnecessarily complex
Q3	I thought the system was easy to use
Q4	I think that I would need technical support to use this system
Q5	I found the various functions in this system were well integrated
Q6	I thought there was too much inconsistency in this system
Q7	I would imagine that most people would learn to use this system very quickly
Q8	I found the system very cumbersome to use
Q9	I felt very confident using the system
Q10	I needed to learn a lot of things before I could get going with this system

III. RESULTS AND DISCUSSION

A. Result

The results of the requirement analysis and system modeling reveal the need for a development approach that is capable of handling dynamic workflows and intensive user interaction in Wedding Organizer systems.

Therefore, an adaptive Scrum model is applied to translate these findings into a structured development process. The model integrates UML-based requirement definition, iterative sprint execution, and continuous client validation.

1) Product Backlog

The use case diagram is employed to support requirement elicitation and feature identification for backlog preparation. The diagram illustrates the interaction between system actors and functionalities (Figure 2), which are subsequently translated into product backlog items used during sprint planning.

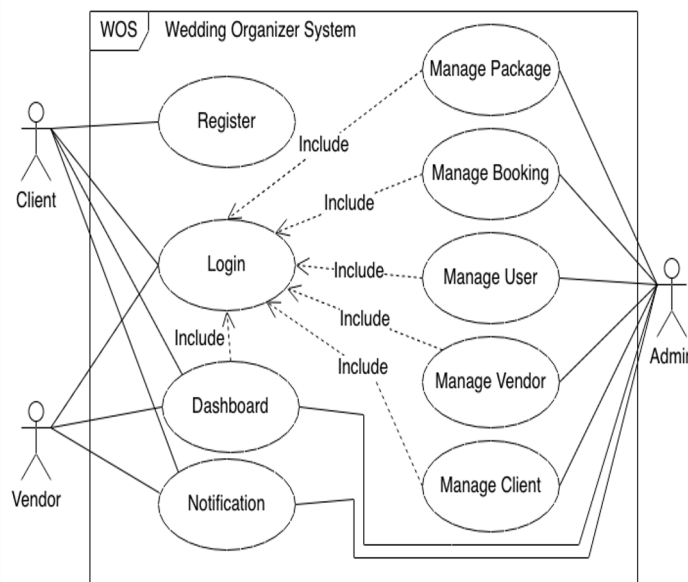


Fig. 2. Use Case Diagram Supporting Product Backlog Identification.

2) *Sprint Planning*

In Scrum, backlog items are generally defined based on priority and estimated effort to support iterative development [6, 4]. While relative estimation techniques are commonly used, this study adopts working hour-based estimation as a practical approach to represent development effort more concretely. This approach supports better workload distribution and enhances sprint planning accuracy, particularly in small-scale development environments. The defined sprint planning purposes and their corresponding effort estimations are presented in Table II.

TABLE II. SPRINT PLANNING PURPOSES.

No	Product Backlog Item	Sprint	Value / Estimation (hours)
1	Implement Authentication Feature	Sprint 1	8
2	Develop User Management Module	Sprint 1	16
3	Develop Booking Registration	Sprint 2	8
4	Implement Notification Service	Sprint 2	16
5	Develop Dashboard Interface	Sprint 2	16
6	Develop Client Management	Sprint 3	16
7	Develop Vendor Management	Sprint 3	16
8	Develop Package Management	Sprint 4	16
9	Develop Booking Management	Sprint 4	16

3) *Sprint Execution*

Development is carried out iteratively by the development team. An analyst, as part of the team, is responsible for refining requirements and producing system models using Unified Modeling Language. A Client Validation Loop is implemented, allowing users to provide feedback at each iteration. The feedback is used to update both system features and UML models.

Sprint 1:

Daily Scrum: The Daily Scrum activities in Sprint 1 resulted in refined task allocation among team members, identification of initial technical constraints in the authentication module, and adjustment of backlog priorities lead by Scrum Master and Analyst. These outcomes contributed to a more structured daily work plan and improved coordination during the early development phase.

Design: The design phase produced initial system models using Unified Modeling Language by Analyst, including activity diagrams for user authentication workflows (Figure 3), class diagrams defining core entities (Figure 4), and initial user interface designs (Figure 5, Figure 6). These artifacts established a clear foundation for system development.

Implementation: The implementation phase generated functional features related to user management, including login and registration modules. The developed components were successfully integrated and aligned with the defined system requirements, as shown in Figure 7.

Review: The review process confirmed that the implemented authentication features satisfied the defined functional requirements. However, several minor improvements related to input validation and user interface clarity were identified to enhance system quality in subsequent iterations.

Sprint 2:

Daily Scrum: The Daily Scrum activities in Sprint 2 resulted in improved synchronization of development tasks, identification of issues related to booking data flow, and refinement of backlog items to better support booking functionality. The team achieved more efficient coordination compared to the previous sprint.

Design: The design phase produced refined system models, including activity diagrams for booking processes (Figure 8), updated class diagrams to accommodate booking entities (Figure 9), and improved user interface designs (Figure 10, Figure 11). These refinements enhanced the clarity and consistency of the system structure.

Implementation: The implementation phase generated core booking features, including package selection and booking management functionalities. The developed features were integrated with the existing system modules and demonstrated improved system capability, as presented in Figure 12 and Figure 13.

Review: The review process indicated that the booking features functioned as expected according to quality assurance testing results. However, several improvements related to user experience and information presentation were identified to further enhance system usability in subsequent iterations.

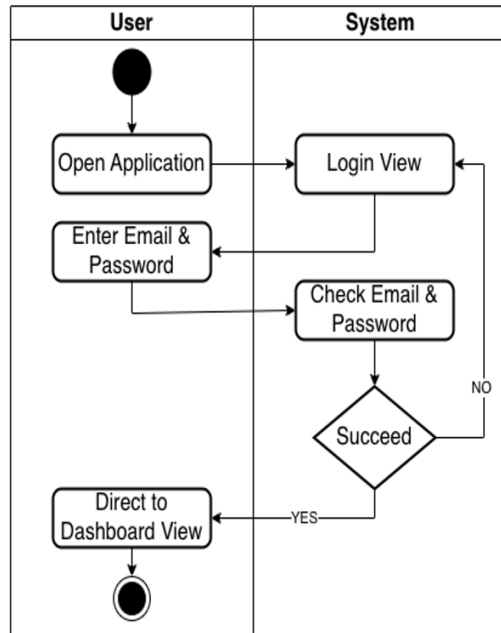


Fig. 3. Activity Diagram Login.

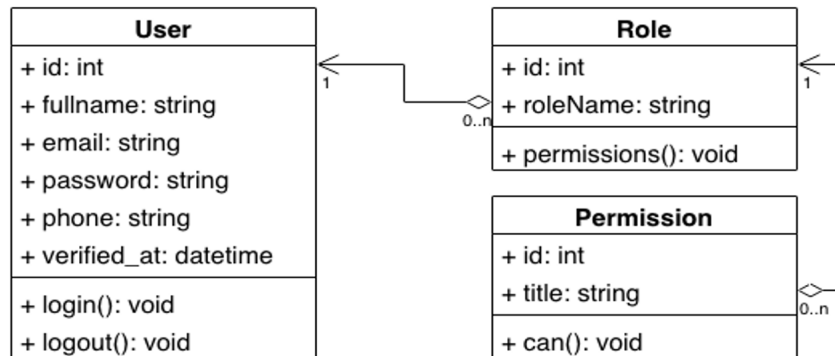


Fig. 4. Class Diagram Login.

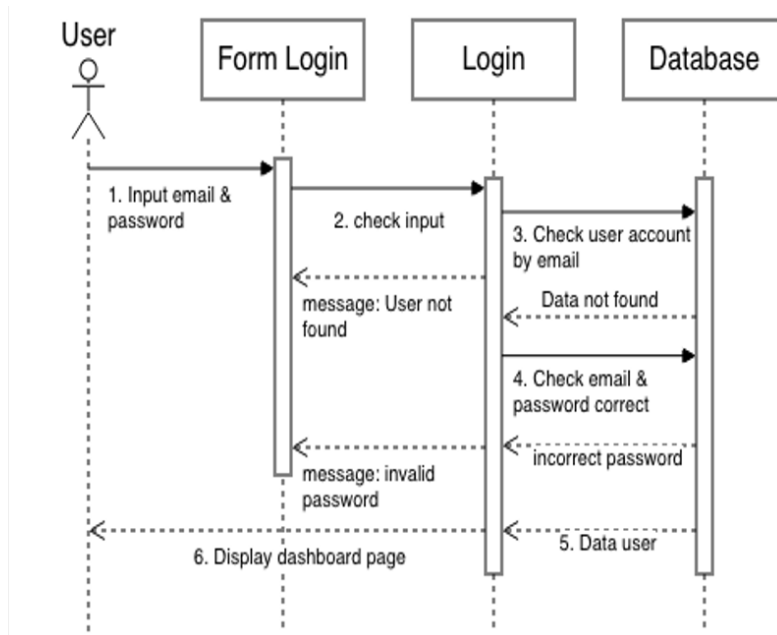


Fig. 5. Sequence Diagram Login.

Log in to your account

Enter your email and password below to log in

Email Address *

Password *

Remember me

[Forgot your password?](#)

Don't have an account? [Sign up](#)

Fig. 6. Desain UI Login.

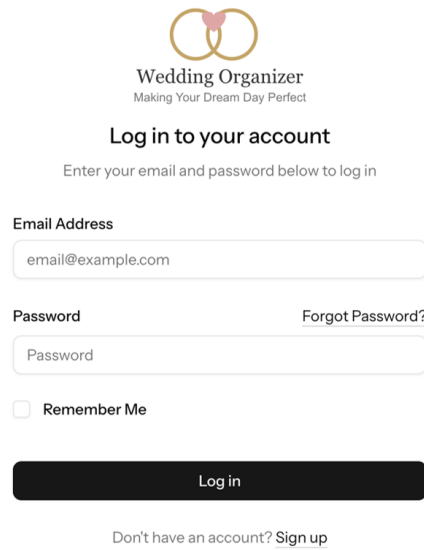


Fig. 7. Login Page.

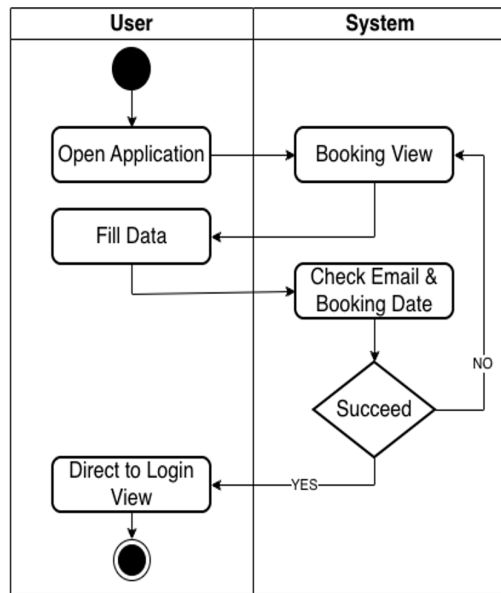


Fig. 8. Activity Diagram Booking.

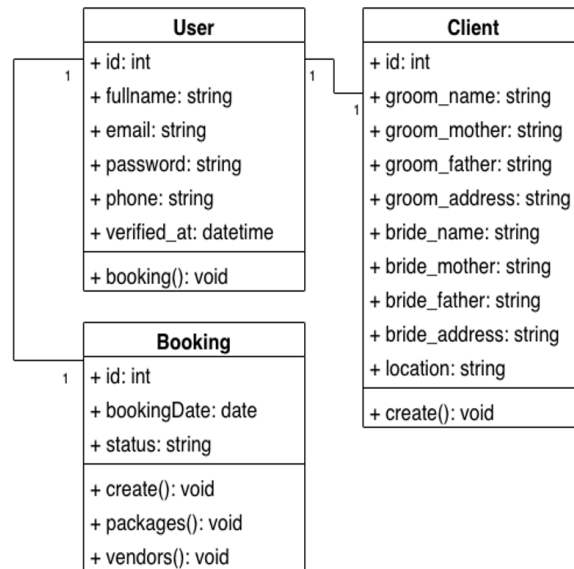


Fig. 9. Class Diagram Booking.

Create your account

Fill in the information below to create a new account

Full Name *

Email Address *

Password *

Confirm Password *

Booking Date *

I agree to the [Terms and Conditions](#) and [Privacy Policy](#)

Sign up

or

Already have an account? [Log in](#)

Fig. 10. User Interface Booking.

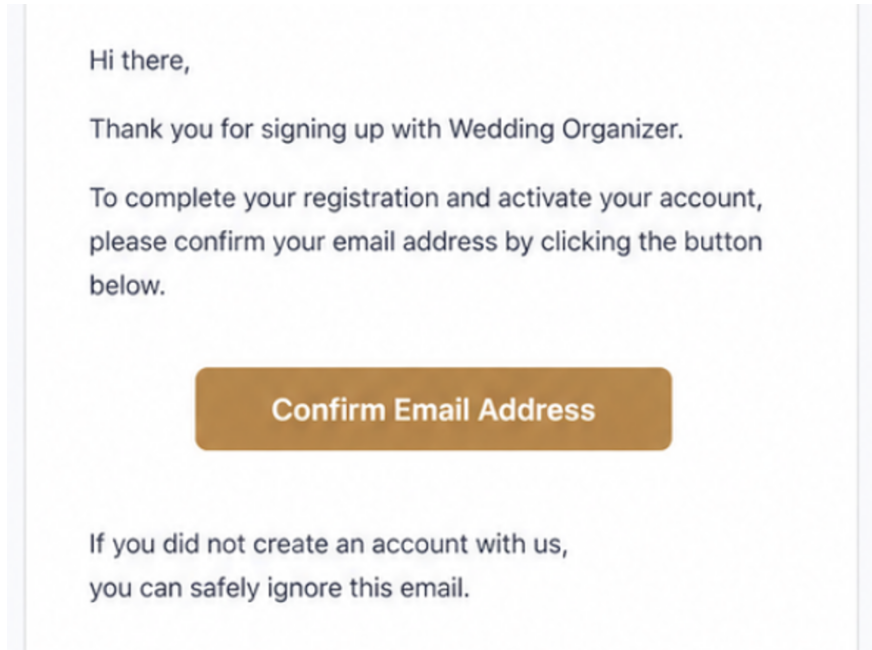


Fig. 11. User Interface Notification.

Wedding Organizer
Making Your Dream Day Perfect

Create Account

Create a new account to get started

Name

Email Address

Password

Confirm Password

Booking Date

Don't have an account? [Log in](#)

Fig. 12. Booking Page.



Wedding Booking Confirmation

Thank you for registering and submitting your wedding booking request.

Please verify your email address to proceed with your booking.

Our Wedding Organizer team will contact you within **24 hours** to confirm your selected booking date and further details.

Verify Email & Confirm Booking

If you did not request this booking, please ignore this email.

Fig. 13. Notification Email.

4) Release

The release process was implemented through an automated continuous integration and deployment pipeline using GitHub Actions. The system was deployed to a web hosting environment after successful validation in each sprint, ensuring consistent delivery, reduced deployment errors, and faster system availability.

5) System Evaluation

1) Functional Testing Results

Functional testing was conducted using the Black Box testing method to evaluate system functionality based on input and output behavior. The results are summarized in Table III.

TABLE III. BLACKBOX TESTING.

No	Feature	Test Scenario	Expected Result	Result
1	Login	Valid credentials input	User successfully logs in	Valid
2	Booking	User creates a booking while registering	Data stored successfully	Valid
3	Notification	The system sends a notification	User receives notification via email	Valid

The testing results indicate that all system features function as expected, demonstrating that the system meets the functional suitability criteria defined in ISO/IEC 25010.

2) Usability Evaluation Results

Usability evaluation was conducted using the System Usability Scale (SUS) involving 25 respondents representing the primary stakeholders of the Wedding Organizer system, including Wedding Organizer staff, clients (brides/grooms), and vendor representatives who directly interacted with the system functionalities.

The SUS score was calculated based on respondent evaluations using the standardized SUS scoring procedure [16]. The obtained result is SUS Score = 82.5.

According to the SUS interpretation model proposed by Bangor et al. [17], a SUS score can be interpreted using adjective ratings, acceptability ranges, and grade scales. In addition, Sauro [18] emphasizes that SUS serves as a reliable usability benchmarking instrument for evaluating user satisfaction and overall system effectiveness across software applications. Systems with scores above 80 are generally categorized as Excellent, indicating high usability and strong user acceptance. Based on the SUS interpretation scale, the obtained score of 82.5 falls into the following categories as shown in Table IV.

TABLE IV. EVALUATION RESULT.

Evaluation Aspect	Result
SUS Score	82.5
Adjective Rating	Excellent
Grade Scale	A
Acceptability Range	Acceptable

B. Discussion

The findings indicate that the proposed adaptive Scrum model provides a structured and context-oriented approach for developing Wedding Organizer systems. Agile software development approaches generally emphasize adaptability, iterative improvement, and stakeholder collaboration as essential elements for handling changing requirements [19]. Unlike conventional Scrum implementation, which is generally domain-independent [6, 5], the proposed model introduces several adaptations to better accommodate the operational characteristics of Wedding Organizer services, including timeline-based workflows, intensive client involvement, and dynamic requirement changes.

One of the main contributions of this study lies in the integration of Unified Modeling Language (UML) to support product backlog definition. In conventional Scrum, product backlog items are commonly represented in the form of user stories or prioritized work items [6]. However, requirement ambiguity frequently becomes a challenge in software projects involving multiple stakeholders and changing expectations [3, 20]. In this study, the UML use case diagram was utilized as a supporting artifact to identify and define system functionalities before translating them into backlog items. This approach contributes to improving requirement clarity and reducing misunderstandings between stakeholders and developers.

The results of sprint execution demonstrate that the proposed adaptation improved development organization and feature delivery. Each sprint was structured according to Wedding Organizer operational priorities, allowing features to be developed incrementally based on actual business needs. Sprint 1 focused on authentication and user management, while Sprint 2 emphasized booking and notification features. This phased implementation aligns with Scrum principles emphasizing incremental delivery and iterative feedback cycles [6, 21, 19]. Agile software development methods generally encourage iterative and adaptive implementation to improve flexibility and responsiveness toward changing requirements [19].

Furthermore, the inclusion of an Analyst role within the Scrum team contributed to maintaining consistency between requirement changes, UML artifacts, and implementation outputs. Although Scrum traditionally emphasizes cross-functional teams without explicit analyst separation, previous studies indicate that Scrum adaptations are frequently applied to fit organizational contexts and project complexity [5, 22]. In Wedding Organizer systems, where user requirements may frequently evolve due to changing client preferences, the analyst role becomes beneficial for maintaining requirement traceability and minimizing ambiguity during sprint execution.

Another important adaptation introduced in this study is the Client Validation Loop, which facilitates continuous stakeholder involvement throughout sprint implementation. Client feedback obtained after each sprint contributed to refining both system functionalities and UML models. This finding supports prior research emphasizing that continuous stakeholder participation in Agile development improves requirement alignment, system acceptance, and software quality [4, 23]. Through iterative validation, the risk of requirement mismatch can be minimized, especially in event-based systems requiring frequent adjustments.

From a project management perspective, the implementation of working-hour estimation also demonstrated practical value during sprint planning. Although Scrum commonly applies relative estimation techniques such as story points [6], this study adopted working-hour estimation to simplify effort calculation and workload distribution in a small-scale development environment. The results indicate that this approach improved sprint predictability and supported more realistic development scheduling. Similar findings have been reported in Agile estimation research emphasizing that contextual adaptation of estimation techniques may improve planning effectiveness depending on project scale and team composition [1, 24].

The testing results further validate the effectiveness of the proposed adaptive Scrum model. Functional suitability testing conducted using the Black Box testing method achieved a score of 100%.

This result indicates that all tested features, including login, booking, and notification services, operated according to expected requirements. From an ISO/IEC 25010 perspective, this finding demonstrates that

the developed system satisfies the functional suitability quality characteristic [12]. Previous studies also emphasize that functional testing is crucial for validating system behavior and ensuring software reliability before deployment [14, 15].

In terms of usability, the developed system achieved a System Usability Scale (SUS) score of 82.5.

According to Bangor et al. [17], this score falls within the Excellent category, achieving Grade A and classified as Acceptable in usability interpretation. This result indicates that users perceived the system as easy to learn, efficient to use, and supportive of Wedding Organizer operational activities. The high usability score may also indicate that iterative development and continuous client validation positively influenced user experience and interface quality. Similar studies suggest that Agile-based iterative refinement contributes significantly to improved usability and user satisfaction [10, 25].

Overall, the findings suggest that the proposed adaptive Scrum model successfully enhances requirement clarity, development flexibility, stakeholder involvement, and software quality. The integration of UML-supported backlog identification, analyst-assisted requirement refinement, client validation loops, and practical effort estimation contributes additional value beyond conventional Scrum implementation. Therefore, this model is considered suitable for event-based information systems characterized by dynamic workflows and continuous interaction between service providers and clients.

IV. CONCLUSION

The results demonstrate that the proposed adaptive Scrum model provides a structured and effective approach to developing Wedding Organizer systems.

The use of Unified Modeling Language-based backlog definition improves requirement clarity, while the timeline-based sprint planning aligns development activities with real-world workflows. In addition, the incorporation of effort estimation in the form of working hours supports better workload distribution and enhances planning accuracy during sprint execution.

The integration of a client validation loop ensures continuous user involvement, reducing the risk of requirement mismatch and enabling iterative refinement of system features. Furthermore, the inclusion of an analyst role within the development team strengthens the connection between system modeling and implementation, which is often not explicitly addressed in conventional Scrum practices.

From the evaluation perspective, the system satisfies the functional suitability criteria defined in ISO/IEC 25010, as all features operate according to expected results. In addition, usability evaluation using the System Usability Scale indicates a high level of user satisfaction, reflecting that the system is easy to use and well accepted by users.

Overall, the proposed model enhances development efficiency, system usability, and stakeholder collaboration, making it suitable for event-based application domains. The findings also suggest that integrating structured modeling and practical estimation techniques within Agile frameworks can provide additional value in improving software development outcomes.

ACKNOWLEDGMENT

The authors would like to express their sincere appreciation to the Wedding Organizer teams involved in this study, as well as to all respondents, including brides and grooms, vendors, and staff members, whose participation and valuable insights significantly contributed to the success of this research.

REFERENCES

- [1] J. G. Rivera Ibarra, G. Borrego, and R. R. Palacio, "Early estimation in agile software development projects: A systematic mapping study," *Informatics*, vol. 11, no. 4, p. 81, 2024. [Online]. Available: <https://doi.org/10.3390/informatics11040081>.
- [2] A. S. Reddy, "A systematic review of software development methodologies in agile and devops environments," *Journal of Recent Trends in Computer Science and Engineering*, vol. 12, no. 5, pp. 50–54, 2024.
- [3] A. A. Zaveri, J. Jaafar, E. Yafi, and S. Shehmir, "Evolution of requirements engineering in agile methodology – literature review," *Journal of Engineering Technology and Applied Physics*, vol. 6, no. 2, pp. 57–65, 2024. [Online]. Available: <https://doi.org/10.33093/jetap.2024.6.2.9>.
- [4] T. Dybå et al., "A systematic literature review of agile software development projects," *Information and Software Technology*, vol. 182, 2025.

- [5] M. Hron and N. Obwegeser, “Why and how is scrum being adapted in practice: A systematic review,” *Journal of Systems and Software*, vol. 183, p. 111110, Jan. 2022. [Online]. Available: <https://doi.org/10.1016/j.jss.2021.111110>.
- [6] K. Schwaber and J. Sutherland, *The Scrum Guide*. Scrum.org, 2020. [Online]. Available: <https://www.scrumguides.org/scrum-guide.html>.
- [7] W. N. Behutiye, P. Rodríguez, M. Oivo, and A. Tosun, “Analyzing the concept of technical debt in the context of agile software development: A systematic literature review,” *Information and Software Technology*, vol. 82, pp. 139–158, Feb. 2017. [Online]. Available: <https://doi.org/10.1016/j.infsof.2016.10.004>.
- [8] A. Jalaluddin, E. K. Budiardjo, and K. Mahatma, “Recommendation for scrum-based software development process with scrum at scale: A case study of software house xyz,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 8, pp. 401–412, Jun. 2024. [Online]. Available: <https://doi.org/10.29207/resti.v8i3.5646>.
- [9] M. Fowler and J. Highsmith, “Agile software development trends and practices,” 2021. updated studies, 2021–2023. [Online]. Available: <https://martinfowler.com/agile.html>.
- [10] R. Alnanih, “Enhancing usability: A grounded theory approach-based scrum framework for mobile application design,” *Procedia Computer Science*, vol. 241, pp. 162–170, 2024. [Online]. Available: <https://doi.org/10.1016/j.procs.2024.08.023>.
- [11] Object Management Group, *Unified Modeling Language (UML) Specification Version 2.5.1*. OMG, Needham, MA, USA, 2017. [Online]. Available: <https://www.omg.org/spec/UML/2.5.1>.
- [12] ISO/IEC, *ISO/IEC 25010:2011 Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE)*. International Organization for Standardization, Geneva, Switzerland, 2011. [Online]. Available: <https://www.iso.org/standard/35733.html>.
- [13] P. Sankhe, N. Patil, N. Patwari, A. Agar, and R. Asati, “Evolution and current trends in agile software development methodologies: A comprehensive analysis of industry adoption and practices,” *International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE)*, vol. 14, Nov. 2025. [Online]. Available: <https://doi.org/10.17148/IJARCCE.2025.1411132>.
- [14] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner’s Approach*. New York, NY, USA: McGraw-Hill, 9th ed., 2019.
- [15] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner’s Approach*. New York, NY, USA: McGraw-Hill Education, 9th ed., 2019.
- [16] J. Brooke, “Sus: A quick and dirty usability scale,” in *Usability Evaluation in Industry*, London, U.K.: Taylor & Francis, 1996.
- [17] A. Bangor, P. Kortum, and J. Miller, “Determining what individual sus scores mean: Adding an adjective rating scale,” *Journal of Usability Studies*, vol. 4, no. 3, pp. 114–123, 2009. [Online]. Available: <https://uxpajournal.org/determining-what-individual-sus-scores-mean-adding-an-adjective-rating-scale/>.
- [18] J. Sauro, *A Practical Guide to the System Usability Scale: Background, Benchmarks & Best Practices*. Denver, CO, USA: Measuring Usability LLC, 2011. [Online]. Available: <https://www.measuringu.com/sus/>.
- [19] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, “Agile software development methods: Review and analysis,” tech. rep., VTT Publications, Espoo, Finland, 2002.
- [20] D. Leffingwell, *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Boston, MA, USA: Addison-Wesley, 2011.
- [21] K. Rubin, *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Boston, MA, USA: Addison-Wesley, 2012.
- [22] M. Hron and N. Obwegeser, “Why and how is scrum being adapted in practice: A systematic review,” *Journal of Systems and Software*, vol. 183, p. 111110, 2022. [Online]. Available: <https://doi.org/10.1016/j.jss.2021.111110>.

- [23] M. Paasivaara and C. Lassenius, “Communities of practice in agile software development,” *Information and Software Technology*, vol. 57, pp. 87–100, 2015.
- [24] S. McConnell, *Software Estimation: Demystifying the Black Art*. Redmond, WA, USA: Microsoft Press, 2006.
- [25] J. Nielsen, *Usability Engineering*. San Francisco, CA, USA: Morgan Kaufmann, 1993.