Implementation of a lecturer agenda management application using the scrum method at UIN Jurai Siwo Lampung

Krisna Widatama
UIN Jurai Siwo Lampung
krisnawidatama@uinjusila.com

Submitted: September 10, 2025; accepted: October 21, 2025

Abstract

Lecturers at UIN Jurai Siwo Lampung faced significant challenges in schedule management due to the lack of a centralized system, leading to frequent agenda conflicts and coordination inefficiencies. This research aimed to develop a functional agenda management application and evaluate the effectiveness of the Scrum method in this context. The software development process utilized the Scrum framework, emphasizing iterative sprints to adaptively handle changing user requirements. The results showed that the developed application successfully minimized scheduling overlaps through automated conflict detection features. Additionally, the Scrum method proved effective in accelerating the development cycle and ensuring the product met specific user needs. This study contributes a practical solution for enhancing academic productivity and offers empirical insights into implementing Agile methodologies within educational institutions.

Keywords*: agenda management, lecturer, scrum method, software.*

1. Introduction

Modern higher education institutions face increasing demands for both operational and academic efficiency. Lecturers, as key pillars of the *tridharma*, carry complex workloads that include teaching, research, community service, and various administrative tasks. Effective agenda management therefore becomes crucial to ensure productivity and prevent scheduling conflicts. Initial observations at UIN Jurai Siwo Lampung indicate that the agenda management process for lecturers is often fragmented and not yet centrally coordinated. The absence of an integrated system has emerged as a major issue, potentially leading to overlapping schedules, difficulties in coordinating meetings, and inefficiencies in lecturers' time allocation.

The selection of the Scrum method in this study is based on the need for high flexibility in responding to the operational dynamics of academic environments. System development in higher education institutions often encounters challenges such as sudden regulatory changes and newly emerging user needs that only become apparent once the system begins to be tested. Traditional development models such as Waterfall, which are sequential and rigid, are considered less effective because they require fully defined requirements at the early stages of development. In contrast, Scrum enables development to be carried out iteratively and incrementally. This approach was chosen because it facilitates intensive involvement of lecturers as users through Sprint Review sessions, allowing mismatches in features to be detected and corrected earlier without waiting for the system to be fully completed.

Various studies [1] indicate that high lecturer workloads, when not managed with a structured system, can be directly correlated with increased work-related stress and may potentially reduce the quality of outputs in the three pillars of higher education (tridharma). Moreover, operational efficiency in universities greatly depends on the ability to effectively coordinate human resources (lecturers). Other studies [2] emphasize that the adoption of information technology to centralize administrative tasks, such as scheduling, is a key factor in enhancing institutional productivity. Therefore, the development of this application is not merely the creation of a technical support tool, but a strategic step toward optimizing organizational performance through transparent and efficient agenda management [3].

To address these coordination and efficiency challenges, the development of a centralized application system is proposed as a solution. This study applies the Scrum method throughout the software development process. Scrum, as part of the Agile framework, is selected because of its high flexibility in responding adaptively to changing user requirements during the course of the project [4]. Although previous studies have extensively examined the development of academic information systems or project management in general, research that specifically focuses on the application of the Scrum method to build a lecturer agenda management application within Islamic higher education institutions in Indonesia remains limited and rarely found. Earlier studies have largely concentrated on the development of academic scheduling systems, often employing complex algorithms such as Genetic Algorithms or Particle Swarm Optimization to optimize the automatic allocation of course schedules [5].

Another study examines the implementation of corporate project management systems to enhance efficiency [6]. From a methodological perspective, the implementation of Scrum has largely been documented within the context of large-scale commercial software industries or in the development of startup applications that prioritize rapid release cycles [7]. Nevertheless, there remains a research gap regarding the practical application of Scrum for developing internal applications with highly specific scopes, such as lecturer agenda management, which relies heavily on adapting to the unique workflow dynamics of non-profit higher education institutions.

Based on the background and research gaps previously described, this study formulates two main research problems: (1) How can a functional lecturer agenda management application be designed and implemented to align with the specific workflow at UIN Jurai Siwo Lampung? (2) How effective is the application of the Scrum method in managing dynamic changes in user requirements throughout the application development process within an academic environment? In line with these research problems, the objectives of this study are to implement a functional lecturer agenda management application and to analyze and document the implementation process of the Scrum method as a software development case study at UIN Jurai Siwo Lampung.

The primary contributions of this study consist of two aspects. Practically, it produces an application system that can be directly utilized by lecturers at UIN Jurai Siwo Lampung to enhance efficiency and transparency in agenda management. Academically, the study contributes a detailed empirical case study on the challenges and success factors of implementing the Scrum method in a higher education institutional context. The results are expected to serve as a reference for other researchers or similar institutions intending to adopt Agile methodologies for internal system development.

2. Method

The design method employed in this research is the Scrum Methodology. Scrum is an Agile framework highly effective for managing and developing complex projects, particularly in environments where requirements may be fluid or not fully defined at the outset [8]. This approach was chosen because the lecturer agenda management application requires rapid iteration, continuous validation from lecturers as *end-users*, and flexibility to dynamically accommodate feature adjustments. Scrum implementation in this study ensures that the developed application is validated and refined at the conclusion of each iteration. This research was conducted through a series of repetitive and structured development cycles known as *Sprints*. The workflow of the Scrum method applied to the application design is described through five main stages.

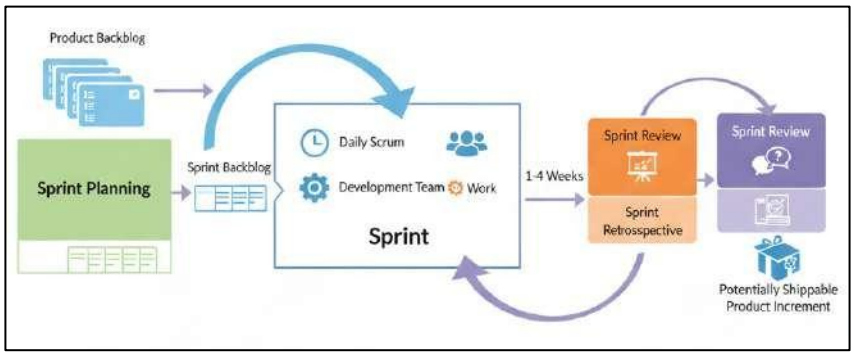

Fig. 1 The Five Stages of Scrum Method

2.1 Product Planning

This initial stage focuses on defining the application's vision and its functional requirements comprehensively [9]. The team collaborates with the *Product Owner* to compile the Product Backlog the master list containing all desired features, enhancements, and requirements (*user stories*) prioritized based on the highest business value. Feature prioritization is crucial, focusing on core functions such as "Synchronizing lecturer teaching schedules with personal calendars" and "Automatic notification system for thesis supervision deadlines. "The *Product Backlog* is dynamic and will be continuously updated throughout the project based on feedback [10].

2.2 Sprint Planning

At the start of each *Sprint*, the team holds a Sprint Planning Meeting to determine the *Sprint Goal* and select the highest priority items from the *Product Backlog* that are realistically achievable within the *Sprint*'s fixed duration [11]. These selected items are then broken down into smaller, specific technical tasks, forming the Sprint Backlog. Decisions are made collaboratively: the *Product Owner* determines *what* needs to be built, and the development team determines *how* to build it. For example, one *Sprint Goal* might be the release of the lecturer dashboard scheduling module.

2.3 Implementation and *Daily Scrum*

This is the execution phase where the development team works to deliver a functional *increment*. During this phase, the workflow is maintained transparent and measurable through the Daily Scrum Meeting (a daily stand-up meeting, maximum 15 minutes) [12]. Each team member answers three key questions: (1) What did I complete yesterday? (2) What will I work on today? (3) What impediments are I facing? This meeting helps identify and remove obstacles (*impediments*) as quickly as possible, ensuring the team remains focused on achieving the *Sprint Goal*.

2.4 Sprint Review

At the end of every *Sprint*, a Sprint Review is conducted as an informal demonstration session for the *Product Owner* and other stakeholders (lecturers). The objective is to inspect the completed and functional *increment* and gather critical feedback. Unlike internal testing, this review measures the product's acceptance from the perspective of its functionality and alignment with the lecturers' real-world needs. The feedback gathered including suggestions for new features, interface modifications, or priority changes—is recorded and injected back into the *Product Backlog* for the subsequent *Sprint* [13].

2.5 Sprint Retrospective

Following the *Sprint Review* and before the next *Sprint Planning*, the internal development team holds the Sprint Retrospective . This is an internal team meeting aimed at continuous process improvement. The team reflects on the entire *Sprint* in terms of people, relationships, processes, and tools [14]. Questions focus on: "What went well?", "What could be improved?", and "How can we apply these improvements in the next *Sprint*?". The result is an action plan for quality enhancement that is immediately implemented, ensuring that development efficiency continually improves as the project progresses [15].

3. Result and Discussion

The implementation process of the lecturer agenda management application at UIN Jurai Siwo Lampung was carried out through the full application of the Scrum method. This study produced two primary outputs that are presented in this section: (1) the results of the Scrum implementation process itself, and (2) the functional software application that was developed.

3.1 Application Architecture and Technology

The lecturer agenda management application was developed as a web-based system to ensure broad accessibility within UIN Jurai Siwo Lampung. The selection of technologies used is critical to ensuring performance, security, and ease of future development.

a. Development Framework. This study employs Laravel (the latest version available at the time of implementation) as the primary framework. Laravel was chosen due to its expressive syntax, robust Model-View-Controller (MVC) structure, and strong support for rapid application development, which aligns well with the principles of Agile Scrum.

b. Primary Programming Language. PHP is used as the server-side (backend) programming language.
c. Database. A MySQL relational database was selected for its reliability, proven scalability, and strong compatibility with the Laravel framework.
d. User Interface (Frontend). To ensure a responsive and user-friendly interface, the frontend was developed using HTML and CSS (with the Bootstrap framework for design consistency), along with JavaScript including the Vue.js library for dynamic interface elements.

This combination of technologies ensures that the developed application is not only functional but also built on a solid technical foundation, enabling smooth maintenance and future enhancements by UIN personnel.

3.2 Implementation and Evaluation of the Scrum Method

The development of this application was divided into four Sprint cycles, each with a duration of two weeks. The research team served as both the Development Team and the Scrum Master, while several representatives of lecturers and academic administrative staff were involved as the Product Owner and key stakeholders. The process began with the preparation of the Product Backlog, which contained 25 main user stories identified from the problem observations described in the Introduction.

During the development process, progress monitoring within each Sprint was carried out using a Burndown Chart. This chart visually presents the amount of remaining work in relation to the available time. As shown in Fig. 2, the Burndown Chart for Sprint 2 (which focused on the core calendar feature) illustrates the team's progress. The *Ideal Work* line represents the expected pace of completion, while the *Remaining Work* line indicates the actual workload. The upward deviation observed on days 5 and 6 reflects the addition of unexpected tasks (impediments), which were identified during the Daily Scrum and subsequently resolved.

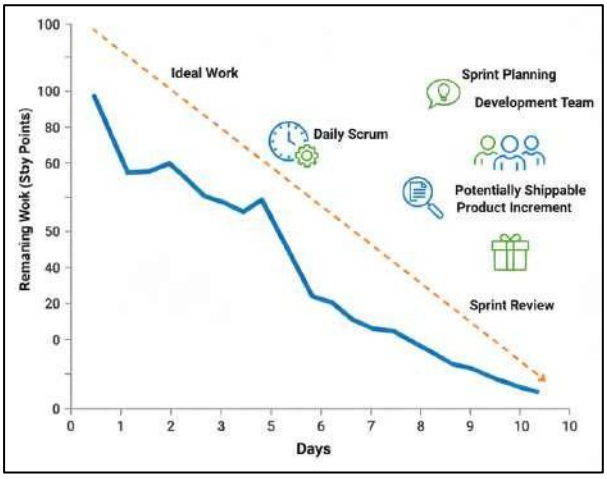

Fig. 2 Burndown Chart Sprint 2

From a methodological perspective, the discussion indicates that the implementation of Scrum was highly effective. Feedback received from lecturers during each Sprint Review session enabled the team to immediately adapt priorities. For example, the "WhatsApp notification" feature was not initially included in the Product Backlog, but it was added in Sprint 3 after lecturers identified it as an urgent need. This flexibility addressed the challenge of frequently changing requirements commonly encountered in academic environments.

3.3 Results of the Development of the Agenda Management Application

The main output of this implementation is a centralized, functional web-based lecturer agenda management application. The system is designed to directly address the issues of schedule overlap and lack of coordination that were previously identified. Figure 3 presents the interface of the application's main dashboard page. After successfully logging in, lecturers are greeted with a summary of their daily agenda, a list of upcoming activities for the week, and important notifications. This design aims to enable users to quickly identify their schedules within seconds.

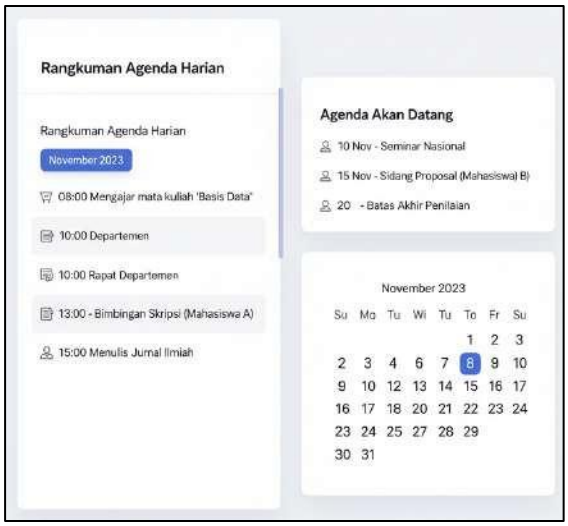

Fig. 3 Interface of Dashboard Page

To support transparency and resolve coordination issues among lecturers, the application provides a shared calendar feature, as shown in Fig. 4. A lecturer (for example, a Head of Study Program) can view the public schedules of other lecturers to identify suitable times for meetings or collaborative activities. This feature effectively reduces the need for repeated manual confirmations and minimizes scheduling errors.

Fig. 4 Interface of Shared Calender

.

The discussion of the application development results shows that the implemented functionalities align well with user needs. The schedule conflict detection feature (Fig. 3) and the shared calendar (Fig. 4) directly address the main issues outlined in the Introduction. The application successfully transforms the previously fragmented agenda management process into one that is centralized and transparent.

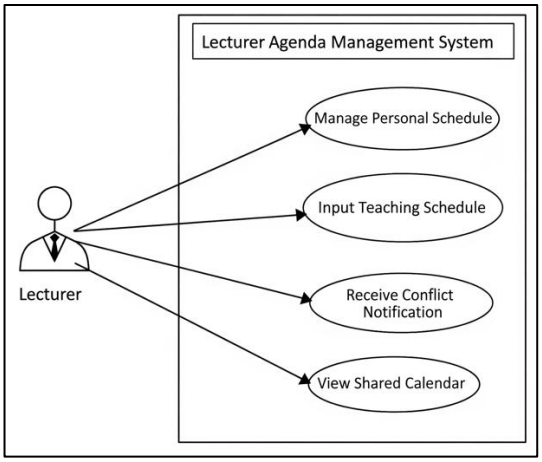

Fig. 5 Use Case Diagram

Figure 5 illustrates the functional interaction between users and the system through a Use Case Diagram. The primary actor in this system is the Lecturer, who has full access rights to manage both academic and personal agendas. The diagram defines the system boundary and includes four main use cases designed to address scheduling inefficiencies:

a. **Manage Personal Schedule**. This use case represents the core function that allows lecturers to perform CRUD (Create, Read, Update, Delete) operations on their non-routine activities. Lecturers can independently add meetings, research activities, or community service tasks. This flexibility is essential to ensure that the system can accommodate the dynamic nature of lecturers' activities, which often change.
b. **Input Teaching Schedule**. Unlike personal agendas, this use case is specifically designed for entering routine and recurring teaching schedules. The system separates this function to simplify the management of fixed academic time slots throughout a semester, which form a priority in lecturers' time allocation.
c. **Receive Conflict Notification**. This is an intelligent automatic feature of the system. When lecturers perform the *Manage Personal Schedule* or *Input Teaching Schedule* functions, the system verifies the entered time against the database. If an overlap with an existing agenda is detected, the system instantly triggers a conflict notification. This use case provides a preventive solution to avoid double-booking, which is one of the main issues addressed in this study.
d. **View Shared Calendar**. This use case is designed to enhance transparency and team coordination. Lecturers are given access to view the availability of their colleagues (for example, within the same study program). This feature facilitates the identification of common available time slots for meetings or collaborative activities without the need for repeated manual communication.

Overall, the interactions depicted in Figure 5 demonstrate how the system supports proactive and collaborative time management, replacing manual processes that are prone to errors.

3.4 Discussion

This subsection focuses on the interpretation and implications of the findings presented in the previous subsections (1 and 2). The successful development of the agenda management application and the effective implementation of the Scrum method provide several important insights.

First, the resulting application directly addresses the main issues identified at UIN Jurai Siwo Lampung, namely schedule fragmentation and coordination difficulties. The automatic conflict detection feature (Fig. 3) and the shared calendar (Fig. 4) are concrete technical solutions that have proven effective in resolving scheduling overlap. This reinforces the notion that improving lecturer efficiency does not always require complex algorithms, but rather a centralized system that is functional and easily accessible. Second, from a methodological standpoint, this case study provides empirical evidence that the Scrum method is highly relevant for software development in non-profit academic environments. Unlike commercial software industries that aim for market releases, the advantage of Scrum at UIN Jurai Siwo

Lampung lies in its ability to manage evolving user requirements. Direct feedback from lecturers during each Sprint Review enabled the team to adjust priorities, such as adding the notification feature that was not originally part of the Product Backlog. This demonstrates the superiority of Scrum over traditional waterfall methods in dealing with the dynamics of higher education institutions.

Nevertheless, the researchers acknowledge several limitations of this study. The evaluation conducted focused primarily on functionality and the development process, without including long-term usability measurements or quantitative assessments of efficiency and productivity improvements among lecturers. Additionally, the study was conducted on a limited scale within a single university, meaning that broader generalization may require further research in other institutions.

The practical implication of this study is that the resulting application is ready for full-scale trial implementation within UIN Jurai Siwo Lampung. Academically, this case study contributes as an empirical reference for the application of Scrum in the development of internal systems within the higher education sector.

4. Conclusion

This study aimed to implement a centralized lecturer agenda management application using the Scrum method to address issues of fragmentation, coordination, and inefficient time allocation at UIN Jurai Siwo Lampung. The resulting application has proven to be functional and capable of resolving these core problems through key features such as automatic schedule conflict detection and a shared calendar among lecturers. Thus, the practical objective of enhancing transparency and efficiency in agenda management has been successfully achieved.

From a methodological perspective, the application of Scrum proved to be highly effective in this case study, as its iterative nature enabled rapid adaptation to evolving user requirements based on feedback obtained during Sprint Review sessions. As a recommendation for future work, subsequent research is encouraged to expand the implementation by fully integrating this application with the existing Academic Information System (SIAKAD) and conducting a more comprehensive user experience (UX) evaluation to optimize the application interface.

Reference

[1] Nuryadi and K. Widatama, "Sistem Pembayaran Tagihan Rumah Sakit Menggunakan Metode Host to Host pada RSUD Bung Karno Kota Surakarta," *Jurnal Informatika, Komputer dan Bisnis*, vol. 2, no. 2, pp. 1–11, 2022, [Online]. Available: https://jurnal.itbaas.ac.id/index.php/jikobis

[2] Y. Kurniawan, K. Widatama, and Murhadi, "Pengembangan Sistem Informasi Perpustakaan Menggunakan Metode Prototype," *JISAI*, vol. 3, no. 2, pp. 202–211, 2023.

[3] D. K. Lintang, K. Widatama, and I. Y. Pasa, "Bridging Data Sistem Informasi Rumah Sakit (SIMRS) Rumah Sakit Dan Laboratory Information System (LIS)," *Jurnal Sistem Cerdas*, vol. 5, no. 2, pp. 92–106, 2022.

[4] R. Suhaimi, N. Santoso, and R. A. Siregar, "Pengembangan Sistem Manajemen Proyek Menggunakan Metode Scrum Berbasis Android," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 4, no. 6, pp. 1897–1905, 2020, [Online]. Available: http://j-ptiik.ub.ac.id

[5] I. Made Budi Adnyana, "Implementasi Algoritma Genetika untuk Penjadwalan Asisten Dosen di STIKOM Bali," *Jurnal Sistem dan Informatika*, vol. 12, no. 2, pp. 166–173, 2018.

[6] J. F. Widjojo, E. Rusdianto, F. Kartika, and S. Dewi, "Pembangunan Sistem Manajemen Proyek pada Perusahaan Konstruksi Bangunan Berbasis Website," *Jurnal Informatika Atma Jogja*, vol. 1, no. 1, pp. 60–70, 2020.

[7] W. Krisna, H. J. Muhammad, and D. Puspitaningrum, "Penggunaan Digital Signature Untuk Absensi Pada Universitas Muhammadiyah Purworejo," *Jurnal Sistem Cerdas*, vol. 5, no. 1, pp. 36–45, 2022, doi: 10.37396/jsc.v5i1.188.

[8] S. Pratama, S. Ibrahim, and M. A. Reybaharsyah, "Jurnal Penggunaan Metode Scrum Dalam Membentuk Sistem Informasi Penyimpanan Gudang Berbasis Web," *Informatika dan Teknologi (Intech)*, vol. 3, no. 1, pp. 27–35, 2022, doi: 10.54895/intech.v3i1.1192.

[9] V. Hema, S. Thota, S. Naresh Kumar, C. Padmaja, C. B. Rama Krishna, and K. Mahender, "Scrum: An Effective Software Development Agile Tool," *IOP Conf Ser Mater Sci Eng*, vol. 981, no. 2, 2020, doi: 10.1088/1757-899X/981/2/022060.

[10] T. Karabiyik, A. Jaiswal, P. Thomas, and A. J. Magana, "Understanding the Interactions Between the Scrum Master and The Development Team: A Game-Theoretic Approach," *Mathematics*, vol. 8, no. 9, pp. 1–21, 2020, doi: 10.3390/math8091553.

[11] M. Rizky and Y. Sugiarti, "Pengunaan Metode Scrum Dalam Pengembangan Perangkat Lunak: Literature Review," *Journal of Computer Science and Engineering (JCSE)*, vol. 3, no. 1, pp. 41–48,

2022, doi: 10.36596/jcse.v3i1.353.
[12] A. I. Ramdhani, Z. M. Subekti, and M. D. Suryadi, "Rancang Bangun Aplikasi Inventory Logistik Berbasis Website Menggunakan Metode Scrum," *J-SISKO TECH (Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD)*, vol. 5, no. 2, p. 161, 2022, doi: 10.53513/jsk.v5i2.5681.

[13] Akhtar Ali Junejo and Sajida Memon, "Case Study on Evolution Performance of Agile Scrum Software Development Life Cycle for Shopping Cart Applications," *World Journal of Advanced Engineering Technology and Sciences*, vol. 10, no. 1, pp. 085–090, 2023, doi: 10.30574/wjaets.2023.10.1.0254.
[14] Anita Wulansari, D. S. Y. Kartika, Agussalim, S. F. A. Wati, E. M. Safitri, and A. S. Fitri, "E-commerce Website Development Using Scrum Methods on Small Business," *Ijconsist Journals*, vol. 3, no. 2, pp. 8–12, 2022, doi: 10.33005/ijconsist.v3i2.69.
[15] M. S. Anggreainy, A. P. Sulaiman, C. Mathew, and K. E. Tirta, "Tasks Management: Approach to Problem Solving and its Relation to the Scrum and Agile Software Development Method," *Jurnal Ilmiah Komputasi*, vol. 20, no. 4, pp. 593–600, 2021, doi: 10.32409/jikstik.20.4.2871.