Jurnal Masyarakat Informatika, 16(1), 2025
DOI:10.14710/jmasif.16.1.73187
ISSN: 2777-0648

Applying the Scrum Method in Software Development for Undergraduate Thesis Project Implementation

Attaf Riski Putra Ramadhan, Indra Waspada*, Nurdin Bahtiar, and Adhe Setya Pramayoga
Informatics Department, Diponegoro University, Semarang, Indonesia
* *Corresponding author: indrawaspada@lecturer.undip.ac.id*

Abstract
The Scrum Method, as one of the frameworks within Agile-based software development, has become the de facto standard in industry practices. However, to date, there is no specific guideline or adaptation model that directs the application of Scrum in undergraduate thesis project settings, particularly within the Bachelor of Informatics Study Program at Diponegoro University. In this program, the final project is carried out individually by a student under the supervision of two academic advisors, forming a small team structure that differs from conventional Scrum configurations. This study proposes an adaptation model of the Scrum method for such a scenario, assigning the roles of Product Owner and Tester to the First Supervisor, Scrum Master and Tester to the Second Supervisor, and Developer as well as Assistant to the Student. The implementation of Scrum in this context facilitates structured communication between supervisors and the student, while also supporting flexibility in accommodating changing requirements throughout the development process. Moreover, active stakeholder involvement during the requirements gathering and Sprint Review stages contributes to the enhanced quality of the final deliverable. The project was executed over four sprints within a total of 40 working days, covering 64 product backlogs and several derivative tasks. The findings indicate that adapting Scrum to the context of a final project enables timely project completion with outcomes that are academically and technically accountable.

Keywords : *Final Project, Software Development, Agile, Scrum*

1 Introduction

The undergraduate thesis is a critical component in completing a Bachelor's degree in Informatics at Diponegoro University. This academic requirement involves students managing and completing an individual research or software development project under the supervision of two faculty advisors [1]. In recent years, agile methodologies have become the dominant approach to software project management, with Scrum emerging as one of the most widely adopted frameworks. Scrum defines clear roles (Product Owner, Scrum Master, and Developer) and structures work into short, time-boxed iterations called sprints [2] [3].

However, Scrum was originally designed for industrial team-based software development projects, not for individual academic projects such as undergraduate theses. As such, a systematic effort is needed to examine how Scrum roles can be redefined in this context particularly considering the unique configuration of two academic supervisors and a single student. Moreover, Scrum routines such as sprint planning, daily stand-ups, and retrospectives must be adjusted to match the academic rhythm, which often lacks daily interactions [4], [5].

Received: 13 May 2025 ; Revised: 29 May 2025; Accepted: 30 May 2025; Published: 31 May 2025

In this context, adapting Scrum to academic environments requires considering specific factors such as schedule flexibility, student autonomy, and the dual role of academic supervisors as facilitators and assessors [6]. This study proposes a case-based exploration of Scrum adaptation in the undergraduate thesis setting at the Informatics Program, Diponegoro University. The research aims to identify the potential benefits and challenges of implementing Scrum as a thesis supervision framework. It is expected that the study will result in an adaptive model that enhances the effectiveness of academic guidance, increases supervisor engagement, and enables students to manage their final projects in a more structured and iterative manner. Ultimately, the findings may contribute to innovation in academic supervision methodologies and inspire similar implementations in other academic programs with comparable characteristics.

2 Related Works

The application of Scrum in small teams has been explored across diverse domains, consistently demonstrating its adaptability and positive impact on communication, responsiveness to change, and product quality. Numerous studies show successful Scrum adoption in both traditional software development environments and unconventional settings such as high schools, social work, and small businesses. For instance, Prasetyo et al. [7] reported how a student team in an Indonesian high school utilized Scrum to develop an e-voting application, improving both the product outcome and civic engagement. Similarly, Borissov [8] detailed how a German social work organization adapted Scrum to coordinate interventions for at-risk youth, enabling flexible responses to evolving needs. These cases exemplify Scrum's capacity to enhance team coordination and outcome relevance beyond its original context. However, these implementations typically involved multi-member teams and short-term deliverables, which may not translate directly to the highly individualized and long-term nature of academic thesis projects.

In the domain of small and medium enterprises (SMEs), Scrum has been implemented to support digital transformation and operational efficiency. Basri et al. [9] and Tokenov [10] reported how laundry services and IT startups, respectively, benefited from Scrum through improved service delivery and internal collaboration. Other studies emphasized the importance of customizing Scrum to fit organizational constraints. Caballero and Calvo-Manzano [11] highlighted how very small enterprises introduced tailored phases like "design sprint" and "test sprint" to align Scrum with contractual limitations. Further, companies in Sri Lanka and Brazil [12] reported increased team motivation and faster delivery cycles after structured Scrum adoption, despite initial challenges with Agile transition. These findings are instructive but still assume a multi-role team setup, which is not present in individual thesis supervision models.

Beyond the software sector, Scrum has been applied in mechanical product development, pharmaceutical IT projects [13], and cross-organizational collaborations [14], revealing its scalability and flexibility. Hybrid models, such as the Agile Framework for Small Projects [15], offer structured guidance for small-scale Agile adoption. Meanwhile, tools like Scrum Essentials Cards [16] and CI/CD pipelines have been shown to enhance Scrum implementation. Collectively, these studies underscore that while the core principles of Scrum remain consistent, successful adoption often requires contextual adaptation, especially in small-team environments where roles, resources, and workflows may significantly differ from traditional Agile settings. However, these studies primarily focus on team-based settings where feedback loops and accountability are distributed among members. The implications for a setting where a single student must simultaneously embody multiple roles remain under-explored.

Although this body of research affirms Scrum's adaptability, there is limited critical attention to how Scrum roles specifically the Product Owner, Scrum Master, and Development Team can be meaningfully reinterpreted in academic contexts. Existing studies often emphasize process tailoring rather than role adaptation, neglecting the unique dynamics of undergraduate thesis projects where the "team" comprises just one student and supervising faculty. Moreover, these works rarely consider how academic constraints such as semester schedules, institutional evaluation criteria, and solo development responsibilities affect Agile implementation. This presents a significant gap that this study aims to address: developing a Scrum role adaptation framework that aligns with the structural and pedagogical characteristics of thesis supervision in higher education.

3 Research Methods

This chapter outlines the methodology adopted to address the research problem, including the research activity plan and the overall flow of the methodology applied in the final project. The methodology consists of four main phases: literature review scrum & adaptation frameworks, scrum adaptation framework, implementation framework application, and empirical evaluation. The methodological flow is structured and sequential, aiming to systematically solve the problem presented in this study. The complete research methodology flow is illustrated in Figure 1.

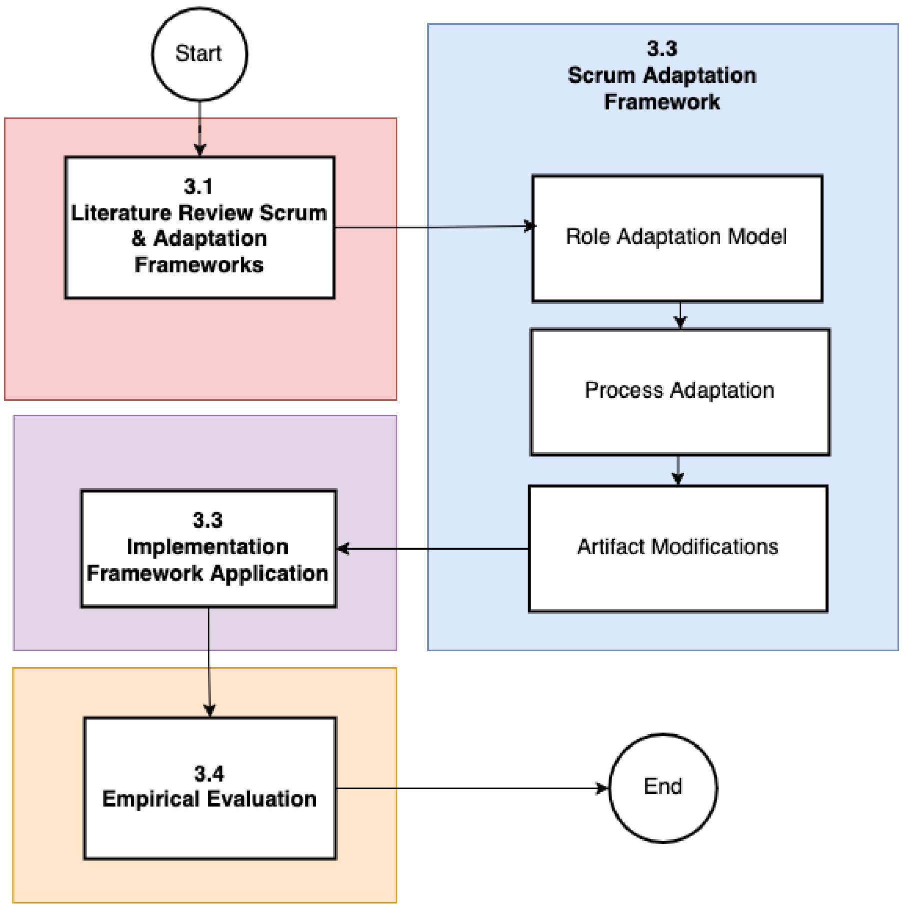


Figure 1 Research Methodology Flow

The research methodology consists of four major components. Each of these components is elaborated in Sections 3.1 to 3.4.

3.1 Literature review scrum & adaptation frameworks

Scrum is one of the most widely adopted Agile frameworks, originally developed to manage complex software projects through iterative development and continuous feedback [2]. It is structured around three key roles (Product Owner, Scrum Master, and Development Team), five events (including Sprints, Sprint Planning, and Retrospectives), and several artifacts such as the Product Backlog and Sprint Backlog. These components work in synergy to promote transparency, adaptability, and delivery of incremental value.

In practice, applying Scrum in non-traditional settings often requires contextual adaptation. Numerous adaptation frameworks have emerged to address challenges when standard Scrum cannot be implemented as-is. These adaptations may involve modifying roles, events, or artifacts to align with specific organizational, technical, or cultural constraints. For instance, in small teams or resource-limited environments, it is common to merge roles, extend sprint durations, or use asynchronous communication channels in place of daily stand-ups [4].

Academic settings, particularly individual thesis projects, present a unique context in which standard Scrum implementations face practical limitations. Unlike industry projects, undergraduate theses are typically undertaken by a single student, guided by one or two academic supervisors. The limited team size, rigid academic schedules, and predefined milestones necessitate a rethinking of Scrum's core structure. Yet, current literature lacks a focused framework for adapting Scrum roles and processes to this academic context. While some educational studies have implemented Scrum in group-based student projects (as discussed in Section 2), there is minimal guidance on how to translate these roles when the development is conducted by an individual under supervision.

This study builds upon existing adaptation frameworks and introduces a tailored Scrum implementation model for individual thesis development. The proposed framework emphasizes role reinterpretation, modified Scrum ceremonies, and academic-specific artifact definitions. By addressing the gap in literature regarding Scrum role adaptation in thesis supervision contexts, the study seeks to provide a practical guide for structured and iterative thesis development supported by Agile principles.

3.2 Scrum Adaptation Framework

This section presents the proposed Scrum adaptation framework specifically designed for academic thesis supervision contexts. The framework addresses the unique challenges of applying industry-standard Scrum practices to individual academic projects. Figure 2 illustrates the Scrum flow with modifications to the Scrum Team, where the Product Owner and Tester are represented by the First Supervisor, the Scrum Master and Tester by the Second Supervisor, and the Developer and Assistant roles are assigned to the Student.

3.3.1 Role Adaptation Model

The implementation of Scrum in this study involved three primary actors, each assuming specific roles and responsibilities. The First Supervisor served as the Product Owner, bearing full responsibility for defining the product vision, managing the product backlog, and supporting the Scrum Master during the testing process. The Second Supervisor acted as the Scrum Master, facilitating the Scrum process, ensuring that Scrum principles were upheld, and actively participating in testing to verify functional requirements. The student took the role of Developer, tasked with implementing the system based on the sprint backlog. With approval from the supervisors, the student was also allowed to assist in responsibilities typically held by the Product Owner and Scrum Master when necessary. The mapping can be seen in the Table 1.

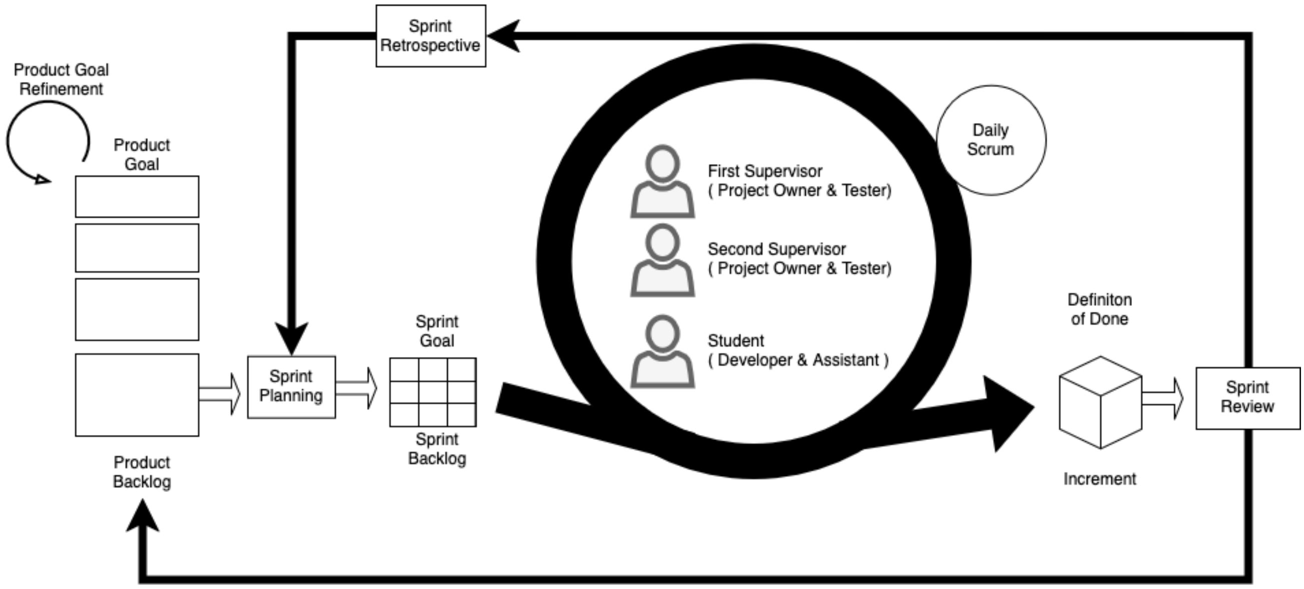


Figure 2 Scrum Flow with Scrum Team Modification

Table 1 Mapping Role

Thesis Project Role	Role	
	Standard Role	Additional Role
1st Supervisor	Project Owner	Tester
2nd Supervisor	Scrum Master	Tester
Student	Developer	Assistant

In addition to the standard Scrum roles, this thesis project also involved additional roles. Specifically, the first and second academic supervisors assumed supplementary responsibilities as testers. This adjustment was made because it is considered inappropriate for developers to test the systems they develop themselves; therefore, the testing role was delegated to the supervisors. Furthermore, the student also took on auxiliary roles as an assistant to each primary Scrum role, such as Assistant Product Owner and Assistant Scrum Master. This arrangement was necessary to ensure that the student, as the sole executor of the thesis project, could fulfil all required activities while still adhering to the structure of the Scrum framework. The supervisors' responsibility was to provide guidance and oversight to ensure that the student, in their assistant roles, performed the tasks correctly and effectively.

3.3.2 Process Adaptation

Standard Scrum ceremonies were also modified to align with academic time constraints and supervisory practices. Daily meetings were reduced from the typical every day to 2 days in a week, accommodating the less frequent but more substantial academic meetings. The adapted event can be seen in the Table 2.

Table 2 Adapted Event

Event	Standard Time	Adapted Time	Academic Event
Daily Meetings	Everyday	Twice a week	-
Sprint Planning	The beginning of sprint	The beginning of sprint	Supervision meetings
Sprint Reviews	The end of sprint	The end of sprint	Progress presentation
Sprint Retrospective	The end of sprint	The end of sprint	Progress presentation

Sprint Planning sessions were conducted during scheduled supervision meetings, while Sprint Reviews were integrated with regular thesis progress presentations. Sprint Retrospectives were adapted into supervision feedback sessions, enabling reflection and iteration within the academic mentoring framework.

3.3.3 Artifact Modifications

In adapting Scrum artifacts to academic needs, several adjustments were made to ensure compatibility with thesis deliverables and institutional expectations. The adapted artifact detail can be seen in the Table 3.

Table 3 Adapted Artifact

Scrum Artifact	Academic Adaptation
Product Backlog	Aligned with the overall scope of the thesis and organized according to academic milestones (e.g., final submission deadlines).
Sprint Backlog	Adjusted to reflect weekly or monthly targets as determined in supervision meetings.
Definition of Done	Reinterpreted to include academic quality standards, such as code functionality, completeness of documentation, and alignment with thesis evaluation criteria.

The Product Backlog was aligned with the scope of the thesis and structured around academic milestones, such as final submission. Sprint Backlogs were synchronized with weekly or monthly targets established during supervision. Finally, the Definition of Done was reinterpreted to reflect academic quality standards, including code functionality, documentation quality, and alignment with thesis evaluation criteria.

To facilitate supervisors in monitoring the ongoing processes within a sprint especially since tasks are primarily added by the student while supervisors provide direction an additional support tool is required. A modified Kanban board was therefore introduced to better align with the specific needs of this academic context. The details of the Kanban board can be seen in Figure 3.

The use of this Kanban board offers significant benefits by facilitating both the supervisor and the student in tracking project progress, thereby streamlining the reporting process during supervision sessions. The Scrum Kanban board consists of several columns, namely: Backlog, Next Sprint, Sprint Backlog, Dev, Testing, Done, and Blocked Task. The student, acting as the Scrum assistant, is responsible for actively updating and moving tasks across the board, with guidance and input from the first and second academic supervisors.

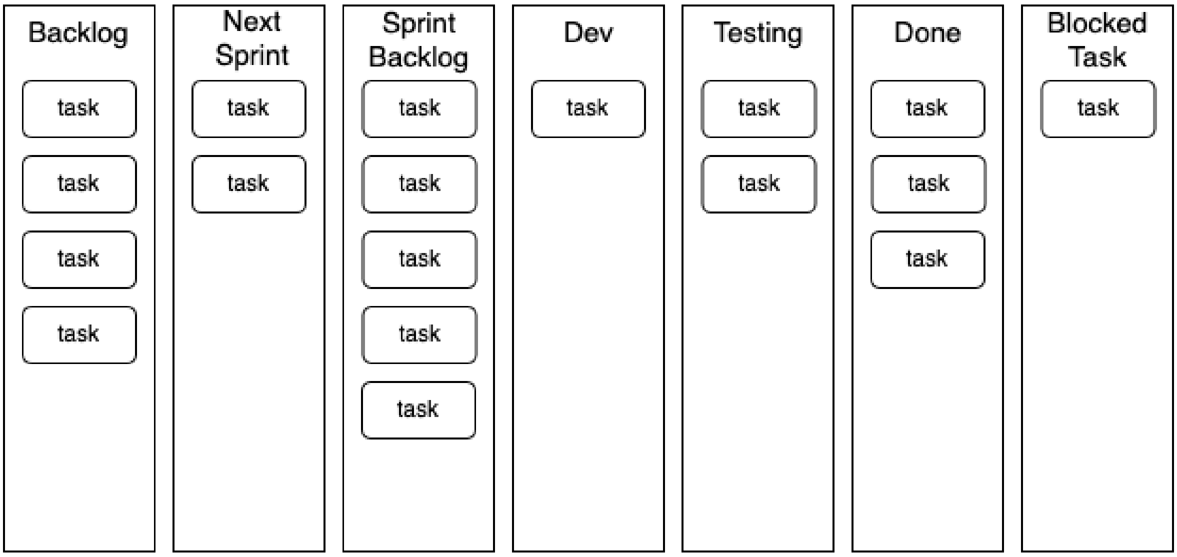


Figure 3 Kanban Board

3.3 Implementation Framework Application

This study adopts a case study approach to explore the application of the Scrum methodology in the context of an undergraduate thesis project. The research was conducted in a natural academic environment, focusing on a real-world software development project carried out by a student under the guidance of academic supervisors.

This study was conducted within the Informatics Program, Faculty of Science and Mathematics, Diponegoro University, during the academic year 2024/2025. The program follows a traditional thesis supervision model where each student is assigned two academic supervisors for their final project. The case selected for this study involves one undergraduate student (the researcher) undertaking a thesis project focused on developing an official correspondence management system. Also involves two faculty supervisor with complementary expertise which is first supervisor, specializing in system information and project management and the Second supervisor, expert in scrum, ui/ux, and software testing. The thesis project involves developing a web-based correspondence management system for the Faculty of Science and Mathematics, with the following characteristics that shown in the Table 4.

Table 4 Project Characteristic

Characteristic	Target
Duration	2 Months
Scope	Full-stack web application using ReactJS and ElysiaJS
Stakeholder	Faculty Administrative staff
Expected Deliverables	Working system

In the span of two months, the application will be developed based on the software requirements gathered through interviews. Interviews were conducted with the staff of the Faculty of Science and Mathematics, Diponegoro University. Within this two months, the expected output is a fully functional application in accordance with the SOP. For details on the SRS of this application, refer to Table 5.

Table 5 SRS List

No	Feature
1	Authentication Feature
2	Letter Submission Form Feature
3	Document Printing Feature
4	System Navigation Feature
5	Applicant Queue Feature
6	Letter List per Officer Access Rights Feature
7	Letter Detail Processing Feature
8	Outgoing Letter Creation Feature
9	Disposition Flow Feature
10	Letter Number Assignment Feature
11	Letter Tracking for Applicants Feature
12	Comment Feature
13	Dashboard Feature

3.4 Empirical Evaluation

To evaluate the effectiveness of the adapted Scrum framework in the context of undergraduate thesis supervision, quantitative data were analysed using descriptive statistical methods. A key instrument employed in this empirical evaluation was the Burndown Chart, which visually represented the remaining workload across the duration of the project. By tracking the progression of completed tasks against planned targets, the Burndown Chart provided a clear overview of sprint performance and enabled the identification of deviations from the expected development pace.

Sprint metrics such as completion rates and velocity trends were further examined to assess consistency and productivity throughout the development lifecycle. The Burndown Chart facilitated a time-based analysis of task completion, highlighting potential bottlenecks and periods of increased progress. In addition, patterns of communication frequency between the student and supervisors were recorded and analysed to evaluate the intensity of supervisory engagement.

Furthermore, project timeline adherence was assessed by comparing actual thesis milestone completion dates against the projected sprint schedule. These quantitative indicators, visualized and supported by the Burndown Chart, provided empirical evidence on how well the Scrum framework maintained progress, structured development efforts, and supported iterative advancement in an academic setting.

4 Results and Discussion

The researcher presents the results obtained from the implementation of the Scrum method in the development of an undergraduate thesis project in Informatics Department.

4.1 Adapted Scrum Framework Implementation

At the initial stage of the implementation, the Product Backlog was developed using user stories, which were then mapped to the required system features. The Product Backlog was aligned with the thesis requirements and constructed by the student, who served in the role of Assistant Product Owner. The first supervisor, acting as the Product Owner, provided guidance and evaluation of the backlog items generated. The Product Backlog was then prioritized and each item was assigned a development status. Prioritization was based on identifying the most critical features that required early verification by stakeholders and the Product Owner. After being grouped into four priority levels because decided to use four sprints in two months, the product backlog was verified by the product owner and deemed ready for inclusion in the sprint backlog. During implementation, the product backlog underwent two revisions due to changes in specifications from the stakeholders at the end of the second sprint. Therefore, adjustments were necessary at the beginning of the third sprint to accommodate these changes. The resulting product backlog example is presented in Table 6.

Table 6 Product Backlog Example

PBI ID	As a/an	I want to…	So that…	Priority	Sprint
1	Student	Input Student Email at the column	Registration information can be sent into my SSO email	Medium	3
2	Academic Supervisor	View letter submission detail	View data requiring approval in more detail	Medium	2
3	Administrative Manager	View submission list	Select which submissions can be approved	High	1
4	Vice Dean	View dashboard	Stay informed on the latest status or letter submissions	Low	4

The division of tasks from each sprint resulted in 25 PBIs carried out in the first Sprint, 30 PBIs carried out in the second Sprint, 60 PBIs carried out in the third Sprint, and 5 PBIs carried out in the fourth Sprint. Some PBIs were delayed due to incomplete preparation. The guidance activities conducted were adjusted according to the activities in the scrum. In the first sprint, activities included sprint planning, daily meetings, sprint review, and sprint retrospective. These activities also produced artifacts tailored to the needs of the thesis. The mapping of the activities carried out in the first sprint can be seen in Table 7.

Table 7 First Sprint Event

ID	Scrum Event	Academic Event	Output	Day
1	Sprint Planning	Supervision meetings	First Sprint Backlog, First Sprint DoDs, First Sprint Subtasks	Sunday first week
2	Daily Meeting	-	-	Tuesday and Thursdays each week
3	Sprint Review	Progress Presentation	-	Friday second week
4	Sprint Retrospective	Progress Presentation	-	Friday second week

To meet academic document requirements, the sprint backlog should be adjusted according to the product owner's directions during sprint planning. Additionally, the DoD document must be tailored to scientific testing methodologies, specifically black box testing. In the Second Sprint, there is a new process which is product backlog refinement because some changes in the product backlog specification. The detail what happen in the Second Sprint can be seen in the Tabel 8.

Scrum events, such as Sprint Planning, Daily Meetings, Sprint Review, and Sprint Retrospective, are implemented up to Sprint 4. Similarly, the artifacts produced also follow each sprint. An example of the sprint backlog generated during Sprint Planning can be referred to in Table 9, which illustrates the Sprint Backlog from the Third Sprint.

Table 8 Second Sprint Event

ID	Scrum Event	Academic Event	Output	Day
1	Sprint Planning	Supervision meetings	Second Sprint Backlog, Second Sprint DoDs, Second Sprint Subtasks	Sunday first week
2	Daily Meeting	-	-	Tuesday and Thursdays each week
3	Sprint Review	Progress Presentation	-	Thursday second week
4	Sprint Retrospective	Progress Presentation	-	Thursday second week
5	Product Backlog Refinement	-	Product Backlog v.2	

Table 9 Sprint Backlog Example from Third Sprint

Tasks	Assigned To	Assigned Hours	Hours Used	available hours remaining	Day 1..10
Disposition button Implementation	Attaf Riski	1	0,5	0,5	0,5
Authorization Validation	Attaf Riski	1	0,5	0,5	0,5
Flow Record	Attaf Riski	1	0,5	0,5	0,5
Movement Notification	Attaf Riski	1	0,5	0,5	0,5

Table 9 Subtasks Example from Third Sprint

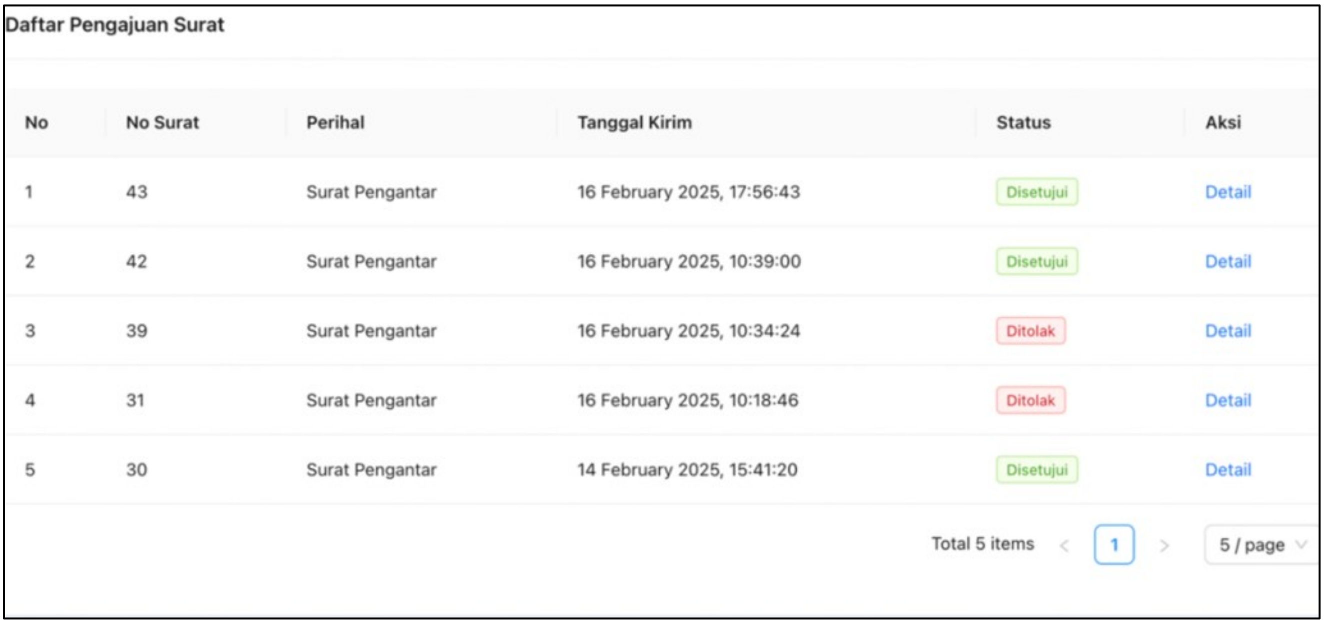
Product Backlog Items	Sub-tasks
Outgoind Letter List Feature	Dataset query
	Filter Implementation
	List rendering
	Pagination
	Privileges validation

Table 10 DoDs Example from Third Sprint

Product Backlog Items	Definition of Done
Outgoind Letter List Feature	Button works
	Filter and sorting works Pagination Works
	UI match the privilege
	Accurate Data

Then an example for Subtasks Third Sprint can be seen in Table 9. Subtasks consist of additional or derivative tasks that can assist developers in completing expected tasks. Each Product Backlog Item may have more than one subtask. Formerly an example for DoDs Third Sprint can be seen in Table 10. DoDs are useful for validating that a feature can be considered an increment if it meets the requirements of the DoDs. Each Product Backlog Item may have more than one DoD. The number of DoDs and Subtasks are sometimes not the same, so they need to be distinguished in their tables.

Subsequently, once a Product Backlog Item (PBI) has been successfully completed, it can be grouped into a collection representing increments. These increments collectively signify progress toward achieving the project's objectives. An example of a completed PBI in the third sprint, which aimed to display a list of submitted letters owned by applicants or students, is illustrated in Figure 4.


Daftar Pengajuan Surat

No	No Surat	Perihal	Tanggal Kirim	Status	Aksi
1	43	Surat Pengantar	16 February 2025, 17:56:43	Disetujui	Detail
2	42	Surat Pengantar	16 February 2025, 10:39:00	Disetujui	Detail
3	39	Surat Pengantar	16 February 2025, 10:34:24	Ditolak	Detail
4	31	Surat Pengantar	16 February 2025, 10:18:46	Ditolak	Detail
5	30	Surat Pengantar	14 February 2025, 15:41:20	Disetujui	Detail

Total 5 items < 1 > 5 / page

Figure 4 Result from some Product Backlog Item in Third Sprint

4.2 Empirical Evaluation

During Sprint I, several features were developed, including the letter submission form, system navigation, letter detail processing, disposition flow, and letter tracking for applicants. The time allocated for Sprint I was 10 working days, with 10 hours allocated per day. The burndown chart for sprint 1 is shown in Figure 5.

The red line declines faster than the blue line, indicating that actual performance was ahead of the planned schedule. This occurred because some tasks had already been completed prior to the official start of Sprint I, thereby accelerating the overall progress. In Sprint II, the features developed included the applicant queue, document printing, and letter listing by access rights. The time allocation remained the same 10 hours per day over 10 days. The resulting burndown chart for Sprint II is presented in Figure 5. Several tasks were not completed as planned during Sprint II, as indicated by the red line that does not reach the bottom threshold in the burndown chart. This shortfall was due to specification changes introduced by the stakeholders and the Scrum Master, causing incomplete tasks to be postponed until the new requirements could be implemented. Sprint III maintained the same allocation of 10 working days with 10 hours per day. Many changes from Sprint II carried over into this sprint. Features addressed during Sprint III included authentication, letter submission form, document printing, system navigation, applicant queue, letter listing by access rights, disposition flow, letter detail processing, and outgoing letter creation. The burndown chart for Sprint III is shown in Figure 5.

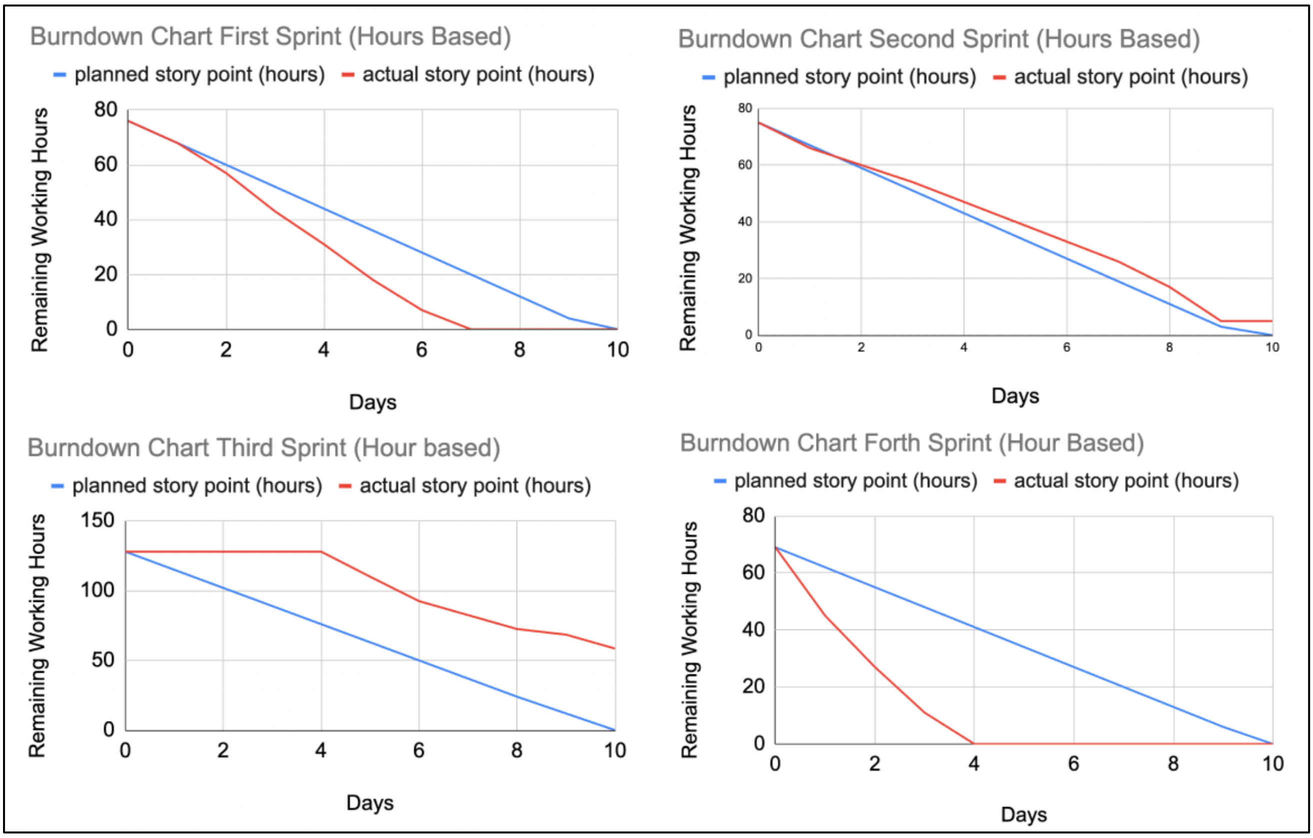


Figure 5 Each Sprint Burndown Chart

The beginning of Sprint III was delayed due to the need to learn new technologies introduced in the previous sprint. The learning process continued until the fifth day. Consequently, some tasks remained incomplete by the end of the sprint, as indicated by the red line remaining significantly above the baseline. Sprint IV was also allocated 10 working days at 10 hours per day. Features developed during this sprint included applicant letter tracking, commenting, and dashboard functionality. The burndown chart for Sprint IV is shown in Figure 5. Tasks in Sprint IV were completed ahead of schedule, as some of the features had been partially developed in prior sprints. As a result, Sprint IV was concluded in just 4 days.

5 Conclusion

This study demonstrates that the Scrum framework, when appropriately adapted, can effectively support the development of individual academic software projects, such as undergraduate final projects. This is evidenced by the actual story point values closely aligning with the planned values, except in the third sprint, which experienced delays due to the developer needing to learn new technologies during the development process. By redefining traditional Scrum roles to fit the academic supervision structure assigning supervisory roles as Product Owner, Scrum Master, and Tester, and the student as Developer the implementation promotes structured communication, iterative progress tracking, and flexibility in accommodating evolving requirements.

Across four sprints conducted over a 40-day period, the development process incorporated 64 product backlogs and a series of iterative adjustments, resulting in the successful delivery of a correspondence application for students at the Faculty of Science and Mathematics, Diponegoro

University. The empirical results, reflected through sprint burndown, indicate that the adapted Scrum approach contributed positively to project management and timely completion.

This adaptation model can serve as a reference for integrating agile methodologies into academic contexts, offering a scalable and pedagogically sound alternative to traditional thesis supervision practices. Future research may further refine this model and explore its application in diverse academic and interdisciplinary project environments.

Acknowledgement

Gratitude is extended to the Faculty of Science and Mathematics, Diponegoro University, for granting permission to conduct this research within the institution.

Bibliography

[1] S. Tenhunen, T. Männistö, M. Luukkainen, and P. Ihantola, "A systematic literature review of capstone courses in software engineering," *Inf Softw Technol*, vol. 159, p. 107191, Jul. 2023, doi: 10.1016/J.INFSOF.2023.107191.

[2] J. Sutherland and K. Schwaber, *The Scrum Paper: Nut, Bolts, and Origin of an Agile Framework*. 2011.

[3] J. G. Schneider, P. W. Eklund, K. Lee, F. Chen, A. Cain, and M. Abdelrazek, "Adopting industry agile practices in large-scale capstone education," *Proceedings - International Conference on Software Engineering*, pp. 119–129, Jun. 2020, doi: 10.1145/3377814.3381715.

[4] V. Mahnic, "Scrum in software engineering courses: An outline of the literature," *Global Journal of Engineering Education*, vol. 17, pp. 77–83, Jan. 2015.

[5] S. D. Zorzo, L. De Ponte, and D. Lucrédio, "Using scrum to teach software engineering: A case study," *Proceedings - Frontiers in Education Conference, FIE*, pp. 455–461, 2013, doi: 10.1109/FIE.2013.6684866.

[6] O. Cico, L. Jaccheri, A. Nguyen-Duc, and H. Zhang, "Exploring the intersection between software industry and Software Engineering education - A systematic mapping of Software Engineering Trends," *Journal of Systems and Software*, vol. 172, p. 110736, Feb. 2021, doi: 10.1016/J.JSS.2020.110736.

[7] N. A. Prasetyo, F. M. Wibowo, R. Adhitama, M. L. Leksono, and A. D. Septiadi, "A Case Study of Using Scrum for Small to Medium Software Development Project E-Voting Student Council Elections," *11th International Conference on ICT for Smart Society: Integrating Data and Artificial Intelligence for a Resilient and Sustainable Future Living, ICISS 2024 - Proceeding*, 2024, doi: 10.1109/ICISS62896.2024.10751566.

[8] D. Borissov, "Agile Method in Social Work with Children and Adolescents Exhibiting Conduct Disorder and Antisocial Behaviour: Case of Kidsköpfe gGmbH," *Health Economics and Management Review*, vol. 5, no. 3, pp. 143–168, Sep. 2024, doi: 10.21272/HEM.2024.3-10.

[9] A. Basri, D. M. U. Atmaja, A. R. Hakim, and A. R. Sanjaya, "Application of Scrum Methodology in The Design of Micro, Small, and Medium Enterprise Systems: A Case Study on Laundry Services," *International Journal Software Engineering and Computer Science (IJSECS)*, vol. 3, no. 3, pp. 537–544, Dec. 2023, doi: 10.35870/IJSECS.V3I3.1968.

[10] D. Tokenov, “Lessons Learned from Implementing Scrum in a Kazakhstani IT Firm: A Case Study at ‘AG TECH’ LLP,” *Authorea Preprints*, Aug. 2023, doi: 10.22541/AU.169221787.70225854/V1.

[11] E. Caballero and J. A. Calvo-Manzano, “A Practical Approach to Project Management in a Very Small Company,” *Communications in Computer and Information Science*, vol. 301 CCIS, pp. 319–329, 2012, doi: 10.1007/978-3-642-31199-4_28.

[12] B. L. Romano and A. D. Da Silva, “Project management using the scrum agile method: A case study within a small enterprise,” *Proceedings - 12th International Conference on Information Technology: New Generations, ITNG 2015*, pp. 774–776, May 2015, doi: 10.1109/ITNG.2015.139.

[13] A. Azanha, A. R. T. T. Argoud, J. B. de Camargo Junior, and P. D. Antoniolli, “Agile project management with Scrum: A case study of a Brazilian pharmaceutical company IT project,” *International Journal of Managing Projects in Business*, vol. 10, no. 1, pp. 121–142, 2017, doi: 10.1108/IJMPB-06-2016-0054/FULL/XML.

[14] N. A. Prasetyo, F. M. Wibowo, R. Adhitama, M. L. Leksono, and A. D. Septiadi, “A Case Study of Using Scrum for Small to Medium Software Development Project E-Voting Student Council Elections,” *11th International Conference on ICT for Smart Society: Integrating Data and Artificial Intelligence for a Resilient and Sustainable Future Living, ICISS 2024 - Proceeding*, 2024, doi: 10.1109/ICISS62896.2024.10751566.

[15] S. Lee and H. S. Yong, “Agile Software Development Framework in a Small Project Environment,” *Journal of Information Processing Systems*, vol. 9, no. 1, pp. 69–88, 2013, doi: 10.3745/JIPS.2013.9.1.069.

[16] J. Sutherland, I. Jacobson, and B. Kerr, “Scrum Essentials Cards,” *Queue*, vol. 18, no. 3, Jun. 2020, doi: 10.1145/3411757.3418775.