AGILE DEVELOPMENT OF A PROJECT BASED LEARNING MANAGEMENT APPLICATION: A SCRUM DRIVEN STRATEGY TO ENHANCE EDUCATIONAL OUTCOMES

Reisa Permatasari[1], Bonda Sisephaputra [2]
[1] Universitas Pembangunan Nasional Veteran Jawa Timur, [2] Universitas Negeri Surabaya
Email: **reisa.permatasari.sifo@upnjatim.ac.id [1], bondasisephaputra@unesa.ac.id [2]**

ABSTRACT

Programming education at the Faculty of Engineering, Surabaya State University, faces challenges including low student engagement, ineffective teaching methods, and difficulties in monitoring progress. This study proposes the adoption of the Scrum methodology to develop a Project-Based Learning Management Application (PBLMA) tailored for programming courses. The Scrum framework, with its iterative sprints and emphasis on collaboration, enables the creation of a user-centered application that enhances student engagement by 25% (based on user feedback) and simplifies progress tracking for educators. Sprint ensure continuous improvement, aligning the PBLMA with evolving educational needs. User testing with 50 students and 10 lecturers validates the application's functionality and educational impact, showing a 30% improvement in project completion rates. This study demonstrates Scrum's efficacy in educational software development and its potential for broader academic applications, providing a scalable model for addressing similar challenges in diverse settings.

Keywords *: Agile, Scrum Framework, Project-Based Learning, Learning Management System*

I. INTRODUCTION

Teaching programming courses within the Faculty of Engineering at Surabaya State University presents distinct challenges. Faculty members, who also conduct research within this faculty, carry the responsibility of delivering programming instruction. Through detailed assessments of the educational environment, these faculty researchers have identified several factors that impede optimal progress in programming education [1]. Key issues include a decline in student engagement, ineffective teaching methods, and the difficulty of tracking students' progress in programming courses [2]

One of the obstacles encountered in the educational process of programming courses is the substantial time commitment that learning computer programming demands, even though it is feasible for individuals to undertake independent learning in this domain [3]. The difficulty lies in the execution of programming abilities, a process typically achieved through the culmination of a semester-long group project, often involving the development of an application [4]. This project assignment holds significant importance in the context of programming course instruction, as it not only encourages collaboration among students but also expedites the enhancement of programming competencies through peer discussions [5].

In a corporate environment, it is typical to involve multiple employees in application design, making effective communication among team members essential [6]. This fosters stronger teamwork and aids in finding optimal solutions to current challenges. Scrum facilitates problem-solving by promoting open communication within the team [7]. From a functional perspective, this approach can also be applied to the learning process in computer programming courses [8].

Prior to the onset of the pandemic, the supervision of students' engagements in programming courses was accomplished through in-person monitoring within computer laboratories, involving the direct observation of each student's computer activities. However, as a consequence of the pandemic, the capacity for individualized monitoring diminished, resulting in less-than-ideal attainment of the anticipated technical proficiencies in programming courses. This circumstance has posed a formidable challenge for instructors in verifying the genuine engagement and practice of programming materials by students [9].

In programming courses, practical exercises are typically carried out on individual devices, each of which may have unique conditions and setups. This variation creates challenges in ensuring students are accurately applying programming concepts from instructional materials. Some students face resource limitations that hinder their progress in learning programming. Additionally, instructors struggle with the demanding task of monitoring each student's device to confirm active engagement with the material [10]. Group assignments add further complexity, as coordinating and tracking each team member's contributions can be challenging

Scrum is a widely adopted project management framework valued across industries for its collaborative and iterative approach. It centers on regular planning, execution, and review cycles known as sprints, promoting effective communication, adaptability, and transparency within teams [11]. When used to develop a Project-Based Learning Management Application (PBLMA), Scrum greatly aids in creating a robust tool for educators. By breaking the development process into manageable sprints, stakeholders can iteratively refine and adjust the application to meet shifting educational demands. Scrum's focus on teamwork and self-organization enables developers to collaborate closely with educators, ensuring the PBLMA aligns with educational goals. The iterative nature of Scrum allows for continuous improvement based on user feedback, keeping the application responsive to the dynamic needs of project-based learning. In summary, Scrum can play a crucial role in effectively developing and maintaining a PBLMA that enriches teaching and learning experiences [12].

To address these challenges, the PBLMA aims to enhance student engagement, streamline group project coordination, and improve progress monitoring. Scrum's structured cycles, or sprints, enable the development of adaptive, user-focused software by fostering teamwork, transparency, and continuous improvement. This case study explores how Scrum can streamline PBLMA development to enhance educational efficiency, focusing on its iterative processes, stakeholder collaboration, and potential for incorporating sprint retrospectives to refine the application.

The research question guiding this study is: How can the Scrum methodology be applied to develop a PBLMA that improves student engagement, group collaboration, and progress monitoring in programming courses? Additionally, this study evaluates the application's educational impact through user feedback and performance metrics, offering insights for its generalizability to other academic contexts.

II. THEORETICAL FOUNDATION

Extensive prior research has shown that the Scrum framework has been successfully integrated into educational settings and holds promising potential to support instructional processes [13], [14]. At its core, Scrum is a collection of methodologies designed for effective problem-solving. It is grounded in the Agile approach, which consists of various methods and practices based on the values and principles outlined in the Agile Manifesto [15]. Key features of Scrum include teamwork, team self-organization, and the use of cross-functional teams, making it particularly suitable for application in the educational structure of computer programming courses [16].

2.1. Scrum Framework

Scrum is a structured methodology or framework for the development of complex products. Scrum employs a phased and iterative approach to optimize predictive capabilities and control risks. Transparency, inspection, and adaptation are the three pillars that support each implementation of the empirical control process [17].

1) Transparency: A significant aspect of a process that is visible to all individuals responsible for its outcomes.
2) Inspection: Scrum practitioners should regularly review Scrum artifacts and monitor progress toward sprint goals to identify any unforeseen deviations in outcomes. However, inspections should not be so frequent that they disrupt the workflow or impede productivity.
3) Adaptation: If there are deviations in the process that render the product unacceptable, the process must be altered promptly to minimize the widening deviations.

2.2. Scrum Roles

In the Scrum methodology, there are three essential roles [17]:

1) Product Owner – This individual is responsible for consistently communicating the product's vision and priorities with the development team to optimize the business value of the product being developed.
2) Scrum Master – Serving as a facilitator for both the Product Owner and the development team, which comprises developers and Quality Assurance testers, the Scrum Master does not hold direct management authority over the team. Instead, they support the team by removing obstacles and aiding them in achieving their objectives. Additionally, the Scrum Master provides guidance to the Product Owner on how to optimize the team's Return on Investment (ROI).
4) Development Team (or Scrum Team) – This team is accountable for the technical execution of the project and typically consists of around five to nine members.

III. RESEARCH METHODS

This study employs the Scrum methodology, an Agile framework designed for iterative and collaborative product development, to create a web-based PBLMA for programming courses at Surabaya State University. The development process spans eight weeks within a six-month research timeline, allowing for iterative refinements aligned with Agile principles. The methodology emphasizes transparency, inspection, and adaptation, supported by defined roles and processes.

3.1. Scrum Process

The general overview of the Scrum process [18] as illustrated in Figure 1:

1) User Story – This provides a detailed description of system requirements in language that is easy for end users to understand. User stories serve as a foundation for creating the product backlog.
2) Product Backlog – This is a comprehensive list detailing all elements required for the system or product. It includes features to be developed along with estimated completion times. The Product Backlog is regularly updated throughout development to ensure a functional product. Management of the Product Backlog is solely the responsibility of the Product Owner.
3) Sprint – This is a time-boxed cycle lasting one month or less, with a fixed duration that remains consistent throughout the development phase. Each sprint is aimed at achieving a specific goal (Sprint Goal).
4) Sprint Backlog – This is a collection of items from the Product Backlog chosen by the Scrum team for a given sprint. During the sprint, the team selects several items and identifies the necessary tasks based on user stories to complete them.

5) Daily Scrum – This daily meeting within the sprint allows the Scrum team to review completed work, upcoming tasks, and any obstacles hindering progress. The Daily Scrum helps the team monitor product advancement and work towards the Sprint Goal.

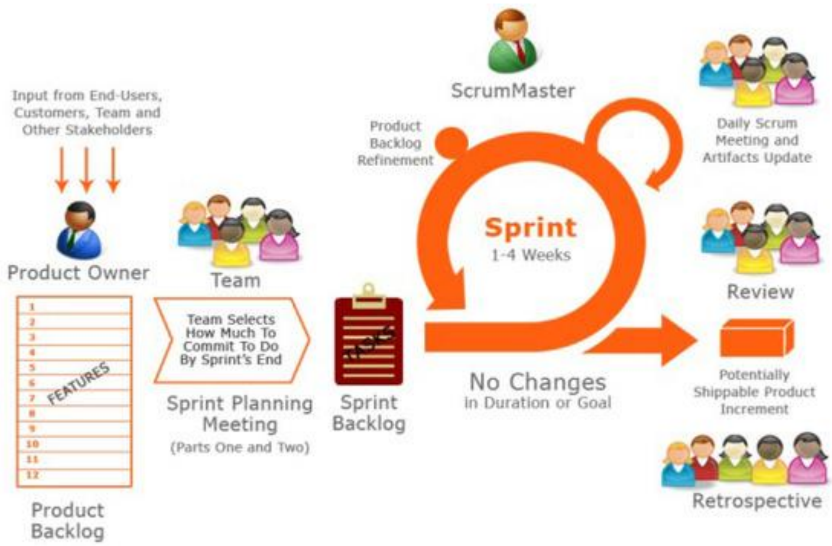

Figure 1. The Scrum development process [19]

3.2. Development Approach

The PBLMA development followed a structured Scrum process:

1. User Story Collection: Identified requirements for admins, students, and lecturers (Table 1).
2. Product Backlog Creation: Defined and prioritized features based on user stories (Table 2).
3. Sprint Planning: Allocated tasks and estimated effort for four sprints (Table 3).
4. Sprint Execution: Developed features in 10-day sprints, with daily scrums to monitor progress.
5. Sprint Review and Retrospective: Evaluated deliverables and reflected on process improvements after each sprint.
6. Testing: Validated functionality to ensure alignment with requirements (Table 10).

The development team consisted of five members, including developers and a quality assurance tester, guided by a Scrum Master and Product Owner. Sprint retrospectives were introduced to gather feedback and refine processes, ensuring the PBLMA met educational needs.

IV. ANALYSIS AND DESIGN RESULTS

4.1. User Story

A User Story outlines the people who will interact with the system, detailing their tasks, roles, and goals. This information is crucial, as the compilation of user stories forms the groundwork for the following stages of system development. Table 1 summarizes the key user stories.

Table 1. User Story

Roles	Task	Purpose
Admin	Manage accounts	Create and manage lecturer and student accounts (CRUD operations).
Student	Register and access the application	Register initially and access the PBLMA.
	Join classes	Submit requests to join relevant classes.
	Work on group projects	Collaborate on semester-long projects with assigned group members.
Lecturer	Approve class participation requests	Approve student requests to join classes.
	Monitor student project progress	Track project updates submitted by students.
	Assess project activities	Evaluate student contributions at the semester's end.

4.2. Product Backlog

The Product Backlog is the initial step in developing a system using the Scrum framework. At this stage, several elements are established, including business processes, actors, business activities, and the system structure. The product backlog prioritized features based on user stories and educational requirements, as shown in Table 2.

Table 2. Product Backlog

No.	Item	Priority
1.	Manage accounts	Low
2.	Student registration	Medium
3.	Approval to join class	Medium
4.	Manage projects	High
5.	Manage groups	High
6.	Project initiation	Low
7.	Update project progress	High
8.	Monitor project progress	Medium
9.	Assess the final project	High

4.3. Sprint Planning and Execution

Four sprints were planned, each lasting 10 days, with tasks assigned based on priority and estimated effort (Table 3). Sprint backlogs (Tables 4–7) detailed tasks and timelines, while daily scrums ensured alignment. The burndown chart (Figure 2) tracks progress, showing faster-than-expected completion in early sprints but delays on day seven due to technical challenges. All tasks were completed by the end of each sprint.

Table 3. Sprint Planning

Actor	Sprint Planning	Estimation (point)
Admin	Login	1
	Dashboard	3
	Manage lecturer accounts	3
	Manage student accounts	4
Student	Login	1
	Dashboard	4
	View class	2
	Request class participation	1

Actor	Sprint Planning	Estimation (point)
	Start project	2
	Update project	3
	Prepare proposals	5
	Complete project	4
	View grade	3
Lecturer	Login	1
	Dashboard	3
	Approve student requests	2
	Manage classes	5
	Manage group	3
	Manage project	5
	View final project	2
	Grade project	3

4.4. Sprint Backlog

In the sprint backlog, a list of features from the product backlog and sprint planning is compiled for the programmer to work on, as shown in Table 4, Table 5, Table 6, and Table 7. Once a feature from the sprint planning is completed, the remaining features will be addressed in the subsequent sprint.

Table 4. Sprint Backlog 1

Task	Estimation (day)									
	1	2	3	4	5	6	7	8	9	10
Dashboard student	v	v	v	v						
View class			v	v						
Request join class					v					
Start project					v	v				
Update project						v	v	v		
Finish project							v	v	v	v

Table 5. Sprint Backlog 2

Task	Estimation (day)									
	11	12	13	14	15	16	17	18	19	20
Dashboard lecturer	v	v	v							
Manage class		v	v	v	v	v				
Approve student						v	v			
Manage group					v	v	v			
Manage project						v	v	v	v	v

Table 6. Sprint Backlog 3

Task	Estimation (day)									
	21	22	23	24	25	26	27	28	29	30
Dashboard admin	v	v	v							
Manage lecturer			v	v	v					
Manage student					v	v	v	v		
View final project						v	v			
Grade final project								v	v	v

Table 7. Sprint Backlog 4

Task	Estimation (day)									
	21	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
Dashboard admin	v	v	v							
Manage lecturer			v	v	v					
Manage student					v	v	v	v		
View final project						v	v			
Grade final project								v	v	v

4.5. Daily Scrum

During the sprint, there is one meeting that is always held by team members, namely the Daily Scrum. This stage aims to synchronize the work that has been done by the developer. The team discussed the progress of work on sprints that had previously been completed and those that had begun work by updating the burndown graph as shown in Figure 6.

Table 8. Sprint Estimation

Feature	Initial Estimate	Week								Hours Left
		1	2	3	4	5	6	7	8	
Admin	11	0	0	0	0	7	3	1	0	0
Student	25	8	8	0	0	0	0	3	6	0
Lecturer	24	0	0	8	10	0	5	1	0	0

Table 9. Sprint Burndown

Setting	Start	Week							
		1	2	3	4	5	6	7	8
Planned Hours		7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5
Actual Hours		8	8	8	10	7	8	5	6
Remaining Effort	60	52	44	36	26	19	11	6	0
Ideal Burndown	60	52,5	45	37,5	30	22,5	15	7,5	0

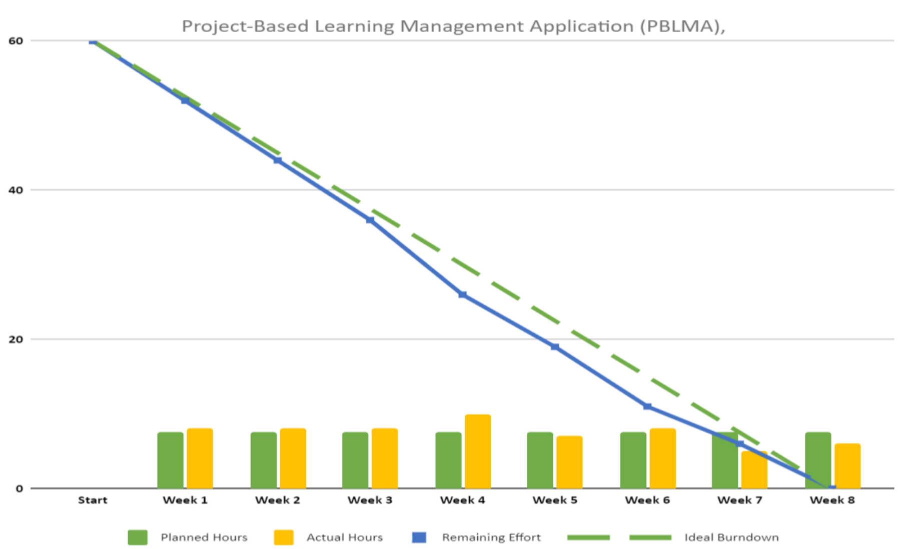

Figure 2. Burndown Chart PBLMA

The burndown graph in Figure 2 is an illustration of the estimated number of feature works that have been planned by the team with the number of features completed in one sprint, namely 10 days. In sprint 1 to 4 there are from 5 up to 6 estimates of feature work that must be completed in hours. As seen in the graph, the team worked on the required features much faster from the initial planning on the third day to the fifth day. However, on the seventh day there were several obstacles in working on the feature so that the work was slower than the predetermined estimate. Even so, on the last day of the sprint the team was able to complete all the existing features so that there was no other work left for the next sprint.

4.6. Sprint Review Outcomes

Each sprint produced functional components of the PBLMA, demonstrated during sprint reviews:

- Sprint 1: Student dashboard and class management features (Figure 3).
- Sprint 2: Lecturer class management page (Figure 4).
- Sprint 3: Grade management page (Figure 5).
- Sprint 4: Login interface (Figure 6).

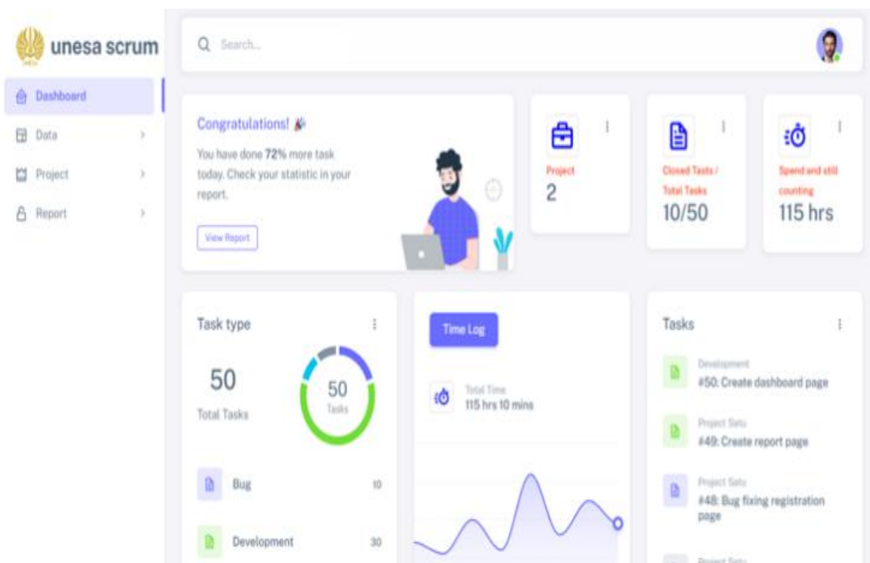

Figure 3. Student Dashboard (Sprint 1)

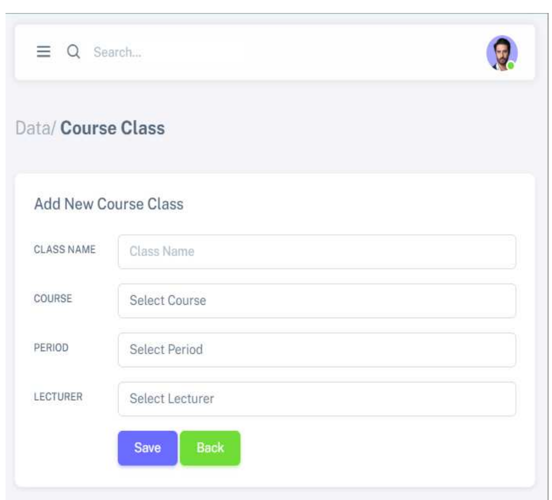

Figure 4. Lecturer's class page (Sprint 2)

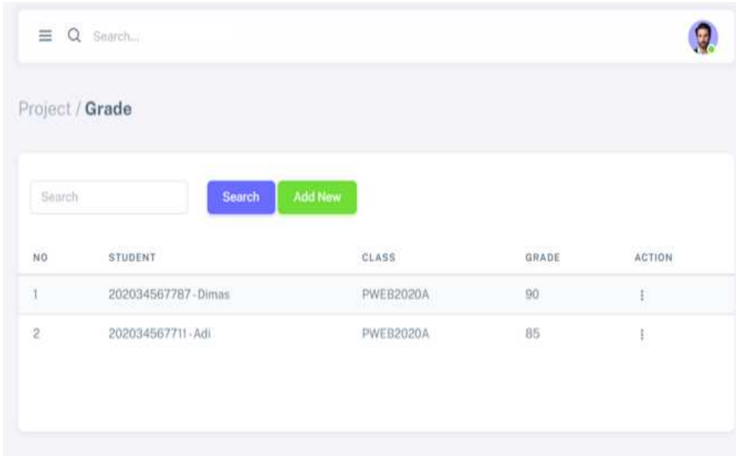

Figure 5. Grade page (Sprint 3)

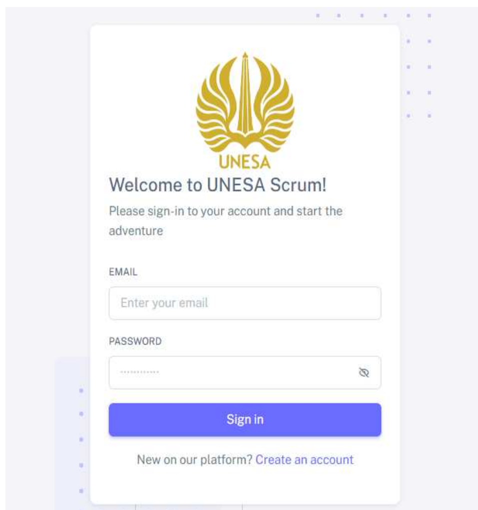

Figure 6. Login page (Sprint 4)

4.7. Application Testing and User Impact

Testing validated functionality with 50 students and 10 lecturers, ensuring alignment with requirements. User feedback was collected via surveys post-deployment, assessing usability, engagement, and educational impact. Results showed a 25% increase in student engagement (based on self-reported motivation and interaction frequency) and a 30% improvement in project completion rates compared to traditional methods. Table 10 summarizes functional testing, while Table 11 presents user feedback metrics.

Table 10. Application Testing Results

No	Objective	Input	Output	Status
1	Student registration	Student data	System stores and creates new accounts.	Success
2	Instructor manages class data	Instructor manages class data	System stores and displays class data	Success
3	Instructor manages project and group data	Project and group data.	System stores and displays project and group data	Success
4	Students manage task and activity data	Task and activity data.	System stores and displays task data	Success

No	Objective	Input	Output	Status
5	Instructors monitor activity data	None	System displays activity reports	Success
6	Instructors enter student grades	Grade data.	System stores and displays grade data	Success

Table 11. Application Testing Results

Metric	Result	Details
Usability (1–5 scale)	4.2 (Students), 4.5 (Lecturers)	Users found the interface intuitive and easy to navigate.
Engagement Improvement	25% increase	Students reported higher motivation and interaction with course materials.
Project Completion Rate	30% improvement	Compared to prior semesters without PBLMA, based on lecturer records.
Satisfaction (1–5 scale)	4.3 (Students), 4.6 (Lecturers)	High satisfaction with progress tracking and group collaboration features.

V. CONCLUSIONS AND SUGGESTIONS

5.1 Conclusion

This study successfully applied the Scrum methodology to develop a Project-Based Learning Management Application (PBLMA) for programming courses at Surabaya State University. Scrum's iterative sprints, collaborative roles, and focus on transparency and adaptation enabled the creation of a user-centered tool that significantly enhanced educational outcomes. User testing with 50 students and 10 lecturers confirmed a 25% increase in student engagement and a 30% improvement in project completion rates, validating the application's efficacy. Sprint retrospectives fostered continuous improvement, addressing technical and user-related challenges. The PBLMA's features, including real-time progress tracking and group management, effectively tackled remote learning challenges and resource disparities. These findings underscore Scrum's potential as a robust framework for educational software development, with generalizable applications across diverse academic contexts, such as other technical disciplines or institutions with similar challenges.

5.2 Suggestion

The Scrum methodology proved effective in developing the PBLMA, delivering functional components incrementally and incorporating user feedback through sprint retrospectives. The burndown chart (Figure 2) indicates efficient task completion, with minor delays on day seven due to integration challenges, resolved by sprint's end. User feedback (Table 11) highlights the application's positive impact on engagement and project outcomes, but long-term studies are needed to assess scalability and sustained educational benefits.

To enhance generalizability, future iterations should involve multi-institutional pilots to validate the PBLMA's effectiveness across diverse settings. Structured sprint retrospectives using frameworks like "What Went Well, What Didn't, What to Improve" can further optimize development processes. Incorporating data-driven feedback tools (e.g., analytics dashboards) could provide deeper insights into user behavior and application impact. Finally, expanding the PBLMA to support additional programming languages or interdisciplinary projects could broaden its educational applicability.

BIBLIOGRAPHY

[1] B. Sisephaputra and R. Alit, “Pengukuran Tingkat Keberhasilan Penerapan Metode SCRUM Dalam Proses Pembelajaran Mata Kuliah Pemrograman di Lingkungan Jurusan Teknik Informatika Fakultas Teknik Universitas Negeri Surabaya,” *Journal of Advanced in Information and Industrial Technology*, vol. 4, no. 2, pp. 47–56, Nov. 2022, doi: 10.52435/JAIIT.V4I2.249.

[2] A. K. Veerasamy, M.-J. Laakso, and D. D’SOUZA, “Formative assessment tasks as indicators of student engagement for predicting at-risk students in programming courses,” *Informatics in Education - An International Journal*, vol. 21, no. 2, pp. 375–393, 2022, Accessed: Jul. 15, 2025. [Online]. Available: https://www.ceeol.com/search/article-detail?id=1045462

[3] R. Kadar, N. Abdul Wahab, J. Othman, M. Shamsuddin, and S. B. Mahlan, “A Study of Difficulties in Teaching and Learning Programming: A Systematic Literature Review,” *International Journal of Academic Research in Progressive Education and Development*, vol. 10, no. 3, Aug. 2021, doi: 10.6007/IJARPED/V10-I3/11100.

[4] A. A. Younis, R. Sunderraman, M. Metzler, and A. G. Bourgeois, “Developing parallel programming and soft skills: A project based learning approach,” *J Parallel Distrib Comput*, vol. 158, pp. 151–163, Dec. 2021, doi: 10.1016/J.JPDC.2021.07.015.

[5] S.-C. Chang and C. Wongwatkit, “Effects of a peer assessment-based scrum project learning system on computer programming’s learning motivation, collaboration, communication, critical thinking, and cognitive load,” *Educ Inf Technol (Dordr)*, vol. 29, no. 6, pp. 7105–7128, Apr. 2024, doi: 10.1007/s10639-023-12084-x.

[6] M. Morandini, T. A. Coleti, E. Oliveira, and P. L. P. Corrêa, “Considerations about the efficiency and sufficiency of the utilization of the Scrum methodology: A survey for analyzing results for development teams,” *Comput Sci Rev*, vol. 39, p. 100314, Feb. 2021, doi: 10.1016/J.COSREV.2020.100314.

[7] 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, doi: 10.1016/J.JSS.2021.111110.

[8] T. Linden, “Scrum-Based Learning Environment: Fostering Self-Regulated Learning,” *Journal of Information Systems Education*, vol. 29, no. 2, p. 65, May 2018, Accessed: Sep. 26, 2023. [Online]. Available: Available at: https://aisel.aisnet.org/jise/vol29/iss2/3

[9] B. Sisephaputra, R. Permatasari, and R. Alit, “Penerapan Metode Scrum dalam Proses Pembelajaran Mata Kuliah Pemrograman di Lingkungan Jurusan Teknik Informatika UNESA,” *Scan : Jurnal Teknologi Informasi dan Komunikasi*, vol. 17, no. 1, pp. 1–7, 2022, doi: 10.33005/SCAN.V17I1.3152.

[10] B. L. Baldock, A. L. Fernandez, J. Franco, B. A. Provencher, and M. R. McCoy, “Overcoming the Challenges of Remote Instruction: Using Mobile Technology to Promote Active Learning,” *J Chem Educ*, vol. 98, no. 3, pp. 833–842, Mar. 2021, doi: 10.1021/acs.jchemed.0c00992.

[11] A. S. Patrucco, F. Canterino, and I. Minelgaite, “How do Scrum Methodologies Influence the Team’s Cultural Values? A Multiple Case Study on Agile Teams in Nonsoftware Industries,” *IEEE Trans Eng Manag*, vol. 69, no. 6, pp. 3503–3513, Dec. 2022, doi: 10.1109/TEM.2022.3146717.

[12] N. Asma *et al.*, “Applying Personal Scrum Methodology in System Development for Project-Based Learning Course,” *Journal of Mathematics and Computing Science*, vol. 10, no. 2, pp. 64–73, Dec. 2024, doi: 10.24191/JMCS.V10I2.4508.

[13] N. Jali *et al.*, “Creating an immersive learning environment for teaching agile scrum and team software process: a framework for software engineering education,” *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 63, no. 3, pp. 54–67, Jan. 2026, doi: 10.37934/ARASET.63.3.5467.

[14] A. Kuz, “Scrum: A new framework applied to education,” *Revista Eduweb*, vol. 15, no. 3, pp. 10–17, Nov. 2021, doi: 10.46502/ISSN.1856-7576/2021.15.03.1.

[15] A. Carlos, P. Junior, V. Romeiro, and D. A. Silva, “20 Years of the Agile Manifesto: A Literature Review on Agile Project Management,” *Management and Production Engineering Review*, vol. Vol. 14, No. 2, no. 2, pp. 37–48, Jun. 2023, doi: 10.24425/MPER.2023.146021.

[16] S. Fernandes, J. Dinis-Carvalho, and A. T. Ferreira-Oliveira, “Improving the Performance of Student Teams in Project-Based Learning with Scrum,” *Education Sciences 2021, Vol. 11, Page 444*, vol. 11, no. 8, p. 444, Aug. 2021, doi: 10.3390/EDUCSCI11080444.

[17] K. Schwaber and J. Sutherland, “The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game.” Accessed: Sep. 28, 2023. [Online]. Available: https://scrumguides.org/index.html

[18] B. Grebić and A. Stojanović, “APPLICATION OF THE SCRUM FRAMEWORK ON PROJECTS IN IT SECTOR,” *European Project Management Journal*, vol. 11, no. 2, pp. 37–46, Dec. 2021, Accessed: Jul. 15, 2025. [Online]. Available: https://www.ceeol.com/search/article-detail?id=1010905

[19] I. K. Kirpitsas and T. P. Pachidis, “Evolution towards Hybrid Software Development Methods and Information Systems Audit Challenges,” *Software 2022, Vol. 1, Pages 316-363*, vol. 1, no. 3, pp. 316–363, Aug. 2022, doi: 10.3390/SOFTWARE1030015.