

JUVOTEK JURNAL
Jurnal Vokasi Teknik | *J. Vok. Tek.*
https://mentech.id/jurnal/index.php/juvotek | e-ISSN 3025-1125
DOI: https://doi.org/10.53905/9j80td53
Original Research Article

Design and Development of a Web-Based Capstone Project Monitoring Application Using an Agile Approach

Gusti Alta Pradana[1], Ajid Haqi Rahmani[1], Shofi Zulfatur Riski[1], Muhammad Rian[*1], Adriansyah Amir[1], Achmad Affandi[1]

[1]Diploma Program in Informatics Engineering, Politeknik Negeri Samarinda, Samarinda, Indonesia.

Received: january 21, 2026| Accpeted: March 28, 2026 | Published: April 01, 2026

ABSTRACT

This study aimed to design and develop a web-based application for monitoring capstone projects in higher education, with particular emphasis on progress visibility, supervision coordination, feedback exchange, and administrative control. The study adopted a design-and-development approach. Requirements were elicited through interviews with relevant lecturers, modeled using a context diagram, data-flow diagram, activity and use-case representations, sequence diagrams, and an entity-relationship diagram, and implemented using PHP/Laravel and MySQL. Development followed an Agile-oriented iterative process. Functional verification was conducted using black-box testing against the documented user requirements. The resulting prototype supported three principal user roles: students, supervisors, and administrators. Core functions included topic submission, milestone and deadline tracking, progress reporting, automated notifications, supervision scheduling, supervisor feedback, account management, and system reporting. The source project reported that the implemented functions operated in accordance with the specified requirements. The system architecture also established traceable information flows among the three user groups and a relational data structure for capstone records. The prototype demonstrates the feasibility of integrating capstone monitoring, academic supervision, and administrative oversight within a single web platform. However, because the available evidence is limited to requirements analysis and functional testing, future studies should evaluate usability, security, reliability, and educational impact with larger user samples and validated instruments.

Keywords:
capstone project; web-based monitoring; academic supervision; Agile development; Laravel; higher education information system.

Article Info

Check for updates

EDITED BY
Dr. Samsurijal Hasan
Universitas Pahlawan, Indonesia.

COPYRIGHT©2026 (Author)
Gusti Alta Pradana
Ajid Haqi Rahmani
Shofi Zulfatur Riski
Muhammad Rian
Adriansyah Amir
Achmad Affandi

***CORRESPONDENCE**
Muhammad Rian
muhammadrianr651@gmail.com

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

To cite this article:
Pradana, G. A., Rahmani, A. H., Riski, S. Z., Rian, M., Amir, A., & Affandi, A. (2026). Design and Development of a Web-Based Capstone Project Monitoring Application Using an Agile Approach. Jurnal Vokasi Teknik (JuVoTek), 4(01), 18-24. https://doi.org/10.53905/9j80td53

INTRODUCTION

Contextual Framework and Global Issue

Capstone projects are widely used in higher education as culminating learning experiences through which students integrate disciplinary knowledge, technical skills, teamwork, communication, and problem-solving in authentic or near-authentic tasks(Rahman et al., 2025; Talmi et al., 2018, p. 47). Their educational value depends not only on the final product but also on the quality of project management, supervision, feedback, documentation, and assessment throughout the project life cycle (Ghonim & Eweda, 2018; Marel et al., 2022)

Despite their importance, capstone projects are frequently managed through fragmented communication channels, manually maintained files, and episodic face-to-face meetings(Farr et al., 2001; Ferrão & Soares, 2026). Such practices can limit real-time visibility, delay feedback, obscure milestone status, and make it difficult for coordinators to identify at-risk projects early(Abayomi et al., 2024; Persson et al., 2009, p. 515). These challenges have become more pronounced as higher education increasingly adopts blended, distributed, and technology-mediated learning environments.

Web-based monitoring systems offer a practical response because they can centralize records, standardize workflows, and provide role-specific access independent of location and device(Cristóbal et al., 2017, p. 147; Nasir & Wahab, 2024, p. 7007). In an educational context, a well-designed monitoring system can support transparency, accountability, timely intervention, and continuity of supervision while preserving an auditable project history.

Critical Examination of Existing Literature

Previous research consistently frames capstone and graduation projects as integrative learning

environments, but also highlights supervision as a complex professional practice. A literature review by Marel et al. (2022) found that effective graduation-project supervision requires a combination of disciplinary expertise, research guidance, process management, and pedagogical judgment(Sá et al., 2021, p. 158). Ghonim & Eweda (2018) similarly emphasized that graduation projects require systematic management, regular supervision, and transparent assessment practices(Yılmaz et al., 2018, p. 15).

Project-based learning research also indicates that authentic projects can strengthen the application of project-management knowledge and support both technical and interpersonal skill development(Ngereja et al., 2020). Marnewick (2023) showed that Agile-oriented project-based learning helps students experience iterative work, collaborative planning, and practical project-management processes. More broadly, Agile approaches have been applied in education to promote collaboration, adaptive planning, continuous feedback, and responsibility for incremental outcomes (López-Alcarria et al., 2019).

However, the use of Agile principles does not automatically ensure effective governance. Software projects require explicit requirements, adequate documentation, user involvement, and continuous validation. Empirical reviews of Agile software development emphasize the importance of iterative delivery and stakeholder collaboration while also identifying risks related to communication, documentation, and changing requirements (Bianchi et al., 2018, p. 552; Burkin, 2023, p. 14; Dybå & Dingsøyr, 2008). These considerations are especially relevant when developing systems for academic administration, where data integrity, access control, and process consistency are critical.

Research Gap, Rationale, and Objectives

The literature provides strong conceptual support for capstone supervision, project-based learning, and Agile development, yet the source project did not identify an integrated institutional application that simultaneously addressed student progress tracking, supervisor feedback, supervision scheduling, topic selection, notifications, and administrative monitoring in its local context. In addition, the original project report presented the system primarily as a course paper and did not organize its contribution according to an international scientific reporting structure.

Accordingly, this study reorganizes and strengthens the original development report as a scientific design-and-development article. The study objectives were to: (1) identify the functional and role-based requirements of a web-based capstone monitoring application; (2) design the system architecture, data flows, and database relationships; (3) implement the prototype using web technologies and an Agile-oriented process; and (4) verify the functional conformity of the application through black-box testing.

MATERIALS AND METHODS

Study Design and Setting

This study used a design-and-development research approach focused on producing and functionally verifying a web-based capstone monitoring prototype. The development context was the Diploma Program in Informatics Engineering, Department of Information Technology, Politeknik Negeri Samarinda. The unit of development was the capstone-project workflow involving students, supervisors, and administrators.

Requirements Elicitation

Requirements were collected through interviews with relevant lecturers concerning competency clusters, research topics, supervision needs, and monitoring procedures for students' final projects. The original report does not provide the number, demographic characteristics, or interview duration of participants; therefore, no unsupported participant statistics are introduced in this article. Interview findings were translated into functional requirements and user-role specifications.

System Analysis and Design

A structured systems-analysis approach was used. The design artifacts included a flowchart, context diagram, data-flow diagram, use-case and activity diagrams, sequence diagrams, and an entity-relationship diagram. These artifacts were used to represent external actors, data movement, application activities, user interactions, process sequences, and database relationships. Figure 1 reconstructs the principal information flows documented in the original context diagram.

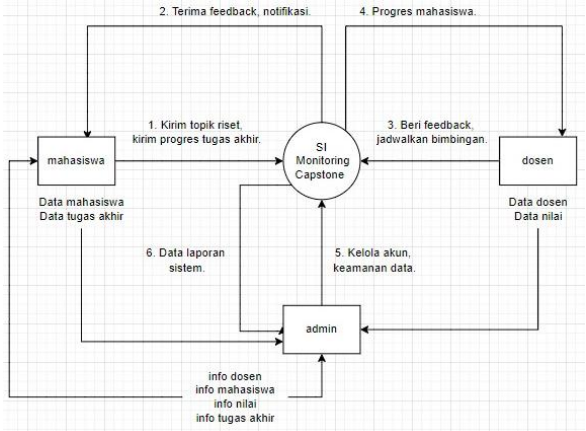


Figure 1. Context diagram of the web-based capstone monitoring application. *Source: Original system-development documentation (2024).*

Figure 1 presents the high-level exchange of information among students, supervisors, administrators, and the monitoring system. Students submit research topics and progress records; supervisors review progress, provide feedback, and schedule supervision; and administrators manage accounts, security, and system reports.

Development Process and Technology

Development followed an Agile-oriented iterative process consisting of requirements analysis, system design, implementation, functional review, and

refinement. The application was implemented as a web system using PHP with the Laravel framework and a MySQL relational database. Visual Studio Code was used as the code editor, and a local web-server environment was used during development. The iterative approach was selected to allow requirements and interface components to be refined in response to stakeholder feedback.

Functional Testing and Data Analysis

Black-box testing was used to verify whether externally observable system functions conformed to the defined requirements. Testing focused on user inputs, expected system responses, and role-specific outputs rather than internal source-code structure. The available source report states that all implemented features operated according to user needs, but it does not provide raw test logs, numbers of test cases, failure counts, response-time data, or usability scores. Consequently, the analysis in this article is descriptive and traceability-based; inferential statistical tests are neither reported nor implied.

Research Ethics

Permission was sought from the course coordinator before lecturer interviews. The development team stated that confidentiality of the information obtained during requirements elicitation was maintained. Because the project report did not document formal ethics-board approval, participant consent forms, or identifiable personal data collection, this article reports only the ethical procedures explicitly supported by the source document.

RESULTS

User Roles and Functional Requirements

Requirements analysis identified three principal roles. Students required direct visibility of milestones, deadlines, requirements, topic options, notifications, supervision schedules, and feedback. Supervisors required real-time access to student progress and tools for scheduling and feedback. Administrators required account management, schedule and notification administration, reporting, and data-security controls. Table 1 summarizes the role-function mapping.

Table 1. Role-based requirements of the web-based capstone monitoring system.

User role	Core functions	Main information input	Expected output
Student	Submit research topic; upload progress; monitor milestones and deadlines; schedule supervision	Topic proposals, progress reports, meeting requests	Status updates, notifications, supervisor feedback
Supervisor	Monitor progress; review submissions; provide feedback; manage supervision schedule	Feedback, approval decisions, schedule information	Student progress dashboard, supervision records
Administrator	Manage users; maintain schedules and notices; generate reports; support data control	User records, configuration data, announcements	System reports, controlled access, consolidated records

System Design Outcomes

The context model defined bidirectional information exchange among students, supervisors, administrators, and the monitoring system. The data-flow design decomposed the workflow into topic submission, progress reporting, supervisor review, notification, scheduling, and administrative reporting. The entity-relationship design linked core records for users, capstone topics, progress submissions, supervision activities, feedback, and related academic data. Together, these artifacts reduced ambiguity between user requirements and database implementation. The detailed workflow and database relationships are presented in Figures 2 and 3, respectively.

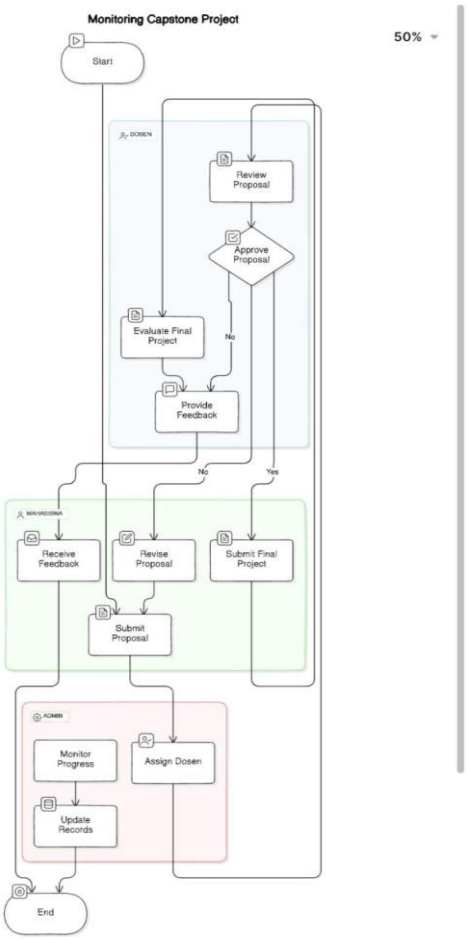


Figure 2. Data-flow and activity representation of the capstone monitoring workflow; *Source: Original system-development documentation (2024).*

Figure 2 details the operational workflow across the three user roles. It depicts proposal submission, supervisor review and approval, feedback and revision cycles, final-project submission and evaluation, and administrative monitoring and lecturer assignment. The iterative return paths visually support the Agile-oriented development logic adopted in the system.

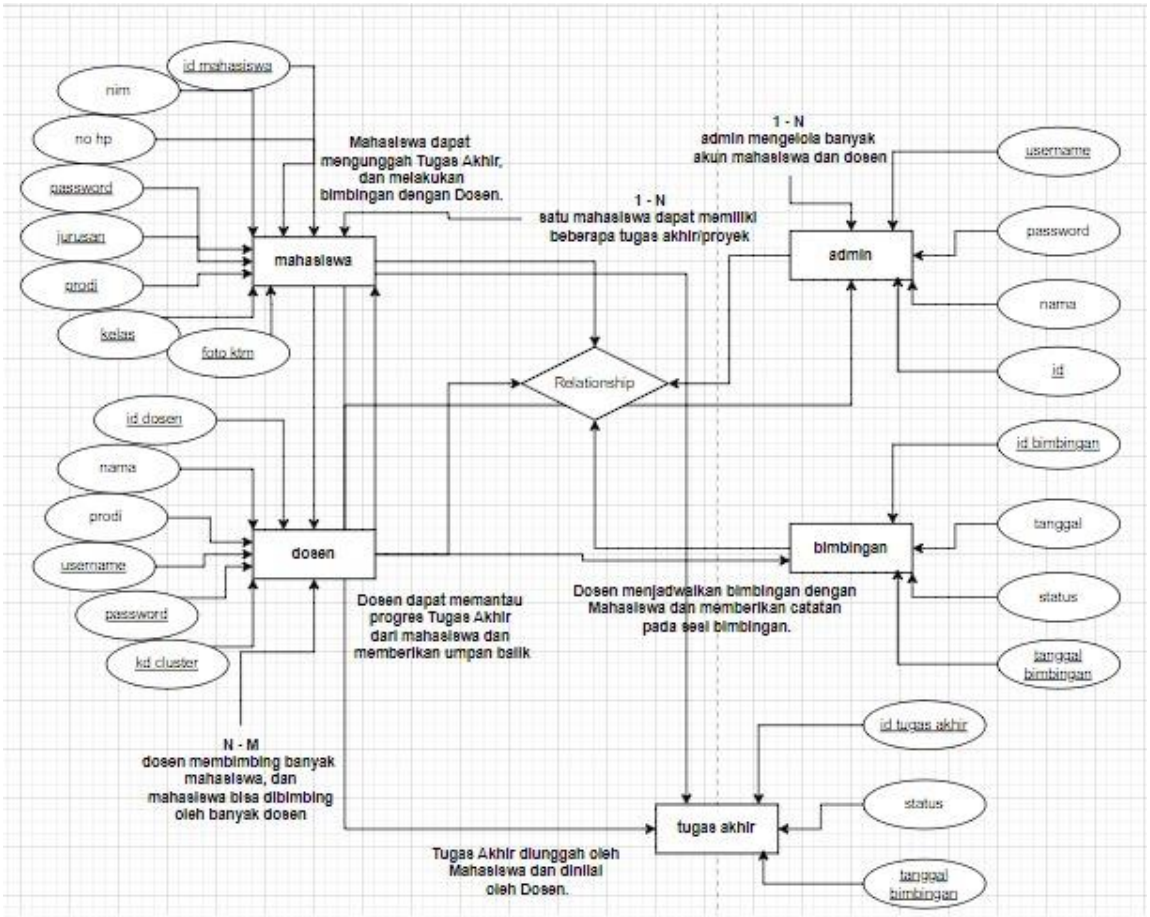


Figure 3. Entity-relationship diagram of the capstone monitoring database; *Source: Original system-development documentation (2024).*

Figure 3 shows the proposed relational structure connecting student, supervisor, administrator, supervision, and final-project entities. The model supports one-to-many and many-to-many relationships required for account management, supervision assignment, progress documentation, scheduling, and project-status tracking.

Functional Verification

The source report states that black-box testing confirmed that the developed functions operated according to the requirements. Table 2 presents a requirements-to-verification synthesis. Because the raw test sheet was not included in the source file, the table reports functional conformity qualitatively rather than inventing numerical pass rates.

Table 2. Qualitative synthesis of black-box functional verification.

Functional domain	Representative test condition	Expected behavior	Reported outcome
Authentication and role access	User signs in under a registered role	Display functions authorized for student, supervisor, or administrator	Functionally conformant
Topic submission	Student submits a proposed research topic	Store and display the proposal for review	Functionally conformant
Progress monitoring	Student uploads progress information	Update milestone/status data on the dashboard	Functionally conformant
Notifications	A deadline, feedback item, or supervision event is recorded	Provide the relevant notification to the user	Functionally conformant
Supervision scheduling	Student or supervisor records a meeting schedule	Save and display the supervision schedule	Functionally conformant
Supervisor feedback	Supervisor enters comments or guidance	Store and deliver feedback to the relevant student	Functionally conformant
Administrative management	Administrator updates accounts, schedules, or notices	Apply changes and maintain system records	Functionally conformant

Significant Findings and Statistical Interpretation

The most significant development outcome was the consolidation of previously dispersed capstone-management activities into one role-based platform. The system connected progress evidence, deadlines, feedback, and supervision schedules, thereby creating a transparent record that could support earlier academic intervention.

A second important finding was the explicit separation of student, supervisor, and administrator responsibilities. This role structure is essential for both workflow clarity and access control, particularly when the same system contains academic progress records and evaluative comments.

No inferential statistical evaluation can be reported from the available data. The source document contains no user-satisfaction sample, pretest-posttest comparison, reliability coefficient, task-completion time, or hypothesis test. Therefore, the evidence supports functional feasibility, not proven improvements in efficiency, learning outcomes, user satisfaction, or completion rates.

DISCUSSION

Interpretation of the Findings

The prototype addresses a central weakness in traditional capstone administration: limited process visibility. By centralizing milestones, progress reports, schedules, and feedback within a shared digital platform, the design mitigates common challenges such as fragmented communication, difficulty in tracking student progress, and the lack of consolidated, transparent documentation (Haris et al., 2025; Leite et al., 2023, p. 42). This approach shifts the focus from managing isolated project artifacts to fostering continuous supervision and active process management, which is essential for ensuring project quality and enabling timely academic intervention (Gary & Koehnemann, 2020; Ghonim & Eweda, 2018; Marel et al., 2022). Furthermore, by making project health visible to all stakeholders, the platform supports improved dyadic supervision and empowers students through increased ownership of their work (Leite et al., 2023, p. 42).

The adoption of an Agile-oriented development process was highly appropriate, as capstone systems involve diverse stakeholder requirements that often evolve throughout the implementation lifecycle (Dhir et al., 2024; Nechaeva & Voshta, 2025). Iterative development facilitates the progressive clarification of workflows and ensures the incremental delivery of features that respond directly to user needs, ultimately enhancing stakeholder engagement and satisfaction (Dhir et al., 2024; Peru. et al., 2025). This interpretation aligns with broader software-engineering and educational research, which underscores that while Agile practices significantly improve system adaptability and responsiveness, their successful integration requires disciplined communication, structured documentation, and methodological support tailored to the educational environment (Dybå & Dingsøyr, 2008; Nechaeva & Voshta, 2025).

Comparison with Previous Studies

The system's emphasis on feedback, progress review, and scheduling reflects the supervisory expertise identified in previous capstone literature. Marel et al. (2022) described graduation-project supervision as requiring the coordination of content expertise, research-process guidance, and student learning support. The present design operationalizes part of that coordination through structured digital records and repeated interaction points.

The system also complements Agile and project-based learning research. (Marnewick, 2023) found that iterative project work can help students apply project-management principles and develop technical and soft skills. López-Alcarria et al. (2019) likewise reported growing educational use of Agile methods to foster collaboration and adaptive learning(Marnewick, 2023). The current system does not itself constitute an Agile pedagogy, but it provides infrastructure that can support iterative milestones, short feedback cycles, and shared accountability.

Implications

For students, the platform may support self-regulation by making expectations, deadlines, and progress states visible. For supervisors, it may provide a consolidated basis for prioritizing feedback and identifying delayed milestones. For administrators, standardized digital records may improve coordination, reporting, and accountability across capstone cohorts.

From a software-design perspective, the role-based architecture and relational data model create a foundation for future integration with institutional academic systems. Potential extensions include single sign-on, document versioning, rubric-based assessment, analytics for delay risk, audit logs, secure file storage, and mobile-responsive interfaces. Such extensions should be evaluated against recognized software-quality dimensions, especially usability, reliability, security, maintainability, and performance efficiency.

Limitations

This study has several limitations. First, the source report does not specify the number or characteristics of interviewed stakeholders, which limits assessment of requirements coverage. Second, black-box testing is reported only in summary form; raw test cases and defect logs were not available. Third, no usability testing, security testing, performance testing, or database stress testing was documented. Fourth, the study did not compare the system with existing capstone-management practices using experimental or quasi-experimental data. Finally, the prototype was developed in a single academic context, so transferability to other institutions remains untested. These limitations mean that the present findings should be interpreted as evidence of design coherence and functional feasibility, rather than definitive evidence of educational effectiveness or organizational efficiency.

CONCLUSION

This study designed and developed a web-based capstone project monitoring application that integrates topic submission, progress tracking, deadline monitoring, notifications, supervision scheduling, feedback, account management, and reporting. The system was structured around three user roles and modeled through context, data-flow, behavioral, sequence, and entity-relationship representations. Functional black-box verification, as reported in the source project, indicated conformity between the implemented functions and the documented requirements.

The study reinforces the proposition that capstone supervision benefits from continuous, transparent, and traceable coordination. The developed prototype provides a technically plausible platform for supporting these processes and for enabling more timely interaction among students, supervisors, and administrators. Nevertheless, the available evidence does not yet establish statistically significant gains in completion time, learning performance, or user satisfaction.

Future authors and researchers are encouraged to submit evidence-based suggestions for system refinement, particularly regarding usability, accessibility, security, integration with academic information systems, and validated evaluation designs. Subsequent studies should involve representative users, report complete testing datasets, and examine whether the application measurably improves supervision quality and capstone completion outcomes.

Acknowledgments

The authors acknowledge Fransisca Angelia Sebayang, M.Kom., as the course lecturer, and the Department of Information Technology, Politeknik Negeri Samarinda, for academic guidance and institutional support during the development of the project.

Conflict of Interest

The authors declare that they have no conflict of interests related to this study.

REFERENCES

Abayomi, A. A., Ogeawuchi, J. C., Gbenle, T. P., Agboola, O. A., & Uzoka, A. C. (2024). Advances in Project Stakeholder Communication and Transparency Using Cloud Collaboration Platforms. *International Journal of Scientific Research in Science and Technology*, *11*(5), 633–652. https://doi.org/10.32628/ijsrst52310373

Bianchi, M., Marzi, G., & Guerini, M. (2018). Agile, Stage-Gate and their combination: Exploring how they relate to performance in software development. *Journal of Business Research*, *110*, 538–553. https://doi.org/10.1016/j.jbusres.2018.05.003

Burkin, V. (2023). Mitigating Risks in Software Development through Effective Requirements Engineering. In *arXiv (Cornell University)*. Cornell University. https://doi.org/10.48550/arxiv.2305.05800

Cristóbal, J., Rodríguez-Triana, M. J., Gallego-Lema, V., Arribas-Cubero, H. F., Asensio-Pérez, J. I., & Martínez-Monés, A. (2017). Monitoring for Awareness and Reflection in Ubiquitous Learning Environments. *International Journal of Human-Computer Interaction*, *34*(2), 146–165. https://doi.org/10.1080/10447318.2017.1331536

Dhir, S., Kumari, S., & Gupta, S. (2024). Agile Approaches and Practices in Higher Education Systems. In *Advances in educational technologies and instructional design book series* (pp. 1–22). IGI Global. https://doi.org/10.4018/979-8-3693-2169-0.ch001

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, *50*, 833–859. https://doi.org/10.1016/j.infsof.2008.01.006

Farr, J. V., Lee, M. A., Metro, R. A., & Sutton, J. P. (2001). Using a Systematic Engineering Design Process to Conduct Undergraduate Engineering Management Capstone Projects. *Journal of Engineering Education*, *90*(2), 193–197. https://doi.org/10.1002/j.2168-9830.2001.tb00590.x

Ferrão, R. C., & Soares, L. P. (2026). Enhancing Capstone Program Workflow: A Case Study on a Platform for Managing Academic-Industry Projects. In *arXiv (Cornell University)*. Cornell University. https://doi.org/10.48550/arxiv.2602.20120

Gary, K., & Koehnemann, H. (2020). *The Benefits Of Transparency In Managing Software Capstone Projects*. https://doi.org/10.18260/1-2--16985

Ghonim, M., & Eweda, N. (2018). Best practices in managing, supervising, and assessing architectural graduation projects: A quantitative study. *Frontiers of Architectural Research*, *7*(3), 424–439. https://doi.org/10.1016/j.foar.2018.06.002

Haris, N. A., Zailani, N. A., Hasim, N., & Rahman, F. A. (2025). An Integrated Approach to Supervising and Managing Students Final Year Projects in Higher Institutions. *International Journal of Advanced*

Research in Computational Thinking and Data Science, *5*(1), 19–27. https://doi.org/10.37934/ctds.5.1.1927

Leite, L. O., Lagstedt, A., Kolog, E. A., & Tsupari, K. (2023). Contextualising thesis process digitalisation at a university in Ghana. *Australasian Journal of Educational Technology*, *39*(4), 33–47. https://doi.org/10.14742/ajet.8801

López-Alcarria, A., Olivares-Vicente, A., & Vilches, M. de F. P. (2019). A Systematic Review of the Use of Agile Methodologies in Education to Foster Sustainability Competencies. *Sustainability*, *11*(10), 2915–2915. https://doi.org/10.3390/su11102915

Marel, I. van der, Munneke, L., & Bruijn, E. de. (2022). Supervising graduation projects in higher professional education – A literature review. *Educational Research Review*, *37*, 100462–100462. https://doi.org/10.1016/j.edurev.2022.100462

Marnewick, C. (2023). Student experiences of project-based learning in agile project management education. *Project Leadership and Society*, *4*, 100096–100096. https://doi.org/10.1016/j.plas.2023.100096

Nasir, N. N. binti M., & Wahab, H. N. bin A. (2024). Enhancing course management and student monitoring through the myStudentStatus application: A user perception study. *E3S Web of Conferences*, *479*, 7007–7007. https://doi.org/10.1051/e3sconf/202447907007

Nechaeva, O. A., & Voshta, V. Y. (2025). Integration of Agile and Scrum Methodologies into Educational Management Systems: A Theoretical and Empirical Analysis of Innovative Approaches. *THE TIDINGS of the Baltic State Fishing Fleet Academy Psychological and Pedagogical Sciences (Theory and Methods of Professional Education)*, 7–15. https://doi.org/10.46845/2071-5331-2025-3-73-7-15

Ngereja, B. J., Hussein, B., & Andersen, B. (2020). Does Project-Based Learning (PBL) Promote Student Learning? A Performance Evaluation. *Education Sciences*, *10*(11), 330–330. https://doi.org/10.3390/educsci10110330

Persson, J. S., Mathiassen, L., Boeg, J., Madsen, T. S., & Steinson, F. (2009). Managing Risks in Distributed Software Projects: An Integrative Framework. *IEEE Transactions on Engineering Management*, *56*(3), 508–532. https://doi.org/10.1109/tem.2009.2013827

Peru., U. N. F. V., Lima, Patricio-Peralta, C., Mondragón, J. E. Z., Peru., U. C. V., Lima, Meneses, J. P., Peru., U. N. de la A. P., Iquitos, Villacorta, J. M. R., Peru., U. N. de la A. P., Iquitos, Angeles, A. E. C., Peru., U. T. del P., Lima, Cortegano, C. A. G., & Peru., U. N. de la A. P., Iquitos,. (2025). Agile Methodology Integrated Programming Planned (PIP) to Optimize the Development of Educational Software. *International Journal of Innovative Technology and Exploring Engineering*, *14*(6), 15–19. https://doi.org/10.35940/ijitee.d1075.14060525

Rahman, M., Sarwar, H., Tin, T. T., & Kader, M. A. (2025). A systematic literature review on final year capstone project in an engineering curriculum. *Edelweiss Applied Science and Technology*, *9*(6), 1408–1440. https://doi.org/10.55214/25768484.v9i6.8146

Sá, M. J., Santos, A. I., & Serpa, S. (2021). The Academic Supervisor of Higher Education Students' Final Projects: A Gatekeeper of Quality? *Academic Journal of Interdisciplinary Studies*, *10*(1), 152–152. https://doi.org/10.36941/ajis-2021-0013

Talmi, I., Hazzan, O., & Katz, R. (2018). Intrinsic Motivation and 21st-Century Skills in an Undergraduate Engineering Project: The Formula Student Project. *Higher Education Studies*, *8*(4), 46–46. https://doi.org/10.5539/hes.v8n4p46

Yılmaz, M., Tasel, F. S., Güleç, U., & Sopaoğlu, U. (2018). Towards a process management life-cycle model for graduation projects in computer engineering. *PLoS ONE*, *13*(11). https://doi.org/10.1371/journal.pone.0208012