



Development of Academic Dashboard Application with Extreme Programming Method to Optimize Decision Making in Higher Education

Denny Jean Cross Sihombing

Faculty of Engineering, Atma Jaya Catholic University of Indonesia

Email: denny.jean@atmajaya.ac.id

Keywords

Academic Dashboard Application,
Extreme Programming,
Higher Education

Abstract. This research aims to investigate the development of an academic dashboard application using the Extreme Programming (XP) method as a response to digital transformation in higher education. Digital transformation presents opportunities and challenges in academic information management, requiring effective and responsive solutions. In this research, the academic dashboard application development approach integrates XP principles. The research method involves planning, design, coding, testing, delivery, and feedback. The results showed that the XP method provided responsiveness to changes, improved code quality, and improved application security. Thorough testing ensures the application is well-tested and can withstand diverse usage loads. The delivery phase ensures that each iteration meets user acceptance criteria. End-user feedback provides valuable insights for continuous improvement, which are integrated into the next iteration.

1. INTRODUCTION

Higher education has become increasingly involved in digital transformation as technology evolves, creating opportunities to enhance learning experiences, manage data, and support decision-making. Technology in higher education is no longer limited to learning management systems or digital libraries but has expanded into various aspects, including administration, communication, and data analysis. The increased use of technology in higher education has resulted in several positive impacts. First, the increased accessibility of information and distance learning allows students to access learning materials from anywhere[1]–[3]. Secondly, technology opens the door to more interactive and innovative approaches to learning, incorporating elements such as simulation, e-learning, and game-based learning[4]–[6].

However, amidst these opportunities, some challenges need to be overcome. First, universities are faced with demands to maintain the security and privacy of student and staff data. Second, technology integration requires significant financial and human resource investments. In addition, the rapid dynamics of technological change require universities to remain adaptive to stay caught up. Colleges today face significant challenges related to the growth of data volumes. Over time, academic data continues to grow, covering various aspects such as students' personal information, academic records, and other administrative data. This growth requires an efficient and easily accessible data management system[7]–[9].

Diversification of user needs is crucial in defining the importance of academic dashboard applications. Students, lecturers, and administrative staff have different information needs. Students need quick and transparent access to their academic progress, while lecturers need tools to monitor and evaluate student performance. Administrative staff also need a solution that can efficiently manage data for administrative purposes. Academic dashboard apps emerge as a solution that effectively addresses this complexity. First, dashboard apps manage the growing volume of data by providing a structured data storage and processing system. Thus, information can be accessed quickly and in an organized manner[10]–[12].

Furthermore, the role of dashboard applications in improving efficiency is indispensable. These applications give students easy access to their academic information and allow lecturers to monitor student performance effectively. In addition, dashboard apps provide the support needed in decision-making, whether in planning, teaching, evaluating student performance, or designing administrative policies. Thus, the development of academic dashboard applications has become necessary to face the challenges of academic information complexity and meet the increasingly diverse needs of stakeholders in higher education[3], [13], [14].

The development of academic dashboard applications cannot be separated from several problems that can affect the success and efficiency of the process. The first technical challenge faced in developing an academic dashboard application is the limitations of existing systems. Colleges may already use various systems and infrastructure, and integrating a new dashboard application with these systems can be complex. Such limitations may include differences in data formats, different technology architectures, or other technical restrictions. Data security and user privacy challenges are critical in developing academic dashboard applications. With the high sensitivity of academic data and students' personal information, data protection and user privacy should be top priorities[15]. The need to ensure data security from external threats and implement proper access controls became a crucial focus in the development of the application. Challenges in the development process include flexibility in dealing with changing requirements. The needs and expectations of users and stakeholders may change over time. Therefore, the academic dashboard application must be designed and developed with sufficient flexibility to adapt to these changing needs without compromising stability and quality. Universities are dynamic environments where policies, procedures, and needs can change rapidly. This challenge includes responding quickly to such changes in application development. Developers must be able to adopt new academic policy changes or administrative needs without hampering the operational continuity of the dashboard application[8].

The Extreme Programming (XP) approach plays a vital role in academic dashboard application development, providing a dynamic framework that is responsive to the challenges faced in a higher education environment. Extreme Programming (XP) is a software development methodology emphasizing team collaboration, flexibility, and speed in responding to change. Some of the critical principles of XP involve iterative development, continuous testing, and close communication between developers, customers, and stakeholders. The advantages of XP involve the ability to adapt to changing needs quickly, improved code quality through continuous testing practices, and active customer involvement. On the other hand, the disadvantages of XP can include the challenges of managing large and complex projects and the need for high commitment from the development team[16], [17].

One of the critical aspects of XP that is relevant in developing academic dashboard applications is responsiveness to changing needs. In a dynamic college environment, needs related to academic information, policies, or application features can change quickly. XP allows the development team to quickly adapt the application to these changes through iterative and flexible development. User engagement is a critical element of academic dashboard application development, and XP supports this by encouraging constant communication between developers and stakeholders. By actively involving students, faculty, and administrative staff in the development process, XP ensures that the application meets users' expectations and can cope with dynamic changes in their needs. Applying the principles of XP in the development of academic dashboard applications can create an adaptive, high-quality solution, providing significant added value to the entire university ecosystem[18].

An in-depth understanding of the higher education context and the crucial role of academic dashboards is an essential foundation for designing and developing practical solutions to the challenges faced by higher education institutions. Identifying challenges in the development and decision-making process is a critical step in designing a responsive and adaptive solution. The justification of using the Extreme Programming (XP) method reinforces this approach as a suitable and effective solution to address these challenges. In identifying the challenges of developing an academic dashboard application, several aspects emerged in the context of higher education. First, the complexity of academic information management includes significant growth in data volume, which requires a system that can manage and present data efficiently. Diversifying user needs, such as students, lecturers, and administrative staff, poses a challenge as each group requires different access and functionality. Another challenge lies in the decision-making process. Universities must overcome the limitations of existing systems, secure user data and privacy, and remain flexible in the face of rapidly changing needs. There is a need for a solution that can accurately monitor student academic

performance and support efficient decision-making processes amidst higher education dynamics[19]–[22].

This research aims to develop an academic dashboard application using the Extreme Programming (XP) method for optimizing decision-making in a college environment. The first objective is to design and develop a responsive academic dashboard application capable of adjusting to rapid changes in academic information needs in higher education. Furthermore, this research aims to optimize the student, lecturer, and administrative staff decision-making process through specially designed features. User involvement is also a focus, increasing the active participation of students, lecturers, and administrative staff in every application development stage. In addition, this research aims to ensure data security and privacy by adhering to high standards and maintaining user confidence in the application. The contributions of this research involve innovative solutions for dynamic universities, improving the efficiency of the decision-making process, providing an open development model for future research, improving user experience, and developing best practices in data security. By achieving these goals, this research is expected to make a positive and sustainable contribution to supporting the development of responsive and adaptive academic dashboard applications in higher education.

2. METHOD

The stages of academic dashboard application development using the Extreme Programming (XP) method can be divided into several steps. In the planning stage of developing an academic dashboard application using the Extreme Programming (XP) method, the team identifies user needs, determines the overall scale and scope of the project, and divides tasks and responsibilities among team members. An iterative plan was created for each development cycle, which involved selecting features and functionality to be implemented at each iteration. The research phase consisted of several steps, as shown in Figure 1.

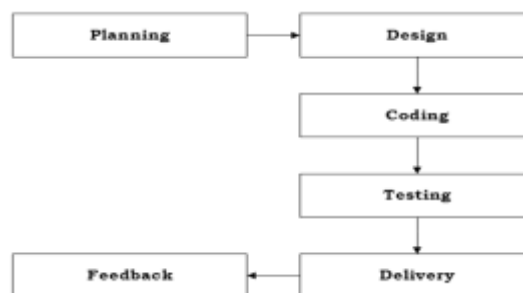


Figure 1. Research Stages

Planning

In the planning stage of developing an academic dashboard application using the Extreme Programming (XP) method, the team identifies user needs, determines the overall scale and scope of the project, and divides tasks and responsibilities among team members. An iterative plan was created for each development cycle, which involved selecting the features and functionality to be implemented at each iteration.

Design

The team conducts collaborative design sessions at the design stage involving developers, testers, and user representatives. They design a responsive and easy-to-use user interface, define the required database structure, and establish an application architecture that conforms to XP principles.

Coding

Once the design was complete, the team entered the coding phase, implementing the key features designed in the design session. Continuous Integration principles are applied to regularly integrate code changes, while pair programming and Test-Driven Development (TDD) practices improve code quality and security.



Testing

Next, in the testing phase, the team implemented continuous unit and integration testing, involving the testing team in every stage of development. Thorough app functionality and load tests were conducted to ensure app performance when used concurrently by multiple users.

Delivery

After that, at the delivery stage, the team ensures that each iteration has met the user acceptance criteria. They provide end-users with an accessible app version, integrate it with the college's existing systems, and implement a regular delivery process according to an iterative schedule.

Feedback

Finally, in the feedback stage, the team held demo sessions and collected user feedback through interviews and surveys. This feedback is used to implement necessary improvements and changes to the app. The team also compiled learning notes from each iteration to apply to the next iteration, making feedback an integral part of the continuous development process. Thus, the academic dashboard application can continue to evolve and improve according to user needs and expectations.

3. RESULTS AND DISCUSSION

Planning

As shown in Table 1, the development of the academic dashboard application using the Extreme Programming (XP) method included an in-depth understanding of user needs and a comprehensive definition of the scale and scope of the project. The team successfully identified user needs by involving representatives from students, lecturers, and the college's administrative staff. This process helped the team gain a more accurate insight into the essential features and functionalities to meet the demands of each party. In addition, creating an iterative plan for each development cycle helps ensure the project runs on schedule and can adapt to changing needs more adaptively. The selection of features and functionality to be implemented at each iteration is based on priorities set together with stakeholders so that the project can add value at each development phase. Thus, the results of the planning phase show a strong foundation for developing a responsive academic dashboard application that meets user expectations. A good understanding of user needs and structured planning will provide a solid foundation for the following stages of development, ensuring the project's overall success.

Table 1. Planning Results

Stage	Description
Identification of User Needs	Number of students who will use the dashboard. List of features desired by lecturers to monitor student performance. Administrative staff preferences regarding academic data management.
Determination of Project Scale and Scope	The time budget available for application development. Human resources that can be allocated, such as the number of developers, testers, and other stakeholders. Scope of the academic dashboard application, such as whether it involves general academic information, attendance records, or financial information
Division of Tasks and Responsibilities	Resource allocation table with roles and responsibilities of each team member. Division of responsibilities for monitoring application development, testing, and collecting user feedback.
Iterative Plan	An iterative schedule with time estimates for each development cycle. List of features or functions scheduled for each iteration. Acceptance criteria to determine the success of each iteration.

Design

The research results of the design phase in developing the academic dashboard application using the Extreme Programming (XP) method include intensive collaboration between developers,



testers, and user representatives. During the design session, the team successfully designed a user interface that was responsive and designed to ensure affordability and optimal usability. The database structure definition process became a crucial element in laying the foundation of the application. The team considered the extensive academic information needs and carefully delineated the relationships between data entities. The selection and mapping of the application architecture was done by ensuring conformity with XP principles, such as clarity, simplicity, and flexibility. The success of the design stage was seen in detailing the user interface layout, creating an efficient database schema, and designing an application structure that could quickly adapt to changing needs. The results of this stage guided the subsequent implementation, forming a solid foundation for developing a responsive academic dashboard application that conforms to XP principles.

Coding

The research results from the coding stage in developing the academic dashboard application using the Extreme Programming (XP) method reflect the intensive collaborative work of the development team to translate the agreed design into a concrete implementation. The team successfully implemented the critical features designed in the design session, taking into account XP principles that promote quality, clarity, and security in the development process. The application of Continuous Integration principles helped ensure that code changes were regularly integrated into the source code. This provides the benefits of early identification of potential conflicts and integration issues, speeds up development, and ensures that the application is always testable.

During the coding stage, the team implemented pair programming practices with two developers working together to write and review the code. The presence of two different minds helps detect and fix errors faster, improving the overall quality and safety of the code. Meanwhile, the application of Test-Driven Development (TDD) helps produce well-tested code, ensuring that any changes do not break existing functionality. The result of this coding stage is the implementation of features by design, with continuously integrated code, improved quality through pairing practices, and guaranteed security through TDD. Overall, the coding stage was essential in developing a responsive academic dashboard application that conforms to XP principles.

Testing

The research results of the testing stage in developing academic dashboard applications using the Extreme Programming (XP) method include the team's collaborative efforts to ensure that the application has been thoroughly tested and can face diverse usage conditions. In the continuous unit and integration testing phase, the team identifies and fixes potential problems at the module level and integration between modules. By involving the testing team in every stage of development, the team can ensure that any changes do not affect existing functionality and that the application remains consistent in performing its tasks.

Thorough app functionality tests are conducted to verify that the app meets all the functional requirements that have been set. This includes testing the key features that have been implemented, as well as testing everyday use cases. Load tests were conducted to assess the application's performance when used concurrently by multiple users, ensuring the application can handle significant load without experiencing performance degradation.

The result of the testing phase is an academic dashboard application that has been thoroughly tested, has a high level of quality and reliability, and can answer the demands of diverse uses. Findings from testing provide valuable feedback to refine and optimize the application to end users before it is launched, thus minimizing the risk of bugs and performance issues after launch.

Table 2. Testing Results

Testing Items	Test Description	Scenario	Results
Integration Testing	Integration between Database and User Interface	Update student grades by lecturers on the user interface.	Correct value updates occur in the database.
	Integration between Financial System and	Tuition payment by students on the user	Student balances are updated in the

	User Interface	interface	financial system.
Functionality Testing	Student Search Feature	Search by NIM	The corresponding student information appears.
	Student Performance Monitoring Feature	View summary of student grades	A graph or table showing the academic progress of students
Load Testing	User Load Simulation	Load data for 500 students simultaneously	Successful

Delivery

Research results from the delivery stage in developing academic dashboard applications using the Extreme Programming (XP) method show that each iteration successfully meets user acceptance criteria. The development team ensures that the developed version of the application can be accessed by end users smoothly and that integration with existing systems in higher education has been successfully carried out. The delivery process is carried out regularly according to a predetermined iterative schedule. Each delivery includes updates or additions of new features, bug fixes, and performance improvements identified during previous iterations. In this process, the team focuses on adding value to the end users with each iteration.

Table 3. Delivery Results

Stage	Description
User Acceptance Criteria	Ensure that any implemented features or changes are in line with end-user expectations. Involve end users in testing or validation sessions to get direct feedback.
Accessibility and Integration	Ensure that end users can access the application version easily without technical barriers. Successful integration with existing systems in the college to ensure data coherence and functionality.
Iterative Schedule	Develop and follow an iterative delivery schedule according to the development plan. Adjust the schedule if any changes in requirements or priorities arise during the iteration.
Continuous Improvement	Implement changes based on feedback from end users after each delivery. Continuously improve the quality and features of the app on an ongoing basis.

With a successful delivery phase, end users can widely use the academic dashboard application, provide the desired benefits, and evolve according to user needs and feedback.

Feedback

In the feedback phase of developing the academic dashboard application using the Extreme Programming (XP) method, the development team has conducted interviews and surveys to obtain valuable feedback from end users. User interviews revealed that the new features were very helpful in tracking academic progress. However, there was a desire to add notifications for important announcements. Surveys conducted on users also provided an overview of the users' level of satisfaction with the new user interface. The survey results showed that most users were satisfied with the new user interface, but some felt it needed improvement, especially the delay in loading the student grades page.

Based on the feedback received, the development team identified several improvement steps. One was optimizing code and resources to address the slowness when loading the student grades page. In addition, the team also decided to add a notification feature for important announcements by requests from users such as Users. These process improvements were also recorded in the team's learning log. Despite challenges like page load slowness, the team overcame these issues through technical optimizations. The learning log also highlighted the success of the user interface improvements and the positive benefits it provided to students in tracking academic progress.

Moving forward, the development team will integrate these changes into the next iteration of the academic dashboard application. This reflects an ongoing development approach responsive to user feedback, making feedback an integral part of the development cycle. As such, the academic dashboard application will continue to evolve and improve according to user needs and expectations.

4. CONCLUSION

This research describes the development of an academic dashboard application using the Extreme Programming (XP) method as a response to digital transformation in higher education. The XP method in developing the academic dashboard application shows excellence in responsiveness to change. The iterative approach, adaptive planning, and application of XP principles allow the development team to adapt the application to the evolving needs in a dynamic educational environment. Secondly, code quality and application security are the main focus of XP implementation. Continuous Integration practices, continuous testing, and Test-Driven Development (TDD) applications contribute to quality improvement. Application security is emphasized throughout the development cycle to protect sensitive data and system integrity. Third, user involvement was crucial in successfully developing the academic dashboard application. End-user participation in every stage, from planning to delivery, provides valuable feedback that supports continuous improvement. This creates a product more in line with user needs and expectations. Overall, this research shows that applying XP methods in developing academic dashboard applications can provide an effective and responsive solution to the demands of digital transformation in higher education. The integration of XP principles paves the way for continuous improvement and innovation in academic information management, positively contributing to the overall advancement of higher education.

REFERENCES

- [1] L. W. Santoso and Yulia, "Data Warehouse with Big Data Technology for Higher Education," in *Procedia Computer Science*, Elsevier B.V., 2017, pp. 93–99. doi: 10.1016/j.procs.2017.12.134.
- [2] J. Petchamé, I. Iriondo, O. Korres, and J. Paños-Castro, "Digital transformation in higher education: A qualitative evaluative study of a hybrid virtual format using a smart classroom system," *Heliyon*, vol. 9, no. 6, Jun. 2023, doi: 10.1016/j.heliyon.2023.e16675.
- [3] J. L. R. Muñoz *et al.*, "Systematic Review of Adaptive Learning Technology for Learning in Higher Education," *Eurasian Journal of Educational Research*, vol. 2022, no. 98, pp. 221–233, 2022, doi: 10.14689/ejer.2022.98.014.
- [4] S. Wan and Z. Niu, "A Hybrid E-learning Recommendation Approach Based on Learners' Influence Propagation," *IEEE Trans Knowl Data Eng*, vol. 4347, no. c, pp. 1–1, 2019, doi: 10.1109/tkde.2019.2895033.
- [5] Q. N. Naveed, M. R. N. Mohamed Qureshi, A. Shaikh, A. O. Alsayed, S. Sanober, and K. Mohiuddin, "Evaluating and Ranking Cloud-Based E-Learning Critical Success Factors (CSFs) Using Combinatorial Approach," *IEEE Access*, vol. 7, pp. 157145–157157, 2019, doi: 10.1109/ACCESS.2019.2949044.
- [6] S. A. Salloum, A. Qasim Mohammad Alhamad, M. Al-Emran, A. Abdel Monem, and K. Shaalan, "Exploring students' acceptance of e-learning through the development of a comprehensive technology acceptance model," *IEEE Access*, vol. 7, pp. 128445–128462, 2019, doi: 10.1109/ACCESS.2019.2939467.

- [7] S. Alharbi and S. Drew, "Using the Technology Acceptance Model in Understanding Academics' Behavioural Intention to Use Learning Management Systems," *International Journal of Advanced Computer Science and Applications*, vol. 5, no. 1, 2014, doi: 10.14569/ijacsa.2014.050120.
- [8] D. J. C. Sihombing, "Academic Data Warehouse Modeling in Higher Education Using Nine-Step Design Methodology," *Journal of Information Systems and Informatics*, vol. 4, no. 4, 2022, doi: 10.51519/journalisi.v4i4.399.
- [9] K. S. Double, J. A. McGrane, and T. N. Hopfenbeck, "The Impact of Peer Assessment on Academic Performance: A Meta-analysis of Control Group Studies," *Educ Psychol Rev*, vol. 32, no. 2, pp. 481–509, Jun. 2020, doi: 10.1007/s10648-019-09510-3.
- [10] A. Calma, G. Suder, and T. P. Kenworthy, "Focus on education: Taking stock of key themes, topics, trends and communities in international business and international management education research," *The International Journal of Management Education*, vol. 22, no. 1, p. 100902, Mar. 2024, doi: 10.1016/j.ijme.2023.100902.
- [11] J. M. Sáez-López, R. Cózar-Gutiérrez, J. A. González-Calero, and C. J. G. Carrasco, "Augmented reality in higher education: An evaluation program in initial teacher training," *Educ Sci (Basel)*, vol. 10, no. 2, 2020, doi: 10.3390/educsci10020026.
- [12] J. Lassa, M. Petal, and A. Surjan, "Understanding the impacts of floods on learning quality, school facilities, and educational recovery in Indonesia," *Disasters*, vol. 47, no. 2, pp. 412–436, Apr. 2023, doi: 10.1111/disa.12543.
- [13] M. Almashhadani, H. Almashhadani, and H. A. Almashhadani, "The Impact of Education on Construction Management: A Comprehensive Review," *International Journal of Business and Management Invention (IJBMI) ISSN*, vol. 12, pp. 284–290, 2023, doi: 10.35629/8028-1206284290.
- [14] D. Yin, "Study on the Simulative Financial Management System in ERP Education," *IERI Procedia*, vol. 2, pp. 642–648, 2012, doi: 10.1016/j.ieri.2012.06.147.
- [15] P. Silva, C. Gonçalves, N. Antunes, M. Curado, and B. Walek, "Privacy risk assessment and privacy-preserving data monitoring," *Expert Syst Appl*, vol. 200, no. January, p. 116867, 2022, doi: 10.1016/j.eswa.2022.116867.
- [16] A. Akhtar, B. Bakhtawar, and S. Akhtar, "EXTREME PROGRAMMING VS SCRUM: A COMPARISON OF AGILE MODELS," *International Journal of Technology, Innovation and Management (IJTIM)*, vol. 2, p. 2022, doi: 10.54489/ijtim.v2i1.77.
- [17] D. J. C. Sihombing, "Analysis and development of the ProTrack application: construction timeline management using Extreme Programming Methodology," Online, 2023.
- [18] J. Chen, T. Yu, L. Yin, J. Tang, and H. Wang, "A unified time scale intelligent control algorithm for microgrid based on extreme dynamic programming," *CSEE Journal of Power and Energy Systems*, vol. 6, no. 3, pp. 583–590, Sep. 2020, doi: 10.17775/CSEEJPES.2019.00100.
- [19] T. Dingsøyr, S. Nerur, V. Balijepally, and N. B. Moe, "A decade of agile methodologies: Towards explaining agile software development," *Journal of Systems and Software*, vol. 85, no. 6, 2012. doi: 10.1016/j.jss.2012.02.033.
- [20] P. Serrador and J. K. Pinto, "Does Agile work? - A quantitative analysis of agile project success," *International Journal of Project Management*, vol. 33, no. 5, 2015, doi: 10.1016/j.ijproman.2015.01.006.
- [21] S. Al-Saqqa, S. Sawalha, and H. Abdelnabi, "Agile software development: Methodologies and trends," *International Journal of Interactive Mobile Technologies*, vol. 14, no. 11, 2020, doi: 10.3991/ijim.v14i11.13269.
- [22] R. Santos *et al.*, "A Comparative Analysis of Agile Teamwork Quality Instruments in Agile Software Development: A Qualitative Approach", doi: 10.18293/DMSVIVA2023-217.