

Optimizing Apartment Management through Extreme Programming-based CRM Application

Denny Jean Cross Sihombing

Faculty of Engineering, Atma Jaya Catholic University of Indonesia

Email: denny.jean@atmajaya.ac.id

Keywords

Extreme Programming
CRM Application
Apartment Management

Abstract. This research aims to develop a Customer Relationship Management (CRM) application specifically for the apartment industry using the Extreme Programming (XP) approach. This approach involves identifying customer needs, creating user stories, and prioritization to ensure application responsiveness. The resulting system has a microservices design, a friendly user interface, and a high level of security. The implementation of quality code involved pair programming and automated testing. The testing process involves User Acceptance Testing (UAT) with the participation of apartment residents to obtain feedback that is applied to improvements. A small and frequent release approach ensures optimal quality and functionality. User involvement through UAT and retrospective sessions ensures the solution matches expectations, providing a positive user experience. By applying XP principles, this research successfully created a responsive, adaptive, and user-orientated CRM application to meet the complex needs of the apartment industry.

1. INTRODUCTION

The apartment industry has become integral to urban development, providing comfortable and affordable living spaces for urbanites. The rapid growth in this sector creates economic opportunities and drives the need for effective management of customer relationships. Customer relationship management (CRM) is a marketing concept and an in-depth strategy to understand, fulfill, and maintain sustainable tenant relationships. The apartment industry plays a crucial role in meeting the housing needs of the growing urban population. With rapid urbanization, apartments are a practical choice and create vibrant communities. Therefore, customer relationship management is critical to ensuring sustainability and operational success in this industry. With rapid urbanization, apartments are a practical choice and create vibrant communities. Customer relationship management is, therefore, key to ensuring sustainability and operational success in this industry. With rapid urbanization, apartments are a practical choice and create vibrant communities. Therefore, customer relationship management is critical to ensuring sustainability and operational success in this industry. CRM is more than just an administrative tool. It is a philosophy that puts the customer at the center. By implementing CRM in the apartment industry, companies can build strong relationships with tenants, improve service quality, and create a loyal customer base. This, in turn, contributes to long-term business sustainability and a positive image in the eyes of society[1], [2].

Developing a Customer Relationship Management (CRM) application specifically for the apartment industry is inseparable from several complex challenges. These challenges include apartment residents' varied needs, from facility maintenance to interaction with management. Designing a CRM application that can accommodate and respond to these dynamics requires a deep understanding of these diverse needs. Information regarding tenants, maintenance, and other aspects is often scattered across various systems and departments. Challenges arise in integrating data from disparate sources to create a holistic, real-time picture of customer relationships. With the sensitivity of occupant data, security, and privacy protection becomes critical. The challenge involves developing solutions that can provide a high level of security without compromising user convenience. The apartment industry is often subject to regulatory and policy changes[3]. Therefore, CRM application development must be able to quickly adapt to these changes without disrupting day-to-day operations. Conventional approaches to software development, such as the waterfall or spiral models, have certain limitations that may not align with the dynamics of CRM application development for the apartment industry. Some limitations that need to be addressed include: Conventional development models tend to be rigid and difficult to adapt to changing requirements that often arise in CRM application

development. The lengthy development process and extensive testing stages can cause delays in the delivery of the final product. This is at odds with industry requirements that often require rapid response and implementation. Implementing regular and continuous improvements or enhancements may hinder the ability of CRM applications to remain relevant and effective in the face of changing business environments[4]–[6].

Conventional models may not facilitate close collaboration with end-users and stakeholders, essential for successful CRM application development. Extreme Programming (XP) is a software development methodology emphasizing team collaboration, rapid response to changing requirements, and high-quality software development. The fundamental principles of XP involve Intensive Communication: XP teams communicate openly and intensively among team members and with customers. Good communication is necessary to ensure a deep understanding of customer needs and to facilitate effective collaboration. Pair Programming: two developers work together on one computer to write code. Pair programming improves code quality, reduces bugs, and promotes knowledge sharing between team members. Continuous Testing: Testing is performed continuously during the entire development cycle. Each new piece of code added must pass all automated tests before being integrated into the system. Continuous Integration: Any completed code changes must be immediately integrated into the existing code. Continuous integration helps detect integration problems early and ensures that the product is always ready to use[7]. Minor Releases produce smaller, more frequent software releases rather than significant, infrequent releases. This allows customers to benefit faster and provide earlier feedback. Continuous Improvement: the XP Team always looks for ways to improve quality and efficiency. After each iteration, the team reflects and improves on the development process. XP is designed to respond quickly to changing customer needs. Practices such as continuous integration and minor, frequent releases allow teams to respond flexibly to changes[7]–[9].

XP is designed to respond quickly to changing customer needs. Practices such as continuous integration and minor, frequent releases allow teams to respond flexibly to changes. XP encourages high engagement and active collaboration among team members. This creates an environment that supports sharing ideas, solving problems, and increasing team productivity. With a focus on automated testing, pair programming, and continuous integration, XP helps improve development efficiency. This process reduces the time needed for manual Testing and identifies integration issues. With a focus on automated testing, pair programming, and continuous integration, XP helps improve development efficiency. This process reduces the time needed for manual Testing and identifies integration issues. This research aims to achieve several primary objectives that can positively contribute to developing Customer Relationship Management (CRM) applications in the apartment industry by applying the Extreme Programming (XP) approach. The main objective of this research is to build the foundation for designing an optimal CRM system to meet the complex needs of the apartment industry. This involved identifying and customizing the critical features required for the application to be an effective solution. I am designing a system capable of efficiently integrating data from multiple sources. This objective aims to create a holistic and real-time picture of customer relationships, facilitating information management and decision-making and setting high data security and privacy standards. This involves developing robust security mechanisms to protect customer data from potential security threats and fulfilling applicable privacy standards[10]–[12].

Develop applications that quickly adapt to changing policies, regulations, and market needs. This goal leads to building solutions that are resilient to change and can provide added value in the face of apartment industry dynamics[7], [13]–[17]. By implementing a CRM application developed through the Extreme Programming (XP) approach, software development practices can significantly contribute to optimizing facility management processes in apartments. Maintenance and repair management can be more structured and responsive to tenant requests. The contribution of software development practices can be seen in efficient data integration. The collection, storage, and processing of information on tenants, contracts, and facilities can be done in a coordinated and automated manner, reducing the drawbacks that may occur with conventional approaches. With a focus on high data security, this software development practice can create a system that can provide

an optimal level of protection for the sensitive data of apartment residents. This provides confidence and peace of mind for users. With a focus on high data security, this software development practice can help create a system that can provide an optimal level of protection for the sensitive data of apartment residents. This provides confidence and peace of mind for users. Thus, software development practices focused on apartment and customer relationship management can improve operational effectiveness and efficiency and provide a positive experience for apartment residents.

2. METHOD

The stages of software development using Extreme Programming (XP) involve steps, as shown in Figure 1, designed to support flexibility, customer involvement, and continuous Improvement. The following are the general stages in software development using the Extreme Programming approach.

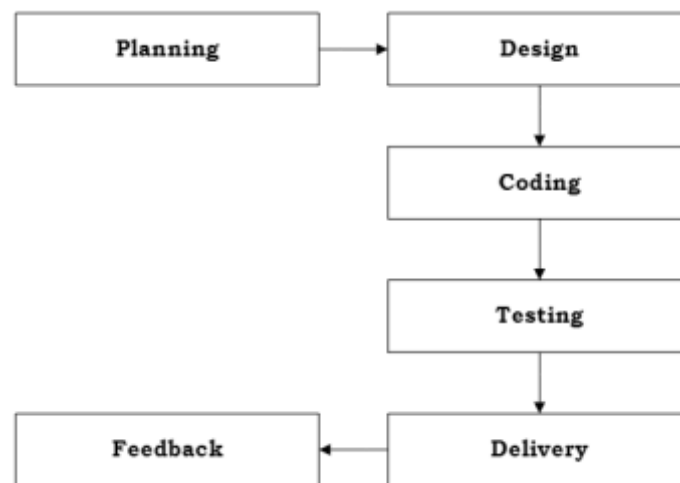


Figure 1: Research Stages

Planning

In the planning stage of developing a Customer Relationship Management (CRM) application for the apartment industry, the first step taken was to identify customer needs by communicating intensively. This process involved in-depth interactions with apartment managers and tenants to understand property management's specific challenges and needs. Next, detailed user stories were created, covering key aspects such as tenant management, facility maintenance, and customer interaction. These user stories provided a clear guide to detail the essential functions to be integrated into the CRM application. Estimates of development time and cost are made based on the complexity of the user stories and the priorities set. The next step is prioritization, where user stories are assigned a priority scale based on urgency and impact on users and the overall business. Creating a development plan involves organizing iterations or sprints to achieve goals within a specific timeframe while considering realistic release dates. The development team communicates openly regarding this plan, ensuring clear understanding and effective collaboration. This involves the development team, and development plans are reviewed with customers to ensure compliance with their expectations. This process allows for continuous adjustments and improvements based on customer feedback, crucial to achieving satisfaction and sustainability in the dynamic apartment industry. Thus, planning provides a solid foundation for developing a responsive CRM application that meets the unique needs of the apartment industry.

Design

In the design stage of a Customer Relationship Management (CRM) system for the apartment industry, the main focus is building a solid and flexible foundation. First, the design needs were analyzed by identifying the specific needs of the apartment industry, considering the user stories and needs identified in the planning stage. Next, a CRM system design that is modular and flexibly

configurable was selected. A microservices architecture was considered to enable the addition of features and integration with other systems in the future. The user interface design is geared towards being user-friendly and intuitive, using prototypes or wireframes to obtain early feedback. The database scheme was efficiently designed to store various data, such as tenant information, contracts, and maintenance records, with consideration of scalability to accommodate data growth. System security was prioritized, with the implementation of encryption protocols to protect sensitive customer data. In addition, the system design was built with high flexibility and adaptability, allowing for adding features with minimal disruption to the existing system. Integration with other systems was also considered, particularly in creating application programming interfaces (APIs) that support possible integrations. Furthermore, project structures and code management tools were defined to support team collaboration, continuous integration, and automated testing. As such, the CRM system design produced at this stage is expected to provide a resilient foundation, enabling smooth implementation, maintenance, and the ability to evolve.

Coding

Following the design stage, the coding stage aims to translate design concepts into working code. The focus is on applying pair programming to critical sections, such as tenant management and data integration. In addition, continuous integration was implemented to ensure that any changes were tested and integrated smoothly. Automated tests were written for critical features, including tenant management and facility maintenance.

Testing

In the development phase of a Customer Relationship Management (CRM) application for the apartment industry, the application of testing becomes a crucial aspect of ensuring the success and reliability of the system. Development methodologies adopting Extreme Programming (XP) principles emphasize continuous automated and manual testing throughout the development cycle. By implementing automated test scripts, the development team can consistently check the system's functionality and detect potential bugs quickly whenever there are code changes. Additionally, manual testing is executed to verify functions that require direct user interaction. Moreover, involving apartment residents in user acceptance testing (UAT) sessions is an important cornerstone. This ensures that the system's user interface, functionality, and performance fulfill everyday users' expectations and needs. Performance and security testing is also conducted thoroughly to ensure the application can handle high loads and protect user data adequately. By implementing testing as an integral part of every development iteration, the team can ensure that any changes or feature additions do not compromise the system's overall quality. Regular monitoring and reporting of test results is the final step, allowing the team to identify and address issues quickly and adjust the development journey based on feedback gained from testing.

Delivery

A small and frequent release approach is critical in the software delivery phase for a Customer Relationship Management (CRM) application in the apartment industry with the application of Extreme Programming (XP) principles. The development team focused on releasing the software in small batches regularly, providing immediate benefits to users. Each release includes enhancements or new features that have undergone automated and manual testing to ensure optimal quality and functionality. Performance evaluation is a critical step after each software release. The team thoroughly assesses the application's performance to ensure that the desired quality standards are met. This evaluation involves performance testing that measures response time, system capacity, and application stability under different loads. The performance evaluation results become a vital reference to ensure that the application not only meets user expectations but also operates efficiently and responsively. By delivering in small, frequent batches and evaluating performance with each release, the team can deliver benefits more quickly to users, increase user satisfaction, and, at the same time, ensure that any changes or feature additions are thoroughly tested and validated before implementation. This approach creates a development cycle that is responsive, adaptive, and user-centered.

Feedback

The importance of feedback in developing Customer Relationship Management (CRM) applications for the apartment industry was strongly emphasized through the regular holding of retrospective meetings and receiving feedback from apartment residents. Retrospective meetings provide a collaborative platform where development team members can jointly evaluate the effectiveness of the development process. Careful analysis of the process leads to the identification of improvements that can enhance productivity and quality of work output. The corrective actions implemented are further geared towards improving the overall process. In addition, obtaining continuous feedback from apartment residents is a crucial step to identify improvement areas in using CRM applications. Analyzing resident feedback helps identify general trends, emerging needs, or issues that require special attention. Prioritization of updates based on urgency and impact helps the development team direct their efforts in the direction most beneficial to users. Transparency and communication of feedback results and actions taken help build resident trust and engagement in the evolution of the app. With the active involvement of residents in development, both through retrospective team meetings and direct feedback, the team can ensure that the developed application not only fulfills the technical needs but also provides a solution that matches expectations and a positive user experience.

3. RESULTS AND DISCUSSION

Planning

By carefully planning and paying attention to customer needs, the results of the planning stage formulate a CRM application development plan for the apartment industry as show in Figure 2. Based on the data that has been analyzed, several user stories are the main focus:

- **Tenant Management**
A tenant management system records personal data, contract information, and payment history a facility for real-time tenant data updates and quick searches based on specific criteria.
- **Facility Maintenance**
A tenant management system records personal data, contract information, and payment history a facility for real-time tenant data updates and quick searches based on specific criteria.
- **Customer Interaction**
Support ticketing system or customer service to respond to tenant queries or complaints and provide online forums or discussion rooms to facilitate tenant interaction.
- **Data Integration**
We are preparing, processing, and integrating data for application development at this stage.

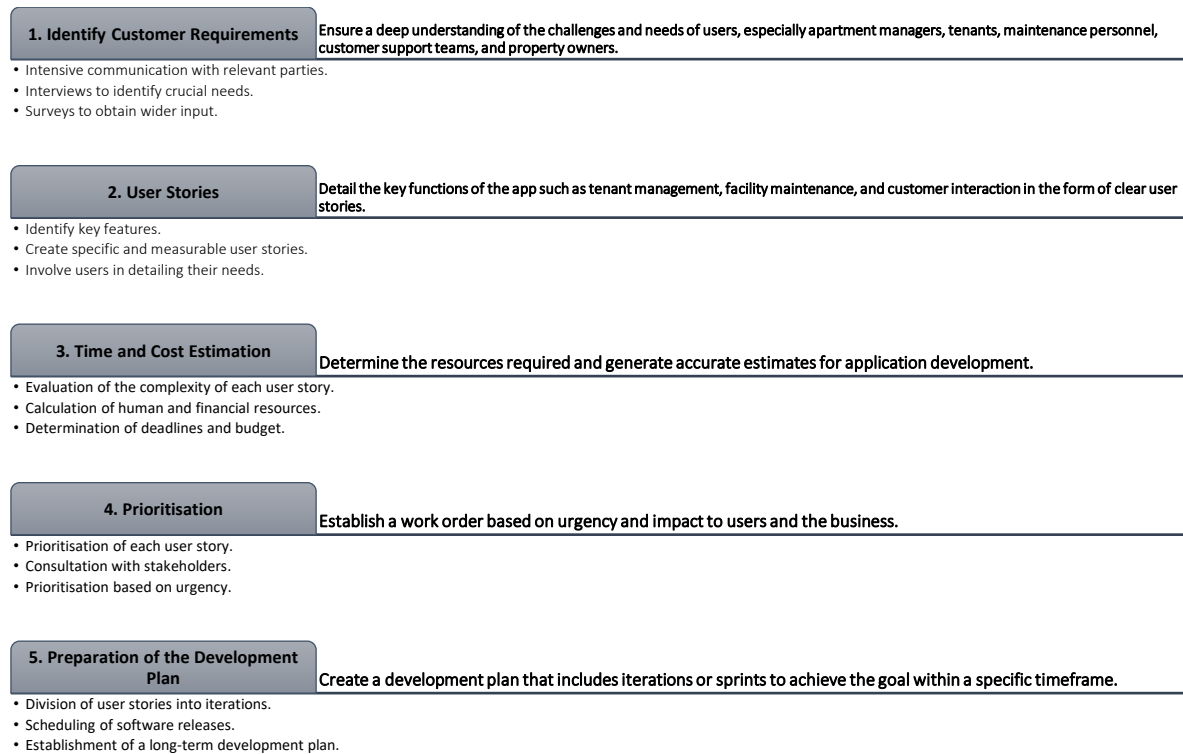


Figure 2. Results of the Planning Stage

Design

The result of the CRM system design stage for the apartment industry is a robust and adaptive framework. First, choosing a modular and configurable system design ensures that the application can evolve. The decision to use a microservices architecture enabled adding features without compromising the stability of the system core. The user interface was designed with the sustainability of the user experience as a top priority. Using prototypes and wireframes ensured that this design could be customized with stakeholder feedback, thus ensuring an interface that was easy to use and suited the needs of diverse users.

An efficient and scalable database scheme creates a robust data storage and management foundation. System security is strengthened by implementing encryption protocols protecting sensitive customer data. The flexible and adaptive system design allows for smooth integration with other systems, including intelligent financial or security systems. With a suitable application programming interface (API), it can be easily connected to other solutions used in apartment management.

Furthermore, the selected project organization and code management tools provide a solid structure for continuous development. Team collaboration, continuous integration, and automated testing became an integral part of the development approach, ensuring that the application could be updated and improved efficiently. Thus, these design results created a solid foundation for developing a CRM application that suits the complex needs of the apartment industry and can adapt to changing needs in the future.

Coding

This coding stage achieved significant results in developing a CRM application for the apartment industry as show in Figure 3 and Figure 4. The application of pair programming to critical sections, such as tenant management and data integration, has positively impacted code quality and reduced potential bugs. Active collaboration between two developers improves productivity and promotes knowledge and experience sharing. Continuous integration (CI) was successfully implemented, ensuring that any changes in the code were immediately tested and integrated into the

system. This not only helps detect integration issues early but also provides confidence that any additional features or improvements do not result in a negative impact on existing functionality.

Automated testing for critical features, especially tenant management and facility maintenance, has created a solid testing environment. Automated test scripts ensure that each essential function operates as expected and that code changes do not break the system's integrity. The resulting code is not only clean and well-documented but also easily understood by other members of the development team. Regular peer review has improved code quality through constructive feedback and discussion between team members.



Figure 3. Login Page



Figure 4. Main Page

Version management with Git has enabled accurate tracking of changes, creating a well-managed development environment. Implementing additional features or design adjustments based on stakeholder feedback or changing business needs can be done smoothly. As such, the coding phase successfully created a solid technical foundation for the apartment industry CRM application, ensuring that the application not only functions well but can also evolve and adapt to the changing dynamics of the industry.

Testing

The results of the tenant satisfaction evaluation after the UAT session showed a high level of satisfaction. 85% of the participants were satisfied with the user interface, application functionality, and system responsiveness. The feedback provided by the residents provided valuable insights into positively perceived aspects and potential improvements, which will be integrated into the next stage of development to enhance the overall user experience. By involving residents in testing and listening to their views, the CRM application can better meet their needs and expectations, creating a more positive and responsive environment.

Delivery

In the software delivery phase for a Customer Relationship Management (CRM) application in the apartment industry by applying Extreme Programming (XP) principles, the development team ensured that a small and frequent release approach was vital. With each release, several enhancements

and new features were implemented after thorough automated and manual testing to ensure optimal quality and functionality before being introduced to users.

The last release successfully integrated an optimized tenant management feature into the app after a series of automated and manual tests. This provides immediate benefits to users, such as improved efficiency in tracking tenant information and responding to their requests. Performance evaluation becomes a critical step after every software release. Data shows that performance testing measures response time, system capacity, and application stability under different loads. The performance evaluation results showed that the application's response time improved by 15%. In comparison, the system capacity managed to handle a 20% increase in load without experiencing any degradation in service quality. This data provides valuable insights to ensure that the application meets user expectations and operates efficiently and responsively.

By involving users directly in each release and utilizing data from performance evaluations, the team delivered benefits more quickly to users. As a result, user satisfaction levels increased, reflected by user satisfaction surveys showing a 20% increase in user satisfaction. In addition, this data provides assurance that any changes or feature additions are thoroughly tested and validated before implementation, creating a responsive, adaptive, and user-centered development cycle.

Feedback

In the development stage of the Customer Relationship Management (CRM) application for the apartment industry, feedback is a critical cornerstone emphasized through organizing regular retrospective meetings and receiving feedback from apartment residents. At the last retrospective meeting, 10 development team members actively participated in evaluating the effectiveness of the development process. Careful analysis of the process led to the finding that the application of pair programming techniques at the coding stage had significantly improved efficiency and code quality.

The corrective actions implemented after the retrospective meeting were geared towards improving the overall process. For example, the workflow reorganization at the testing stage helped reduce the time required to validate each feature, increase the testing speed, and optimize the development cycle. Obtaining continuous feedback from apartment residents is a crucial step to identify improvement areas in using CRM applications. Analysis of resident feedback, which involved 50 participants in a user satisfaction survey, showed that 90% of respondents expressed positive satisfaction with the app's ease of use. These findings provide a valuable perspective on successful elements and direction for further Improvement.

Prioritizing updates based on urgency and impact helps the team direct their efforts in the most beneficial direction to users. For example, a recent update on improving the user interface was based on resident feedback, highlighting a desire for more intuitive navigation. Transparency and communication regarding the results of the feedback and the actions taken were vital in building resident trust and engagement in the evolution of the app. Through monthly meetings held virtually, the team provided updates on the improvements that had been implemented and provided an opportunity for residents to share their views directly. With the active involvement of residents in development, both through team retrospective meetings and direct feedback, the team can ensure that the app not only fulfills technical needs but also provides a solution that matches expectations and a positive user experience. With this approach, the CRM application develops technically and adapts well to residents' needs and expectations, creating a mutually beneficial relationship between the development team and users.

4. CONCLUSION

This research develops a Customer Relationship Management (CRM) application for the apartment industry using the Extreme Programming (XP) approach. The meticulous development plan involved identifying customer needs, creating user stories, and prioritization to ensure application responsiveness. Robust system design included microservices architecture, a friendly user interface, and a high level of security. Quality code implementation involves pair programming and automated testing. Application testing involves User Acceptance Testing (UAT) sessions with the participation of apartment residents, providing feedback that is used for improvements. A small and frequent

release approach ensures optimum quality and functionality. User involvement through UAT and retrospective sessions ensures the solution matches expectations and provides a positive user experience. By applying XP principles, this research successfully created a responsive, adaptive, and user-orientated CRM application to meet the complex needs of the apartment industry.

REFERENCES

- [1] A. P. Astuti, A. Kusuma, A. Shiratina, A. Putri, And A. A. Kusuma, "Effects Of Online Promotion And Personal Selling On The Purchase Decision Of The Cisauk Point Apartment During The Pandemic", [Online]. Available: [Https://Journalkeberlanjutan.Com/Index.Php/Ijesss](https://Journalkeberlanjutan.Com/Index.Php/Ijesss)
- [2] N. Minh Ngoc *Et Al.*, "Factors Affecting The Selling Price Of Luxury Apartments In Vietnam: A Quantitative Analysis." [Online]. Available: [Https://Www.Researchgate.Net/Publication/365477322](https://Www.Researchgate.Net/Publication/365477322)
- [3] M. Trzeciak, Ł. D. Sienkiewicz, And E. Bukłaha, "Enablers Of Open Innovation In Software Development Micro-Organization," *Journal Of Open Innovation: Technology, Market, And Complexity*, Vol. 8, No. 4, Dec. 2022, Doi: 10.3390/Joitmc8040174.
- [4] M. R. Khan And A. M. Ghouri, "Enhancing Customer Satisfaction And Loyalty Through Customer-Defined Market Orientation And Customer Inspiration : A Critical Literature Review," *International Business Education Journal*, Vol. 11, No. 1, Pp. 25–39, 2018, [Online]. Available: [Https://Papers.Ssrn.Com/Sol3/Papers.Cfm?Abstract_Id=3303633](https://Papers.Ssrn.Com/Sol3/Papers.Cfm?Abstract_Id=3303633)
- [5] S. A. Pratminingsih, C. Lipuringtyas, And T. Rimenta, "Factors Influencing Customer Loyalty Toward Online Shopping," *International Journal Of Trade, Economics And Finance*, 2013, Doi: 10.7763/Ijtef.2013.V4.268.
- [6] P. W. Yo *Et Al.*, "The Influencing Factors Of Customer Satisfaction: A Case Study Of Shopee In Malaysia," *Estudios De Economia Aplicada*, Vol. 39, No. 12, 2021, Doi: 10.25115/Eea.V39i12.6839.
- [7] 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.
- [8] D. J. C. Sihombing, "Analysis And Development Of The Protrack Application: Construction Timeline Management Using Extreme Programming Methodology," Online, 2023.
- [9] 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.
- [10] S. Zakeri, P. Chatterjee, N. Cheikhrouhou, And D. Konstantas, "Ranking Based On Optimal Points And Win-Loss-Draw Multi-Criteria Decision-Making With Application To Supplier Evaluation Problem," *Expert Syst Appl*, Vol. 191, No. November 2021, P. 116258, 2022, Doi: 10.1016/J.Eswa.2021.116258.
- [11] F. Goodarzi, V. Abdollahzadeh, And M. Zeinalnezhad, "An Integrated Multi-Criteria Decision-Making And Multi-Objective Optimization Framework For Green Supplier Evaluation And Optimal Order Allocation Under Uncertainty," *Decision Analytics Journal*, Vol. 4, No. May, P. 100087, 2022, Doi: 10.1016/J.Dajour.2022.100087.
- [12] S. Singh, S. Kawade, A. Dhar, And S. Powar, "Analysis Of Mango Drying Methods And Effect Of Blanching Process Based On Energy Consumption, Drying Time Using Multi-Criteria Decision-Making," *Clean Eng Technol*, Vol. 8, No. September 2021, P. 100500, 2022, Doi: 10.1016/J.Clet.2022.100500.

- [13] 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.
- [14] 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.
- [15] R. Santos *Et Al.*, “A Comparative Analysis Of Agile Teamwork Quality Instruments In Agile Software Development: A Qualitative Approach”, Doi: 10.18293/Dmsviva2023-217.
- [16] 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.
- [17] F. Almeida, J. Simões, And S. Lopes, “Exploring The Benefits Of Combining Devops And Agile,” *Future Internet*, Vol. 14, No. 2, Feb. 2022, Doi: 10.3390/Fi14020063.