Developing Ordering Management System in an Indonesian Multinational Packaging Machinery Company

Samantha Tiara Widjaja[1(✉)], Tanika Dewi Sofianti[2], and Aditya Tirta Paratama[2]

[1] Master of Mechanical Engineering, Engineering Management, Swiss German University, Tangerang, Indonesia
samantha.tiara@yahoo.com

[2] Department of Industrial Engineering, Swiss German University, Tangerang, Indonesia

Abstract. Ordering Management System (OMS) is the fundamental system that is required by companies that sell goods. A company that sells packaging machinery require the development of their information system and their ordering management system. In this research, the company is still using the traditional method for handling their ordering management system to minimize lead time. The company wants to improve their ordering management system to help them in future inventory control development. This research aims to develop a web-based application that can allow the company to have more structured data for future improvement in the company. The chosen methodology for this research is an agile methodology, which consists of six steps, requirements listing, planning, design, development, deployment, and assessment. The research found that to perform management improvement, the company has to own a detailed and structured database that keeps the track record of their business activities. The developed application provides the company with a system that allows the company to auto-generate several processes and keep the information in the database that can be retrieved whenever it is necessary. Thus, the result from user acceptance test showed that the developed application eliminates several business processes in the company to become more efficient.

Keywords: Agile Methodology · Information System Development · Ordering Management System · User Acceptance Test · Web-based Application

1 Introduction

Packaging is the most essential part of a business that plays a big role in the world economy. There are several types of packaging depending on the business type, such as plastic boxes, plastic bottles, glass bottles, paper cups, etc. Packaging is also a crucial part of daily aspects to organize things, keeping materials from contamination, etc. In business, packaging can be found everywhere as the final touch of the product. Packaging also becomes the way for businesses to improve their product or brand impression to their clients. The quality of packaging will reassure clients that the product inside the package is safe and is protected in its prime condition.

G. W. Bhawika et al. (Eds.): IConBEM 2022, AEBMR 249, pp. 203–216, 2023.
https://doi.org/10.2991/978-94-6463-216-3_16

The packaging industry has always been projected to increase every year. The European packaging market in 2018 produced €195 billion turnover with an average annual growth rate (AAGR) of 2% with the increasing market projection to €214 billion turnover by 2023. The global market size in the packaging industry is also projected to increase by more than $1 trillion by 2023. Further projection in global packaging market size has also been projected to keep on increasing until 2028 with the additional $150 billion (Abejón, Bala, Vázquez-Rowe, Aldaco, & Fullana-i-Palmer, 2020).

The packaging market in Indonesia is also projected to keep increasing by the average annual growth rate (AAGR) of 8% from 2014 to 2020. The development in the services and industrial business is the largest type of business in Indonesia (Indonesian Packaging Federation, 2019).

The increasing demand in the packaging industry leads to the increasing manufacturing sector. Manufacturers in the packaging industry are pressured to increase their performances to compete with other manufacturers and to fulfil customer demands. Thus, companies are pressured to improve their management system and their information system. The management system and information system of the company have to cover all of the company activities within the company and the transactions with the customers. The company has to analyze the transaction history with the customers, the manufacturing and the delivery process of the ordered spare parts to perform improvement.

2 Literature Review

Software development or information system development requires a strong foundation in the specification of the software. Many software development incidents happened due to the lack of clarity in its software requirements. Thus, developing software requires a document that specifies the software requirements before executing the software development process. Building SRS also requires an in-depth analysis of the business process to ensure the developed software meets its purpose.

2.1 Information System Development (ISD)

A study found that developing an information system (IS) requires direct interaction between the users to ensure communication clarity. Developing IS in industries also requires a lot of interactions between each department of the company. The communication in ISD must ensure that it is two-way communication. Each department has to contribute to the communication to integrate data (Taipalus, Seppänen, & Pirhonen, 2020).

2.2 Requirements Engineering (RE)

Requirements engineering (RE) is an engineering design process that produces a document for the development process in engineering. One of the documents that can be generated using RE is the SRS document (Oktaviyani, 2020). This document will serve as the initial step of building a system. Thus, defining RE must be systematically and structurally established to ensure its relevance (Curcio, Navarro, Malucelli, & Reinehr, 2018).

2.3 Software Requirements Specification (SRS)

Software requirements specification (SRS) is a document that specifies programs to perform specific functions (Medeiros, Vasconcelos, Silva, & Goulão, 2018). The building or creating SRS requires an in-depth analysis of its concept and understanding of its structure. Several standardized guidelines can be followed in developing SRS structures, such as ISO, IEC, and IEEE. The created SRS can be verified and validated using the requirement quality verification (RQV) tool (Boyarchuk, Pavlova, Bodnar, & Lopatto, 2020).

Generating an SRS document requires a meticulous process to ensure the clarity of the document when given to the software developer. SRS document must possess consistent, complete, and unambiguous wordings, meaning that developing an SRS document requires excellent literacy in describing software to prevent failure in the software development process (Oktaviyani, 2020).

2.4 Software Development Life Cycle (SDLC)

A software development life cycle (SDLC) is a framework used by software developers to defines the development process from the beginning until the end of its cycle (de Vicente Mohino, Bermejo Higuera, Bermejo Higuera, & Sicilia Montalvo, 2019). Generating design documents is an essential and critical part of SDLC. The SDLC process involves six steps that begin with requirement analysis. This step is followed by planning the desired software development, designing and building the software (Kumar & Rashid, 2018). The established software must be tested to verify and validate its properties. The testing process also ensures the functionality of the developed software and ensures that the developed software meets the requirements. The last step will be deploying the software or maintaining the functionality of the developed software (Karunajeewa, 2020) (Fig. 1).

There are five models in implementing SDLC, which are agile, waterfall, v-shaped, iterative, and spiral methodology (Marsner, 2019). The most common methodology used

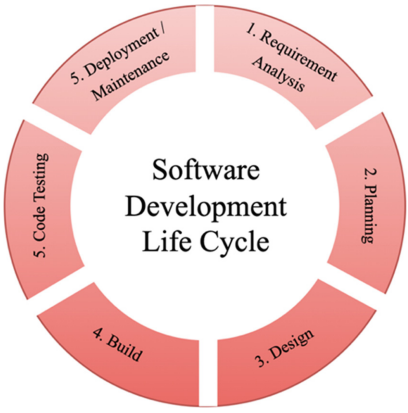

Fig. 1. Process of Software Development Life Cycle (Ergasheva & Kruglov, 2020).

for software development projects is agile methodology and waterfall methodology. In 2019, a study showed that agile methodology performs 16% better in successful rate compared to waterfall methodology and 13% less in failure rate. However, the use of agile methodology is not always suitable for every software development project (Boyarchuk et al., 2020).

2.5 Unified Modelling Language (UML)

The Unified Modelling Language (UML) is a model that visualizes and specify a software system. It has the functionality to help users understand the purpose of the system (Fauzan, Siahaan, Rochimah, & Triandini, 2018). UML is a tool to define discrete systems, such as software, firmware, and digital logic. It is used as a representation of a system that wants to be developed. It can also be used for guidance towards system specifications (Jacobson & Booch, 2021).

UML diagram can be classified into structure diagram, behavior diagram, and interaction diagram. The structure diagram consists of a class diagram, component diagram, object diagram, profile diagram, a composite structure diagram, deployment diagram, package diagram (Maylawati, Darmalaksana, & Ramdhani, 2018). The behavior diagram consists of an activity diagram, use case diagram, state machine diagram, and interaction diagram. Interaction diagram consists of a sequence diagram, communication diagram, interaction overview diagram, and timing diagram (Pradigm, 2018).

2.6 User Acceptance Test (UAT)

User Acceptance Test (UAT) is an assessment that must be addressed to the end-user before the software can be launched (Sualim, Yassin, & Mohamad, 2017). The assessment includes the features compatibility of the developed software with the company needs. UAT is performed at the end of the development process that will become the final assessment in the development process after functional testing and system testing are done (Yunanto, 2018).

However, conducting UAT in the agile methodology of software development faces several problems in time, motivation, and knowledge. Getting the end user to invest their time analyzing the developed software is difficult, especially when there is a large number of end-user and stakeholders (Long, 2020). Conducting UAT in an agile environment could create less motivation for the developers, especially in a long-term and high-pressure project. The invested time to analyze UAT might not result in a good output due to several users who might prefer the traditional method (Otaduy & Diaz, 2017).

3 Research Methodology

The chosen research methodology for this research was using SDLC with the agile software development methodology. The methodology was chosen due to its flexibility of requirements changes in the development process. The chosen methodology also allows the development process to focus on customer satisfaction if the application (Fig. 2).

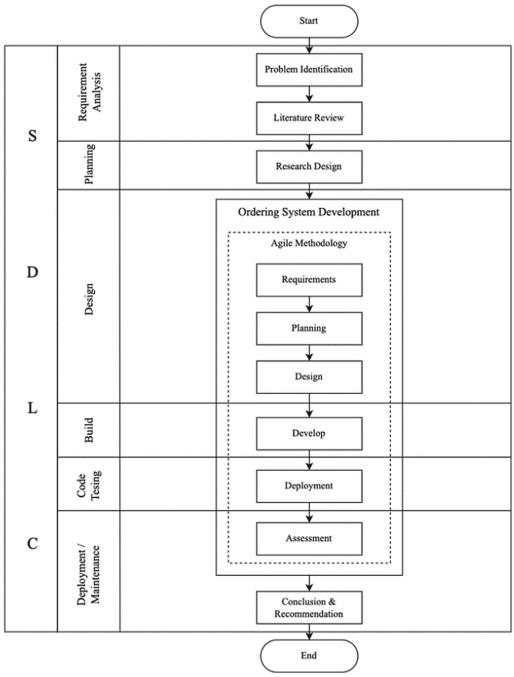

Fig. 2. Research methodology using SDLC and agile software development method.

The chosen methodology consists of requirements analysis, planning, designing, building, testing, and deployment. The requirements analysis process was done by identifying problems and literature review. The planning process was done by research design. The designing process used an agile methodology that consists of listing the application requirements, planning the design, and creating the design. The building process was done by developing the application based on the design. The testing process was done by deploying the developed application. The deployment process was done by assessing the developed application.

4 Result and Discussion

4.1 Company Profile

The research is conducted in a European packaging machinery & equipment company that has worked in this industry for years. The company has many branches all around the world that specialize in manufacturing packaging machinery. The company's business process will involve the parent company that is headquartered in Germany, a subsidiary company of the Indonesia branch, and customer companies in Indonesia.

4.2 Problem Identification

The company has several problems in the business process. The company has difficulty in predicting the customer demand and delivering it on time to the customers. The

company mainly sells spare parts of their machines with its maintenance service. The customers who buy a machine from the company have to use its service in maintaining its performance. This is due to the spare parts of the machines being unique and cannot be replaced by other brands. Replacing machine parts with another brand will risk the production performance and the performance of the machine itself.

The ordered spare parts are being manufactured and sent from the headquarter. The company has an average lead-time of 4 weeks or less in a normal transaction. However, several parts might be out of stock and require more time in the delivery process. The company has a system to display the availability of the parts. However, the system can only be accessed internally. Thus, the stock availability and the estimated time of arrival of the parts can only be seen after the customer places a booking or an order.

4.3 Research Design

4.3.1 Current Business Process

The current process begins with the quotation request from the customer to the company. The responsible person for handling the quotation request is still not defined. The request can be made through the sales manager, service manager, or sales administrator. However, requests that were made through the sales manager and service manager will be forwarded to the sales administrator. The sales administrator will process the request by checking the spare parts availability and price list of the requested items in the company's system.

The sales administrator will inform the customer by email regarding the availability of the spare parts and the price list of the spare parts. The customers can complete their quotation and send it back to the company as a purchase order. The purchase order sent by the customer must be confirmed by the director before it can be sent to the headquarter as a complete purchase order from the customer.

After the order has been confirmed by the director, then the sales administrator will have to retype each spare part number, machine type, and quantity to the system. The headquarter will receive order notification by the system and will immediately send the order confirmation with the complete estimated time of arrival. The estimated time of arrival that will be noted in the order confirmation from headquarters might differ from the first estimated time of arrival that has been informed to the customer. The common delivery process after headquarter receive the purchase order is between 2–4 weeks. However, several items might be unavailable during that time. This causes prolonged delivery lead-time due to the production time that may vary according to the ordered items.

Once the ordered items are ready to be delivered, the headquarter will immediately send the softcopy file of the invoice and delivery note to the Indonesia branch. The hard copy of the file will be given to the courier as they pick the items up to be shipped. The courier will prepare the shipping process and will notify the Indonesia branch regarding the shipping information and tracking number of the item that can be traced through their website.

Customers that wish to know the shipping status of the ordered items have the option to ask the company or make a request to obtain the item's tracking number. Customers

will not be notified regarding the tracking number or the shipping status unless they make a request for it. Customers will only be notified when the items have already arrived at Indonesia's airport. Thus, customers can prepare for the maintenance schedule that will be given by the service manager.

4.3.2 Proposed Business Process

Requests from customers will be directed to the sales admin. The sales manager and service manager will not handle customer requests. Sales admin will be able to directly input the necessary information in the application. The completed requests will be able to become purchase orders after the quotation has been confirmed by the customer. Confirmation from the director will be auto-generated and the purchase order can be processed by the sales admin.

Sales admin will receive an order confirmation from headquarter and delivery notes from headquarter after processing customer orders. Sales admin will be able to input the necessary data and item's tracking number to the application. The necessary data will be displayed to the customers. Thus, the customer does not have to reach out to the company to obtain the necessary information regarding their order.

Delivery notes will be generated after the item's arrival and ready to be shipped to the customer site. The customer will have to approve the delivery note after maintenance has been completed. The invoice will be auto-generated to the customer after maintenance completion. There will be a place for customers to upload their payment proof in the application to complete their transaction process with the company.

The proposed improvement will change the ordering and approval process to be done by the system. This will also allow the company to store the transaction data and retrieve the transaction data whenever it is necessary. However, there are several processes in the company that cannot be modified due to the company system that connects the branch to the headquarter. This is also due to the access limitation to connect the developed system to the company's system. Thus, several processes must remain manually received and inputted.

4.4 Ordering System Development

4.4.1 Requirements

The requirements list of this project was defined based on the company needs for their database storage based on the daily transaction with the customers. The company has several requirements regarding their databases that will be included in the system, such as spare part name, spare part number, machine type, quotation, delivery time, harmony system/commodity code, amount per piece (IDR & EUR), total amount (IDR & EUR), purchase order, delivery note, delivery tracking number, discounts, order confirmation, invoice.

4.4.2 Planning

The research was conducted in three months. The research plan was created to manage the project timeline. The first month was used to identify the system's problem in the

company. The second month was used to set application requirements, design development, and application development. The third month was used for system testing and application assessment.

4.4.3 Design

The designing process consists of several processes. The application design began by creating a UML diagram, such as a use case diagram, activity diagram and sequence diagram. Use case diagram was made to explain the interaction between the user and the application. The application user consists of the customers and the company's employees. An activity diagram was made to provide an activity explanation of the developed use case diagram. The activity diagram provides information regarding the system activity details from the beginning until the end.

A sequence diagram was made to explain the objects that will be included in the application. There was customer, system, supplier, and database. The sequence begins with the customer logging in to the system and creating an order. The supplier follows by inputting the necessary information to the customer based on the order. The sequence continues until the customer completes the payment.

The designing phase continued to the application user interface and user experience (UI/UX) design, and development of SRS document. UI/UX design and SRS document were completed before entering the development phase. However, the designing process continues during the development process to ensure the application compatibility with the requirements. The SRS document explains the system designs of the web-based application. This document explains further information regarding the detailed information of functions, interface, and other resources that were used in developing the application (Figs. 3, 4 and 5).

4.4.4 Development

The application development process began by analyzing design documents and turning them into a programming language. The development process emphasizes the main feature function of the application. There are two access paths to the application. Customers and admin have a different links to access the application.

The application development process referred to the previously developed UI/UX designs. The development process began by developing the customer view and continuing to the company view. The development process of the customer view and the company view was done separately by initially completing each view. The application for customer view and company view was furtherly interfaced. However, there were several changes in the designs during the development process to simplify and shorten the development process due to the limited time in developing the application.

The application development began with developing the customer application view. Customer application view consists of a login page, home page, online booking page, booking plan page, order history page, and account page. Meanwhile, the application development of the company view consists of a login page, customer orders history page, insight page to download database, and account page.

Fig. 3. Use Case Diagram of the Developed Application.

4.4.5 Deployment

The developed application was deployed after the development process has been completed. There was customer view application deployment, company view application deployment, and the whole completed application deployment. The first deployment was the customer view. However, it is still incomplete due to the undeveloped company view that must be interfaced with the customer view. Thus, several functions in the application were still not defined in this phase.

Furthermore, the completed company view that has been interfaced with the customer view was deployed as a whole functioning application. In this phase, the whole completed application can immediately be released for the overall application testing. The deployed application is ready to be assessed at this stage.

4.4.6 Assessment

The assessment process continues throughout the development process after the application is ready to be deployed. There was assessment in the development process and there was also UAT. Assessment that was done during the development process was done to validate the function of the application features, whereas UAT was done to validate the overall application functionality is compatible with the company needs.

The primary assessment result referred to the UAT checklist that will be filled by the user. The user of the application will be the company and their customers. However,

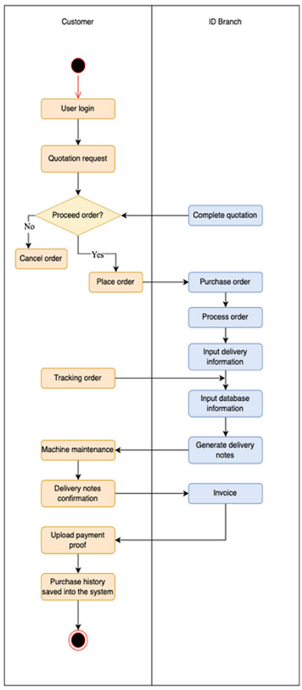

Fig. 4. Activity Diagram of the Developed Application.

this application emphasizes the business process improvement for the company. Thus, the UAT checklist focuses on the application used for the company's admin.

The application was reviewed by the director and the spare parts administrator of the company. The result showed that the developed application covers all business processes and activities in the company. The application makes daily tasks more efficient. The user can log in to the application using the registered email. The user can update their personal information that will be displayed in the transaction forms. The user can view the transaction history and the order status in the application. The user can generate and download new necessary documents in the application.

The developed application provides a transaction database that shows separate information regarding the transaction between the company and the customers, with the transaction between the company and the headquarter. The data record provides codes and dates for every transaction. The data record also allows the company to view customer orders based on the customer's name/company.

Based on the conducted UAT, the feedback from the user is well. The developed application showed efficiency in handling customer orders. Customers are no longer required to create order forms by themselves. Thus, the developed application allows the company to eliminate several validation processes that were done manually. The developed application also allows the company to send customer orders/requests directly to the responsible person. The developed application eliminates the necessity of the company to input the ongoing or completed transaction to their database.

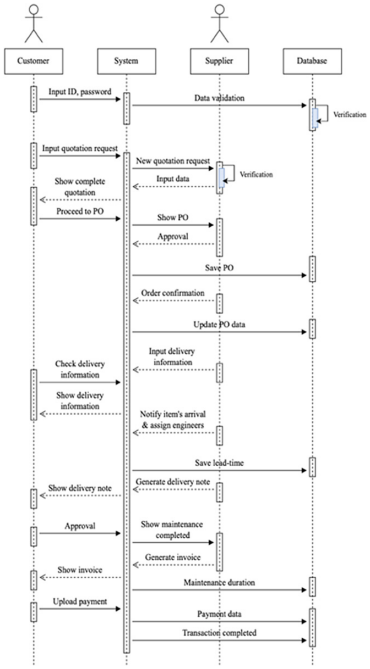

Fig. 5. Sequence Diagram of the Developed Application.

However, the application still requires further development in the application features, UI/UX, and security system. Application features still require improvements to ensure compatibility with the business processes, such as booking item modifications, customer support, real-time display feature, etc.

It is important to add customer support features to the application, such as a chatting feature for the customers to reach out to the company whenever they need assistance. Thus, customers can receive immediate support, primarily in urgent situations. UI/UX design of the application still requires improvements to ensure customer satisfaction in the better ordering process and obtaining the necessary information of the transaction. The security system also still requires improvements to ensure data confidentiality of the user, such as addresses, price, signature, etc.

5 Conclusion and Recommendation

5.1 Conclusion

The developed application improved the company system by integrating their transaction data history and simplifying the current business process. The business process of the company can be shifted to become auto-generated by the system. This system minimizes input errors and requires fewer tasks validating data accuracy from customer requests. The system allows customers and the company to create a faster ordering system without requiring the customers and the company to create long emails and forms in creating and

responding to customer requests. The system also allows the company to easily view and select order history in the system. The system has a database that will store their transaction information as processed data. The developed system serves as a purpose in providing the company with accessible historical data for further management system improvement.

The developed application creates several changes in the company business process. Several processes were eliminated due to the auto-generation feature in the application. The UAT result shows that the main features of the application cover all ordering process activity in the company. The developed application simplifies the business activities to become more efficient. The system helps the company in viewing and storing order history based on customers names. The developed system also reduces processing time by directing customer orders to be handled directly by the responsible person in the company. Employee job functions in handling customer orders become more defined, which will be handled only by the spare parts administrator. Employee workloads are also being reduced by eliminating the necessity of validating orders, generating new documents, and generating database due to the application auto-generation feature. Generating new documents and database are automatically done based on customer order.

5.2 Recommendation

Further study is still required to improve the quality of the application performance. Improvements in the application layout should be performed to improve customer satisfaction in using the application. Several features should be added to the application, such as a chatting feature for the customer to communicate with the company easily. The security system of the application still requires several improvements to ensure the data confidentiality of the company. The application should undergo beta testing with real data cases to ensure the application efficiency and effectiveness.

The company should perform improvements for their maintenance performance. The company should perform preventive maintenance for each machine. The company can begin by registering the lifespan of each spare part. Furthermore, the company should create a maintenance book for each machine. The book must consist of the spare parts list required by the machine and the latest repair date of each spare part. Moreover, the company shall add a maintenance notification feature in the developed application. The feature will help the company and the customer to reorder new spare parts before reaching the wear-out phase. The feature will also help the engineer's performance in the machine assessment process.

Acknowledgement. Countless people have supported my journey in completing this work. The assistance of Mr Antoni Wijaya and Ms Leoni Andriyati was also greatly appreciated as the director and the sales administrator of the company. Their support and encouragement make this project successful by providing the necessary data to be analyzed.

I wish to extend my special thanks to Angellina Hermawan that have introduced me to Gifhary Syidhqa Hamim, who helped me a lot as a developer expert in this project. His support and guidance make this project successful by providing the necessary information regarding the software development process.

References

Abejón, R., Bala, A., Vázquez-Rowe, I., Aldaco, R., & Fullana-i-Palmer, P. (2020). When plastic packaging should be preferred: Life cycle analysis of packages for fruit and vegetable distribution in the Spanish peninsular market. Resources, Conservation and Recycling, 155, 104666. https://doi.org/https://doi.org/10.1016/j.resconrec.2019.104666

Boyarchuk, A., Pavlova, O., Bodnar, M., & Lopatto, I. (2020). Approach to the Analysis of Software Requirements Specification on Its Structure Correctness. IntelITSIS, 85–95.

Curcio, K., Navarro, T., Malucelli, A., & Reinehr, S. (2018). Requirements engineering: A systematic mapping study in agile software development. Journal of Systems and Software, 139, 32–50. https://doi.org/https://doi.org/10.1016/J.JSS.2018.01.036

de Vicente Mohino, J., Bermejo Higuera, J., Bermejo Higuera, J. R., & Sicilia Montalvo, J. A. (2019). The application of a new secure software development life cycle (S-SDLC) with agile methodologies. Electronics, 8(11), 1218.

Ergasheva, S., & Kruglov, A. (2020). Software Development Life Cycle early phases and quality metrics: A Systematic Literature Review. Journal of Physics: Conference Series, 1694(1), 12007. IOP Publishing.

Fauzan, R., Siahaan, D., Rochimah, S., & Triandini, E. (2018). Activity diagram similarity measurement: A different approach. 2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 601–605. IEEE.

Indonesian Packaging Federation. (2019). Packaging Industries in Indonesia. Retrieved from https://packindo.org/2019/04/08/packaging-in-indonesia/

Jacobson, L., & Booch, J. R. G. (2021). The unified modeling language reference manual.

Karunajeewa, S. (2020). Software Development Life Cycle methods, their advantages and disadvantages.

Kumar, M., & Rashid, E. (2018). An efficient software development life cycle model for developing software project. International Journal of Education and Management Engineering, 8(6), 59–68.

Long, R. (2020). Human Centric User Acceptance Testing.

Marsner. (2019). Agile vs. Waterfall: Which Is the Right Software Development Methodology for Your Project? Retrieved October 17, 2021, from https://marsner.com/blog/agile-vs-waterfall-which-is-the-right-software-development-methodology-for-your-project/

Maylawati, D. S., Darmalaksana, W., & Ramdhani, M. A. (2018). Systematic design of expert system using unified modelling language. IOP Conference Series: Materials Science and Engineering, 288(1), 12047. IOP Publishing.

Medeiros, J., Vasconcelos, A., Silva, C., & Goulão, M. (2018). Quality of software requirements specification in agile projects: A cross-case analysis of six companies. Journal of Systems and Software, 142, 171–194. https://doi.org/https://doi.org/10.1016/J.JSS.2018.04.064

Oktaviyani, E. D. (2020). Redundancy Detection Of Sentence Pairs In The Software Requirements Specification Documents With Semantic Approach. IOP Conference Series: Materials Science and Engineering, 879(1), 12078. IOP Publishing.

Otaduy, I., & Diaz, O. (2017). User acceptance testing for Agile-developed web-based applications: Empowering customers through wikis and mind maps. Journal of Systems and Software, 133, 212–229. https://doi.org/https://doi.org/10.1016/J.JSS.2017.01.002

Pradigm, V. (2018). What is Unified Modeling Language (UML)? Retrieved December 1, 2021, from https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/

Sualim, S. A., Yassin, N. M., & Mohamad, R. (2017). Comparative evaluation of automated user acceptance testing tool for web based application. International Journal of Software Engineering and Technology, 2(2).

Taipalus, T., Seppänen, V., & Pirhonen, M. (2020). Uncertainty in information system development: Causes, effects, and coping mechanisms. Journal of Systems and Software, 168, 110655. https://doi.org/https://doi.org/10.1016/J.JSS.2020.110655

Yunanto, R. (2018). Android-based Social Media System of Household Waste Recycling: Designing and User Acceptance Testing. IOP Conference Series: Materials Science and Engineering, 407(1), 12139. IOP Publishing.