Jurnal Ekonomi
Volume 13 , Number 01, 2024, DOI 10.54209/ekonomi.v13i01
ESSN 2721-9879 (Online)
https://ejournal.seaninstitute.or.id/index.php/Ekonomi

Enhancing Delivery Efficiency in Cargo Companies through Agile-Based Scheduling Apps

Denny Jean Cross Sihombing
Information System Study Program, Atma Jaya Catholic University of Indonesia

Article Info
Keywords:
Scheduling Apps,
Agile Methodology,
Cargo.

ABSTRACT

This research discusses the challenges cargo companies face in improving the efficiency of shipping goods amid increasingly fierce market competition. Using a qualitative approach and focusing on applying Agile methodology in application development, this research collected data through literature study and interviews with industry experts. The results show that the application of Agile methodology has a positive impact on improving delivery efficiency, enabling the company to adjust to changes in customer demand quickly, increasing flexibility in planning, and ensuring high application quality through integrated testing. The contribution of this research lies in a better understanding of the importance of applying Agile methodologies in the cargo industry, which can guide cargo companies in optimizing their shipping processes, reducing operational costs, and increasing customer satisfaction through efficient application development.

Corresponding Author:
Denny Jean Cross Sihombing
Atma Jaya Catholic University of Indonesia
Jakarta, Indonesia
denny.jean@atmajaya.ac.id

INTRODUCTION

As a vital element in the global supply chain, Cargo companies offer services to deliver goods from one point to another. The services they offer cover a wide range of shipping methods, including land, air, and sea, as well as the provision of warehousing facilities. They serve a wide area, from local to international, with a complex distribution network. In their operations, cargo companies usually involve many employees, from truck drivers to warehouse personnel and upper management. Challenges faced in the shipping process include complex coordination between the parties involved, efficient inventory management, and handling workloads that vary according to demand(Farafontova et al., 2022; Jörgensen et al., 2023; Schünemann et al., 2022).

Efficient shipment scheduling is not just a necessity but a must for cargo companies. Efficiency in shipment scheduling significantly impacts customer satisfaction, which is the desire for timely and seamless delivery of goods. Furthermore, good scheduling can result in substantial cost savings by reducing fuel, fleet maintenance, and other operational costs. More efficient use of resources, such as truck fleets, also increases their utility and optimizes companies' overall performance, helping them maintain and improve their competitiveness in an increasingly tight market(du Plessis et al., 2024; Hunt et al., 2023; Merzlikin et al., 2022; Tseremoglou et al., 2022).

Traditional approaches to scheduling shipments have several disadvantages that can be difficult for cargo companies. The manual processes typically used in scheduling are time-consuming and prone to human error. Not only does this hamper the company's responsiveness to rapid changes in demand or operational conditions, but it can also lead to inaccuracies in scheduling, leading to delayed or imperfect deliveries. In addition, the lack of visibility and control makes it difficult for companies to track and manage the delivery process efficiently.

The Agile approach promises a solution for cargo companies to improve their shipping efficiency. It offers the flexibility needed to address the dynamic challenges in the cargo industry, such as demand fluctuations and delivery schedule changes. With a focus on iterative development and intensive team collaboration, the Agile approach enables cargo companies to develop more adaptive and responsive scheduling solutions. In addition, the focus on quality and risk reduction in application development is another added value offered by the Agile approach(Batliner et al., 2022; Baxter et al., 2023; Humpert et al., 2022; Kantola et al., 2022; López et al., 2022; Ouriques et al., 2023).

The reason for choosing an Agile approach for this research is reflected in its ability to overcome the weaknesses identified in traditional approaches(Akhtar et al., n.d.; Alami et al., 2022, 2023; Almeida et al., 2022; Estrada-Esponda et al., 2024; Hasan et al., 2013; Meiliana et al., 2023; Najihi et al., 2022; Rindell et al., 2021). By adopting a more adaptive and collaborative approach, cargo companies can improve their ability to manage the complex dynamics of shipping operations. The flexibility offered by Agile approaches allows companies to respond quickly to changes while focusing on quality and risk reduction, which can help ensure successful implementation. The research question raised in this study is how implementing an Agile-based scheduling application can improve shipping efficiency within a cargo company. In addition, this research will explore what factors influence the successful implementation of Agile approaches in the context of cargo companies. By understanding the factors that influence success and the barriers that may be faced, this research is expected to provide valuable insights for cargo companies looking to improve their operational efficiency through Agile approaches.

METHODS

Developing an Agile-based scheduling application to improve shipping efficiency in a cargo company went through four main stages. The first stage, which needs definition, involves requirements analysis, eco-risk analysis, and setting goals and Key Performance Indicators (KPIs). The second stage, planning, involved designing the system architecture, designing the user interface (UI) and user experience (UX), and creating a work plan and timeline. The third stage, development, involved programming and implementation, functional and non-functional testing, and document preparation. Finally, the testing stage includes user acceptance testing (UAT), rollout and implementation, and ongoing maintenance and support.

Thus, using the Agile approach in the development of scheduling applications is expected to improve the efficiency and effectiveness of cargo company operations through a structured and measurable process.

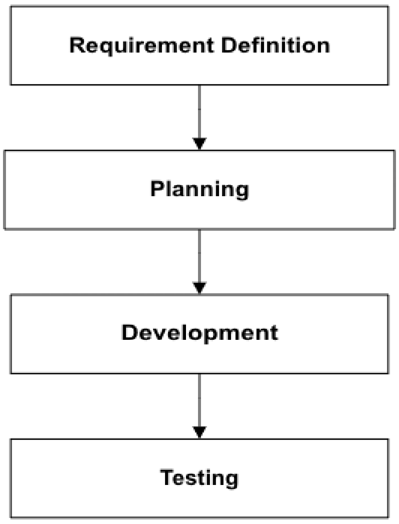

Figure 1. Research Stages

Definition of Need

The initial stage of the project is the requirements analysis, where user and stakeholder needs are identified in depth. The analysis involves an in-depth understanding of existing business processes and workflows to ensure the application adequately fulfills functional and non-functional needs. Next, a risk analysis is conducted to identify potential risks and bottlenecks in the project. Risk mitigation strategies are formulated to address risks that may arise during development. Lastly, goal setting and Key Performance Indicators (KPIs) are crucial to setting project objectives and parameters for evaluating its success.

Planning

Once a clear need is established, the focus turns to the planning stage. First, a system architecture design is undertaken to establish the required application framework and infrastructure. Appropriate technologies and tools are selected to support the designed architecture. Next, user interface (UI) and user experience (UX) design came into focus to create an intuitive UI and an engaging UX for users. Creating a work plan and timeline is the final step in the planning stage, where the project schedule and resource allocation are carefully established.

Development

The development phase begins with the programming and implementation of the application based on the approved design. Unit and integration testing is performed regularly to ensure code quality and application functionality during the development process. Once the application is built, the functional and non-functional testing phase is carried out to ensure that the application meets all the requirements that have been set. In addition, technical documentation and user manuals are prepared to guide users in the application.

Testing

User acceptance testing (UAT) is the first step in the testing phase, where users are invited to test the application and provide feedback. Based on the UAT results, revisions and improvements are made to ensure that the app meets users' expectations. After passing the UAT, the application is launched into the production environment. Training is provided to users to ensure they can use the application effectively. After the launch, maintenance and support are carried out regularly, including monitoring and bug fixing and providing technical support to users who need assistance.

RESULTS AND DISCUSSION

Definition of Need

The results of the Requirements Definition stage include three main components. First, in Requirements Analysis, the needs of the users and stakeholders involved in the project are identified. This involves an in-depth understanding of the existing business processes and workflows that must be considered to ensure that the functional and non-functional needs of the application are adequately met. Secondly, Risk Analysis focuses on identifying potential risks and obstacles in the project. Strategic risk mitigation measures are carefully prepared, including introducing possible risks and efforts to reduce or avoid their negative impact. Finally, Goal Setting and Key Performance Indicators (KPIs) are the stage where short-term and long-term goals are set along with relevant KPIs to measure the application's success. This helps ensure the project moves towards the desired direction and measurably achieves the predefined results. Thus, the Requirements Definition stage becomes a strong foundation for the success of the application development project.

Table 1. Results of Needs Definition

Stage	Results
Requirements Analysis	- Identification of the needs of users and stakeholders involved in the project. - Deep understanding of existing business processes and workflows. - Understanding the functional and non-functional requirements of the application.
Risk Analysis	- Identification of potential risks and obstacles in the project. - Development of risk mitigation strategies. - Introduction of possible risks. - Measures to reduce or avoid their negative impact.
Goal Setting and Key Performance Indicators (KPI)	- Establish the project's short-term and long-term goals. - Establishment of relevant KPIs to measure the success of the application. - Helping to ensure the project moves towards the desired direction and achieves measurable predetermined results.

Table 1 presents the results of the three stages in the requirements definition process in application development. Requirements Analysis underscores the importance of deeply understanding user needs and existing business processes to ensure the application meets functional and non-functional needs. Risk Analysis highlights the essential steps in identifying and managing potential risks and obstacles in the project, including the mitigation strategies required to reduce their negative impact. Goal Setting and KPIs emphasize the need for setting specific and measurable goals and using relevant KPIs to measure the success of the application. As such, the table provides a clear view of the critical steps that must be taken in the early stages of application development to ensure overall project success.

Planning

The outcome of the Planning stage includes three main components that are essential steps in preparing for application development. First, in the System Architecture Design, the architecture and infrastructure of the application are established by considering both functional and non-functional requirements. The selection of technologies and tools is also carefully considered to ensure the right solution is chosen according to the characteristics of the project. Secondly, in User Interface (UI) and User Experience (UX) Design, the main focus is on intuitive UI design and engaging user experience. It aims to create a positive user experience and fulfill user needs well. Finally, in Work Plan and Timeline Creation, a detailed work plan is drawn up by setting timelines and project milestones. The estimated time and resources required are also carefully determined to ensure the smooth execution of the project and achieve the set goals. Thus, the Planning stage provides a strong foundation for the overall viability of the application development project.

Table 2. Planning

Stage	Results
System Architecture Design	- Establish application architecture and infrastructure by taking functional and non-functional requirements into account. - Selection of appropriate technologies and tools to ensure the right solution is selected for the project.
User Interface Design (UI/UX)	Intuitive UI design and UX that appeals to users are the main focus. - The ultimate goal is to create a positive user experience and fulfill the user's needs well.
Making Work Plan and Timeline	- Preparation of a detailed work plan by setting project timelines and milestones. - Estimation of time and resources required to ensure smooth project implementation.

Table 2 summarizes the results of the three stages in the application development planning process. In the System Architecture Design stage, the focus is on defining the architecture and infrastructure of the application and selecting technologies that suit the needs of the project. This stage is essential to ensure that the chosen solution can effectively support the application's functions. Meanwhile, user interface design (UI/UX) emphasizes the importance of intuitive and appealing interface design to users. It aims to create a positive and satisfying user experience. Finally, Work Plan and Timeline Creation highlights the need to develop a detailed work plan by setting project timelines and milestones. Time and resource estimates are also required to manage project execution efficiently. Thus, the table provides an overview of the critical steps that need to be considered in the planning stage of app development to achieve overall project success.

Development

The results of the Application development stage, which consists of four main components, are shown in Table 3. First, programming and implementation cover the process of coding the designed solution, ensuring that the application is realized according to the needs and specifications that have been set. Second, Unit and Integration Testing provides an overview of the testing efforts carried out during the development process, where unit and integration testing is carried out periodically to ensure that each application component runs well and interacts correctly with each other. Third, Functional and Non-Functional Testing highlights the importance of thoroughly testing the application in terms of functionality according to user requirements and non-functional aspects such as performance and security. Finally, Documentation underscores the importance of compiling technical documents and user guides to support users in effectively using the application. This analysis shows that the Development stage is an essential and complex stage in application development, which requires detailed attention and coordinated efforts to ensure the project's overall success.

Table 3. Results of the Development Stages

Development Phase	Results
Programming and Implementation	This stage involves programming and implementing the application based on the previously approved design. The developers will code the designed solution to match the requirements and specifications. This implementation involves the use of programming languages and technologies that have been selected during the planning stage.
Unit and Integration Testing	During the development process, unit and integration testing is performed periodically. Unit testing aims to test each component or unit of the program separately to ensure that small functions run correctly. Meanwhile, integration testing tests the interaction between various units to ensure the application functions properly. These tests help ensure the code's quality and the application's suitability to the user's needs.
Functional and Non-Functional Testing	After the application is built, the functional and non-functional testing phase is carried out. Functional testing tests the application's functions according to the specifications set in the requirements analysis. Meanwhile, non-functional testing tests aspects such as

	the application's performance, security, and scalability. These tests ensure that the application meets all the requirements and functions adequately under various conditions.
Documentation	During the development phase, technical Documentation and user manuals are prepared to assist users in using the application. The technical Documentation contains detailed information about the application architecture, source codes, and other technical instructions. The user manual is designed to provide clear and easy-to-understand instructions on how to use the application. This Documentation is essential to support users in understanding and utilizing the application effectively.

Testing

The Testing phase is a critical stage in application development that ensures that the application is ready to be used by end users effectively. User acceptance testing (UAT) becomes an essential first step in validating the application before launching it into the production environment. Launch and implementation are then carefully carried out, followed by an ongoing maintenance and support phase to maintain application performance and user satisfaction; the test results are presented in Table 5.

Table 5. UAT Testing Results

Testing Phase	Results
User Acceptance Testing (UAT)	This stage begins with user acceptance testing (UAT), where users are invited to test the application and provide feedback. UAT aims to ensure that the application meets user expectations and can be used effectively in production. Based on the UAT results, revisions and improvements are made to ensure the application is ready to launch to end users. After UAT, the application is prepared for launch into the production environment.
Launch and Implementation	The application is launched into the production environment after successfully passing the UAT and any necessary revisions. This rollout process includes training users to ensure they can use the application effectively. Implementation is done carefully to minimize disruptions and ensure a smooth transition to the new application.
Maintenance and Support	After launch, the maintenance and support phase is carried out regularly. This includes monitoring and fixing bugs that may arise after launch and providing technical support to users who need assistance. Effective maintenance and responsive support are essential for maintaining the performance and reliability of the app in the long run, as well as increasing user satisfaction with their user experience.

Table 5 provides an overview of the results of the Testing stage in application development, which consists of three main stages. First, User Acceptance Testing (UAT) emphasizes the importance of involving end users in validating the application before launching it into the production environment. This process allows for identifying issues and final fixes before the app is made widely available. Secondly, Launch and Implementation highlight the crucial stage of bringing the application to the production environment by training the users. This step ensures a smooth and effective transition to the new application. Finally, Maintenance and Support shows the importance of continuous maintenance and technical support to ensure the application continues to run well after launch and assist users in dealing with problems or queries. This analysis shows that the Testing phase is not the end of the development process but an ongoing step to ensure the application remains relevant, high-performing, and meets users' needs in the long run.

CONCLUSION

Agile methodology can be applied in application development to improve efficiency in cargo companies. This research found that the Agile approach in planning, developing, and testing the application positively impacted improving the efficient delivery of goods. The planning stage helps set clear objectives and adjust to changing customer demands flexibly. Designing a system architecture and user interface responsive to user needs is a strong foundation for the application's success. Regular integrated functional and non-functional testing ensured the high quality of the application before launch. The testing phase results confirmed the importance of user participation in the testing process before the launch into production. The rollout of the application to the production environment was followed by intensive user training, which ensured that users could make the most of the application. Post-launch support and maintenance are essential steps in ensuring the continued performance of the application. This research shows that the application of Agile methodology in cargo application development has a positive impact on improving shipping efficiency, reducing operational costs, and increasing customer satisfaction. By understanding the challenges faced by cargo companies and adopting an Agile approach, companies can optimize their operations and remain competitive in an increasingly competitive market.

REFERENCE

Akhtar, A., Bakhtawar, B., & Akhtar, S. (N.D.). Extreme Programming Vs Scrum: A Comparison Of Agile Models. *International Journal Of Technology, Innovation And Management (Ijtim)*, *2*, 2022. Https://Doi.Org/10.54489/Ijtim.V2i1.77

Alami, A., Krancher, O., & Paasivaara, M. (2022). The Journey To Technical Excellence In Agile Software Development. *Information And Software Technology*, *150*. Https://Doi.Org/10.1016/J.Infsof.2022.106959

Alami, A., Zahedi, M., & Krancher, O. (2023). Antecedents Of Psychological Safety In Agile Software Development Teams. *Information And Software Technology*, *162*. Https://Doi.Org/10.1016/J.Infsof.2023.107267

Almeida, F., Simões, J., & Lopes, S. (2022). Exploring The Benefits Of Combining Devops And Agile. *Future Internet*, *14*(2). Https://Doi.Org/10.3390/Fi14020063

Batliner, M., Boës, S., Heck, J., & Meboldt, M. (2022). Linking Testing Activities With Success In Agile Development Of Physical Products. *Procedia Cirp*, *109*, 146–154. Https://Doi.Org/10.1016/J.Procir.2022.05.228

Baxter, D., Dacre, N., Dong, H., & Ceylan, S. (2023). Institutional Challenges In Agile Adoption: Evidence From A Public Sector It Project. *Government Information Quarterly*, *40*(4). Https://Doi.Org/10.1016/J.Giq.2023.101858

Du Plessis, F., Goedhals-Gerber, L., & Van Eeden, J. (2024). The Impacts Of Climate Change On Marine Cargo Insurance Of Cold Chains: A Systematic Literature Review And Bibliometric Analysis. *Transportation Research Interdisciplinary Perspectives*, *23*. Https://Doi.Org/10.1016/J.Trip.2024.101018

Estrada-Esponda, R. D., López-Benítez, M., Matturro, G., & Osorio-Gómez, J. C. (2024). Selection Of Software Agile Practices Using Analytic Hierarchy Process. *Heliyon*, *10*(1). Https://Doi.Org/10.1016/J.Heliyon.2023.E22948

Farafontova, E., Safronov, V., Scheblyakov, E., Babaeva, A., & Kubrikova, A. (2022). Problems Of The Carrier's Liability For Violation Of The Terms Of Cargo Delivery. *Transportation Research Procedia*, *68*, 223–228. Https://Doi.Org/10.1016/J.Trpro.2023.02.030

Hasan, R., Ta, A.-, & Razali, R. (2013). Prioritizing Requirements In Agile Development : A Conceptual Framework. *Procedia Technology*, *11*(Iceei), 733–739. Https://Doi.Org/10.1016/J.Protcy.2013.12.252

Humpert, L., Röhm, B., Anacker, H., Dumitrescu, R., & Anderl, R. (2022). Method For Direct End Customer Integration Into The Agile Product Development. *Procedia Cirp*, *109*, 215–220. Https://Doi.Org/10.1016/J.Procir.2022.05.239

Hunt, J. D., Nascimento, A., Tong, W., Zakeri, B., Jurasz, J., Patro, E. R., Đurin, B., De Jesus Pacheco, D. A., De Freitas, M. A. V., Filho, W. L., & Wada, Y. (2023). Perpetual Motion Electric Truck, Transporting Cargo With Zero Fuel Costs. *Journal Of Energy Storage*, *72*. Https://Doi.Org/10.1016/J.Est.2023.108671

Jörgensen, A. M., Wibel, R., Veider, F., Hoyer, B., Chamieh, J., Cottet, H., & Bernkop-Schnürch, A. (2023). Self-Emulsifying Drug Delivery Systems (Sedds): How Organic Solvent Release Governs The Fate Of Their Cargo. *International Journal Of Pharmaceutics*, *647*. Https://Doi.Org/10.1016/J.Ijpharm.2023.123534

Kantola, K., Vanhanen, J., & Tolvanen, J. (2022). Mind The Product Owner: An Action Research Project Into Agile Release Planning. *Information And Software Technology*, *147*. Https://Doi.Org/10.1016/J.Infsof.2022.106900

López, L., Burgués, X., Martínez-Fernández, S., Vollmer, A. M., Behutiye, W., Karhapää, P., Franch, X., Rodríguez, P., & Oivo, M. (2022). Quality Measurement In Agile And Rapid Software Development: A Systematic Mapping. *Journal Of Systems And Software*, *186*. Https://Doi.Org/10.1016/J.Jss.2021.111187

Meiliana, Daniella, G., Wijaya, N., Putra, N. G. E., & Efata, R. (2023). Agile Software Development Effort Estimation Based On Product Backlog Items. *Procedia Computer Science*, *227*, 186–193. Https://Doi.Org/10.1016/J.Procs.2023.10.516

Merzlikin, I., Zueva, A., Kievskaya, S., Shkoropat, E., & Popov, K. (2022). The Market Of Air Transportation And Cargo Transportation In The Investment Strategy Of Transport Enterprises. *Transportation Research Procedia*, *63*, 1420–1430. Https://Doi.Org/10.1016/J.Trpro.2022.06.153

Najihi, S., Elhadi, S., Abdelouahid, R. A., & Marzak, A. (2022). Software Testing From An Agile And Traditional View. *Procedia Computer Science*, *203*, 775–782. Https://Doi.Org/10.1016/J.Procs.2022.07.116

Ouriques, R., Wnuk, K., Gorschek, T., & Svensson, R. B. (2023). The Role Of Knowledge-Based Resources In Agile Software Development Contexts. *Journal Of Systems And Software*, *197*. Https://Doi.Org/10.1016/J.Jss.2022.111572

Rindell, K., Ruohonen, J., Holvitie, J., Hyrynsalmi, S., & Leppänen, V. (2021). Security In Agile Software Development: A Practitioner Survey. *Information And Software Technology*, *131*. Https://Doi.Org/10.1016/J.Infsof.2020.106488

Schünemann, J., Finke, S., Severengiz, S., Schelte, N., & Gandhi, S. (2022). Life Cycle Assessment On Electric Cargo Bikes For The Use-Case Of Urban Freight Transportation In Ghana. *Procedia Cirp*, *105*, 721–726. Https://Doi.Org/10.1016/J.Procir.2022.02.120

Tseremoglou, I., Bombelli, A., & Santos, B. F. (2022). A Combined Forecasting And Packing Model For Air Cargo Loading: A Risk-Averse Framework. *Transportation Research Part E: Logistics And Transportation Review*, *158*. Https://Doi.Org/10.1016/J.Tre.2021.102579