IT Strategic Planning to Design and Implementat a Billing Dashboard Using Agile Methodology

Jefri Abdurrozak Ismail [a,1,*]

[a] Magister Teknologi InformasiUPN "Veteran" Jawa Timur, Jl.Raya Rungkut Madya, Gunung Anyar, Surabaya60294,Indonesia
[1] 23066020025@student.upnjatim.ac.id
* corresponding author

ARTICLE INFO	ABSTRACT (10^{PT})
	Type your abstract here (10 pt).

1.   Introduction

In the current digital era, the use of information systems has become a fundamental aspect of how businesses manage their operations. This is particularly evident in the IT infrastructure services industry, where speed, accuracy, and transparency in billing account management are critical factors. Inefficiencies in billing systems can adversely impact a company's cash flow and erode customer trust. Therefore, there is an urgent need for a digital solution capable of streamlining billing processes while simultaneously enhancing the customer experience. In response to this challenge, a project has been initiated to develop a comprehensive billing dashboard aimed at increasing operational efficiency and improving internal workflows.

This study focuses on strategic IT planning in the context of billing system development, with a particular emphasis on the design of a billing dashboard utilizing the Agile methodology. Agile is selected due to its capacity to support iterative and flexible development, allowing continuous user feedback to be incorporated throughout the development lifecycle. This approach ensures that the resulting system is not only functionally robust but also adaptable to the evolving needs of the business environment. As such, Agile plays a critical role in maintaining alignment between system capabilities, end-user expectations, and broader organizational goals.

The primary objective of this project is to deliver a project plan of creating billing dashboard that is functional, efficient, and user-centered. The system is designed to simplify end-to-end billing management processes, including transaction recording, invoice tracking, and real-time financial reporting. Additionally, the dashboard will incorporate automation features to minimize manual intervention and reduce the likelihood of human error to

improve the overall user experience. In the long term, the system is expected to contribute to the company's strategic business objectives by integrating seamlessly with existing systems and processes, delivering measurable improvements in operational accuracy, efficiency, and customer satisfaction.

2. Method

This research adopt Agile methodology which testing continuously at the end of each sprint to ensure every functions are working as expected. Research flow of IT strategic planning to design a billing dashboard using agile methodology can be found at figure 1 below.
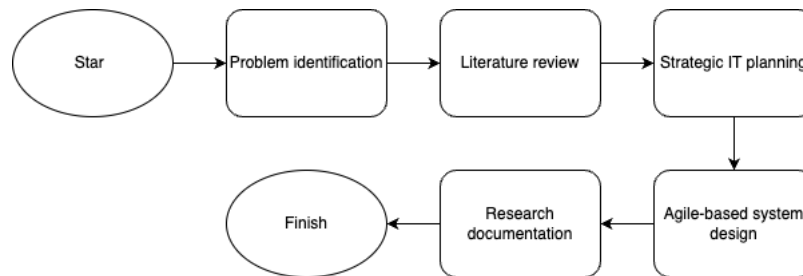

Figure 1. Reseach workflow

2.1. Problem Identification

One of the main challenges faced by the IT infrastructure service industry is the efficient, accurate, and integrated management of billing systems. Common issues include time-consuming manual processes, a lack of transparency in financial reporting, and limitations in providing real-time data. Identifying these problems forms the foundation of this research and serves as a critical step in developing a system-based solution that addresses both operational dynamics and customer expectations.

2.2. Literature Review

Previous studies have highlighted the importance of digitizing billing systems to enhance business efficiency. According to Sari et al. (2021), automated information systems can improve billing accuracy and speed by up to 35% compared to manual methods. Nugroho (2020) emphasized that the implementation of interactive dashboards contributes to greater transparency in financial reporting and facilitates better managerial decision-making. The literature also underscores the significance of user-centered design principles in system interface development and the relevance of Agile methodology in supporting iterative and adaptive software development.

2.3. Strategic IT Planning

Strategic IT planning refers to the process of aligning technological initiatives with organizational goals to ensure long-term sustainability and impact. As outlined by Luftman (2000), alignment between business strategy and IT strategy is a key determinant of successful technology implementation. In the context of billing dashboard development, strategic planning involves defining functional system requirements, ensuring integration with existing digital infrastructure, and establishing key performance indicators (KPIs) to measure implementation outcomes.

2.4. Agile-Based System Design

Agile methodology was chosen for this project due to its adaptability and responsiveness to change. Specifically, the Scrum framework facilitates development in short, focused sprints that deliver incremental system features. System design in this context involves formulating user stories, building prototypes, validating usability through user feedback, and continuously refining the system. This iterative process ensures that the developed system remains responsive to evolving user needs and enhances the overall product quality.

2.5. Research Documentation

All stages of the research—from initial analysis to implementation—are systematically documented. This documentation includes detailed records of system requirements, technical specifications, sprint outcomes, and system performance evaluations. Beyond serving as a formal archive, the research documentation provides a reference point for future studies and supports transparency and replicability in academic and professional contexts.
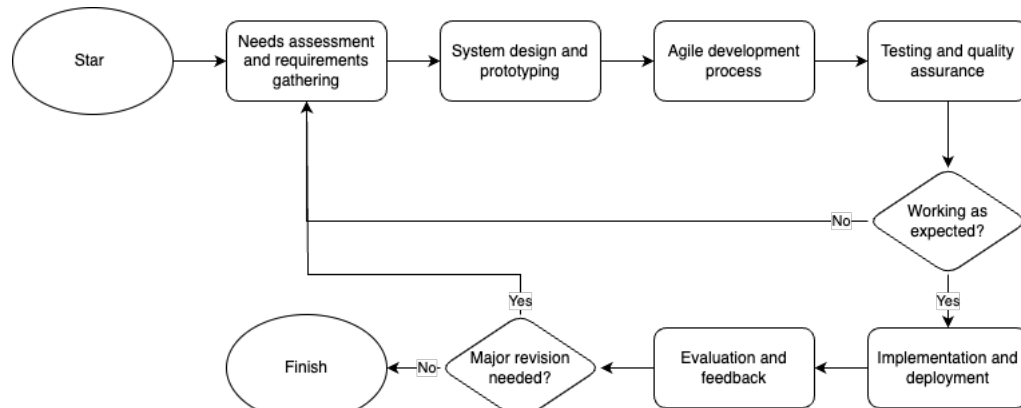
3.   Results and Discussion

Figure 1. Project workflow

3.1.   Needs Assessment and Requirements Gathering

The Needs Assessment and Requirements Gathering phase ensuring that the billing dashboard for Elitery aligns with both business objectives and user expectations. This process begins with the identification of relevant stakeholders, including business leaders, IT teams, end-users such as billing staff, and external stakeholders like customers[1]. Understanding their perspectives is important to gathering comprehensive and accurate data[2]. Various data collection methods will be conducted, such as interviews, where one-on-one discussions will take place with key stakeholders to uncover their expectations, pain points, and desired features. Surveys and questionnaires will be used to gather quantitative data from a larger group of users, helping prioritize the necessary features based on feedback[3]. Workshops will bring stakeholders together to collaboratively discuss needs and agree on the most critical features, while observations of existing workflows will allow for the identification of inefficiencies and areas that could benefit from automation.

Once the data is collected, the next step is to classify it into functional and non-functional requirements. Functional requirements include specific features the system must support, such as real-time data updates, automated invoice generation, user-friendly interfaces, and integration with existing accounting software. Non-functional requirements define system characteristics like scalability, security, and usability, ensuring the dashboard can grow with the company, protect sensitive data, and be intuitive for users[4]. After gathering and categorizing the requirements, they will be prioritized the MoSCoW method (Must Have, Should Have, Could Have, Won't Have) to identify the most crucial features for the system[5].

3.2.   System Design and Prototyping

The System Design and Prototyping phase is focused on transforming the gathered requirements into a functional and structured design for the billing dashboard, ensuring that the system is user-friendly and scalable to meet Elitery's strategic goals. The system will be built on a multi-tier architecture, with the frontend developed using React for real-time data display and a Node.js backend handling user requests and database interactions. A MySQL database will store billing data, and the dashboard will integrate with external systems via RESTful APIs. The UI will focus on simplicity and clarity, with features such as automated invoice generation, real-time notifications, and role-based access control[6].

An interactive prototype will be created using Figma, allowing stakeholders to explore the system and provide feedback, which will be incorporated in iterative updates. This process follows the Agile methodology, ensuring that the system can adapt to changing requirements based on feedback from

each sprint. The prototype will help refine the design, ensuring the final product meets both business objectives and user expectations[7].

3.3.   Agile Development Process

The Agile Development Process for the billing dashboard is designed to ensure that the project remains flexible, adaptive, and responsive throughout its lifecycle. By following Agile principles, the development team will work in iterative cycles, known as sprints, which allows for continuous delivery of functional features, regular feedback from stakeholders, and adjustments as necessary[8]. This approach guarantees that the final product aligns with both business objectives and user needs, while also staying on schedule and within scope.

The process begins with Sprint Planning, where the development team, business leaders, and stakeholders define the scope of work for each sprint. A sprint typically lasts 1 to 4 weeks, depending on the complexity of the tasks. During sprint planning, the team will review and break down the prioritized requirements from the previous phase into manageable user stories. These user stories will be assigned estimated effort and resources, ensuring that tasks are achievable within the sprint. Each user story will include clear acceptance criteria, which will guide the development team's work[9].

Once planning is complete, the Design and Development phase begins. The development team will focus on building the features identified during the planning session. Frontend developers will focus on creating the user interface, ensuring the design is responsive and intuitive. Backend developers will implement the necessary business logic, handle data processing, and ensure that the system integrates smoothly with external services. This phase involves ongoing collaboration and communication within the team to address any challenges and ensure that the system aligns with the functional requirements outlined in the sprint[8].

As development progresses, the team will regularly test the system to ensure its functionality meets expectations. Automated tests will be implemented to verify that new features do not disrupt existing functionality. In addition to automated testing, manual testing will be conducted for user interface elements, ensuring they are intuitive and meet the usability standards. At the end of each sprint, the development team will conduct Sprint Reviews, during which they will demonstrate the newly completed features to stakeholders. This allows for immediate feedback, enabling the team to make necessary adjustments before the next sprint begins[10].

After each sprint review, stakeholders will provide feedback on the newly developed features. This feedback will guide the team in refining the system, ensuring it meets both business goals and user expectations[10]. The Agile methodology encourages continuous improvement, so if any features need adjustments or new functionality is required, the development team can prioritize these changes in future sprints.

After each sprint, Testing and Quality Assurance (QA) will be conducted. This phase ensures that the system is working as expected and meets the necessary quality standards[10]. The QA team will conduct performance, security, and usability testing to ensure the system can handle a large number of users and secure sensitive data. User Acceptance Testing (UAT) will also take place, where end-users will interact with the system in real-world scenarios to validate the functionality and usability. Any issues identified during QA will be addressed before moving forward to the next phase.

Once the system has been fully developed and all features have been tested and validated, the final product will be ready for Deployment and Release. The deployment will occur at the end of the last sprint, where the system will be moved to the production environment and made available to users[11]. The development team will ensure a smooth transition and address any issues that arise during the release process. Post-deployment, the team will continue to monitor the system to ensure it functions properly and resolve any issues that arise in the live environment.

After deployment, the Agile process does not end. The team will continue to provide ongoing support and improvements. Post-deployment sprints will be scheduled to implement new features, fix bugs, and refine existing functionalities. This ensures that the billing dashboard remains up-to-date and continues to evolve based on user feedback and changing business needs. The Agile methodology ensures that the system is adaptable and capable of meeting both current and future requirements.

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

References

[1]   J. S. Harrison, J. B. Barney, R. E. Freeman, and R. A. Philips, *The Cambridge Handbook of Stakeholder Theory*. Cambridge University Press, 2019. doi: 10.1017/9781108123495.

[2]   *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*, 7th ed. Newton Square: Project Management Institute, Inc., 2021.

[3]   R. S. Simhadri and M. Shameem, "Challenges in Requirements Gathering for Agile Software Development," in *Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering*, New York, NY, USA: ACM, Jun. 2023, pp. 406–413. doi: 10.1145/3593434.3594237.

[4]   A. M. Striuk, S. O. Semerikov, H. M. Shalatska, and V. P. Holiver, "Software requirements engineering training: problematic questions," in *Proceedings of the 4th Workshop for Young Scientists in Computer Science & Software Engineering (CS&SE@SW 2021)*, A. E. Kiv, S. O. Semerikov, V. N. Soloviev, and A. M. Striuk, Eds., Ukraine: CEUR Workshop Proceedings, Dec. 2021, pp. 3–11.

[5]   Suchetha Vijayakumar, Krishna Prasad K, and R. Holla M., "Assessing the Effectiveness of MoSCoW Prioritization in Software Development: A Holistic Analysis across Methodologies," *EAI Endorsed Transactions on Internet of Things*, vol. 10, Oct. 2024, doi: 10.4108/eetiot.6515.

[6]   H. Rianto, "Analysis and design of a bill notification system using rapid application development," *Manajemen, Teknologi Informatika dan Komunikasi (Mantik)*, vol. 7, no. 2, pp. 3895–3905, Feb. 2024.

[7]   Lasminiasih, G. E. Saputra, Rooswhan Budhi Utomo, and Elbi Wiseno, "USING PROTOTYPING METHOD FOR ANALYSIS AND DESIGN OF INFORMATION SYSTEMS FOR STUDENT REGISTRATION IN SEKOLAH MASTER," *International Journal Science and Technology*, vol. 1, no. 2, pp. 19–29, Jul. 2022, doi: 10.56127/ijst.v1i2.140.

[8]   S. Alsaqqa, S. Sawalha, and H. Abdel-Nabi, "Agile Software Development: Methodologies and Trends," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 14, no. 11, p. 246, Jul. 2020, doi: 10.3991/ijim.v14i11.13269.

[9]   L. Trihardianingsih, M. Istighosah, A. Y. Alin, and M. R. Ghonim Asgar, "Systematic Literature Review of Trend and Characteristic Agile Model," *JURNAL TEKNIK INFORMATIKA*, vol. 16, no. 1, pp. 45–57, May 2023, doi: 10.15408/jti.v16i1.28995.

[10]   I. A. Dewi, Y. Miftahuddin, M. A. Fattah, C. B. Palenda, and S. Fathurrahman Erawan, "Point of Sales System in InHome Café Website using Agile Methodology," *Journal of Innovation and Community Engagement (ICE)*, vol. 01, no. 01, pp. 1–19, Mar. 2021.

[11]   V. P. Munteanu and P. Dragos, "The Case for Agile Methodologies against Traditional Ones in Financial Software Projects," *European Journal of Business and Management Research*, vol. 6, no. 1, pp. 134–141, Feb. 2021, doi: 10.24018/ejbmr.2021.6.1.741.