Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 22, No. 3, July 2023, pp. 567∼582
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 200/M/KPT/2020
DOI: 10.30812/matrik.v22i3.2830

E-Mortality Using Agile Scrum Method to Improve Healthcare Information Services Effectiveness

Ardafa Ihromi, Yuda Syahidin, Erix Gunawan, Neneng Yuniarty
Politeknik Piksi Ganesha, Bandung, Indonesia

Article Info

Article history:

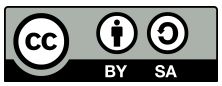
Received March 10, 2023
Revised April 07, 2023
Accepted Juny 06, 2023

Keywords:

Agile
Scrum
Information System
Mortality

ABSTRACT

The advancement of information system technology is presently used extensively in many disciplines, including the field of health care or hospitals. This research aims to create an information system to handle hospital data to facilitate processing of patient data that has been declared dead with quality. In this study, it was found that there was no information system in the form of a program or application to handle death data. The management process still relies on Microsoft Excel, which is considered less efficient. In addition, the development of this information system is assisted by choosing the suitable software development methodology and considering existing needs. This research uses the Agile Development Method with the Scrum framework for software development. This research is qualitative descriptive and uses the observation method in data collection. C# programming language and MySQL database are also used in this system. This research produces an information system to handle death data by the product backlog. It is intended to meet user needs to assist in processing death data more effectively and efficiently and reduce the error rate associated with recording death data manually.



Corresponding Author:

Ardafa Ihromi,
Health Information Management ,
Politeknik Piksi Ganesha, Bandung, Indonesia,
Email: ardafaihromi@gmail.com

How to Cite:
A. Ihromi, Y. Syahidin, E. Gunawan, and N. Yuniarty, ”E-Mortality using Agile Scrum Method to Improve Information Services Effectiveness”, MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer, vol. 22, no. 3, pp. 567-582, Jul. 2023.

1. INTRODUCTION

Rapid advancements in science and technology profoundly affect all aspects of existence. Digital technology contributes to progress, including in the field of health technology. In recent years, the technology of the healthcare industry has undergone numerous significant changes or upheavals. Therefore, it is required that public health services be more responsive to modern technology. Managing a hospital is a difficult task, especially with the advent of globalization and economic changes. One of them is an application system for managing mortality data (death). The mortality data management system was designed to simplify, enhance efficiency, and assure security. Its functions include the automatic capture and logical validation of numbers and basic information. Additionally, the electronic mortality case report module must be tailored to the specifics of each medical institution.

The information system used to handle mortality data seeks to resolve issues with managing death data by enhancing the efficiency of medical record information services and developing a computerized system for processing death data. In this research, software development was conducted using the Agile software development methodology and the Scrum framework. The Scrum software development method was selected because it can solve the problems of development time and human resources, allowing the efficacy of the information system to be measured in terms of user satisfaction. This information system can produce data, including death reports, death index reports, and death certificates. In addition, the death information system displays data regarding the number of deaths in a population. The designed system is anticipated to improve mortality data processing activities' effectiveness and efficiency and reduce the error rate of manually recording death data.

Death reports archive mortality data as hospital data sources for monthly, quarterly, semiannual, and annual reports [1, 2]. The flow of reporting mortality (death) begins with the patient who has passed, followed by the entry of death information data into the mortality information system, which can exhibit the number and details of deceased patients. Unfortunately, the system that combines the patient data input system in the treatment area and emergency unit with the Medical Record Installation's data management has not fulfilled the criteria for producing death certificates. Creating a death certificate still requires a paper form written by a doctor, and it does not include information on the patient's cause of death, so the information is not continuous. This results in poor quality and inaccurate information within the obtained medical records.

Considering prior research by [3], Using the System Development Life Cycle (SDLC) development method, a system was created to assist medical record officers with managing inpatient death data and preparing quality medical record data reports. According to the findings of this study, the system can operate effectively by system requirements. However, there is still a lack of application of the SDLC method in analysis. What distinguishes this research from the author's is that it focuses on the application of SDLC and the implementation of web-based requirements analysis.

According to research by [**?**], The Microsoft Visual Studio-based Death Information System assists hospitals in producing quality medical record reporting. However, the method used to develop the system is currently unknown. So relevant updates are required for the current era for the system to be optimally utilized. The difference with the author is in the results of the data processing flow and monotonous design.

In research by [4], the Waterfall methodology was applied to create a death record system. This system's value lies in its ability to improve the reliability of internal and external reports. However, this system has several disadvantages, one of which is that it is challenging to experience changes in the needs desired by the user. In contrast to the author's study, this one uses the waterfall system development method in conjunction with the chosen database. A death index is a result.

According to a study by [5], create and design death certificate applications using the Waterfall, which is anticipated to facilitate work and organize system development. However, more study is needed to further develop the information system because of flaws in the design, specifically the lack of maintenance on a system that minimizes errors that can be monitored. The difference with the author is in the system's design, the database used, the development method, and the form of the final product, which is a death certificate.

According to the findings of a previous study on information systems about technologically-based mortality, previous studies used the waterfall technique to develop an information system. However, the waterfall method has several drawbacks, including a lengthy development cycle and the inability to make any modifications. Consequently, using the agile software development methodology with the Scrum framework is appropriate and in keeping with the requirements of the present research because several features can be utilized without having to wait to complete the development process, particularly in a brief amount of time. As a result of the development of an electronic death information system, hospital information services are expected to become more efficient and effective. By comparing previous research, this study can contribute to developing an efficient death information system that presents comprehensive data about death reports, death indices, and death certificates to enhance an effective and well-organized death data processing information system. However, because this is the first research on death information systems using the agile scrum method and there has been little prior research on death information systems, there is little that can be compared.

This research is presented and put together in different parts. In the second section, researchers discuss how Agile Scrum is

put together and how it works. In the third section, the results of the investigation into the methodology are given. The last part of this article discusses the conclusions and suggestions from this research.

2. RESEARCH METHOD

This research project was prepared using descriptive qualitative research methodology. Qualitative research methods include interviews, observation, and library analysis. The development method used is the agile method with the scrum framework.

2.1. Requirements Collection

Data collection using observation, namely collecting data by observing and researching directly to obtain the data needed according to the application currently running at Hospital X. To get information, researchers conducted data collection interviews with authorized parties who handle problems with this research, especially regarding death data management. Meanwhile, to add references to the studied system, researchers conducted a literature study by researching and collecting data from several documents, books, and the internet related to the death data management system. Based on the observations obtained, researchers found problems that exist in the information system for managing death data at X Hospital, namely, the system that connects inputting patient data in the treatment room and emergency department with data processing in the Medical Records Installation has not met the needs of death data management. Because there is no interconnected digitized form, the death data management process is still done manually, whereas data entry is computerized using Microsoft Excel. However, this is less effective because it cannot immediately display the death data management output quickly if needed. To solve the problems, the author employs Microsoft Visual Studio to design an information system, which is supposed to lead to an application for the mortality information system to help solve the problems.

2.2. System Development Method

The Agile Scrum method is used for system development in this research. Agile software development uses a gradual and iterative process to achieve goals. Agile software development emphasizes changeability. This research used Agile Software Development because the system can adapt to changes. Implementation projects also depend on IT infrastructure design and capacity [6]. Agile Software Development includes Scrum. Thus, agile has Scrum. Scrum helps create high-value products efficiently and creatively by managing complex product development. Even projects with tight deadlines, complex requirements, and high individuality can use [7, 8]. Product Owners, Developers, and Scrum Masters are crucial to Scrum. The Product Owner sets expectations and owns product success. These items will go into the Product Backlog. However, the interdisciplinary Development Team must meet the Product Owner's needs. The third role, the Scrum Master, manages everything, coaches the team, and ensures Scrum is appropriately used [9].

The stages depicted in Figure 1 demonstrate that Scrum comprises scrum teams with their respective roles and predetermined scrum implementation of the scrum methodology for the E-mortality application development. Scrum components have a specific function and role in achieving success.

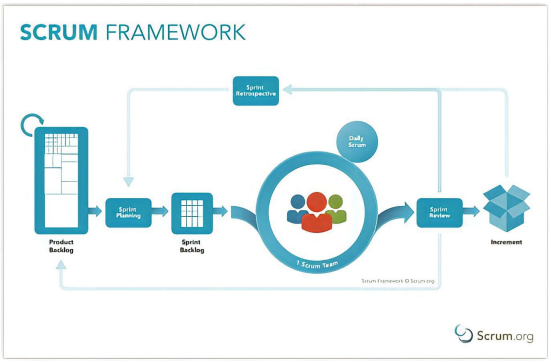


Figure 1. Development of the Agile Scrum System

The following is included in each stage of the Scrum, and Product Backlog is the only source of all changes that must be

implemented on the product. The Product Owner oversees the backlog and must ensure its completeness, accessibility, and prioritized order [10]. Prioritize the features that will be developed for the mortality data management information system that will be constructed. Content can be introduced at any time to features [11]. Sprints Activities must be organized to satisfy the requirements specified in the backlog over a 30-day work period. Sprint activities are essential to achieving the product goal, including the Sprint Plan, Daily Scrum Meeting, Sprint Review, and Retrospective. Each occurs in activities sprints. Each sprint is typically 2-4 weeks long [8]. Sprint Planning.

Determine the sprint implementation schedule, select the product backlog items expected to be finished during the sprint, and create a list of the desired tasks. One product backlog item can be subdivided or divided into multiple work lists. Still, it is conceivable that various similar product backlog items will be merged into a single task item. Sprint Backlog Lists the product backlog opted for Practice. The scrum team then used the prioritized list to inform their sprint planning session, which produced the sprint backlog list. The Sprint Backlog manifest will remain unchanged until the end of the sprint [12].

Scrum Meeting (Daily Scrum) Hold a gathering to go over the status of the task of creating an information system for handling death data with the team that was assigned. The frequency of these meetings can be sporadic, with a minimum of once per week or daily (if more coordination is needed due to dependency) [13]. Sprint Review, The entire Scrum team holds meetings to evaluate the outcomes of all completed sprints and determine whether they align with the product backlog [14]. Sprint Retrospective Conduct an evaluation by the scrum team to identify process constraints. This is the final activity where the Scrum is prepared for launch and performance evaluation throughout the sprint. Increment During the sprint phase, results undergo a series of iterative enhancements carried out in response to specific requirements. Finally, the software features developed are displayed for user evaluation at a predetermined time.

3. RESULT AND ANALYSIS

The research results are based on a logical Proses Management of death data (mortality) in the current information system is still performed manually. In contrast, the proposed information system's process is computerized, with a database serving as the information system's storage medium. The following procedures guide the development of this system based on the preceding steps.

3.1. Needs analysis

The Hospital x requirements analysis will incorporate all mortality data management into a centralized database, facilitating the user's ability to accommodate mortality data. Unified Modeling Language (UML) and Flowchart diagrams are provided below to provide an overview of system development (Unified Modeling Language). Use Case Diagram, Establishing a Use Case Diagram for the mortality data management system explains how users interact with the system to determine its functionality can be seen in Figure 2.

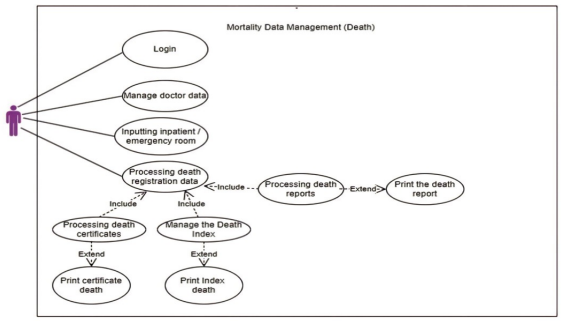


Figure 2. Actor Use Case Diagrams

Flowchart, The software architecture will be presented as a system flowchart. A flowchart is a graphical representation of the logic behind a computer program or system procedure. A flowchart is a visual representation of a sequence of operations in a computer system. User-friendliness was a primary design goal in creating this app [15, 16]. The researcher will present a flowchart model of the mortality data management information system in Figure 3.

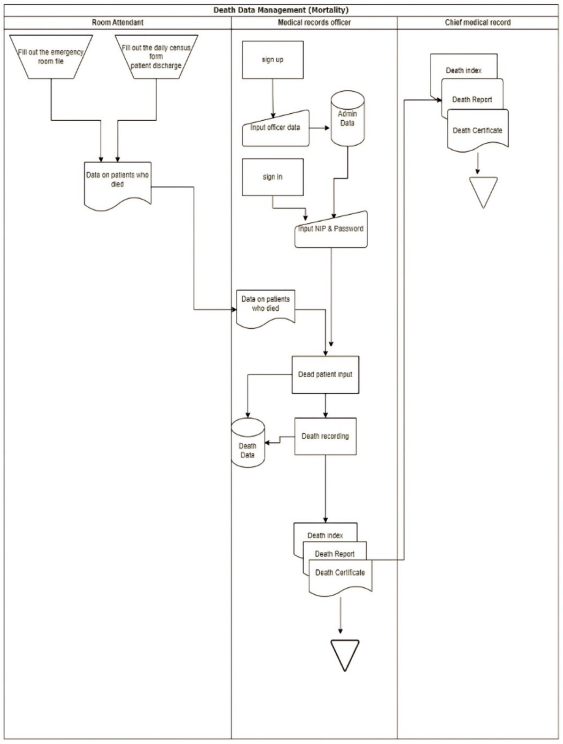


Figure 3. Flowchart Diagram

Activity Diagram, Activity diagrams show the workflow or actions of a software-based system. The steps of the people and machines in the system are described using activity diagrams. The functional architecture's activity diagrams are modified to include the infection modes for the simulation so activities can be performed by the system instead of those served by individual actors. For example, see the diagram below to explain how a user can enter the system with a single login and access the primary menu. The information system for managing mortality statistics has the action diagram in Figure 4 until 9.

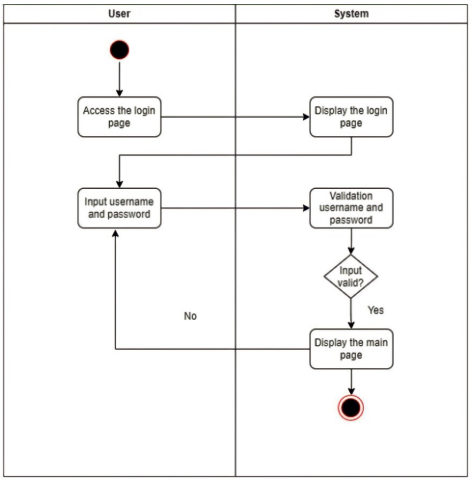


Figure 4. Login Activity Diagrams

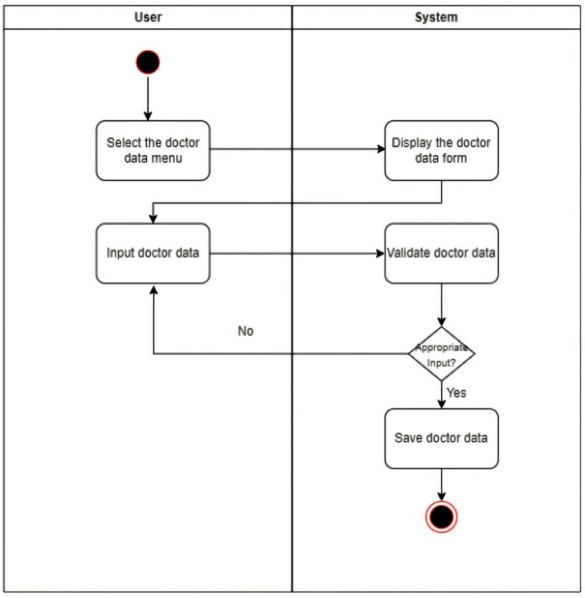


Figure 5. Activity Diagram Doctor

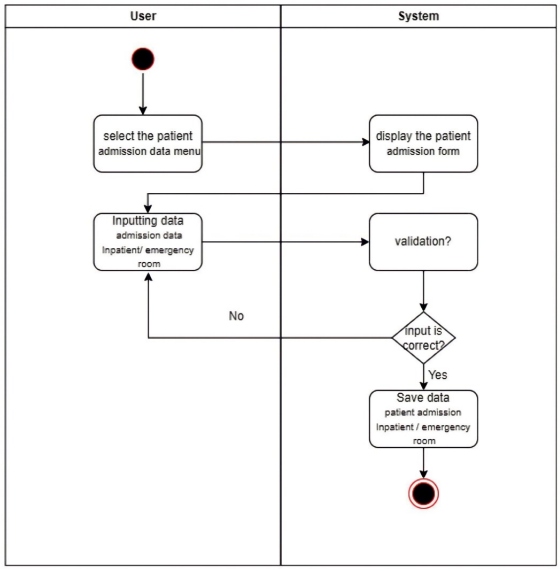


Figure 6. Admission Records Activity Diagram

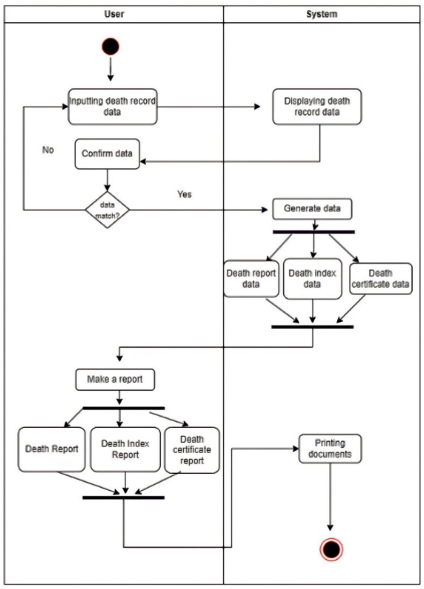


Figure 7. Death Recording Activity Diagram

3.2. Application Development

Product Backlog, A Product Backlog must be created as a first stage. The product backlog is a list of potential orders and activities that must be completed to make changes to the product. The developed product backlog also contains an estimated completion time for each feature or work grouping. The information system is ranked according to customer objectives in the product backlog researcher. Table 1 displays key information system specifications that must be created for this system based on the findings of the analysis and survey.

Table 1. Product Backlog

No.	Product Backlog	Priority
1.	Diagram Design	High
2.	Application Login	High
3.	Manage Menu Data	High
4.	Manage User Data	High
5.	Manage Doctor Data	High
6.	Manage admission records Data	High
7.	Manage Death Record Data	High
8.	Death Report	High
9.	Manage Death Index	High
10.	Manage death certificates	High

Sprint Planning, Discussions between the development team and the scrum master regarding the strategy to create an information system for handling death data were conducted as part of the sprint planning [17]. Following the product backlog, development begins with designing the system's requirements by creating a flow diagram. The sprint backlog is the next stage after sprint planning. It is created based on the results of the sprint planning process. This design divides the dashes into three phases. Here are the specifics of intervals 1 through 3.

Table 2. Sprint Backlog 1

Product Backlog	Task	Est. (days)
Diagram Design	Analysis of functional system requirements & flowchart	0,5
	Create a Use Case	0,5
	Create an Activity Diagram	1
Application Login	Create a database schema	0,5
	Create login UI design	1
	Implementing UI into coding	1
	Testing	0,5
Manage Menu Data	Create a database schema	0,5
	Create menu UI design	1
	Implementing UI into coding	1
	Testing	0,5

Table 2 contains three backlog items: designing UML designs, creating application login menus, and menu data. In addition, every backlog item includes 3 or 4 tasks. As a result, the total number of tasks in sprint 1 is 11. Therefore, the estimated processing time for backlog items obtained in Table 2 is eight days.

Table 3. Sprint Backlog 2

Product Backlog	Task	Est. (days)
Manage User Data	Create a database schema	0,5
	Create user data UI design	1
	Implementing UI into coding	1
	Testing	0,5
Manage Doctor Data	Create a database schema	0,5
	Create doctor data UI design	1
	Implementing UI into coding	1
	Testing	0,5
Manage Incoming Patient Record Data	Create a database schema	0,5
	Create a UI design for recording admission patients	1
	Implementing UI into coding	1
	Testing	0,5
Manage Death Record Data	Create a database schema	0,5
	Create UI design for a death record	1
	Implementing UI into coding	1
	Testing	0,5

Table 3 consists of four backlog items: handling medical data, inbound patient registration data, death registration data, and user data. The overall number of tasks in sprint 2 is 16, with each queue item containing four tasks. Table 3 shows that the expected working period for backlog products is 11 days. Table 4 includes three backlog itemsdeath reports, handling death files, and death certificatesand two tasks for each backlog item, totaling six tasks for sprint 3. For the backlog products found in Table 2, the expected working period is five days.

Table 4. Sprint Backlog 3

Product Backlog	Task	Est. (days)
Death Report	Implementing death report features into coding	1
	Testing	1
Manage Death Index	Implement the manage death index feature into the Coding	1
	Testing	0,5
Manage Death Certificate	Implementing death certificate management into coding	1
	Testing	0,5

The perspective of agile results from applying the Trello application is shown in Figure 8. The following activity in the scrum stage is the daily Scrum, also known as the scrum meeting. This activity is an evaluation of team members' daily data submissions. This activity is routinely scheduled daily with a 15-minute time limit, from 10:00 to 10:15. This activity was also conducted standing up (stand-up meeting) because standing in a standing position improved creativity and participation. This meeting aims to evaluate

work that has been completed, constraints and solutionsfaced, and the completion target as a reference for the next meeting. Every week, during sprint evaluations, the development team describes the challenges encountered during the sprint and how the solution addresses them in the product backlog that has been accomplished and that the project manager has yet to complete. In addition, the system analyst provides an overview of the week's accomplishments [18].

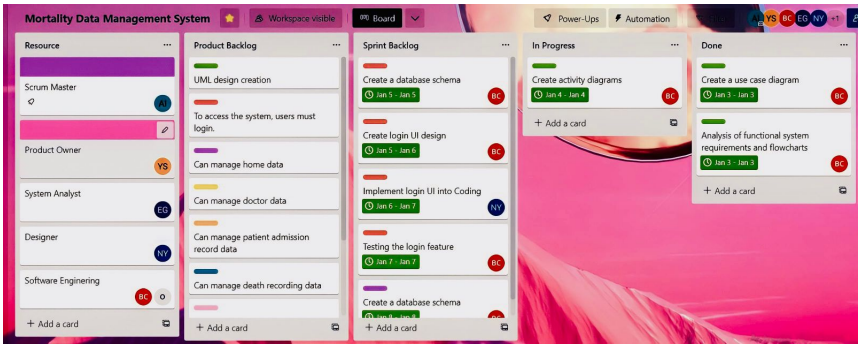


Figure 8. Scrum process in the Trello app

Sprint Retrospective, Once the Sprint Review is complete, The Developer Team and Scrum Master conduct a Sprint Retrospective to assess the team's performance during one sprint. The objective of the concluding phase of this sprint is to evaluate all of the development team's results in reverse order. The Sprint Retrospective does not debate the product to be produced; instead, it concentrates on the development team's performance, both individually and in collaboration with one another. The Scrum Master assignment is to develop answers to evaluation results such as unequal task distribution, individual capabilities disparities, and work estimates recalculations. Continue the conversation with the development team [19, 20]. Figure 9 depicts the final outcome of implementing agile scum in the Trello application.

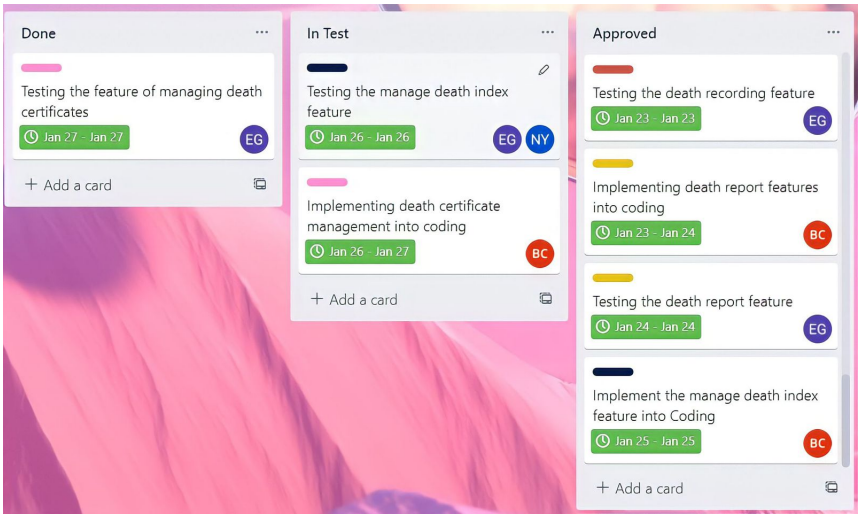


Figure 9. Scrum final state on Trello

Researchers use Microsoft Visual Studio-based C# programming language for the user interface and MySQL database as the database and information from the system during the implementation stage of this mortality data management information system. This stage results from the mortality data management information system's design of the previous phase. The following diagram illustrates how the fatality data management information system displays its output. Every user of this system to administer mortality data must first log in with the username and password depicted in Figure 10. After a successful login, the system will automatically display the primary menu following the user's role.

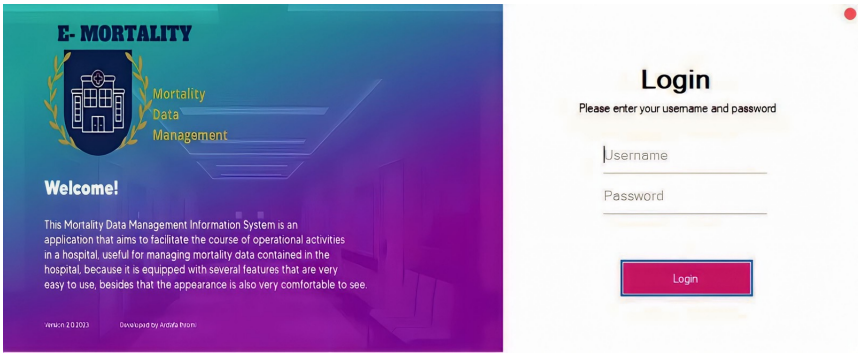


Figure 10. Login Page

Figure 11 depicts the primary menu display, which consists of two primary sections: 1) Master data consisting of user data, doctor data, incoming patient registration data, and mortality registration data. 2) Reports consisting of death certificates, death reports, and death index reports. As depicted in Figure 12, the user list page displays data consisting of user data information. Figure 13 displays doctor information and allows users to contribute doctor information.

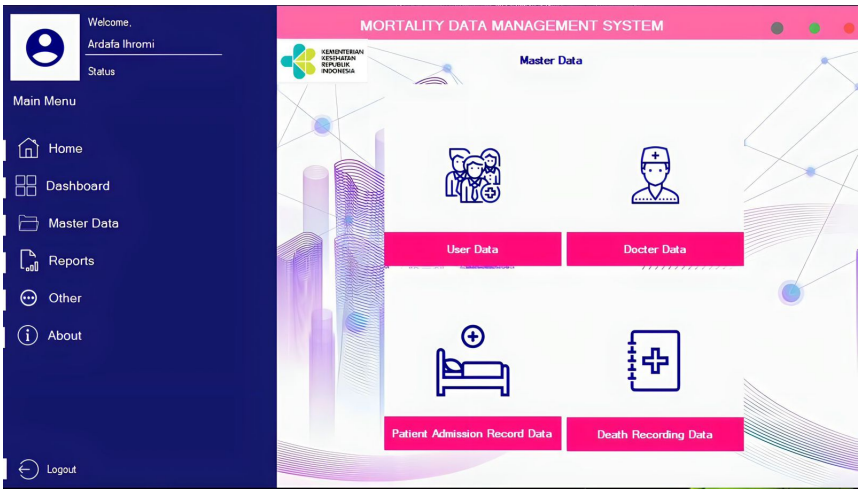


Figure 11. Main Menu Page

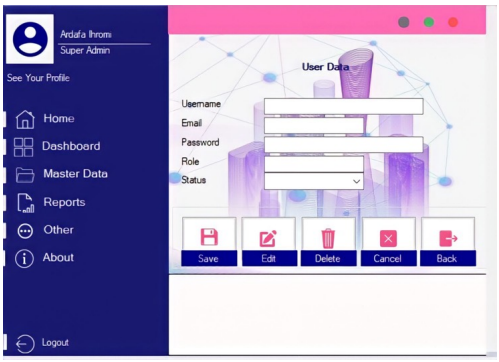


Figure 12. User Page

Figure 13. Doctor Data Page

Figure 14 demonstrates the application interface for recording inpatients or emergency department patients who enter (begin treatment), which includes the patient's name, the administering physician's name, and the chamber where the patient is being treated. This is a crucial component of applications for processing electronic death records because it functions as the application's data repository. In line with Figure 15, it identifies an inpatient or emergency room patient pronounced deceased during hospital treatment and whose information will be administered through a death registration form. This form also documents the primary diagnosis of the cause of mortality and any additional examinations.

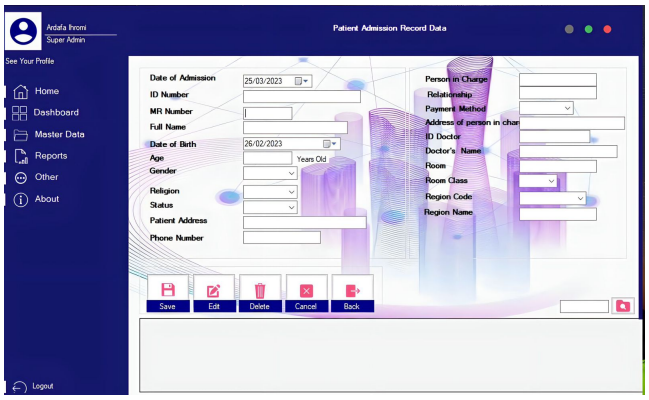


Figure 14. Admission Patient Record Page

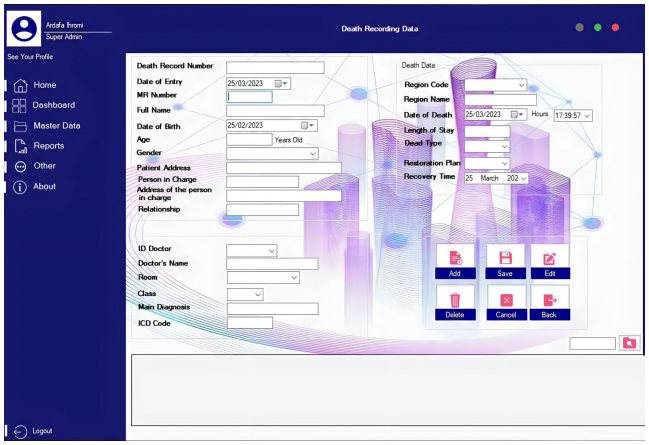


Figure 15. Death Record Data Page

The death report depicted in Table 5 provides an overview of inpatients or emergency department patients who were declared deceased, including their principal diagnoses, cause of death, and discharge plan. While Table 6 is a mortality index displaying a catalog of inpatients or emergency department patients categorized by cause of death. According to what is shown in Figure 16 depicts the processing of mortality data in the preceding phases on a death certificate. The death certificate contains the identity of the deceased, the chronology of death, and the primary diagnosis of the cause of death, which is generated automatically by the system.

Table 5. Death Report

Death Record	Mr Number	Age	Gender	Entry Data	Date of Death	Diagnosa	Doctor	ICD	Room	Death Type
RM0001	263499	44 Y.O	Male	02/11/2022	02/12/2022	Respiratory failure	Rona Kania Utami, dr	J96.9	ER	<48
RM0002	260219	30 Y.O	Female	06/11/2022	06/11/2022	Hypoglycaemia, Unspecified	Adhika Putra, dr	E16.2	ER	<48
RM0003	171240	63 Y.O	Male	22/11/2022	22/11/2022	Cardiac arrest, unspecified	Segal Abdul Aziz, dr, SpPD	I46.9	ER	<48
RM0004	110937	73 Y.O	Female	07/11/2022	09/11/2022	Cardogenic shock	Segal Abdul Aziz, dr, Sp.PD	R57.0	14	>48
RM0005	263869	66 Y.O	Male	11/11/2022	14/11/2022	Chronic kidney disease	Budhi Prihartanto, dr, Sp.PD	N18.9	ICU	>48
RM0006	263901	57 Y.O	Female	10/11/2022	16/11/2022	Pleura effusion,	Budhi Prihartanto, dr, SpPD	J90	25	>48
RM0007	263329	42 Y.O	Female	28/11/2022	08/12/2022	Septicaemic, unspecified.	Amelia Andriani dr., Sp.PD	A41.9	9	<48
RM0008	259251	42 Y.O	Female	08/12/2022	09/12/2022	Hepatic failure,unspecified	Segal Abdul Aziz, dr., Sp. PD	K72.9	14	<48
RM0009	177402	42 Y.O	Male	09/12/2022	12/12/2022	Respiratory failure, unspecified	pen Afgani, dr.Sp.Pd	J96.9	11	<48
RM0010	125621	52 Y.O	Male	06/12/2022	12/12/2022	Pulmonary oedema	pen Afgani, dr.Sp.Pd	J81	ICU	<48

Table 6. Death Index

Number	Mr Number	Age	Gender	Entry Data	Date of Death	Diagnosa	Doctor	ICD	Room	Death Type	Address
1.	263499	44 Y.O	Male	02/11/2022	02/12/2022	Respiratory failure	Rona Kania Utami, dr	J96.9	ER	<48	Bandung
2.	260219	30 Y.O	Female	06/11/2022	06/11/2022	Hypoglycaemia, Unspecified	Adhika Putra, dr	E16.2	ER	<48	Bandung
3.	171240	63 Y.O	Male	22/11/2022	22/11/2022	Cardiac arrest, unspecified	Segal Abdul Aziz, dr, SpPD	I46.9	ER	<48	Bandung
4.	110937	73 Y.O	Female	07/11/2022	09/11/2022	Cardogenic shock	Segal Abdul Aziz, dr, Sp.PD	R57.0	14	>48	Soreang
5.	263869	66 Y.O	Male	11/11/2022	14/11/2022	Chronic kidney disease	Budhi Prihartanto, dr, Sp.PD	N18.9	ICU	>48	Bandung
6.	263901	57 Y.O	Female	10/11/2022	16/11/2022	Pleura effusion,	Budhi Prihartanto, dr, SpPD	J90	25	>48	Cimahi
7.	263329	42 Y.O	Female	28/11/2022	08/12/2022	Septicaemic, unspecified.	Amelia Andriani dr., Sp.PD	A41.9	9	<48	Bandung
8.	259251	42 Y.O	Female	08/12/2022	09/12/2022	Hepatic failure,unspecified	Segal Abdul Aziz, dr., Sp. PD	K72.9	14	<48	Bandung
9.	177402	42 Y.O	Male	09/12/2022	12/12/2022	Respiratory failure, unspecified	pen Afgani, dr.Sp.Pd	J96.9	11	<48	Bandung
10.	125621	52 Y.O	Male	06/12/2022	12/12/2022	Pulmonary oedema	pen Afgani, dr.Sp.Pd	J81	ICU	<48	Bandung

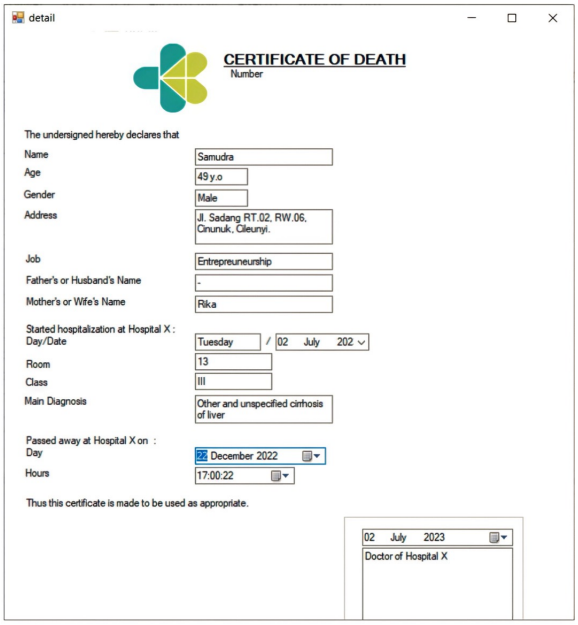


Figure 16. Output Death Certificate

3.3. Comparison with Other Methods

The results of the accuracy value comparison between the proposed model and the various datasets are presented in Table 5. It is unmistakable that the method that was suggested obtained a higher accuracy value. This demonstrates that combining Agile practices with the scrum workflow can improve the validated method's performance and overall effectiveness. Previous research methods that use the waterfall method are arguably the most ancient or conventional because of the static nature of needs. If requirements change, the entire process must be reworked from the beginning, which is time-consuming. In contrast, the Agile Scrum method is more appropriate than the Waterfall method due to its novelty and unpredictability. If the user employs agile Scrum, there is no need for a complete overhaul, and the software development process becomes efficient.

Table 7. Comparison of Method Performance to Previous Studies

Method	Database	Source	Accuracy Rate
SDLC	MySQL	[3]	98%
-	Microsoft Access	[**?**]	87,5%
Waterfall	Microsoft Acess	[4]	89,29%
Waterfall	Microsoft Acess	[5]	92,76%
Agile Scrum	MySQL	The method that the author proposes	99.02%

4. CONCLUSION

Based on research that has been conducted on the creation of a mortality data management information system (death), it can be concluded that the purpose of using a mortality data management information system (death) is to make it easier for medical record officers to handle death data in a computerized manner more effectively in making this software using the Agile Development Method with the Scrum approach pattern to facilitate development so that a faster review of the features being developed is obtained. The stages of the scrum framework are Product Backlog, Planning sprints, Sprint Backlog, Sprint, Scrum meeting, increment, Scrum Review, and Sprint Retrospective, which can overcome all implementation obstacles and create this death statistics data management system. This data system has been designed in an Open-Source manner so that it can be developed and improved by creating additional methods by adding some other features. The best option to develop this system is to use the scrum framework. Users can present it to the development team at any time, and the development team will immediately implement the changes in the next iteration. Some features can be used without waiting to complete the development process, especially in a short time. This information system can generate death data, including death reports, death index reports, and cause-of-death certificates. The system is designed to increase the effectiveness and efficiency of death data processing activities by reducing the error rate in recording death data. Because this system requires maintenance actions, it is recommended that future researchers be more thorough, especially in carrying out system maintenance, so that the update gap from the previously created system can be maximized effectively and efficiently.

5. ACKNOWLEDGEMENTS

All individuals and organizations that have directly or indirectly offered moral and material support to complete this article.

6. DECLARATIONS

AUTHOR CONTIBUTION

The first author was involved in developing the research concept, analyzed the data, drafted the bulk of the paper, and drew the study's conclusions. Next, the proposed system model was developed with input from the second and third authors, who also provided initial guidance. Finally, the study's fourth author completed the study's translation.

FUNDING STATEMENT

No dedicated funding for this study was from any public, commercial, or non-profit organization.

COMPETING INTEREST

All authors declare that they have no financial/non-financial, professional, or personal conflicts that could influence this article's research.

REFERENCES

[1] N. S. Rini and A. Pujihastuti, "Tinjauan Proses Pelaporan Eksternal di Bagian Pelaporan Rumah Sakit Umum Daerahpandan Arang Boyolali," *Jurnal Manajemen Informasi Kesehatan Indonesia*, vol. 3, no. 2, pp. 21–26, 2015.

[2] K. Marzuki, A. Apriani, and N. G. A. Dasriani, "Pemberdayaan Pengrajin Perak Melalui Market Place sebagai Media Penjualan Online di Desa Ungga,Kabupaten Lombok Tengah," *ADMA : Jurnal Pengabdian dan Pemberdayaan Masyarakat*, vol. 1, no. 2, pp. 107–114, 2021.

[3] K. Mustofa and I. Novitasari, "Rancang Bangun Perangkat Lunak Pengelolaan Data Mortalitas Pasien Rawat Inap," *Infokes*, vol. 12, no. 2, pp. 19–24, 2022.

[4] Y. Yuliani, Y. Syahidin, and L. Herfiyanti, "Perancangan Sistem Informasi Indeks Kematian Pasien Covid-19 di RSU Proklamasi," *Bianglala Informatika : Jurnal Komputer dan Informatika Akademi Bina Sarana Informatika Yogyakarta*, vol. 9, no. 2, pp. 79–83, 2021.

[5] S. Rizkita, L. Herfiyanti, and F. Abdussalaam, "Perancangan Sistem Informasi Keterangan Kematian di Rumah Sakit Bhayangkara Sartika Asih," *Cerdika: Jurnal Ilmiah Indonesia*, vol. 1, no. 10, pp. 1377–1388, 2021.

[6] D. M. Wiechmann, C. Reichstein, R. C. Haerting, J. Bueechl, and M. Pressl, "Agile Management to Secure Competitiveness in Times of Digital Transformation in Medium-Sized Businesses," *Procedia Computer Science*, vol. 207, pp. 2353–2363, 2022.

[7] J. Sutherland, "The Scrum Guide ," no. July, 2013.

[8] L. Neelu and D. Kavitha, "Estimation of Software Quality Parameters for Hybrid Agile Process Model," *SN Applied Sciences*, vol. 3, no. 3, pp. 1–11, 2021.

[9] H. Thiele, S. Weber, J. Reichwein, J. A. Bartolo, Y. Tchana, L. Jimenez, and J. C. Borg, "A Scrum Agile Integrated Development Framework," *Proceedings of the Design Society: DESIGN Conference*, vol. 1, pp. 747–756, 2020.

[10] C. Unger-Windeler and K. Schneider, "Expectations on the Product Owner Role in Systems Engineering-A Scrum Team's Point of View," *Proceedings - 45th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2019*, no. June, pp. 276–283, 2019.

[11] F. F. Kurniawan, F. R. Shidiq, and E. Sutoyo, "WeCare Project: Development of Web-Based Platform for Online Psychological Consultation using Scrum Framework," *Bulletin of Computer Science and Electrical Engineering*, vol. 1, no. 1, pp. 33–41, 2020.

[12] Muhammad Nur Afandy, Widodo, and Bambang Prasetya Adhi, "Pengembangan Sistem Informasi Akademik Berbasis Web di SMK Bakti Idhata Jakarta," *PINTER : Jurnal Pendidikan Teknik Informatika dan Komputer*, vol. 4, no. 2, pp. 19–26, 2020.

[13] A. Mundra, S. Misra, and C. A. Dhawale, "Practical Scrum-Scrum Team: Way to Produce Successful and Quality Software," *Proceedings of the 2013 13th International Conference on Computational Science and Its Applications, ICCSA 2013*, pp. 119–123, 2013.

[14] W. K. Pradana, Yopi Firdha, "Rancang Bangun Aplikasi Android Penilaian Kinerja Guru Dengan Metode Agile Development di MAN 1 Kabupaten Malang," *RAINSTEK : Jurnal Terapan Sains dan Teknologi*, vol. 2, no. 3, pp. 182–191, 2020.

[15] A. R. Isnain, D. A. Prasticha, and I. Yasin, "Rancang Bangun Sistem Informasi Pembayaran Biaya Pendidikan (Studi Kasus : SMK Pangudi Luhur Lampung Tengah)," *Jurnal Ilmiah Sistem Informasi Akuntansi*, vol. 2, no. 1, pp. 28–36, 2022.

[16] R. Krishnan and S. V. Bhada, "Integrated System Design and Safety Framework for Model-Based Safety Assessment," *IEEE Access*, vol. 10, no. July, pp. 79 311–79 334, 2022.

[17] A. Ardiansyah, F. Y. Yahya, A. R. Irawati, and M. Yusman, "Pengembangan Sistem Informasi Terpadu Fmipa Universitas Lampung (Simipa) Menggunakan Metode Scrum," *Jurnal Teknoinfo*, vol. 15, no. 2, p. 112, 2021.

[18] R. Gutama and T. Dirgahayu, "Implementasi Scrum pada Manajemen Proyek Pengembangan Aplikasi Sistem Monitoring dan Evaluasi Pembangunan (SMEP)," *Informatics Departement Universitas Islam Indonesia*, vol. Vol 2, p. 7, 2021.

[19] I. Tahyudin and Zidni Iman Sholihati, "Pengembangan Aplikasi Tiga-Tingkat Menggunakan Metode Scrum pada Aplikasi Presensi Karyawan Glints Academy," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 1, pp. 169–176, 2022.

[20] W. Supriyanti and D. A. Pertiwi, "Implementasi Scrum dalam Pengembangan Sistem Informasi Pengelolaan Nilai Siswa," *REMIK: Riset dan E-Jurnal Manajemen Informatika Komputer*, vol. 6, no. 3, pp. 547–560, 2022.