

Personal Extreme Programming with MoSCoW Prioritization for Developing Library Information System

1st Gita Indah Marthasari
Informatika

Universitas Muhammadiyah Malang
Malang, Indonesia
gita@umm.ac.id

2nd Wildan Suharso
Informatika

Universitas Muhammadiyah Malang
Malang, Indonesia
wsuharso@umm.ac.id

3rd Frendy Ardiansyah
Informatika

Universitas Muhammadiyah Malang
Malang, Indonesia
frendy.a@webmail.umm.ac.id

Abstract— Software development projects require experience and knowledge of the developer or clients related to the system which will be developed. Unclear clients' needs potentially emerge many changes of needs during the process of development which can not be resolved by using conventional software development methodology. The implementation of the less significant requirements either from the clients or the other stakeholders causes the project development becomes longer. Therefore, a technique is needed to arrange the priority of software requirements. In this paper, we combine personal extreme programming (PXP) methodology with Moscow technique to overcome those problems. PXP is suitable to use in small to medium-sized projects if the clients do not know in detail about the needs in the development of applications, applications needed in relatively quick time, and the development phase is adjusted to use by a single programmer. Moscow technique was used for prioritizing requirements elicited in PXP methodology. Moscow is a method to determine priority needs based on cost, risk, and business value. This technique was applied during the planning phase of PXP to develop library application, thereby it reduced the time of project completion. The result was a library application suited the needs of clients to support business processes at Batu State Attorney's library.

Keywords—personal extreme programming, Batu state attorney, library information system, moscow prioritization, test driven development.

I. INTRODUCTION

Agile Development is a collection of application development methodology which has an incremental and iterative concept [1], [2]. It focuses on processing application and communication with clients. Its goal is to make faster response change and will reduce the completion time of application development projects. The changes are in the form of cost, requirements, schedule, and team members [2]. Examples of methodologies in Agile Development are SCRUM and Extreme Programming (XP) [3], [4]. The comparison between them is that SCRUM focuses on project management and team member [3], while XP focuses on application programming, feedback, and communication with clients. XP is suitable to use in small to medium-sized projects if the clients do not know in detail about the needs in the development of application and application needed in relatively quick time. In the development, the phase in XP practice is adjusted to use by a single programmer called Personal Extreme Programming (PXP) [5], [6]. In PXP, the priority requirements need to be

determined so that the application is completed on time as user's request. The priority setting in PXP is based on technical risk, and business value [7].

State Attorney is one of the institutions that play an essential role in the control of the law and the welfare of the people, especially in the territory of its power in Indonesia [8]. Based on the preliminary survey, we have found a few weaknesses in the service process in Batu State Attorney. One of the weaknesses is found in library services. A member who is going to search for books should look into the catalog in the library. After getting the book code, the member can start searching for books from the shelves according to the code. This sort of searching process can take some time, especially if the book collection is overwhelming. This makes it difficult for members (as prosecutors) when they need a source of information in a short time to support their work. Another problem is the catalog which material made from paper and has a risk of being lost or damaged. When that happens, a librarian has to rewrite each transaction and collection list from scratch, so it takes a lot of energy and time. Besides, the library has no media to remind the members when the duration of the borrowing is up, so they often forget to return the book which potentially lost. The solution to overcome the problems mentioned above is to make an improvement of the library service. Therefore, we need to develop a web-based system to help library management.

The initial stage of creating an application is to clearly define all needs, controls, and functions [9], but it is known that the clients, as well as the users (librarian), do not have a detailed knowledge of the functional requirements for the library application to be developed. As a result, many changes and adjustments of requirements will potentially occur. Changes to needs that have lack significant functionality in the application will increase the project duration. The solution is involving the clients' roles through intensive communication throughout the project. One of the clients' roles is describing all the desirable needs and determining its priority based on the impact on the functionality of the application. Priority setting of requirements also has an impact on project completion time. In the PXP project, clients play an essential role to determine priority. Referring to the needs and clients' requirements, it is necessary to select the appropriate priority method. A quick and easy approach to help clients set priorities is MoSCoW. MoSCoW is a method for prioritizing requirements based on cost, risk, and business

value. These requirements will be grouped into four categories: Must have, Should have, Could have, and Will not have [10], [11]. Therefore, we created a web-based library system by involving the role of clients during the process of making the application. PXP and MoSCoW were used as the development methodology to support the roles of the clients. The result is a web-based library system of Batu State Attorney to assist the performance of library management and the staff and to overcome the problems mentioned before.

II. RESEARCH METHOD

This research used Personal Extreme Programming (PXP) as the development methodology combine with the MoSCoW as requirements prioritization technique (Figure 1). Overall, we conduct requirements elicitation from clients to identify functional specification of the system then the iteration phase is started. This phase starts with design step and continue until the final step, retrospective, which is the end of iteration stage.

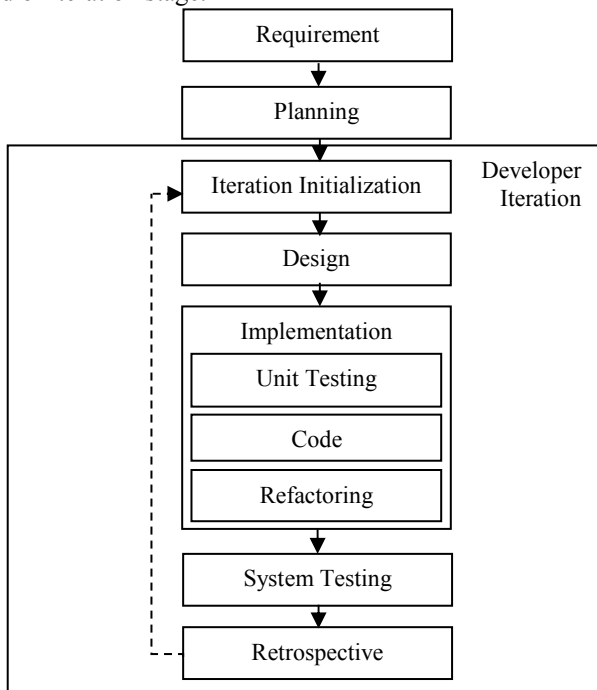


FIG 1. STAGES OF PXP METHODOLOGY [6].

III. RESULT AND DISCUSSION

A. Requirements

This stage was done by collecting data at Batu State Attorney Office. We asked the developers and the clients to describe the needs based on the problems encountered. These needs will be converted into a functional specification in the application library. In the PXP methodology, needs refer to user stories written on cards which are called user story cards. The format of the writing is "As [role] I can [action] so [purpose]" [13]. User story criteria are negotiable, estimable, and testable.

B. Planning

Planning stage describes the procedure on the user story obtained from the requirements stage. We determined the estimated time of implement the user story, the priority needs using Moscow with technical risk criteria, and business value based on [14], [15], and to create a list of release planning. The results of prioritizing Moscow refer to [16] as follows:

Must have criteria lists all of user story meets the criteria of business value and technical risk as follows:

- a. Registering members, this user story must be implemented in the system. It explains the business processes of borrowing books and to do that a user must be registered as a member.
- b. Signing up the book, the user story becomes a significant function because it helps the clients' need of storing and backing up book data, searching, and borrowing books.
- c. Borrowing the book, this user story is the main feature of the system for processing, storing, and backing up data, storing. This functional specification is required for due date notification.
- d. Automatic reminder messages, this user story also becomes the main feature to prevent members who are late returning the book.
- e. Book searching by title, this user story helps users finding the availability of books to be borrowed.
- f. Calculating automatic fines, this user story is essential to help admins calculate the cost of fines for late return of books.
- g. Notice of due date on the admin side, user story plays a role to help admins view the list of borrowed books that have been in due date which is difficult to do manually.
- h. New book notification, user story becomes a key feature to make users aware of the latest book list.
- i. User-specific access, user story restricts and differentiates user permissions.
- j. Restoring the book, user story must exist as a critical feature of the business process of borrowing books on library applications.
- k. Registering a new admin, user story stores admin data and permissions to log into the data library in library application.

Should have category is a user story criteria that does not meet the technical risk criteria, but still has business value or can help the business process of library users. User story in the "should have" category is implemented within the specified time frame, but the project is not considered failed even if not implemented. User story included in this category are as follows:

- a. Changing the website information data, this user story provides library information of Batu State Attorney, such as logos, agency names, addresses, and contacts.
- b. Reminding due date, user story can help users to see the due date of the borrowing through the application so that they might not forget.
- c. Proposing a new book, user story helps members submit book suggestions through library application to complete library book collections.

d. Uploading a book file, this user story is essential as it helps users save book files in softcopy form into the system.

Could have category is a user story that is not included in the criteria of technical risk or business value, but can help to complete the application features. The user story that belongs to this category is viewing a list of new books. The last category is “would not have” criteria that not using user story.

Referring to [17], the result of the release planning list used to develop the application of Batu State Attorney library is shown in table 1.

TABLE 1. LIST OF RELEASE PLAN

	Use Stories	Priority	Point
Iteration 1	Registering the members	Must have	1
	Registering the books	Must have	1
	Inputting the book borrowed	Must have	1
	Automated reminder messages	Must have	1
Velocity			4
Iteration 2	Finding the books based on the title	Must have	1
	Counting the fine automatically	Must have	1
	Accepting the notification of the new books	Must have	1
	Observing the due date of borrowing by admin	Must have	1
Velocity			4
Iteration 3	Registering the new admin	Must have	1
	Special access for a user	Must have	1
	Returning the books	Must have	1
	Changing the website information data	Should have	1
Velocity			4
Iteration 4	Suggesting the new books	Should have	1
	Uploading the book file	Should have	1
	Observing the new books	Could have	1
	Observing the due date of borrowing	Should have	1
Velocity			4

Table 1 shows the results of the release plan list which is determined by the developer along with the clients. There are 4 iterations that must be done in developing the application of Batu State Attorney library. Table 1 shows the title of each user story, each of them has a story point used to describe the difficulty level of working, a user story by the number 1 is estimated to be done within 3 days. One time of iteration has a velocity value of 4, so to complete 16 story points, it takes 4 times of iteration. There is reiteration on iteration development to work on 4 iterations in Table 1.

One iteration development is provided for one time iteration. This paper explains the results of working on the first iteration. Based on the list of planning released in Table 1, the user story done in the first iteration is to register the members, register the books, input the book loans, and automated reminder messages.

C. Iteration Initialization

Based on the list of planning released in Table 1, the user story done in the first iteration is to register the members, register the books, input the book loans, and automated reminder messages.

D. Design

Figure 2 shows a database design table for user stories that consist of a member entity for registering the members, a book entity for registering the books, a loan entity for inputting the loan books, and an automated reminder message using an email attribute in a member entity. In the next phase, the developers begin to write the process of program code according to the design that has been made.

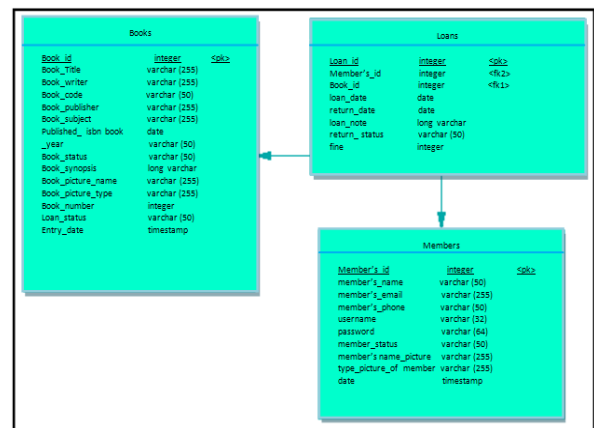


FIG II. DATABASE DESIGN IN THE FIRST ITERATION

E. Implementation

The implementation of the program using Test Driven Development (TDD) consists of 3 stages: unit testing, code generation, and refactoring. The unit testing results of each user story are shown in Figure 3. Member Controller (Controllermember.php) is used for the user story of member registration, Book Controller (Controllerbook.php) for book registration, Loan Controller (Controllerloan.php) for loaning the book, and Email (Email.php) for automatically sends reminding messages. All codes have passed unit testing shown by value of 100%. The process of completing the program code or code generation is done after the program code of each user story passes the unit testing. Besides, it is also created user views from each user story. The user view in one of the users story is shown in Figure 4.

	Lines	Code Coverage				
		Functions and Methods	Classes and Traits			
Total	100.00%	170 / 170	100.00%	15 / 15	100.00%	4 / 4
Controllermember.php	100.00%	42 / 42	100.00%	4 / 4	100.00%	1 / 1
Controllerbook.php	100.00%	61 / 61	100.00%	5 / 5	100.00%	1 / 1
Email.php	100.00%	22 / 22	100.00%	2 / 2	100.00%	1 / 1
Controllerloan.php	100.00%	45 / 45	100.00%	4 / 4	100.00%	1 / 1

FIG III. TESTING RESULT IN FIRST ITERATION

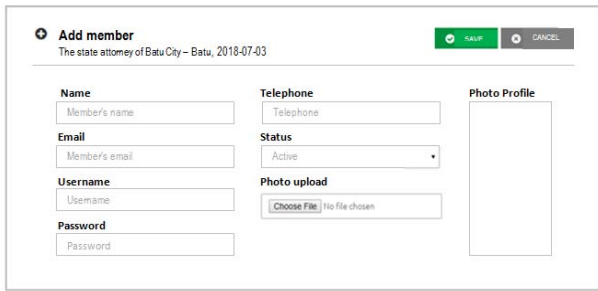


FIG IV.USER VIEW OF REGISTER MEMBER REQUIREMENT

F. System Test

This stage tests the results of the implementation of features that have been obtained. The test applies the user acceptance test criteria. It is based on the user story. The test is done by the client. The test results are shown in Table 2

TABLE II. USER ACCEPTANCE TEST

Iteration 1			Results	
User Stories	Acceptance test criteria	Priority	Yes	No
Registering the members	Notification of member's data is accepted	Must have	√	
	The account can only be registered by admin			
Loaning the books	Notification of data is successfully saved	Must have	√	
	Loan data can be deleted			
	The loan number by member is only allowed for one time			
Registering the books	Data of the books can be deleted	Must have	√	
	There is an image for the book cover			
Reminder message of the due date of loan	The message is sent by email	Must have	√	
	The message is sent constantly until the book returned			

G. Restrospective

Based on the results of the first iteration implementation, verification is done at this stage to ascertain whether the realization time is equal to the initial estimation time. The verification results are shown in table 3.

TABLE III. TIME VERIFICATION RESULT ON THE FIRST ITERATION

Iteration 1				
User Stories	Priority	Story Point	Estimation Time	Realization Time
Registering the members	Must have	1	3	3
Registering the books	Must have	1	3	3
The book loans	Must have	1	3	3
Automated	Must	1	3	6

reminder messages	have			
Velocity		4	12	15

IV. CONCLUSION

Personal extreme programming (PXP) could be used to develop an application fast because it could overcome changing requirements flexibly. MoSCoW method was useful in planning and processing which focused on the main needs. The methodology used in this study was personal extreme programming in developing application of Batu State Attorney library which clients were still unfamiliar with software technology. Based on the research that has been discussed, PXP is able to meet the needs of clients with the construction of Batu State Attorney library applications to overcome the problems encountered. Batu State Attorney library application which has been built using PXP is only based on the client's needs without adding another feature that might be needed in the future. The Moscow approach is able to assist the developers and the clients in determining the priority needs, but it cannot prevent the potential delay time that occurs during the development of Batu State Attorney library applications. It is found that most of the needs have priority "must have" when the determining of Moscow priority is done. However, this makes the processing time longer.

REFERENCES

- [1] S. Balaji, "Waterfall vs v-model vs agile: A comparative study on SDLC," *WATEERFALL Vs V-MODEL Vs Agil. A Comp. STUDY SDLC*, vol. 2, no. 1, pp. 26-30, 2012.
- [2] D. Greer and Y. Hamon, "Agile Software Development," *Softw. - Pract. Exp.*, vol. 39, no. 7, pp. 701-736, 2009.
- [3] J. Highsmith, "What Is Agile Software Development?," *J. Def. Softw. Eng.*, vol. 15, no. 10, pp. 4-9, 2002.
- [4] V. Devedzic and S. R. Milenkovic, "Teaching Agile Software Development: A Case Study," *IEEE Trans. Educ.*, vol. 54, no. 2, pp. 273-278, 2011.
- [5] R. Agarwal and D. Umphress, "Extreme programming for a single person team," *Proc. 46th Annu. Southeast Reg. Conf. XX - ACM-SE 46*, no. August, p. 82, 2008.
- [6] Y. Dzhurov, I. Krasteva, and S. Ilieva, "Personal Extreme Programming-An Agile Process for Autonomous Developers," *Int. Conf. software, Serv. Semant. Technol.*, no. August 2016, pp. 252-259, 2009.
- [7] K. Beck and C. Andres, *Extreme Programming Explained: Embrace Change*. Addison Wesley Professional, 2004.
- [8] P. R. Indonesia, Keputusan Presiden Republik Indonesia Nomor 86 Tahun 1999 tentang Susunan Organisasi dan Tata Kerja Kejaksaan Republik Indonesia. 1999, pp. 1-13.
- [9] R. Fojtik, "Extreme programming in development of specific software," *Procedia Comput. Sci.*, vol. 3, pp. 1464-1468, 2011.
- [10] I. Stamelos and P. Sfetsos, *Agile Software Development Quality Assurance*, no. February. 2007.
- [11] S. Dimitrijević, J. Jovanovic, and V. Devedžić, "A comparative study of software tools for user story management," *Inf. Softw. Technol.*, vol. 57, no. 1, pp. 352-368, 2015.
- [12] B. W. Boehm, "Software Engineering Economics," *IEEE Trans. Softw. Eng.*, vol. SE-10, no. 1, pp. 4-21, 1981.
- [13] M. Kohn, *User Stories Applied for Agile Software Development*. Boston: Pearson Education, Inc, 2004.
- [14] IIBA, *Agile Extension to the BABOK® Guide*. Toronto, Ontario, Canada, 2005.
- [15] IIBA, *A Guide to The Business Analysis Body of Knowledge*. Toronto, 2015.

- [16] Y. Sugianto, S. Tjandra, and P. Method, "Aplikasi Point of Sale Pada Toko Retail Dengan Menggunakan Dynamic Software Development Method," vol. 8, no. 1, pp. 1-8, 2016.
- [17] H. Rizal, S. Adhy, and P. W. Wirawan, "Perancangan Dan Pembuatan Mobile Learning Interaktif Berbasis Android Dengan Metode Personal Extreme Programming," *J. Informatics Technol.*, vol. 2, no. 3, pp. 1-10, 2013.