

# Comparing Traditional and Agile Software Development Approaches: Case of Personal Extreme Programming

Sri Andriati Asri<sup>1</sup>, I Gusti Made Agung Sunaya<sup>2</sup>, Putu Manik Prihatini<sup>3</sup>

Department of Electrical Engineering  
Politeknik Negeri Bali  
Bali, Indonesia

<sup>1</sup>sriandriati@pnb.ac.id, <sup>2</sup>a\_sunaya@pnb.ac.id,

<sup>3</sup>manikprihatini@pnb.ac.id

Widyadi Setiawan<sup>4</sup>

Department of Electrical Engineering  
Udayana University  
Bali, Indonesia

<sup>4</sup>widyadi@unud.ac.id

**Abstract**— Generally, there are two approaches to software development, traditional and agile. Traditional is heavy and well documented, agile is light and less documentation. Personal Extreme Programming (PXP) is one of the agile approaches that iterative, more flexible and responsive to changes. This research compared Waterfall as one of the traditional methods and PXP as agile in the requirement analysis, planning and implementation processes. The project that used to compares those approaches is job training monitoring application. PXP used a closed questionnaire with weighted questions to determined which one of user requirement has the highest priority. The planning phase determined the time estimation for completed each task of making the application. PXP provided more short developing time less than three months. Implementing phase in PXP is a unit testing, code generation, and code refactoring when a user story finished, there for a mistake or an error can be detecting earlier, unlike the waterfall that often tested the application at the end of project development. The result is the agile approach (PXP) gives more convenient and flexibility for developing the application.

**Keywords**— agile; PXP; waterfall; job training application

## I. INTRODUCTION

Comparison between traditional and agile software development methodologies have become an interesting topic for many researchers. The researchers (1,2,3) compared the methodologies by analyzing and deep exploring about the comparison through another researcher results on the topic. They gave a summary of the comparison of traditional and agile methodologies. The Waterfall model is a traditional method which is heavy and well documented. Waterfall first introduced by Winston W. Royce in 1970 (4). This model is a sequential software development process, each phase must be completed before next one can start.

Unlike the traditional one, agile software development methodologies are light and less documented. Agile methodology has various techniques such as scrum, crystal, Extreme Programming (XP) and Personal Extreme Programming (PXP). PXP is a software development process

designed to be applied by software engineers individually and aims a lightening PSP by reducing the number of the script being followed and the amount of data to be filled in the forms (5).

## II. METHOD

### A. Waterfall and PXP

Waterfall model essentially consists of five phases: analysis, design, implementing and maintenance (6). Fig.1 shows the phases of the waterfall model.

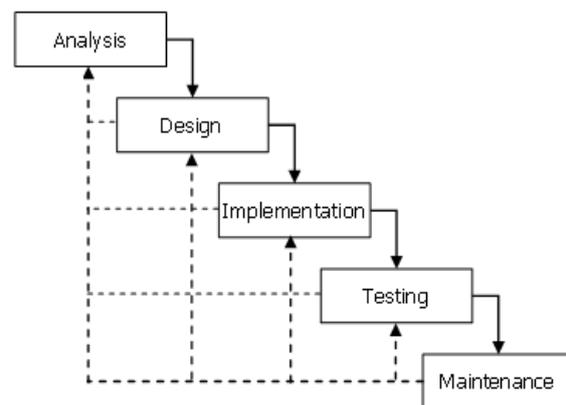


Fig.1 The Waterfall Model phases

According to Dzurov, there are several phases in PXP, requirements, planning, iteration initialization, design, implementation, system testing, and retrospective. Fig. 2 shows the PXP process phases.

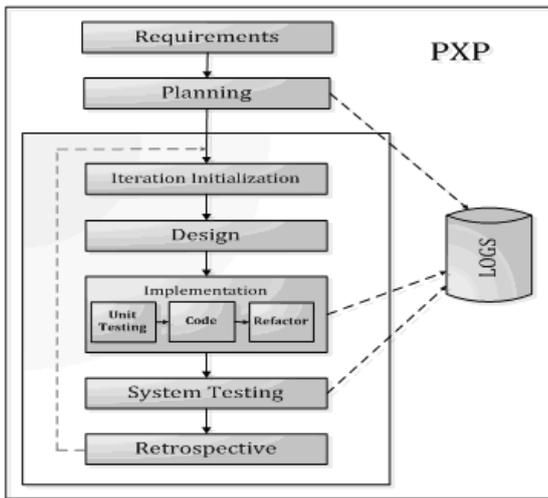


Fig. 2 PXP process phases

The analysis in a waterfall known as Software Requirement Specification (SRS). SRS described a comprehensive description of the software’s behavior to developed. Functional and non-functional requirement. The functional requirements defined the scope, function, user characteristics, interface requirement and database requirement, user interaction with the software. A requirement in PXP has the same function as the waterfall one. The functional and non-functional are created in this phase. The difference is in agile the requirement can adapt the requirements change according to the circumstances, once the requirements are changed the task planning also revised.

Design phase in waterfall model is the phases to defined the software architecture, algorithm, database conceptual and database and user interface design. Before the design phase took place in the PXP model the are two more phases to accomplished first. The Planning phase and Iteration Initialization. Unlike the traditional one, in the planning phase, the estimation time of the project’s modules are detailly defined. Therefore the time’s needed to develop the project can be known as well. The design phase in PXP the developer is modeling the system’s modules and classes. The developer is not allowed to guesses what would the future required.

Implementation phase both waterfall and PXP is where the actual generation code took place through programming. The implementation in PXP the implementation consist of three sub-phases: unit testing, code generation, and refactoring. In the PXP. Unit testing can help the developer knows the error of the code earlier.

**III. RESULT AND ANALYSIS**

This paper compared the waterfall and the PXP in the perspective of how to accomplish the requirement, design and implementation phases. The case is the information system for job training activities (7).

**A. Requirement**

There are several ways to define the requirements, by observation, interview or communication and conduct a survey using a questionnaire. Requirements analysis results in the

specification of software’s operational characteristics indicate software’s interface with other system elements and establish constraints the software must meet (8). Waterfall model needs to well define the requirements at the beginning of the project. it can be done in several ways above. PXP need to gives the priority of each user requirement. The priority is needed to set which one of the requirement has to be deployed first, to get the requirement priority a weighted questionnaire is needed. Observation and interview gave a general point of view of the problems that appeared during job training activities, there are time, guidance and monitoring activities. A weighted questionnaire purposed to get the requirements priority. When the observation and interview methods can not prioritize actual requirements exactly, the weighted questionnaire can provide it. The results of a weighted questionnaire survey on job training activities provide the must-have features of the application are shown in TABLE I (7).

TABLE I. MUST-HAVE FEATURES

Feature	Priority Score
Guidance	1,86
Reporting activities	2,09
Monitoring	2,12
Assessment	2,85

The feature with the lowest point has the highest priority to be deployed first. The main problem with the job training activities is to get guidance from the supervisor and vise versa. The assessment has the highest point which means has because this feature only required by supervisor. Waterfall model often used a use case to described and interpret the requirement, agile PXP used a user story. A user story is used to capture a description of software feature from an end user perspective. It’s containing sufficient information for developers to estimate the effort and time are needed to accomplish the requirement. According to (9) user stories are composed of three aspects: a written description of the user story used to plan and as a reminder, conversation about the story that serves that the flesh out the detail of the user story and tests that convey and document details that can be used to determine when a story is complete. The questionnaire results from TABLE I then break into several user stories. The prioritization of user stories can use any technique, for example, weighted questionnaire. We used small user story. Small user stories have a better change being completed so they reduce the risk of failing in the sprint and also accurately estimated (10). TABLE II shows user stories priority that generated from must-have features in TABLE I. Level priority 1 is the stories to be deployed first which is based on must-have features value.

TABLE II. USER STORIES PRIORITY

Level Priority	User Story
1	<ul style="list-style-type: none"> <li>- As a student, I can get guidance from supervisor so that I can ask about the job training topic.</li> <li>- As a supervisor, I can give guidance to the student so that I can give advice to job training’s students</li> </ul>
2	<ul style="list-style-type: none"> <li>- As a student, I can report my daily activity so that my supervisor lecturer knows.</li> <li>- As a supervisor, I can monitor the student activity so that I can know the daily activity of the job</li> </ul>

Level Priority	User Story
	training's student.
3	As a supervisor, I can fill the assessment score so that I don't need to fill it on the paper

Unlike the waterfall, the PXP's requirement phase is not only determined the requirement but also defines a set of planning task to accomplish a user story and it can adapt the requirements change. If there happen, then the task planning will be revised.

### B. Design

The design phase in the waterfall is to generate a plan to solve or to complete the requirements, which includes architecture, database, graphical user interface design. In PXP, the design phase took place after the planning and iteration initialization. In the planning phase, the developer defines a set of tasks based on the requirements that broke into user stories. It's also estimating time to complete a task (user story). Iteration initialization indicates the beginning of each iteration (sprint). Iteration length could vary from 1 to 3 weeks. After the set of task, iteration initialization of the set of tasks being generated, the developer modeling the system modules and classes. In this research three modules are being developed based on the must-have features of the application: guidance, monitoring, and assessment.

### C. Implementation

The main different of implementing in waterfall and PXP is in PXP the implementation divided into three sub-phases: unit testing, code, and refactoring. Before code generation, a unit testing must be created. The code generation based on the user stories that have been defined in the early phase. It aims to guide the code to complete the requirements specification. Creating a unit testing help the developers to detect an error that occurred at the beginning of development. In waterfall model there no unit testing. A software or application are tested at the end of the project so that an error is detected almost at the end of the project.

### D. Requirement's Change

The advantage of agile methods is the ability to adapt to the changes in requirement. The requirement's changes are adopted in the job training case study. Where there is a new information to share over the application, a new user story has to be created. TABLE III shows a new user story being added.

TABLE III. USER STORIES DESCRIPTION

No	User Story (US) code	Description
Iteration 1		
1.	US 01	Login
2.	US 02	Edit and view profile
3.	US 04	Add guidance
4.	US 05	Send a question text
5.	US 06	Upload a file
6.	US 07	Send an answer text
Iteration 2		
1.	US 08	Add activities
2.	US 09	Approve activities

No	User Story (US) code	Description
3.	US 10	Add assessment score
4.	US 11	Edit assessment score
Iteration 3		
1.	US 12	Download Job Training regulation
2.	US 13	Download Job Training report template.
3.	US 03	Change Password

Iteration 3 shows the user stories from the requirement's change. The change occurs when the student asked about the regulation and the template report of the job training. The developer must to redefined the time estimation to complete the software development.

### IV. CONCLUSION

PXP is more convenient and gave flexibility to developers to develop the software application, especially in requirement and implementation phase. The PXP can adapt the change of the requirement and the revised the time estimation. In the implementation phase the unit testing help developer to detect an error at the beginning of software deployment.

### ACKNOWLEDGMENT

The authors would like to thank P3M Politeknik Negeri Bali and the Ministry of Research and Technology of Higher Education of the Republic of Indonesia for the funding this research.

### REFERENCES

- [1] M. Stoica, M. Mircea, and B.G. Micu, "Software Development: Agile vs Traditional," *Informatica Economica*, vol 17, no.4, pp 64-76, 2013.
- [2] A.B.M. Moniruzzaman and S.A. Hossain, "Comparative Study on Agile Development Methodologies," *Glob. J. Comp. Sci. and Tech.*, vol. 13, no. 7 Issue, pp 5 - 18, 2013.
- [3] N. Bhadoriya, N. Mishra, and A. Malviya, "Agile Software Development Methods, Comparison with Traditional Methods and Implementation in Software Firm," *Int. J. Eng. Res. Tech.*, vol. 3, no. 7, pp 1656-1662, 2014.
- [4] W.W. Royce, "Managing the Development of Large Software Systems," *Proceed. IEEE Wescon*, pp 1- 9, 1970.
- [5] D.Y. Krastieva and S. Ileava, "Personal Extreme Programming – An Agile Process for Autonomous Developers," *Proceed. pp 252-259 [International Conference on Software, Services and Semantic Technology, 2009]*.
- [6] Y. Bassil, "A Simulation Model for Waterfall Software Development Life Cycle," *Int. J. Eng. Tech.*, vol. 2, no. 5, pp 742 - 749, 2012.
- [7] S.A. Asri, I.G.A.M. Sunaya, E. Rudiastari and W. Setiawan, "Web Based Information System for Job Training Activities Using Personal Extreme Programming," in *Journal of Physic: Conf. Series 953 (2018) 012092*, 2018
- [8] R.S. Pressman, *Software Engineering: a Practitioner's Approach*, Seventh Ed, MacGrow\_Hill, New York, 2010.
- [9] M. Cohn, *User Story Applied For Agile Software Development*, Addison Wesley, 2009.
- [10] V. Gaikwad and P. Joeg, "An empirical study of writing effective user stories," *Int. J. Software Eng. and Its App.*, vol 10, no. 11, pp 387 - 404, 2016.