Insert Your Title Here∗

Insert Subtitle Here

FirstName Surname†  
 Department Name  
 Institution/University Name  
 City State Country  
 email@email.com

FirstName Surname  
 Department Name  
 Institution/University Name  
 City State Country  
 [email@email.com](mailto:email@email.com)

FirstName Surname  
 Department Name  
 Institution/University Name  
 City State Country  
 email@email.com

ABSTRACT

Throughout the development of the computer architecture and software complexity, it becomes inevitable for computer scientists or engineers to develop a software following a rigorous process.

After several years researching and improving regarding the process of software development, engineers classified software process into these fixed categories: Waterfall model, Incremental model and Iterative model. In practice, none of those models per se are suitable for developing a software product. However, by combining partial feature of those model, software engineers could optimize their workflow based on the developing context.

In this paper, we will discuss the method of collaborating between several iterative model, specifically for *Agile* process. We divide our discussion into 4 parts: general background of software development process, the importance of software architecture and design, collaborating method for agile teams, and finally the conclusion and summary.

In the first section, several processes are proposed. We will compare those models parallelly and most importantly, the comparison between those process with Agile process. The difference of those processes are important and substantial among the process evolution, which are solid factors of influencing the collaborating model for the later processes, including the well-known Agile model.

The second section introduces the general architecture of the software using *Agile* developing method. Before we dive into the topic of our paper, we believe it is essential to review the essence of Agile development process. Rather than comparing Agile model to other models, this section focusing on introducing the formal Agile development for a single software development project.

After reviewing the formal process of Agile development architecture, this section includes several methods of collaborating in different agile teams. The methods of the collaboration, even in Agile solely, are diverse based on the fundamental software as well as the clients/businesses models. In this section, we will first introduce some characteristic of collaborating models and then include the high-level collaborating process based on Agile methodology.

CCS CONCEPTS

• Insert CCS text here • Insert CCS text here   • Insert CCS text here

KEYWORDS

Communication, 12 Agile Principles

ACM Reference format:

FirstName Surname, FirstName Surname and FirstName Surname. 2018. Insert Your Title Here: Insert Subtitle Here. In *Proceedings of ACM Woodstock conference (WOODSTOCK’18). ACM, New York, NY, USA, 2 pages.* https://doi.org/10.1145/1234567890

1. Background

2. The Importance of Design and Architecture

2.1 Continuity in Design and Architecture

3. Important Characterizes of Agile Teamwork

3.1 Communication

Since the requirements and goals can be changed in the agile developing process, every team should keep good communication. And there is a daily stand up meeting in the agile process, which can reflect the progress and status of every team. Then, the Scrum Master should take responsibility to exchange information with other agile teams and update the information to team members. “Agility requires that teams have a com-mon focus, mutual trust, and respect; a collaborative, but speedy, decision-making process; and the ability to deal with ambiguity.”[[1]](#footnote-2) There are many conflicts and problems to deal in the agile process because every agile team is self-organized and independent so that every team probably have different ideas and comprehension to the same thing. Therefore, keeping communication between each team is the most important thing during developing. In order to prevent endless argument, only Scrum Master or representative of each team can swap the advises if it occurs some conflicts or misunderstanding.

3.2 Continuity

3.3 Consistency

ACKNOWLEDGMENTS

Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here.

REFERENCES

[1] A. Cockburn and J. Highsmith, "Agile software development, the people factor," in Computer, vol. 34, no. 11, pp. 131-133, Nov. 2001.  
doi: 10.1109/2.963450

[2] Sten Andler. 1979. Predicate path expressions. In *Proceedings of the 6th. ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL '79)*. ACM Press, New York, NY, 226-236. DOI:https://doi.org/10.1145/567752.567774

[3] Ian Editor (Ed.). 2007. *The title of book one* (1st. ed.). The name of the series one, Vol. 9. University of Chicago Press, Chicago. DOI:https://doi.org/10.1007/3-540-09237-4.

[4] David Kosiur. 2001. *Understanding Policy-Based Networking* (2nd. ed.). Wiley, New York, NY..

Conference Name:ACM Woodstock conference

Conference Short Name:WOODSTOCK’18

Conference Location:El Paso, Texas USA

ISBN:978-1-4503-0000-0/18/06

Year:2018

Date:June

Copyright Year:2018

Copyright Statement:rightsretained

DOI:10.1145/1234567890

RRH: F. Surname et al.

Price:$15.00

1. A. Cockburn and J. Highsmith, "Agile software development, the people factor," in Computer, vol. 34, no. 11, pp. 131-133, Nov. 2001.  
   doi: 10.1109/2.963450 [↑](#footnote-ref-2)