

Open Source Tools to Support Teaching Agile Software Development

Panji Wisnu Wiryawan

Lecturer at Jurusan Informatika, Universitas Diponegoro Semarang, maspanji@undip.ac.id

Satriyo Adhy

Lecturer at Jurusan Informatika, Universitas Diponegoro Semarang, satriyo@undip.ac.id

Abstract - Learning agile software development methodologies are important due to the popularity of agility in software industry. Agile software development has several practices and each practice needs specific tools to work with. Fortunately, there are plenty of open source tools to support working with the agile practices. However, each tool is a separate tool and there is no information about the interrelation of those open source tools. In this paper we propose a set of open source tools to support agile software development course. We start from identifying the principles and practices of agile software development and continue with examining open source tools that fit with agile practices. The relationship between the open source tools is also determined, based on their functionalities.

Keyword - Open Source Tools, Agile Software Development, Course

1. Introduction

Agile Software Development (ASD) has become an interesting study to become a part of the curriculum in higher education [1], [2], [3], [4]. Convening this course because ASD is a method that able to handle the uncertain changing needs of consumers. The interest of the Agile Software Development was based on its ability to address the uncertain needs of consumers [1], [5].

In the course of ASD, the Extreme Programming (XP) method can be used as a reference. The reasons for selecting XP is this method designed to adapt the agile principles and practices to be used with either [6]. So expect to know one method, at least can represent ASD.

Agile Practice is an activity that is clear and has a goal which is carried out to develop software [7]. These Agile Practice needs to be delivered in the course so that students can develop the software in agile. Some Agile Practice such as test first programming and metaphor system require tools.

To facilitate students in getting the tools, open source tools selected which can be downloaded via the Internet for free. These tools will be used to support the explanation of Agile Practice and student can apply agile practice by themselves to develop the software.

There are several open source tools to support Agile Practice, but each tool is a separate tool. It is necessary to study the relation of each tools so the student can used them to learn and apply agile practices in agile software projects.

This article reviews the linkages between tools to support the delivery of agile practices in the classroom. Starting from identifying Agile Practice and which Practice that requires and doesn't require tools. The discussion continued by identifying several open source tools for each practice. Finally, mapping the function of each tools and also explained the activities which the student can do with the tools.

2. Agile : Principles and Practice

Agile software development has the values, principles and practice. Reference [7] shows that the value is something more desirable or not in software development. While the practice is a translation of these values through the principles.

Reference [8] defines 17 agile practices were grouped by phase in the software development phase. While Reference [5] defines the 12 XP practices which are considered as "best practice". What is revealed by Schneider and Mike is disjoint. In this article, used 12 XP Practice as agile practices which then adapted to the course. Practices are shown in Table 1 and included to illustrate the description of each practice.

Table 1. Agile Practice and description

Practice	Description
System Metaphor	Programmer with customer communication media.
Always Use The Simplest Solution that add s business values	Using the number of classes and minimal methods to create a test passed.
The Planning Game	Planning a release that will be done through stories.
Small, Frequent release	Release something that has business benefits.
Forty-Hour week	Work quickly with maintaining quality.
Continuous	Integration code frequently.

Practice	Description
integration	
Collective code ownership	Code belongs to all programmers
Refactoring	Improving/changing structure without changing functionality
Test First Programming	Test before develop code.
Pair Programming	Code together in one machine.
Coding standard	Rules in writing code to make readable code.
On-Site customer	Working in customer place.

Student will learn the twelve practices in Table 1. In addition, the student will apply these practices in developing software in the course of ASD. However, not all practice require tools. Pair programming, coding standards, and onsite customer is a practice that doesn't require tools.

3. Open Source Tools To Support Agile Practice

Reference [9] lists several open source tools that can be used to implement agile practices, especially in the use of Extreme Programming and Java programming. However, in the explanation, all the tools described are likely to the testing framework, has not touched all the aspects of existing practices.

Open source tools mentioned in this article are adapted from some of the tools that have been mentioned by Burke, coupled with some of the tools associated with Extreme Programming practices. Table 2 shows some of the open source tools that can support the agile practices. Description of each of the tools in Table 2, taken from the website of the each tools.

Table 2. Open source tools

Open Source Tools	Description
UMLet [10]	Java-based UML tool to draw UML diagram. This tools can export its document to several format like PDF, JPG and SVG.
Agilefant [11]	Web based tools for agile management. It support multi team, user story writing and release planning.
Jenkins [12]	Continuous integration system that can build and also test software continuously
Cruise Control [13]	Continuous integration system that includes a variety of source control and build technology.
SVN Server [14]	Version control system (VCS) that manages document (file or folder) version. Support collaboration for many users.

Eclipse IDE [15]	Integrated Development Environment (IDE) that supports many programming language for development and supports JUnit plugin.
Netbeans IDE [16]	Integrated Development Environment (IDE) that supports many programming language for development and supports JUnit plugin.
JUnit [17]	Open source testing framework to work with Java.

Please note that tools such as UMLet, Agilefant, Jenkins, Cruise Control, as well as the SVN server mentioned above are tools that are independent of the programming implementation. While the Eclipse IDE, NetBeans IDE, and JUnit is a tool that is primarily used to create Java programs. In this article, the tools used for development is limited only to Java programming.

4. Relationship Among Tools

In this section will be mapped open source tools that support each practice. In fact, some tools can support more than one practice. This further facilitates the execution of any practice because it can be related to one practice to another. This section begins with an explanation of the need for these tools, followed by its use for software development, and ends with the table presentation of each tools linkage.

At the time of course implemented, student not only required to understand the twelve agile practice / XP practices, but are also required to be able to implement each practice in an agile software development project. Student need these tools for software project development. Although the tools are different, but still can be used for project development and its use can be orchestrated. Orchestration use of these tools is based on Extreme Programming development lifecycle.

Before the student can run the XP practices, the student needs to do some set up in the lab or in other places that support the use of server and network access. Tools such as CruiseControl or Jenkins, SVN Server and Agilefant need to be set up in a machine that can be used by the student on the same team.

Student will start the project by forming a team and get a case. The problem in such cases can be described using a user story. The use of user stories is an attempt to implement the Always Use The Simplest Solution that add s business values.

Then, each story will be given the following estimation, priority, and release scheduling. These ability is owned by Agilefant tools. Practices that can be implemented is the planning game, small frequent releases, and forty-hour week, because in agilefant allows small release plan.

Some user story and possibly the system itself, may be difficult to understand. Therefore, a system metaphor needed to explain about the system can be more elusive. UML can be used to understand the purpose of the system. To draw UML used UMLet. UML images

can be stored on the SVN server documentation for future development needs.

Table 3.Mapping Practice with open source tools.

Supported Practice Tools	System Metaphor	Use The Simplest Solution	Planning Game	Small, Frequent Release	Forty -Hour Week	Continuous integration	Collective Code ownership	Refactoring	Test First Programming
UMLet	Yes								
Agilefant		Yes	Yes	Yes	Yes				
Jenkins						Yes			
Cruise Control						Yes			
SVN Server							Yes		
Eclipse							Yes	Yes	Yes
Netbeans							Yes	Yes	Yes
JUnit									Yes

Furthermore, the student can proceed with the development of each of these stories as a team. The development of each of these stories could use Netbeans IDE and Eclipse IDE. Because there is a practice test first development, JUnit testing framework is used as a tool to test first development. Another practice that can be used when development is pair programming.

In software development, student will continually integrate code with fellow members of the team. It required the practice of continuous integration. Student can use Jenkins or CruiseControl in performing this practice.

Each time the code has been integrated, in-commit code to CVS so that code can be read by version and other team members (collective code ownership practice). Activities ranging from test first development to commit code into the CVS which repeated constantly until the complete release (small release) and the completed project.

An overview of the implementation of the development by the student above can be summarized in the activity diagram figure 1.

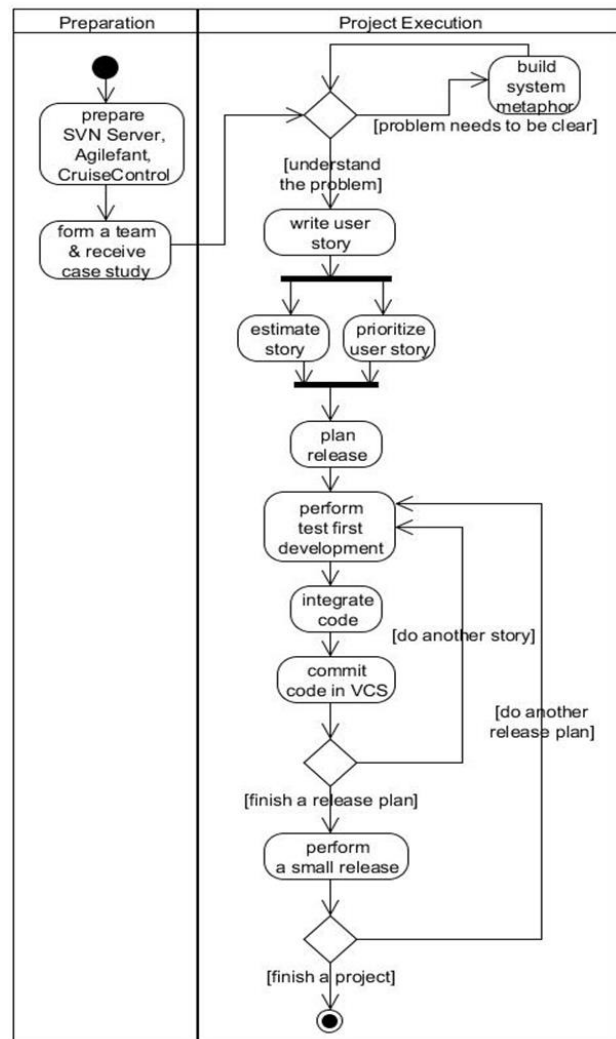


figure 1. activity diagram

5. Conclusion

Agile Software Development course requires tools, and luckily there are tools that are open source and can support all existing practices in Agile Software Development. Basically, these tools are vary but can be used simultaneously by knowing the agile practice. Not

enough with just the knowledge of agile practices, knowledge of one of the Agile methods, in this case Extreme Programming, needed to be able to use these tools simultaneously.

6. References

- [1] Buchalcevoca, Alena, "Where in the curriculum is the right place for teaching agile methods?", Prague, In: Proceedings 6th ACIS International Conference on Software Engineering Research, Management & Applications (SERA 2008), pp 205–209, 2008.
- [2] Andreas Schroeder, Annabelle Klarl, Philip Mayer, Christian Kroiß, "Teaching agile software development through lab courses," Global Engineering Education Conference (EDUCON), 2012 IEEE , vol., no., pp.1,10, 17-20 April 2012 doi: 10.1109/EDUCON.2012.6201194.
- [3] Orit Hazzan, Yael Dubinsky, "Why software engineering programs should teach agile software development", SIGSOFT Softw. Eng. Notes Volume 32, Number 2, pp 1-3, March 2007, doi=10.1145/1234741.1234758.
- [4] Carlson, B., "An Agile Classroom Experience: Teaching TDD and Refactoring," Agile, 2008. AGILE '08. Conference , pp.465,469, 4-8 Aug. 2008, doi: 10.1109/Agile.2008.39.
- [5] Mike Holcombe, Running an Agile Software Development Project, A John Wiley & Sons Inc., 2008.
- [6] Gert van Valkenhoef, Tommi Tervonen, Bert de Brock, Douwe Postmus, "Quantitative release planning in extreme programming, Information and Software Technology", Volume 53, Issue 11, November 2011, Pages 1227-1235, ISSN 0950-5849, <http://dx.doi.org/10.1016/j.infsof.2011.05.007>.
- [7] Kent Beck, Cynthia Andres, Extreme Programming Explained, 2nd edition, Addison-Wesley, November 2004.
- [8] Jean-Guy Schneider, Rajesh Vasa, "Agile practices in software development - experiences from student projects," Software Engineering Conference, 2006. Australian , vol., no., pp.10 pp.,410, 18-21 April 2006. doi: 10.1109/ASWEC.2006.9
- [9] Eric M. Burke, Brian M. Coyner, "Java Extreme Programming Cookbook", O'Reilly, 2003.
- [10] Free UML Tool For Fast UML Diagram, <http://umlet.com>.
- [11] Open source web based tool for agile management, <http://agilefant.com/open-source>.
- [12] An extendable open source continuous integration server, <http://jenkins-ci.org>.
- [13] A continuous integration tool and an extensible framework for creating a custom continuous build process , <http://cruisecontrol.sourceforge.net>.
- [14] Enterprise-class centralized version control for the masses, <https://subversion.apache.org>.
- [15] Eclipse IDE for Java developers , <https://www.eclipse.org/downloads/>.
- [16] Netbeans IDE, <http://netbeans.org>.
- [17] JUnit : Simple framework to write repeatable tests, <http://junit.org>.

7. AUTHOR'S PROFILE



Panji Wisnu Wirawan, S.T, M.T

Lecturer at Jurusan Informatika
Universitas Diponegoro Semarang

maspanji@undip.ac.id
+62 813 2805 3844



Satriyo Adhy, S.Si, M.T

Lecturer at Jurusan Informatika
Universitas Diponegoro Semarang.
Active research in : Software
Engineering and Information System.

Email : satriyo@undip.ac.id
Phone : +62 896 99 120 338