Work in Progress - Project Practices of Agile Software Development for Undergraduate Students

Kazuhiro Matsuo, Shota Anzawa
Kanazawa Institute of Technology, kmatsuo@neptune.kanazawa-it.ac.jp, anzawa.shota@gmail.com

***Abstract* - As information system becomes core infrastructure of our society, the size and complexity of software products increase rapidly. So dependable and efficient software development methods are demanded and new methods such as an agile method are introduced. It is, however, not easy to teach and excise processes of software development. Most of undergraduate students in computer and software engineering courses are busy for mastering basic programming skills and for understanding computer and DB systems. There is almost no room for learning skills for software development in Japan. There are, however, always some percentages of students who can easily master basic skills and knowledge and are ready to learn more. So we decided to start a project practices as an extracurricular activity. The project is aimed for students to learn an agile method and actual experiences of group development processes. In this presentation we report our work in progress of extracurricular activities for project practices of agile software development to highly-motivated undergraduate students.**

Index Terms – Agile method, extracurricular project, extreme programming, ticket driven development.

INTRODUCTION

As information system becomes core infrastructure of our society, the size and complexity of software products increase rapidly. So dependable and efficient software development methods are demanded and new methods such as an agile method are introduced. It is, however, not easy to teach and excise processes of software development. Most of undergraduate students in computer engineering and software engineering courses are busy for mastering basic programming skills and for understanding computer and DB systems. There is almost no room for learning skills for software development in Japan.

There are, however, always some percentages of students who can easily master basic skills and knowledge and are ready to learn more. So we decided to start a project practices as an extracurricular activity. The project is aimed for students to learn an agile method and actual experiences of group development processes. At first trial we recruited several undergraduate students with good programming skills. A project manager was assigned by one of the authors who learned it during his six-month internship at an IT vender. At the beginning the project, we asked the same IT vender to supervise the project. By preparing software development environment, we started the three-month project to develop KIT Groupware, groupware system for university laboratories. The reason to choose the system is the one which students can easily figure out actual requirements from their school life. Last December we have accomplished first three-month project with multi-iterations. Here we present preliminary analysis of our activities with the overview of our project practice.

AGILE SOFTWARE DEVELOPMENT METHOD

Agile software development method is a method of iterative development in which development processes are composed of numbers of iterations within a master schedule of the development, shown in Fig. 1.

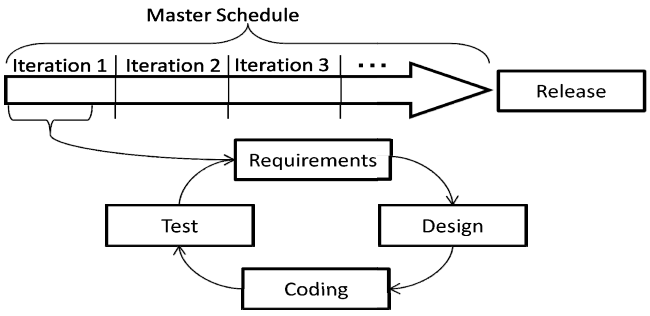

Fig. 1 Agile Software Development Process

The method is quite effective in a case when frequent changes in specifications occur during development process from customer's requests. There exist many variations of the method. XP (Extreme Programming) [1] and Scrum are well known among others.

OVERVIEW OF PROJECT PRACTICES

In the end of July we selected seven project members from many applicants by checking their programming skills. They are two sophomores and five juniors. During summer vacation we ask them to work on preliminary tasks of software development as well as programming drills. As mentioned, one of the authors becomes a project manager with his internship experience. We asked the same IT vender to supervise and to review the procedure at the end

of iteration. These supports we have received are keys to success in the practice.

We picked up Ruby on Rails as our software framework and other software development environments are as follows;

- Development environment: Eclipse, MySQL, Mongrel.
- Programming Language and framework: Ruby on Rails.
- Management tools: Redmine, Subversion.

ACTUAL PRACTICES

The practice in the project ends in three months in which six iterations are included.

Iterations: Iteration is composed of two weeks. The first day of iteration is the day for planning and the last day for reporting back. On the day of reporting back project members show their demos of their software developed and discuss about process improvements using KPT Kanban.

TiDD: TiDD stands for Ticket-Driven Development, which is a management method of agile development using error reporting tools such as Redmine and Trac. [2] In our case we use Redmine and manage development tasks and individual tasks by dividing them into task units which we call Ticket.

Pair Programming: Project members do their programming tasks by pairs. This programming style provides high quality codes, smooth knowledge transfer, efficient learning, and teamwork upgrading. [3]

Automated Testing: Testing procedures are automated for developed codes. At end of the day, all codes developed are checked to work properly by Test Server and test results are automatically reported to client PCs.

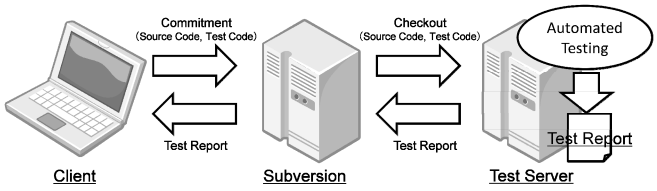

Fig. 2 Automated Testing Process

Code Coverage: We measured the percentage of code coverage obtained from test scripts generated. We use rcov, code coverage tool in Ruby.

Kanban Board: Kanban board used for visualizing what to do and which tasks members are doing in the iteration.

Burn-down Chart: This is a progress chart for evaluating work progress. There are an ideal progress line and an alarm line. The former is initially estimated progress line without any risk considerations. The latter is initially estimated progress line with empirical risk estimations. At the end of iteration, actual progress result is marked on the chart for checking.

Reviewing: The last day of iteration is the day of reviewing. Members report good and bad points of development procedures in the iteration. KPT Kanban board is used in reviewing.

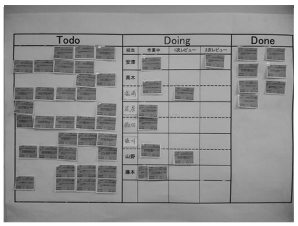

Fig. 3-1 Kanban Board

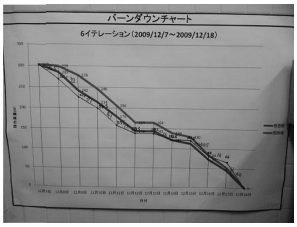

Fig. 3-2 Burn-down Chart

REVIEW OF PROJECT PRACTICES

After the three months practice we collected performance data. We analyze the performance from the QCD points of view, i.e., Quality (code quality), Cost (working hours), and Delivery (work achievement).

Code Quality: Ninety six stand-along tests and seven hundred forty two function tests were done. The code coverage is 91.1%.

Operation Cost: We could not estimate operation cost of student project, so we compared resulted total working hours with estimated working hours in iterations. Overall working hours are 13% shorter than the estimated hours.

Work Achievement: Average work achievement of iterations is 96.8%.

CONCLUSIONS

The project practices from requirement analysis to code testing were accepted to undergraduate students with great satisfaction. They were excited to learn in the project. The success might come from the selection of the students. We found out there are some trials in regular classroom in Japan, but they are not so successful as ours. We started the second trial and will be able to report the comparison between two trials at the conference.

ACKNOWLEDGMENT

The authors express their gratitude to Fujitsu Limited for helping the project practices. The project is partly supported by Program for Promoting University Education Reform of MEXT, Japanese Government.

REFERENCES

[1] Beck, Kent and Andres, Cynthia. 2004. *Extreme Programming Explained 2nd Ed.*, Addison-Wesley Professional.

[2] Sakai, Makoto. "TiDD: Ticket Driven Agile Development Method" (in Japanese), 2009. *5th Workshop of Forum on Reliable Computer Software*. http://sakaba.cocolog-nifty.com/PDF/TiDD_FORCE09.pdf

[3] Williams, Laurie and Kessler, Robert. 2002. *Pair Programming Illuminated*. Addison-Wesley Professional.

AUTHOR INFORMATION

Kazuhiro Matsuo Professor, Kanazawa Institute of Technology, kmatsuo@neptune.kanazawa-it.ac.jp.

Shota Anzawa, Graduate Student, Kanazawa Institute of Technology, anzawa.shota@gmail.com.