

IMPROVEMENT OF COLLABORATIVE WORK ON SOFTWARE DEVELOPMENT PBL IN A DISTRIBUTED ENVIRONMENT

Satoru Kizaki¹ and Yoshihide Chubachi²

¹ Graduate School of Information Systems,
The University of Electro-Communications,
Japan

² Advanced Institute of Industrial Technology,
Japan

ABSTRACT

In this paper, we propose an effective management method. The project manager would lead a project to a successful conclusion on software development PBL in a distributed environment. This method focuses on the real-time nature of collaborative work under the ticket-driven development. Ticket-driven development is a development style which manages by dividing work into tasks and assigning tickets of BTS (Bug Tracking System). To initiate a ticket is published first. By basing on a ticket can be reduced to disagreement of communications. We have investigated university-industry collaboration project "Collaborative Management Laboratory", intended for undergraduate students of Keio University in the fiscal year 2010. The attempt was to implement a training course where both undergraduate students and engineers in industry collaboratively learn a construction of information systems through project-based learning from the fiscal year 2005. The project was composed of three to five students and the project managers from an IT company. There were companies or personal clients that actually needed the developed systems in this project. We aimed for the development of software for general consumers. The project has aimed at the information systems engineer's training by the PBL execution from the educational perspective; however, the results are requested to be put out efficiently within a short term (about four months) as the development period. Therefore, the agile software development method was the first choice. It was considered effective. In this project, we previously used the agile software development method, several practices from the project was used. We did not succeed, because this project had communication problems within the team. In the project for the fiscal year 2011, the communication between project manager and students have been improved by applying the proposed method.

KEYWORDS

Project-Based Learning, Agile software development, Task management, Collaborative work, Ticket-driven development

INTRODUCTION

In this paper, we speak about the university-industry collaboration project "Collaborative Management Laboratory"[1], intended for undergraduate students of the Keio University in fiscal year 2010 and fiscal year 2011. The attempt was to implement a training course where both undergraduate students and engineers in the industry collaboratively learn a construction of information systems through Project-Based Learning (PBL) that began in fiscal year 2005. The project was composed of three to five students and the project managers from an IT company. There were companies or personal clients that actually needed the developed systems in this project. We aimed for the development of software for general consumers. The project has aimed at the information systems engineer's training by the PBL execution from the educational perspective; however, the results were requested to be put out efficiently within a short development period (about four months). Therefore, the agile software development method was the first choice. It was considered effective. 4 teams were formed in the fiscal year 2010. One of the authors (Kizaki) participated in one project as a project manager. However, this project had problems in task management and communications within the team. We proposed to focus on the real-time attribute of collaborative work under what we call the ticket-driven development. The ticket-driven development technique was taken in, and also the educational environment was changed in the fiscal year 2011. Although the fiscal year 2010 was only instructed by a teacher, specialists of agile development participated from actual companies. Moreover, one engineer, a working member of information technology society, was assigned to each team, and performed technical support. 3 teams were formed in the fiscal year 2011. The author Kizaki participated in the project realizing "the next-generation mobile network service" using the Android phone as a project manager. This project used Robot Service Network Protocol (RSNP), a robot service-oriented protocol specification. The valuation method analyzed the comparison with the last fiscal year by EVM. Questionnaire was carried out for qualitative evaluation in the project.

THE PROJECT IN THE FISCAL YEAR 2010

This chapter describes the outline of the project in the fiscal year 2010. One of the authors (Kizaki) participated in a project. He was appointed Project Manager (PM) for this project in October, 2010. There were three members (Third year student A, Second year student B & First year student C). Students had programming experiences, Java grammar skills, and object-orientation skills. This project aimed at extending the existing number of users by repairing the present system. This system is called KATAZOU (<http://katazou.jp/>). It is a smart phone application which resembled the Purikura. Purikura is properly called "print club," and it is a machine which automatically produces stickers from photographs. We analyzed the present system, extracting the problem, and repaired it by mounting an additional function. Moreover, the Android terminal and the website (Ruby on Rails) were set as the development target. Only the PM met with the customer, and when PM directed work to a student, the member improved performance by concentrating only on the development. The target of this system is to increase the number of users, and aimed to public presentation by Android Market (<https://market.android.com/>) and andronavi (<http://andronavi.com/>). The development process combined use-case driven development with the basis of extreme

programming (XP), therefore adopting each practice. We used Eclipse as the development-tool and performed version management with Subversion.

Adaption of PBL using extreme programming

Extreme programming (XP) is a software development methodology which is intended to improve software quality and responsiveness to changing customer requirements [2]. As a type of agile software development, it advocates frequent "releases" in short development cycles (timeboxing), which is intended to improve productivity and introduce checkpoints where new customer requirements can be adopted. XP has been described as having 12 practices, grouped into four areas. We selected the several practices. However, as for the practice in connection with the quality of software, such as pair-programming, refactoring, test-driven development, we judged that it was difficult to apply by a place, time, and human restrictions, and did not adopt.

The result of the project

The repair scale of the software developed in this project can be seen in Table 1 & Table 2. The repair range was an Android application and a website. The additional function of an Android application was a Twitter linkage function. The student B took charge of this function. The repair of the web page performed a layout change of a top page, the addition of a picture search service, and the addition of the Facebook linkage function. In addition, the student C took charge of layout change and a picture search service, and the student A took charge of Facebook linkage. Among these, the Twitter linkage function was not able to be released. Moreover, it was postponed, without the ability to release other functions on schedule. The lack of communication between project manager and students, lack of test cases, and each members' experience with the cause were insufficient.

Table 1
The repair scale of software

Category	Language	Lines of code *read as "after repair (before repair)"
Android	Java	7570 (6902)
Web	PHP	206202 (206044)

Table 2
Additional functions

Additional functions	Contents
Twitter linkage	The function posts message to Twitter from KATAZO of Android Application
Layout change	The picture displayed on a website is increased
Picture search	Keywords are inputted and pictures are searched
Facebook linkage	A website and Facebook are made to cooperate and the contribution from a website to Facebook is enabled

Management result

In this project, the schedule was managed by Work Breakdown Structure (WBS). Work instructions were performed in response to the weekly progress reports sent by students via e-mail, followed by weekly meetings. We used Earned Value Management (EVM) which is one of the management techniques. We managed the progress of project activities. Table 3 is the management data for four months. Although the actual working hours of PM were exceeding 148 hours and student A has also spent many hours comparatively, student B and student C had little time to apply to this project.

Table 3
Management data (unit: time)

	Planned Value	Actual Cost	Earned Value
Project Manager	127.0	148.0	120.5
Student A	127.5	123.5	99.0
Student B	74.5	40.0	34.5
Student C	72.5	55.5	57.0

THE PROPOSAL OF PROBLEM SOLVING

We improved the environment which specialized the education of agile development, and created solutions to the problem which occurred in the fiscal year 2010. The effect brought visualization to the sharing of information, increase in communication, improvement in motivation, and progress, etc. by adopting a ticket-driven development in this project.

Educational Environment

In the project in the fiscal year 2011, the professionals of agile development joined the members of instruction. They performed a lecture on agile development, and they taught how to use BTS (Bug Tracking System) at the beginning of class. Furthermore, the IT engineer, working members of society, joined the members of each project as Project Leader (PL). They performed technical support for each student.

Application of Ticket-driven development (TiDD)

Ticket-driven development is a development style which manages by dividing work into tasks and assigning tickets from BTS (Bug Tracking System). To initiate, a ticket is first published. By using a ticket disagreement of communications can be reduced. By adopting ticket-driven development, information is sharable by the persons concerned, and the role for every worker in charge is also clear, and we can equalize work load, and we can understand who is in what kind of situation.

Problems

The work load allotment begins with the PL's registering tasks into BTS (Bug Tracking System). Although the BTS is a slightly difficult system to learn, once it understands the

process was considerably easier. At the start of this project, the agile coach instructed the members on how to use the BTS. Therefore the ticket management could well-received. The PM and students were able to understand their tasks using ticket.

Requirements for the project

In the project in the fiscal year 2011, the creation of a program, completion, and delivery of goods which realized "a next-generation mobile network service" was carried out. We are creating a program that uses Robot Service Network Protocol (RSNP) that is robot service-oriented protocol specification [3]. As of November, 2011, this project is currently in progress.

RELATED WORK

With the possibility of adapting agile software development methodology, many opportunities arise for the proposal of teaching methodologies in college-level programming courses. According to [4], using extreme programming (XP) could improve success rates in introductory programming courses by adapting approaches. According to [5], the use of agile development methodologies to create small programs in introductory programming courses can bring many benefits to students. In Kanazawa Institute of Technology of Japan, the experience which used ticket-driven development (TiDD) and the Agile Software development together, and performed software engineering education has been reported to [6] which used the agile software development for the study of software engineering. Moreover, [7] practiced the collaborative project-based learning to develop a 3D fighting game in cooperation with Konan University and Osaka University of Arts. Project activities were carried out by using computer game work for software engineering education, taking charge of each special field of study, and mutual cooperation. The experience which makes a computer game that is the subject matter of PBL, raises a student's motivation, and makes software engineering education available to others. An agile development process is applied to a game work project, and software engineering can be studied in the University of Texas at Dallas [8]. By the PBL activity in the programming course of the Federal University of Goias (UFG) in Brazil, Extreme programming was able to introduce PBL activity and the student's capability was successfully raised. Moreover, the application called TaskBoard was developed and the student's group activity was supported [9]. The difference among these related researches premised on the project undertaken for performing agile development at one place in the existing research. However, it is difficult to practice the education by agile development in a distributed environment. In this paper, the enforcement method of the project corresponding to distributed type PBL proposed.

CONCLUSIONS

In this paper, by the distributed type software development PBL, one of the authors (Kizaki) participated in one project as a project manager in the fiscal year 2010. However, this project had problems in task management and communications within the team. We performed two approaches as corrective method. The 1st, we proposed to focus on the real-time attribute of collaborative work under what we call the ticket-driven development. The 2nd, we prepared the educational environment which specialized in agile development. In the fiscal year 2011, specialists of agile development participated from actual companies. Moreover, one engineer, a working member of IT society, was assigned to each team, and performed technical support. By using this technique, the task management and communication within a team improved.

REFERENCES

- [1] Yoshiaki Matsuzawa & Hajime Ohiwa (2007). Learning Information Systems Engineering and Its Management from Experience of a Tiny Project through University-Industry Collaboration. *International Conference on Advanced Learning Technologies 2007 (ICALT)*, 538 - 542.
- [2] Kent Beck & B, Boehm. (2003). Agility through discipline A debate. *IEEE Computer*, 44 - 46.
- [3] Soichiro Ushio, Yuka Ito, Kazunori Okada, Tomoki Kitahara, Hidenori Tsuji, Satoko Moriguchi, Msahiko Narita & Yuka Kato (2011). The Digital Travel Diary System Using the Network Service Platform. 2011 *Workshops of International Conference on Advanced Information Networking and Applications (WAINA)*, 890 - 895.
- [4] Jovanovic, V., Murphy, T. & Greca, A.(2002). Use of extreme programming (XP) in teaching introductory programming. 32th *Frontiers in Education Conference (FIE)*, FIG-23 vol.2.
- [5] Karen Keefe, Judithe Sheard & Martin Dick(2006). Adopting XP practices for teaching object oriented programming. *8th Australasian Conference on Computing Education (ACE)*, 91 - 100.
- [6] Kazuhiro Matsuo & Shota Anzawa(2010). Work in progress — Project practices of agile software development for undergraduate students. 40th *Frontiers in Education Conference (FIE)*, S2D-1 - S2D-2.
- [7] Naoya Nitta, Yasuhiro Takemura & Izuru Kume(2009). A practice of collaborative project-based learning for mutual edification between programming skill and artistic craftsmanship. 39th *Frontiers in Education Conference (FIE)*, 1 - 5.
- [8] Tucker Smith, Kendra M.L., Cooper, C. & Shaun Longstreet .(2011). Software Engineering Senior Design Course: Experiences with Agile Game Development in a Capstone Project. 1th *International Workshop on Games and Software Engineering*, 9 - 12.
- [9] H.W.A.S. Gondim, A.P.L. Ambrósio & F.M. Costa(2011). TaskBoard - Using XP to Implement Problem-Based Learning in an Introductory Programming Course. 12th *International Conference on Agile Software Development (XP2011)*, 162 - 175.