A Software Development Group Exercise Support Environment, EtUDE:  
The System Overview and the System Evaluation through Applying to Classes

HIROAKI HASHIURA†, KOTARO YAMASHITA†,  
TATSUYA ISHIKAWA†, YUKA ISOZAKI**, SEIICHI KOMIYA†  
Graduate School of Engineering†  
Department of Information Science and Engineering*  
Shibaura Institute of Technology  
Komiya Lab, 13F, Kenkyu-to, 3-7-5 Toyosu Koto-ku, Tokyo  
JAPAN

Abstract: In the recent years, software development has become larger in scale and more complicated. Furthermore, development with faster delivery and lower cost is required, thus the software development environment is becoming more and more complex. Accordingly, companies seek high potential students in universities who have a practical sense. The information engineering department of Shibaura Institute of Technology provides classes that adopt a more practical approach to software development so that students can obtain knowledge and skills necessary for software development. However, as class hours assigned for learning software engineering are not sufficient, a support system that enables students to practice outside the classroom or school is required. This support system should have work such that it enables each member belonging to the same group to collectively perform tasks just like in the classroom. In order to solve this problem, the authors developed EtUDE [1][2], the group exercise support environment for software development. Although group exercise helps to reduce the burden of instructing compared to individual exercise, it is difficult to offer instructions that meet the individual’s needs. However, it is the goal of the exercise-based classes to accomplish the obtainment of knowledge and skills for software development according to each student’s level. Therefore, the authors developed the group exercise support environment for software development, EtUDE which features various functions necessary for group exercise support as well as the function that detects learners who do not benefit from the group exercise and need individual instruction. With this, the software development exercise in more practical form will be available, and at the same time, it is made possible to acquire the knowledge and skills necessary for software development according to each student’s level. This essay presents the overview of EtUDE system and the outcome of the application of the system.

Key-Words: Software Development Group Exercises, Software Development Environment

1 Introduction

In recent years, software development in the actual world has become larger in scale and more complicated due to highly sophisticated technology and diversified system platforms. Furthermore, development with faster delivery and lower cost is required, thus the software development environment is becoming more and more complex. Because of these drivers it is becoming more important to develop manpower with more practical knowledge and skills necessary for software development. However, although companies in the IT industry fully acknowledge that it is imperative to nurture such manpower, their daily work loads force them to assign practical tasks to new workers who have not spent enough time in training or they ask contract companies for temporary engineers for a software development projects. The same situation occurs to the contract companies in that those temporary engineers are not trained enough and in some cases even the temporary engineers cannot make up for manpower shortage and more engineers have to be provided by sub-subsidiaries. In reality, software engineers learn skills through experience, but not through in-company training. Therefore, due to these problems, companies seek such students that have a practical sense and high potential.

In the real world, however, it is common to build a project team for software development. For this reason, universities and other institutes often have
students experience practical software development through quasi projects of software development and exercises worked on by several groups. The students can learn practical skills for software development by voluntarily solving various problems using the software engineering knowledge they obtain from lectures.

The Information science and engineering department of Shibaura Institute of Technology (the “Institute”) has carried out the practical software development exercise (the “exercise”) which is performed by several project teams made up of juniors to learn about the knowledge and skills for software development since 2003. In the exercise, to do assignments, each member of the same group shares various responsibilities for developing software. As such, strictly speaking, it is not group exercise but rather team exercise.

We propose improving the exercise quality by introducing EtUDE (Environment for Ultimate software Development Exercise), the software development group exercise support system which we developed to solve problems of software engineering education at Institute. The function of EtUDE is composed of the following four elements: “communication support,” “deliverable management support,” “project management support,” and “software development support.” In addition, to monitor the students’ activities, it also provides a log acquiring function to automatically collect the data showing the students’ activity status without troubling the students.

Now, let’s discuss the related work. Unlike regular lectures software development exercise requires frequent asynchronous communication between students and instructors. Also, it is necessary for team members having different schedules to communicate even after class. For these reasons, many exercise support systems have been suggested so far.

(1) TeamSCOPE [3]

Jang and his colleagues of Michigan State University developed a system called TeamSCOPE to conduct an experiment of remote software development by a team composed of members in Asia, America, Europe and all over the world. Jang pointed out that as the software development by remote members lacks awareness, real-time communication and group meeting scheduling is difficult and gaps may be created in deliverables and their analysis. They also suggested that the support for the shortage of awareness is important.

(2) Waltz [4]

Chiken and his colleagues pointed out that not only communication between students but also communication between students and instructors is important in software development, and developed a system called Waltz that features a communication support function not only for in-group communication but also communication between students and instructors. In the exercise using Waltz, the instructors conduct the inspection and the acceptance tests to evaluate students’ progress or performance at a checking point, and Waltz features such a function that supports these tasks. Chiken and his colleagues made it clear that the communication support function for students and instructors is of help in such tasks as inspection and tests.

(3) Research by Matsuura and His Colleagues [5][6]

Matsuura and his colleagues suggested that the software development process in the exercise is composed of the demand analysis phase, the system analysis phase, the system design phase, and the implementation and test phase, and that it is important to let students learn to consider the purposes of these phases and develop software step-by-step while producing intermediate deliverables. They aimed to accomplish the above goal by adding a reporting template to the system.

The following is the contents of this essay. Chapter 2 discusses the problems in university software engineering education. Chapter 3 figures out the required functions of EtUDE that were developed to solve the problems described in Chapter 2, and explains the functions provided. Chapter 4 discusses the architecture to realize these functions. In Chapter 5, the EtUDE system is applied to the actual exercise, and the effectiveness of each function provided and the actual exercise with EtUDE is evaluated. Chapter 6 presents the conclusion of this essay.

2 Problems in Institute Software Engineering Education

Software engineering is a constrained subject in the sense that it is based on the assumption that learners have already obtained programming knowledge and skills and are able to utilize them to some degree. In junior high schools, “information and computer” were made a requirement in home economics education in 2002, and in high school, “information” was made a requirement in 2003. The students who take such information classes entered Institute after 2006. However, these students only
received information literacy education like the operations of applications such as word processing and spreadsheet, but do not learn programming. As a result, even information engineering students in Institute have to start learning programming before learning software engineering. Programming skills mean to understand and memorize the programming language grammar and learn how to materialize various functions using the language. The total hours spent in information engineering classes are not sufficient for this, and it is not unusual that there are students who have good grades but are not good at programming. However, even those students not good at programming would be able to write a program to some degree after spending some time programming on daily basis at work. This fact shows that mastering programming takes time and anyone would be able to write a program after spending time.

It is not easy to make even information engineering students master the programming skills and software engineering in four years at Institute. As the information engineering students have so many requirements they do not have enough time for classes to learn about programming skills and software engineering. Therefore, to make up for the shortage of class hours, such a support system would be able to write a program to some degree after spending some time programming on daily basis at work. This fact shows that mastering programming takes time and anyone would be able to write a program after spending time.

3 Group Exercise Support Function for Software Development

We built EtUDE as a Web application system such that each student can use without restriction of time or place. In addition, EtUDE is made to provide “communication support system” so that the members of the same group can do the work collectively even if they are remote or use the system at different time. Also, EtUDE is provided with “deliverable management support function” and “project management support function” so that several members can collectively produce one deliverable even if they do the work separately. Meanwhile, the “software development support” function only offers the functions that are necessary for an individual to develop software, and it is not different from general software development support tools at all. Therefore, this essay does not cover the description of that.

3.1 Communication Support Function

Chiken and his colleagues figured out that after the observation of the exercise, the communication occurring during the exercise is divided into the following three types according to the purpose and also based on between whom the communication would take place.

(1) Question and answer
Takes place to make up for technically unclear points. Further divided into in-group communication, communication between an instructor and each student, and communication between an instructor and the group.

(2) Suggestion and discussion
Communication taking place within the group.

(3) Notification from instructor and confirmation
Communication between an instructor and an individual student or the group.

We set up the bulletin board function to support the above three communication patterns. There are two types of bulletin board on this system: one for each group and one for question and memorandum for all learners. Both are the thread-type bulletin boards with tree view. The thread-type bulletin board has a merit that comments can be made by the thread, the related subject. Also, the tree view makes it possible to grasp to which comment the reply is made. Fig. 1 shows the screen of the bulletin board function. As in Fig. 1, when you select the comment you want to see, all the comments in the thread are displayed and the selected comment is highlighted to make it easy to use.

The group bulletin board shows the deliverables uploaded in the “deliverable management function” which is described later.

The bulletin board function also has the feature of sending notice e-mail to participants as soon as a message is posted. This is the function designed for awareness.
3.2 Deliverable Management Support Function

In software development, various kinds of tasks are done, including development planning, analysis, design, coding, and testing, and as a result, such deliverables as plan documents, requirement specific documents, specifications, test cases, source codes, and minutes are produced. Among these deliverables, there is a (intermediate) deliverable which determines the relation between preceding tasks and following tasks (that is, the process order of these tasks), for example, the following task is performed by referring to the (intermediate) deliverable produced by the preceding task. Not only should such a (intermediate) deliverable should definitely be shared within the group which produces or uses it, but it also should be shared by the instructor who refers to the (intermediate) deliverable to grasp the progress of the exercise and give proper advice to the students. Furthermore, in cases where there are many groups, it is a quite burden for the instructor to receive all deliverables from the students and manage them for progress reviewing and evaluating. Therefore, such a system that electronically consolidates all the deliverables to which both the students and the instructor have access at any time from any where is necessary. As a result, the three functions, “deliverable management function,” “module management function,” and “source code function,” were developed.

(1) Deliverable Management Function
Deliverable management function uploads and manages document deliverables other than source code.

(2) Module Management Function
Source code is a deliverable that is very difficult to manage as it consists of many files compared to other deliverables and is modified quite often. Therefore, the module management function was developed to make it easier to manage source code. Module in this sense refers to Java packages and source code, and it is represented in tree structure.

(3) Source Code Management Function
“Source code management function” was developed for the version management and consolidation of the source code. Source code management function uploads the source code of the module produced by the module management function.

3.3 Project Management Support Function

“Report preparation support function” and “failure management function” were developed for project management support.

(1) Report Preparation Support Function
The report preparation support function provides templates with the items to be reported to help the students who are not sure what to report to the instructor. The students can fill in information according to the template so that they can provide the information the instructor needs. This also makes it possible to consolidate the information. There are five kinds of reports: development plan, work item management, work report, minutes, and development completion report. Work item management is a report that helps to understand the progress of work items, and the most important in managing a project.

(2) Failure Management Function
The failure management function (bug tracking function) manages bugs in coding in the implementation phase and the test phase. Bugs will occur most of the time in a proper test after coding. To deal with this problem, it is fundamental to create a list of bugs occurred at the time and debug according to the priority. Creating the bug list helps to clarify the role-sharing of debugging as well as manage the progress of debugging.

4 Architecture of System Building

4.1 Basic Policy of System Building
EtUDE has been enhanced by applying it to software development exercise classes every year and analyzing and evaluating the application result. We wish to firmly maintain the policy. Therefore, EtUDE is built up on the premise of addition and modification of functions. To make such a system structure possible, we set up the following five goals.

(1) To build highly reusable system structure.
(2) To make the system structure maintenance easy.
(3) To make the system structure suitable for development by multiple persons.
(4) To make the system structure easy to test.
(5) To make it easy to collect log information from the system.

In order to accomplish the above goals, we set it up as the basic policy of system building to reduce the interdependence of the system components to be reused to enhance the independence of the components.

The basic policy above
(1) Implementation with POJO
The implementation with POJO (Plain old Java Objects) refers to implementing using the components of POJO. This will lower the interdependence of components and enhance the independence of them.

(2) Implementation using interface (polymorphism)
The merits of operating the objects with only the interface defined by abstract class are as follows:

- Clients can implement without knowing the object type as long as they follow the interface required by the object to use.
- Clients can implement without knowing the class that implements these objects.

As the implementation using interface makes it easier to produce a mock, testing the program and developing the system by multiple persons is made easier.

(3) Aspect Oriented Programming [7]
By applying Aspect Oriented Programming, those functions that were commonly contained in several modules are separated from those modules and put into Aspect module, making it possible to develop each module without disturbance of the common functions.

4.2 System Architecture
As we have explained so far, we built EtUDE as a Web application system so that learners can use the system without restriction in time or place. Also, we applied Java as a system development language which provides great performance for building a Web application system. Java has an affluent class library, mass production component technology like JavaBeans, various tools to facilitate system operation such as filters and custom tags, and various frameworks (i.e., JavaEE [8]) for building server application program on a large scale system. In addition, it provides full of know-how on software design, such as design pattern and JavaEE pattern. These are the main reasons why we used Java.

Sun Microsystems, which developed Java, formulated Java BluePrints or a guideline called “Java Platform, Enterprise Edition application design guide”[9] for designing applications using JavaEE. Java Blueprints introduces multilayer system building model, and recommends multilayer architecture. Therefore, our system also use the multilayer architecture, and we decided to develop the system with 5-layer architecture as in Fig. 2.

(1) Client layer
The client layer receives the input and output from users on the client side or displays performance consequence of logic. It is equivalent to a Web browser on the Web system.

(2) Presentation layer
The presentation layer receives the input from users on the server side and transfers the content to the business layer to request the execution of business logic. Then, it returns the performance consequence to the client. It is equivalent to Servlet or JSP on the Web system. Our system uses JSP Model2 architecture on this layer. This clearly separates HTML from logic. The Filter, which performs preprocessing and post-processing of the request, converts character codes.

(3) Business layer
The business layer executes the business logic according to the information the user inputs on the presentation layer, and make access to the integration layer, if necessary. The result is transferred to the presentation layer. In executing the business logic, it calls for the processing of the business logic through Façade (the window).

(4) Integration layer
The integration layer manages the communication with external resources, such as database, CRM system, and legacy application. It makes access to the database through DAO (Data Access Object).

(5) Resource layer
The resource layer refers to database, resource, external service, and B2B.
5 System Effectiveness Evaluation Based on the Application to the Class

We applied EtUDE to the exercise class for software development to evaluate the effectiveness of each support function as well as the exercise class with EtUDE. We made use of the log data regarding the use of EtUDE and the result of the questionnaire after the completion of the exercise for analysis and evaluation. We also analyzed and evaluated each deliverable by metrics. The exercise class is targeted to analyze and design Web application system for “conference room reservation system” using OOSE (Object-Oriented Software Engineering) [10] and develop a program using Java. JUDE [11], a system design support system, is to be used in analysis and evaluation process by OOSE, and Eclipse, programming development environment, in programming process.

5.1 Evaluation of Communication Support Function

We consider the evaluation of effectiveness regarding the bulletin board function to be the evaluation of effectiveness of communication support. We used a 1-to-4 scale (4 being the best) for the evaluation of the bulletin board function based on the questionnaire regarding “operating efficiency” and “usability.” Besides, as no one expressed complaint in the comment space on the questionnaire, it is considered that it served sufficiently well as communication support.

5.2 Evaluation of Deliverable Management Support Function

We consider the evaluation of effectiveness of both the deliverable management function and source code management function to be the evaluation of effectiveness regarding deliverable management support. The result shows, as in Fig. 4 and 5, that more than 80 % of the students gave the excellent rating for both functions regarding operating efficiency. Besides, the analysis of the log result shows that these functions were used most frequently during the winter vacation when the students could hardly work together.
5.3 Evaluation of Project Management Function

We consider the evaluation of effectiveness regarding the progress management function to be the evaluation of effectiveness of the project management support. The result shows, as in Fig. 5, that the students did not give as good rating to the progress management function as other functions. This is because many students are not familiar with the task of managing the progress. Also, the result shown in Table 2 suggests that more than half of the students responded that they could not accomplish the work as planned to the question, “Could you accomplish the work as planned?” The result could contribute to making the function less valued.

However, it is necessary to have the students understand the importance of progress management through making a plan and being engaged in progress management. Therefore, it is necessary to have them experience progress management.

In addition, for the progress management function, we had the students evaluate the progress status of their project using weather descriptions once a week so that the instructor could grasp the progress of each project. The weather rating is five-scale from “sunny,” “fine,” “cloudy,” “rain,” and “thunder storm,” from 5 to 1 respectively. Table 2 shows the average of ratings of the progress of each project. As you can see, the ratings became worse week after week. The same result is observed in most projects. Analyzing the reason of such a result, we found out that the students gave the worse ratings when they could not have meetings with others because the class was canceled or it took them more time for programming as their programming skills were not sufficient. Having the students report the progress of their projects by way of the weather scaling, the instructor could easily understand the progress of the projects.

It is found out that the progress management function above is sufficiently effective support for the instructor.

Table 1 Accuracy of actual performance of plan

<table>
<thead>
<tr>
<th>Response</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accomplished as planned</td>
<td>80 (1%)</td>
</tr>
<tr>
<td>Accomplished as planned to some degree</td>
<td>50 (34%)</td>
</tr>
<tr>
<td>Did not go well as planned</td>
<td>20 (55%)</td>
</tr>
<tr>
<td>Planning was of no use</td>
<td>0 (8%)</td>
</tr>
<tr>
<td>Did not make a plan</td>
<td>0 (2%)</td>
</tr>
</tbody>
</table>

Table 2 Average value of the progress status in the whole project

<table>
<thead>
<tr>
<th>Date</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>11/16</td>
<td>3.4</td>
</tr>
<tr>
<td>11/30</td>
<td>3.1</td>
</tr>
<tr>
<td>12/7</td>
<td>3.2</td>
</tr>
<tr>
<td>12/14</td>
<td>2.8</td>
</tr>
<tr>
<td>12/21</td>
<td>2.9</td>
</tr>
<tr>
<td>12/28</td>
<td>2.5</td>
</tr>
<tr>
<td>1/4</td>
<td>2.6</td>
</tr>
</tbody>
</table>

5.4 Evaluation of the Exercise with EtUDE

We evaluated the effectiveness of the EtUDE applied exercise based on the question, “Did you feel the exercise was fulfilled?” As Table 3 shows that more than 80% of the students responded that it was “very fulfilled” or “fulfilled,” it is obvious that the exercise was very meaningful to the students. We are assured of the effectiveness of the exercise using EtUDE.

Table 3 Evaluation of fulfillment of the exercise

<table>
<thead>
<tr>
<th>Response</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very fulfilled</td>
<td>7 (13%)</td>
</tr>
<tr>
<td>Fulfilled</td>
<td>36 (68%)</td>
</tr>
<tr>
<td>Not so fulfilled</td>
<td>9 (17%)</td>
</tr>
<tr>
<td>Not fulfilled</td>
<td>1 (2%)</td>
</tr>
</tbody>
</table>

At the same time, we found new issues to discuss by applying EtUDE to the exercise.

5.4.1 Application of Communication Support

Table 4 shows the question regarding what communication tools the students used other than the EtUDE bulletin board and the responses to it. The responses are “mail,” “messenger,” and even “mix.”
which is not included in the answer choices. Considering that the efficiency of the bulletin board is not rated so well, it is assumed that the students prefer the communication tool which they are familiar with. As the instructors are required to use EtUDE in order to properly understand the progress of the project, it is necessary to provide the communication tools which are easier to use for the students.

Table 4 Communication tools on EtUDE other than bulletin board (Multiple answers allowed)

<table>
<thead>
<tr>
<th>Tool</th>
<th>2006</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mail</td>
<td>13</td>
</tr>
<tr>
<td>Messenger</td>
<td>12</td>
</tr>
<tr>
<td>Telephone</td>
<td>5</td>
</tr>
<tr>
<td>Other BBS</td>
<td>1</td>
</tr>
<tr>
<td>Conversation</td>
<td>10</td>
</tr>
<tr>
<td>Others</td>
<td>6</td>
</tr>
</tbody>
</table>

5.4.2 Support for Producing Deliverables in Each Process

EtUDE has made it possible to some degree to understand the progress of the exercise, and the instructor has been able to spend more time for teaching the students. This also helped to obtain better ratings towards the exercise. However, human resource is still insufficient, especially the instructor has to spend a lot of time teaching for deliverable production. Therefore, if there is an a function which automatically picks up errors that would often occur in deliverable production, the students can fix the errors by themselves, or if the instructor can guide those students in person, even the group exercise would improve the quality of learning effect.

6 Conclusion

This essay first discussed the current situation of software engineering education in Japanese companies, and stated that Shibaura Institute of Technology practiced the group exercise for software development in order to nurture highly potential students with the practical sense required by companies. Then, we clarified the problems of software engineering education in Japanese universities, for which EtUDE, the support system for software development group exercise, was developed and introduced to the exercise. We presented the overview of EtUDE and the effectiveness of each support function of EtUDE by applying it to the exercise and evaluating it. Furthermore, we showed the effectiveness of the exercise with EtUDE. Finally, we presented the issues that are to be discussed.

References: