

Title	A Proposal of Integrating Personal and Community Support with Learning Environment for Group-based Software Engineering Course
Author(s)	Miura, Masato; Kobayashi, Yusuke; Shimada, Kazuyuki; Takahashi, Koichi; Seiki, Susumu; Hazeyama, Atsuo
Citation	
Issue Date	2007-11
Type	Conference Paper
Text version	publisher
URL	http://hdl.handle.net/10119/4090
Rights	
Description	The original publication is available at JAIST Press http://www.jaist.ac.jp/library/jaist-press/index.html , KICSS 2007 : The Second International Conference on Knowledge, Information and Creativity Support Systems : PROCEEDINGS OF THE CONFERENCE, November 5-7, 2007, [Ishikawa High-Tech Conference Center, Nomi, Ishikawa, JAPAN]

A Proposal of Integrating Personal and Community Support with Learning Environment for Group-based Software Engineering Course

Masato Miura[†] Yusuke Kobayashi[†] Kazuyuki Shimada^{*}
Koichi Takahashi^{*} Susumu Seiki^{*} **Atsuo Hazeyama[†]

[†]Graduate School of Education, Tokyo Gakugei University

^{*}Department of Information Education, Tokyo Gakugei University

**hazeyama@u-gakugei.ac.jp

Abstract

With the advancement of information and communication technologies, the needs for software development increase. Along this trend, training of human resources with high expertise is required. Based on the background, we developed a support system for group-based software engineering project course and applied it to an actual university course. From its application, we found it was difficult to ascertain individual activities and activities beyond the group. This paper proposes a support system for group-based software engineering project course that supports individuals and communities in addition to group activities.

Keywords: Group-based software engineering project course, Learning support system, Community, personal support

1 Introduction

It is recently recognized practical software development education and training is important because of shortage of IT engineers, especially shortage of software engineers with high expertise in Japan [9, 11]. Based on the recognition, we have been tackling group-based software engineering project course since 1997 [6]. In this course, developers who are undergraduate students complete a software development task via a software life cycle starting from requirement analysis, design, implementation, and testing. The goal of the course is that learners experience lots of activities in software development conducted in a real world as follows:

- * they experience the software development process through practice
- * software development is generally performed

by organizing a group by several persons, therefore they are required to experience social processes such as communication, collaboration, problem solving, and/or project management

- * they perform quality assurance activities such as inspection and testing

The support environment is indispensable for this kind of course to be run in an effective and efficient manner, therefore, a lot of support environments have been constructed [2, 4, 7, 8, 10, 12, 13, 14]. These environments support group activities. However in practice, activities by individuals should be ascertained from the viewpoint of group work and instructor. In addition, as information sharing beyond group was observed in the course, we think such activities should be supported.

Software engineering environments that support software development by team are provided. For example, SourceForge.net supports open source software development. It provides the following tools: CVS for source code control, mailing lists, and bug tracking [1]. However, artifacts managed by the tools are not inter-related. As an artifact has relationships with other artifacts and/or communication messages in software development, these relationships need to be managed by the environment.

We propose a learning environment for group-based software engineering project course integrating personal and community support.

The rest of this paper is organized as follows: section 2 describes some requirements for the environment. Section 3 presents a conceptual model for our learning support environment. Section 4 explains the environment. Finally we summarize this paper in section 5.

2 Requirements for a project course support environment

We identified the following three issues from applying our current environment to an actual project course.

2.1 Requirement to development support of individual

In software development, not only activities by group but also lots of activities by individual exist, for example, design document creation, implementation, testing, bug fixing, and so on for assigned usecases. However, traditional support environments mainly support activities by group such as project management, progress monitoring, and artifacts management [2, 4, 7, 8, 10, 12, 13, 14]. Activities by individual were not fully supported, therefore, the following issues occurred.

- * When a learner was beyond the schedule, it is difficult to grasp his/her situation.

- * Responsibility of individuals was unclear.

Therefore, for example, when a defect was detected, it was difficult to determine who was in charge of it.

In order to deal with these problems, the environment needs to support development activities of individuals.

2.2 Requirement to progress monitoring support for the instructor

It is difficult to ascertain activities of individuals in traditional environments. It is therefore difficult to give feedbacks to learners and evaluate their activities.

Abilities and/or motivations that participants have vary in this course. Some learners have difficulty to catch up the project. Some slackers may appear [5]. The teaching staff therefore needs to pay attention to learners' activities. To determine the grade of learners and to validate the instructional method, the instructor needs to evaluate activities of learners. Hayes et al. proposed five methods with respect to grading in group-based software engineering project course [5]. However they supposed to determine learners' grade by means of some sort of indirect data. We think the data that represent personal activities in practice should also be used to determine it. The environment should support that purpose.

2.3 Requirement to support for informal collaboration

We observed in practice of the project course that some learners conducted informal collaboration across group. A learner asked his/her friend for giving advice for the problem (s)he faced. Some other learners read design documents and/or source programs their friends created and imitated them. Traditional environments did not support such kind of activities across group. From the results, the following problems occurred:

- * many learners encountered similar problems and/or defects and such problems and/or defects caused delay

- * the teaching staff were asked similar questions from learners several times

In order to solve these problems, the environment should support the abovementioned informal collaboration. On the other hand, from the viewpoint of learning effectiveness, it is not desirable to give learners mere answer, because they do not learn a good learning style such as investigation. Therefore the environment should support informal collaboration over taking learning effectiveness into consideration.

3 Conceptual model for the support environment

We propose a conceptual model for the learning environment as shown in Figure 1 to solve the abovementioned requirements. We classify the activities in group-based software engineering project course into those by group, those by individual, and those beyond group. Each learner, group, and community is assigned its area for its activity. In the group area, the learners conduct group activities such as project management, progress monitoring, artifacts management, peer review, and so on. In the personal area, learners perform their personal activities such as creation of personal schedule, progress reporting, and artifacts management. In the community area, they make informal communication across group.

Each area is related across the area. The links between the group area and the personal area, and between the community area and the personal area represent affiliation relationship. The links that connect the personal areas represent cooperation relationship.

4 The support environment

This section describes our learning support environment, which embodies the conceptual model we proposed in the previous section. In section 4.1, we describe overview of the functions the environment provides according to the model elements. In the subsequent subsections, we explain the functions in more detail.

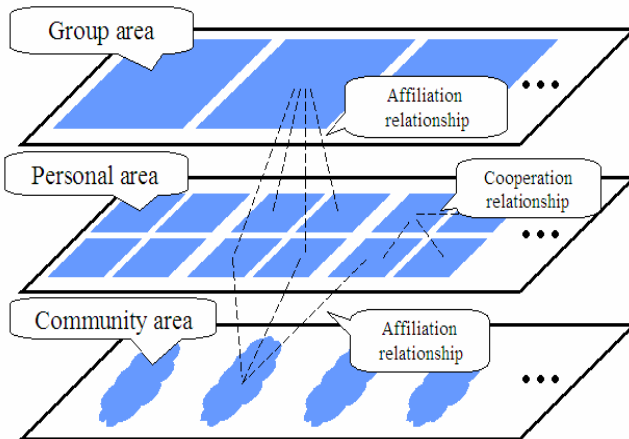


Figure 1: Conceptual model for the support environment

4.1 Overview of the environment

Figure 2 shows a set of functions our environment provides.

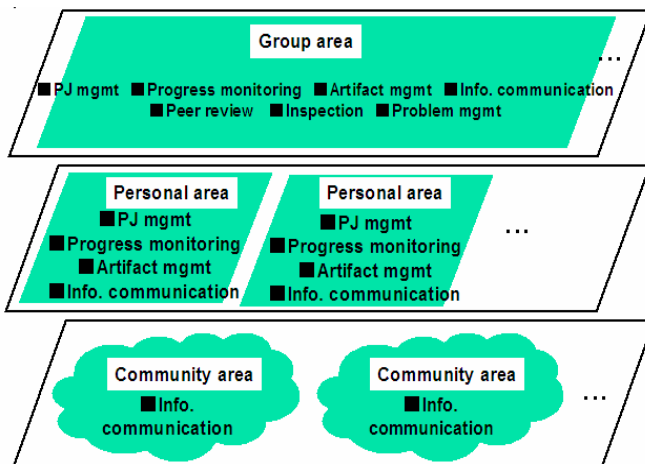


Figure 2: Set of functions of the support environment

* Group area

This area supports development activities by

group comprehensively like traditional learning support environments. Those are project management that defines a group's development schedule and work assignment, progress monitoring, management of artifacts that should be shared within a group, communication function that exchanges messages among group members and/or between the teaching staff and group members, functions that support quality assurance activities, such as peer review and inspection, problem reports management. These functions are inter-related centered around artifact management. Communications in software development often relate to artifacts. Therefore users can access to artifacts from the communication function (asynchronous communication support by BBS). The target artifacts for inspection are those under artifact management.

* Personal area

This area supports personal development activities that are conducted based on the group schedule and assignment. We provide three functions: project management, which specifies personal schedule, personal progress monitoring based on the personal schedule, and artifacts management that manage the artifacts created by individuals.

* Community area

This area supports information sharing across group. For that purpose, some communication functions are provided.

4.2 Development support

This environment supports group and personal activities by providing the group and personal area respectively. Figure 3 shows the data that are managed in each area and those across the areas. Figure 4 shows a screen shot of the personal area.

In the group area, members decide the development plan and work assignment using the project management function. The assignment is reflected on project management function of personal area. Each learner makes his/her personal schedule based on group development plan (Figure 4 (1)). When a personal schedule is registered, it is notified of other group members (Figure 4(4)). That facilitates progress monitoring.

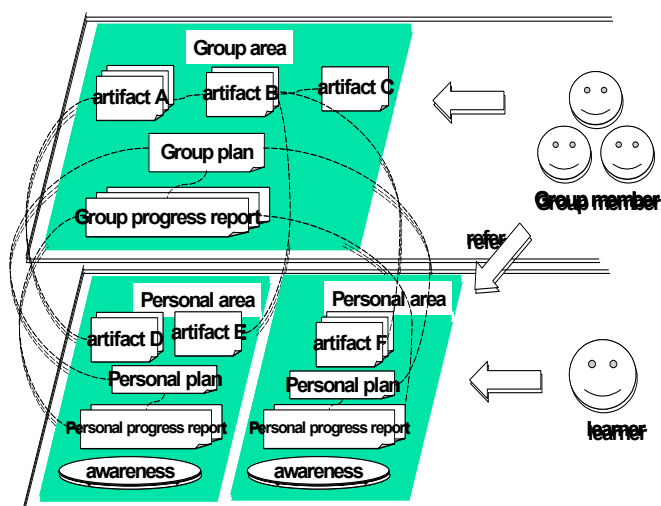


Figure 3: Support of development activities

The progress data of personal and group are managed in the personal area and group area respectively (Figure 4(2)). The related data can be followed by links. Managing the progress data in each area allows the progress data to be arranged. We can ascertain the progress on how a person participated in the group and the personal progress by traversing the group area and personal area. When the progress data of group or personal is registered, it is notified of the personal area of the group members (Figure 4(4)).

Outputs of development activities are managed as artifacts. This environment manages artifacts created by group in the group area and those created by personal in the personal area (Figure 4(3)). At this time, related artifacts are managed by links like progress monitoring. Once artifacts are registered, it is notified of the personal area of the group members (Figure 4(4)).

Our environment supports project management, progress monitoring, and artifact management for not only groups but also personals. Personal areas are related with group area by affiliation relationship. By this relationship, when a member is beyond the schedule, the group members go to the personal area for the member and can ascertain the detailed progress. Since our environment manages development activities in each area, responsibility for activities becomes clear. For example, defects are detected in an artifact by peer review and/or inspection, the member who is in charge of it is specified. These features solve the requirements that were described in section 2.1.

4.3 Progress monitoring support by the instructor

In our environment, a personal area manages a personal plan, personal progress report, and artifacts for a learner. The instructor can access to all of them from the instructor's area.

When quality problem for artifacts or delay is found, the instructor can visit to the personal area for the learner and ascertain his/her plan, progress and the artifacts (s)he created (Figure 4(1,2, and 3)). According to such observation, the instructor gives his/her advice or feedback if necessary. Once the instructor sends the learner a message for advice or feedback, it is notified of him/her (Figure 4(6)).

After the project course, the instructor evaluates the artifacts that are stored in the group area and personal area. Evaluation for group is determined by the artifacts in the group area. Evaluation for personal is determined by the artifacts in the personal area. This function enables the teaching staff to give feedback and/or to evaluate learners' activities, therefore, solve the requirement raised in section 2.2.

4.4 Information sharing support across group

Our environment provides two types of information sharing support, that is, by community from the viewpoint of problem solving, and by cooperation relationship among learners from the viewpoint of quality according to the requirements in section 2.3.

4.4.1 Information sharing support by community

Our environment supports to create a community area and communicate with peers in the area. This function enables to participate in a community a learner has interests and to share information. As such, our environment facilitates collaborative information sharing among learners.

The abovementioned community may not fully support those who have some problems because social relationships in communities are loose and various types of communications are conducted, therefore turn around time for responses to a request is not always short.



Figure 4: Screen image of the personal area

We therefore provide a problem solving community that supports specific problem solving in addition to a community of general purpose. We explain problem solving process using this community.

(i) Initial phase (Figure 5)

In this phase, a user creates a community, and calls for participation.

- 1) Creation of a problem solving community: a learner who encountered a problem creates a community to get advice from the whole class.
- 2) Calling for participation: the environment

informs the whole class of a notification that a learner encountered a problem. The notification message is shown in Figure 4(5).

- 3) Registration of participation for the problem solving community: the learners who had interests in the problem register their participation in the community. By the registration, affiliation relationship is established between them and the community. When the relationship is established, an icon to access the community will be shown in Figure 4(8).

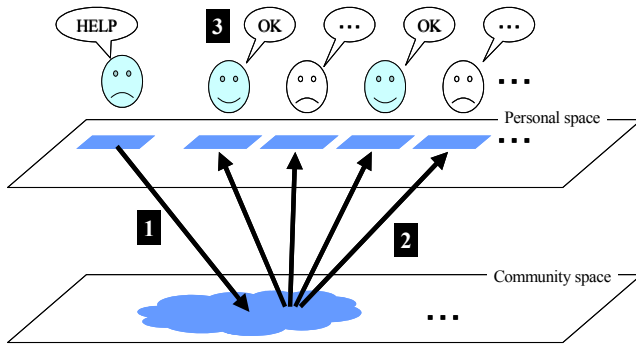


Figure 5: Initial phase

(ii) Activation process (Figure 6)

This is a phase participants in the community discuss to solve the problem.

- 4) Discussion: discussions on the problem are held centered around the person who asked for help. To facilitate discussions, awareness information is provided as shown in Figure 4(5) when messages are posted to the community area. When the discussions are at a standstill, a function that asks the teaching staff for help is provided.

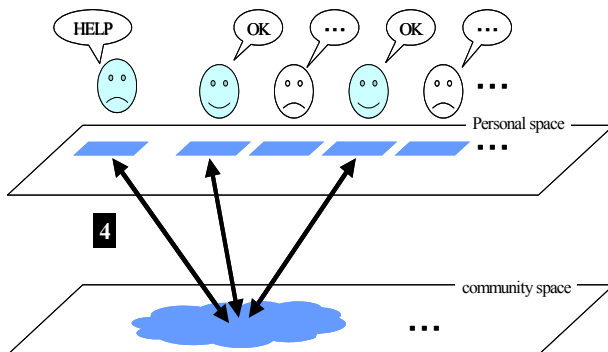


Figure 6: Activation phase

(iii) Convergence phase (Figure 7)

This is a phase to terminate discussions in the corresponding community after the problem has fixed and a learner who asked for help registers the problem resolution information.

- 5) Registration of the problem resolution information: the learner who asked for help registers the problem resolution information, i.e., cause and disposition of the problem, and the lessons learned from the problem resolution, after (s)he resolved it.
- 6) Notification of the problem resolution: the learners who possess the affiliation relation-

ship with the community receive a message that the problem has fixed. Learners who have a similar problem to this can utilize this community, i.e., they can refer stored information or pose questions to those who participate in the community.

By providing the abovementioned two types community area, our environment supports both loose information sharing and information sharing for specific problems. They facilitate communications across groups, therefore, we expect to improve efficiency of learners' development. By sharing the problem resolution information for specific problems in the community, the teaching staff can avoid to have to answer duplicate questions. Thus by supporting informal communication with community, we expect to solve the requirements in section 2.3.

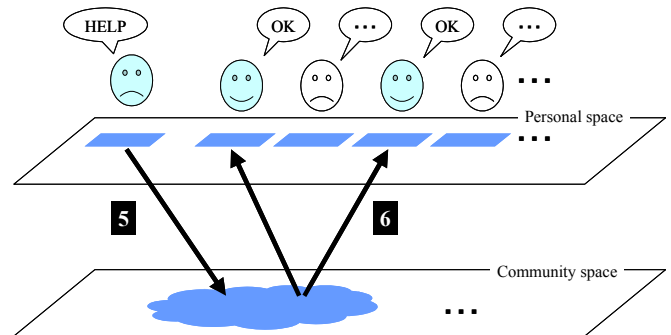


Figure 7: Convergence phase

4.4.2 Information sharing support based on cooperation relationship

We observed some learners referred to artifacts their friends created. A learner X implemented a function that makes a user register "book" data from a web browser, uploads the data to a server, and then stores them into a database. Another learner Y, who is a friend of learner X, was in charge of a function that registered the user information to a database in a server. Both functions were similar in that the data input via a web browser were uploaded to a server and then stored into a database. Learner Y referred to the source codes learner X created. The learner Y implemented his/her assigned function successfully. By such cooperation, most of them could accomplish their assignment, even if they lack specific technical skills. Such actions are said to be effective in the field of learning theory [3]. Therefore we support information sharing among

individuals in our environment. Our environment opens the artifacts, which are stored in the personal area to those who participated in the course. At this time, we introduce access control, that is, everyone can't always access all artifacts. In this study, we call the relationship that opens the information each other cooperation relationship. The relationship is shown by icons in the personal area like Figure 4(7). By selecting the icon, a learner can access to the personal area of the learner the icon represents. The cooperation relationship is established only among group members from the outset. However, by agreement among learners this relationship can be established at any time. We suppose this relationship will be established by not only friendship in the real world but also by relationship that was constructed in a community.

Thus by exchanging information between learners who established cooperation relationship, overall quality improvement will be expected. As the result, efficiency of development will improve, and the load of the teaching staff with respect to review will reduce. We think the requirement of section 2.3 will be solved by supporting the relationship for informal collaboration.

5. Summary and future work

This paper described the necessity for supporting not only group activities but also personal activities, for monitoring by the teaching staff, and information sharing across group in group-based software engineering project course. We proposed a conceptual model for the learning support environment that includes personal area, group area, and community area. We also described some major functions for the environment.

We will complete its implementation and apply it to an actual course.

Acknowledgment

This study is supported by the Grant-in Aid for No. (C) 18500701 from The Ministry of Education, Science, Sports and Culture of Japan.

References

[1] L. Augustin, D. Bressler, and G. Smith, Ac-

celarating software development through collaboration, Proceedings of the 24th International Conference on Software Engineering (ICSE2002), 559-563, ACM Press, 2002.

[2] K. Chiken, and A. Hazeyama, Awareness Support in Group-based Software Engineering Education System, Proceedings of the Tenth Asia-Pacific Software Engineering Conference (APSEC2003), 280-289, IEEE Computer Society Press, 2003.

[3] A. Collins, J. S. Brown, and S. Newman, Cognitive Apprenticeship: Teaching the Craft of reading, writing, and mathematics. In L. Resnick (ed.), Knowing, Learning, and Instruction: Essays in Honor of Robert Glaser. Hillsdale, NJ: Lawrence Erlbaum, 1989.

[4] S. A. Drummond, and C. Boldyreff, The Development and Trial of SEGWorld : A Virtual Environment for Software Engineering Student Group Work, Proceedings of the IEEE 13th Conference on Software Engineering Education and Training (CSEET2000), 87-97, IEEE Computer Society Press, 2000.

[5] J. H. Hayes, T. C. Lethbridge, and D. Port, Evaluating Individual Contribution Toward Group Software Engineering Projects, Proceedings of the 25th International Conference on Software Engineering, 622-627, IEEE Computer Society Press, 2003.

[6] A. Hazeyama, An Education Class on Design And Implementation of an Information System in a University And Its Evaluation, Proceedings of the 24th Annual International Computer Software and Applications Conference (COMPSAC2000), IEEE Computer Society Press, 21-27, 2000.

[7] A. Hazeyama, K. Osada, Y. Miyadera, and S. Yokoyama, An Education Support System of Information System Design and Implementation, Proceedings of the 7th Asia-Pacific Software Engineering Conference (APSEC2000), IEEE Computer Society Press, 393-396, 2000.

[8] A. Hazeyama and A. Nakano, Communication Support System for Software Engineering Education, *Journal of IPSJ*, Vol. 42, No. 11, pp. 2550-2561, 2001 (In Japanese).

[9] Keidanren (In Japanese), <http://www.keidanren.or.jp/japanese/policy/2005/039/honbun.pdf>.

[10] N. Matsushita, S. Matsuura, and Y. Hayashi, Tool Support for Inspection on Group Work Software Development 3Z-3, Proceedings of the 67th IPSJ Congress, 2005 (In Japanese).

- [11] Ministry of Education, Culture, Sports, Science and Technology, 2006 (In Japanese), http://www.mext.go.jp/a_menu/koutou/it/index.htm.
- [12] S. Shoenig, Supporting a Software Engineering Course with Lotus Note, Proceedings of the International Conference on Software Engineering Education and Practice (SEEP1998), IEEE Computer Society Press, 1998.
- [13] C. Steinfield, C.-Y. Jang, and B. Pfaff, Supporting Virtual Team Collaboration: the TeamSCOPE System, Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work, 81-90, ACM Press, 1999.
- [14] K. Yamashita, H. Hashiura, R. Chiba, Y. Inoue, S. Umebayashi, and S. Komiya, EtUDE: A Software Environment for Supporting Exercise in Software Development Conducted in Units of Groups -, IEICE Technical Report AI2006-62, 9-12, 2007 (In Japanese).