Project Replayer – An Investigation Tool to Revisit Processes of Past Projects

Keita Goto[1], Noriko Hankawa[2], and Hajimu Iida[1]

[1] Graduate School of Information Science, Nara Institute of Science and Technology, Japan
keita-g@is.naist.jp, iida@itc.naist.jp
[2] Faculty of Management Information, Hannan University, Japan
hanakawa@hannan-u.ac.jp

Abstract. In order to help knowledge acquisition and accumulation from past experiences, we propose a KFC (Knowledge Feedback Cycle) framework among engineers and researchers. Three tools (Empirical Project Monitor, Simulator, and Replayer) are used to circulate captured knowledge in KFC. Project Replayer is a most characteristic tool used to review data of past projects derived from development logs; version control, bug reports and e-mails. With Project Replayer, past projects can be easily revisited and complicated phenomena of past projects can be investigated. As a result of preliminary experiments, we have confirmed that Project Replayer helps researchers construct and validate hypotheses of software process. We also confirmed that developers have acquired new knowledge about a certain problem extracted from past projects.

1 Introduction

Recently, software development scale becomes bigger, and software quality's impact to our society is significantly increasing. On the other hand, lifetime of software is getting shorter. In order to develop software with certain qualities in a limited time, developers require various knowledge such as cost estimation or risk management, as well as other software engineering techniques. Some of such knowledge should be extracted and accumulated through their own experiences. However, acquiring and accumulating such knowledge require long time and large efforts. In other words, it is very difficult for developers to become matured engineers in a short period.

In order to help knowledge acquisition and accumulation for novice software engineers, we propose a framework for *cycling* engineering and management knowledge among experienced developers, software engineering researchers and novice developers. We call this cycle KFC (Knowledge Feedback Cycle). In the KFC environment, knowledge, mainly concerning risk management and cost estimation, is extracted from past experiences for future reuse. Three tools, EPM (Empirical Project Monitor), Project Replayer, and Project Simulator, are used to capture and circulate knowledge in KFC. EPM [1] is a tool to automatically collect project data from source code repository, bug-reports and e-mails. Project Replayer is a tool used to review data of past projects. Project Simulator is used to provide actual feedback to developers. Developers can avoid mistakes that is not happened in their experiences but happened in other developers' experiences before.

Q. Wang et al. (Eds.): SPW/ProSim 2006, LNCS 3966, pp. 72 – 79, 2006.

This paper mainly describes the concept of the KFC and features of Project Replayer. Section 2 shows related works. In section 3, we present a conceptual environment of KFC. In section 4, outline of Project Replayer and its feature are explained. In section 5, preliminary experiments to evaluate capability of the Replayer are shown. In section 6, we discuss experiment result. Finally, in section 7, conclusions and further work are shown.

2 Related Work

Recently, many works are published in the field of the software process simulations. Some of them focus to help understanding of process behavior and training of process management using software process simulators [2].

For example, Pfahl et al. present the system with integrated simulation component called CBT that is designed for software engineering education [3]. CBT provides an interactive environment using standard web browsers to learn knowledge of project management. The CBT simulation module employs the model to represent the characteristic of a project is generated by the event diagram based on a COCOMO model. RoleEnact is a tool to support simulation, evaluation and improvement of software development processes [4]. RoleEnact focuses on developers' roles. Once part of the existing process has been captured in the model generator, the model stepper and simulator evaluate the results of running process while RoleEnact revise the process.

Most of these works treat abstracted model to show behaviors of software processes, even though they are obtained from real project experiences. Though real project data is not directly handled by those systems, investigation of real project data is very important to understand the behavior of the project in detail. Real project investigation also plays a major role in construction and validation of process simulation models, especially when we construct organization-specific models based on their own experiences. Usually, project investigation is very time-consuming task, and reducing the cost of investigation task is the key factor of knowledge extraction.

3 Knowledge Feedback Cycle

The purpose of KFC environment is to circulate knowledge from experience of past projects to future projects. Developers are supposed to acquire new knowledge while experiencing software development projects. If such knowledge can be transferred to future projects at low-cost, it is quite valuable and helpful for the members.

To establish such cycle, KFC employs three tools; EPM, Project Replayer and Project Simulator (See Fig. 1). KFC also involves two human roles – software developers and software engineering researchers. Developers utilize the KFC environment in order to acquire new knowledge from past projects while researchers utilize the KFC environment in order to construct simulation models which are embedded to the Project Simulator.

A typical scenario in KFC would be as follows;

Step1: Various development data (records of code modification, bug tracking, and e-mails) is automatically captured by EPM during the project enactment (See "EPM" part of Fig. 1).

Step2: Researchers analyze collected data to construct various simulation models using Project Replayer and analysis tools (See "Researcher" part of Fig. 1).

Step3: Using Project Replayer, developers review past projects. Events and accidents that are not recorded by EPM are also clarified in interview with developers (See "Project Replayer" part of Fig. 1).

Step4: Regarding results of reviews and interviews, researchers refine their simulation models that were made in Step2. The models are embedded into the Project Simulator (See "Simulation Model" part of Fig. 1).

Step5: Using the Project Simulator, novice developers learn complicated phenomena in past projects. Developers can also utilize the Project Simulator to make their next project plans. The planned project is regarded as the target of Step1 of the next cycle (See "Project Simulator" part of Fig. 1).

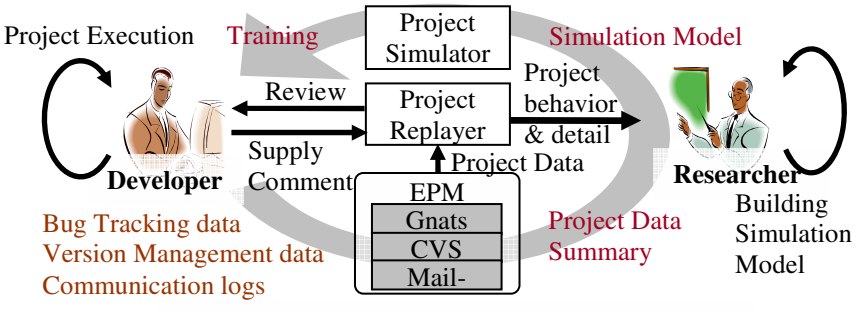

Fig. 1. Knowledge Feedback Cycle environment

The whole mechanism of the KFC environment is currently under development in our group, while a prototype of Project Replayer has been implemented at first to evaluate its capability. Following parts of this article describe the feature of Project Replayer and results of preliminary experiments using the Replayer.

4 Project Replayer

4.1 Purpose of the Tool

Project Replayer is a tool to *replay* project data collected by EPM in order to help understanding behavior of past projects. Project Replayer accelerates knowledge circulation by supporting both of two roles in KFC; developers can use Project Replayer to revisit their past projects for postmortem evaluations, while researchers can use Project Replayer to deeply understand and analyze dynamic behavior of the

Fig. 2. Screen image of Project Replayer

projects. Replaying past real project is also important for education and training because simulators sometimes provide practitioners with quite less reality that is derived from abstract and ideal models. Project Replayer faithfully replays various behaviors of past projects. Developers are more familiar with the behaviors of past projects than behavior of the virtual projects.

4.2 Features

Current implementation of Project Replayer has four views (Event list view, File view, Graph view and Member view) and a time-control bar.

Event list view shows various (CVS: Concurrent Versions System, bugs, and e-mail) events collected through EPM are listed in order of time (See "Event list view" part of Fig. 2). The first column of each line works as a button to jump to the time of the event, which is indicated in the second column. The third column indicates the type of the event, the fourth column shows owner of the event, and the fifth column shows related filename.

File view presents source files in CVS repository (See "File view" part of Fig. 2). Each file item is shown with its name and progress bar. The progress bar shows rate of progress calculated as ratio of current CLoC (Cumulative modified Lines of Code) to the final CLoC. Graph view shows transitions of various data including total LoC (Line of code) and CLoC (See "Graph view" part of Fig. 2). Y-axis of the line chart indicates quantitative value such as LoC or CLoC, while x-axis indicates calendar time (days) of the project.

Member view lists project members with their role names and current actions (See "Member view" part of Fig. 2). The first row of a member item shows the member's

name, the second row shows the member's portrait (or avatar), third row indicates current action s/he has performed, and fourth row shows active files that are currently being modified by the member.

The time control bar indicates the time (date) currently shown in replaying. Moving the slider changes the time currently displayed. The bar also provides buttons such as start and pause.

5 Preliminary Experiments

5.1 Planning of the Experiments

To evaluate features of Project Replayer, we conducted preliminary experiments. The aim of the experiments is to observe how Project Replayer's features help the researcher and developers to make a new simulation model and to acquire new knowledge respectively. Since experiments focused two viewpoints, i.e. developers' benefits, and researchers' benefits, we prepared four subjects; Subject0 is a researcher, and Subject1~3 are developers (graduate school students). Experiments form three phases, researcher's analysis, developers' review, and construction of simulation model by the researcher.

In the first phase, a researcher analyzes project's phenomena using Project Replayer. The researcher is requested to construct simulation models regarding the analyzed phenomena of the project. The researcher may have some questions about phenomena because information exposed by Project Replayer still does not include all detailed events that occurred in the project. Therefore, the researcher draws some questions about specific phenomena. The developers' answer to the questions, which is provided in the second phase, will help the researcher make a simulation model.

In the second phase, Project Replayer is used by developers to search answers to the questions. Project Replayer also helps to extract their recall. Originated questions are also important because they provide the developers with practical focus to deeply review the specific phenomena in past project. Just reviewing projects without any specific focus would be a very hard task.

In the third phase, the researcher tries to improve the simulation model according to the answer provided by developers.

The target project was for development of a typing-game. The project is operated for 24 days by six developers (students at NAIST). Program consists of 105 modules (.cpp files) and the final code size was 9,578 lines in total.

5.2 Results of the Experiments

In the first phase, the researcher (Subject0) made two hypotheses as follows;

(**H1**) If developers start to develop modules (.cpp files) on the end stage of project, the quality of the module is low.

(**H2**) If CVS's event behavior does not match to bug reports and e-mail data, the project is in confusion, and resulting software quality is low.

Meanwhile, the researcher also issued following questions about the phenomena of the projects:

(**Q1**) Why did not CVS data change from May 25th to June 5th?
(**Q2**) Why did the members delete many files from June 4th to 6th?
(**Q3**) How was the quality of these four modules: RankingScore.cpp, ScoreManager.cpp, ClickSocre.cpp, and GameSceneClickScore.cpp?
(**Q4**) Why was not CVS renewed during the last three days?
(**Q5**) How was the quality of the completed program?

In the second phase, the developers searched for the answers using Project Replayer. Resulting answers are shown as follows;

(**A1**: answer to **Q1**)
Subject1~3: I remember that we had an examination in that period so we had to suspend the development.
(**A2**: answer to **Q2**)
Subject1: I remember that we deleted image and sound files because it took long time to checkout from CVS repository.
Subject2: We deleted image files because we changed image format from BMP to PNG. I confirmed deletion of many image files by the Replayer.
Subject3: I think that some trouble occurred in the multimedia files, because the Replayer presented deletions of many image file and sound files.
(**A3**: answer to **Q3**)
Subject1: **H2** is doubtful, because the Replayer shows that any modules were not changed after the module completion.
Subject2: Most of them have good quality except of one module that was developed in only one day. Other modules were not revised once after they were completed. Therefore, the developers made carefully those modules, and those qualities were good. I confirmed this by the Replayer.
Subject3: It is bad. I expected the modules were developed in a hurry at the end of project. I confirmed in Project Replayer.
(**A4**: answer to **Q4**)
Subject1: Because the last three days were maintenance phase.
Subject2: Maybe, the last three days were demonstration periods.
Subject3: (No Answer)
(**A5**: answer to **Q5**)
Subject1: I remember that the total quality is not high because bugs occurred in scoring functions.
Subject2: The quality is not high. I remember there were bugs. I also found the existence of bugs using Project Replayer.
Subject3: Not good. I realized that the LoC graph of Project Replayer indicates the growth of the curve was not to meet the deadline.

In the third phase, regarding provided answers, the researcher validated the hypotheses. The validation results and new findings are discussed in the following section.

6 Discussion

In this section, we discuss the validation of the researcher's hypotheses and the new finding in the experiments of Project Replayer.

At first, validation of the researcher's hypotheses is discussed. The first hypothesis **H1** was not clearly backed up by the developers' answers. Especially Subject2 said definitely that quality was good (See **A3**) though other subjects had doubt to the module quality. Therefore, we would say that the researcher should have to consider other factors, not just two factors, i.e. development period (only one day) and calendar time (the end of project). After additional analysis of the four modules in detail using Project Replayer, the researcher found that those were developed at the end of the project. Two of them that use other modules handling game-scores were assigned suddenly to two new developers, while game-scoring modules were completed later. It can be assumed that the two new developers didn't have sufficient knowledge about game-scoring specification. In fact, first two modules couldn't properly handle game-scores. Other two modules were assigned to another developer who developed game-scoring. Therefore, Hypothesis: **H1** should change to the following;

(**H1'**) If developers start developing modules at the end of project and if the developers have little experience of developing the similar functions, the modules' qualities are not good (This hypothesis may be regarded as a concrete case of Brooks' law that is "adding people to a late project makes it later"[5].)

Next, the second hypothesis **H2** is discussed. The researcher at once considered that the deadline was the 16th of June because CVS data was recorded until the 16th of June. Then the researcher set the hypothesis **H2** because it seems strange that the growth of LoC has stopped before the last day. After regarding the developers' answer **A4**, the researcher realized that true deadline was the 14th of June. With true deadline, no modification to the source code during June14th-16th seems quite natural now. Therefore, the researcher has withdrawn the hypothesis **H2**.

In general, researchers can validate their hypotheses in many projects with help of Project Replayer just like this way. After the hypotheses have been refined in many validations with Project Replayer, the hypotheses will be raised to simulation models.

Now, we discuss the usefulness of Project Replayer in the developer's viewpoint. The developers replayed past project using Project Replayer in search for the answers to the questions. The developers can review past projects when developers acquire new knowledge in past projects' phenomena. All developers successfully recognized the problem of the file size in BMP format in **Q2**. Subject2 and Subject3 identified this problem using Project Replayer. They will use different file format (PNG) to avoid the problem in future projects.

In addition, Subject1 and Subject3 noticed the problem of program quality in **Q3** and **Q5**. If the developers do not review past project, they would not have any rethinking about their program quality. Once the project has completed, the developer's matter of concern moves to other topics, or real engineers in industry do not have enough time to review past projects. Project Replayer provides developers with easy way to review past projects in very short time. Subject1 and Subject3

deeply re-thought the program quality. They searched logically, not intuitionally, for the problematic programs in short time. Project Replayer also helps the developers think logically in various situations.

7 Conclusion

We have proposed the KFC concept to circulate valuable knowledge acquired from past project processes. In the KFC concept, valuable knowledge is finally formalized as a simulation model that will be used in future projects. Project Replayer and Project Simulator are key tools of KFC to accelerate the knowledge circulation.

This paper mainly described features of Project Replayer. We also conducted preliminary experiments of Project Replayer, and showed that developers can acquire knowledge from past project with help of Project Replayer. Project Replayer was also applicable to support researchers make simulation models. In order to establish the KFC environment, we regard other tools such as Project Simulator also to be implemented and embedded. Moreover, we perform further evaluation and validation of the KFC's effectiveness by more controlled experiments in many organizations.

Acknowledgements

We would thank the anonimous reviewers for their valuable comments and suggestions to improve this paper. We would like to thank Kimiharu Ohkura in Software Design Lab. at NAIST for his contribution in implementation of the Replayer

This research was partially supported by the Japan Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for Scientific Research (C) 17500024, and also by the EASE project [6] in Comprehensive Development of e-Society Foundation Software program of the Japan Ministry of Education, Culture, Sports, Science and Technology.

References

1. Ohira, M., et al.: Empirical Project Monitor: A System for Managing Software Development Projects in Real Time. in Proceeding of ISESE2004, Vol.2 (2004) 37-38.
2. Navarro, O., E., van der Hoek, A.: SIMSE: An Interactive Simulation Game for Software Engineering Education. in Proceeding of CATE, (2004)12-17.
3. Pfahl, D., Klemm, M., Ruhe, G.: A CBT module with integrated simulation component for software project management education and training. The Journal of System and Software, No.59, (2001)283-298.
4. Henderson, P., et al.: A tool for evaluation of the software development process. The Journal of System and Software, No.59,(2001)355-362.
5. Brooks, P., F.: The Mythical Man-Month. Addison-Wesley Pub, 1995.
6. EASE Project, http://www.empirical.jp/