

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

Keita Goto¹, Noriko Hankawa², and Hajimu Iida¹

¹ Graduate School of Information Science, NAIST, Japan
keita-g@is.naist.jp, iida@itc.naist.jp

² Faculty of Management Information, Hannan Univ., Japan
hanakawa@hannan-u.ac.jp

Abstract. In order to help knowledge acquirement 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. The Project Replayer is a most characteristic tool used to review data of past projects derived from development logs; CVS, bug reports and e-mails. With the Project Replayer, past projects can be easily revisited and complicated phenomena of the past projects can be investigated. As a result of preliminary experiments, we have confirmed that the 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 and technologies. 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 acquirement 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.

This paper mainly describes the concept of the KFC and features of Project Replayer. Section 2 shows related works. In section 3, we present an conceptual environment of Knowledge Feedback Cycle. In section 4, outline of the 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 Works

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. presents the system with integrated simulation component called CBT that is designed for software engineering education [3]. CBT provides interactive environment using standard web browser 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 (Knowledge Feedback Cycle) 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 (Empirical Project Monitor), Project Replayer and Project Simulator (See Fig. 1). KFC also involves two

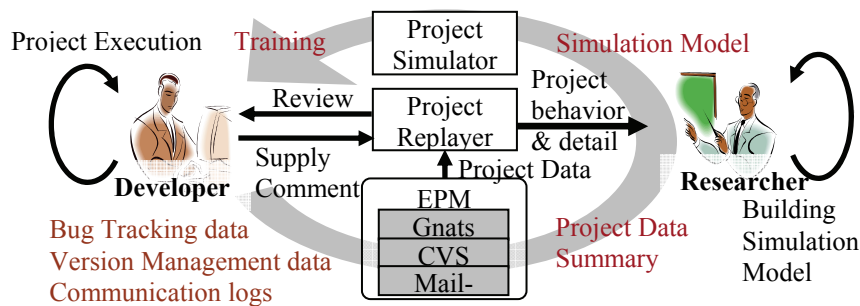


Fig. 1. Knowledge Feedback Cycle environment

human roles – software developers and software engineering researchers. Developers utilize the KFC environment in order to acquire new knowledge from the 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 the KFC would be as follows;

- Step1:** Various development data (records of code modification, bug tracking, and emails) 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 the Project Replayer and analysis tools (See “Researcher” part of Fig. 1).
- Step3:** Using the Project Replayer, developers review the 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/Estimation Model” part of Fig. 1).
- Step5:** Using the Project Simulator, novice developers learn complicated phenomena in the 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).

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

4 The Project Replayer

4.1 Purpose of the tool

The 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 Re-

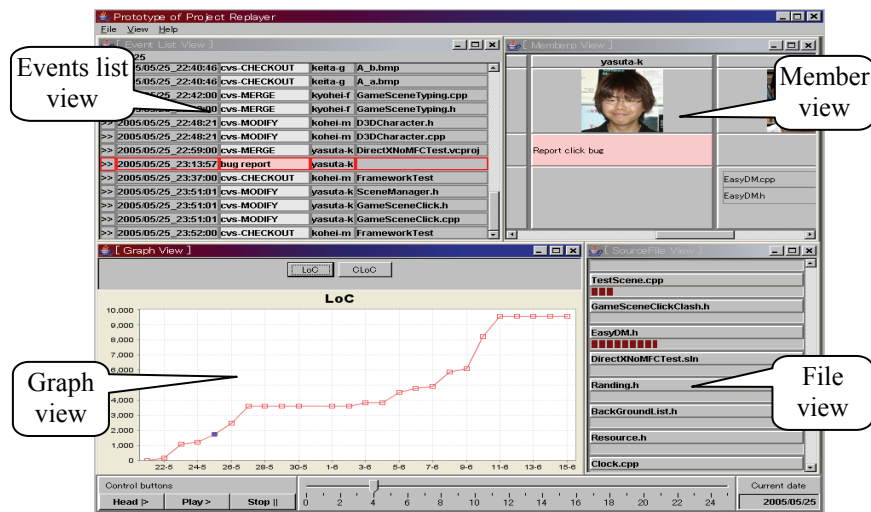


Fig. 2. Screen image of Project Replayer

player to revisit their past projects for postmortem evaluations, while researchers can use Project Replayer to deeply understand and analyze dynamic behavior of the projects. Replaying past real project is also important for education and training because simulators sometimes provide quite less reality for developers because they just provide abstract and ideal models. Project Replayer replays faithfully 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 the Project Replayer has four views (event list views, source file view, graph view and member view) and a time-control bar.

Event list view shows various (CVS, Bugs, and E-mail) events collected through EPM are listed in order of time. 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, fourth column shows owner of the event, and fifth column shows related filename.

Source file view presents source files in CVS repository. 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. 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. 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 he/she 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 the 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 the past project. Without any specific focus, just reviewing projects would be very hard task.

In the third phase, the researcher makes a complete simulation model according to the answer provided by developers.

The target project was development of a typing-game program used in our open-campus. 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 LoC 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 had 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 were the qualities of these four modules: RankingScore.cpp, ScoreManager.cpp, ClickScore.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 is 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 the Project Replayer, the researcher found that they were developed at the end of the project; two of them were assigned suddenly two new developers at the end of project. These two developers made these modules without regarding "Scoring" function up to the end stage of the project. The researcher may assume that these two developers don't have sufficient knowledge about "Scoring". The other two modules were assigned to another developer who developed other modules having "Scoring" functions. In fact, first two modules (RankingScore.cpp and ScoreManager.cpp) were not able to calculate the correct score of the game. 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.

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 the 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. We would say that the Project Replayer is very valuable tool for researchers.

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

In addition, Subject1 and Subject3 noticed the problem of program quality in **Q3** and **Q5**. If the developers do not review the past project, they would not have any re-thinking about their program quality. Once the project has completed, the developer's matter of concern moves to the other topics, or real engineers in industries do not have enough time to review the past projects. The Project Replayer provides develop-

ers with easy way to review the past project in very short time. Subject1 and Subject3 re-thought deeply 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 Knowledge Feedback Cycle concept to acquire and circulate valuable knowledge from past project processes. In the KFC concept, valuable knowledge is finally formalized as a simulation models that will be used in future project. Project Replayer and Project Simulator are key tools of the 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 the 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, further evaluation and validation of the KFC's effectiveness by more controlled experiments in many organizations.

Acknowledgements

We would like to thank Kimiharu Ohkura in Software Design Lab. at NAIST for his contribution in implementation of a parsing module 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 [5] in Comprehensive Development of e-Society Foundation Software program of the Japan Ministry of Education, Culture, Sports, Science and Technology.

References

1. Masao O., 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. EASE Project, <http://www.empirical.jp/>