

ReBEC: a Method for Capturing Experience during Software Development Projects

Gerardo Matturro¹, Andrés Silva²

¹ Universidad ORT Uruguay, Cuareim 1451, 11200 Montevideo, Uruguay
matturro@uni.ort.edu.uy

² Universidad Politécnica de Madrid, 28660 Boadilla del Monte, Madrid, Spain
asilva@fi.upm.es

Abstract. In software organizations, usual ways to capture the experience project team members acquire are based on methods such as project postmortem analysis, post-project revisions and others alike. Their main drawback is that the experience capture is done (if ever) after project completion, which leads to the risk of losing it if, as usually occurs, team members are finally not available. This paper introduces ReBEC (Reflection-Based Experience Capture), an approach that enables organizations to integrate the experience capture activities into daily software project tasks. We also present the case study of the implementation of this approach in a software organization. The study results show that ReBEC allows an earlier capture of knowledge and experience compared to existing approaches, and identify sources of knowledge as well as lessons learned and proposals of best practices.

Keywords: Knowledge management, software engineering, experience capture.

1 Introduction

The knowledge and experience that team members acquire as they carry out a software project conforms a valuable asset for organizations that aim to improve their software practices and processes for future projects [1]. For this to happen, software organizations should assure that the knowledge gained in a project is not lost, and it must be first captured and then, stored and managed for reutilization [2] [3]. Common approaches for capturing that knowledge and experiences are based on techniques such as semi-structured interviews [4], [5] or by applying methods such as project postmortem analysis [6], post-project revisions [7] and legacy sessions [8], among others. The main drawback of these approaches is that the capturing process usually takes place at a later time than of the occurrence of the experience itself and it is required that the people who own the experience be available to participate in this capturing process, which in general is not possible.

In this paper we present an approach to capture software project experience that differs from the above mentioned methods mainly in two aspects: the way the experience is captured, and the moment in the project life cycle this capture takes place.

This remainder of this article is organized as follows. In section 2 we present a general overview of the existing approaches for capturing software project experiences along with their main weaknesses. In section 3 the concept to reflective practice is introduced as a key element in the design of our approach. Section 4 is devoted to present a general overview of ReBEC, with a description of the phases it is structured, and to introduce the “reflective guides”, the proposed knowledge management tool used to capture experience. In section 5 we present the case study we conducted in a software organization to study the application of our approach. In section 6 we discuss the conclusion of the study. Finally, in section 7 we present the conclusions of the article and describe the future works and research lines.

2 Related Works and Criticisms

Several methods have been proposed for capturing of knowledge and experiences that project team members acquire as they carry out a software project.

The project post-mortem analysis [6] comprises three phases: preparation, data collection, and analysis. In the preparation phase all the documentation generated during the project is reviewed in order to determine the goals for the postmortem analysis. The data collection phase is the moment in which the relevant project experience is gathered and, once the important topics have been identified, they are prioritized before proceeding with the analysis phase. During this last phase, a feedback session is conducted in order to analyze the data collected and to find the causes for positive and negative experiences.

The legacy sessions [8] refers to the working sessions where project team members identify innovations and improvements that they have performed in their projects and that are potentially valuable for future users. A legacy session consists of four parts. The first part consists of a brainstorming session to identify potential legacies (apprenticeships that have the potential of being re-used by the members of the project team or by other members of the organization). In the second part, the participants synthesize the results of the former phase categorizing them as “processes”, “products” or “people”, and an element is chosen for subsequent discussion. The third phase is the detailed discussions of the chosen element for, in the fourth phase, create a summary of the revision done.

The post-project reviews [7] are a way to provide a formal mechanism to transfer experience from a project team to an organizational memory once the project has finished and while these experiences are still fresh in the minds of the participants. The captured experience is stored in a repository of learned lessons whose purpose is to facilitate the organization, maintenance and spread of the captured knowledge.

All the above described methods are characterized by the fact that the capture of experience is done later in time with regard to the actual occurrence of the experience, generally after finishing the project or by the time it has reached a relevant milestone.

The main problem with these approaches is that they arrive very late in the life of a project, if they are ever done, because when a project is finished, team members are almost immediately reassigned to a new one, taking with them their individual knowledge, and there is no time for such reviews [9], [10], [11], [12], [13].

Thus, to separate in time the experience and its subsequent capture involves some risks. Experience might be lost if team members are no longer available because they have been assigned another project or they have abandoned the organization.

Based on these considerations, a new approach is needed as long as: 1) it previously establishes the specific types of knowledge and experiences that are interesting to capture and 2) the capture of the knowledge and experience happens while the project is under execution.

Another topic that is central to our approach is the concept of “reflection”. Raelin defines *reflection* as the practice of periodically stepping back to ponder the meaning to self about what has recently transpired. Reflection illuminates what has been experienced by self, providing a basis for future action [14]. Reflection is considered to be an essential part of a learning process and reflective practice is the method by which reflection is made a conscious and structured activity. Reflection should be built into every activity, project or work piece in order to maximize learning from everyday activity [15]. Schön introduced the reflective practitioner perspective in which professional rethink and examine their work during and after accomplishing the creative process [16].

According to Hazzan and Tomayko, an analysis of the software engineering field, and the kind of work that software engineers usually perform, supports applying the reflective practitioner perspective to software engineering [17].

One traditional tool used in reflection activities is the reflective journal. A reflective journal records a learning item that took place as a result of reflecting on experiences and situations [15].

3. Proposed Approach for Experience Capture

Based on the considerations detailed in the previous section, we have defined a different viewpoint for capturing the knowledge and the experience that is acquired during software projects execution. This new viewpoint is characterized by a) incorporating the perspective of the reflective practice and the use of reflective diaries in the activities of software projects, b) take into account the problems pointed out in section 2 related to the traditional forms of post-mortem analysis and similar methods and c) integrating the experience capture process with the activities of software projects and software processes improvement efforts.

3.1 General Overview

Our approach is structured in four phases, as shown in Figure 1. From the set of the software practices and processes in use in the organization, in the Capture Objectives Definition phase a set of knowledge and experience capture objectives are establish for those software practices and processes that the organizations wants to improve. Based on these objectives, in the next phase the reflective guides are created. These guides contain a series of questions or sentences whose purpose is to guide and facilitate the analysis and reflection over the realization of the project tasks by the members of the project team.

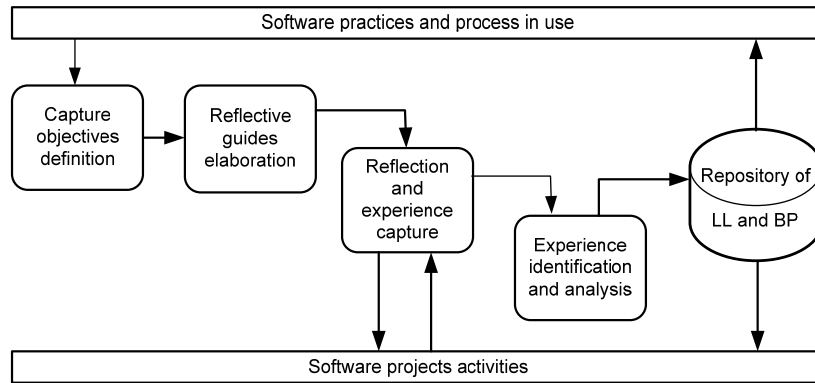


Fig. 1 Graphical representation of the proposed approach

Once these guides have been elaborated, the next step consists of assigning those guides to the members of the project teams who, during the execution period of their project tasks, use the guides to register their reflections and impressions, the difficulties found, unexpected events and similar considerations in relation to the way they those tasks are actually performed.

Once the project tasks are finished and the questions or sentences have been answered, the guides are collected for analysis and for identifying new knowledge and experiences captured in the answers. This last activity provides, for the next phase, the inputs of new knowledge and personal experience in order to carry out the process of identifying and taking out the lessons learned during the execution of the project activities and the identification of the proposal of best practices that, later, will be incorporated to a Repository of Learned Lessons and Best Practices.

These new captured experiences and knowledge will impact the manner in which project activities will be carried out in the future. They will become a basis for incorporating improvements to current practices and processes in use. This sequence of activities can be repeated in an iterative way to incrementally manage the creation of knowledge and the organizational learning based on experience, integrating the knowledge and experience management activities with software projects and software process improvement initiatives.

3.2 The Experience Capture Objectives Definition Phase

The purpose of this phase is to define the objectives of knowledge and experience capture for the target practices, techniques or software processes. To define these objectives we propose to use the well-known Bloom's taxonomy of educational objectives [18]. The way to express the capture objectives is to formulate questions and sentences that, taking into account the cognitive processes associated with the different levels of Bloom's taxonomy, point to those aspects of the practices or processes from which we will try to capture experience. Asking questions is nothing new. The difference here is that these questions are asked "before" the team members perform their project activities and not "after" those activities has been performed. In

this way, respondents know in advance the questions he/she will have to answer later, and find them in a better position to reflect on and to give a more detailed answer.

To elaborate these reflective questions or sentences, it is necessary to take into account: a) the concepts related to the practices, techniques or software processes in relation to which the experience is going to be captured and b) certain keywords usually associated with each level of Bloom's taxonomy, that express corresponding cognitive operations. Lists of key words can be found in [18].

3.3 The Reflective Guides Elaboration Phase

Once the experience capture objectives are defined, the next step is to elaborate the reflective guides. As we define them, these guides are a knowledge management tool whose purpose is to guide the reflection done by project team members about those project aspects from which it is desired to gather experiences, according to the experience capture objectives previously defined. We consider the reflective guide as a special kind of reflective diary that incorporates questions and statements elaborated in the previous step, and whose purpose is to guide the analysis and the reflection of team members with regard to those project activities that were assigned to them, and to focus their attention on those aspects of their activities from which we want to gather experience. For each question or statement, team members will be able to provide an answer with those reflections, comments and problems encountered during the execution of his/her project tasks.

3.4 The Reflection and Experience Capture Phase

Once the reflective guides have been elaborated, they are handed in to the members of the project team who are responsible to carry out the project activities respective of which the objectives to capture experience were formulated. This delivery is done in a brief meeting in which team members are given an explanation about the purpose and content of the guides, and how they are supposed to use them as part of their project activities. During this phase then, team members should use the guides as an aid to analyze and to reflect on the execution of their project activities, and to record their reflections and experiences as a way to answer the reflective questions or statements.

3.5 The Experience Identification and Analysis Phase

Once the project activities related to the reflective guides have finished and the team members have answered the questions, the guides are collected back for their analysis. This analysis consists in extracting from the answers those passages that might be considered "lessons learned" during the realization of project activities, and those passages that have the potential of becoming proposals for "best practices". The learned lessons and the proposals of best practices obtained will be incorporated into a repository to make them accessible to the rest of the organization (see Fig. 1).

3.6 The Repository of Lessons Learned and Best Practices

This repository, in our proposal, has a tree-like structure, based on the software engineering knowledge areas provided by the Guide to the Software Engineering Body of Knowledge [19]. Each leaf of the tree points to the actual lessons learned and/or best practices related to their corresponding knowledge area. Any team member in a new project will be able to access the repository and find the lessons learned and best practices relevant for the projects activities he or she is carrying out.

To ensure these knowledge assets are used in new software projects, when the reflective guides are assigned to team members they will include references to the lessons learned and best practices in the repository related to the project activities they are going to perform. In Figure 1, this is represented by the arrow that goes from the Repository to the “software project activities”.

Going through several iterations over the same project activities, those lessons learned and best practices can be continuously refined and, when they became stable, the knowledge and experiences they represent can be formally integrated into the organizational software process specification. In Figure 1, this is represented by the arrow that goes from the Repository to “Software practices and process in use”.

4. Empirical Study

To illustrate our approach and the use of reflective guides, we present here the study conducted from April to July of 2009 at the Software Factory (ORTsf), an academic unit within the Software Engineering department of the Universidad ORT Uruguay.

4.1 Research Questions and Projects Selected for the Study

The two research questions for the study were stated as follows: 1) How to capture the knowledge and experience that team members acquire during software projects, in a way that such capture is done earlier than with existing approaches; 2) What kind of knowledge and experience can be captured by using the reflective guides.

The strategy chosen was a case study [20]. The conditions of the context are particularly important to be considered because we aim to study the proposed approach as embedded into the daily working activities of the members of a real software project team working in real software development.

Three independent software development projects were considered in this study, as shown in Table 1. The project teams were integrated by 3 to 5 students of the last course of Systems Engineering career at the University. In each team, typical roles in software projects, like project manager, requirements engineer, architect, developer and tester, were distributed among its members. A working condition for the teams was to work together on-site (in the facilities of the University) for at least 10 hours weekly, in order to promote team cohesion and also to have a similar working ambience to that of a software organization. The remaining 20 expected weekly hours in the project, the students had the freedom to work at the University or in any other alternative place at their choice. Each project had a “real” customer, namely, an

organization, independent of the University, to which the products was targeted. This characteristic of the chosen projects makes the work of the project groups similar to that in software development organizations: several projects being developed simultaneously, not uniform development practices through the different projects, and different deadlines, handing in dates and commitments with the respective customers.

Table 1 Projects selected for the empirical study

Name	Description	Persons
COODESOR	Management system for a dentistry medical organization	4
GESA	Management system for the Uruguayan accreditation organism	3
SCPI	Investment projects follow up and control system	5

4.2 Reflective Guides Content and Elaboration

The first step in defining the contents of the reflective guides, is to define which practices and processes they will focus on. In ORTs^f there is historical data regarding the software practices that are usually assessed as “inefficient” when performed by project teams. Based on this data, we use ReBEC to capture experiences aimed to improve the processes of defining metrics for project management.

We used this software engineering activity as a base for the definition of the aims of capturing the experience and for elaborating the reflective guides for the case study. The questions and statements included in the guides were elaborated by the first author in cooperation with a member of the ORTs^f staff who, based on working experiences with other former project teams, has good knowledge about the specific aspects of the chosen practices that are important to take into account.

4.3 Data Collection and Analysis

The reflective guides were elaborated and given to the project managers of the competing teams, who used them during their project activities and returned them back with all questions answered. What follows are extracts of those answers for some of the reflective questions. The full guides with the complete answers can be obtained from the first author.

Question 3 and extracts of answers are presented in Table 2.

Table 2 Answers to question 3 (Level 4, Analysis, of Bloom’s taxonomy)

Q. 3: According to your experience, how could you overcome the difficulties for identifying metrics useful for your project?
COODESOR: ... today the difficulty we have is identifying some metric that is missing from our point of view ... to overcome this I will be reading the literature on the issue and I also want to meet with the Manager role tutor to learn how we are situated related to metrics.
GESA: ... we used documentation from previous projects, meetings with role tutors for project management tasks and meetings with the group tutor.
SCPI: With the help of the project’s tutor, of the reviewer and the SQA role tutor ... the information available in the Software Factory web site was very useful ...

Question 5 and extract of answers are presented in Table 3.

Table 3 Answers to question 5 (Level 6, Evaluation, of Bloom's taxonomy)

Q. 5: Report, in a few lines, the lessons learned during the metrics planning process.
COODESOR: ... what I can say is that, given that I work in a maintenance project, I have observed the differences between a maintenance project and a development project ...
GESA: ... the metrics used in other projects cannot always be reused, each project must be evaluated by itself and individual metrics defined for that project. What we hear in class or read in books may not always be applicable to the project we are carrying out.
SCPI: ... we realized that just the "theoretical" framework is not enough, because metrics need to be adapted to the project's reality.

Finally, question 7 and extract of answers are presented in Table 4.

Table 4 Answers to question 7 (Level 5, Synthesis, of Bloom's taxonomy)

Q. 7: Of those metrics-related management activities you carried out, which ones you consider your performance was adequate and which should be improved?
COODESOR: ...having a talk with the team to make them understand the significance of time records...define the metrics to be used at the very beginning of the project, something that I did not do because of lack of experience and, establishing them after X time after the project had begun it is harder to collect the information needed for metrics to represent reality ... the metrics to use have been well selected ... avoiding collecting metrics that do not contribute too much and that consume more time from the project.
GESA: I consider that the activity records of the group members were correctly carried out. I would improve the iteration estimation task ...
SCPI: ... we still cannot say what we did right or wrong ... later on, when we use the collected data, we may actually know the errors we made in planning.

To answer the research questions asked in sub-section 5.1, we proceeded to perform a qualitative analysis of the answers given by the people taking part. This analysis comes from abstracting away those elements that are considered to be important or pertinent to answer the research questions [21]. In our case, these elements were chosen from those answers that present chunks of experience acquired during project activities. To refer to these sections of the answers, in the following we will use the convention (Project Name, Question number from which the answer was extracted).

With regard to the first research question, the analysis of the answers indicates that they refer to accurate aspects derived from the experience of having executed the project activities, elaborated during the period in which the definition of the project management metrics activities were performed. One of the criticisms formulated to the existing approaches for capturing experience is that the process of capturing usually happens once the project is finished. With the reflective guides, this capturing process occurred while projects unfold, because team members were asked to use the reflective guides and to elaborate the answers as part of their project tasks.

With regard to the second research question, the analysis of the answers allows us to identify sources of knowledge, learned lessons and proposals of best practices.

Regarding the identification of the source of knowledge, the answers allow us to identify sources of explicit as well as of tacit knowledge. Expressions such as

“...meetings with role tutors... and with the group tutor...” (GESA,3), “...with the help of the project tutor, the reviewer and the SQA role tutor...” (SCPI,3) enable identification of tacit knowledge sources. Similarly, expressions such as “...we used documentation from previous projects...” (GESA,3), “...the information available in ORTs web site was very useful...” (SCPI,3) indicate sources of explicit knowledge.

The knowledge and experience captured in the guides enable us the identification of lessons learned, derived also from carrying out the project tasks. Expressions such as “...something I did not do because of lack of experience and that were more difficult to collect a long time after the project had begun...” (COODESOR,7), “...metrics need to be adapted to the project’s reality...” (SCPI,5) show learned lessons during the project activities.

With regard to the identification of proposals of best practices, expressions such as “...having a team meeting to make them realize the significance of keeping time records...” (COODESOR,7), “...establishing the metrics to be used at the very beginning of the project...” (COODESOR,7) may be considered recommendations to follow that, adequately developed, will allow the formulation of best practices.

5 Conclusions and Further Work

In this article we presented ReBEC, a new approach for capturing experiences in software projects, by using a knowledge management tool we named “reflective guides”. Different from the pre-existing approaches discussed in section 2, the proposed method is based on a previous establishment of the specific types of knowledge and experiences that are interesting for capture, and on the capture of this knowledge and experience as projects unfold, instead of waiting to capture them once the project is over. In this way, problems and inconveniences that arise from postponing the capture of experience after the project ends, as discussed in section 2, are solved.

We also presented an empirical study for an implementation of the proposed approach in a software organization, with the purpose of showing that the reflective guides constitute an adequate tool for the capture of knowledge and experience that members acquire during the realization of a software project. The reflective guides were used by the project managers of the selected projects as an aid to facilitate the reflection on their respective project activities, and as guidelines to capture the knowledge and experience they acquired while doing their project activities. The qualitative analysis of the answers allowed us to gather, at least, three kinds of knowledge artifacts: sources of knowledge in the organization, lessons learned during the execution of project activities, and proposals of best practices.

Based on these positive results, in ORTs we are planning to extend the use of the reflective guides to capture experiences from other software engineering activities.

One aspect that requires additional work is the possibility of enrichment of the answers to the reflective questions, in order to allow team members to add more details or enhanced descriptions of their experience. At first sight, this can be done by introducing the role of a facilitator to help team members in improving their abilities to reflect and to write better answers.

Another question that also deserves further research relates to the time consumed by team members in answering the reflective guides and the other activities of ReBEC, such as reflective guides preparation and the analysis of the answers given, and how it compares with the times required to carry out the other methods presented in section 2. We consider that it is necessary not only to have quantitative data about this timing. Qualitative data is also needed in order to take into consideration the quality and richness of the results obtained with ReBEC, compared to others methods.

References

1. Briand, L.: On the many ways software engineering can benefit from knowledge engineering, Proc. 14th International Conference on Software Engineering and Knowledge Engineering, pp. 3-6 (2003)
2. Rus, I., Lindvall, M.: Knowledge Management in Software Engineering, IEEE Software, 19(3), pp. 26-38 (2002)
3. Antunes, B., Seco, N., Gomes, P.: Knowledge management using semantic web technologies: An application in software development, Proc. 4th International Conference on Knowledge Capture, pp. 187-188 (2007)
4. Komi-Sirvio, S., Mantyniemi, A., Seppanen, V.: Toward a practical solution for capturing knowledge for software projects, IEEE Software, 19(3), pp. 60-62 (2002)
5. Scott, L., Jeffrey, R.: An anatomy of an experience repository, Proc of the 2003 International Symposium on Empirical Software Engineering, pp. 162-173 (2003)
6. Birk, A., Dingsøyr, T., Stalhane T.: Postmortem: never leave a project without it, IEEE Software, 19(3), pp. 43-45 (2002)
7. Harrison, W.: A software engineering lessons learned repository, Proc 27th Annual NASA Goddard/IEEE Software Engineering Workshop (2003)
8. Cooper, L., Majchrzak, A., Faraj, S.: Learning from project experiences using a legacy-based approach, Proc 38th Hawaii International Conference on System Sciences (2005)
9. Basili, V. Lindvall, M., Costa P.: Implementing the Experience Factory as a set of experience bases, Proc 13th International Conference on Software Engineering and Knowledge Engineering, pp. 102-109 (2001)
10. Desouza, K., Dingsøyr, T., Awazu, Y.: Experiences with conducting project postmortems, Proc 38th Hawaii International Conference on System Sciences, (2005)
11. Cooper, L.: Converting project team experience to organizational learning, Proc 40th Hawaii International Conference on System Sciences, (2007)
12. Bjørnson, F.: Knowledge management in software process improvement, Doctoral Thesis, Norwegian University of Science and Technology, (2007)
13. Zedtwitz, M.: Organizational learning through post-project reviews in R&D, R&D Management, 32(3), pp. 255-268 (2002)
14. Raelin, J.: Work-based learning, Prentice Hall, Upper Saddle, New Jersey (2002)
15. Clifford, J., Thorpe, S.: Workplace learning and development, Kogan Page, London (2007)
16. Schön, D.: Educating the reflective practitioner, Jossey-Bass, San Francisco (1987)
17. Hazzan, O., Tomayko, J.: Reflection and abstraction in learning software engineering's human aspects, IEEE Computer, 38(6), 39-45 (2005)
18. Bloom, B., Engelhart, M., Furst, E., Hill, W., Krathwohl, D.: Taxonomy of educational objectives. Handbook I: Cognitive domain, David McKay, New York (1956)
19. Abran, A., Moore, J.: Guide to the software engineering body of knowledge, IEEE Computer Society, Los Alamitos, California (2004)
20. Yin, R.: Case study research. Design and methods, Sage, Thousand Oaks (2003)
21. Blaxter, L., Hughes, L., Tight, M.: How to research, Open University, Philadelphia (1996)