A Qualitative Study about the Life Cycle of Lessons Learned

Davi Viana, Jacilane Rabelo, Tayana Conte
USES Research Group
Federal University of Amazonas
Manaus, Amazonas - Brazil
{davi.viana, jaci.rabelo, tayana}@icomp.ufam.edu.br

Andréia Vieira, Ellen Barroso, Mário Dib
Fundação Des. Paulo Feitoza
Manaus, Amazonas - Brazil
{avieira, ellen.barroso, mario.silveira}@fpf.br

***Abstract*—Software activities are executed by people and demand great knowledge. For this reason, knowledge dissemination activities are important in software organizations. One of the ways of sharing knowledge is through the practice of lessons learned dissemination. The form of dissemination could help to clarify questions about how lessons learned are shared in the organization. This paper aims to analyze the life cycle of lessons learned in a software organization, in order to understand how they are treated during the course of the projects. Results show that discussions of the lessons encourage the exchange of knowledge between the team members. However, it is necessary to improve the form of knowledge dissemination in all the organization, as well as encourage learning this knowledge.**

***Keywords*—Knowledge management, lessons learned, qualitative analysis, interviews, ethnography**

I. INTRODUCTION

Software development and maintenance are activities executed by people and demand an intensive effort in knowledge [1,2]. Nevertheless, carrying out activities that require the application of much knowledge is not trivial [3]. Professionals must have knowledge of the technologies and software engineering methods. This knowledge has to be analyzed and shared between the employees. Thus, it is possible to react to the client's and the market's demands in a better way [1].

Knowledge Management (KM) contributes to the organizational knowledge treatment process [4], apart from making this process less complicated [5]. KM consists in systematically collecting and storing the acquired knowledge and sharing it through a corporate memory [6,7]. One way of approaching KM in an organization is through lessons learned.

Lessons learned could describe discussions and solutions to resolve problems or opportunities of improvement [8]. They contribute to the prevention of similar faults in the projects [9, 10]. An organization which does not register lessons learned is subject to repeating faults [6]. The loss of this relevant knowledge could occur because of failure to register and forgetting the solutions [11]. This way, analyzing the activities of creation and sharing lessons learned could help improving the execution of KM in software projects. In this paper, we search to understand in which situations lessons learned are created and used in the software organization. This research seeks to understand how lessons learned are created and maintained in software projects, as a form of knowledge dissemination.

To find this better understanding, a qualitative study was carried out based on interviews and ethnography. The study was executed in a Research and Development Institute which encourages the creation of lessons learned in the work environment. The initial results of the analysis show that the employees define the lessons learned through discussions and consent of the whole team at the end of each development stage. The life cycle of lessons learned contributes to the sharing of important project issues.

The rest of this paper is structured as follows: Section 2 presents works that seek to identify the knowledge flow and activities. Section 3 describes the research method defined and the study setting. Section 4 discusses the preliminary results of the study. Finally, Section 5 presents the conclusions and future works.

II. BACKGROUND

Managing knowledge can be considered a competitive strategy for an organization, allowing it to help professionals improve their productivity [2]. Mitchell and Seaman [12] use the KM technique of "knowledge mapping" as a research technique to characterize the knowledge flow in software projects. This characterization indicates improvements to software process. These researchers verified that the software engineers strongly depended on explicit and implied knowledge resources. Furthermore, they verified that there is great confidence in the knowledge flow internal to the teams.

Boden *et al*. [13] analyze four ways of knowledge sharing: status meetings and maintaining awareness, the collaborative use shared artifacts and repositories, spending time at the other site and human "bridges" that mediate between people and cultures. The authors seek to understand how cultural and social questions can influence in the way of knowledge exchange in global software development (GSD). As a result, it was noticed that the ways of knowledge sharing need to be studied in context, for a better understanding of sharing.

In addition to the research identified above, it is necessary to understand how this knowledge is shared and maintained in the everyday working life, in different contexts.

III. Research Method and Study Setting

One way to investigate the point of view of professionals is to use qualitative methods. Such methods support a better comprehension of the issues that need a more specific and detailed analysis. According to Seaman [14], the use of qualitative methods allows the researcher to consider human behavior and thoroughly understand the object studied. This way, we hope to obtain a more adequate understanding of the employees' practices as they approach the lessons learned in the organization in question. This qualitative study was carried out through interviews and direct observations.

The usage of both data collection methods, interviews and observations, allowed the triangularization of the data [15]. Thus, there is a reduction in the researchers' bias effects [16].

A. Study Setting Details

The interviews and observations took place at a Research and Development Institute in Manaus in the North Region of Brazil. Since 1998, this institute has worked with the technological development of various sectors, including, industrial, medical, apps for mobile devices, business process flow, besides developing research related to accessibility and digital inclusion of disabled people. Furthermore, this institute values the KM, including specific activities in its software process.

The Institute has a software process defined, containing a few practices of Scrum methodology, such as the lessons learned meeting. At the end of each sprint, the lessons learned meeting takes place. This is where important issues of the project are discussed.

One author followed six lessons learned meetings of five different projects. Furthermore, interviews were made with eleven employees with the aim to obtain richer results. The results collected during the course of the research were analyzed together with the data obtained from the interviews.

For the interviews, a semi-structured questionnaire was used with open questions. The questions are related to the way the lessons learned are approached in the software projects. The next section describes the preliminary results of this study.

IV. Our Findings

In each development sprint, the project team identifies important points about the project execution. These issues are discussed during the lessons learned meetings, always at the end of each sprint. The team classifies the lessons into two categories: "worked well" and "needs to be improved".

A "worked well" lesson is related to any issue or event that has worked well during the execution of the last sprint. Normally, this type of lesson is related to a new approach or new way of carrying out a certain activity. For example: "*Use XYZ framework helped in the developer's productivity*".

A "needs to be improved" lesson is any issue which did not work out well in the last sprint. This issue needs to be improved in the next sprints. These lessons need to have an action plan with the improvement proposed. These actions contain information about: what can be done, when it can be done, person responsible, the action status. An example of this kind of lesson learned is: *"Improve the communication with the client, as well as register this communication"*. A possible defined action for this lesson is: "*Present the communication plan to the client. This will take place from the next meeting onwards. The person responsible for this action will be the project leader*". The following quotation illustrates the definitions of the types of lessons presented by the employees.

> *"[lessons of the type "needs to be improved"] are all the issues you can use to improve the project's development or to maintain something that is already working [lessons of the type "worked well"](...)* – Interviewee 1

The two types of lessons learned are important for the project. They expose points of views from the project teams about issues which can effect (positively or negatively) the entire project's development. The collaborators register the lesson learned in a spreadsheet standardized by the organization. It is possible to register lessons learned related to various development stages, including technologies, estimation definitions, project planning and management, and requirement treatment.

The lessons learned of the type "need to be improved" are the most critical to the project. They are related to the issues that can negatively affect software development. When questioned about which type of lesson is most interesting for the project, the collaborator answered:

> *"Those which impact the development (...), are those which we try to solve the most"* – Interviewee 3

The following subsections present the three phases of the lessons learned' life cycle.

A. Lessons learned definition

We noticed there are two ways to start the lessons' definition: previous identification and identification during the meeting.

Previous identification of lessons learned – each collaborator defined his/her lessons learned before the meeting. When the meeting starts, the project leader consolidates all the lessons learned defined by the team. Next, these lessons are discussed. If the lesson learned is of the type "needs to be improved", the collaborator who defined it also describes the action to be executed. The interviews allowed us to identify a point which was not possible to obtain with the observations: i.e. which sprint moments the collaborators raise the lessons learned. By the collaborators' answers, we noticed there is not a consensus as to when to raise the lessons.

> *"(...)We register at the end [of the sprint], but I have the habit of, when I see something happening which isn't good, I write it down (...). So when I fill out [the spreadsheet] I already remember where it is"* – Interviewee 1

> *"[The identification] is at any moment. Depends on what is being developed (...) and also on the team"* – Interviewee 4

Lessons identification during the meeting – The second type of identification is similar to the first type, with the difference that the lessons learned were thought and defined during the meeting.

During the observations, we noticed a bigger integration between the collaborators when the lessons learned were defined before the meeting. However, when the lessons were defined during the meeting, not all the collaborators participated with an adequate definition of the lessons learned. The following quotation confirms the observations:

> *"Because they leave it to be done too late, it's that point that the person isn't going to remember. So lessons which could be collected in advance aren't learnt or are forgotten. And, collecting before, wins you some time in the meeting to discuss instead of taking notes."* – Interviewee 7

After identification, the lessons learned go on to the monitoring phase. In this phase, the lessons learned of the type "needs to be improved" are monitored.

B. Monitoring the lessons learned

We observed that the lessons of the type "worked well" aren't verified again, for they are issues which worked adequately. In this type of lesson, a plan of action doesn't need to be created. An employee is questioned about the treatment of the learnt lessons after the meetings.

> *"What do you do after a meeting learnt with these lessons? The points which worked well, nothing. The ones which need improvement, we try to follow the actions (...)in the next meeting we verify if [the action]it was effective or not."* – Interviewee 4

This monitoring phase occurs only with the lessons learned of the type "needs to be improved". There is not a consensus as to how the lessons are treated. The following quotations present examples of lessons' monitoring.

> *"After the learnt lesson, we talk informally and try to improve; we try to apply what needs to be improved as a suggestion."* – Interviewee 6
>
> *"(...) Is there a specific moment when you remind the employees what has to be done [after the definition of the lessons]?*
>
> *In the daily meeting"* –Interviewee 3

In the following lessons learned meeting, the collaborators analyze the lessons that have the action to be carried out and verify the effectiveness of the action. An action is effective when the issue truly improved. There has to be a consensus between all collaborators. An interviewed affirmed that an action is considered effective when an action which needed to be improved did not happen again and the action was understood by the team.

> *"In the next cycle that action didn't occur anymore, it worked out, so ok. It was taken care of."* – Interviewee 11

There are a few reasons which explain the ineffectiveness of the actions, the study made us identify the following reasons: (1) end of project: at the end of the project, we noticed it was not possible to improve a certain issue; (2) the relationship of the project with the lesson learned no longer existed: decisions in the project made it so that the lesson learned was no longer necessary, such as a change of technology; (3) the purpose of the action/lesson is not clear: the description do not have enough detail to close the lesson.

Some collaborators stated the use of some lessons learned made by collaborators from other teams. This use can offer the sharing and learning of certain practices.

> *"In the tool, in kanban, you can assign colors to tasks. (...) When you have to visualize it, it is easier. And they [a team] adopted this and it was a lesson learned for them. I saw them using it (...) I got this and passed it on to other projects that use that tool, as a good practice."* – Interviewee 11

Even though the organization's process only supports sharing knowledge inside the team, we noticed there is sharing between teams. This type of sharing was informed by a collaborator who is hierarchically above two or more teams. Normally, these superiors do not participate in lessons learned meetings, but they have access to the lessons created. As they verify that one lesson learned can be applied to other projects, this employee passed the lesson, through personal communication. It is good to highlight that each team only has access to the lessons learned created in their project, this way we noticed the tacit knowledge was shared. Similar information was obtained by collaborators who have participated in many projects in the organization in leadership roles.

C. End of lessons learned

If a certain action was efficient, the project leader changes the status to "concluded". From this moment onwards, the lesson is not monitored; however it remains available to the team. When an action is considered inefficient, new actions must be identified, aiming to solve the problem.

In case the action was not efficient but the problem did not happen again, or, for another reason, that lesson learned stopped being analyzed, the leader cancels the lesson, according to the reasons presented in the monitoring phase. However these reasons for cancelling were not registered.

The collaborators affirmed that, when they ended the project, the lessons with actions to be carried out were also "finished", even though they were not concluded.

D. Results discussion

According to Sharp and Robinson [17], the ethnography study can uncover implicit features of practice and emphasize the social order of practice. In this study, we noticed there is much capture and knowledge sharing inside the projects. This sharing happens through the collaborators' tacit knowledge discussions.

The organization also previews a way of sharing this knowledge between the members of the team, through the lessons learned spreadsheet. The explicit knowledge is available through the lessons described in the spreadsheet, but there were no reports or observations which confirm the consultation of the spreadsheet. Therefore, we need to improve the promotion of this knowledge dissemination between the teams. This is a goal for future research.

A way of stimulating the use of lessons created by other teams is through the creation of an infrastructure which allows knowledge dissemination. With this, one may also avoid creation of repeated lessons learned. An interviewed affirmed

that a repository is being created with this goal. However, it is necessary to check if this repository is going to be enough to disseminate knowledge, once the spreadsheet was not used in other projects. This way, it is necessary to analyze way to stimulate consultation and use of the knowledge and make this part of the organizational culture. This is another issue for future investigation.

V. Conclusions and Future Work

This paper discussed the life cycle of lessons learned in software projects in a Research and Development Institute in the North Region of Brazil. A qualitative study was executed involving interviews and observations seeking to understand knowledge treatment during the project's execution.

The life cycle of lessons learned identified has three main phases: definition, maintenance and conclusion. In the definition, the lessons learned are created and classified into two types: "worked well" and "needs to be improved". In the maintenance occurs the execution of the actions related to the lessons of the type "needs to be improved". Lastly, in the conclusion, the lessons are classified according to their efficiency. This cycle shows the way in which the lessons learned are approached in the organization. In addition, it was possible to obtain findings about tacit knowledge dissemination.

Apart from that, it is necessary to analyze how the explicit knowledge can be encouraged. This way, one can investigate possibilities of using the knowledge generated to benefit the organization.

The study has some limitations related to the scope observed. We observed the lessons learned meeting environment, which could have limited the vision of lessons learned' life cycle. The researchers carried out interviews as a way of reinforcing what was observed and seek to increase the access to collected data.

Each qualitative study contributes to advancing the state of art in a research area, providing evidence and hypothesis that can be later tested using quantitative methods. In short, each study helps building a body of knowledge about the life cycle of lessons learned. In this paper, we present evidences about the behavior of lessons learned in a software organization. Hypotheses rose as results were discussed, this will make it possible to continue this investigation. Furthermore, we plan to go deeper in the qualitative analysis of the data obtained aiming to analyze organizations in different contexts. This extension will allow us to gain more knowledge about lessons learned and identify other factors involved in the creation, maintenance and assimilation of the organization's knowledge, as support strategies to organizational learning in software companies.

Acknowledgment

We would like to thank the Research & Development Institute. Also, we would like to thank the financial support granted by: FAPEAM through of Support Program for Pos-Graduated Human Resources - RHTI - Doctorate, N.009/2012; and FAPEAM through Project N.021/2011 - Universal Amazonas. Finally, the authors especially thank Cleidson de Souza for the original idea of the research and for all his support during this and other researches.

References

[1] K. Schneider. *Experience and Knowledge Management in Software Engineering*. Berlin: Springer, 2009

[2] M. Levy and O. Hazzan. "Knowledge Management in Practice: The Case of Agile Software Development", in: 2nd Intern. Workshop on Cooperative and Human Aspects on Soft. Eng. (CHASE) - ICSE Workshop. Vancouver, 2009, pp. 60-65.

[3] M. Gomes, S. Carolyn, V. Basili, and Y. Kim. "A Prototype Experience Management System for a software Consulting Organization", in: 13th Conference on Soft. Eng. and Knowledge Eng. SEKE'01. Buenos Aires, 2001, pp. 29-36.

[4] G. Probst, S. Raub, and K. Romhardt. *Managing Knowledge: Building Blocks for Success*. 1. ed. New York: Wiley, 1999.

[5] T. Davenport, L. Prusak. *Working Knowledge: How Organizations Manage What They Know*, 1. ed., Boston: Harvard Business School Press, 1998.

[6] L. Borges and R. Falbo. "Managing Software Process Knowledge", in: International Conference on Computer Science, Soft. Eng., Information Technology, e-Business, and Applications (CSITeA'02), Foz do Iguaçu, 2002, pp. 227-232.

[7] G. Avram. "At the Crossroads of Knowledge Management and Social Software". *The Electronic Journal of Knowledge Management*. vol 4, issue 1, pp. 1-10, 2006.

[8] D. O'Leary. "Enterprise Knowledge Management". *IEEE Computer*, vol. 31, pp. 54-61, 1998.

[9] O. Bjørnson and T. Dingsøyr. "Knowledge Management in Software Engineering: A Systematic Review of Studied Concepts, Findings and Research Methods Used". *Journal of Information and Software Technology*, pp. 1055 – 1068, 2008.

[10] A. Natali and R. Falbo. "Knowledge Management in Software Engineering Environments", in: Proceedings of the XVI Brazilian Symposium on Software Engineering (SBES'2002), Gramado, 2002, pp. 238-253.

[11] M. Mendonça, C. Seaman, V. Basili, and Y. Kim. "A Prototype Experience Management System for a Software Consulting Organization", in 13th Conference on Soft. Eng. and Knowledge Eng (SEKE'01), Buenos Aires, 2001, pp. 29-36.

[12] S. Mitchell and C. Seaman. "Software process improvement through the identification and removal of project-level knowledge flow obstacles", in: 34th International Conference on Software Engineering (ICSE), Zurich, 2012, pp. 1265-1268.

[13] A. Boden, G. Avram, L. Bannon and V. Wulf. "Knowledge sharing practices and the impact of cultural factors: reflections on two case studies of offshoring in SME". J*ournal of Software: Evolution and Process*, v. 24, n. 2, pp. 139-152, , 2012.

[14] C. Seaman, "Qualitative Methods", in: Guide to Advanced Empirical Software Engineering, Shull et al. (eds.): Springer, 2008.

[15] K. Rönkkö. *Ethnography*. Encyclopedia of Software Engineering, pp. 278-286, 2010.

[16] D. Fetterman. *Ethnography: Step-by-step*. 3rd ed. Los Angeles: Sage, 2010.

[17] H. Sharp and H. Robinson. "An ethnographic study of XP practice". *Empirical Software Engineering*, v. 9, n. 4, pp. 353-375, 2004.