Using Post Mortem Analysis to Evaluate Software Architecture Student Projects

Alf Inge Wang and Tor Stålhane
Dept. of Computer and Information Science,
Norwegian University of Science and Technology,
Sem Sælandsv. 7-9, NO-7491 Trondheim,
Phone: +47 73594485 , email: alfw/stalhane@idi.ntnu.no

Abstract

In this paper we present results from using the post mortem analysis (PMA) to evaluate student projects in a software architecture course at the Norwegian University of Science and Technology (NTNU). The PMA gave students a change to evaluate their own work as well as evaluating the project exercise itself. The results of the analysis revealed several positive and negative issues related to the project that could be used to improve the course next year. We also discovered that the PMA gave us a much more detailled evaluate than using more traditional course evaluation methods.

Keywords: *Post mortem analysis, Software architecture, Education, Student project.*

1 Introduction

To learn from your mistakes is a key to improve. This is also true for teaching software engineering. However, in order to improve software engineering related courses we must identify the weak and the strong points. Traditionally, course evaluation, where students must fill in evaluation forms, is used as a tool to improve courses. In software project, a method called post mortem analysis (PMA) has been used to elicit strong points and weak points of the project. In this paper we describe experiences and results from applying the same method on a project being a part of a software architecture course. By using PMA to evaluate a student project, the students can learn this method, they can evaluate their own accomplishment of a project, and they can evaluate the project given in the course as a whole.

2 The Context and the Experiment

The PMA was conducted as a part of a software architecture course for 4th year master students at the Norwegian University of Science and Technology (NTNU). The course teaches students the main principles, methods and models of software architecture through lectures, exercises and a project where most of the theory must be used to succeed. The textbook "Software in Practice, Second Edition" by Bass, Clements and Kazman [1] is used along with some additional papers on architectural description and modelling and design patterns. Every year about 50 students attend the course where most of the students are

4th year master students and some are 3rd year students. The credit of the course is 25% of the total workload for one semester. For the exercises and the project, the students are divided into groups of 4. The project plays a major role of the course and spans over 8 weeks and constitute 30% of the grade of the course. The goal of the project is to create and implement a software architecture for a mobile robot that should search a maze for randomly placed balls and bring them all to a light source within the maze. The robot has distance and light sensors and can move by using the engine on its two wheels. The WSU Khepera robot simulator [7] for Java is used to implement and test the robot. The final architecture of the different students group should reflect the fact the groups should focus on various quality attributes like availability, performance, maintainability, and permutations of the three. The project was divided into four deliveries:

1. Delivery 1: A requirement specification, a quality attribute description and the architectural design.
2. Delivery 2: Evaluation of another group's architecture using the Architecture Tradeoff Analysis Method (ATAM). Here to groups evaluated the other group's architecture in turn and it was up to the involved groups to find time and room to do the evaluation.
3. Delivery 3: The whole project including revised delivery 1 and 2, and implementation of the robot controller.
4. Delivery 4: A PMA with comments.

In 2004, we decided to introduce and use PMA to evaluate the student project. The motivation for using PMA was based on three main reasons. Firstly, we wanted to have a deeper and more thorough evaluation to understand the reasons for positive and negative student experiences of their performance of the project. Secondly, we wanted to teach the student the PMA and give them practice in using the method so that they could use it themselves on real software projects when they start to work for software companies. Thirdly, we wanted to evaluate the project itself and identify how the project could be improved for next year.

The PMA of the software architecture course was scheduled at the very end of the course after finishing the project and lectures. The students where requested to use 3 hours on the PMA that was compulsory to get the project delivery accepted. In 2004, we had 13 groups where most groups were consisting of 4 students and some groups with 3. In the PMA we focused on two questions: What went wrong and what went well in the robot project?

After a short introduction to PMA, the 13 groups were split into four larger groups that were sent to four group rooms equipped with a blackboard or a whiteboard. Each group room had a number of felt pens along with post-it notes ready to be used by the students. The outcome of the PMA from each group room was two affinity diagrams with comments (positive and negative) and two root cause diagrams with comments (positive and negative). During the PMA, instructors were available in the group rooms for questions and guidance.

3 The Method

One of the main problems in this work was to collect and analyse the students' responses in an efficient manner. Since we have long experience in using methods from the TQM in collecting information during industrial software process improvement - SPI - activities, we decided to let each student group perform a post mortem analysis - PMA - of their project.

For a further discussion of PMA and its use in SPI, see for instance [6, 3]. The PMA, as we use it, consists of two traditional TQM methods - affinity diagrams - the KJ method - and root cause analysis - RCA. For a more complete description of these two methods, see [2]. The following description is just meant to be a quick introduction to these two important methods.

Before the post mortem started, all students got a short introduction - 20 minutes - to the two methods that they should use. Judged from the results, the introductions were a bit too short even though most of the teams did a good job.

3.1 The Affinity Diagram

The affinity diagram is also called the KJ method in honour of its inventor, the Japanese ethnographer Kawakita Jiro [4]. Broadly stated - the KJ process is a special way to perform a brainstorming. It has two points that make it more efficient than a traditional brainstorming. First, all participants get the same amount of attention. Second, the analysis of the input is done by the participants in course of the meeting.

Even though the affinity diagram in most cases is used as a problem solving method, it works equally well applied to successes. We have noted that many persons seem to forget this and turn the activity into a soul searching "mea culpa" session. In order to give the students a more optimistic view on their work and the course the groups had to analyse both positive and negative events and relations in their architectural project.

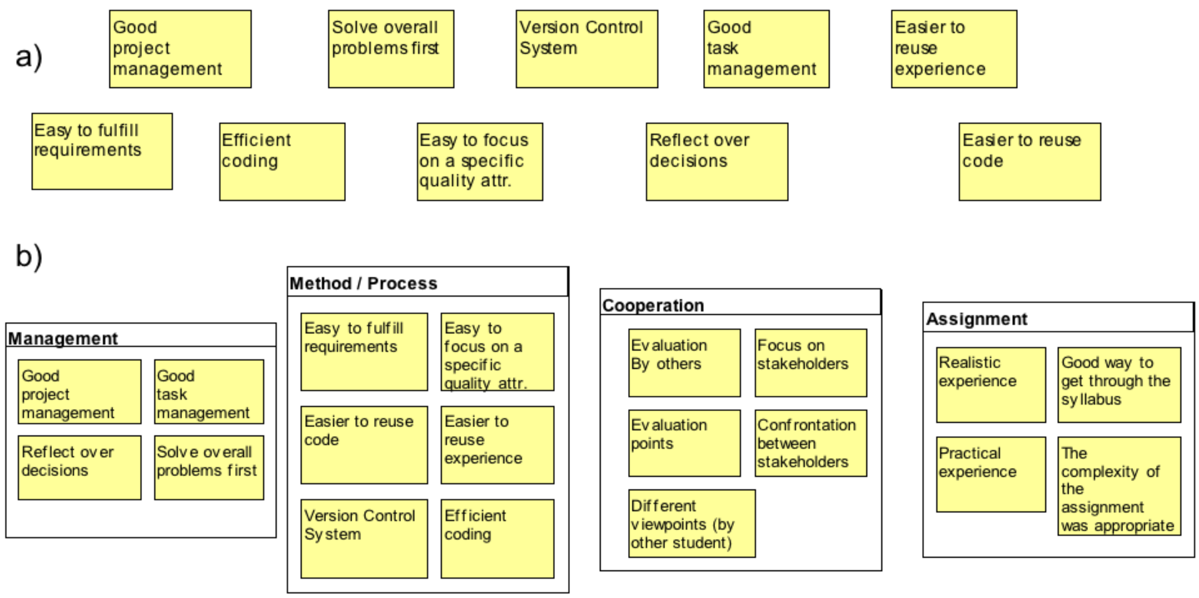

Figure 1. Illustration of students post-its on a whiteboard

In the version of the method used here and in most practical SPI work, the following process is used to develop the KJ diagram:

1. State the topic of the KJ. This might be broad, like what are the most important lessons learned from this project, or quite focused, like what are the main causes for cost overruns in this project.

2. The participants pair up to discuss the topic and may be write down some important points that surface during this discussions. This is mainly done in order to kick-start the next step in the process which is where the real generation of ideas take place.
3. Each participant write down ideas, comments etc. on post-it notes. The rules are one idea per post-it note. Ideally, each idea should be stated in a sentence that, as a minimum, has both a verb and a subject. As will be seen from the example shown below, this rule is difficult to enforce. In our experience it is practical to set an upper limit on the number of post-it notes each participant can produce.
4. Each participant put his post-its on a whiteboard and gives a short explanation of why this particular idea is brought up (see Figure 1a).
5. When all post-its have been placed on the whiteboard, all the participants go to the whiteboard and start organizing the notes. This is free-for-all process - there is no such thing as "my post-its". In most cases this process converges fairly quickly. The goal of this step is to create groups of post-its that have a common theme. The participants then give names to the groups and draw connections between them showing relationships and influences (see Figure 1b).
6. The last step in the process is to assign a priority or importance to each group. Since most of the student teams generated just a few groups, this step was left out in this case.

Five of the thirteen student teams performed a KJ analysis - most of them have been done according to the rules.

3.2 The Root Cause Analysis

The RCA was invented by the Japanese engineer Ishikawa and is thus also called the Ishikawa diagram [5]. For obvious reasons it is also called the fish bone diagram.

There are two standard approaches to an RCA - just starting with the problem and starting with a set of predefined main causes. The diagram below shows a RCA diagram with a set of standard causes included. Typical standard causes are for instance personnel, equipment, methods and material. Our experience that using a partly ready-made diagram inhibit the creativity of the analysis process. Thus, as a rule, we start the process just by drawing a horizontal line and the problem statement.

The process of building the RCA diagram starts with the question "What are the main causes for the problem?" These causes are them inserted on the main lines in the diagram. The process is then repeated for each main cause by asking "What are the causes for this main cause?" Figure 2 shows a RCA diagram with causes inserted.

The process is repeated - first for all the main causes and then for the sub-causes. The process can be repeated as many times as one like but practical experience shows that there seldom makes any sense to go below level three of causes.

Figure 3 two Ishikawa diagrams - one for a negative issue (project course management) and one for a positive issue (good process). The main method in the project - ATAM - is included in both diagrams.

Eight of the thirteen student teams performed an RCA - most of them were done according to the rules. Of the eight teams, four had a positive issue as starting point and four had a negative one.

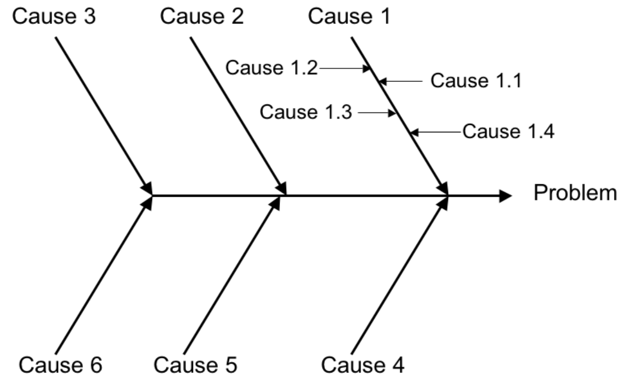

Figure 2. RCA diagram with sub-causes for main cause 1 filled in

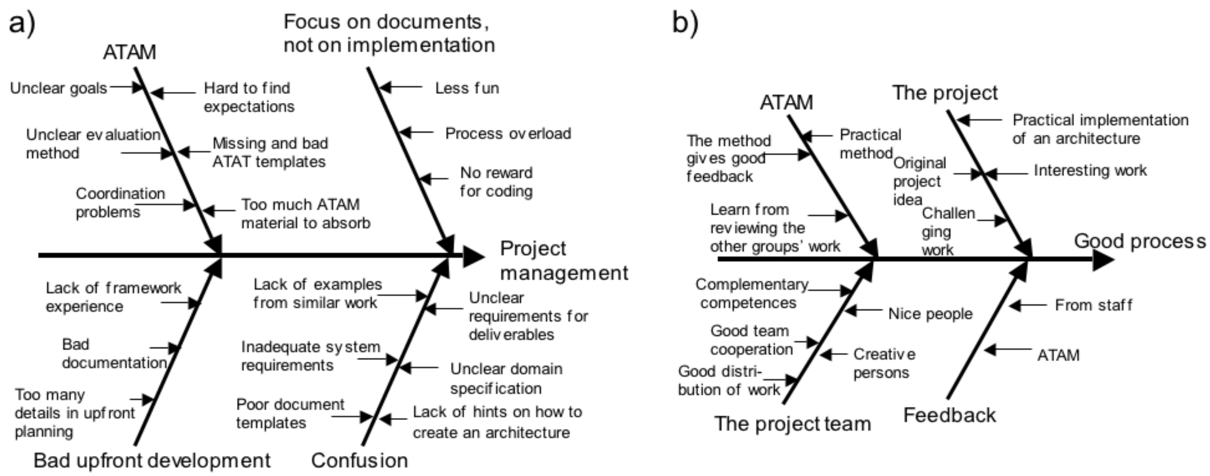

Figure 3. Ishikawa diagrams for bad project management and for good process

4 Presentation of Data and Suggested Improvements

In order to get an overview over the good and bad points of the course, as perceived by the students, we summed up all issues mentioned in the PMAs and the RCAs. A total of 13 groups participated in the event - three of the groups analysed both positive and negative issues, while we had five groups analysing only positive issues and five groups analysing only negative issues. This gives us a total of eight groups on each side. We decided that all issues brought up by at least half the groups (four groups or more) should be included in the analysis. Table 1 show the issues identified by the PMA (negative issues are numbered 1-7).

Based on the outcomes of the PMA, the following prioritised actions are suggested to improve the course for next year (sorted by order of negative issues with most votes):

1. **Improve the provided project delivery templates:** To make the delivery documentation templates comply with the standards of architectural documentation and to avoid misunderstandings these templates must be revised.
2. **Improve the assignment text to be easier to comprehend:** The PM-analysis revealed that the assign text could be unclear and difficult to understand. We there-

Negative issues	
Votes	**Issue**
8	1. Variation in document templates, Too much focus on document problems
6	2. Not good enought description of the exercise, unclear spesification
6	3. Too much (implementation) work - implementation suffered, Not enough time
4	4. Robot sensor problems
4	5. Bad simulator - badly documented or did not work
4	6. Bad coordination, bad communication with teaching staff
4	7. ATAM did not work, too few stakeholders, ATAM hard to use, too litle ATAM introduction
Positive issues	
Votes	**Issue**
5	Learned a lot about architecture, The architecture worked
5	Feedback on the architecture made a better architecture (ATAM)
3	Everybody did their share of the work
3	Good communication, good feedback

Table 1. Results from PMA

fore suggest making a revised version of the assignment text proofread by a person external to the course.

3. **Provide examples and libraries for the robot simulator:** Since much of the frustration of the project is related to difficulties implementing a robot controller, we suggest creating code examples and libraries that help students to faster and easier implement a useful robot controller. The main focus of the course is on creating and implementing a software architecture and not dealing with robot simulator problems.
4. **Require a small prototype implementation to be a part of the first delivery:** To force the students to understand the robot simulator and environment before designing the architecture and to avoid having the implementation started in the last week before final delivery, we propose to add a simple implementation exercise to be a part of the first delivery. This exercise should involve getting the robot controller to do simple actions.
5. **Improve the documentation of the robot simulator:** From the PM-analysis we found that the lack of documentation of the simulator environment is a very severe problem in order to complete the project. This problem has caused the students to create an architecture that does not fit the simulator environment and not to take the characteristics of the simulator environment in to account at an early stage of the project.
6. **Assign time slots and group rooms for performing ATAM:** The PMA revealed that coordination of two groups to perform the ATAM was difficult. The course staff can simplify this coordination by assigning time slots and group rooms for the involved groups. As a bonus, fixed time slots make it possible for the course staff to guide the students in the ATAM process.
7. **Tailor ATAM to fit the size of the project:** A problem of many software engineering methods is that they do not fit very well into smaller projects causing such methods to be tiresome and overkill to use. To avoid this problem we suggest down-

sizing ATAM to fit the size of the robot project to make it an efficient tool for an evaluation of the software architecture. It is important that the students get a positive attitude towards ATAM so they can use it later in software companies in Norway that usually are small companies.

One of the important things when improving a process - in this case a course - is not to destroy the positive effects observed. In our case, the positive effects were to a large degree related to social factors - good communication; everybody did their share - and the ATAM process itself. None of the changes mentioned above will jeopardise this. On the other hand it might be possible to increase the positive effects, for instance by improving group dynamic factors, i.e. cooperation and communication. One of the things that we could do is to focus more on the importance of the communication and feedback aspects of ATAM. The suggestions 6 and 7 above should improve the communication and feedback aspects of ATAM. In a technically focused environment like NTNU, it is all too easy to forget or downplay the social side, such as the communication part of a method.

5 Evaluation of Method and Conclusion

To get an impression of the students' attitude towards the PMA, they were asked to briefly describe their experience with the method in a report. In addition, the teacher was visiting the group rooms and asking the students about how they were doing. One problem that was discovered was that it was difficult to get students from different groups to do a PMA together. Although the group projects had the same goals, each group meant that their project experience was unique. However, after some discussion, the different groups in the same group room found that they had much in common. The students that were assigned to analyse the positive aspects of their project had a new experience. In prior projects they have always only looked at the negative aspects about the project in an evaluation, and it was therefore challenging and useful to focus on the positive aspects. Some students also said that the PMA was the first time they have done a proper analysis of a project after they had delivered the product. Further, that such evaluation would help in new projects to improve the group performance. In one group room, we had a mixture of foreign and Norwegian students. This caused some problems in performing the PMA. The main problem was for the Norwegian students to express themselves naturally in English and to discuss orally in English. An interesting observation the development of KJ-diagrams was the hesitation of the students to start organizing the others' notes. Usually, they asked the creator of the note politely before moving it on the whiteboard. After some time, the notes could be moved without asking.

Considering the short introduction the students got to the two methods, most of the groups did fairly well. Neither method is complicated to understand - as a matter of fact, for TQM it is important that the methods are simple to use so that they do not exclude anybody from the analysis and decision processes. We have noted the following problems when using the two methods:

- RCA - the chosen goal or problem was at too high a level, resulting in a lot of rather trivial causes. This was the case for all the teams dealing with positive issues and for one of the teams dealing with negative issues. For students as for people working in industry, it seems to be a problem to identify positive issues. Too much of our courses focus on problems and problem solving.

- KJ - no indication of connections between groups and no priorities assigned to the groups. In addition, one group mixed positive and negative issues in the same diagram.

Since the RCA starts with a defined structure - the fish bone - and a process - ask "why did this occur" - in order to get to the next level, this is hardly surprising. It is also consistent with what we have observed in when applying the two methods in industrial settings. As is the case in industry all the problems that occurred would have been prevented if each group had been supervised during the process, for instance by a teaching assistant that had been trained in the two methods.

When using the PMA in an industrial process improvement process, we have used first the KJ to get an overview of the problems and opportunities and then the RCA to analyse the two most important ones. In this case, the limited amount of time available forces us to use a simpler version where five teams only did a KJ while eight teams only did the RCA. The teams that did only the RCA had problems finding an interesting and important starting cause. This problem will not occur in an industrial setting since the goal is selected during the KJ part of the PMA.

All in all, the PMA was a positive experience for everybody involved and it is a practical way to assess a course. There is, however, always room for improvement. In our case, the following improvements should be done:

- A teaching assistant, trained in PMAs should be assigned to each team or group of teams. His job should be to see that the process is followed and, if necessary guide the team back on the right track.
- The students should be made aware of the most important and common pitfalls and how they can be avoided, before they start on the PMA.
- If at all possible, all teams should do both the KJ - to identify all areas of interest - and the RCA - in order to look at one or two issues in depth.

References

[1] Len Bass, Paul Clements, and Rick Kazman. *Software Architecture in Practice - Second Edition.* Sei series in Software Engineering. Addision-Wesley, 2003.

[2] Andreas Birk, Torgeir Dingsøyr, and Tor Stålhane. Postmortem: Never leave a project without it. *IEEE Software*, 19(3):43–45, May/June 2001.

[3] Torgeir Dingsøyr, Nils Brede Moe, and Øystein Nytrø. Augmenting Experience Reports with Lightweight Postmortem Reviews. In *3rd Int'l Conf. on Process Focused Software Process Improvement (PROFES2001)*, Kaiserslautern, Germany, 10–13 September 2001.

[4] Raymond Scupin. The KJ Method: A Technique for Analyzing Data Derived from Japanese Ethnology. *Human Organization*, 56(2):233–237, 1997.

[5] David Straker. *A Toolbook for Quality Improvement and Problem Solving.* Pretence Hall International (UK) Limited, 1995.

[6] Tor Stålhane and Nils Brede Moe. Post Mortem - An Assessment of Two Approaches. In *European Software Process Improvement Conference (EuroSPI2001)*, Limerick, Ireland, 10–12 October 2001.

[7] Wright State University. WSU Khepera Simulator. Web: http://ehrg.cs.wright.edu/ksim/ksim.html, 2003.