

Copyright ©2002 IEEE. Reprinted from IEEE Software May/June 2002.

This material is posted here with permission of the IEEE. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to [pubs-permissions@ieee.org](mailto:pubs-permissions@ieee.org).

By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

# Postmortem: Never Leave a Project without It

Andreas Birk, *sd&m*

Torgeir Dingsøy, *Sintef Telecom and Informatics*

Tor Stålhane, *Norwegian University of Science and Technology*

**I**n every software project, the team members gain new knowledge and experience that can benefit future projects and each member's own professional development. Unfortunately, much of this knowledge remains unnoticed and is never shared between individuals or teams.

Our experience with project *postmortem analysis* proves that it is an excellent method for knowledge management,<sup>1</sup> which captures experience and

improvement suggestions from completed projects and works even in small- and medium- size companies that cannot afford extensive KM investments. However, PMA has been mainly advocated for situations such as completion of large projects, learning from success, or recovering from failure.<sup>2-4</sup>

When used appropriately, PMA ensures that team members recognize and remember what they learned during a project. Individuals share their experiences with the team and communicate them to other project groups. Additionally, PMA identifies improvement opportunities and provides a means to initiate sustained change.

We have applied a lightweight approach to PMA in several projects<sup>5,6</sup> by focusing on a few vital principles:

- PMA should be open for participation from the entire team and other project stakeholders.
- Goals can—but need not—provide a focus for analysis.
- The PMA process comprises three phases: preparation, data collection, and analysis. For each phase, team members

can apply a number of fairly simple methods, such as the KJ method (after Japanese ethnologist Jiro Kawakita)<sup>7</sup> that collects and structures the data from a group of people.

## Preparation

When we conduct PMA in software companies, two software process improvement group members work as facilitators together with two to all project team members. Facilitators organize the analysis, steer the discussion, and document the results. They can be employees in the company where the PMA is conducted or external, as we are. External facilitators often have an advantage performing the PMA because participants regard them as more neutral and objective. However, they might not know the company as well as internal facilitators, so preparation is important.

During the preparation phase, we walk through the project history to better understand what has happened. We review all available documents, such as the work breakdown structure, project plans, review reports, and project reports.

We also determine a goal for the PMA.

Although primarily used for large projects and companies, postmortem analysis also offers a quick and simple way to initiate knowledge management in small- or medium-size software projects.

**Once the group identifies the important topics, we must prioritize them before proceeding with the analysis.**

Goals might be “Identify major project achievements and further improvement opportunities” or “Develop recommendations for better schedule adherence.” If a PMA does not have a specific focus to guide our preparation, we briefly discuss the project with the project manager and key engineers.

We find it practical to distinguish between two PMA types: One is a general PMA that collects all available experience from an activity. The other is a focused PMA for understanding and improving a project’s specific activity, such as cost estimation. It helps to explicitly state goals for both of these PMA variants during this phase.

### **Data collection**

In the data collection phase, we gather the relevant project experience. Usually, project team members and stakeholders have a group discussion, or experience-gathering session. We can often conduct data collection and the subsequent analysis within the same session. You shouldn’t limit experience gathering to the project’s negative aspects, such as things to avoid in the future. Instead, maintain a balance by identifying a project’s successful aspects, such as recommended practices. For example, during a PMA at a medical software company, the team realized that the new incremental software integration process significantly improved process control and product quality. Integration had been so smooth that without the PMA, its important role might have gone unnoticed.

Some techniques that we find useful for data collection include

- *Semistructured interviews.* The facilitator prepares a list of questions, such as “What characterizes the work packages that you estimated correctly?” and “Why did we get so many changes to the work in package X?”
- *Facilitated group discussions.* The facilitator leads and focuses the discussion while documenting the main results on a whiteboard.
- *KJ sessions.* The participants write down up to four positive and negative project experiences on post-it notes. Then they present their issues and put the notes on a whiteboard. The participants rearrange all notes into groups according to topic and discuss them.

Once the group identifies the important topics, we must prioritize them before proceeding with the analysis. This will ensure that we address the most significant issues first.

For example, during a PMA we performed in a satellite software company, frequent and late requirements changes emerged as an important topic. A software developer commented that during the project, team members found it difficult to identify when the requirements had changed, so much so that the code had to be rewritten completely. In such situations, they made a few wrong decisions, which reduced the software’s quality. After this PMA session, other project members made requirements changes a high-priority topic for analysis.

### **Analysis**

In this phase, as facilitators, we conduct a feedback session in which we ask the PMA participants: “Have we understood what you told us, and do we have all the relevant facts?”

When we know that we have sufficient and reliable data, we use Ishikawa diagrams<sup>6</sup> in a collaborative process to find the causes for positive and negative experiences. We draw an arrow on a whiteboard, which we label with an experience. Then, we add arrows with causes—which creates a diagram looking like a fishbone. In our example from the satellite software company, we found four causes for changing requirements: poor customer requirements specification, new requirements emerging during the project, little contact between the customer and software company, and the software company’s poor management of requirements documents.

Because PMA participants are a project’s real experts and we have time limitations, we perform all analysis in this step.

### **Results and experience**

Facilitators document the PMA results in a project experience report. The report contains

- A project description, including products developed, development methods used, and time and effort needed
- The project’s main problems, with descriptions and Ishikawa diagrams to show causes
- The project’s main successes, with descriptions and Ishikawa diagrams

## About the Authors

- A PMA meeting transcript as an appendix, to let readers see how the team discussed problems and successes

In an example from the satellite software company, facilitators wrote a 15-page report in which they documented the problem with changing requirements with an Ishikawa diagram that showed the four main causes. After facilitators submit a report, the knowledge management or quality department must follow up.

In our experience, PMA is suitable when a project reaches a milestone and when the company is looking for qualitative experience that will help improve a similar, future project. You should not apply PMA in situations with unfinished activities, or when serious underlying conflicts might remove the focus from improvement. If the atmosphere isn't appropriate for discussing a project's problems, we prefer using approaches other than PMA, such as those outlined in *Project Retrospectives: A Handbook for Team Reviews*.<sup>2</sup> When there have been serious conflicts in the project, this is more appropriate for managing the risk that discussions degenerate into a hunt for scapegoats. Finally, you must have enough time for following up on PMA results.

In our experience, if teams apply PMA in the right setting, it is an excellent step into continuous knowledge management and improvement activities. It makes project team members share and understand one another's perspectives, integrates individual and team learning, and illuminates hidden conflicts. It documents good practice and problems, and finally, it increases job satisfaction by giving people feedback about their work.

Performing a PMA can even improve project cost estimation. We applied PMA to three projects in an Internet software development company, which all had serious cost overruns. The company could not allocate workers with skills specific to the project. This led to a need for courses—the team's experts had to act as tutors for the rest of the team and were distracted from their roles in the project. By performing the PMA, the company realized the gravity of the qualification issue and how it led to the project going over budget. As an improvement action, a training budget was set up on the company level instead of the project level. The company no longer charged staff qualification to the pro-



**Andreas Birk** is a consultant and software engineering professional at sd&m, software design and management. His special interests include software engineering methods, knowledge management, and software process improvement. He holds a Dr.-Ing. in software engineering and a Dipl.-Inform. in computer science and economics from the University of Kaiserslautern, Germany. He is a member of the IEEE Computer Society, ACM, and German Computer Society. Contact him at sd&m, Industriestraße 5, D-70565 Stuttgart, Germany; andreas.birk@sdm.de.

**Torgeir Dingsøy** is a research scientist at Sintef Telecom and Informatics research foundation in Trondheim, Norway. He wrote his doctoral thesis on "Knowledge Management in Medium-Sized Software Consulting Companies" at the Department of Computer and Information Science, Norwegian University of Science and Technology. Contact him at Sintef Telecom and Informatics, SP Andersens vei 15, NO-7465 Trondheim, Norway; torgeir.dingsoyr@sintef.no.



**Tor Stålhane** is a full professor of software engineering at the Norwegian University of Science and Technology. He has a MSc in electronics, and a PhD in applied statistics from Norwegian University of Science and Technology. He has worked on compiler development and maintenance and software reliability, and on software process improvement and systems safety. Contact him at Department of Computer and Information Science, Norwegian University of Science and Technology, NO-7491 Trondheim, Norway; tor.stalhane@idi.ntnu.no.

ject's budget, and now views it as an investment into quality and competitive advantage. As a result of this PMA, management realized the strategic importance of staff qualification and knowledge management—a truth that often gets buried in the hectic rush of Internet software business.

**W**e received a lot of positive feedback from PMA participants in different companies. Particularly, they like that PMA offers a simple yet effective way to uncover both achievements and improvement opportunities. One developer at the satellite software company noted, "If you do a PMA on the project...you have to think through things," which is a crucial part of knowledge management. So, never leave a project without it! 🍷

## References

1. C. Collison and G. Parcell, *Learning to Fly: Practical Lessons from One of the World's Leading Knowledge Companies*, Capstone, New York, 2001.
2. B. Collier, T. DeMarco, and P. Fearey, "A Defined Process For Project Post Mortem Review," *IEEE Software*, vol. 13, no. 4, July/Aug. 1996, pp. 65–72.
3. N.L. Kerth, *Project Retrospectives: A Handbook for Team Reviews*, Dorset House Publishing, New York, 2001.
4. A.J. Nolan, "Learning from Success," *IEEE Software*, vol. 16 no. 1, Jan./Feb. 1999, pp. 97–105.
5. T. Stålhane et al., "Post Mortem—An Assessment of Two Approaches," *Proc. European Software Process Improvement (EuroSPI 01)*, ICSN, Bray, Ireland.
6. T. Dingsøy, N.B. Moe, and Ø. Nytrø, "Augmenting Experience Reports with Lightweight Postmortem Reviews," *3rd Int'l Conf. Product Focused Software Process Improvement (Profes 01)*, Lecture Notes in Computer Science, vol. 2188, Springer-Verlag, Berlin, pp. 167–181.
7. D. Straker, *A Toolbook for Quality Improvement and Problem Solving*, Prentice Hall International, London, 1995, pp. 89–98 and 117–124.