

Postmortem reviews: purpose and approaches in software engineering

Torgeir Dingsøy^{*}

SINTEF Information and Communication Technology, SP Andersens vei 15b, 7465 Trondheim, Norway

Received 21 June 2004; revised 17 August 2004; accepted 25 August 2004

Abstract

Conducting postmortems is a simple and practical method for organisational learning. Yet, not many companies have implemented such practices, and in a survey, few expressed satisfaction with how postmortems were conducted. In this article, we discuss the importance of postmortem reviews as a method for knowledge sharing in software projects, and give an overview of known such processes in the field of software engineering. In particular, we present three lightweight methods for conducting postmortems found in the literature, and discuss what criteria companies should use in defining their way of conducting postmortems.

© 2004 Elsevier B.V. All rights reserved.

Keywords: Postmortem review; Knowledge management; Software engineering

1. Introduction

Knowledge management has received much attention in the software engineering field during the past years, partly as a promising field for software process improvement in order to increase quality and decrease costs in software development.

Software process improvement has its roots in general improvement philosophies like total quality management, which has been tailored to software engineering in the Quality Improvement Paradigm [1], and in efforts on standardisation like the ISO 9001 and the Software Engineering Institute's Capability Maturity Model [2].

A common factor in knowledge management and in software process improvement is to learn from past successes and failures in order to improve future software development. Experience Factory [3] has been a central term in focusing organisational learning on improving software development processes.

Most companies that develop software organize the development in projects. In the Experience Factory, the projects are seen as the main arena for learning, and

experience which appears in the projects is to be shared with other projects.

In this article, we will discuss practical methods to harvest experience from projects that are either completed or have finished a major activity or phase. We refer to these methods as 'postmortem reviews' as this is a common term.

The main objective of this article is to highlight the importance of group-processes as a method for knowledge sharing in software projects, and to give an overview of known such processes in the field of software engineering.

In the rest of this article we will discuss some fundamental issues in knowledge management and learning, and then present work on postmortems from the literature. We further present three processes for conducting postmortems in detail, as well as an example of results from one postmortem review using one of the methods. We discuss what such processes should contain: What are requirements for a good postmortem process, who should be invited to a postmortem meeting, should the postmortem involve homework for participants, what should be the role for the facilitator, should the discussions be open or structured, should management participate in the meeting and should the postmortem focus on tacit or explicit knowledge? Finally, we discuss how the processes relate to fundamental issues in knowledge management and learning.

^{*} Tel.: +47 73 59 29 79; fax: +47 73 59 29 77.

E-mail address: torgeir.dingsoyr@sintef.no.

1.1. Knowledge management

A recent improvement trend has been knowledge management, which is related to creating ‘learning organisations’, in software engineering: ‘learning software organisations’.

A learning organisation is ‘an organisation skilled at creating, acquiring, and transferring knowledge, and at modifying its behaviour to reflect new knowledge and insight’ [4]. George Huber gives some advice on what managers can do to make their organisations more ‘learning’ [5]:

- Learn from experience—systematically capture, store, interpret and distribute relevant experience gathered from projects; and also to investigate new ideas by carrying out experiments.
- Using a computer-based organisational memory—to capture knowledge obtained from experts to spread it through the organisation.

A research area that is linked to organisational learning is research on ‘communities of practise’ as a basis for learning. Etienne Wenger writes: “learning is an issue of sustaining the interconnected communities of practise through which an organization knows what it knows” [6].

In the much-cited book on learning organisations, *The Fifth Discipline* [7], we find further characteristics of learning organisations: the ability of ‘systems thinking’—to see more than just parts of a system. This often means to involve people in an organisation to develop a ‘shared vision’, some common grounds that make the work meaningful, and also serve to explain aspects that you yourself do not have hands-on experience in. Another way of improving communication in an organisation is to work on ‘mental models’ that support action, ‘personal mastery’; that people make use of their creativity and abilities. And finally, ‘group learning’, to enhance dialogue and openness in the organisation.

1.2. Learning

The process of transferring knowledge between people is usually referred to as ‘learning’. Webster’s [8] defines learning as ‘to acquire knowledge of or skill in by study, instruction, or experience, to become informed of or acquainted with’ or ‘to memorize’. In organisational literature, it is often defined as a ‘purposefully change of action’.

What does it mean to say that an organisation as a whole learns? This differs from individual learning in two respects [9]: first, it occurs through shared insight, knowledge and shared models. Second, it is not only based on the memory of the participants in the organisation, but also on ‘institutional mechanisms’ like policies, strategies, explicit models and defined processes. We can call this the ‘culture’

of the organisation. These mechanisms may change over time, which is a form of learning.

Argyris and Schön distinguish between what they call single and double-loop learning [10] in organisations. Single-loop learning implies a better understanding of how to change (or ‘tune’), say a process, to remove an error from a product. It is a single feedback—loop from observed effects to making some changes (refinements) that influence the effects. Double loop learning, on the other hand, is when you understand the factors that influence the effects, and the nature of this influence, what they call the ‘governing values’ [11].

In software engineering, a ‘learning software organisation’ has been defined as an organisation that has to ‘create a culture that promotes continuous learning and fosters the exchange of experience’ [12]. Dybå puts more emphasis on action in his definition: “A software organisation that promotes improved actions through better knowledge and understanding” [13].

In the following sections, we will present two models from the literature on how knowledge is transferred between individuals in organisations, what we can describe as ‘learning’ on an individual level, and ‘organizational learning’ for a community. We do not aim to cover the whole range of theories of learning, but will focus on two approaches that we consider interesting, and has been used in the knowledge management field, namely:

- Learning through participation: communities of practise.
- Learning as a conversion process between tacit and explicit knowledge.

1.2.1. Learning through participation: communities of practise

The traditional view of learning has been that it best takes place in a setting where you isolate and abstract knowledge and then ‘teach’ it to ‘students’ in rooms free of context. Etienne Wenger describes this view of learning as an individual process where for example collaboration is considered a kind of cheating [6]. In his book about communities of practise, he describes a completely different view: learning as a social phenomenon. A community of practise develops its own ‘practises, routines, rituals, artifacts, symbols, conventions, stories and histories’. This is often different from what you find in work instructions, manuals and the like. In this context, Wenger defines learning as:

- For individuals: learning takes place in engaging in and contributing to a community.
- For communities: learning is to refine the practise.
- For organisations: learning is to sustain interconnected communities of practise.

We find communities of practise everywhere: at work, at home, in volunteer work. And it can be a challenge to

sustain such networks of people, for example in turbulent organisations that undergo reorganisation processes.

The work on communities of practise is closely linked to work on situated learning [14].

1.2.2. Learning as a conversion process between tacit and explicit knowledge

In the much-cited book ‘The Knowledge-Creating Company’, where Nonaka and Takeuchi explains the success of Japanese companies by their effort at ‘organizational knowledge creation’. They also offer a model of how knowledge is transformed and converted in an organisation [15].

When we discussed the word ‘knowledge’, we divided between tacit and explicit knowledge. Nonaka and Takeuchi claims that knowledge is constantly converted from tacit to explicit and back again as it passes through an organisation. They say that knowledge can be converted from tacit to tacit, from tacit to explicit, or from explicit to either tacit or explicit knowledge as shown in Fig. 1.

We now describe each of these four modes of conversion:

- Socialization means to transfer tacit knowledge to tacit through observation, imitation and practice, what has been referred to as ‘on the job’ training. Craftsmanship has usually been learned in this way, where oral communication is either not used or just plays a minor part.
- Internalisation is to take externalised knowledge and convert it into individual tacit knowledge in the form of mental models or technical know-how. ‘Documents and manuals facilitate the transfer of explicit knowledge to other people, thereby helping them experience the experiences of others indirectly (i.e. ‘re-experience’ them)’.
- Externalisation means to go from tacit to explicit knowledge. Explicit knowledge can ‘take the shapes of metaphors, analogies, concepts, hypotheses or models’. This conversion is usually triggered by dialogue or collective reflection, but can also be the result of individual reflection, for example in a writing process.
- Combination is to go from explicit to explicit knowledge, that is, to combine and systemize knowledge from different sources such as documents, meetings, telephone conferences and bulletin boards. Systematising this kind

of explicit knowledge is to reconfigure it by sorting, adding, combining or categorising the knowledge.

According to Nonaka and Takeuchi, knowledge passes through different modes of conversion in a spiral which makes the knowledge more refined, and also spreads it across different layers in an organisation. Hansen et al. [16] discusses two strategies for knowledge management, one relying on codification, the other relying on sharing tacit knowledge, what they call personalisation.

1.3. The project as a learning arena

In software engineering, to reuse life cycle experience, processes and products for software development is often referred to as having an ‘Experience Factory’ [3]—a separate organisational entity with responsibility for capturing and reusing experience. This approach has been much cited in the software engineering field. Experience is collected from software development projects, and packaged and stored in an experience base. By packaging, we mean generalising, tailoring and formalising experience so that it is easy to reuse.

The Experience Factory organisation assists software developing projects with earlier experience both in upstart and during execution, and can suggest improvements in the development processes, based on collected experience.

2. Postmortem reviews

We first define what we mean by a ‘postmortem’. Then, we describe postmortem reviews from the software engineering literature, before presenting some methods for conducting postmortem reviews.

2.1. What is a ‘postmortem’?

By a postmortem, we mean a collective learning activity which can be organised for projects either when they end a phase or are terminated. The main motivation is to reflect on what happened in the project in order to improve future practise—for the individuals that have participated in the project and for the organisation as a whole. The physical outcome of a meeting is a postmortem report.

This type of processes has also been referred to as ‘project retrospectives’, ‘post mortem analysis’,



Fig. 1. Conversion of knowledge according to Nonaka and Takeuchi. We can imagine knowledge going through all conversion processes in a spiral form as it develops in an organisation.

‘postproject review’, ‘project analysis review’, ‘quality improvement review’, ‘autopsy review’, ‘Santayana review’, ‘after action reviews’ and ‘touch-down meetings’.

Researchers in organisational learning sometimes use the term ‘reflective practice’, which can be defined as ‘the practice of periodically stepping back to ponder on the meaning to self and others in one’s immediate environment about what has recently transpired. It illuminates what has been experienced by both self and others, providing a basis for future action’ [17]. This involves uncovering and making explicit results of planning, observation and achieved practice. It can lead to understanding of experiences that have been overlooked in practice.

The two theories of learning that we presented in Section 1.2 put different emphasis on this kind of learning. In the model of Nonaka and Takeuchi, postmortems are a combination of learning through socialisation and through externalisation. In listening to others you employ socialisation and in reflecting and sharing your own experience you externalise your tacit knowledge. Postmortems are also a method for leveraging knowledge from the individual level to the organisational level.

In a community of practise view, postmortems are an arena for the individual to contribute with knowledge to the community, and also for the community to discuss changes of practise on key areas.

In a survey on essential practises in research and development-companies, ‘learning from post-project audits’ are seen as one of the most promising practises that could yield competitive advantage [18].

A survey on post-project reviews in research and development companies show that only one out of five projects received a post-project review [19]. Also, the reviews tend to focus on technical output and bureaucratic measurements. Process-related factors are rarely discussed.

As a knowledge management and software process improvement tool, postmortem reviews are simple to organise. The process focuses on dialogue and discussion which is a central element in knowledge transfer. Von Krogh et al. writes that “it is quite ironic that while executives and knowledge officers persist in focusing on expensive information-technology systems, quantifiable databases, and measurement tools, one of the best means for knowledge sharing and creating knowledge already exists within their companies. We cannot emphasize enough the important part conversations play” [20].

An example of postmortem reviews are ‘after action reviews’ conducted by the US army since after the Vietnam war, focusing on a ‘professional discussion of an event’ to provide insight, feedback and details about the event [21].

Kransdorff [22] criticizes postmortems because people participating do not have an accurate memory, which can lead to disputes. He suggests collecting data during the project, for example through short interviews, in an effort to get more objective material.

2.2. Postmortem reviews in software engineering

There are several ways to perform Postmortem Reviews. Apple has used a method [23] which includes designing a project survey, collecting objective project information, conducting a debriefing meeting, a ‘project history day’ and finally publishing the results. At Microsoft they also put much effort into writing ‘Postmortem reports’. These contain discussion on ‘what worked well in the last project, what did not work well, and what the group should do to improve in the next project’ [24]. The size of the resulting document is quite large, ‘groups generally take 3–6 months to put a postmortem document together. The documents have ranged from under 10 to more than 100 pages, and have tended to grow in length’.

In a book about team software development, Watts Humphrey suggests a way to do postmortems to ‘learn what went right and wrong, and to see how to do the job better the next time’ [25].

A description of another lightweight approach which seeks to elicit experience using interviews, and not a group process, is described by Schneider [26].

Norman Kerth lists a total of 19 techniques to be used in postmortems [27], some focusing on creating an atmosphere for discussion in the project, some for reviewing the past project, some for helping a team identify and embrace change during their next project, and some for dealing with the unique effects of a failed project. Kerth recommends using three days in order to effect a lasting change in the company.

Tiedeman [28] suggests three types of postmortems, related to a waterfall model of software development, one for ‘planning’, one for ‘design/verification’ and one ‘field postmortem’ to provide feedback after the developed system has been in use for some time.

The *Game Developer* magazine publishes postmortems on game development projects in most issues, see for example a postmortem on the game ‘Aggressive Inline’ [29]. The articles contain a brief description about the game developed and the project organisation, and then usually five issues that ‘went right’ and five issues that ‘went wrong’.

2.3. Methods for conducting postmortem reviews

We now present three methods for conducting postmortem reviews from the literature. We have selected three methods that can be performed in short time, and are thus suitable even for small and medium-size companies. They can also be a good start for companies wanting more in-depth methods later.

Neal Whitten suggests the following process for conducting postproject reviews [30]:

- (1) *Declare intent*—the project head should state his or her intention to have a postproject review at the completion

of the project, by a letter to all project participants. The letter should describe the postproject review process.

- (2) *Select participants*—participants from each major participating organisations should be selected: from planning, development, test, publications, performance, usability, module build group, etc. Managers should not participate in the postproject review team, as they are also evaluating the performance of people, and this might hinder topics from surfacing in the process.
- (3) *Prepare for workshop*—participants are asked to do homework before the workshop: to respond to a set of questions, like ‘What level of productivity was achieved for your tasks? How did it compare with what you expected?’. Many questions can be asked from various areas like staffing, mission objectives, education and training, tools, quality to support from outside groups.
- (4) *Conduct workshop*—the workshop can last from half a day to two days, and include: (a) 10–30 min presentations of feedback on the questions from each participant. (b) Construction of a things that ‘went right’ list with the most beneficial items placed at the top. (c) Construction of a ‘went wrong’ list in priority order. (d) Develop proposals that address the problems-either in groups or collectively.
- (5) *Present results*—results of the workshop are first presented to the project leadership. First and second level of the project leadership should at least be invited. Secondly, the results are presented to all participants in a meeting.
- (6) *Adopt recommendations*—a postproject review report is completed, which includes information from the workshop and recommendations from the project leadership. The report is either distributed to project leaders or to all personnel. The project leadership is responsible for acting on the committed recommendations.

Collison and Parcell [31] suggest the following steps for organising a retrospect meeting:

- (1) *Call the meeting*—hold the meeting as soon as possible after the project ends, and make the meeting a physical meeting rather than a videoconference.
- (2) *Invite the right people*—if a similar project is underway, invite the new project team also. The project leader needs to attend, as well as key members of the project. In the call to attendees, announce the purpose as to ‘make future projects run more smoothly, by identifying the learning points from this project’.
- (3) *Appoint a facilitator*—appoint one that is not closely involved in the project, but who is outside the line-management structure, as the meeting is to be clearly separate from any personal performance assessment.
- (4) *Revisit the objectives and deliverables of the project*—find the original criteria for success, and ask whether the project delivered these.

- (5) *Revisit the project plan or process*—in complex projects, it can be useful to construct a flow chart of what happened to identify tasks, deliverables and decision points.
- (6) *Ask ‘What went well’*—ask “what were the successful steps towards achieving your objective? What went really well in the project?”. Ask “why” several times to answers.
- (7) *Find out why these aspects went well, and express the learning as advice for the future*—identify the success factors and base future recommendations on agreed facts. The facilitator should press for specific, repeatable advice. The facilitator can either organize a conversation through probing questions, or identify issues and then work on each as a team.
- (8) *Ask “what could have gone better”?*—ask “what were the aspects that stopped you from delivering even more?”. Start by asking the project leader, then go round the room.
- (9) *Find out what the difficulties were*—identify stumbling blocks and pitfalls to be avoided in the future. Ask “given the information and knowledge we have today, what could we have done better?”
- (10) *Ensure that the participants leave the meeting with their feelings acknowledged*—ask people to rate the project: “looking back, how satisfied were you with this project, marks out of ten“. Follow up by asking “what could have made it a ten for you?”.
- (11) *‘What next’*—if the team is going straight into a new project, it is useful to follow the retrospect with a planning session for this.
- (12) *Recording the meeting*—a well-structured account of the meeting can contain: (a) guidelines for the future, (b) history from the project to illustrate the guidelines, (c) names of the people involved, for future reference, and (d) any key artifacts (documents, project plans). Use direct quotes to capture the depth of feeling and to create a summary that is easily read.

Birk et al. have used Postmortem Reviews as a group process [27,32–35], where most of the work is done in one meeting lasting half a day. They try to get as many as possible of the persons who have been working in the project to participate, together with two process consultants, one in charge of the Postmortem process, the other acting as a secretary. The goal of this meeting is to collect information from the participants, make them discuss the way the project was carried out, and also to analyse causes for why things worked out well or did not work out. A further description of this method can be found in the ‘results’ section.

The ‘requirements’ for this process is that it should not take much time for the project team to participate, and it should provide a forum for discussing most important experience from the project, together with an analysis of this experience. The main findings are documented in a report.

All participants in a project are invited to a half-day postmortem meeting without any requirements for preparation. Birk et al. use two techniques to carry out the Postmortem Review. For a focused brainstorm on what happened in the project, a technique named after a Japanese ethnologist, Jiro Kawakita [36]—called ‘the KJ Method’ is used. For each of these sessions, the participants are given a set of post-it notes, and asked to write one ‘issue’ on each note. Five notes are handed out to each person. After some minutes, the participants are asked to attach one note to a whiteboard and say why this issue was important. Then the next person presents a note and so on until all the notes are on the whiteboard. The notes are then grouped, and each group is given a new name.

Root Cause Analysis, also called Ishikawa or fishbone-diagrams are used to analyse the causes of important issues. The process leader draws an arrow on a whiteboard indicating the issue being discussed, and attach other arrows to this one like in a fishbone with issues the participants think are causing the first issue. Sometimes, also underlying reasons for some of the main causes are attached as well.

The postmortem meeting has following steps:

- (1) *Introduction*. First, the consultants introduced the agenda of the day and the purpose of the postmortem review.
- (2) *KJ session 1*. Consultants hand out post-it notes and ask people to write down what went well in the project, hear presentations, group the issues on the whiteboard, and give them priorities.
- (3) *KJ session 2*. Consultants hand out post-it notes and asked people to write down problems that appeared in the project, hear presentations, group the issues on the whiteboard, and give them priorities.
- (4) *Root cause analysis*. The process consultant leading the meeting draws fish-bone diagrams for the main issues, both from the things that went well and the things that were problematic.

Birk et al. use a tape recorder during the presentations, and transcribe everything that is said. The consultants write a postmortem report about the project, which contain an introduction, a short description of the project analysed, how the analysis was carried out, and the results of the analysis. The result is a prioritised list of problems and successes in the project. Statements from the meeting are used to present what was said about the issues with highest priority, together with a fishbone diagram to show their causes. In an appendix, everything that was written down on post-it notes during the KJ session is included, as well as a transcription of the presentation of the issues that were used on the post-it notes. Such reports are usually between 10 and 15 pages in length.

The day after the meeting, the consultants present the report to the people involved in the project to gather feedback and do minor corrections.

3. Case: postmortem in a medium-sized company

Above, we have seen different approaches to conducting postmortem reviews. In order to get a better understanding of such reviews, we now present results from one review. First, we present the company, then the project on where the review was carried out, and finally extracts from the postmortem report.

The case reported here was selected because of a wide data collection as a part of an action research [37] project on software process improvement. All written material from the postmortem meeting was photographed, and discussions were recorded on tape and transcribed. In the project, researchers and industry participants collectively discussed problems, identified possible solutions, tried out a solution and together reflected on the results.

3.1. A satellite software company

The company makes software and hardware for stations receiving data from meteorological and Earth observation satellites. Since the company was founded in 1984, they have delivered turnkey ground station systems, consultancy services, feasibility studies, system engineering, training, and support. The company has been working with large development projects, both as a prime contractor and as a subcontractor. The company possess a stable and highly skilled staff, many with master’s degrees in Computer Science, Mathematics or Physics, and have an ‘engineering culture’. Approximately 60 people are working in the company, and the majority is working with software development. Projects are managed in accordance with the European Space Agency PSS-05 standards, and are usually fixed price projects.

The company had problems with estimating the size of new software projects. Many people in the company also felt that they did not transfer enough experience between their software development projects. Every project wrote an ‘experience report’, but these were seldom considered interesting, and were not read very often. To improve this, the company decided to try postmortem reviews at the end of projects.

3.2. The project

We organised a postmortem in one project, which had developed a software system for a satellite that was recording environmental data. The project had developed a module that was to analyse data from this satellite, from European Space Agency specifications. This was a critical project for the company, as it was the first in a line of new services. The project lasted 36 months, and employed four people in the analysis phase, 8–12 people in the design phase, and 5–9 people in testing. The project spent a total of 47,000 work hours.

The five people in a core-team participated in the postmortem review, including the project manager. This was the first time the people in the project had participated in a postmortem meeting.

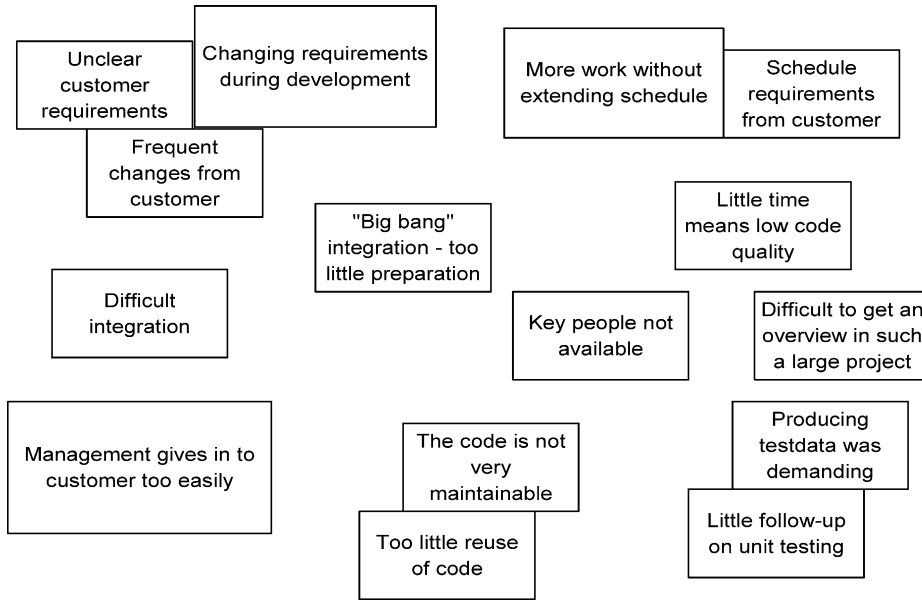


Fig. 2. Post-it notes showing some of the problems in a software development project, after a KJ session. The notes are grouped thematically. Each group was later given a new name, for example the three notes in the upper left corner were named ‘changing requirements’.

3.3. The postmortem

We organised the postmortem as described by Birk et al. in Section 2.3, and will now present some of the results.

Because the participants in the postmortem meeting knew each other well, we started with a brief introduction, followed by a KJ brainstorm session to identify issues that went well.

One result from the KJ session on problems that appeared in the project, was three post-it notes grouped together and

named ‘changing requirements’. They are shown in the upper left corner of Fig. 2. When presenting these notes, participants gave the following statements for two of the notes:

“Another thing was changes of requirements during the project: from my point of view—who implemented things, it was difficult to decide: when are the requirements changed so much that things have to be made from scratch? Some wrong decisions were taken that reduced the quality of the software”.

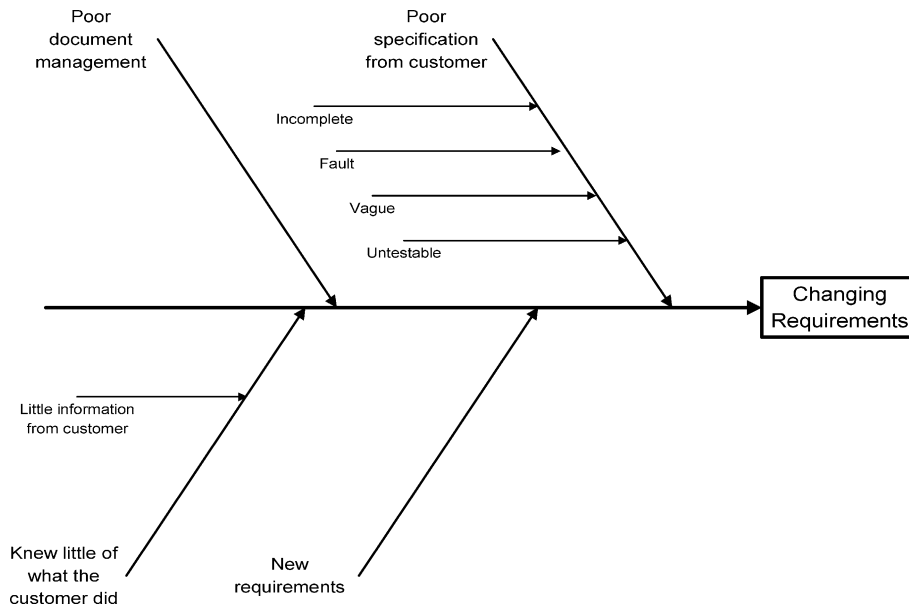


Fig. 3. Ishikawa diagram showing main and sub-caused for ‘changing Requirements’. For example, participants in the postmortem meeting thought that changing requirements was partly a problem because of a poor original specification from the customer. The specification was poor because requirements were incomplete, contained faults, were vague and untestable.

“Unclear customer requirements—which made us use a lot of time in discussions and meetings with the customer to get things right, which made us spend a lot of time because the customer did not do good enough work.”

When we later brought this issue up again in order to find some of the root causes for ‘changing requirements’, we ended up with the fishbone diagram in Fig. 3.

The root causes for the changing requirements, as the people participating in the analysis saw it, was that the requirements were poorly specified by the customer, there were ‘new requirements’ during the project, and the company knew little of what the customer was doing. Another reason for this problem was that documents related to requirements were managed poorly within the company. In Fig. 3, we have also listed some sub causes.

After the postmortem meeting was finished, we asked people to state what they thought of the process. All participants had got new insights on the project—were able to see issues from new perspectives. Also, many stated that the way of conducting postmortem was motivating in itself because it was unusual (their normal workday would be to develop software in cell offices and attend normal meetings).

Given the time restrictions to use only half a day, we did not give recommendations to management in the company, other than stating the successes and problems in the project. The report was later discussed in a meeting where all project managers in the company were invited, where they discussed changes in how projects were carried out based on what was learned in this project.

4. Summary and discussion

In a study in 19 companies across Europe in industries such as management consulting, engineering, construction and telecommunications on project-based learning practices, Keegan and Turner [38] found that ‘project team members frequently do not have the time for meetings, or for sessions to review lessons learned. Often, project team members are immediately reassigned to new projects before they have had time for lessons learned sessions or after action reviews’. They did not find a single company where employees expressed satisfaction with the postmortem

process. Keegan and Turner do not discuss what kind of postmortem processes existed in the companies, but the main finding was that the processes were seldom used in practice.

We think there is a need for helping software companies choosing simple and practical methods for conducting postmortems, to make it easier to perform postmortems to a higher degree. The benefit of conducting postmortem reviews are mainly that it provides a learning forum where discussions are relevant to the project and to the company. It can also be a way for management to show that they listen to what the employees say, and are willing to discuss improvement efforts.

We will now discuss the three approaches described (some of the discussion points are summarized in Table 1) more in depth. In the discussion, we use the material from the medium-sized satellite software company for examples.

4.1. Requirements for a good postmortem process

Openness, patience, the ability to listen, experimentation with new words and concepts, politeness, the formation of a persuasive argument and courage are some ingredients for a good discussion [20]. In a postmortem this is sought by having a skilled process leader who encourages open dialogue, and should prevent critique of individuals and that dominating people get the most of the meeting time.

Norman Kerth [27] emphasizes the importance of a good atmosphere through the ‘prime directive’: “Regardless of what we discover, we must understand and truly believe that everyone did the best job he or she could, given what was known at the time, his or her skills and abilities, the resources available, and the situation at hand”. For longer postmortems, exercises such as ‘create safety’ and ‘understanding differences in preferences’ [27] can be used to further focus on creating a good atmosphere.

In the satellite software case, we only used a short introduction, as team members knew each other well from working closely for a long period of time. Participants spent 5 h each, a total of 25 h. Two facilitators spent ten hours each, giving a total time expense of 45 h, which is less than 0.1% of the total time spent on the project.

It is difficult to give advice on what is the ‘optimal’ time usage for a postmortem. More time will allow more issues to be discussed deeper, thus increasing the learning effect. The

Table 1
Summary of selected differences between three methods for conducting postmortem reviews

	Whitten	Collison and Parcell	Birk et al.
Who to invite?	From each major participating organization	All project members, possibly new project	All project members
Homework?	Yes	No	No
Type of discussion?	Open	Open	Structured
Output?	Recommendations	Guidelines Histories Names of people Key artifacts	Structured report with issues that went well and could be better

time used for a postmortem should depend on what kind of strategy a company has—codification requires more work than personalisation. It should also depend on the size of the project, as there should be more issues to discuss in a larger project than in a small.

4.2. *Who to invite*

The three methods described all argue that one should invite a broad audience for a postmortem. Whitten mentions participants from planning, development, test, usability and module build as example roles to invite. Collison and Parcell suggests that people from similar projects that are underway should be invited as well as key members from the project. The method of Birk et al. recommends getting ‘as many people as possible’ from the project to participate.

Looking back at the methods for knowledge sharing, it seems reasonable that all participants in a project can contribute with knowledge that is relevant for future projects through socialisation. Inviting many people can also broaden existing communities of practise within an organisation, especially if people from new projects are also invited. If the postmortem is conducted as a lightweight process, the cost will not be high.

Inviting external stakeholders such as a customer will move the focus from internal events to stakeholder relations. This makes it difficult to blame stakeholders that are not present in the postmortem meeting, like the people in the satellite software company partly blame the customer for a poor requirement specification and management for giving into customer demands too easily.

4.3. *With or without homework?*

Should a postmortem include ‘homework’ for the attendants? Whitten recommends that all attendees go through a set of questions to prepare themselves for the workshop. Collison and Parcell do not put emphasis on homework, neither do Birk et al. A reason for doing homework is that the learning process is taken over a longer period of time. People who prepare can also easier contribute in an open meeting session. In the method suggested by Birk et al., all participants are given time for reflection during the workshop, to identify main successes and problems. Given a high number of participants, there is a high probability that the most important issues are dealt with, even without homework. But homework can stimulate individual reflection, externalisation, but will also require more time.

Another question is whether the facilitator should do homework. With the method of Birk. et al., the facilitator does not need to know much about the project, as the main intention is to use techniques to get the participants to reflect. However, if the facilitator style is more intrusive, asking questions that is to stimulate reflection, preparation is necessary.

In the satellite software case, the attendants did not do any homework, and the facilitators had little information about the project—only a short discussion with the project manager before the postmortem review meeting.

4.4. *Facilitator*

All methods recommend using a facilitator for the meeting. The question is what kind of person is the right to use. The project manager can be one option, but this person is so much involved in the project that it can be difficult to allow everyone to express opinions without commenting. Also, issues that people think can be sensitive to the project manager might not appear. It is probably wise to use someone from outside of the project, whom the participants trust. It can also be someone who is external to the company. A benefit of using an external person is that participants have to explain issues to this person more thoroughly than they would to an insider. This can cause different interpretations within the project to be uncovered. The facilitator should also be properly trained in order to follow-up when statements from people are unclear.

4.5. *Open or structured discussion?*

Another question is whether to have an open or structured discussion of the experience from the project. An open dialogue as suggested by Whitten is seen as a central learning instrument in the works of Senge. However, this form can easily take a lot of time, and might focus on a limited number of issues. It might also be that these issues are only interesting to the most dominant people taking part in the meeting. Birk et al. are using the KJ method in order to give each participant the possibility to influence equally on the topics. The KJ method is equally strong whether the participants prefer thinking about ideas through quiet introspection or interactive brainstorming.

A drawback with the KJ method can be that it takes time to reach consensus on new names for groups of issues. In the case with the satellite software company, however, this was not time-consuming. This might be because we did not encourage discussion about whether an issue was a problem or not, we focused on discussing ‘what would be a proper name for a set of issues that some participants felt were important’.

4.6. *With or without management?*

Should the management or the project manager take part during a postmortem? We do not think the management should take part in the postmortem, as the intention is to focus on learning, and management also has a role of evaluating employees. This can be a problem as we saw in the satellite software example, where management was blamed for some of the problems in the project. But this

kind of problems can be discussed with management after the postmortem meeting is over.

The project manager is very useful to include because this person has a more overall view of the project than the rest of the participants. But this person can also be quick to defend all decisions taken during the project, and make it difficult to have a free exchange of ideas on how to improve the next project. Given a strong facilitator that is aware of the possible problems with the project manager, we think a project manager should be invited to get a more complete overview in the postmortem.

4.7. What should be output?

What should the output of a postmortem be? Whitten describes a list of recommendations that are given to the company management in order to ensure learning in other projects. Collison and Parcell also mention such guidelines for the future, but also mention histories to illustrate the guidelines, names of people involved and key artifacts. They also recommend using direct quotes to capture the depth of feeling and to create a summary that is easily read. Birk et al. suggests writing a report which describes the project, what went well, what went wrong, and the causes of what went well and wrong. They also transcribe much of what is said during the meeting in order to give more context for future readers. If the intention of the postmortem mainly is to come up with improvement suggestions, probably the method described by Whitten is sufficient. But if the intention is to transfer knowledge also to people who did not take part in the postmortem, the method of Birk et al. is more appropriate.

There are many examples of postmortem reports not being used. Kerth [27] argues that the participants in the postmortem meeting should write the report, otherwise they loose commitment to the content. The Cross-Affinity Exercise [27] produces proposals for change, which identifies people willing to work on the change.

4.8. Learning focus: tacit or explicit knowledge?

An area related to the previous discussion is what kind of knowledge transfer is intended from the postmortems. If we go back to the two strategies suggested by Hansen et al., we can view postmortems as supporting personalisation in that it provides an arena for ‘reflective practice’ where participants can discuss past events.

From a community of practice-view, a postmortem can be one arena to engage in and to contribute to the community. The main aim of the postmortem is to discuss changes that will lead to refined practice.

We can also see postmortems as an attempt to codify knowledge from projects, where the main output is the report, which should provide insight to other project teams (as a part of systematically capturing, storing, interpreting

and distributing relevant experience from projects as seen as an important learning mechanism by Huber [39]).

How postmortems are used should depend on what strategy the company has. Smaller companies should focus on sharing tacit knowledge, as a codification strategy is expensive. Larger companies are more dependent of codified knowledge, and should invest more in the documentation.

5. Conclusion and future work

We have investigated postmortem reviews from a knowledge management perspective, and presented three methods for conducting postmortems from the literature. We have also presented example results from a postmortem report.

The methods vary in several dimensions. They put different emphasis on who to invite, how to prepare, how to facilitate the postmortem meeting, how to structure discussions, and what the written output of the postmortem is to be.

Companies wanting to conduct postmortems should decide on the method to use after what general strategy they have for knowledge management. They should also decide whether they want to focus purely on internal project affairs, or also to include relations to project stakeholders. A general advice is to use people who are not directly involved in the project to facilitate the postmortem meeting.

Acknowledgements

I am grateful to many people for having discussions on postmortem reviews, particularly the research group at SINTEF ICT: Nils Brede Moe, Tore Dybå, Geir Kjetil Hanssen, Hans Westerheim and Tor Erlend Fægri. I would further like to thank Tor Stålhane at the Norwegian University of Science and Technology, Kevin Desouza from the University of Illinois, Max von Zedtwitz from Tsinghua University and especially Norman Kerth for comments on this article.

This work was conducted through the Software Process Improvement based on Knowledge and Experience (SPIKE) project, supported by the Research Council of Norway.

References

- [1] V.R. Basili, Quantitative evaluation of software engineering methodology, Proceedings of the First Pan Pacific Computer Conference, 1985.
- [2] M.C. Paulk, C.V. Weber, B. Curtis, M.B. Chrissis, The Capability Maturity Model: Guidelines for Improving the Software Process, Addison-Wesley, Boston, 1995.

- [3] V.R. Basili, G. Caldiera, H.D. Rombach, The experience factory in: J.J. Marciniak (Ed.), *Encyclopedia of Software Engineering*, Wiley, New York, 1994, pp. 469–476.
- [4] D. Garvin, Building a learning organization, *Harvard Business Review* 1993; 78–91.
- [5] G. Huber, Organizational learning: a guide for executives in technology-critical organizations, *International Journal on Technology Management, Special Issue on Unlearning and Learning for Technological Innovation* 11 (1996) 821–832.
- [6] E. Wenger, *Communities of Practise: Learning, Meaning and Identity*, Cambridge University Press, Cambridge, UK, 1998.
- [7] P.M. Senge, *The Fifth Discipline: the Art and Practise of The Learning Organisation*, Century Business, 1990.
- [8] Webster's Encyclopedic Unabridged Dictionary of the English Language, Gramercy Books, New York, 1989.
- [9] R. Stata, Organizational learning: the key to management innovation in: K. Starkey (Ed.), *How Organizations Learn*, Thomson Business Press, London, 1996, pp. 316–334.
- [10] C. Argyris, D.A. Schön, *Organizational Learning II: Theory, Method and Practise*, Addison Wesley, Reading, MA, 1996.
- [11] C. Argyris, *Overcoming Organizational Defences: Facilitating Organizational Learning*, Prentice Hall, Englewood Cliffs, NJ, 1990.
- [12] R.L. Feldmann, K.-D. Althoff, On the status of learning software organisations in the year 2001, *Learning Software Organizations Workshop*, 2001, pp. 2–6.
- [13] T. Dybå, Enabling software process improvement: an investigation on the importance of organizational issues, Dr. ing thesis, Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, 2001, pp. 332, ISBN 82-471-5371-8.
- [14] J. Lave, E. Wenger, *Situated Learning*, Cambridge University Press, Cambridge, New York, 1991.
- [15] I. Nonaka, H. Takeuchi, *The Knowledge-Creating Company*, Oxford University Press, New York, 1995.
- [16] M.T. Hansen, N. Nohria, T. Tierney, What's Your Strategy for Managing Knowledge?, in: *Harvard Business Review on Organizational Learning*, Harvard Business School Press, Boston, 1994, pp. 61–86.
- [17] J.A. Raelin, Public reflection as the basis of learning, *Management Learning* 32 (2001) 11–30.
- [18] M.M. Menke, Managing R&D for competitive advantage, *Research Technology Management* 40 (1997) 40–42.
- [19] M. Zedtwitz, Organizational learning through post-project reviews in R&D, *R&D Management* 32 (2002) 255–268.
- [20] G.v. Krogh, K. Ichijo, I. Nonaka, *Enabling Knowledge Creation*, Oxford University Press, New York, 2000.
- [21] P.L. Townsend, J.E. Gebhart, *How Organizations Learn*, Crisp Publications, 1999.
- [22] A. Kransdorff, Using the benefits of hindsight—the role of post-project analysis, *The Learning Organization* 3 (1996) 11–15.
- [23] B. Collier, T. DeMarco, P. Fearey, A defined process for project post mortem review, *IEEE Software* 13 (1996) 65–72.
- [24] M.A. Cusomano, R.W. Selby, *Microsoft Secrets—How the World's Most Powerful Software Company Creates Technology, Shapes Markets, and Manages People*, The Free Press, 1995.
- [25] W.S. Humphrey, The postmortem, in: *Introduction to the Team Software Process*, Addison Wesley Longman, Reading, MA, 1999, pp. 185–196.
- [26] K. Schneider, LIDs: a light-weight approach to experience elicitation and reuse, Second International Conference on Product Focused Software Process Improvement, PROFES 2000, 2000, pp. 407–424.
- [27] N.L. Kerth, *Project retrospectives: a handbook for team reviews*, Dorset House Publishing, New York, 2001.
- [28] M.J. Tiedeman, Post-mortems—methodology and experiences, *IEEE Journal of on Selected Areas in Communications* 8 (1990).
- [29] R. Condon, Postmortem: Z-axis's aggressive inline, *Game Developer* 2002; 42–49.
- [30] N. Whitten, *Managing Software Development Projects: Formula for Success*, Wiley, New York, 1995.
- [31] C. Collison, G. Parcell, *Learning to Fly: Practical Lessons from one of the World's Leading Knowledge Companies*, Capstone Publication, 2001.
- [32] A. Birk, T. Dingsøy, T. Stålhane, Postmortem: Never leave a project without it, *IEEE Software, Special Issue on Knowledge Management in Software Engineering* 19 (2002) 43–45.
- [33] T. Dingsøy, N.B. Moe, Ø. Nytrø, Augmenting experience reports with lightweight postmortem reviews in: F. Bomarius, S. Komi-Sirviö (Eds.), *Third International Conference on Product Focused Software Process Improvement*, Springer, Kaiserslautern, Germany, 2001, pp. 167–181.
- [34] T. Stålhane, T. Dingsøy, N.B. Moe, G.K. Hanssen, Post mortem—an assessment of two approaches, *EuroSPI*, 2001, pp.
- [35] T. Dybå, T. Dingsøy, N.B. Moe, *Process Improvement in Practice—a Handbook for it Companies*, Kluwer, Boston, 2004.
- [36] R. Scupin, The KJ Method: a technique for analyzing data derived from Japanese ethnology, *Human Organization* 56 (1997) 233–237.
- [37] D.J. Greenwood, M. Levin, *Introduction to Action Research*, Sage, Beverly Hills, CA, 1998.
- [38] A. Keegan, J.R. Turner, Quantity versus quality in project-based learning practises, *Management Learning* 32 (2001) 77–98.
- [39] G.P. Huber, Organizational learning: the contributing processes and the literatures, *Organizational Science* 2 (1991) 88–115.