

A Review of Small and Large Post-Mortem Analysis Methods

Mauri Myllyaho¹, Outi Salo², Jukka Kääriäinen², Jarkko Hyysalo¹, Juha Koskela²

¹ Department of Information Processing Science,
P.O.Box 3000, FIN-90014 University of Oulu, FINLAND
Phone: +358 8 553 1900
Fax: +358 8 553 1890
mauri.myllyaho@oulu.fi, jarkko.hyysalo@oulu.fi

² VTT Technical Research Centre of Finland
P.O.Box 1100, FIN-90571 Oulu, FINLAND
outi.salo@vtt.fi, jukka.kaariainen@vtt.fi, juha.koskela@vtt.fi

Abstract Post-mortem analysis (PMA) is an empirical study method in software engineering. It is an important, but often forgotten, way of gathering empirical knowledge. PMA is ideally performed either soon after the most important milestones and events or at the end of a project, both in successful and unsuccessful software development projects. The benefit is that post-mortems can often reveal findings more frequently and differently than project completion reports alone. This paper reviews PMA as a project based learning technique, and discusses how such a technique can be used to perceive the different settings of software development projects. In this study, two techniques were used to gain information about the post-mortems: firstly an extensive literature search was carried out on software engineering and management literature, and secondly post-mortem sessions were held in small software development projects. Based on the findings, we present a general post-mortem analysis process as an example from a formal framework, used for identifying innovation possibilities based on lessons learnt. Different PMA techniques are affected by the size of the project, as well as the scope and density of the PMAs: previously post-mortem reviews have been suggested to be used after the average-sized or large projects; and recently some agile methods have proposed using team reflection techniques with small teams, for instance to adjust the processes at end of each iteration (release). Furthermore, we conclude from our findings that PMA is worthwhile in small and medium-sized software development projects for which lightweight versions of PMA can be tailored.

Keywords : Post-mortem analysis, agile software development, process improvement

1. INTRODUCTION

“Learning lessons from projects is important, but the practice of post-mortems does not occur frequently in practice.” [1]

Recently it has been shown that carefully planned and organised post-mortem analyses (i.e. not ad-hoc) can bring unexpected benefits in practice. For instance at VTT [2] Technical Research Centre of Finland, PMA has significantly helped to improve and optimize practices in extreme programming. In another study by Stålhane et al. [3], one major finding was that “a software company had not analysed properly, and with all the relevant participants, what was reusable from their earlier projects”. In their case PMA helped to identify new improvement possibilities in the development process. For example, it became clear that some key documents, such as project completion reports, were not remembered afterwards in detail by everyone. Usually the report writing is the task of a project manager, and the information may not be further discussed after writing or publishing the report. PMA is a simple and quick way of revealing information and knowledge about the about the various project aspects.

There are many different ways of organising project reviews. Some PMA tools and techniques are available on the “Project Review Web Site” [4], and a description of the post-mortem process for small projects can be found in “A practical guide to Lightweight Post Mortem Reviews” [5]. Below is a list of examples illustrating benefits of applying both small and large post-mortems, based on the experiences described in [6] and [7]:

- 1) it helps project team members share and understand each other’s perspectives;
- 2) it integrates individual and team learning;
- 3) it identifies hidden problems;
- 4) it documents good practices and problems (so as not to repeat bad practices);
- 5) it increases job satisfaction by giving people feedback about their work;
- 6) and in some cases, PMA can even improve project cost estimation.

In the Spacetec case [3], four researchers (facilitators) and three software project members participated in a lightweight review session, which was carried out in one day. They used the KJ-method [8] with post-it notes together with fishbone diagrams for idea generation and analysis. Fishbone diagrams are also known as *cause and effect* diagrams, and are a part of well known business problem and analysis techniques introduced by Cross [9]. The aim of the Spacetec analysis was to present both the positive and negative findings with recommended practices in compact PMA reports (10-15 pages).

Unlike small companies and small projects, the post-mortem analyses in large software companies like Apple and Microsoft are often harder to finalise. Dingsøyr and Hanssen [10] state that generally PMAs in very large companies “take [at least] three to six months”. A well known PMA approach [for large projects] is defined by Collier, DeMarco and Fearey [11]. It starts with surveys and finishes by writing masses of review documents.

This paper is composed as follows. First, a general description of post-mortem analysis is presented. The differences and similarities between the small (also referred to as lightweight) and large PMAs based on our experience are discussed and compared, and the literature is reviewed. There is a discussion of the general findings related to the projects, where ideas of lightweight PMA have been applied. Finally the conclusions explain why PMA is an important empirical research method, and how it supports process improvements.

2. POST-MORTEM ANALYSIS

PMA, or its main, activities have many terms such as: blame and flames, debriefing, lessons learned, post implementation review, post project review, postpartum, project audit, project review, retrospective, team retrospective. We use the term *post-mortem*, as it is the most common term used in the recent publications (see e.g. [12],[13]), and in Collier [4] project reviews on software projects are most often referred to as post-mortems.

It has been established by Conradi and Wang [14] that four kinds of empirical software engineering studies (see Table 1) often follow a certain sequence. Firstly, post-mortems or surveys are carried out to pre-analyse the experience using a specific software technology or process. Then, controlled experiments are conducted to test out the more concrete hypotheses of the chosen study object, often with university students. Lastly, case studies are used to examine the refined and adapted hypotheses in industrial settings.

Table 1. Taxonomy of the most common empirical study methods, based on [13], [12].

Category	Description	Example(s)
Observational	Collect relevant data as project develops.	Case study: monitoring a case within the specific setting (“research-in-the-typical”)
Historical	Collect data from “completed” projects.	Post-mortem: analyzing the outcome in the retrospect (“research-in-the-past”) Survey: gathering information from a large number of participants (“research-in-the-large”)
Controlled	Provide multiple instances of an observation.	Experiments: observation with a narrow focus and controlled circumstances (“research-in-the-small”)

As lightweight post-mortem reviews are said to suit small-and-medium sized companies (see e.g. [6]), it is important to determine first how the different sizes of the software projects are defined. Commonly used methods for starting a software project’s evaluation in large software projects, are function points or lines of code (LOC). Thereafter the work effort is estimated. However, the question arises as to what determines whether the project is small, medium, average or large. It is difficult to find a common definition, because project size is a very subjective measure depending on variable factors such as culture and context. However, below are some example definitions found in the literature:

- small projects ... are generally defined as having a staff size of one or two people and a schedule of less than six months [15];
- with medium-sized companies ... 5-10 people usually participate in a project, ranging in size from about 8 to 50 man months [16];
- an average sized project team has 150 members [4].

Post-mortem analysis can be defined as a series of steps aimed at examining the lessons to be learned from products, processes and resources to benefit on-going and future projects. Post-mortems enable individual learning to be converted into team and organizational learning. It is necessary to search for innovation opportunities, e.g. imaginative and useful ideas, systematically. If innovation is defined as hard work, in [17] it means having to: “establish the right roles and processes, set clear goals and relevant measures, and review progress at every step”.

In addition to innovation, reflective reviews and the exploitation of many small opportunities or failures in retrospect can bring out new paradigms. According to Drucker [17], among the history-making innovations the ones based on new knowledge have the longest lead times, and it may take many years before all the elements suddenly converge. Like Rome, extreme programming (XP) was not developed in a day. It was initially invented and tested during the DaimlerChrysler’s C3 project, after Kent Beck joined the team as head coach in 1996. The roots of XP are the result of a combination of many previously known practices, as explained by Beck [18].

2-1 General process description and its steps

Depending on the size and the schedule of the project, there are many alternative ways and tools for conducting a PMA. To ensure a successful project review, it is essential to capture the right (unbiased) information and to plan the process carefully. In [11], the PMA process for average-sized and large projects is defined in five steps: 1. project survey, 2. collect objective information, 3. debriefing meeting, 4. project history day (= post-mortem review), and 5. publication of the results; whereas Birk et al. present [6] the PMA process for small and medium sized projects in four steps: 1. preparation, 2. data collection, 3. analysis, and 4. results and experience.

The thorough analysis of the proposed PMA process descriptions provided a framework for creating a general post-mortem analysis process (Figure 1). It illustrates the main steps needed for all systematic post-mortem analyses. Other example guidelines from the fairly similar and successful PMA process used at Microsoft can be found in [19]. In addition, a good PMA-toolbox, applicable to both small and large projects, is provided by Collier [4]. A good example in the toolbox, which shows the aims of every project’s final review is the sample of the “open letter”, also called as the letter of *good, bad and ugly*. In summary, the aim of any post-mortem is to provide answers to the following four key questions [20]:

- what did we do well that does not have to be further discussed (the positive aspects)?
- what did we learn (the negative aspects)?
- what should we do differently the next time (the negative aspects which require improvements)?
- what still puzzles us (share the things that are confusing or do not make sense)?

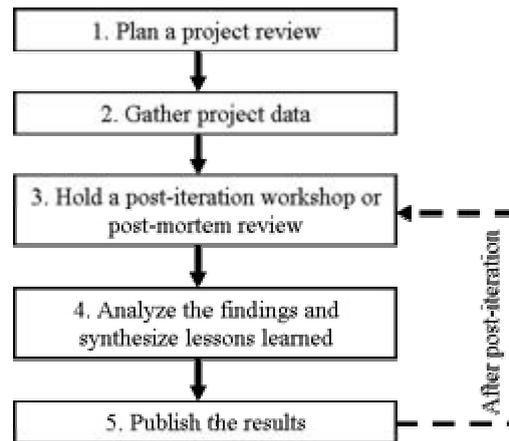


Figure 1. A general and iterative post-mortem analysis process.

The general post-mortem process has the following five fundamental steps (adapted from [4]):

1. a project review is planned to identify the most suitable methods and tools used in the other steps. The post-mortem reviews, the reasons for the review, the focus and the participants are defined;
2. both objective and subjective data are collected from all the project participants via pre-defined metrics, surveys, debriefings, etc. to identify the useful information for the “following step” (workshop/review);
3. a “project history day” is the most important step, and it is held to combine reflective analysis of project events with the actual project data after a project’s major milestone (post-iteration), or after a project has finished (post-mortem). In the case of large projects, only a few key people participate in this session;
4. the findings are analysed, prioritized and synthesized as lessons learned. This is often started during the project history day after identifying and prioritising the positive events and problems ;
5. the summary of the findings is published and presented in a way that enables future projects to know what processes or tools are important to continue, and to turn problems into improvement activities.

3. SMALL AND LARGE POST-MORTEMES

Pfleeger [21] mentions that, for every project, processes can be evaluated by collecting and analyzing large amounts of data during the development stages and after the project has ended. In small projects, there are often fewer software development artefacts such as design documents or open issues, when compared to bigger projects. These artefacts partially define how the review process can be carried out. Additionally, how do we make a distinction between small and large post-mortems with so many definitions of the software project’s size? We have set this limit to be approximately 30 people. That is close to the maximum sizes of I) manageable agile software project, and II) reasonable debriefing meeting, which is one reliable way of gathering project related data (step 2 in the general PMA process).

2-1 Small PMAs

In agile software development projects, post-mortems can be conducted in the traditional way, but also iteratively during the project. The agile principles [22] suggest that “at regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly”. Furthermore, agile proponents have noted that “each situation calls for a different methodology” [23]. Therefore, iterative and light post-iteration analysis sessions have been suggested for use in agile projects (e.g. [10],[23],[24]). One of the key differences in large PMAs is their focus on improving and adapting current software development processes based on the experiences of previous iteration, rather than just learning from the experiences of finished projects. Despite this, the results from post-iteration workshops could also be utilized in broader perspectives in organizations.

Extreme programming (XP) [25] is an agile software development method designed to suit small development teams from two to ten developers. It also emphasizes short development iterations from one to four weeks. These factors effect how such a workshop can be conducted. For example, in order to enable such a frequent event, the workshop should be short and effective, i.e., not taking too much effort from the project team, yet yielding immediate and visible outcomes to motivate the project team for further such activity.

The previously suggested lightweight post-mortem techniques include Cockburn’s [23] methodology-growing technique that includes a team reflection workshop, after each of the iterations. Dingsøy and Hanssen [10] have suggested a learning mechanism called post-mortem reviews to be used as an extension for agile software

development methods in an iterative manner. Additionally, Gottesdiener [26] suggests using team retrospectives for the rational unified process (RUP) assessments after each RUP phase such as elaboration and construction. All of these techniques aim to generate both the positive and negative experiences from the previous iterations for process improvements based on knowledge. In other words, the team is allowed to change their daily working practices to better suit the current project. The facilitator is preferably outside the project team and has an important role regarding the effectiveness of the workshop, usually in keeping the discussion on track.

2-1-1 VTT Case Studies (or post-iteration workshops)

In an ongoing series of agile case studies conducted at VTT Technical Research Centre of Finland, a lightweight post-iteration technique (hereafter referred as post-iteration workshops) has been tried out. It combines valuable elements from both the team reflection workshops [23] and the post-mortem reviews [10] and also includes some additional alterations (more details in [2]). As suggested in the post-mortem review technique, the problem-solving brainstorming method called KJ method was adopted in the post-iteration workshops. It was used for generating and collecting both positive and negative experiences in the project team and structuring them into labelled groups.

Naturally, these kinds of workshops may lead to a large number of findings and groups, especially if the number of findings is not limited or prioritized in any way. For example, in the first workshop of the second case project the five team members generated over 40 negative findings that were distributed into approximately 20 topic areas. Thus, it was really important to effectively process such a large number of issues. Although the Root Cause Analysis technique is suggested for the post-mortem review technique, it was replaced by a focus group technique [27], which is a group interview technique based on a free and open discussion lead by a facilitator.

The results of the VTT case studies indicate that such a workshop can be effective both regarding the results and the effort. For example, the data from the three first case studies reveal (Figure 2) how the negative findings of the project teams as well as the number of changes needed in the process declines towards the end of the projects in terms of improved process and more satisfied project teams. These case studies supported Cockburn's [23] findings that the teams "did not just talk about changing their way of working, they actually changed their way of working".

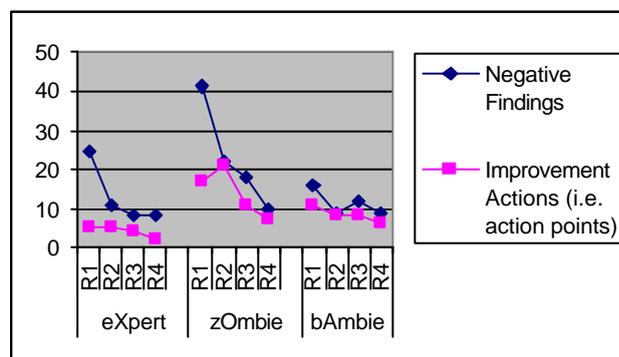


Figure 2. Efficiency of the post-iteration workshops in three VTT case studies.

The group interview technique was found to be an effective way of discussing all the findings and to generate collective process improvements for the following iteration by the project team. For example, the 40 negative findings mentioned earlier became 17 concrete process improvements of various kinds – usually relatively small ones. Furthermore, the role of the facilitator is very important, both in terms of their knowledge of the software development context as well as the post-mortem technique, in order to achieve results from the experiences of the project team.

In these case study projects, the average duration of the workshops varied between 1.82 and 1.15 hours. The shortest workshop lasted less than one hour, and the longest one lasted close to 2.5 hours. This covered the generation of the experiences and specific process improvement actions with allocated responsibilities for the project team members (i.e. action points), as well as follow-up of previous action points. Furthermore, the duration of the workshops was shortened towards the end of each project. In one case, this was because fewer changes were needed for the process, i.e. the negative experiences of the project teams had lessened. Additionally, the increasing experience of the facilitator could be a factor affecting the shorter duration of the workshops.

It could be claimed that lightweight post-mortem sessions require a certain amount of effort from the project team to produce actual improvements in the process. The data from the workshops indicate that in the latter stages of the projects, the number of process improvement actions diminishes. This should mean that at some point of the longer-lasting projects, the frequency of the post-iteration workshops could be reduced. However, the real benefits of these sessions may lie in their providing regular and systematic project gatherings where positive issues as well as the unfinished and problematic issues are disclosed and discussed together.

2-2 Large PMAs

According to Zedtwitz [28] a survey, carried out between 1997 and 2001 in research and development (R&D) projects [in well recognized global companies or organizations], revealed that only one out of five projects had conducted a post-project review. While today R&D depends on the usage of software or computer hardware, these results can be said to be relevant to the software industry also. In addition, a survey [29] of software industry professionals found the following: “fifty-two percent of the respondents [from a total of 106] said that they had participated in one [post-mortem] at some time of their career and the others had not”. From this sample close to 57% had more than 5 years work experience in the software industry.

We should now consider why post-mortem reviews happen so rarely, even though many books (e.g. [30]) recommend them. The reason may be that conducting a post-mortem review properly “is not such an easy task one would expect”. Zedtwitz [28] mentions that the studies with big international companies like Hewlett-Packard, Daimler-Chrysler, and Roche have identified the following difficulties with post-project reviews:

- the frequency of reviews varied between business units ;
- the quality and results depended on the skills and talents of the review facilitator;
- some organizations had so many projects coming in that potential project managers were immediately reassigned to the new projects [after finishing the previous ones];
- when many top-managers were participating, the reviews often developed into marketing events rather than post-project analyses.

Additionally, there can be a lot of resistance to the learning aspect from post-project reviews. Time constraints and the lack of time are quite often mentioned as the main reasons for skipping over the whole post-mortem analysis. Zedtwitz states that [28] “people are unlikely to devote time and effort to yesterday’s problems since natural incentives favour moving ahead to the next problem instead of spending valuable time on reviewing a recently completed project”.

However, Dürr et al. [31] provide an example of the kind of process improvement possibilities carried out in Austrian Aerospace. The size of the team was only 12 persons, but because of the length of the project, the post-mortem process followed the guidelines of a large post-mortem review. Their analysis led to a further discussion of alternative ways to be used in subsequent projects so as to collect the lessons learned.

2-2-1 Software game industry

PMA was found to be common practice in the game development industry. It is used in both small and large projects. For example, the game development site Gamasutra offers over 70 articles on “the good, the bad and the ugly” of game development projects (see [32]). The experiences vary from one-man-projects to over 70 person projects with a large number of contractors and over five years of development time. There are also other game development sites offering insights into development by providing a community of peer professionals to exchange experiences based on real life projects.

Gamasutra’s PMA articles follow a structure proposed for the Open Letter Template (see [4]) offering a brief project description with information covering the most critical development issues. Lessons learned identify the following aspects:

- The Good. Good solutions, improvements, suitable tools etc. including areas of project management, communication, marketing and engineering practices;
- The Bad. List of key issues and problems and cause of the problems ;
- The Ugly. Few critical issues that absolutely must be corrected.

The collection of this data creates a knowledge base for developers with real life examples and experiences. It helps find experiences in comparable situations and builds a community for knowledge sharing. Typically the following game project metrics are offered in addition to the experiences:

- Publisher and developer
- Number of full-time developers, part-time developers, and contractors
- Length of the development, and release date

- Game platforms, and development hardware and software used

The post-mortem article “Tom Clancy’s Splinter Cell” [33] provides a good example of how these metrics and experiences are expressed, based on lessons learned. Further ideas about how to conduct a post-mortem analysis, and which techniques to use are given in Norman L. Kerth’s book “Project Retrospectives” (see [34]).

4. DISCUSSION

Zedtwitz [28, page 257] presents how knowledge becomes specific in individual, team, and organizational levels of learning. He further proposes a post-project review maturity model to demonstrate how this reflective process can be brought from the individual to organizational level. The levels of learning and the proposed maturity model could be a good starting point for further research related to software project improvements. In addition to PMA, which is a retrospective approach, there are also other approaches such as the intuitive, incidental and the prospective approaches to learn from past experiences (see e.g. [35]).

It is essential to recognize a need for the small or lightweight post-mortem review process, often providing quick feedback as in the case of post-iteration workshops. One reason is the size of the manageable work units as Krol and Kruchten state [36] “for larger software development organizations, say 25 people or more, it is common to have the organization broken into smaller teams”. In addition, it has been claimed in [37] that today “the vast majority of software development organizations have fewer than 20 software people”. To check, if post-mortem analysis really adds value via process improvement possibilities, it is useful for the small organization to try it out by first defining and modifying the ways of collecting objective data to provide fairly exact metrics.

It is necessary to make a clear distinction between the closely related terms of post-mortem analysis (PMA) and post implementation evaluation (PIE). In the former, the primary purpose is commonly a source for project based learning to identify possible product or process improvements, whereas in the latter (see e.g. [38] and [39]) the purpose is to evaluate the implementation success of the finished information system.

There are many possibilities for further studies in this area. For example, a study could be carried out on the experience of repositories, and how to make the best use of them, e.g. via learning logs; another study could be done on how widely recognised the use of the post-mortem process is in the software industry; another area of research could be to study whether these types of lessons learned (knowledge) from software projects are really useful in the long term.

5. CONCLUSIONS

The existing literature confirms that PMA efforts are worthwhile and can be highly recommended. According to our initial experience, the results of the project reviews in small projects are also promising. The chosen PMA practice and ideas were found to work well in short, and especially in agile, software development projects with tight schedules. Competence building and the creation of further knowledge is needed, if post-mortem analysis is to be used systematically in experience and project-based learning. When time is a constraint there is often no possibility of going through all the steps of the full analysis. In this situation a simplified or 'lightweight' version of the post-mortem analysis has the potential advantage of revealing information which might otherwise remain unnoticed.

REFERENCES:

- [1] Terry Williams : Identifying the hard lessons from projects – easily; International Journal of Project Management, Volume 22, Issue 4, pp. 273-279, May 2004
- [2] Outi Salo, Kari Kolehmainen, Pekka Kyllönen, Jani Löthman, Sanna Salmijärvi and Pekka Abrahamsson: Self-Adaptability of Agile Software Processes: A Case Study on Post-iteration Workshops; Fifth International Conference on Extreme Programming and Agile Processes in Software Engineering (XP 2004), pp. 184-193, June 6-10, 2004
- [3] Tor Stålhane, Torgeir Dingsøy, Geir Kjetil Hanssen, and Nils Brede Moe: Post Mortem – An Assessment of Two Approaches; In Reidar Conradi and Alf Inge Wang (Eds.). Empirical Methods and Studies in Software Engineering: Experiences from ESERNET, LNCS 2765. Springer Verlag, pp. 129-141, 2003
- [4] Bonnie Collier: Project Review Process Web Site - Postmortem Toolkit; <http://www.projectreview.net/prtoolkit.asp> [Retrieved March 28, 2004], 1996
- [5] Torgeir Dingsøy, Tor Stålhane and Nils Brede Moe: A practical guide to Lightweight Post Mortem Reviews. University of Oslo. [For teaching]; http://www.uio.no/studier/emner/matnat/ifi/INF3120/h03/undervisningsmateriale/Prosjektoppgave/PMA_practical_guide.pdf. [Retrieved March 30th 2004], 2003

- [6] Andreas Birk, Torgeir Dingsøy, and Tor Stålhane: Postmortem: Never leave a project without it; IEEE Software, Special Issue on Knowledge Management in Software Engineering, pp. 43-45, May-June 2002
- [7] John S. Reel: Critical Success Factors in Software Projects; IEEE Software, pp. 18-23, May/June 1999
- [8] Raymond Scupin: The KJ Method: A Technique for Analyzing Data Derived from Japanese Ethnology; Human Organization, 56(2), pp. 233-237, 1997
- [9] Anthony T. Cross: Management Through Visibility: Business Improvement; Murton Group, 2003
- [10] Torgeir Dingsøy and Geir Kjetil Hanssen: Extending Agile Methods: Postmortem Reviews as Extended Feedback; 4th International Workshop on Learning Software Organizations (LSO '02), August 6, 2002,
- [11] Bonnie Collier, Tom DeMarco, and Peter Fearey: A Defined Process for Project Post Mortem Review; IEEE Software, pp. 65-72, July 1996
- [12] ESERNET: Empirical methods for software Engineering: A tutorial; [On-line publication], <http://www.esernet.org/c/esernet/tutorial?tutorialpage=frame.jsp>, [Retrieved March 15, 2004], 2003
- [13] Claes Wohlin, Martin Höst and Kennet Henningsson: Empirical Research Methods in Software Engineering; In Reidar Conradi and Alf Inge Wang (Eds.). Empirical Methods and Studies in Software Engineering: Experiences from ESERNET, LNCS 2765, Springer Verlag, pp. 7-23, 2003
- [14] Reidar Conradi and Alf Inge Wang: Introduction; In Reidar Conradi and Alf Inge Wang (Eds.). Empirical Methods and Studies in Software Engineering: Experiences from ESERNET, LNCS 2765, Springer Verlag, pp. 1-6, 2003
- [15] Kathleen Peters: Software Project Estimation; [web-publication], Software Productivity Center, <http://www.spc.ca/resources/estimate/>, [Retrieved September 15, 2004], 1999
- [16] Torgeir Dingsøy: Knowledge Management in Medium-Sized Software Consultant Companies; [Ph.D work], Norwegian University of Science and Technology, 2002
- [17] Peter F. Drucker: The Discipline of Innovation; [initially 1985], Harvard Business Review, August 2002
- [18] Kent Beck. Embracing Change with Extreme Programming. IEEE Computer, pp. 70-77. October 1999
- [19] Steven Sinofsky and Stefan Thomke: Learning from Projects: Note on Conducting a Postmortem Analysis; HBS 9-600-021. Boston: HBS Press, 1999
- [20] Norman L. Kerth: Retrospectives; <http://www.retrospectives.com>, [Retrieved September 19, 2004], 2000
- [21] Shari Lawrence Pfleeger: Software Engineering: Theory and Practice (2nd edition); Prentice Hall, 2001
- [22] Kent Beck et al.: Principles behind the Agile Manifesto; <http://agilemanifesto.org/principles.html> [Retrieved August 15, 2004], 2001
- [23] Alistair Cockburn: Agile Software Development; Boston: Addison-Wesley, 2002
- [24] Diana Larsen: The Manager's Role in Starting and Ending Projects: Charters and Retrospectives; 21st Pacific Northwest Software Quality Conference, 2003
- [25] Kent Beck: Extreme Programming Explained: Embrace Change; Addison Wesley, 2000
- [26] Ellen Gottesdiener: Team Retrospectives – for better iterative assessment; The Rational Edge, April 2003
- [27] Fred N. Kerlinger, Howard B. Lee: Foundations of Behavioral Research (4th edition); Harcourt College Publishers, 2002
- [28] Maximilian von Zedtwitz: Organizational learning thorough post-project reviews in R&D; R&D Management, 23(2), pp. 255-268, 2002
- [29] Gloria H. Congdon: Techniques and Recommendations for Implementing Valuable Postmortems in Software Development Projects; [M.Sc. Project], University of Minnesota, May 1999
- [30] Project Management Institute: A Guide to the Project Management Body of Knowledge (PMBOK Guide)- 2000 Edition; Project Management Institute, 2001
- [31] Walter Durr, Franz Griesauer and Alfred Werni: Postproject Review as a Source for Improvement Proposals; Proceedings of the 29th Euromicro Conference, pp. 360- 367, 1-6 Sept. 2003
- [32] Gamasutra: Featured Topics: Postmortem; CMP Media. http://www.gamasutra.com/php-bin/article_display.php?category=5 [Retrieved September 22, 2004], 2004
- [33] Wu Dong Hao: Postmortem: Tom Clancy's Splinter Cell; Gamasutra, July 14, 2003.
- [34] Norman L. Keth: Project Retrospectives: A Handbook for Team Reviews; Dorset House Publishing, 2001
- [35] Alan Mumfold: Four approaches to learning from experience; Industrial and Commercial Training, 27(8), 1995.
- [36] Per Kroll and Philippe Kruchten: A Role-Based Guide to the Rational Unified Process. In: The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP, Addison-Wesley, pp 271-286, 2003
- [37] Roger S. Pressman: Software Engineering: a Practitioner's Approach (6th edition); McGraw-Hill, 2004
- [38] Kuldeep Kumar: Post Implementation Evaluation of Computer-Based Information Systems: Current Practices. Communication of ACM, 33(2), pp. 203-212, 1990.
- [39] Margaret Gemmell and Rosane Pagano: A Post-Implementation Evaluation of a Student Information System in the UK Higher Education Sector; The Electronical Journal of Information Systems Evaluation, 6(2), December 2003