A Method to Draw Lessons from Project Post-mortem Databases†

Joost Schalken¹, Sjaak Brinkkemper², and Hans van Vliet¹

¹Vrije Universiteit, Department of Computer Science,
De Boelelaan 1083a, 1081 HV Amsterdam, The Netherlands
(jjp.schalken.jc.van.vliet}@few.vu.nl

²Utrecht University, Institute of Information and Computing Sciences,
Padualaan 14, 3584 CH Utrecht, The Netherlands
s.brinkkemper@cs.uu.nl

Abstract

Post-mortem project reviews often yield useful lessons learned. These project reviews are mostly recorded in plain text. This makes it difficult to derive useful overall findings from a set of such post-mortem reviews, for example to monitor and guide a software process improvement program. We have developed a five-step method to transform the qualitative, natural language type information present in those reports into quantitative information. This quantitative information can be analyzed statistically and related to other types of quantitative project-specific information. In this paper we discuss the method, and show the results of applying it in the setting of a large industrial software process improvement initiative. Through the application of the analysis method in the case study, improved questions for a new evaluation procedure were discovered. The analysis also showed that in this organisation team cooperation and the architecture of the infrastructure had a major impact on project performance.

Keywords

Software Process Improvement, Post-mortem evaluations, Grounded Theory, Empirical Software Engineering

† An earlier, shorter version of this article appeared in the proceedings of the EuroSPI 2004 conference (Schalken et al. 2004).

‡ Send proofs to: Joost Schalken. Address: Vrije Universiteit, Department of Computer Science, De Boelelaan 1083a, 1081 HV Amsterdam, The Netherlands. E-mail: jjp.schalken@few.vu.nl. Telephone: +31 20 598 7751. Mobile phone: +31 6 45 544 859. Fax: +31 20 598 7653.
1 Introduction

Most software project management methods recommend that projects be evaluated. Although the advice to perform project evaluations is sound, most project management methods offer no guidance as to how to analyze the data collected. This paper fills this gap and describes an exploratory data analysis method to extract useful information from qualitative post-mortem project evaluation reports using an approach that is based on Grounded Theory (Glaser and Strauss 1967, Strauss and Corbin 1990).

Two kinds of project evaluations can be distinguished (Jurison 1999): intermediate project evaluations that take place periodically during the course of the project, and post-mortem project evaluations that take place at the end of a project, when the actual task of the project has been finished.

The objectives of these evaluations differ. Intermediate evaluations are used by senior management to periodically assess whether the project's goals and objectives are still relevant (Jurison 1999) and to monitor project risks (McConnell 1996). Post-mortem project evaluations on the other hand have the objective to “evaluate past experience and develop lessons learned for the benefit of future projects” (Jurison 1999). The context of this paper is project post-mortem reviews as a means to develop lessons learned for future projects.

The insights gained in the course of the project are made explicit and are recorded in the post-mortem project evaluation report. This evaluation can be useful to the people that have been directly involved with the project, but also to the organization at large. People directly
involved in the project may gain an understanding of the factors that attributed to and/or undermined the project's success. One of the strengths of post-mortem project evaluations is that they force people to reflect on their past work and at the same time the evaluations also provide feedback from other team members. These lessons learned can also be used outside the group of people directly involved in the project, to reuse improvements and to avoid pitfalls in future projects throughout the organization.

Before insights from project evaluations can be used by the rest of the organization, they first need to be packaged for reuse (Seaman 1993). For the organization as a whole, post-mortem project reports usually contain too much project specific details. Therefore, the information in post-mortem project reports needs to be consolidated before it can be quickly understood and used throughout the rest of the organization. The consolidation of project evaluations is usually based on software metrics, since software metrics allow for easy consolidation.

There is a difference between the body of knowledge regarding the *elicitation* of knowledge during a project post-mortem review and the body regarding the *analysis* of post-mortem reviews to package this knowledge for the rest of the organization. An overview of the elicitation process, in which tacit knowledge is externalized into explicit knowledge (Nonaka and Takeuchi 1995) in project post-mortem reviews is given in (Dingsøyr 2005). The packaging of this tacit knowledge in project post-mortem reviews, so that it can be internalized by employees in other parts of the organization, is the topic of this paper.
Software metrics, both objective and subjective, can be used to evaluate projects. The advantage of objective metrics is that they do not require the judgment of an expert. However, their formal definitions usually require strict adherence to a measurement procedure and frequently require a lot of data to be collected and aggregated in order to measure the attribute. Without the required data no measurement for the attribute is possible, which explains why for many projects not all potentially useful objective metrics are available. Subjective measures on the other hand require no strict adherence to measurement rules, the judgment of an expert suffices. This explains the higher availability of subjective measurements of projects (Wohlin and Andrews 2001).

Even when resorting to subjective measurements for project attributes the distillation of knowledge from past experience is not easy. Without careful upfront (McIver and Carmines 1981) design of the subjective metrics, chances are that the scales of the subjective measurements are meaningless. On top of that potentially interesting project attributes are often missing from the metrics database. This leaves the analyst with missing data and measurements that are meaningless.

In this article we address the research question: “How can IT institutions learn from past project experiences when no objective nor subjective measurements have been performed on project aspects of interest?”

To solve the stated problems we propose a new method to explore potentially interesting relations present in project evaluations. Instead of limiting ourselves to just the project metrics database we propose to use the natural language post-mortem project reports as an additional source of data. Using concept hierarchy trees we are able to recode the
qualitative information in the post-mortem project reports into quantitative information. This quantitative information can be analyzed statistically to discover correlations between factors. This proposed method has been tested extensively in a case study involving 55 projects at the internal IT department of a large financial institution.

The remainder of this paper is structured as follows. In section 2, we discuss some related work in this area. We present our method in section 3 and section 4 describes the general concerns with the validity of findings found with our method. Section 5 contains a case study in which we applied the method to a substantial set of real-world post-mortem review reports. We end with our conclusions in section 6.

2 Related Work

Wohlin and Andrews (2001) have developed a method to evaluate development projects using subjective metrics about the characteristics of a project, collected using a questionnaire that can even be conducted at the end of the project. They have created a predictive model for certain success indicators, based on subjective metrics of project factors. Our work differs from their work in that our approach does not even require the collection of subjective measurements at the end of the project. Instead, we extract subjective metrics from qualitative data as found in post-mortem review reports.

As our method places lower demands on the required information, the method proposed in this paper might be applicable to an even wider range of projects than Wohlin and Andrews' method. On the other hand, as our data has less structure, our method results in a large percentage of missing data, which limits the analysis that can be performed on the
data (regression model building and principal component analysis are not feasible when our method is used).

Damele et al. (1996) have developed a method to investigate the root causes of failures during development. Their method uses questionnaires to obtain quantitative information on failures, which are analyzed using correlation analysis and Ishikawa diagrams. Their method differs from ours in that it uses Ishikawa diagrams to present the results of the analysis and we use the diagrams similar to Ishikawa diagrams as an intermediate structuring technique.

In (Van der Raadt et al. 2004) we used Grounded Theory to interpret and analyze data from a large set of semi-structured interviews with practitioners in the area of software architecture. The Grounded Theory method is a qualitative approach to inductively distil theory from a dataset (Glaser and Strauss 1967, Strauss and Corbin 1990). This approach is not meant to test an existing hypothesis, but provides a method for emerging a theory from collected data. The basic idea of Grounded Theory is to read and reread some textual database, and iteratively ‘discover’ and label a set of concepts and their interrelationships. In the research described in this paper, we apply a method related to Grounded Theory when populating the concept hierarchy trees.

3 A Method to Analyze Postmortem Data

Before we can describe the analysis method and insights the method attempts to spot, we first need to introduce some definitions. These definitions follow the definitions given by Wohlin and Andrews (2001). A factor is a general term for project aspects we would like to
study. Factors are either *project factors* or *success factors*. Project factors describe the status, quality, or certain characteristics of a project (e.g. as the testing tool used and team morale), and their value can either be determined prior to starting the project or during the execution of a project. Success factors capture an aspect of the outcome of a project (e.g. the timeliness of a project).

The method described below attempts to expose the effects of project factors on the success factors of a project. The method could try, for example, to discover the effect of using a certain programming language (which is a project factor) on the productivity of those projects (which is a success factor). Our method for discovering insights in natural language post-mortem evaluations consists of the steps listed in Table 1.

### 3.1 Identify Success Factors

To determine whether a project is a success or a failure, one needs to know what the important aspects (or factors) are of a project in the eyes of the stakeholders. Examples of success factors are timeliness, productivity and stability. Success factors can both be measured objectively and subjectively. The success factors we used in our case study are listed in Table 4.

In our article we have assumed that quantitative data is available for the key success factors metrics that have been defined. If the organization does not have quantitative data on the key success factors, one can let experts rank the projects based on the success factor (effectively measuring the success factor on an ordinal scale). If no consensus exists on the key success factors, one could start a separate analysis phase using step 3 and 4 to obtain quantitative assessments of the outcomes of a project.
3.2 Selection of Project Evaluations

Within the scope of an analysis, it usually will not be feasible to analyze all projects of which a post-mortem project evaluation is available. Therefore a sample of the available project evaluations needs to be drawn. Both random sampling and theoretical sampling can be used to select a sample of project evaluation reports. With random sampling chance determines if a post-mortem report is included in the sample. With theoretical sampling a report is not drawn blindly based on chance, but instead those reports are selected that offer the greatest opportunity of discovering relevant project factors. Although theoretical sampling has its own disadvantages, we prefer to use theoretical sampling over random sampling, as it increases our chances of finding noteworthy results.

As we use up-front theoretical sampling of project post-mortem reviews, we need to have a method to select promising post-mortem reviews. To achieve this, we first stratify the projects based on success factors identified in the previous step. The stratification process selects a proportion of projects that score high or low on the success factor and another proportion of projects that score average on the success factor. The stratification selects a disproportionately large group of projects that are an extreme case for one of the selected success factors, as these projects yield most information.

In the stratification process projects with the most extreme values on certain dimensions are found by selecting those projects that deviate more than $\lambda$ standard deviations from the average for that dimension. $\lambda$ is chosen such that a manageable number of projects are selected.
The stratification should in this case not lead to different conclusions than an analysis of a random sample, since stratification does not disrupt the coefficients in a regression equation model as long as the other assumptions of the regression model are not violated (Allison 2002).

3.3 Identify Project Factors

The open questions in project evaluations do not have numerical answers. We thus have to look for a transformation from the natural language texts available to numerical scores on project factors, which can subsequently be analyzed. We use concept hierarchy trees to bring structure in the remarks in the project evaluations to find the underlying project factors in the answers to the open questions. The project factors that are identified as relevant in this step, will be used in the next analysis step to code all selected project evaluations.

Concept hierarchy trees, which have a great resemblance with fish-bone or Ishikawa diagrams (Ishikawa 1984), organize individual remarks made in project post-mortem reviews. The difference between Ishikawa diagrams and concept hierarchy trees is that Ishikawa diagrams focus on the effect of project factors on a single specified success factor, whereas concept hierarchy trees merely organize project factors that have an influence on any of the identified success factors.

Concept hierarchy trees do not indicate what effect the project factors have on the success factor. Remarks that identify similar project factors are located near each other in the tree, whereas remarks that refer to different project factors are placed in separate positions in the concept tree. To synthesize the concept hierarchy tree, we take a sample (say, 30%) from
the set of selected project evaluations. The content of these selected project post-mortem reviews is used to populate the concept hierarchy tree. To synthesize the concept hierarchy tree it is usually not required to examine all the project evaluations that will be analyzed in the subsequent analysis step, as we expect relevant project factors to occur frequently.

To synthesize a concept hierarchy tree based on a set of selected project post-mortem reviews, one starts by breaking the review text into lines or paragraphs each describing a single project factor that influences project outcomes. In our case study review, the text originates from the open questions listed in Table 2. The mapping of lines to project factors is not always one-on-one, sometimes multiple lines describe a single project factor, sometimes a single sentence can describe multiple project factors. This process of breaking the text into single topic remarks is called open coding in Grounded Theory. We should be careful not to include descriptions of success factors in the concept hierarchy tree, as we do not want to confuse cause and effect.

The result of breaking the post-mortem review up into single topic remarks is a large list of descriptions of project factors. This list likely contains duplicate entries for a single project factor and contains both overly broad remarks and very project specific remarks regarding the project factors. To eliminate duplicates in the list of project factors and to obtain a list of project factors of a roughly similar level of abstraction, the individual remarks need to be organized. To do so, the individual remarks are placed into a concept hierarchy tree in such a way that similar remarks end up near one another (share more ancestors in the concept hierarchy tree). This organization of remarks by comparing individual remarks and
grouping similar remarks together is called \textit{comparative analysis} and the whole process of hierarchically ordering remarks is called \textit{axial coding} in Grounded Theory.

To facilitate the ordering of the remarks in our case study we started with a predetermined set of four main categories: processes, tools, people, and environment, see figure 1. Our top-level structure was derived from a classification from Balanced Score Card (Kaplan and Norton 1996) schemes used elsewhere within the company. If no such structure is available up front, it can be obtained as a byproduct of a Grounded Theory-like investigation. Researchers who are familiar with Grounded Theory may have noted that our main classification of remarks into project factors and success factors and our subsequent classification of remarks in four main categories differs from the \textit{paradigm model} suggested by Strauss and Corbin (1990, p. 99). The rationale for this difference is that our classification scheme is specialized for software engineering research and that the absence of intermediate constructs (such as causal conditions that cause a phenomenon) makes the statistical analysis of the gathered data easier. Models including intermediate constructs require complex factor analytical analyses, whereas with our approach correlation coefficients suffice.

After all remarks have been placed in the concept hierarchy tree, nodes of conceptually similar remarks are replaced by a node containing a keyword or a short statement that describes the underlying project factor. To avoid ending up with overly specific or overly broad concepts we have used the following rule of thumb: each keyword must be observed in at least 20\% and at most 50\% of the project post-mortem reviews under study. Overly specific or overly broad concepts can hamper the subsequent analysis steps as we will only
be able to determine the effect of the project factor on the success factors if the presence/absence of project factors varies. Only discriminating concepts are interesting as we can't learn much from the observation that, say, “user involvement is a crucial factor” if this remark is made for almost all projects.

The keywords from the adjusted concept hierarchy tree describe the answers to the open questions in a project post-mortem review. Since these keywords classify the answers to the questions, and not the questions themselves, one should not expect a one-to-one relation between the open questions in the post-mortem report and the keywords in the concept hierarchy tree. For reasons of space (we distinguished over 80 categories in the case study), the resulting list of categories is not included in the paper. Figure 1 contains a few typical sub categories (such as: test tooling used) and subsubcategories (such as: cooperation in IT team) we found during the case study.

3.4 Interpret Project Evaluations

After the keywords and patterns have been distilled from the project evaluations, the subjective interpretation can start. During the subjective interpretation step, selected project evaluations are read by the researcher. This researcher interprets in which categories from the concept hierarchy tree the remarks from the evaluation fit. Next, it is determined whether the project scores positive or negative on this category. For example, one of the categories we distinguished is change management. A project evaluation report may for instance contain phrases that pertain to the notion of change management. From these phrases, the researcher may deduce that change management has been implemented well for the project, or that problems with respect to change management occurred during the
project. Using Likert scaling, we transform natural language text in the answers to open questions into numerical information. For each project, the scores of each category are included in a spreadsheet, together with information from other sources on project aspects. This leaves us with a concise numerical characterization of each project.

3.5 Analyze Correlations

The interpretation of the post-mortem project evaluations yields a spreadsheet with a quantitative encoding of the information from the project evaluation database. As said above, this spreadsheet may be coupled with other numerical information from the project administration database. Using a statistical package, we may next determine correlations between the project evaluations on the one hand, and other quantitative information on the other hand. In our case study, we had quantitative information on productivity, conformance to budget and schedule, and satisfaction of different sets of stakeholders.

The matrix that results from the subjective interpretation of the project evaluations unfortunately has a high number of variables in relation to the number of observations. Normally we would use a technique such as principal component analysis to reduce the number of variables. However in our case a large percentage of the data is missing, which makes principal component analysis infeasible.

Instead of first reducing the number of variables in the data matrix, we directly measure the correlation between the project factors (the independent variables) and the success factors (the dependent variables). This leads to a matrix of correlations between project characteristics and success indicators. We used Kendall's tau (Liebetrau 1983) measure for the correlation, since this measure is suited for ordinal data. We use pair-wise deletion
when encountering missing data, instead of list-wise deletion (Allison 2002), to make optimal use of the available data.

The correlation coefficients in the matrix are not all based on the same number of observations. Correlation coefficients that are based on a larger number of observations offer more certainty that the observed correlation is really present in the underlying population. This certainty is shown in the level of significance of the correlation coefficient, which can be calculated by statistical packages (Siegel 1956).

Rather than examining the correlation coefficients for all identified project factors, we may opt to restrict ourselves to observing the most significant coefficients. The sheer amount of numbers in the correlation matrix can distract attention away from the most influential project factors. Several selection criteria can be used to reduce the size of the correlation matrix.

The most straightforward selection criterion for rows in the correlation matrix is to select only those rows that contain a single correlation coefficient that has the highest significance level. An alternative to this criterion is to look for rows that contain the highest or lowest single correlation coefficient and not to look at the significance of the coefficient. More advanced selection criteria can look at the overall effect of a project factor on all success factors. To achieve this one can either select those rows that have the maximum squared average correlation coefficient or that have the highest average significance level for the correlation coefficients. Even more sophisticated criteria can take weight factors for each success factor into account to average the overall correlation coefficients or significance levels.
Note that the statistical significance observed in this type of analysis often is not very high, due to the small sample sizes. As we make multiple comparisons we should apply a Bonferonni or Sidak correction to compensate for the multiple comparisons if we want to use the technique as a confirmatory instead of a exploratory technique. As the statistical significance of the results is rather low, we need to have a theory for the correlations observed, in order to derive useful information. Correlation by itself does not imply the direction of the causality.

4 Threats to Validity

The described method to find correlations between project factors and success factors is by nature an exploratory analysis technique to find starting points for software process improvements. Before actual software process improvements activities are undertaken, the correlations found should be verified in a conformational study.

Having explained that exploratory analysis techniques should not be used to gather evidence, but instead to find starting points for new research, it is still useful to examine which sources of noise and bias could potentially influence the results of the analysis. These sources of noise and bias can lead to spurious correlations where in reality there exist none. Understanding how the validity of a study can be impaired is useful in trying to prevent these problems from occurring.

In this section we will only discuss the method specific threats to the validity of outcomes of studies using the method of analysis described this paper. Generic threats to the validity of an experiment will not be discussed as the subject is too broad to describe in a single
paper. There are however excellent books that explain generic threats to the validity of a study, see for example the seminal work by Cook and Campbell (1979).

4.1 Representation Condition of Evaluation Reports

One of the strengths of (partially) open-ended project post-mortem evaluation processes is their ability to discern success or failure conditions that were not anticipated during the design of the development process and the evaluation process. Being able to detect unanticipated sources of failure could help in preventing unforeseen causes of problems to be ignored until they cause irreparable damage to the software product of project.

Open-ended evaluation processes are, however, a double-edged sword. While the process allows unanticipated remarks, open-ended reviews depend on the subjective evaluation of the project's success and failure source by the individual project members. Subjective evaluations in general are known to be less dependable than objective evaluations. The quality of the analysis that is based on these open-ended reviews can only be as good as the quality of project post-mortem reports. If the project post-mortem reports do not offer an accurate view of the project reality, the result of the analysis will also not reflect that reality.

Sometimes project post-mortem reviews do not reflect reality because the project members do not have a good overall picture of the entire project or because the project members have a vested interest in not accurately depicting the project. Most project members are only involved in part of the project or their tasks might only involve one aspect of the entire project. From the point of view of a single project member certain aspects of the project might not seem relevant or productive (for example change control or requirements
management) because the benefits of those actions are not for the person involved, but are none-the-less useful for other stakeholders in the project. This could lead to project post-mortem reviews indicating problems when in reality there exist none. Another potential source of disturbance is that for certain project members it could be interesting to not give an accurate description of the project. Even if the project manager knows that the planning was mediocre and caused downstream problems in the project, he manager might not mention this fact if he feels that this revelation could have negative consequences for him. Conflicts of interest of the participants of a project could cause biased findings in the final analysis.

As project post-mortem reviews are the only source of information, there is no external oracle that can be used to validate the contents of the project post-mortem report. Although no external validation source exists, there do exist internal sources that can be used to validate the findings. If other project members agree upon the findings of the project evaluation, this corroborates the validity of the findings. Especially if the participants of the project post-mortem review have different/conflicting interests, this will decrease the chance of manipulation of the review findings. Still, the dependence on subjective interpretations of the project reality remains a weak point in the entire technique.

4.2 Validity of Identified Project Factors

Not only the project members can cause noise and bias in the findings of the study, also the researchers can introduce bias in the findings of the study. Two points of concern are the identification of project factors and the interpretation of project evaluations. If the identification of project factors is flawed, the resulting concept hierarchy tree will be
invalid, causing problems in interpreting and coding the individual project post-mortem reports during the interpret project evaluations step. Even with a solid concept hierarchy tree the researchers can misunderstand project post-mortem reports or make mistakes in coding the reports for later analysis.

Although the exact form of the concept hierarchy tree is not of direct influence on the final analysis results, it does have a major impact on the subsequent interpretation of the project evaluations. Therefore it is crucial that the concept hierarchy tree is a useful representation of the project factors and that all remarks from the open coding phase can be placed in the concept hierarchy tree with ease.

To gain insight in the quality of the concept hierarchy tree, two researchers can independently create that tree and later compare them with each other to see where there are points that require clarification. To formally compare the resulting concept hierarchy trees of the two researchers, one can use mathematical techniques used in systematic biology to compare taxonomy trees. For more information we refer to the article of Penny and Hendy (1985).

### 4.3 Validity of Project Evaluation Interpretation

Even with a proper concept hierarchy tree researcher can and will misunderstand project post-mortem reports. This will lead to these mistakes being encoded into the data matrix, which will lead to invalid results of the correlation analysis.

There are two strategies that can be used to eliminate this problem. The first method involves letting the original project members code the project post-mortem review under
the supervision of the researcher. The project member uses the supplied concept hierarchy tree to classify and code the post-mortem report. As the project member has a deeper understanding of the project, it is less likely that the statements in the project post-mortem report are misunderstood. This approach has been used in our case study, where a project support officer from the organisation in which the study took place, assisted in coding the project post-mortem reviews.

A second strategy to eliminate this problem is to let two researchers code each post-mortem review and to compare the interpretations of each project post-mortem review. If two or more researchers arrive at the same conclusion, this strengthens the results. As full overlap in interpretation of each post-mortem report is unlikely, Cohen's kappa (Cohen 1960) coefficient can be calculated to determine the degree of agreement in interpretation of both researchers. This coefficient gives an impression of how valid the analysis results will be.

5 Case Study

In this section we discuss a case study in an organization in which we performed the method described above.

5.1 Context of Case Study

This study has been performed within an internal Information Technology department of a large financial institution. This department employs over 1500 people. The organization primarily builds and maintains large, custom-built, mainframe transaction processing systems, most of which are built in COBOL and TELON (an application-generator for COBOL). Besides these mainframe systems, an extensive variety of other systems are
constructed, implemented, and maintained by the organization. These systems are implemented in various programming languages (such as Java and COOL:Gen), run under a variety of operating systems (such as Microsoft Windows and UNIX) and are distributed over different platforms (batch, block-based, GUI-based and browser-based).

The organization’s agile development process is based on the Dynamic Systems Development Method (Stapleton 2002). In addition to the DSDM framework, the organisation has added its own practices to the development process an the quality system, in order to meet the demands of Level 2 of the SW CMM (Paulk et al. 1993).

5.2 Evaluation Process

The organization has developed its own post-mortem project evaluation method. The evaluation method consists of an online administered questionnaire composed of both open and closed questions. In the evaluation process three groups of stakeholders are addressed: the customer who has commissioned the project, the IT personnel that participated in the project and the involved middle management of the IT department.

At the end of each development project a mandatory evaluation cycle is initiated by the IT project office of the organization. Upon request of the project office the project leader invites involved stakeholders by e-mail to fill out the evaluation questionnaire. When a sufficient number of stakeholders has filled out the questionnaire, the project leader activates an evaluation consolidation routine in the evaluation program, which anonymizes the responses and calculates averages of all the closed questions.
The evaluation questionnaire contains both open and closed questions. The open questions in the questionnaire are listed in Table 2. To get an understanding in what kind of answers are given in response to the open questions, an excerpt from a project post-mortem report is given in Table 3. As there are over 150 closed questions in the questionnaire, only the categories used to group the closed questions are included in this paper. The categories of the closed questions are: Time Management, Risk Management, Project results, Project Task (Work), Organization, Work environment, Quality/scope, Project management, and Information, Project. The categories of the closed questions originate from a Balanced Score Card (Kaplan and Norton 1996) initiative that has been conducted at the organization.

5.3 Bottlenecks

Although the organization has invested large amounts of time in both developing the original evaluation method and evaluating the projects themselves, the organization was unfortunately not able to fully benefit from the lessons learned that lie hidden in the more than 600 evaluation reports that are stored in the project evaluation database.

The organization used the complete information in the project evaluation only as feedback to the project leader responsible for the project. The organization outside the project used the project evaluations only as Balanced Score Card indicators of satisfaction of the commissioning customer and the satisfaction of the IT employees. These overall, consolidated satisfaction ratings made it hard to pinpoint what is going wrong when e.g. the employee satisfaction drops.

The inability to use the project evaluation database can be attributed to four major reasons:
- The database system containing the evaluation database was not available electronically. Manual copying of the data from the database to a statistical tools was required. This made analysis a labour-intensive task.

- As the evaluation method included open questions, some of the answers contained textual answers instead of quantitative data. Textual data is inherently harder to analyze than quantitative data.

- The wording and grouping of the closed questions was inadequate. The grouping of the questions, that was derived from the Balanced Score Card items, was such that many of the categories measured simultaneously different concepts, which makes the interpretation of the average on a category infeasible.

- As the individual answers on closed questions contribute to the average customer, employee or management satisfaction, one cannot state that the scores on individual questions are independent from the satisfaction measurements. These scale-subscale dependencies make the interpretation of correlation coefficients difficult at least.

The low quality of the closed questions and their clustering combined with the problem of scale-subscale correlations led to the decision to extract project characteristics from the answers to the open questions, using the method outlined in the previous section. Analyzing the open questions had as an added advantage that the answers from every respondent were available, which gave insight into the degree to which the stakeholders in the project agreed on certain issues.
5.4 Experiences

For the analysis of the project information database 55 project evaluations have been selected out of a database containing over 600 project evaluations. The selection of projects included ‘normal projects’, projects with a specific programming environment, and projects that deviated on: productivity, conformance to budget or conformance to schedule. For the deviant projects, an equal ratio of over-performing and underperforming projects has been selected. The exact distribution of the number of projects on the selection criteria is given in Table 4.

5.5 Results

The result of the analysis steps performed in the case study can be found in Table 5. The statistical analyses have been performed with dedicated software written for statistical environment R version 1.9.1 (R Development Core Team 2004). The table contains both the Kendall's tau correlation coefficients between project factor and success factor, as well as the p-values of those correlation coefficients. The correlation coefficients indicate if there is a strong positive (1), or negative (-1) relation between the factors, or no relation (0). The p-value indicates the strength of the evidence of the relation between the factors, varying form 0 (indicating very strong evidence) to 1 (indicating very weak evidence).

To reduce the correlation matrix we have sorted the factors with respect to the highest overall level of significance. Having sorted the project factors we selected the top 20% from this list.

The analysis has given us a lot of insight into the quality and organization of the set of closed questions used so far, and suggested a number of additional closed questions to ask.
For each of the project factors that was in the selected top 20% of most relevant project factors a question has been added to the updated project evaluation questionnaire, so that, for future evaluations, more quantitative information will be directly available.

At a concrete level, the study showed some interesting, albeit weak, relations between project characteristics and success indicators. For example, high productivity occurs frequently when there is a good cooperation within the team, and when the infrastructure architecture is elaborated well. These relations need further study, though.

6 Conclusions

We have presented a method to analyze qualitative, natural language information as often encountered in post-mortem project reports. The method has five steps: identify success factors, select project evaluations, identify project factors, interpret project evaluations, analyze correlations.

This method provides a structured way to deal with qualitative information such as present in post-mortem project reviews. This information can next be related to other, quantitative, information present in the company's project database, such as information related to schedule and cost. Information gained from this analysis can be used to improve closed questionnaires that might also be in use, and it gives additional clues that provide useful guidance in a process improvement initiative.

Note that the statistical significance observed in this type of analysis varies, due to the exploratory nature of the analysis. So we need to have a theory for the correlations
observed or confirm the results using experiments. Correlation by itself does not imply causality.

**Acknowledgments**

This research is supported by ABN AMRO Bank N.V. We thank the ABN AMRO Bank for her cooperation. We are especially grateful to Jean Kleijnen, Ton Groen and Cosmo Ombre for their valuable comments and input. We also appreciate the comments of Geurt Jongbloed (Department of Stochastics at the Vrije Universiteit).

**References**


Figure 1: Example of concept hierarchy tree containing project characteristics.
Table 1: Analysis process to discover relations in post-mortem reports.

<table>
<thead>
<tr>
<th>Process steps</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Identify success factors:</strong> Identify the factors that determine the success of a project in the eyes of the stakeholders.</td>
</tr>
<tr>
<td><strong>Select project evaluations:</strong> Select project evaluations for further analysis. To obtain meaningful results, one might select projects with extreme values along certain success factors.</td>
</tr>
<tr>
<td><strong>Identify project factors:</strong> Identify repeating patterns in project factors by screening a subset of the selected projects (from step Select project evaluations). These repeating patterns will be structured using an concept hierarchy tree.</td>
</tr>
<tr>
<td><strong>Interpret project evaluations:</strong> Read and interpret all project evaluations (from step Identify project factors) using the concept hierarchy tree created in the previous step. After the interpretation, the project is evaluated on the project factors that are present in the concept hierarchy tree.</td>
</tr>
<tr>
<td><strong>Analyze correlations:</strong> Analyze correlations between project factors and success factors. Sort through the correlations between the project factors to find interesting results.</td>
</tr>
</tbody>
</table>
Table 2: Case study: Open questions in project evaluation questionnaire.

<table>
<thead>
<tr>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>- What are the 3 most positive points of the project? Explain them.</td>
</tr>
<tr>
<td>- What are the 3 most important learning points of the project? Explain them.</td>
</tr>
<tr>
<td>- Can you give 3 suggestions by which the project could have been carried out (even) better?</td>
</tr>
<tr>
<td>- Was there sufficient input documentation at the beginning of the functional design phase? Which inputs were not available? Indicate the reasons.</td>
</tr>
<tr>
<td>- Which testing tools were used? What were the advantages and disadvantages?</td>
</tr>
<tr>
<td>- Was test ware available? If not, what were the reasons?</td>
</tr>
<tr>
<td>- Which configuration management tools were used? What were the advantages and disadvantages?</td>
</tr>
</tbody>
</table>
**Table 3:** Excerpt from an actual project post-mortem evaluation report.

<table>
<thead>
<tr>
<th>Project Post-mortem</th>
</tr>
</thead>
<tbody>
<tr>
<td>What are the 3 most positive points of the project? Give your explanation about them.</td>
</tr>
</tbody>
</table>

**Project Manager**
- clear business drive and support
- changes could be put trough quite quickly by flexible project approach. By this compensating the relatively high amount of changes in requirements
- thorough planning and monitoring of this and related projects has proved to be key in delivering results during busy periods.

**Project Leader**
- Deliverables/FCR output data provide real added value with regard to financial management reporting; ABN ESVP's and MB are interested and involved. They use the FCR reports for analysis.
- Good co-operation with the project team members
- Delivered on time & within budget

**Customer**
- quality of delivery
- respected the time schedule
- communication

**Customer**
- think the dedication of the team during development as well as on follow-up issues was very good
- Good resources within the team
Table 4: Case study: Selection criteria for inclusion of project evaluations.

<table>
<thead>
<tr>
<th>Selection criterion</th>
<th>Number of projects selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extreme budget under spending/over-spending</td>
<td>10</td>
</tr>
<tr>
<td>Extreme planning deviations finished early and late)</td>
<td>8</td>
</tr>
<tr>
<td>Extremely high and low productivity</td>
<td>10</td>
</tr>
<tr>
<td>Using COBOL/TELON programming environment</td>
<td>4</td>
</tr>
<tr>
<td>Using Java programming environment</td>
<td>4</td>
</tr>
<tr>
<td>Using Cool:Gen programming environment</td>
<td>4</td>
</tr>
<tr>
<td>Average productivity, no extreme budget or planning deviations</td>
<td>15</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>55</strong></td>
</tr>
</tbody>
</table>
Table 5: Case study: Results of the correlation analysis on the evaluation matrix.

<table>
<thead>
<tr>
<th>Factor name</th>
<th>Conformance to</th>
<th>Satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Productivity</td>
<td>Budget</td>
</tr>
<tr>
<td>change management</td>
<td>0.05</td>
<td>-0.18</td>
</tr>
<tr>
<td>project management</td>
<td>-0.13</td>
<td>0.12</td>
</tr>
<tr>
<td>quality planning</td>
<td>-0.16</td>
<td>0.13</td>
</tr>
<tr>
<td>quality schedule</td>
<td>-0.41</td>
<td>0.24</td>
</tr>
<tr>
<td>project control</td>
<td>-0.28</td>
<td>0.17</td>
</tr>
<tr>
<td>test-ware reused</td>
<td>-0.11</td>
<td>0.50</td>
</tr>
<tr>
<td>quality infrastructure architecture</td>
<td>0.52</td>
<td>0.37</td>
</tr>
<tr>
<td>communication efficiency</td>
<td>0.01</td>
<td>-0.3</td>
</tr>
<tr>
<td>cooperation</td>
<td>0.08</td>
<td>-0.20</td>
</tr>
<tr>
<td>cooperation within IT</td>
<td>0.39</td>
<td>-0.13</td>
</tr>
<tr>
<td>appropriateness team size</td>
<td>0.12</td>
<td>-0.51</td>
</tr>
<tr>
<td>team stability</td>
<td>-0.58</td>
<td>0.14</td>
</tr>
<tr>
<td>team stability organization</td>
<td>-0.27</td>
<td>0.29</td>
</tr>
<tr>
<td>test-tool expediter used</td>
<td>-0.27</td>
<td>0.29</td>
</tr>
</tbody>
</table>