Discovering the Relation between Project Factors and Project Success in Post-mortem Evaluations

Joost Schalken[1], Sjaak Brinkkemper[2], and Hans van Vliet[1]

[1] Vrije Universiteit, Department of Computer Science,
De Boelelaan 1083a, 1081 HV Amsterdam, The Netherlands
{jjp.schalken,jc.van.vliet}@few.vu.nl
[2] Utrecht 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 industrustrial software process improvement initiative.

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.

Two kinds of project evaluations can be distinguished [1]: *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 [1] and to monitor project risks [2]. 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"* [1]. 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 [3]. 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.

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 [4].

Even when resorting to subjective measurements for project attributes the distillation of knowledge from past experience is not easy. Without carefull upfront [5])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.

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 Ishikawa or fishbone diagrams [6] 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 invovling 55 projects at the internal IT department of a large financial institution.

The remainder of this paper is structured as follows. In Sect. 2, we discuss some related work in this area. We present our method in Sect. 3, and Sect. 4

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 Sect. 5.

2 Related Work

Wohlin and Andrews [4] 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. have developed a method to investigate the root causes of failures during development [7]. 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 as an intermediate structuring technique.

In [8] 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 distill theory from a dataset [9]. 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 fishbone diagrams [6].

3 Method

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 [4]. 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

Table 1. Analysis process to discover relations in post-mortem reports.

Process steps
Identify success factors Identify the factors that determine the success of a project in the eyes of the stakeholders.
Select project evaluations Select project evaluations for further analysis. To obtain meaningful results, one might select projects with extreme values along certain success factors.
Identify project factors 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 Ishikawa diagram.
Interpret project evaluations Read and interpret all project evaluations (from step Identify project factors) using the Ishikawa diagram created in the previous step. After the interpretation, the project is evaluated on the project factors that are present in the Ishikawa diagram.
Analyze correlations Analyze correlations between project factors and success factors. Sort through the correlations between the project factors to find interesting results.

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 3.

3.2 Selection of Project Evaluations

Within the scope of an analysis, it might not be feasible to analyze all projects of which a post-mortem project evaluation is available. Therefore we first need to 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

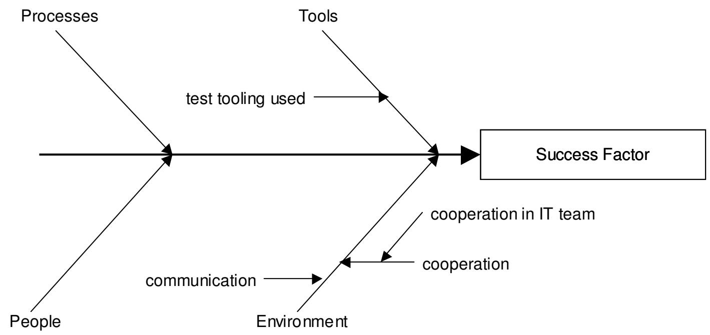

Fig. 1. Example of fishbone diagram containing project characteristics.

on the 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 X standard deviations from the average for that dimension. X is chosen such that the manageable number of projects are selected.

The stratification should not lead to different conclusions than an analysis of a random sample, since stratification does not disrupt the coefficients in a regression equational model as long as the other assumptions of the regression model are not violated [10].

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 fishbone diagrams 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.

Fishbone diagrams, also known as Ishikawa diagrams, are a well-known technique from quality management to detect the causes of disruptions and production problems [6]. To fill the fishbone diagram, we take a sample (say, 30%) from the set of selected project evaluations. From this small set, we select those keywords which best characterize the project factors in the project evaluations. To identify which project factors are relevant we do not need to examine the full set of projects that will be used in the next analysis step, as we expect relevant project factors to occur frequently.

Next, we replace keywords that describe the same pattern by a more general keyword, one which represents all observations from that category. As a rule of thumb for the granularity of keywords we use the following: each keyword must be observed in at least four projects, and at most 50% of the projects must score either positive or negative on this keyword. The reason for the latter restriction is that only keywords which discriminate projects are interesting. 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 found are placed in a fishbone diagram. The main categories we used to organize the diagram in the case study are: processes, tools, people, and environment, see Fig. 1. Our top-level structure was derived from a classification from Balanced Score Card [11] 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. The keywords describe the answers to the open questions listed in Table 2. Since these keywords classify the answers to the questions, and not the questions themselves, these keywords do not have a one-to-one relation to the open questions. For reasons of space (we distinguished over 80 categories in the case study), the 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 used. Please note that the organization of the fishbone diagram only reflects the classification of the remarks in the project evaluations and does not imply any causal links between project factors and success factors.

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 fishbone diagram 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 [12] 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 [10], 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 [13].

Rather than examining the correlation coefficients for all identified project factors, we may opt to restrict ourselves to the most significant coefficients. As the sheer amount of numbers in the correlation matrix can distract attention from the most influential project factors. The factors can either be selected based on the highest overall level of significance or based on the highest level of significance on a single factor. To calculate the overall significance of a project factor the average is taken of the squared correlation coefficients between a project factor and all success factors.

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 conformatory 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 Case Study

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

Table 2. Case study: Open questions in project evaluation questionnaire.

Question
– What are the 3 most positive points of the project? Explain them. – What are the 3 most important learning points of the project? Explain them. – Can you give 3 suggestions by which the project could have been carried out (even) better? – Was there sufficient input documentation at the beginning of the functional design phase? Which inputs were not available? Indicate the reasons. – Which testing tools were used? What were the advantages and disadvantages? – Was test ware available? If not, what were the reasons? – Which configuration management tools were used? What were the advantages and disadvantages?

4.1 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).

4.2 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 programme, 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. 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[11] initiative that has been conducted at the organization.

4.3 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.

Table 3. Case study: Selection criteria for inclusion of project evaluations.

Selection criterion	*Number of projects selected*
Extreme budget under spending/overspending	10
Extreme planning deviations (finished early and late)	8
Extremely high and low productivity	10
Using COBOL/Telon programming environment	4
Using Java programming environment	4
Using Cool:Gen programming environment	4
Average productivity, no extreme budget or planning deviations	15
Total	55

4.4 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 overperforming and underperforming projects has been selected. The exact distribution of the number of projects on the selection criteria is given in Table 3.

4.5 Results

The result of the analysis steps performed in the case study can be found in Table 4. 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 coeffients 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 sofar, and suggested a number of additional closed questions one might ask. This will be used to update and improve the 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.

Table 4. Case study: Results of the correlation analysis on the evaluation matrix.

		Conformance to			*Satisfaction*		
Factor name	*Productivity*	*Budget*	*Schedule*	*Duration*	*Management*	*Employee*	*Customer*
change management	0.05 p=0.71	-0.18 p=0.16	-0.20 p=0.11	0.20 p=0.11	0.33 p=0.006	0.51 p<0.001	0.40 p=0.004
project management	-0.13 p=0.35	0.12 p=0.32	0.39 p< 0.001	-0.26 p=0.02	0.39 p<0.001	0.45 p<0.001	0.10 p=0.42
quality planning	-0.16 p=0.30	0.13 p=0.36	0.34 p=0.01	-0.20 p=0.14	0.43 p=0.001	0.27 p=0.04	0.10 p=0.48
quality schedule	-0.41 p=0.02	0.24 p=0.13	0.23 p=0.13	-0.06 p=0.69	0.10 p=0.52	0.29 p=0.06	-0.19 p=0.30
project control	-0.28 p=0.08	0.17 p=0.25	0.29 p=0.04	-0.34 p=0.02	0.08 p=0.56	0.39 p=0.008	-0.33 p=0.04
testware reused	-0.11 p=0.66	0.50 p=0.02	0.53 p=0.008	-0.38 p=0.06	-0.17 p=0.39	0.39 p=0.05	0.20 p=0.43
quality infrastructure architecture	0.52 p=0.05		0.37 p=0.09	-0.33 p=0.13	0.50 p=0.02	0.08 p=0.70	-0.24 p=0.36
communication efficiency	0.01 p=0.94	-0.36 p=0.06	-0.26 p=0.14	0.30 p=0.10	0.16 p=0.38	0.33 p=0.07	0.25 p=0.20
cooperation	0.08 p=0.58	-0.20 p=0.13	0.25 p=0.06	-0.26 p=0.04	0.22 p=0.08	0.40 p=0.002	0.27 p=0.08
cooperaton within IT	0.39 p=0.03	-0.13 p=0.45	0.46 p=0.006	-0.24 p=0.15	0.59 p<0.001	0.20 p=0.23	0.44 p=0.04
appropriateness team size	0.12 p=0.71	-0.51 p=0.11	-0.93 p=0.004	0.84 p=0.008	-0.14 p=0.65	-1.00 p=0.005	-1.00 p=0.005
team stability	-0.58 p=0.10	0.14 p=0.62	-1.00 p<0.001	0.50 p=0.08	-0.36 p=0.21	0.36 p=0.21	0.18 p=0.56
team stability organisation		0.25 p=0.31	0.36 p=0.13	-0.13 p=0.58	-0.61 p=0.009	-0.27 p=0.24	-0.57 p=0.02
testtool expediter used	-0.27 p=0.03	0.29 p=0.01	0.58 p=<0.001	-0.34 p=0.001	-0.03 p=0.76	0.11 p=0.30	-0.11 p=0.35

5 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

1. Jurison, J.: Software project management: The manager's view. Communications of AIS **2** (1999)
2. McConnell, S.: Rapid Development: Taming Wild Software Schedules. Microsoft Press, Redmond, WA, USA (1996)
3. Seaman, C.B.: Opt: Organization and process together. In: Proceedings of the 1993 conference of the Centre for Advanced Studies on Collaborative research, IBM Press (1993) 314–324
4. Wohlin, C., Andrews, A.A.: Assessing project success using subjective evaluation factors. Software Quality Control **9** (2001) 43–70
5. McIver, J.P., Carmines, E.G.: Unidimensional Scaling. Volume 07-024 of Sage University Paper series on Quantitative Applications in the Social Sciences. Sage Publications, London, UK (1981)
6. Ishikawa, K., ed.: Quality control circles at work. Asian Productivity Organization, Tokio, JP (1984)
7. Damele, G., Bazzana, G., Andreis, F., Arnoldi, S., Pess, E.: Processe improvement through root cause analysis. In Bologna, S., Bucci, G., eds.: Proceedings of the 3rd IFIP International Conference on Achieving Quality in Software (AQuIS'96), IFIP Working Group 5.4 (1996) 35–47
8. van der Raadt, B., Soetendal, J., Perdeck, M., van Vliet, H.: Polyphony in architecture. In: Proceedings of the 26th International Conference on Software Engineering (ICSE2004), IEEE Computer Society Press (2004) 533-542.
9. Glaser, B.G., Strauss, A.L.: The Discovery of Grounded Theory: Strategies for Qualitative Research. Observations. Weidenfeld and Nicolson (1967)
10. Allison, P.D.: Missing Data. Volume 07-136 of Sage University Paper series on Quantitative Applications in the Social Sciences. Sage Publications, Thousand Oaks, CA, US (2002)
11. Kaplan, R.S., Norton, D.: The Balanced Scorecard. The Harvard Business School Press, Boston, MA, USA (1996)
12. Liebetrau, A.M.: Measures of Association. Volume 07-032 of Sage University Paper series on Quantitative Applications in the Social Sciences. Sage Publications, London, UK (1983)
13. Siegel, S.: Nonparametric Statistics for the Behavioral Sciences. McGraw-Hill Series in Psychology. McGraw-Hill Book Company, New York, NY, USA (1956)