C. Wohlin and A. Amschler Andrews, "Assessing Project Success using Subjective Evaluation Factors", Software Quality Journal, Vol. 9, No. 1, pp. 43-70, 2001.

Assessing Project Success using Subjective Evaluation Factors

Claes Wohlin
Dept. of Software Engineering and Computer Science
Blekinge Institute of Technology
Box 520
SE-372 25 Ronneby, Sweden
+46 457 38 58 10
claes.wohlin@ipd.hk-r.se

Anneliese Amschler Andrews
Computer Science Department
Colorado State University
Fort Collins, CO 80523-1873
USA
+1 970 491 7016
aaa@cs.colostate.edu

ABSTRACT

Project evaluation is essential to understand and assess the key aspects of a project that make it either a success or failure. The latter is influenced by a large number of factors, and many times it is hard to measure them objectively. This paper addresses this by introducing a new method for identifying and assessing key project characteristics, which are crucial for a project's success. The method consists of a number of well-defined steps, which are described in detail. The method is applied to two case studies from different application domains and continents. It is concluded that patterns are possible to detect from the data sets. Further, the analysis of the two data sets shows that the proposed method using subjective factors is useful, since it provides an increased understanding, insight and assessment of which project factors might affect project success.

Keywords

Project success, subjective measures, project assessment.

1. Introduction

Software project success is influenced by many project characteristics. They include stability of requirements, knowledge of developers and management, inherit difficulty of the project, techniques and tools used (and how appropriate they are) to name just a few. The project characteristics influence project outcome and success indicators. Success indicators include timeliness of delivery, quality of software, as well as more long term properties such as maintainability.

To complicate matters further, it is usually not obvious how these project characteristics interact. In addition, it may be difficult or impossible to develop and use objective quantitative measures for many of these project characteristics and success indicators. Even when quantitative measures exist, they may not be usable for a variety of reasons such as inconsistent measurement and lack of priorities in selecting measures. This can lead to not measuring important project characteristics or success indicators at all. It is often quicker and easier to collect subjective measures during or after project completion. Not surprisingly, many software project databases (for example the NASA-SEL database) contain a large number of subjective project variables and success variables. It should be noted that even if a variable may be measured objectively, as for example delivery precision, it may be chosen to simply determine how good the precision is on a five grade scale.Thus, we would like to view subjective variables as those that are assigned a non-objectively measured value.

Subjective evaluations or expert judgement are not used that often in software engineering. A possible explanation may be the problems related to formulating suitable scales and also to collect trustworthy data, which is addressed by for example (Valett 1993). Some exceptions exist, primarily in relation to effort estimation, see for example (Gray et. al. 1999) and (Höst and Wohlin 1998), and also lately in risk management (Ropponen and Lyytinen 2000). The use of expert judgement as an estimation method is further discussed in (Hughes 1996). To the best of our knowledge subjective variables have not been used to map project characteristics to project success as proposed in this paper.

Several questions should be addressed in the analysis of subjective factors:

- Which project characteristics drive a specific success indicator?
- How do different success indicators relate to each other?
- How do we prioritize between success indicators and hence between project characteristics? For example, is it more important to deliver on time or that the number of defects is below a certain threshold? In this case, success indicators have to be prioritized to ensure the best combination for a specific project. In this example, we assume that the variables are judged on a scale based on how good they are, i.e. the variables are treated as subjective measures.
- Are there any common patterns in subjective evaluations across projects and organizations?
- Is it possible to aggregate subjective evaluation factors from different organizations?

Before going further, we introduce the notation used and give an example for illustration purposes. A *factor* is a general term for an aspect we would like to study. The factor is either a *project factor* or *success factor*. A project factor may be divided into a number of *project characteristics*, and a success factor into a number of *success indicators*. Project characteristics provide a view of the status or quality of the project, and they can either be estimated prior to starting the project or during the execution of the project. A success indicator captures the outcome of the project, and it is hence measured after project completion. The project characteristics and success indicators are measured through *variables*. A factor is subjective, if a subjective measure is used to measure the corresponding variables.

Example: An example of a project factor may be project management. This factor may include project characteristics such as the quality of the project plan and experience of the project manager. Examples of success indicators include timeliness of delivery and quality of the delivered software. The experience of the project manager may be studied through different variables, for example, number of times as project leader or through a survey among participants in previous projects. The first variable may be measured through calculating an absolute number, but it is also possible that it is judged that it is better to capture the experience on a five-point scale, since the differences between having been project leader, for example, 10 or 11 times are negligible.

More formally, let $prc_1, prc_2, \ldots, prc_i, \ldots, prc_k$ be the project variables and $s_1, s_2, \ldots, s_j, \ldots s_m$ be the success variables. The objective is to find estimators that are able to predict success from project variables. This is illustrated through the formulas below.

$$f_1 \langle prc_1, \ldots, prc_k \rangle = s_1$$

$$\ldots$$

$$f_m \langle prc_1, \ldots, prc_k \rangle = s_m$$

Both prc_i and s_j are rank order variables, and the objective is to identify which project variables are good estimators for the different success variables. The project and success variables are shown in Figure 1 in relation to a software project. The project variables may be estimated prior to starting a project and then tracked as the project is being run. When the project is completed, the success variables are measured.

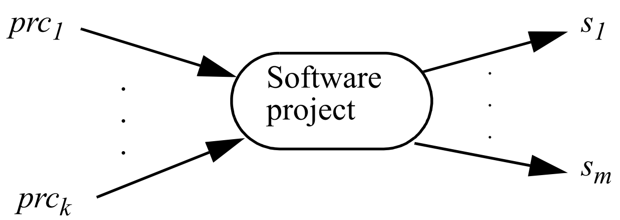

FIGURE 1. Project and success variables.

Analysts are sometimes frustrated when they try to make sense of this data (Valett 1993), and wonder if it is random and useless. Two options exist, either subjective evaluations are discarded or we try to make sense and understand the relationships between subjective project evaluations and project success. The latter seems to be a much more sensible approach although we should be cautious when using subjective evaluations. Some of the subjective measures may well have problems. There are, however, ways to evaluate the ability of using a given set of subjective measures and to determine whether they are useful in indicating success or failure.

First, correlation analysis can be used to show which project variables are positively correlated with project success. In other words, this analysis evaluates to which degree the values for project variables vary together with the success variables. The correlation analysis captures the relationship between one project variable and one success variable.

Second, Principal Component Analysis (PCA) (Kachigan1986) can be used to study which project variables vary together with the success variables. This is done to capture multivariate aspects. PCA is used to analyse which project variables vary together with the success variable(s). We then classify the projects and use an agreement index to determine to which degree project variables "predict" project success as measured by the success variables.

Section 2 describes the analysis method. It consists of five basic steps, and an additional step if several success indicators are used. Section 3 presents the first case study.

It consists of 10 projects from one company. The analysis points out several important relationships between project characteristics and success indicators, particularly with regards to stability and project lead time. Section 4 applies our method to a larger set of data, 46 projects spanning 5 years. This data set includes a different set of project characteristics and success indicators. Several options for applying the method are presented. Some provide better assessment results than others. In either case, the analysis method leads to a better understanding of what influences project success. Finally, in Section 5 some conclusions are presented.

2. Analysis method

Before the analysis, important aspects to study in the analysis should be identified. For our purposes, it must be decided which success and project aspects should be evaluated subjectively. It is usually better that variables are measured quantitatively, and hence it is essential to determine what to measure quantitatively and what to evaluate subjectively. The next step is to determine success indicators and how to define variables. This may include both internal success indicators (for example efficiency and maintainability) and external indicators (for example timeliness and reliability). Thus, it is important to decide what to evaluate when judging the success of a project. Then, the project variables (project characterization) believed to influence the success variables are identified and suitable metrics are determined.

For measuring subjective variables various rating schemes exist, for example, Likert and ordinal scales (Fenton and Pfleeger 1996). The meaning of the different values on the scale should be determined and these should provide a good differentiation between projects. Methods to define scales reliably are described in (von Mayrhauser 1990). Ideally, subjective metrics for such variables should be evaluated for inter-rater and retest reliability (Allen and Yeh 1979).

As with the definition of subjective evaluation factors, analysis should be conducted with care. Thus, it is important to have a systematic analysis approach. We propose the following five steps for each success indicator, and an additional step when having multiple success indicators. The six steps are as follows and they are described in more detail below.

1. The project variables are screened based on each success variable.
2. A principal component analysis (PCA) is conducted to determine which project variables that vary together with each success variable.
3. The projects are ranked based on the outcome of the PCA. Projects are ranked both based on each success variable and the project variables that vary together with each success variable. This results in two ranks for each success variable.
4. Based on the ranking, projects are divided into different classes. Here we have chosen to divide them into two halves, denoted successful respectively unsuccessful projects.
5. The classification forms the basis to evaluate the agreement between classifying the projects based on project variables and each success variable respectively. An agreement index is computed.

6. The analysis in step 1-5 is repeated doing the screening of project variables based on all success variables. The steps in the joint analysis are denoted 6.1-6.5. It should be noted that two different screening criteria are used in the joint analysis (method A and method B).

The objective of the above outline of the method is to provide a context when each step is described in more detail. The steps are subsequently described in more detail and the steps are also illustrated in two case studies presented in Sections 3 and 4.

1. Project variable screening

   We assume that all project and success measures have been formulated so that a higher value on a subjective variable is better. Hence, it is expected that all project variables have a positive correlation with the success variables. If that is not the case, there are either some underlying factors that were not captured, or the scale does not represent what we think. For example, one may expect that higher competence in a project will make it more likely to be successful, but on the other hand the most competent personnel may be assigned to the most difficult and demanding projects. To address issues like this, it is useful to screen the data, i.e. project variables that do not have a positive correlation with the success variable are removed from the analysis.

2. Principal component analysis (PCA)

   Data in too many databases are collected without a purpose (except to collect data). This often leads to measures that are highly correlated because they measure almost the same thing. To be able to extract a useful subset different techniques exist. One such technique is Principal Component Analysis (PCA) (Kachigan1986). PCA groups correlated variables or variables with the same behavior into a number of principal components where each component accounts for the maximum possible amount of the variance for the variables being analysed. The number of principal components extracted may vary depending on the data set and the method chosen for extraction. PCA can help in reducing the number of variables, and it can also provide support in identifying variables that vary together. It can, for example, be applied to analyse similarities between software development projects (Khoshgoftaar and Lanning 1994). It has also been used when building models for prediction of fault-prone software components, see for example (Khoshgoftaar et.al. 1995), and for code decay (Ohlsson et. al. 1999).

   There are some limitations to using PCA. The analysis requires three or more interval variables because it is defined as a linear model. While non-parametric PCA methods have been developed, they are not as commonly used. One reason for this is that the results often turn out to be the same as the standard PCA. In addition, when using parametric PCA with rank order variables we can measure whether the PCA is good by the significance level of the results (Briand et. al. 1996). When applying PCA to non-interval variables, it is necessary to be a little cautious. This may, for example, include not accepting the results from the analysis without making sure that the results correspond to intuitive expectations.

   In this step, the project variables and a single success variable are analysed. This allows us to both determine which project variables that vary together and in particular to identify which project variables that vary together with the success variable. Commonly, variables with a loading of 0.7 or higher are of particular interest since they explain most of the variation of the principal component. Thus, the components are primarily interpreted from the variables with a loading of 0.7 or higher, but other variables in the principal component are also considered. The latter primarily refers to variables having their highest loading in the component.

The following may happen when performing the analysis of the project variables and a success variable:

a) The success variable and one or several project variables have a loading above 0.7. This is the ideal case.

b) The success variable has a loading below 0.7 and one or several project variables have higher loadings. In this case, the project variables with higher loadings than the success variable are considered.

c) The success variable has the highest loading, but one or several project variables have their highest loading in the principal component although their loading are below the threshold of 0.7.

d) The success variable is in a separate principal component from all project variables, i.e. all project variables have higher loadings in other principal components. This means that none of the project variables is behaving similarly to the success variable. Thus, the project variables are not good in predicting success. In other words, the wrong variables were chosen, or the scale for the success variable is not well-defined to obtain a differentiation between projects.

3. Ranking and correlation

   The projects are ranked twice. First, the projects are ranked based on the success variable. Second, they are ranked using the project variables that are grouped in the same principal component as the success variable. The latter ranking is ideally based on the sum of the project variables with a loading higher than 0.7, see a) above. If no project variable has a loading higher than 0.7, then the ranking is based on the sum of the project variables with a loading higher than the success variable, see b) above. If the success variable has the highest loading and no project variable has a loading higher than 0.7, then we have failed to capture the success variable with the project variables, see c) and d) above. Finally, the Spearman correlation is determined between the two rankings.

4. Classification

   Each of the two rankings represents a different model for evaluating a project. The first model ranks the project success based on one success variable. The second model also ranks the projects based on project success, but this time based on the values of key project variables. If we want to predict project success based on project variables, these two rankings (or models) must show some degree of agreement, for example, in terms of classification of projects. Thus, the projects should be classified using the two rankings.

   One opportunity is to classify projects into two classes, which for simplicity are denoted successful and unsuccessful respectively. This classification may be represented in the form of a diffusion matrix or contingency table, see Table 1. Other classification schemes may be used as, for example, in (Ohlsson et. al. 1999 and Wohlin et. al. 2000).

5. Agreement index

   The agreement in terms of classification can be measured by an agreement index, often referred to as kappa statistic (Altman 1991). In software engineering, the kappa statistic has been applied to inter-rater agreement of process assessments (El Emam 1999).

Briefly, the kappa statistic can be explained as follows for the simple case with two raters (or classifications) and two levels (successful or unsuccessful project). Table 1 illustrates this. The cells state the proportions of the projects with a given rating according to model 1 and model 2. For example, p_{11} = 0.20 means that 20% of the projects are considered successful according to the subjective project variables and successful according to the success variable. The columns and rows are summarized (last column and last row respectively in Table 1), which is indicated with p_{01}, p_{02} p_{10} and p_{20}.

TABLE 1. A diffusion matrix for successful projects.

		Success variable		
		Successful	Unsuccess-ful	Sum
Project variables	Successful	p_{11}	p_{12}	p_{10}
	Unsuccess-ful	p_{21}	p_{22}	p_{20}
	Sum	p_{01}	p_{02}	

The entries in Table 1 are used to derive an agreement index. Let P_A be the proportion in which there is agreement. Then, P_A becomes

$$P_A = \sum_{i=1}^{2} p_{ii}$$

This agreement includes cases in which the agreement is obtained by chance. To remove the effect of chance behaviour, the extent of agreement that is expected by chance is defined as

$$P_E = \sum_{i=1}^{2} p_{i0} \times p_{0i}$$

The agreement index is then defined as

$$\kappa = \frac{P_A - P_E}{1 - P_E}$$

To be able to understand the degree of agreement, the kappa statistic is usually mapped into a rank order scale describing the strength of agreement. Several such scales exist, although they are by and large minor variations of each other. Three scales are presented in (El Emam 1999). Here the scale suggested by Altman (Altman 1991) is used. It is shown in Table 2.

TABLE 2. The Altman kappa scale.

Kappa statistics	**Strength of agreement**
< 0.20	Poor
0.21-0.40	Fair

TABLE 2. The Altman kappa scale.

Kappa statistics	Strength of agreement
0.41-0.60	Moderate
0.61-0.80	Good
0.81-1.00	Very good

The agreement index requires comparable scales for the two classifications, i.e. based on project variables or the success variable, to determine which projects are successful in both classifications. This prevents setting thresholds for each classification, since it is not possible (or at least not easy) to map one scale into another directly. To circumvent this problem, we use each classification to rank the projects by success and then consider the upper half of projects successful and the lower half unsuccessful projects. In case of ties, the number of projects considered as a success by the success variable and the project variables may be different.

In the case of several success indicators, we need ways to consider multiple success indicators.

6. Joint analysis for all success variables

   The joint analysis means doing steps 1-5 again, but this time the analysis is done using all success variables as input. The steps in the joint analysis are referred to as 6.1-6.5. A joint analysis of all success variables influences how the data screening in step 1 should be done. There are two options: Method A) to include only variables that have a positive correlation with all success variables or Method B) variables that have a positive correlation with at least one of the success variables. To increase understanding of the relationships between the variables, we recommend performing both types of analysis. This is done for the second case study presented in Section 4.

3. Case study I

3.1 Data

The data in the first case study is from 12 software projects from one company with many divisions. These divisions work in the telecommunication domain. Data was collected for 10 project variables judged critical in improving the predictability of time to market. Thus, there is only one success variable, i.e. the lead time. However, to make lead times comparable, they are normalized with the project effort before being ranked on a five-point scale. The 10 project variables and the success variable are listed in Table 3. The ordinal scales of Table 3 represent subjective judgement, where the objective was to let a higher grade represent something which is believed to be better. The scales were formulated based on expert judgement. This is shown in Table 3 by illustrating the meaning of the lowest grade (1) and the highest grade (5). The complete definition of the scales can be found in (Wohlin and Ahlgren 1995).

Project staffing and quality requirements for the software product are not taken into account as the data is collected from one company, and the applications being developed within the projects have similar quality requirements. The software process is stable. This implies that a number of important factors are stable for these projects and therefore need not be included in the analysis. This assumption is the same as that made in COCOMO, (Boehm 1981).

TABLE 3. Description of data.

	Variables	Brief description	Ordinal scale
Project variables	Complexity	Problem complexity	1: many very difficult subsystems 5: no difficult subsystems
	Competence	Competence of the project personnel	1: all newly employed 5: all experienced
	Req. stability	Stability of requirements	1: major and many changes 5: no changes
	Turnover	Staff turnover	1: Š 10% 5: ð 1%
	Geo. distribution	Geographical distribution of the project	1: > three organizations 5: all work in one organization
	Methods and tools	Quality of methods and tools used	1: disaster 5: advanced, no problems
	Time pressure	Schedule constraints	1: very low 5: very high
	Information flow	Project communication	1: poor 5: very good
	Priority	Top management priority	1: very low 5: very good
	Project management	Performance of management	1: bad, no control or motivation 5: very good, full control and highly motivated
Success variable	Success	Lead time normalized with project effort	

3.2 Analysis

1. Project variable screening

   The scales have been formulated so that positive correlations are expected between the success variable and single project variables. Four variables have negative correlation with success, and are removed from the analysis: competence, complexity, geographical distribution, and methods and tools. The highest correlation with the success variable is obtained for requirements stability which becomes 0.678. The second highest correlation is 0.606 which is the variable denoted priority. It seems that stable requirements are important and that the priority perceived from top level management are important for project success (i.e. short lead times).

   The screening leaves six variables for further analysis together with the success variable.

2. Principal component analysis

   The principal components identified are shown in Table 4. There are three principal components. Loadings above the threshold of 0.7 are shaded. Both the first and third principal components are related to management issues. The second principal component contains the success variable together with requirements stability and staff turnover.

   A separate analysis without the success variable was also conducted, but it basically showed the same results, although it was worth noting that requirements stability had two loadings that were very close to each other (0.626 and 0.608). This is not uncommon, hence underlining the choice of having a loading threshold of 0.7 to avoid having to place one variable in two principal components.

   TABLE 4. PCA including the success variable.

Orthogonal solution	Component 1	Component 2	Component 3
Requirement stability	0.489	0.750	-0.028
Staff Turnover	-0.172	0.836	-0.065
Time pressure	0.955	0.001	0.109
Information flow	0.805	0.275	0.350
Priority	0.075	0.134	0.968
Project management	0.515	-0.258	0.730
Success variable	0.309	0.706	0.495

   Success for this particular data set seems to be highly dependent on project stability, including both stable requirements and low staff turnover.

3. Ranking and correlation

   The projects are ranked twice: (1) based on the success variable, and (2) on the sum of the two project variables having a loading above 0.7 for the same principal component as the success variables, i.e. requirements stability and staff turnover. The correlation between the rankings is 0.544.

4. Classification

   Projects ranked in the upper half represents successful projects, while the lower half represents unsuccessful projects (at least we have chosen to define them this way). In this particular case, a successful project is a project with a grade of 3 or higher. This results in seven successful projects (due to ties between projects). These are compared with the seven projects with the highest rankings based on the project factors. This results in the classification shown in Table 5.

   TABLE 5. A diffusion matrix for successful projects in case study I.

	Agreement index	Success variable	
	0.31 (Fair)	Successful	Unsuccessful
Project variables	Successful	5	2
	Unsuccessful	2	3

5. Agreement index

   The agreement index is calculated based on the outcome of the classification. The agreement index and its interpretation, using the Altman scale, is shown in Table 5.

3.3 Interpretation summary

In summary, it can be noted that the procedure identifies project stability (requirements and personnel) as the key drivers of the chosen success variable. The agreement index turns out to be fair. This is not very high, but it would seem that when important projects are undertaken it is crucial to try to keep them stable. Thus, the assessment of the first company's subjective variables provides an understanding of which project characteristics need special attention when lead time is the key success variable.

4. Case study II

4.1 Data

The case study is based on data from the NASA-SEL database (NASA-SEL 1992). Further information about NASA-SEL can be found in (Basili et. al. 1995). Projects in the second database consist of a rich set of project descriptors. They include parameters related to schedules and estimates, resource use (both manpower and computer use), a variety of product characteristics related to structure, size, growth, and change data. The focus here is however on using the subjective data to understand what drives project success.

Subjective evaluations rank the projects in terms of problem complexity, schedule constraints, nature of requirements, team ability, management performance, discipline, software quality, etc.

In total, the database contains data from more than 150 projects that span five years. Of these, we selected 46 for analysis, based on completeness of project data recorded. These 46 projects represented a variety of types of systems, languages, and approaches for software development. None of the projects is primarily in the telecommunication domain as opposed to the first case study.

The subjective variables are measured on a five-point scale with the higher value denoting more of the quality ranked. The subjective variables can be grouped as indicated in Table 6. The measurement areas are classified into five areas of project variables: Problem, Team, Management, Execution and Infrastructure, and the success variables (Outcome). In total, 27 project variables and 6 success variables are measured for the 46 projects. The third column in Table 6 is related to the screening of the data, see Section 4.2.

TABLE 6. Description of data.

Area	Variables	Brief description	Eliminated for success variable
Problem	COMP	Problem complexity	QDES, QDOC and ACCTEST
	SCHE	Schedule constraints	QDOC
	RSTAB	Stability of requirements	
	RQUA	Quality of requirements	
	RDOC	Documentation requirements	
	RREW	Rigor of requirements reviews	
Team	TABI	Development team ability	
	TAPP	Development team application experience	AGGRE, QSOFT and TIMELI
	TENV	Development team environment experience	AGGRE and TIMELI
	TSTAB	Stability of development team	QSOFT
Management	MPER	Management performance	
	MAPP	Management application experience	AGGRE and QDOC
	MSTAB	Stability of management team	
	PROPL	Project planning discipline	
	PPCOMPL	Project plan compliance	
Execution	PROGP	Programming practices	
	REMET	Requirements methodology	QDOC and ACCTEST
	DEMET	Design methodology	
	TEMET	Test methodology	
	TEPLAN	Use of test plans	
	QA	Quality assurance	

TABLE 6. Description of data.

Area	Variables	Brief description	Eliminated for success variable
Infrastructure	CM	Configuration management	ACCTEST
	DEVSYS	Access to development system	QDES, QDOC and ACCTEST
	DEVTERM	Ratio of developers to terminals	QDOC
	MEM	Memory constraints	AGGRE, QDOC and ACCTEST
	RESTIM	System response time	AGGRE, QDES, QDOC and ACCTEST
	SHWSS	Stability of hardware and support software	AGGRE, QDOC and ACCTEST
Outcome (Success)	AGGRE	Agreement of software with requirements	
	QSOFT	Quality of software	
	QDES	Quality of design	
	QDOC	Quality of documentation	
	TIMELI	Timeliness of delivery	
	ACCTEST	Smoothness of acceptance testing	

4.2 Analysis

1. Project variable screening

   This case study involves six success variables. The data is screened for all six variables separately. Table 7 shows how many project variables are positively and negatively correlated with each success variable (and thus included in the analysis).

TABLE 7. The number of project variables positively and negatively correlated with each success variable.

Success variable	Number of positively correlated project variables	Number of negatively correlated project variables
AGGRE	21	6
QSOFT	25	2
QDES	24	3
QDOC	18	9
TIMELI	25	2
ACCTEST	20	7

The negatively correlated variables are eliminated from further analysis. The eliminated variables are shown in the third column of Table 6 by listing the success variable for which the project variable is eliminated. It is interesting to note that most of the negative correlations relate to the project variables in the Infrastructure area, see Table 6. This could be an indicator that infrastructure was basically sufficient and thus did not have a detectable influence on the success variables.

2. Principal component analysis

   As indicated by Table 7, most variables have a positive correlation with the success variables. In the interest of space, we focus on interpreting the principal components briefly and on the component containing the success variable in particular. The results from the analyses are summarized in Table 8, and analysed below for each success variable. Variables with a loading higher than 0.7 are shown in boldface and other variables in the principal component with plain text. The principal components containing the success variables are shaded.

TABLE 8. Results of the principal component analysis in the second case study.

	Analysis for AGGRE	**Analysis for QSOFT**	**Analysis for QDES**	**Analysis for QDOC**	**Analysis for TIMELI**	**Analysis for ACCTEST**
Comp. 1	**MPER** **PROPL** **PPCOMPL**	**QSOFT** MPER **PROPL** **PPCOMPL** PROGP	**QDES** PROPL PPCOMPL **TEMET** **TEPLAN**	MSTAB TEMET	**TIMELI** MPER **PROPL** **PPCOMPL**	RDOC **RREW** QA
Comp. 2	COMP RSTAB TABI MSTAB CM **REMET** **DEVSYS** **DEV-TERM**	COMP SCHE RSTAB TABI MSTAB **REMET** CM **DEVSYS** **DEV-TERM** RESTIM	**SCHE** **RQUA** TABI PROGP REMET CM DEVTERM	**QDOC** RSTAB PPCOMPL	COMP SCHE RSTAB TABI **REMET** CM **DEVSYS** DEV-TERM **RESTIM**	**ACCTEST** **RSTAB** TSTAB PROPL PPCOMPL
Comp. 3	**RDOC** **RREW** QA	RDOC **RREW** QA MEM	RDOC **RREW** DEMET QA MEM	**TAPP** **TENV**	RDOC **RREW** QA MEM	**SCHE** **RQUA** **PROGP** DEVTERM
Comp. 4	AGGRE **TSTAB**	**SHWSS**	**TAPP** **TENV**	RQUA RDOC **RREW**, DEMET **QA**	**SHWSS**	**TAPP** **TENV**
Comp. 5	**RQUA** SCHE	RQUA **TENV**	**SHWSS**	**TABI** **MPER** PPLAN TEPLAN CM	**RQUA** PROGP	TABI **MPER** **MAPP**

TABLE 8. Results of the principal component analysis in the second case study.

	Analysis for AGGRE	Analysis for QSOFT	Analysis for QDES	Analysis for QDOC	Analysis for TIMELI	Analysis for ACCTEST
Comp. 6	PROGP **DEMET** **TEMET** TEPLAN	**MAPP**	MPER **MAPP**	**TSTAB** PROGP	**DEMET** **TEMET** **TEPLAN**	**MSTAB**
Comp. 7	-	DEMET **TEMET** **TEPLAN**	TSTAB	-	**TSTAB**	DEMET **TEMET** **TEPLAN**
Comp. 8	-	-	RSTAB **MSTAB**	-	**MAPP** MSTAB	-

The principal component analysis for each success variable is shown in Table 8 and may now be interpreted. Two aspects are of particular interest: i) Project variables in the same principal component as the success variable, and ii) The division into principal components as such, i.e. whether it is possible to identify commonalities between the variables in the different components.

Agreement of software with requirements (AGGRE)

i) The most important issue is of course to study how the success variable relates to the project variables. AGGRE is placed together with TSTAB (stability of the development team). These two variables result in one factor. It should, however, be noted that the loading for AGGRE is only 0.586 and the loading for TSTAB is 0.791. A possible explanation of the outcome is that it is fairly important that we have a stable development team to fulfill the requirements. It may be that new team members have a tendency to interpret requirements slightly differently to those who have worked on the project from the beginning. The results indicate that stability of the development team is important to meet requirements.

A potential problem with this success variable is that the differentiation between the scores for the different projects is fairly small.

ii) PCA identifies six principal components. The first component is a project planning factor. The second factor is less obvious. It seems to be a collection of variables which relate to project characteristics in general. The third component is related to requirements. The fourth component is the one containing the success variable. The fifth component is also related to requirements, but primarily the quality of the requirements. The sixth factor is clearly a methodology factor including design, programming and testing. Thus, for five of the six components, we see certain commonalities among the variables in the components.

Quality of software (QSOFT)

i) QSOFT is grouped with variables primarily related to project management (MPER, PROPL and PPCOMPL) and the programming practice variable (PROGP). QSOFT, PROPL and PPCOMPL have loadings above 0.7, so it is quite clear that project planning is highly important to obtain high quality software.

ii) PCA identifies seven principal components. The first component is once again related to project planning. The second component is a diverse collection of variables. The third component is primarily related to requirements. The fourth component is a support component containing a single variable. The fifth component contains two variables, which are not easily interpreted. The sixth component contains a single variable, i.e. the management´s application experience. Finally, the seventh factor is a methodology factor with a particular focus on testing.

Quality of design (QDES)

i) QDES is grouped together with the project planning variables (PROPL and PPCOMPL) and the variables related to test methodology and test planning (TEMET and TEPLAN). The loadings are above 0.7 for the test variables and QDES. In this case, it is hard to determine whether the high quality of the design has driven the test or if good testing routines have put high requirements on the design. Independently, a close relationship is found between testing and design quality.

ii) PCA identifies eight principal components. The first component contains the success variable. The second component is once again a collection of several variables making it hard to interpret the factor. The third factor is also a collection of variables. The fourth component is related to the experience of the team. Components five to eight all only contain one variable each.

Quality of documentation (QDOC)

i) QDOC is grouped together with requirements stability (RSTAB) and the compliance to the project plan (PPCOMPL). It is the only success variable that has a loading above 0.7, which means that we obtain an outcome described by case c) in step 2 of the method, see Section 2. This makes it difficult to interpret the outcome, and it is obvious that the project variables are not fully able to capture the success variable related to the quality of the documentation. This variable also suffers from being fairly similar for the different projects.

ii) When analyzing the project variables having a positive correlation with the quality of the documentation, we obtain six principal components. The interpretation of the components is as follows. The first component contains two variables with no obvious relation. The second component includes the success variable. The third component is solely related to the experience of the team. The fourth component is primarily a requirement component. Both the fifth and sixth components are collections of variables. The interpretation of the components is, for several of the components, not obvious.

Timeliness of delivery (TIMELI)

i) The timeliness of delivery variable (TIMELI) is grouped with the project planning variables. Both PROPL and PPCOMPL are in the same component as TIMELI with loadings above 0.7. This means that timely delivery is highly dependent on the planning of the project and the ability to follow the plan.

ii) PCA identifies eight principal components. This time the components can be interpreted as follows. The first component is primarily a project planning component. The second component is once again a collection of several variables that seem to be unrelated. The third component is primarily a requirements component, although including two other variables. The fourth component describes stability of the software and hardware support. The fifth component contains two variables and it is not obvious how to interpret the component. The sixth component is the methodology component, primarily testing. The seventh component consists of a single variable (TSTAB). The eighth component is clearly related to the experience of management.

Smoothness of acceptance testing (ACCTEST)

i) The smoothness of acceptance testing is grouped with the component related to stability and project planning. In other words, the smoothness of the test depends on two main things: a good project plan and stability in the project, especially stable requirements (RSTAB). The latter and the success variable (ACCTEST) are the only ones having a loading above 0.7. This indicates that the smoothness of the acceptance test is closely related to the stability of the requirements. This is no surprise since it is difficult to develop the software and plan the acceptance testing if the requirements are unstable.

ii) PCA identifies seven components. The first component is a requirements component. The second component contains the success variable. The third component is not easily interpreted. The fourth component is related to the experience of the team. The fifth component is primarily related to the experience of the project manager. The sixth component describes the stability of project management. Finally, the seventh component is a methodology component with a particular focus on testing.

Summary of PCA

i) The six success variables have been analysed together with the project variables using principal component analysis. We were able to identify which project variables are most important for the various success variables. The outcome is:

- AGGRE: Stability of development team
- QSOFT: Project planning
- QDES: Testing
- QDOC: Weak relation to requirements stability and the compliance to the project plan.
- TIMELI: Project planning
- ACCTEST: Stability of requirements

The above results do not really include any big surprises, but on the other hand they were not obvious either. The results show that it is necessary to focus on different project characteristics depending on what are the most important success indicators. The next step is to study the rankings we obtain using the project variables respectively the success variables, see step 3.

ii) We have seen that most of the principal components have an intuitive interpretation. The components show that the subjective variables are indeed able to capture important aspects related to the characterization of software projects. Several of the components are almost identical for a number of the success variables. This indicates that the collection of subjective project variables may indeed lead to an increased insight into the dependence of different project characteristic on project success.

3. Ranking and correlation

All projects are ranked based on two schemes:

1) Each of the six success variables,

2) Each of the six sets of project variables which the PCA grouped with each success variable.

This results in 12 rankings, which then are compared pairwise, i.e. for each success variable. The Spearman rank correlations are computed for comparison. The results are summarized in Table 9.

TABLE 9. A summary of the success variables and the corresponding project variables.

Success variable	Corresponding project variables	Spearman rank correlation
AGGRE	TSTAB	0.163
QSOFT	PROPL and PPCOMPL	0.576
QDES	TEMET and TEPLAN	0.640
QDOC	No corresponding variable	-
TIMELI	PROPL and PPCOMPL	0.602
ACCTEST	RSTAB	0.564

The first correlation is low, but we need to take into account that the loading for the success variable is below 0.7. Four of the correlations are acceptable, although not very good. Unfortunately, these correlation values are fairly common in software engineering (Zhao et. al. 1998). The positive side is that the correlations for the subjective variables and their ranking is comparable to those obtained for objective measures (Zhao et. al. 1998). Thus, to some extent, the subjective variables are doing as well as the objective variables. Hence it seems worthwhile working with subjective as well as objective measures.

4. Classification

   The projects are divided into halves. For this data set, it means that a project having a rank of 23 or lower is considered a success and consequently a rank higher than 23 means that the project is considered a failure. Due to the ties in the ranking, we will not obtain exactly 23 projects in each half. The results of classifying the projects are shown in Tables 10 to 15.

5. Agreement index

   Based on the classifications, it is possible to compute the agreement indices. The indices are fairly low for some of the success variables. The reasons for this are explained further below when treating each success variable separately. In addition to this, there are two other reasons why the index is lower than hoped for. First, the agreement index is used to compare two different classifications using different scales. Normally, the agreement index is used to compare different classifications of people using the same scale. Secondly, some of the misclassifications are due to that some projects are classified close to the boundary between successful and unsuccessful, and they end up on different sides of the boundary for the two classifications. The latter is addressed in a recent study, where a third class was introduced for the classification based on project characteristics. The objective was to capture the uncertainty in the projects being classified close to the boundary in this study. Further information regarding the extended classification model can be found in (Wohlin et. al. 2000). The extended classification model improves the agreement index, which shows that it is important to handle the projects close to the boundary between the two classes here carefully.

TABLE 10. A diffusion matrix for AGGRE as the success variable.

Agreement index		Success variable: AGGRE	
0.14 (Poor)		Successful	Unsuccessful
Project variables	Successful	23	1
	Unsuccessful	18	4

The agreement index for AGGRE is poor. This is what can be expected given that the correlation also was low. Another reason is that the number of ties was very high, which actually means that 41 of the projects are regarded as successes using the ranking based on AGGRE. A positive aspect is that most projects seemed to score well on this variable. The distribution of the scores is skewed, i.e. most projects have either a score of four or five. In other words, it may not be necessary to worry about this variable, since most projects had high scores.

For the next two success variables, see Tables 11 and 12, the agreement index becomes moderate, when estimating project success from related project variables.

TABLE 11. A diffusion matrix for QSOFT as the success variable.

Agreement index		Success variable: QSOFT	
0.47 (Moderate)		Successful	Unsuccessful
Project variables	Successful	23	1
	Unsuccessful	11	11

TABLE 12. A diffusion matrix for QDES as the success variable.

Agreement index		Success variable: QDES	
0.47 (Moderate)		Successful	Unsuccessful
Project variables	Successful	20	6
	Unsuccessful	6	14

Due to the weak relation between QDOC and the project variables, it is infeasible to make a classification and hence compute an agreement index. Once again the distribution of scores is skewed towards high scores; only one project has a score below three. Two options exists, either it is not useful to include this variable in the analysis or the scale has to be reformulated to ensure a better differentiation between projects. The good news is that the projects mostly are successful in terms of this variable.

The agreement index for TIMELI is shown in Table 13. The index is only fair. This may be because the large number of ties in the ranking resulting in that 38 projects are viewed as successes based on the success variable and only 24 projects are estimated as successful using the project variables.

TABLE 13. A diffusion matrix for TIMELI as the success variable.

Agreement index		**Success variable: TIMELI**	
0.28 (Fair)		Successful	Unsuccessful
Project variables	Successful	23	1
	Unsuccessful	15	7

The agreement index for ACCTEST, see Table 14, is moderate when trying to predict the outcome from the project variables.

TABLE 14. A diffusion matrix for ACCTEST as the success variable.

Agreement index		**Success variable: ACCTEST**	
0.49 (Moderate)		Successful	Unsuccessful
Project variables	Successful	27	4
	Unsuccessful	6	9

From the agreement index analysis, it is clear that it is important to either have scales that differentiate between projects or the success variables can be removed from the analysis. The formulation of the scales is crucial when using the method. On the other hand, if all projects succeed based on a specific success variable then we should not formulate the scale to force differentiation. The solution is probably to remove the success variables, with similar scores for most projects, from the analysis, but continue to monitor them so that they do not become a problem.

The next question is whether the analysis could be improved, if all success variables are analysed together.

6. Joint analysis for success variables

This analysis is done with all success variables included in the analysis. The objective is to improve our understanding of how the variables vary together and relate to each other.

6.1. Screening

Two different screenings have to be done because we apply two different methods (analysis A and B respectively). Analysis A only includes project variables with a positive correlation with all six success factors. This includes 14 variables for further analysis, and removes 13 variables. The highest correlation between one of the screened project variables and the success variables is 0.526 (between CM and the quality of the software).

In analysis B, all project variables are included that have a positive correlation with at least one of the success variables. The latter results in retaining all project variables, since no project variable has a negative correlation with all success variables.

6.2. Principal component analysis

Method A

Analysis A results in six principal components as follows shown in Table 15. Components containing success variables are shaded.

TABLE 15. Results of the principal component analysis using method A.

Component 1	Component 2	Component 3	Component 4	Component 5	Component 6
MPER **PROPL** PPCOMPL QSOFT QDES **TIMELI**	**RDOC** **RREW** TABI QA	AGGRE **QDOC**	DEMET **TEMET** **TEPLAN**	**RQUA** PROGP	RSTAB **MSTAB** ACCTEST

Once again, there is two aspects to consider: i) Project variables in the same principal component as the success variable, and ii) The division into principal components as such, i.e. whether it is possible to identify commonalities between the variables in the different components.

i) The success variables are distributed over three components (Component 1, Component 3 and Component 6). Component 3 includes two success variables (AGGRE and QDOC) but no project variable, although the loading of requirements stability is fairly high. This finding is consistent with the fact that the agreement indices for these two success variables were the lowest, and they also had a correlation lower than 0.5 when considering the ranking. This is a result of these two success variables having high scores for most projects as discussed previously. The first component includes three project variables MPER, PROPL and PPCOMPL as well as three success variables QSOFT, QDES and TIMELI. Component 6 contains RSTAB and MSTAB as well as the success variable ACCTEST.

ii) The first component is primarily related to project management and several success variables. The second component is primarily a requirements component. The third component only contains two success variables. The fourth component is clearly a methodology component, especially related to testing. The fifth component contains two not obviously related variables. The sixth component is related to one of the success variables and stability, both in terms of requirements and management.

Method B

In analysis B, all project variables are included. This results in nine principal components, see Table 16.

TABLE 16. Results of the principal component analysis using method B.

Comp. 1	Comp. 2	Comp. 3	Comp. 4	Comp. 5	Comp. 6	Comp. 7	Comp. 8	Comp. 9
MPER **PROPL** **PPCOMPL** AGGRE **QSOFT** **QDES** QDOC **TIMELI** **ACCTEST**	COMP SCHE RSTAB TABI **REMET** CM **DEVSYS** **DEV-TERM** RESTIME	**TAPP** **TENV**	RDOC **RREW** QA MEM	**SHWSS**	MAPP MSTAB	DEMET **TEMET** **TEPLAN**	**TSTAB**	**RQUA** PROGP

i) The analysis places all six success variables in the first component together with MPER, PROPL and PPCOMPL. The two latter have a loading above 0.7 and so do QSOFT, QDES, TIMELI and ACCTEST. Once again this indicates that it is harder to find project variables explaining the agreement to requirements and the quality of the documentation. This may be compared with both Analysis A, and with the results found in step 5 when analysing the success variables one at the time. The results obtained are hence fairly stable for the different types of analyses done.

ii) Most components have an intuitive interpretation. There are components representing primarily: project management, requirements, team experience, support, management experience, testing and staff turnover. In addition, one component seems to be a collection of variables (Factor 2).

6.3. Ranking and correlation

In analysis A, three pairs of rankings are determined based on the three components into which the PCA grouped the success variables. Since some of the success variables have a loading lower than 0.7 in the first component, it was decided to lower the threshold to 0.66 to include all of the success variables in the component. This resulted in a correlation of 0.633 for project variables and success variables in Component 1. Component 3 only includes two success variables, but no project variables. Thus, no correlation analysis can be conducted, unless we choose to correlate the project variable with the highest loading (RSTAB) with the success variables. (This would result in a correlation of 0.354.) In Component 6, we used a threshold of 0.55 to include the success variable. The correlation was 0.518.

Analysis B only has one pair of ranking, since all six success variables ended up in the same principal component. To include all success variables, the loading threshold had to be set to 0.564. The rankings are based on the sum of the grades for the project variables included and the sum of the grades for the six success variables. The collection of all success variables in one component and the fact that all of them have fairly high loadings show that the success variables are closely related. Simplistically, this means that a good project is a good project independent of which success variable is measured. Only two project variables have loadings above the threshold, namely project planning (PROPL) and compliance to the plan

(PPCOMPL). The correlation between the rankings is 0.665. In summary, it must be concluded that project planning and compliance to the plan are clearly two very important project variables for a successful project.

6.4. Classification

As before, the projects are divided into halves. The results of classifying the projects are shown in Tables 17 to 19.

6.5. Agreement index

For analysis A, the agreement index is determined for three rankings. In other words, we have decided to include the third factor in the analysis too using RSTAB as the project variable. The results are shown in Tables 17-19. It can be seen from Table 17 that the agreement index is rather good for the three success variables most closely related to the project planning variables.

A problem with the joint analysis is that it is hard to determine rules of how to perform the computation of the agreement index. For example, it is difficult to determine how the threshold of 0.7 should be used when it is in most cases probably is necessary to lower the threshold to include all success variables. The lowering of the threshold is most likely the reason why the agreement indices become fairly low, at least in two of the three cases.

TABLE 17. A diffusion matrix for success variables QSOFT, QDES and TIMELI.

	Agreement index	**Success variables: QSOFT, QDES and TIMELI**	
	0.47 (Moderate)	Successful	Unsuccessful
Project variables	Successful	20	5
	Unsuccessful	7	14

TABLE 18. A diffusion matrix for success variables AGGRE and QDOC.

	Agreement index	**Success variables: AGGRE and QDOC**	
	0.34 (Fair)	Successful	Unsuccessful
Project variables	Successful	23	4
	Unsuccessful	10	9

TABLE 19. A diffusion matrix for the success variable ACCTEST.

	Agreement index	**Success variable ACCTEST**	
	0.19 (Poor)	Successful	Unsuccessful
Project variables	Successful	19	12
	Unsuccessful	6	9

Next, for analysis B, the agreement index is moderate 0.43, see Table 20. The study in (Wohlin et. al. 2000) shows that it is possible to get substantially better agreement indices by introducing a third class when classifying projects based on project characteristics. This means that we neither classify the project as successful nor as unsuccessful. Two classes are however used with regard to the outcome of the project.

From Table 20, it can be observed that project variables are able to identify 18 of 24 successful projects and 15 unsuccessful projects out of 22. This is fairly good. The incorrectly classified projects are mostly projects that are close to the borderline, i.e. projects which are ranked on at least one of the scales in the range 20-27 (remembering that the border is between ranks 23 and 24). Only four of the misclassified projects have ranks further away from the border. The largest misclassification is one project where the project variables rank the project as 8, and the success variables give the project a rank of 34. If only these four misclassifications are regarded as misclassifications, since the other “misclassifications” are close to the border between the classes, the agreement index becomes as high as 0.82, which is very good.

TABLE 20. A diffusion matrix for all six success variables.

Agreement index		**All six success variables**	
0.43 (Moderate)		Successful	Unsuccessful
Project variables	Successful	18	6
	Unsuccessful	7	15

To study this further, the objective was to study the ten highest and lowest ranked projects from the project variables, and see if these have a tendency to end up being successful and unsuccessful respectively. Given that we have 46 projects, the following procedure was used:

- Top = projects with a rank above or equal to 10 (because of ties the top is equal to 17 projects)
- Bottom = projects with a rank below or equal to 37 (because of ties the bottom is equal to 7 projects)

Seventeen projects have rankings below ten based on the project variables and only seven projects (out of the 46) have a ranking above 37; 13 of the 17 projects actually became successful and 6 of the 7 projects did end up being unsuccessful. This results in an agreement index of 0.55. Thus, the highest and lowest ranked projects from the project variables are fairly likely to become a success or a not so successful project as predicted from the project variables. This underlines that we actually have a fairly good opportunity to judge the forthcoming success of a software project based on the planning of the project and by the compliance to the plan.

4.3 Interpretation summary

This case study shows that it is feasible to identify a few project characteristics with a major influence on a given success indicator. Four of the success variables investigated can be mapped successfully to one or two key project characteristics. This is very encouraging given that the case study includes 27 different project variables. In other words, the method is able to extract a few vital characteristics from a large data set. The other two success variables are harder to capture, which is a result of that they both mostly have high scores and hence it is difficult to differentiate between projects. High scores may either mean that the variables do not have to be as closely monitored, although some tracking is still essential, or that the scales for these two variables have to be redefined.

The actual outcome of the case study is interesting in the sense that different project characteristics relate to different success indicators. Thus, it shows that by determining which success variables are most important should help focusing on the key project characteristics to have a successful project.

5. Conclusions

Subjective measures for project characteristics and project success are often collected, but are step-children in subsequent analysis. We developed a method to estimate project success based on project characteristics when both are based on subjective measures. It was applied to two case studies. The method is based on the collection of subjective measures of project characteristics and success indicators. While it by no means is a silver bullet for software project success, it provides important support for decision makers in planning and controlling software projects successfully.

The method may be used in many different ways: 1) to assess the relationship between characteristics and success, as presented in this paper, and 2) to use historical data from prior projects to determine the classification model and then apply it to new projects by estimating project characteristics early and predicting certain success indicators. The latter may include aiming at a certain score for some key project characteristics to reduce the project risk.

The method is clearly capable of identifying key project characteristics, which influence a certain success variable. The steps in the method provide an opportunity to define your own success variables and identifying whatever project characteristics believed to be important. The method is then applicable to identify the most important project characteristics in a specific organisation to increase the likelihood of success. The method has been illustrated in two case studies illustrating its use and also showing that the method is indeed able to pinpoint key project characteristics in different organisations. It has been shown that the method is independent of organisation and specific measures, and it should be useful for any software organisation that wants to understand the underlying reasons for why some projects turn out better than others.

The results have been presented separately for two organisations. A major challenge would be to perform a meta-analysis, but this is obviously very hard given that we have:

- different measures

  Most organisations collect different project measures and also measure success differently.

- different scales

  Even if two companies measured the same variables, it is not likely that they would use the same scale.

- inter-rater reliability

  Interater reliability is a challenge within an organisation and it becomes an even greater challenge between organisations.

Given these issues, it is not likely that a meta-analysis is possible or not even desirable, since different companies and application domains most certainly have different ways of measuring success and also characterizing software projects. Thus, the method presented should primarily be seen as useful within a company to reduce project risk.

In summary, we have shown that it is feasible to use subjective factors for understanding and assessing software project success. The case studies have highlighted that the method is useful to understand and assess the relationship between project characteristics and success. The method was able to identify a few key project characteristics with a major impact on a given success indicator.

Future work includes weighting of different success indicators. This requires that the method is complemented with a way of prioritizing the success indicators in relation to each other to try to maximize success.

Acknowledgment

We are grateful the anonymous reviewers who helped us improve the paper through their comments. We also thank Frank McGarry for giving us access to the NASA-SEL data which we used as one of the case studies to illustrate our approach.

References

(Allen and Yeh 1979) Allen M., and Yeh W. 1979, Introduction to Measurement Theory, Brooks/Cole Publishing.

(Altman 1991) Altman D. 1991, Practical Statistics for Medical Research, Chapman-Hall.

(Basili et. al. 1995) Basili V., Zelkovitz M., McGarry F., Page J., Waligora S., and Pajerski R. 1995. SEL's Software Process-Improvement Program, IEEE Software, November 1995: 83-87.

(Boehm 1981) Boehm B. 1981, Software Engineering Economics, Englewood Cliffs, N.J., USA, Prentice-Hall.

(Briand et. al. 1996) Briand L., El Emam K., and Morasca S. 1996. On the Application of Measurement Theory in Software Engineering, Journal of Empirical Software Engineering, Vol. 1, No.1: 61-88.

(El Emam 1999) El Emam K. 1999. Benchmarking Kappa for Software Process Assessment Reliability Studies, Empirical Software Engineering: An International Journal, Vol. 4, No. 4: 113-133.

(Fenton and Pfleeger 1996) Fenton N., and Pfleeger S. L. 1996, Software Metrics – A Rigorous & Practical Approach, International Thompson Computer Press.

(Gray et. al. 1999) Gray A. R., MacDonell S. G., and Shepperd M. J. 1999. Factors Systematically Associated with Errors in Subjective Estimates of Software Development Effort: The Stability of Expert Judgement, Proc. of the Sixth Int. Software Metrics Symposium, Boca Raton, Florida, USA, pp. 216-227.

(Hughes 1996) Hughes R. 1996. Expert Judgement as an Estimation Method, Information and Software Technology, Vol. 38: 67-75.

(Höst and Wohlin 1998) Höst M., and Wohlin C. 1998. An Experimental Study of Individual Subjective Effort Estimations and Combinations of the Estimates, Proc. IEEE Int. Conf. on Software Engineering, Kyoto, Japan, pp. 332-339.

(Kachigan 1986) Kachigan S. K. 1986, Statistical Analysis – An Interdisciplinary Introduction to Univariate & Multivariate Methods, Radius Press.

(Khoshgoftaar and Lanning 1994) Khoshgoftaar T. M., and Lanning D. L. 1994. Are the Principal Components of Software Complexity Stable Across Software Products?, Proc. of the Int. Symposium on Software Metrics, London, United Kingdom, pp. 61-72.

(Khoshgoftaar et. al. 1995) Khoshgoftaar T. M., Allen E. B., Kalaichelvan K. S., Goel N., Hudepohl J. P., and Mayrand J. 1995. Detection of Fault-Prone Program Modules in a Very Large Telecommunications System, Proc. of the Int. Symposium on Software Reliability Engineering, Toulouse, France, pp. 24-33.

(von Mayrhauser 1990) von Mayrhauser A. 1990, Software Engineering: Methods and Management, Academic Press.

(NASA-SEL 1992) NASA-SEL 1992. Software Engineering Laboratory Database Organization and Users Guide, Revision 2, Goddard Space Flight Center, Greenbelt, MD, USA, NASA-SEL Series, SEL-89-201.

(Ohlsson et. al. 1999) Ohlsson M. C., von Mayrhauser A., McGuire B., and Wohlin C. 1999. Code Decay Analysis of Legacy Software through Successive Releases, Proc. IEEE Aerospace Conf., Snowmass, Colorado, USA.

(Ropponen and Lyytinen 2000) Ropponen J., and Lyytinen K. 2000. Components of Software Development Risk: How to Address Them? A Project Manager Survey, IEEE Trans. in Software Engineering, Vol. 26, No. 2: 98-112.

(Valett 1993) Valett J. D. 1993. The (Mis)use of Subjective Process Measures in Software Engineering, Proc. Software Engineering Workshop, NASA/Goddard Space Flight Center, Greenbelt, Maryland, USA, pp. 161-165.

(Wohlin and Ahlgren 1995) Wohlin C., and Ahlgren M. 1995. Soft Factors and Their Impact on Time to Market, Software Quality Journal, No. 4: 189-205.

(Wohlin et. al. 2000) Wohlin C., von Mayrhauser A., Höst M., and Regnell B. 2000. Subjective Evaluation as a Tool for Learning from Software Project Success, Information and Software Technology, Vol. 42, No. 14: 983-992.

(Zhao et. al. 1998) Zhao M., Wohlin C., Ohlsson N., and Xie M. 1998. A Comparison between Software Design and Code Metrics for the Prediction of Software Fault Content, Information and Software Technology, Vol. 40, No. 14: 801-809.

Dr. Claes Wohlin

Dr. Wohlin is a professor of software engineering at the Department of Software Engineering and Computer Science, Blekinge Institute of Technology. He has a Ph.D. in Communication Systems from Lund University. His research interests include empirical methods in software engineering, software metrics, software quality and systematic improvement in software engineering. Claes Wohlin is the principal author of the book "Experimentation in Software Engineering – An Introduction" published by Kluwer Academic Publishers in 1999. He is on the editorial board of the journal of Information and Software Technology. Dr. Wohlin is the program chair for the IEEE International Symposium on Software Metrics in London, UK in 2001.

Dr. von Mayrhauser

Anneliese von Mayrhauser received a Dipl.-Inf. degree in Informatik (1976) from the Technical University in Karlsruhe and the A. M. (1978) and Ph.D. (1979) in Computer Science from Duke University. She is currently a Professor of Computer Science at Colorado State University and Director of the Colorado Advanced Software Institute. Dr. von Mayrhauser has published on software testing, software metrics, software maintenance, reliability and performance modeling.