C. Wohlin and A. Andrews, "Analysing Primary and Lower Order Project Success Drivers", Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering, pp. 393-400, Ischia, Italy, July 2002.

Analysing Primary and Lower Order Project Success Drivers

Claes Wohlin
Dept. of Software Eng. and Computer Science, Blekinge Institute of Technology, Box 520
SE-372 25 Ronneby, Sweden
+46 457 38 58 20
claes.wohlin@bth.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 success is influenced by many factors. Some are primary drivers of project success, others secondary. However, these secondary factors may be no less important to understand and learn from than the primary factors. In addition, project success drivers are often qualitative and subjective, eluding analysis through traditional statistical methods. This paper presents a method that analyses the influence and nature of primary and lower order project success drivers. A case study illustrates the usefulness of this analysis and the additional understanding gained from including lower order success drivers in the analysis. The method extends existing work that has been restricted to primary drivers.

Categories and Subject Descriptors

D2.9 [**Management**] Software Process Models, D2.8 [**Metrics**] Process metrics.

General Terms

Management, Measurement, Experimentation

Keywords

Project success, subjective measures, project assessment.

1. INTRODUCTION

Some of the project characteristics that influence project success include stability of requirements, knowledge of developers and management, inherent difficulty of the project, techniques and tools used (and how appropriate they are), tightness of schedule and type of application etc. These project characteristics influence success indicators and through them project outcome. Success indicators include timeliness of delivery, quality of software, as well as more long term properties such as maintainability and evolvability.

Some of the project characteristics and success indicators are difficult to measure objectively and quantitatively. Further, project characteristics and success indicators may reflect what could be measured rather than actual importance of the characteristic or indicator. Measurement may be incomplete as well. Sometimes it is easier and leads to more complete measurement data to collect project characteristics and success factors in the form of a questionnaire at selected times during and after development. This is why software project databases contain sizeable amounts of subjective measures, both for project characteristics and success indicators. We consider a measure subjective when it assigns a value non-objectively (usually on a rank scale, e.g. the five point Likert scale).

While subjective measurement can (and should) be systematic it lacks the rigor of objectively measurable and quantitative scales. To account for this, one normally develops reliability indicators for such scales (e.g. inter-rater reliability) [1]. High reliability attests to quality of the scale. Unfortunately, few scales used for measuring subjective variables for software projects have been evaluated like this.

Analysts are sometimes frustrated when they try to make sense of subjective data [15], 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.

As a result, subjective measures have only recently been analysed more extensively in empirical software engineering, specifically in software project assessment and evaluation. Part of the problem stems from issues related to collecting trustworthy data [15]. On the other hand, subjective measures have been used successfully for effort estimation [7] and [9], and risk management [14]. Expert judgement in estimation tasks has been discussed in [8].

More recently, subjective variables have been used to map project characteristics to project success. First [16], subjective factors were evaluated alongside objective quantitative ones to evaluate both efficiency and success of software production for projects from the NASA-SEL database [3]. The paper identifies which successful projects were also efficient and determines primary (subjective) drivers of project success. In [18], the method for analysing subjective project characteristics and success indicators is refined and discussed in-depth, and two case studies are presented. The primary success drivers amongst the project characteristics are identified and an agreement index is established that quantifies to which degree the project characteristics that were identified as primarily connected to project success are able to predict project success. The results identified that about one third of the successful projects could be predicted accurately. We believe that this is due to the limitations of the approach, specifically that projects are classified into two categories: upper half (based on project characteristics) and lower half (based on project characteristics). The halves are also denoted "good" and "bad" respectively although it is really up to each individual organization to judge where the limit between "good" projects and "bad" projects is. It is reasonable to assume that projects around the border between "good" and "bad" exhibit more uncertainty with respect to project outcome (success or failure). This could account for the misclassification. To circumvent this problem, an extension was proposed in [19], where a third class was introduced to try to avoid classifying projects close to the border between "good" and "bad". This solution circumvents rather than addresses the problem, and hence we would here like to address the problem.

Thus, the main research question is: Are there secondary or tertiary project characteristics that might reduce uncertainty? Positive answers to these issues would make it possible to use the classifications of primary and lower order project success drivers to identify projects based on their project characteristics as likely successes, failures or "at risk" (uncertain) projects while trying to keep the last category as small as possible.

Thus, the method proposed in this paper should not only be useful to evaluate projects after the fact, but to assess the likely success of ongoing projects based on their project characteristics. Beyond, the analysis prioritizes which are primary drivers of project success, which are secondary, and which do not drive success at all.

Before going further, we introduce the notation used and give an example for illustration purposes. The notation is the same as used in [18]. A *factor* is a general term for an aspect we would like to study. The factor is either a *project factor* or a *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 is measured after project completion. A project characteristic identified as driving success is denoted *success driver*. Project characteristics and success indicators are measured through *variables*. A factor is subjective, if subjective measures are 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 measured by the 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 or through a a five-point Likert scale, since the differences between having been project leader, for example, 10 or 11 times are negligible.

Here, we extend the method in [18] to allow for analysis of secondary and lower order project factors that influence project success. We also consider success not as a binary variable, but one that has three categories: Green, Yellow and Red. "Green" means almost certain project success, "Red" means likely that the project is not very successful (failure), and "Yellow" means that we are uncertain whether this project will succeed or fail after having taken into account both primary and lower order drivers of project success.

The paper is structured as follows. Section 2 describes the analysis method. The use of the method is illustrated by a case study (Section 3). We also compare the result of the extended method to the more limited method in [18]. The case study consists of 12 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. We also show that lower order success drivers (project characteristics) influence the success indicator and help reduce the class of "Yellow" projects. Additionally, the introduction of a class of projects considered "yellow" or "at risk" provides better assessment of the project before completion. In terms of guiding management actions, the method is also able to identify project characteristics that, while correlated positively with the project success indicator, clearly do not drive project success. Section 4 presents conclusions and suggests further work.

2. ANALYSIS METHOD

The analysis method consists of four phases, see Table 1. The first two phases are identical to those described in [18] with the extension introduced in [19]. The first phase consists of screening project variables to determine which project characteristics should be included in the analysis. One needs to decide which project characteristics influence project success. It is also important to define what constitutes success. Success measures can include internal or external indicators. Examples of internal ones are efficiency and maintainability. Examples of external ones are timeliness and reliability.

Table 1. Analysis method

Phases	Steps
I: Relevance	1. Screen project variables
II: Primary drivers	2. Identify primary drivers of success variable
	3. Rank projects based on primary drivers
	4. Classify into green, yellow and red (1/3, 1/3 and 1/3)
III: Lower order drivers	5. Remove lower level drivers
	6. Repeat Phase II for the current driver level
	7. Repeat Phase III until no more drivers can be identified
IV: Analyse and interpret	8. Re-classify projects based on all drivers

Once measures have been taken, it is important to determine whether the project variables thought to influence success actually do so. This constitutes phase I (step 1) of the analysis, "determining variables relevant to the analysis". Like [18], we consider positive correlation between a project variable and the success variable as a (preliminary) indicator that project and success variables are related. Project variables with a negative correlation are screened out.

Phase II determines the primary drivers of project success. It also uses them to analyse the relationship between projects ranked high on the basis of project variables identified as primary success drivers versus the ranking of the success variables itself. Thus, phase II consists of the following steps:

2. Identify primary success drivers, as in [18]. A principal component analysis (PCA) [10] groups correlated variables, i.e. it groups variables with a similar behaviour into a number of principal components where each component accounts for the maximum possible amount of variance for the variables being analysed.

   The number of principal components varies depending on the data and the particular extraction method. PCA in our analysis is of interest, because it identifies variables that vary together. In addition to its use to identify project variables and success variables that behave similarly [18]. PCA has also been used to analyse similarities between projects [11] for building models for prediction of fault-prone components [12], and for code decay [13].

   In our case, project variables that are grouped by PCA into the same principal component as the success variable are considered primary drivers of success. 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. For a more detailed discussion see [18].

   We recognise the limitations to using PCA in this context. 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 [5]. 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.

3. In this step, projects are ranked based on primary drivers and based on success variables [18]. The projects are ranked twice. First, the projects are ranked based on the success variable. The first ranking may actually be done directly after the data collection. Second, they are ranked using the project variables that were identified as primary drivers in step 2. The latter ranking is ideally based on the sum of the project variables with a loading higher than 0.7 [18]. 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 [18]. 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. This represents a failure to capture the project variables that drive success. Finally, the Spearman correlation is determined between the two rankings.

   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 primary success drivers. 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. This is the objective of step 3.

4. Here, a comparison is made between project classification based on primary success drivers versus the success variable. In [18], classification was merely based on identifying projects as successful or unsuccessful, i.e. upper and lower half respectively. Then a diffusion matrix was constructed to determine which projects were ranked in the same class by both rankings, and which were not. This is the basis for computing an agreement index [2] to determine to which degree a ranking based on project variables identified as primary success drivers could be used to predict success/failure as (later) measured by the success variable.

   This approach has several shortcomings. First, and foremost, it does not allow for "mediocre" projects or projects whose outcome is uncertain. This was addressed by introducing a third class for uncertain projects in [19]. The drawback with the extension using a third class was that it simply, based on the primary drivers, divided the projects into three classes with one third of the projects in each class. The problem that occurs is that other drivers may be able to say whether a specific project should be considered to be in one of the other classes rather than in the uncertain class (yellow class). This leads us to the second issue.

   Second, secondary success drivers may influence whether these projects end up as successes (green) or failures (red). Not considering lower order drivers may lead to less accurate prediction. To deal with these issues, we use the extended classification based on the primary success drivers before addressing the secondary and lower order drivers. The three classes are identified as follows:

   - Green: top third of the projects in the ranking based on the primary success drivers (possibly adjusted by ties in rank and with the objective of trying to keep the classes of equal size)
   - Red: bottom third of projects in the ranking based on the primary success drivers (possibly adjusted by ties in rank and with the objective of trying to keep the classes of equal size)
   - Yellow: remaining middle-ranked projects

   This is the starting point for addressing lower order success drivers.

Phase III accounts for secondary and lower order success drivers, that would have been ignored in the original method [18]. Phase III consists of two steps that are repeated until no more success drivers can be identified.

5. In this step, the primary success drivers (and possibly lower order success drivers depending on the possible repetition of this step) are removed. This includes success drivers that have been identified in Phase II and prior iterations of steps 5 and

6. This leaves lower level project variables for further consideration.
6. Phase II is performed on these variables and the project success variable.
7. If we in step 6 are unable to identify any more secondary or lower order project success drivers, phase II ends.

In this case, the remaining project variables are positively correlated with the project success variable, but do not drive project success.

Phase IV (step 8) takes the rankings and Green/Yellow/Red classifications based on primary, secondary and lower order project success drivers and combines them for an overall success analysis based on project variables. It then compares its prediction to post-project success as measured by the success variable.

For each success drivers (primary, secondary and so on), the projects are ranked and assigned a color: red for failure, green for success, and yellow for an uncertain or mediocre outcome. Thus each project is assigned a color for each success driver. These are then aggregated to determine a cumulative color as follows: ignoring yellow, assign the majority color (either green or red). If there is no majority color, assign yellow.

The basic idea is that this procedure should better identify projects correctly. It may be the case that a project is uncertain (yellow) for the primary success drivers, but on a secondary success driver it is clearly pinpointed as a successful project (green). In this case, it would probably be wrong to classify it as yellow and we believe that it is likely to be a green project. This is the motivation why we use majority rules and also ignoring the yellow classifications when possible. Using more than primary success drivers means that we will only obtain yellow projects when all levels of success drivers indicate uncertainty or there are contradictions between the success drivers, i.e. one success driver may indicate a green project and another driver a red project.

Thus, when all classifications have been aggregated, every project is assigned a classification of either Green (G), Yellow (Y) or Red (R) based on the project success drivers. G reflects projects that should turn out to be successful, R are those that most likely will not be very successful, and Y indicates projects about whose success there is uncertainty. The latter means that, given the project success drivers measured, it is not possible to determine whether these projects will succeed or fail.

To evaluate the quality of the prediction, we now use the same agreement index as in [6] and [18], but only for the projects classified as R or G (since there is no prediction for the ones labelled Y or uncertain). This was done successfully in [19], although there was a price in terms of the number of projects that could be classified as either green or red. The objective is really to use all levels of project success drivers in order to try to minimise the size of the yellow class. In [19], the yellow class contained one third of the projects. The different classifications are illustrated in a case study in the following section.

Several situations are possible in the final classification:

- There are a limited number of Y ("uncertain") projects. This means that the success drivers identified agree enough with each other, or complement each other enough to enable a clear classification for most projects. A high agreement index with the success indicator ranking speaks for the close relationship between success drivers and success variable. A low agreement index indicates a model that is not suited for predicting project success. This may happen for a variety of reasons, including:

  - identifying less appropriate project variables (i.e. selecting the wrong variables to start with)

  - low PCA values.

- Many project classified as Y ("uncertain"). This means that evaluation based on project variables identified as drivers for project success do not agree with each other and result in very different ranks - and thus classifications - for the projects analysed. This may be due to a variety of reasons, including:

  - inherent lack of predictability. Successful projects simply are successful for too many single reasons, or reasons not captured by the analysis.

  - contradicting rankings. One indicator may result in a high ranking (G) while another classifies the project as unsuccessful (R). One may be correct, or both may be wrong.

  When the analysis flags too many projects as "uncertain", the method is not much help. This is however no surprise since no method will be able to help all the time. So, what we are looking for is a method that:

  - makes good predictions for R and G projects for most projects.

  - does not classify too many projects as "uncertain".

Next, we illustrate and evaluate the extended method on a case study.

3. CASE STUDY

3.1 Data

The data in the 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 natural logarithm of the project effort before being ranked on a five-point scale. For all variables, both project variables and the success variable, the intention is that a higher score means that it is better, where better should be interpreted from an intuitive understanding of the variables. The scales were formulated based on expert judgement. For example, it is assumed that higher competence is better and hence the scale is used so that a higher score means higher competence in the project. The complete definition of the scales can be found in [17].

The 10 project variables are: complexity, competence, requirements stability, personnel turnover, geographical distribution, methods and tools, time pressure, information flow, top management priority of project, and project management.

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, [4].

The values of the success variable are listed in Table 2. Based on the success variable, the 12 projects may be ranked. The ranking may also be used to assign the project to successful and less successful projects, which we here denote green (G) and red (R). Originally, the intention was to divide the projects into two halves with six projects being green and six projects being red respectively. However, given the values of the success variable it was decided to include the seventh project among the green projects since its values for the success variable is close to that of the sixth project. The actual division is a matter of judgement and one might have chosen to include even the eighth project among the green ones. We did not do this.

Table 2. Success variable

Project	Value	Rank	Color
1	3.28	4	G
2	5.26	8	R
3	7.90	9	R
4	3.10	3	G
5	3.76	7	G
6	3.60	6	G
7	2.53	2	G
8	8.76	11	R
9	2.21	1	G
10	8.49	10	R
11	3.36	5	G
12	9.90	12	R

3.2 Analysis

3.2.1 Phase I: Relevance

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. This result is discussed in [18]. The highest correlation (0.678) with the success variable is obtained for requirements stability. The second highest is with project priority (0.606). 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.

3.2.2 Phase II: Primary drivers

2. Principal component analysis

   The principal components identified are shown in Table 3. 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.

Table 3. Identification of first order success drivers using PCA.

Orthogonal solution	Comp. 1	Comp. 2	Comp. 3
Requirement stability	0.504	0.724	-0.061
Staff Turnover	-0.188	0.845	-0.065
Time pressure	0.956	-0.002	0.121
Information flow	0.786	0.307	0.383
Priority	0.064	0.163	0.959
Project management	0.491	-0.248	0.754
Success variable	0.366	0.744	0.392

   The primary success drivers for this data set are identified as “Requirement stability” and “Staff turnover”. Both of these project variables are related to the stability of the project. In other words, changes in requirements or personnel may reduce the likelihood of success.

3. Ranking and correlation

   The projects have already been ranked based on the success variable, see Table 2. This ranking is now complemented with a ranking based 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 ranking is shown in Table 4. It should be noted that several ties appear in the sum of the two project variables, which is the reason why several ranks are the same in the table.

4. Classification

   The projects may now be classified into our color scheme using both the original proposal of having only two classes [18] and with the extension when a third class is introduced [19]. The results of the classifications are shown in Table 4.

   It is now possible to compare the ranking based on the primary success drivers and the ranking obtained for the actual success variable. This is done using a diffusion matrix for the two classification schemes used, see Table 5 and Table 6.

   The classification scheme in Table 5 shows that four projects are misclassified.

   It should be noted that for three classes, the projects classified as yellow based on the project variables are viewed as uncertain and hence it is not possible to compare them with the actual outcome using the classification based on the success variable. This is the reason why only seven projects are shown in Table 6.

Table 4. Ranking based on primary success drivers.

Project	Success rank	Color 2 classes	Color 3 classes
1	4	G	Y
2	2	G	G
3	4	G	Y
4	4	G	Y
5	4	G	Y
6	2	G	G
7	1	G	G
8	12	R	R
9	9	R	R
10	10	R	R
11	8	R	Y
12	10	R	R

Table 5. A diffusion matrix for the project classification using two classes.

Agreement index		Success variable	
0.31 (fair)		Green	Red
Project variables	Green	5	2
	Red	2	3

Table 6. A diffusion matrix for the project classification using three classes.

Agreement index		Success variable	
0.42 (moderate)		Green	Red
Project variables	Green	2	1
	Red	1	3

The advantage with the introduction of a third class is that fewer projects are misclassified on the other hand, we are unable to classify five projects.

These two tables show the necessity of trying to use lower order success drivers to increase the number of projects that are classified as green and red in comparison to Table 6, but without having too many misclassifications as in Table 5.

Based on the classifications, agreement indices are calculated. This is primarily done here for comparison with the results obtained when including lower order success drivers in the analysis. The agreement indices for the two classifications and their interpretation, using the Altman scale [2], are shown in the upper left corner of Tables 5 and 6. The agreement index increases with the introduction of a third class, although not substantially.

3.2.3 Phase III: Secondary and lower order drivers

5. For the analysis of the second order success drivers, the first order success drivers are removed from the analysis.
6. The steps in Phase II are redone. This results in two principal components as shown in Table 7. Here, the pattern is not as clear. The first component clearly includes “time pressure” and “information flow”, but the second component is not as clear, i.e. only one variable with a loading above 0.7. In addition, the success variable has loadings in the two components that are fairly close to each other. Here, we have two options, either we can state that we do not have any lower order success drivers since our main rule of a threshold of 0.7 does not hold, or we may include those variables with a loading higher than the loading of the success variable in the second component. The latter is in accordance with the rules formulated in [18], and we have chosen to view “Priority” and “Project management” as the second order success drivers. The way we have chosen to assign variables to principal components is indicated by the shaded cells in Table 7.

Table 7. Identification of second order success drivers using PCA.

Orthogonal solution	Comp. 1	Comp. 2
Time pressure	0.948	0.102
Information flow	0.858	0.394
Priority	0.059	0.984
Project management	0.457	0.658
Success variable	0.441	0.555

7. This concludes the analysis of second order success drivers, and the analysis is continued with step 5 again to see if it is possible to identify tertiary order success drivers.

This means that steps 5-7 are repeated.

5. The second order success drivers are now removed to allow for an analysis of the remaining variables.
6. This time the PCA results in only one component. This means that the two remaining project variables are grouped together

with the success variable, and hence both these variables are identified as tertiary order success drivers, see Table 8.

Table 8. Identification of tertiary order success drivers using PCA.

Orthogonal solution	Component 1
Time pressure	0.862
Information flow	0.932
Success variable	0.771

7. Out of the original 10 project variables collected for the projects, four variables were screened out based on having a negative correlation with the success variable, two variables each were identified as primary, secondary and tertiary order success drivers respectively.

The identification of success drivers forms the basis for classifying the projects according to the rules outlined in the method description. Moreover, it is possible to see how the use of the project variables for classification fits with the classification based on the actual success variable.

3.2.4 Phase IV: Analysis and interpretation

The projects are now classified for one set of success drivers at a time. The projects are classified into green, yellow and red according to the method. The classification according to the primary success variables has already been done, see Table 4. A similar classification is done based on the secondary and tertiary success drivers. This results in the classifications shown in Table 9. The three classifications are aggregated into a final classification of the projects, see Table 9, as discussed in step 8 of the method description in Section 2. In the final classification, it is interesting to note that only two projects are classified as being yellow. It should be remembered that one of the objectives was to decrease the size of the yellow class. The aggregate variable has managed to push projects from the yellow class into either green or red projects.

The classification based on the aggregate variable is now evaluated by comparing the classification with the actual outcome as listed in Table 2. The result of the evaluation can be found in Table 10. Generally, the results are positive. Only two projects are misclassified (out of 12 projects), and two projects are still yellow. The agreement index is still moderate, but on the high side (0.6 is the limit for moving into being good). However, the agreement index is clearly better and we are able to classify more projects. Thus, the method has for this particular data set succeeded, both in terms of increasing the agreement index and classifying more projects than we were able to when introducing a third class to take the uncertainty in the classification into account.

It is also interesting to note the following:

- The two best projects (7 and 9) are classified correctly.
- Project 4 (ranked as number 3) is the most misclassified project, i.e. it is furthest away from being correctly classified.
- The projects (1 and 11) ranked as 4 and 5 respectively are classified correctly.
- For the projects closest to the boundary based on the success variable, i.e. projects 6, 5, 2 and 3, two projects are pinpointed as yellow, which is no great surprise, one is classified correctly and one is misclassified. This shows that the projects are borderline and whether the classification is correct or not is probably more of a random nature.
- The three worst projects (8, 10 and 12) are classified correctly.

Table 9. Classification based on primary, secondary and tertiary success drivers.

Project	Primary	Secondary	Tertiary	Final
1	Y	G	G	G
2	G	R	G	G
3	Y	Y	Y	Y
4	Y	R	Y	R
5	Y	Y	G	G
6	G	R	Y	Y
7	G	G	Y	G
8	R	Y	R	R
9	R	G	G	G
10	R	R	R	R
11	Y	G	G	G
12	R	Y	Y	R

Table 10. A diffusion matrix for the project classification using three classes.

Agreement index		Success variable	
0.58 (moderate)		Green	Red
Project variables	Green	5	1
	Red	1	3

In summary, it is positive that the best and the worst projects are classified correctly. Moreover, it is no surprise that there is a problem with the projects close to the boundary between green and red projects. There is only one project that is starkly misclassified: project 4.

4. CONCLUSIONS

A method for using subjective variables to assess project success or failure has been introduced previously [18]. It has since been extended regarding project classification [19]. The extended method had a weakness: it classified many projects as being uncertain regarding outcome, i.e. the method is too conservative. Moreover, the extended method only takes primary success drivers into account. This paper has shown that it is possible to further refine the extended method and take lower order success drivers into account.

The method extension has been introduced and evaluated in a case study. The extension presented here is promising because it manages to reduce the number of projects being classified as uncertain in terms of project success, and it increases the number of correctly classified project as measured by an agreement index. In other words, the method is capable of quite accurately, in the conducted case study, to identify different types of projects based on the project variables.

The method may be used in many different ways: 1) to identify success drivers by studying the relation between project 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, as extended here, has increased the capability of an existing method to identify key project success drivers. The method as such is generic in the sense that it provides an opportunity to define your own success variables and identifying whatever project characteristics believed to be important to drive project success.

In summary, we have shown that it is feasible to use subjective factors for understanding and assessing software project success. The case study has highlighted that the method is useful to understand and assess the relationship between project characteristics and success. The method was able to identify several orders of success drivers.

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.

References

[1] Allen M., and Yeh W., “Introduction to Measurement Theory”, Brooks/Cole Publishing, 1979.

[2] Altman D., “Practical Statistics for Medical Research”, Chapman-Hall, 1991.

[3] Basili V., Zelkovitz M., McGarry F., Page J., Waligora S., and Pajerski R., “SEL's Software Process-Improvement Program”, IEEE Software, November pp. 83-87, 1995.

[4] Boehm, B., “Software Engineering Economics”, Englewood Cliffs, N.J., USA, Prentice-Hall, 1981.

[5] Briand L., El Emam K., and Morasca S., “On the Application of Measurement Theory in Software Engineering”, Journal of Empirical Software Engineering, Vol. 1, No.1: 61-88, 1996.

[6] El Emam K., “Benchmarking Kappa for Software Process Assessment Reliability Studies”, Empirical Software Engineering: An International Journal, Vol. 4, No. 4: 113-133, 1999.

[7] Gray A. R., MacDonell S. G., and Shepperd M. J., “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, 1999.

[8] Hughes R., “Expert Judgement as an Estimation Method”, Information and Software Technology, Vol. 38, pp. 67-75, 1996.

[9] Höst M., and Wohlin C., “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, 1998.

[10] Kachigan S. K., “Statistical Analysis – An Interdisciplinary Introduction to Univariate & Multivariate Methods”, Radius Press, 1986.

[11] Khoshgoftaar T. M., and Lanning D. L., “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, 1994.

[12] Khoshgoftaar T. M., Allen E. B., Kalaichelvan K. S., Goel N., Hudepohl J. P., and Mayrand J., “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, 1995.

[13] Ohlsson M. C., von Mayrhauser A., McGuire B., and Wohlin C., “Code Decay Analysis of Legacy Software through Successive Releases”, Proc. IEEE Aerospace Conf., Snowmass, Colorado, USA, 1999.

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

[15] Valett J. D., “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, 1993.

[16] von Mayrhauser, A., Wohlin, C. and Ohlsson, M. C., “Assessing and Understanding Efficiency and Success in Software Production”, Empirical Software Engineering: An International Journal, Vol. 5, No. 2, pp. 125-154, 2000.

[17] Wohlin C., and Ahlgren M., “Soft Factors and Their Impact on Time to Market”, Software Quality Journal, No. 4: 189-205, 1995.

[18] Wohlin, C. and Amschler Andrews, A., “Assessing Project Success using Subjective Evaluation Factors”, Software Quality Journal, Vol. 9, No. 1, pp. 43-70, 2000.

[19] Wohlin C., von Mayrhauser A., Höst M., and Regnell B., “Subjective Evaluation as a Tool for Learning from Software Project Success”, Information and Software Technology, Vol. 42, No. 14: 983-992, 2000.