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

Subjective Evaluation as a Tool for Learning from Software Project Success

Claes Wohlin[1], Anneliese von Mayrhauser[2], Martin Höst[1] and Björn Regnell[1]

[1]Dept. of Communication Systems
Lund University
Box 118, SE-221 00 Lund
Sweden

[2]Computer Science Department
Colorado State University
Fort Collins, CO 80523
USA

Abstract

This paper presents a method for using subjective factors to evaluate project success. The method is based on collection of subjective measures with respect to project characteristics and project success indicators. The paper introduces a new classification scheme for assessing software projects. Further, it is illustrated how the method may be used to predict software success using subjective measures of project characteristics. The classification scheme is illustrated in two case studies. The results are positive and encouraging for future development of the approach.

Keywords: Software success, project characteristic, subjective evaluation and project classification

1. Introduction

Information may be lost in software organisations if the human knowledge of software projects is not captured properly. Measurement of software projects and their success is a difficult undertaking. Often people “know” things that are very hard to measure with objective measures of effort, defects and the like. Examples include project competence or project management performance. This paper highlights how knowledge and experience of software developers, managers and customers may be captured and used. More specifically, this paper illustrates how subjective measures may be used as a complement to objective data.

A subjective measure may be defined as a measure that depends only on the knowledge and expertise of the people involved. A potential advantage of a subjective measure is that it is fairly easy to collect through interviews or questionnaires, and no extensive measurement program has to be in place. The obvious disadvantage is that a subjective measure may be less exact. Additionally, it may be hard to use the measures to draw conclusions. Instead of focusing on the use of objective measures alone, we would like to propose using subjective measures to complement objective measures. It appears that the main use of subjective measures in software engineering has been in conjunction with effort and cost estimation, see for example [2, 5, 8].

The following notion is used and an example is given below for illustration purposes. A subjective 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 may be divided 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 subjective measure is then defined to measure the 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 to judge it on a Likert scale.

The paper is outlined as follows. In Section 2, a newly proposed assessment method based on subjective evaluations is summarised and research questions are posed. An extension to the assessment method is presented in Section 3.

The extended method is illustrated and evaluated using data collected from two different organisations. In the case studies, it is also illustrated how the method may be used for prediction purposes. The two case studies are presented in Sections 4 and 5.

In summary, the paper shows how subjective measures can be used successfully to plan and control software projects to increase the likelihood of project success. Some conclusions from the method and the case studies are presented in Section 6.

2. Background

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 [12]. Some exceptions exist, primarily in relation to effort estimation, see for example [5, 8], and also lately in risk management [11]. The use of expert judgement as an estimation method is further discussed in [7]. Recently, a subjective evaluation method for assessing software has been developed [15].

The subjective evaluation method uses subjective measures in order to judge which project characteristics are most essential for project success. The method provides input to project planning and control as well as risk management [6], where the method could provide valuable input to which projects are particularly difficult to turn into a success. It should not be viewed as a stand-alone method, but rather as a complement and support to project and risk management techniques used today

This section presents a summary of the evaluation method. This means that some details have intentionally been omitted. A full description of the method is available in [15].

The first step is to identify which subjective factors to evaluate and identify project characteristics and success indicators that are believed to be important. Suitable variables should be identified for both project characteristics and success indicators. It must be decided which project and success variables should be evaluated subjectively, i.e. measured on the subjective scales defined. It is usually better if the variables are measured quantitatively, and hence it is essential to determine what to measure quantitatively and what to evaluate subjectively. Success variables may include both internal (development-oriented) success variables, for example efficiency and maintainability, and external indicators (customer-oriented), for example timeliness and quality. Thus, it is important to decide what to evaluate when judging the success of a project. In other words, it should be determined which success variables to measure. The decision may be based on certain objectives, for example, the key issue may be timeliness of delivery, or the objective is simply to capture project success in general. The project variables believed to influence the success variables should be identified and it should be determined how to measure them.

The scales for the subjective variables must be determined carefully. Different subjective rating schemes exist, for example, Likert and ordinal scales [4]. The meaning of the different values on the scale should be determined and these should preferably be defined so that we obtain a good spread between projects. Methods to define scales reliably are described in [13].

The following procedure is proposed in [15] to study the relationship between project characteristics and project success using subjective evaluation factors. The following analysis steps are then conducted for each success variable.

1. Correlation analysis (data 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 should have a positive correlation with the success variables. If that is not the case, then there are either some underlying aspects that were not captured, or the scale does not represent what we think. 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. We have chosen to screen variables with a negative correlation whether it is significant or not. Variables with low correlation (positive or negative) to the success variable are not expected to be pinpointed as important for success anyway.

2. Principal component analysis

Different measures are often highly correlated because they measure almost the same thing. Different techniques exist to extract a useful subset. One such technique is Principal Component Analysis (PCA) [9]. PCA groups correlated variables into a number of principal components, where each principal 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 criteria for extraction. PCA gives each variable a loading for each principal component. The loading is a measure of the variables correlation with the resulting principal component. When applying PCA to non-interval variables, it is necessary to be a little cautious, but it is normally not a problem as long as the outcome makes sense. PCA is also applied to variables measured on Likert scale in [11]. The caution may, for example, include not accepting the results from the analysis without making sure that the results correspond to intuitive expectations. The analysis is conducted for the combination of project variables and the success variable. The objective is to identify which project variables vary together with the success variable. The loadings in the principal component containing the success variable are of particular interest in this analysis. A project variable is considered being in the same principal component as the success variable if the project variable has its highest loading in this component.

3. Ranking and correlation

The projects are ranked twice. First, the projects are ranked based on the success variable, and secondly they are ranked using the project variables in the same principal component as the success variable. The latter ranking is based on the sum of the project variables with a loading higher than a certain threshold. Finally, the Spearman correlation is determined to evaluate the level of relation between the two rankings.

4. Classification

The two rankings reflect different ways for evaluating a project. The first ranking models the project success based on one success variable. The second ranking also describes the projects based on project success, but this time based on the values of key project variables. The rankings are used to classify projects. In the method, it is proposed to use two classes to classify projects as either being successful or unsuccessful. This is done using both the ranking from the project variables and the ranking from the success variable. The correspondence between the two classifications can be shown in a classification table; sometimes this type of table is referred to as a diffusion matrix or contingency table. Basically, the table compares classifications and optimally all cells except for the diagonal should be zero. An example is shown in Table 1.

Table 1: A classification table.

		Classification based on success indicators	
		Successful	Unsuccessful
Classification based on project characteristics	Successful		
	Unsuccessful		

5. Agreement index

If, in the long run, we want to learn how to predict project success based on project variables, the classifications based on the two rankings must show some degree of agreement. Agreement can be measured by an agreement index, often referred to as kappa statistic [1]. In software engineering, the kappa statistic has, for example, been applied to inter-rater agreement of process assessments [3]. A brief description of the kappa statistic is provided in [15].

To be able to understand the degree of agreement, it is necessary to interpret the kappa statistic on a scale describing the degree of agreement. Several such scales exist, but there are only minor differences. Three scales are presented in [3]. Here the benchmark suggested by Altman [1] is used, see Table 2.

Table 2: A mapping between the kappa statistic and an interpretation [1].

Kappa statistics	≤ 0.20	0.21-0.40	0.41-0.60	0.61-0.80	0.81-1.00
Strength of agreement	Poor	Fair	Moderate	Good	Very good

The agreement index is normally calculated for the agreement between, for example, different people using the same scale. This is not the situation here, since the two rankings are based on different variables and scales, and hence the classifications based on the rankings are based on different scales. This means that the agreement index may be lower than if the same scales were used. All the same, the kappa statistic is used to compare the two classifications i.e. one from project variables and the other from success variables. The use of the kappa statistic under these circumstances is an area for further study.

These five steps form the basic evaluation method for subjective factors. In the above steps, it is assumed that projects are classified as being successful or unsuccessful based on project characteristics and success indicators. This results in a 2 * 2 matrix for which the kappa statistic is calculated. One drawback with this approach is that borderline projects are easily misclassified as a potentially successful or unsuccessful project, although it turns out that it should have been classified the other way around. This leads to two research questions that need to be answered:

- How do we best introduce third class when classifying the projects based on project variables, i.e. a potentially problematic class? This additional class is only needed when classifying the projects using the project characteristics, since after having measured the success indicators it is assumed that it is known whether the project became a success or not. The middle class is introduced to cope with uncertainty in the classification, and projects belonging to this class are not viewed as being misclassified.

- How good is the new classification scheme in comparison with the original proposal?

In addition, the basic method has been presented as an assessment method, but it is essential to be able to use the method for prediction purposes. If we were able to estimate the project variables in advance, then we could predict the level of anticipated success for different projects. Another option would be to formulate requirements on the project variables given that we have certain objectives regarding the project success. If the method could be applied for both these situations, it would be even more valuable. This leads to the third research question.

- Is it possible to use the method successfully for prediction rather than only assessment?

The three research questions are answered through the description of the new classification scheme in the next section, and through illustration of the new classification scheme and comparison with the original scheme in two case studies, see Sections 4 and 5.

3. Method extension: Green, yellow and red projects

The new classification scheme divides projects into green, yellow or red based on their project characteristics. Threshold values between the different classes may be chosen by each specific organisation using the method. The scheme is illustrated by classifying the projects based on project characteristics into three equally large groups, i.e. upper third (green), middle third (yellow) and lower third (red). Completed projects are classified as either green (upper half) or red (lower half) based on the success indicators.

The use of three classes allows us to indicate the uncertainty in classifying projects. This is more realistic than having only two classes. Two classes imply that we view the world in only black and white (or in our case red and green). The subjective evaluations are unable to capture all facets of a software project and hence there must be room for some uncertainty.

Most classification schemes try to make exact classifications, and the success of the schemes is evaluated based on their ability to make exact classifications, see for example [10]. This is an ambitious and commendable approach, but it may also be very difficult to obtain low misclassification rates. The approach, suggested in this paper, tries to partly avoid this problem by accepting that there is an uncertainty.

The classification of projects may hence be summarised in a table with three rows and two columns, which can be compared with Table 1.

A set of projects may now be used to evaluate which project characteristics should be used for the classification. This is done using the basic evaluation method outlined in Section 2. The projects may then be classified by extending Table 1. Now it is possible to evaluate the correctness of the classification, since the success indicators are known. These projects are referred to as the fit projects, since they are used to build the classification and hence determine the boundaries between the different classes. New projects should then be classified using the project characteristics. The classification can then help in planning and tracking the project to achieve whatever success indicators are most important. Examples of usage are further discussed in the case studies presented in Sections 4 and 5.

Including a yellow class also means that the calculation and interpretation of the kappa statistic need adaptation. Previously, in the basic evaluation method, it was simple to calculate the agreement index; projects were simply classified as either green or red for both project characteristics and success indicators. This is no longer the case, since the scales are different. Two options exist:

- The first option is to disregard the projects that are classified as yellow using the project characteristics, and calculate the kappa statistics only based on the projects classified as either green or red for both project characteristics and success indicators.
- Another option is to view projects classified as yellow as always being correctly classified. The use of a yellow class has given managers on different levels a warning that a project is at risk of becoming unsuccessful. It should be noted that this option would in most cases result in a higher kappa statistic than the first option.

Regardless, the classification table provides valuable input in terms of expectations of the projects given that the project characteristics are not changed. In particular, it provides an opportunity of trying to improve certain project characteristics based on the importance of the projects becoming a success or not. Further, the project characteristics should preferably be tracked and re-estimated regularly in the project. Of course, it would be beneficial if all projects were successful, and may be they all are successes, but there are always projects that are more successful than others. In other words, the method presented here provides important input for informed decisions and prioritisation between different projects.

Next the complete method, including the extension of the classification table, is illustrated in two case studies.

4. Case study I

4.1. Environment and data

The data in the first case study is from 12 software projects from one company consisting of many organisations. The company works with projects in the telecommunication domain. Data is collected for 10 project variables that were considered critical in improving the predictability of time to market, and they are measured on a scale from 1 to 5 based on project reports and interviews with project participants. Only one success variable is important in relation to this objective, i.e. the lead-time. However, to make the lead-times comparable, they are normalised with the project effort before being judged on a five-point scale.

The 10 project variables are Problem Complexity, Competence, Requirements Stability, Staff Turnover, Geographical Distribution, Method and Tools, Time Pressure, Information Flow, Top Management Priority and Project Management. Once again, it may be appropriate to emphasise that the objective has been to formulate subjective ordinal scales in accordance with intuition. This means that a higher value for a variable is assumed to be better. The complete scales can be found in [14]. The scale is formulated based on experience from the company where the data is collected. The same person collected the data. The sources of information are final project reports in combination with interviews with some key personnel in the projects.

4.2. Method application

The five steps in the method are first conducted for 9 of the 12 projects to create an experience base that can be used when new projects are planned. To be fair, nine projects are chosen at random to create the experience base. The number of projects in the case study is fairly limited, but sufficient to illustrate the use of the method. Because of the small sample size, the random selection may affect the results in terms of screening and the principal component analyses. The steps results in the following:

1. Data screening

Seven of the ten project characteristics are positively correlated with the success variable. The other three variables are removed from the analysis. The three variables with negative correlation are Competence, Geographical Distribution, and Methods and Tools. This outcome may sound surprising, but it is possible to identify potential explanations. For example, one potential explanation regarding competence is that the people with the most experience are assigned to the most difficult project, and hence standard management practices have influenced the outcome. This is one possible explanation, and it illustrates why it is important to realise and accept that the input to the method is not independent of other decisions within an organisation. In particular, it illustrates why data screening is essential so that the remaining data corresponds to intuition.

The highest correlation (0.73) is between Requirements Stability and the success variable, and second highest is with Priority (0.70). It should be noted that the following principal component analyses could only be carried out if the number of subjective variables is less than the number of projects. If this is not the case, variables have to be removed based on other criteria.

2. Principal component analysis

As a rule of thumb, based on experience, the focus is on loadings above 0.7. This choice is based on the objective that we want each variable to clearly relate to one principal component. The key issue in the principal component analysis is which project variables that are grouped (loading above 0.7) together with the success variable. It may also be interesting to study which variables are grouped together in general to study which variables that vary together. The latter is however outside the scope of this paper.

The success variable is placed together with Requirements Stability. This is the only project variable having a loading above 0.7. The success variable also has a loading above 0.7, so it is quite clear that the success variable and Requirements Stability seem to vary together. Thus, it is expected that the Requirements Stability should be a good predictor of success, i.e. with this particular success variable. Given the analysis here it is now possible to rank the projects.

3. Ranking and correlation

The projects are ranked based on the project characteristic(s) in the same principal component as the success variable given the threshold of 0.7. This means, in this particular case, that the nine projects are ranked based on the Requirements Stability and the success variable. The Spearman correlation between the two rankings becomes 0.78 (corrected for ties in ranks). Thus, it seems, as Requirements Stability may be a good indicator of potential success.

4. Classification

Based on the two rankings, it is now possible to classify projects using both the original classification scheme and the new extension introduced in Section 3. We use the colouring scheme, although in the original classification scheme, we only have green and red projects. The results are shown in Tables 3 and 4 respectively.

The tables indicate the quality of fit between the predictive model (project characteristics) and the actual outcome (success indicator). Since the rankings are only based on one variable, there are several ties, which influence where to draw the boundaries between the classes. Letting projects belong to the better class solves this. For example, in Tables 3 and 4 three projects are ranked as number three and they are all classified as green projects based on project characteristics. This results in five projects being classified as green based on project characteristics. A similar case occurs based on the success variable. The ranks are given within parenthesis for each project with the first rank denoting the rank based on project characteristics.

Table 3: A classification table for the nine projects, denoted P1-P9.

		Classification based on Lead Time	
		Green	Red
Classification based on Requirements Stability	Green	P6(1,1), P5(2,4), P1(3,4), P3(3,3) and P8(3,1)	
	Red	P4(6,4)	P2(6,7), P9(6,9) and P7(9,8)

The fit is good with the original classification scheme, i.e. only one project is misclassified. P4 is classified as red based on the project variables, but it is viewed as a success when using the success variable. It should be noted that P4 is ranked fairly close to the boundary so it is likely that the new classification scheme pinpoints P4 as an uncertain project, i.e. a yellow project. In Table 4, we can see that this is exactly the case.

The fit, indicated by Table 4, is very good. It is noticeable that no project is placed in Green/Red and Red/Green respectively. P4 is now classified as yellow. The drawback is that two projects, which were correctly classified as red, now are viewed as being yellow. On the other hand, it is better to express the uncertainty instead of taking a chance that place a project in green or red, although we have a 50% chance of guessing correct.

Table 4: A classification table for the nine projects, denoted P1-P9.

		Classification based on Lead Time	
		Green	Red
Classification based on Requirements Stability	Green	P6(1,1), P5(2,4), P1(3,4), P3(3,3) and P8(3,1)	
	Yellow	P4(6,4)	P2(6,7) and P9(6,9)
	Red		P7(9,8)

5. Agreement index

The agreement index may now be calculated for the two classification schemes. For the classification in Table 3, the agreement index becomes 0.77, which is good according to Table 2. For the classification scheme in Table 4, it is possible to calculate the agreement index using the two options discussed in the extension of the method. In this particular case, the agreement index actually becomes 1.0 for both options, due to that there is no projects in the cells Green/Red and Red/Green. This is fairly unexpected, although very positive. The kappa statistic is not likely to be as good for all data sets, but all the same it is encouraging to see that the wrongly classified project in the original classification scheme becomes a yellow project for the new scheme. The price to pay is that two project that were correctly classified now become yellow projects.

4.3. Prediction of success

The five steps of the method have now been gone through to create classification tables that can be used when new projects are to be executed. The use of the tables can be evaluated using the three projects not included in the previous analysis, although we have chosen only to illustrate how the new classification scheme can be used for prediction. The ranks are re-calculated and the placement of the new projects determines their class. Initially, the actual value on the success variable is unknown and hence the planning of the project has to be done based on the classification from the project characteristics. It should be noted that one could argue that Requirements Stability is unknown when starting the project, but we would like to argue that it is never completely unknown. Several options exist:

- The Requirements Stability may be estimated, based on for example previous experience regarding this specific customer or application type.
- The required stability could be estimated given the aim in terms of success. In other words, how important is the project?
- The knowledge that Requirements Stability is crucial for success could be used in discussion with the customer. We should make the customer aware of the coupling between Requirements Stability and success.

Assuming that it is possible to estimate Requirements Stability, the results are summarised in Table 5.

Table 5: Results from using the extended classification table.

	P10	P11	P12
Class from Requirements Stability	Green	Red	Yellow
Class from Lead Time	Red	Red	Green

From Table 5, it may be noted that one of the new projects (P10) was expected to become a success (green), but became unsuccessful (red). In this particular case, it was not enough with good requirements stability; other factors have obviously made the project less successful. P11 is predicted to become a red project and this is true. If project P11 was really crucial and important, management could (based on the result of this analysis) try to ensure that the requirements become more stable. There is also an opportunity to talk to the customer and explain the probable outcome based on that unstable requirements are expected. The method has for P11 provided management with important input to their decision process. Finally, P12 is likely to become either successful or unsuccessful; it is classified as yellow. Thus, management has information that means that other precautions may be taken. This may, for example, mean assigning one of their best project leaders or performing a very thorough risk assessment. If the original classification scheme is used then the projects classified as green and red would not change, but the project classified as yellow in the extended classification scheme would be wrongly classified as red although it turns out to be green. This indicates the usefulness of acknowledging uncertainty in the classification based on project characteristics.

5. Case study II

5.1. Environment and data

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 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. In this particular case, we have chosen to limit the study to two of the success variables. The two variables are Timeliness of Delivery, which is the project variable closest to the success variable in the first case study, and Quality of the Software. The latter is chosen since it illustrates another key success indicator of a software project.

5.2. Method application

In a similar way as in the first case, a subset of the projects is selected at random to build an experience base before the method is applied for prediction for the remaining projects. The method is applied to 36 projects, out of 46 projects, chosen at random. The descriptions of steps 1-3 and 5 are deliberately kept brief to focus on the use of the classification scheme and the ability of the method to make predictions. Each step in the method is conducted for each of the two success variables. They are denoted a) and b).

1. Correlation analysis (data screening)

a) Timeliness of Delivery

Three variables are removed from the analysis based on the screening. They are Team Experience, Experience of the Development Environment and Staff Turnover. It is interesting to note that once again there is a problem with variables related to competence and experience.

b) Quality of the Software

For this variable, two variables are removed. They are Team Experience and Staff Turnover. In other words, two of the variables that were removed from the other success variable studied in this case study.

2. Principal component analysis

a) Timeliness of Delivery

The success variable is grouped together with two of the remaining 24 project variables. All three variables have a loading above 0.7. The two project variables are Project Planning and Compliance to Project Plan. It seems likely that we have a timely delivery of the software is obtained if having a good project plan and ability to follow the plan.

b) Quality of the Software

The result of the PCA is very similar to the one obtained for the first success variable. The same two project variables are pinpointed as crucial for success.

3. Ranking and correlation

As before, the projects are ranked based on the project characteristics, i.e. the sum of Project Planning and Compliance to Project Plan, and the success variable.

a) Timeliness of Delivery

The Spearman correlation for Timeliness of Delivery with the sum of the two project characteristics becomes 0.65 (corrected for ties in ranks).

b) Quality of the Software

The Spearman correlation for Timeliness of Delivery with the sum of the two project characteristics becomes 0.61 (corrected for ties in ranks).

4. Classification

a) Timeliness of Delivery

The classification of the 36 projects using the original model is shown in Table 6, where it is shown how 11 projects are misclassified, although a large number of projects are classified correctly. It should be noted that considerably more than half of the projects is considered to be green, using the success variable, due to ties.

Table 6: A classification table for the 36 projects, denoted P1-P36.

		Classification based on Timeliness of Delivery	
		Green	Red
Classification based on Project Plan + Compliance to Project Plan	Green	P2, P4, P6, P9, P12, P13, P15, P17, P21, P22, P24, P26, P28 and P31-P35.	
	Red	P1, P5, P8, P10, P11, P14, P23, P27, P29, P30 and P36	P3, P7, P16. P18, P19, P20 and P25

The main intention with introducing a third class for project variables is to lower the number of misclassified projects, although there is a price to pay in terms of projects that were correctly classified earlier now move into the yellow class too. In Table 7, it can be seen how nine of the misclassified projects become yellow (only two projects are misclassified), but on the other hand seven projects that were correctly classified now have become yellow projects. The projects are obviously on the boundary, although most of them turn out to be green in the end. The latter is due to the fact that there are many ties.

Table 7: A classification table for the 36 projects, denoted P1-P36.

		Classification based on Timeliness of Delivery	
		Green	Red
Classification based on Project Plan + Compliance to Project Plan	Green	P4, P9, P12, P13, P15, P17, P21, P22, P24, P26, P28, P32 and P35	
	Yellow	P2, P5, P6, P8, P10, P11, P14, P23, P27, P30, P31, P33, P34 and P36	P19 and P25
	Red	P1 and P29	P3, P7, P16, P18 and P20

b) Quality of the Software

In a similar way as for Timeliness of Delivery, it is possible to classify the projects based on Quality of the Software. The results, when using the original classification scheme, are shown in Table 8. In this case 8 projects are misclassified.

Table 8: A classification table for the 36 projects, denoted P1-P36.

		Classification based on Quality of Software	
		Green	Red
Classification based on Project Plan + Compliance to Project Plan	Green	P2, P4, P6, P9, P12, P13, P15, P17, P22, P24, P26, P28 and P31-P35.	P21
	Red	P10, P14, P19, P23, P25, P29 and P36	P1, P3, P5, P7, P8, P11, P16, P18, P20, P27 and P30

Introducing the new classification scheme means that six of the misclassfied projects move into the yellow class. This is according to the objective, but at the same time some projects that were correctly classified also move into the yellow class.

Table 9: A classification table for the 36 projects, denoted P1-P36.

		Classification based on Quality of Software	
		Green	Red
Classification based on Project Plan + Compliance to Project Plan	Green	P4, P9, P12, P13, P15, P17, P22, P24, P26, P28, P32 and P35	P21
	Yellow	P2, P6, P10, P14, P19, P23, P25, P31, P33, P34 and P36	P5, P8, P11, P27 and P30
	Red	P29	P1, P3, P7, P16, P18 and P20

The actual choice of which classifications scheme to use is up to the user of the method. It should, however, be remembered that only having two classes means that we have 50% chance of guessing the right class. Guessing is not a good method, and hence it is important to apply a method that outperforms guessing. To measure the level of agreement between the classifications based on project variables versus success variable, it is possible to use the agreement index, which take randomness into account in the calculation.

5. Agreement index

- Timeliness of Delivery

  The kappa statistic is shown in Table 10 for the original classification scheme and for the two different options of calculating kappa for the new classification scheme. It is notable that the kappa statistic improves considerably when introducing the new scheme. It shows that the new scheme decreases the number of misclassifications. This indicates that the new scheme is able to identify the borderline projects in particular. The negative aspect is, of course, that some projects that are correctly classified with the original scheme become yellow projects.

Table 10: Kappa statistic for Timeliness of Delivery.

	Original scheme	New scheme Option 1	New scheme Option 2
Kappa statistic	0.39	0.76	0.84
Interpretation of kappa	Fair	Good	Very good

- Quality of the Software

  The kappa statistic is shown in Table 11. The results are similar as for Timeliness of Delivery, although the difference between the original scheme and the new scheme in terms of agreement index is not as big.

Table 11: Kappa statistic for Quality of Software.

	Original scheme	New scheme Option 1	New scheme Option 2
Kappa statistic	0.56	0.78	0.88
Interpretation of kappa	Moderate	Good	Very good

5.3. Prediction of success

The 10 projects not used in identifying the key project characteristics for success are now used for prediction purposes. The two key characteristics are summarised for the ten projects and the rank is determined and compared with the 36 existing projects. The rank determines the colour of the projects. The classification obtained based on the characteristics is then compared with the actual outcome. The results are shown in Table 12 for both success variables, which is possible due to that the same key characteristics are pinpointed for the two success variables.

Table 12: Results from using the extended classification table with G = Green, Y = yellow and R = Red.

	P37	P38	P39	P40	P41	P42	P43	P44	P45	P46
Class from Project Plan + Compliance to Project Plan	G	G	Y	G	Y	Y	Y	Y	Y	G
Timeliness of Delivery	G	G	G	G	G	G	G	G	G	G
Quality of Software	G	G	R	G	G	G	G	G	G	G

The random selection of projects in this particular case resulted in a very large number of green projects, which is partly due to random effects and partly can be explained with ties in the ranking. All projects classified as green from the project characteristics also become green projects. The yellow projects mostly become green projects too with one exception where project P39 become green for Timeliness of Delivery and red for Quality of Software.

If using the original classification scheme then all green projects would still be green, but the yellow projects would have been forced into being classified as either green or red based on the project characteristics. The results from the original classification scheme for the six projects classified as yellow are very interesting. P39 would have been classified as a green project. This would have been correct for Timeliness of Delivery, but wrong when looking at the Quality of the Software. Thus, the yellow class is really valuable in this case when the outcome is different for the two success variables. P41 would have been correctly classified as green, but project P42-P45 would have been wrongly classified as red projects. All four projects have a rank of 25, and the cut off between green and red is 23 so it is clear that the projects are on the boundary, which is acknowledged by the new classification scheme through the introduction of the yellow class.

In summary for the second case study, the introduction of the yellow class is really valuable. It helps both in achieving a higher agreement index, but it is also useful in the prediction of project success.

6. Summary and conclusions

This paper has briefly presented a method for using subjective factors in project evaluation and planning. The method may be used during the early phases of a project. At this stage of the project, it is necessary to determine a set of project characteristics on an ordinal scale to be able to use the proposed method. The method is based on that subjective factors are investigated in a number of projects to form an experience base that may be used in new projects. It is also expected that the experience base be updated as new projects are conducted.

In particular, this paper introduces a new classification table where projects may be classified into three classes based on project characteristics and two classes based on a success indicator. The main benefit with the classification table is that it is possible to indicate uncertainty in the classification rather than just dividing projects into successful and unsuccessful respectively. In addition, the paper has compared the original classification scheme with the extension proposed in this paper. Moreover, the method has for the first time been applied for prediction of software project success instead of being applied only to assess software projects.

The case studies, illustrating the method, showed that it is possible to use subjective evaluations to help in project planning. The extension into three classes allows for a more realistic division of projects based on project characteristics than using two classes, where there is no room to model uncertainty. The uncertainty is needed in the model, since it is unrealistic to believe that the model is able to capture all different aspects of as software project.

The results can be summarised by returning to the three research questions posed when describing the existing evaluation method:

- Q: How do we best introduce third class when classifying the projects based on project variables, i.e. a potentially problematic class?
  A: The new classification scheme, with a yellow class, helps us to capture the uncertainty in the classification.
- Q: How good is the new classification scheme in comparison with the original proposal?
  A: A higher agreement index is achieved with the new classification scheme, which shows that the new scheme is particularly good at identifying projects close to the boundary between green and red projects in the original classification.
- Is it possible to use the method successfully for prediction rather than only assessment?

The method is clearly useful for prediction purposes, in particular with the new classification scheme. The new scheme is, in particular for the second case study, very valuable since it is able to classify several projects as yellow instead of misclassifying them, as would have been the result with the original classification scheme.

Further work includes new case studies, prioritisation between different success indicators and further improvements of the method.

7. References

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

[2] Briand, L C, El Emam, K and Bomarius, F, COBRA: A Hybrid Method for Software Cost Estimation, Benchmarking, and Risk Assessment, Proceedings IEEE International Conference on Software Engineering (1998), 390-399.

[3] El Emam, K, Benchmarking Kappa: Interater Agreement in Software Process Assessment, Empirical Software Engineering, (1999) Vol. 4, 113-133.

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

[5] 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, Proceedings of the Sixth International Software Metrics Symposium (1999), 216-227.

[6] Hall, E M, Managing Risk: Methods for Software Systems Development (Addison-Wesley, 1998).

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

[8] Höst, M and Wohlin, C, An Experimental Study of Individual Subjective Effort Estimations and Combinations of the Estimates, Proceedings IEEE International Conference on Software Engineering (1998) 332-339.

[9] Kachigan, S K, Statistical Analysis: An Interdisciplinary Introduction to Univariate and Multivariate Methods (Radius Press, 1986).

[10] Khoshgoftaar, T M and Allen, E B, A Comparative Study of Ordering and Classification of Fault-Prone Modules, Empirical Software Engineering: An International Journal (1999), Vol. 4, No. 2, 159-186.

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

[12] Vallett, J and Condon, S E, The (Mis)use of Subjective Process Measures in Software Engineering, Proceedings of the Eighteenth Annual Software Engineering Workshop (1993), Software Engineering Laboratory Series SEL-93-003, NASA Goddard Space Flight Center, 161-168.

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

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

[15] Wohlin, C and von Mayrhauser, A, Assessing Project Success using Subjective Evaluation Factors. In review. Submitted to Software Quality Journal (1999). A copy may be obtained from C. Wohlin.