Success or Failure? Modeling the Likelihood of Software Process Improvement

Khaled El Emam
Fraunhofer IESE
Sauerwiesen 6
Kaiserslautern
D-67661
Germany
elemam@iese.fhg.de

Dennis Goldenson
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890
USA
dg@sei.cmu.edu

James McCurley
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890
USA
jmccurle@sei.cmu.edu

James Herbsleb
Software Production Research Dept.
Lucent Technologies
Naperville, IL
USA
herbsleb@research.bell-labs.com

International Software Engineering Research Network technical report ISERN-98-15

Success or Failure? Modeling the Likelihood of Software Process Improvement*

Khaled El Emam
Fraunhofer IESE
Sauerwiesen 6
Kaiserslautern
D-67661
Germany
elemam@iese.fhg.de

Dennis Goldenson
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890
USA
dg@sei.cmu.edu

James McCurley
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890
USA
jmccurle@sei.cmu.edu

James Herbsleb
Software Production Research Dept.
Lucent Technologies
Naperville, IL
USA
herbsleb@research.bell-labs.com

Abstract

In this paper we present the results of a reanalysis of a study of factors that influence the success of software process improvement [6]. The initial report relied on simple statistical analytic methods, largely univariate and bivariate statistics. The current multivariate analysis replicates the basic results of the earlier study, while adding additional insights about the interactions among and comparative importance of the factors that make process improvement efforts likely to succeed or fail.

1 Introduction

Software process improvement has become big business worldwide. Based in part on roots in the total quality movement following World War II, the first SEI-assisted assessment was conducted by the Software Engineering Institute (SEI) in 1987 [10]. By now thousands of organizations have been formally assessed either as part of competitions for source selection and/or by internal initiatives to improve their own organizational process capabilities.

Software process assessments can be a powerful tool for initiating and sustaining software process improvement (SPI) [7][13][4][6][8][9]. There is in fact a growing body of evidence that improving process capabilities can pay off substantially in product quality and business value. However, evidence also shows that SPI programs sometimes fail as well. Moreover, there remain few systematic empirical investigations about the conditions under which SPI initiatives vary in their results. Guidance about implementing process improvement still is based largely on anecdotal evidence and expert judgment.

In this paper we present a multivariate model of the conditions (e.g., organization and funding of improvement efforts) that can explain the successes and failures of software process improvement efforts. The model is constructed using a classification tree algorithm which identifies the contribution of factors that affect the outcome of SPI efforts, and describes how those factors interact with each other to influence success or failure.

Briefly, our results indicate that the most important factor in distinguishing between success and failure of SPI efforts is the extent to which the organization is focused in its improvement effort, with clearly defined goals and consistent directions set by senior management. However we also found that the impact of focus in SPI initiatives depends on how it is combined with organizational commitment to process improvement and with the existence of organizational politics.

In Section 2, we present our research method, including the source of the data and the analysis methods we use. This is followed in Section 3 with the detailed results and their interpretation. We conclude the paper in Section 4 by summarizing the analytical results and providing directions for future work.

* This work is sponsored in part by the United States Department of Defense.

2 Research Method

Most empirical studies of the impact of software process improvement rely on very simple statistical and analytical methods such as percentage tables, charts, and related univariate and bivariate statistics [23][6]. Sample sizes typically are small, with correspondingly few degrees of freedom to support multivariate analysis. Moreover process improvement professionals tend to ignore or downplay issues of statistical rigor.

A good deal of insight can in fact be gained from simple analytical techniques, but such methods also can miss important subtleties. Moreover, they can understate the magnitude of relationships by relying on inaccurate measures with a low signal to noise ratio. It is our task as software engineering empirical analysts to define and conduct methodologically defensible studies and present their results in a manner that is both widely understandable and actionable for practitioners in our field.

The data we use in our study were originally collected in a sample survey of organizations that conducted assessments based on the Capability Maturity Model for Software (CMM®) [6][9]. The original analysis relied largely on simple univariate and first-order bivariate statistics. Here we construct a multivariate model that takes into account interactions among several explanatory factors.

2.1 Data Source

The survey took place one to three years after the assessments were conducted, allowing sufficient time for changes to occur yet recent enough to expect accurate recall from the respondents. The organizations vary considerably in size and are from a wide variety of sectors and domains ranging from embedded military systems through commercial MIS.

The sample was drawn in September 1994 from the SEI's Process Appraisal Information System (PAIS) database. The assessments were conducted in the USA and Canada during calendar years 1992 and 1993. As described more fully in [6] not all of the original points of contact from the database were easily accessible, and obtaining individual contact information was sometimes difficult. However, there is no *a priori* reason to expect any bias in the sample of 61 appraisals.

In particular, the appraisals in the sample do not appear to be self-selected. Moreover, the survey respondents report widely varying degrees of success in their SPI efforts subsequent to their appraisals. Even if the organizations included in our analysis are somehow more successful than others in their SPI efforts, there would have to be very substantial bias in the sample to invalidate our basic results. All told, we received completed questionnaires from 138 respondents (83% of those sent, representing 92% of the organizations that we sampled).

2.2 Roles of Respondents

People who fill different roles in an organization may differ in their perspective about the same events. Hence we purposefully constructed the sample to include individuals who might be expected to differ as a result of their roles in the organization: (1) the project level software manager most knowledgeable about the appraisal; (2) the most knowledgeable and well-respected senior developer or similar technical person; (3) an organizational level SEPG manager, or someone with equivalent responsibilities for SPI.

Contrary to our original expectations, we did not find any consistent role differences among the survey respondents. We found only two statistically significant relationships ($p < 0.05$ using a chi-square test) among all of the survey questions – far fewer than would be expected by chance alone. While we continue to look for more subtle role differences in multivariate response patterns, we have excluded role differences from the analyses reported in this paper.

2.3 Model Specification

Our objective was to construct a multivariate model that characterizes SPI Success, as graphically represented in Figure 1. In this model, we hypothesize that there are two classes of independent variables that influence SPI success: organizational factors and process factors. We also expect there to be interactions among the two classes of independent variables.

® CMM is registered in the U.S. Patent and Trademark Office. Capability Maturity Model is a service mark of Carnegie Mellon University.

Since there are no existing theories in the SPI literature that are amenable to direct empirical confirmation (i.e., they are not stated explicitly in terms of a causal model that can readily operationalized), we specify our model at a high level only. We then use exploratory data analysis techniques to operationalize the high level model.

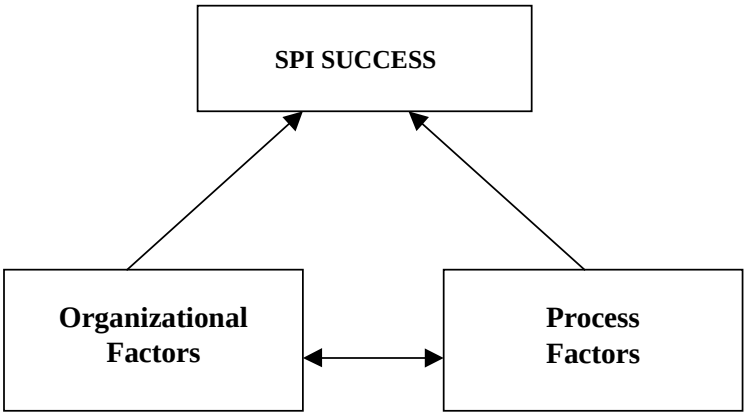

Figure 1: Overall specification of model being tested.

Organizational factors are those variables that characterize the organization undergoing SPI, and the characteristics of the organizational SPI effort itself. The variables we selected for analysis are summarized in Table 1. These represent the types of organizational issues that are commonly recommended for ensuring a successful process improvement effort [5][16][14][15].

Process factors comprise those variables that characterize activities or infrastructure that are believed to be necessary for a successful SPI effort. Process variables included in our analysis are summarized in Table 2. Explicit action planning (PROC1) is often said to provide a necessary basis for process improvement after an assessment [19]. The establishment of a functioning SEPG (PROC3) is strongly recommended for ensuring that there is an infrastructure to support SPI [5][16]. It has also been suggested that having an SEPG in a parent organization supports the functioning of an SEPG in the organization being assessed (PROC4).

Variable ID	Question
ORG1	Are there tangible incentives or rewards for successful software process improvement?
ORG2	How much does “turf guarding” inhibit the progress of software process improvement?
ORG3	Is there much organizational politics?
ORG4	Does senior management actively monitor the progress of software process improvement?
ORG5	Is there a feeling among the technical staff that process improvement gets in the way of the real work?
ORG6	To what extent are process improvement goals clearly stated and well understood?
ORG7	How would you characterize the organization’s staff time / resources dedicated to process improvement since the appraisal?[1]
ORG8	Has there been turnover in key senior management?
ORG9	Has there been involvement of technical staff in the process improvement effort?
ORG10	Have the people who are involved in process improvement been respected for their technical and management knowledge, and their ability to get things done?
ORG11	Has there been clear, compensated assignment of responsibilities for process improvement?
ORG12	Has there been a major reorganization(s) or staff down-sizing?
ORG13	How much turnover has there been among middle management?
ORG14	How much turnover has there been among the technical staff?

Table 1: Organization questions. The response categories for these questions were “Substantial”, “Moderate”, “Some”, and “Little if Any”.

[1] The response categories for this question where: “Excellent”, “Good”, “Fair”, and “Poor”.

Variable ID	Question
PROC1	Did the organization that was assessed create an action plan for improving its software process based on the results of the assessment?
PROC2	Were Process Action Teams (PATs) or similar working groups established as a result of the assessment to address specific process improvements?
PROC3	Does the organization that was assessed have a software engineering process group (SEPG), or other unit(s) that performs similar functions?
PROC4	Does the parent organization of the organization that was assessed have a software engineering process group (SEPG), or other unit(s) that performs similar functions?

Table 2: Process questions. The response categories for all questions were "yes" and "no".

2.4 Resampling Strategy

Since there are multiple responses from each organization in most cases, we chose to follow an explicit resampling strategy. There are three primary reasons for resampling. First, the unit of analysis is the organization rather than the individual, so the analysis should be performed at the same unit about which we wish to draw conclusions. Second, in the current context of exploratory (non-inferential) analysis, pooling responses from different roles is inappropriate because doing so would give more weight to some organization's responses relative to the others, and could therefore bias the results in their favor. Third, response pooling artificially inflates the statistical power of inferential tests.

Since there were no apparent differences in response patterns among the three different roles, a simple resampling strategy would be to randomly select one response from each organization. A random sample constructed in this way would potentially have 56 observations, one from each organization represented in the sample of completed questionnaires. However, since the responses are similar across roles, a better strategy can be used. One problem with the simple strategy is that some respondents had missing data for some questions and not for others. So, instead of selecting a respondent from an organization at random, it is better to select a random respondent who does not have missing data. This is done for each question/variable that is included in the sample. If all the respondents have missing data, then anyone of them could be selected. For example, if for a particular organization there were three respondents for each one of the roles. Also, let's say on a given question one of the respondents did not answer or answered "Don't Know", and hence this is considered as missing data for the purpose of our current analysis. We therefore randomly select one of the responses of those two respondents whose responses were not missing data. Following this resampling approach ensured that the total number of observations remained at least 50.

One issue of concern here is whether the results of our analysis are a consequence of the particular sample selected. To test for this, we generated five different samples and performed the same analysis on each sample. The results are in fact stable and our conclusions do not change with the different samples. Hence we present the results for only one sample here.

One issue of concern, however, in using a resampling approach is its consequence on our data analysis method. We do not use inferential statistics to draw conclusions in the current study. The techniques we use are exploratory in nature and aim to identify patterns in the data. This is reasonable given that our aim is to generate hypotheses rather than to confirm a theory. Therefore, our analytical approach remains valid in conjunction with the resampling strategy we employ.

2.5 Data Analysis Methods

We employ two different data analysis techniques: principal components analysis and classification trees. Principal components analysis (PCA) [11] is used to identify a reduced set of organizational components relative to the original set of variables. Intuitively, it would seem that some of these variables are measuring the same construct, and therefore should be combined into one composite

dimension[2]. PCA provides us a systematic way for performing this reduction. Note that we are following an exploratory strategy here, as opposed to a confirmatory one since the association structure is not stipulated beforehand.

The algorithm that we used to construct classification trees was CART (Classification and Regression Trees) [1]. Constructing classification trees with CART requires that the dependent variable be dichotomized. We did so around the median value, differentiating between "low" success organizations and "high" success organizations. Classification tree algorithms have a number of analytical advantages. First, they do not require a detailed specification of the model to be tested beforehand, making them most suitable for exploratory analysis. This is most appropriate in the formative stages of research where detailed and testable theories do not yet exist. Second, the tree is a visually interpretable structure, making the results more accessible to non-specialists in data analysis. Third, the tree construction process takes into account potentially complex interactions among the independent variables. Fourth, it can easily accommodate categorical and continuous independent variables.

The tree construction process starts off by building a large tree, and then proceeds to prune it from the bottom up for simplification. The CART algorithm constructs binary trees. Tree-building involves the recursive partitioning of the data set. A splitting criterion is used to decide on which independent variable to split and where to make the split in the case of continuous independent variables. An exhaustive search for a "good" split is performed. For example, for a k value ordered categorical variable (which all our independent variables are), there are k-1 possible positions to make a split.

There are a number of different ways in which the "goodness" of a split can be judged. In practice, the literature suggests that not much difference exists between the commonly used splitting criteria in terms of the accuracy of the tree that is constructed [17][1].

The splitting criterion we use is the Gini measure of node impurity [1]. The Gini measure reaches a value of zero when only one dependent variable class is present at a node. This measure is computed as the sum of products of all pairs of class proportions for classes present at the node. It reaches its maximum value when class sizes at the node are equal. Tree construction stops when a minimal number of observations in a terminal node has been reached[3]. In the present analysis, we set this minimal number at 5 (representing approximately 10% of the respondents in each of the five resampled analyses). Prunning starts when all terminal nodes satisfy this criterion.

During pruning, trees are generated by removing splits upward until the tree with only a root node is formed. The optimal tree can be selected from this sequence of trees. An initial choice of an optimal tree is often the one that has the highest cross-validation accuracy (we use 3-fold cross validation). However, the accuracy estimate has some uncertainty associated with it. Therefore, the optimal tree is the smallest tree (with the smallest number of terminal nodes) that is within one standard error of the smallest misclassification rate[4] (this is also referred to as the SE rule). This approach results in a tradeoff between tree accuracy and tree complexity.

After a tree is constructed, it is judged to be good or bad. A commonly used criterion for evaluating a classification model is it's overall classification accuracy (e.g., see [12][21]). A number of different approaches can be used for estimating classification accuracy. The easiest method is to use the resubstitution estimate, whereby the accuracy of the tree is evaluated using the same data set that was used to construct it. This technique, however, is known to be an optimistic estimate of accuracy on unseen cases [24]. However, one can argue that it is a measure of the tree's "goodness of fit", in much the same manner as an R^2 value in an ordinary least squares regression model. For estimating accuracy on unseen cases, the approach that we use is a three-fold cross-validation[5]. Here, we split

---

[2] We do not perform a PCA on the process variables because, even though they may be correlated, they do not appear to measure the same construct.

[3] A cost complexity parameter for controlling tree growth was not used in our analysis since reduction of computation time was not a major concern.

[4] It is not necessary to use the one standard error value. Later in this paper we investigate the sensitivity of our model to this value by varying this parameter.

[5] We use a three-fold cross-validation as an extension of the common practice of using one third of a sample as a holdout sample for testing. Because our sample size was not very large, we refrain from using a holdout sample approach. The three-fold cross-validation also protects against fluctuations in using only one of the three subsamples for testing. The implementation of the CART algorithm (see http://www.statsoft.com) does not support leave-one-out cross-validation.

the sample into three disjoint subsamples. Two of these subsamples are used as a training set, with the third as the test set. Accuracy is calculated by classifying the observations in the test set. This is repeated by making each of the subsamples a test set, and the average accuracy across all test sets is used as the overall tree accuracy.

3 Results

3.1 Descriptive Summary of Respondents and Organizations

The survey respondents represent a variety of software organizations. The largest single proportion (37%) are from organizations from organizations that do contract work for the US federal government. Another 22% are from the federal government and US military services. Firms selling in the commercial market are the second largest category (36%) of software organizations represented by our respondents. Another 5% fall into the "other" category.

The organizations represented in our sample vary considerably in size. Approximately one-third of the survey respondents say they come from organizations that have 200 or more software employees. Another third come from organizations that employ 70 or fewer people who are primarily engaged in software.

Firms selling products in the commercial market are smaller than those in the military and federal government; 43% of the commercial organizations have 70 or fewer software employees as opposed to only 14% of the government organizations. The government contractors vary more in size; 40% have 200 or more software employees, while 34% have 70 or fewer.

The survey respondents were roughly evenly distributed among the roles that were sampled: 31% are SEPGers and other process champions; 34% each are software managers or senior technical people respectively. One person filled both the management and SEPG roles concurrently.

The respondents have a considerable amount of software experience. Half of them have worked on software for 16 years or more; a quarter of them have worked in the field for 22 or more years. All but the least experienced 10% of our respondents have worked on software for 10 years or more.

3.2 Dimensions of Organizational Factors

The results of the PCA are shown in Table 3. Note that for the results in this table all the variables within each factor were coded so that they are pointing in the same direction (i.e., higher score on each variable means a higher score on the factor).

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5
ORG1	**0.73**	-0.02	0.06	-0.02	0.03
ORG2	-0.19	0.06	**-0.86**	-0.08	-0.005
ORG3	-0.09	-0.16	**-0.81**	0.02	-0.14
ORG4	**0.74**	0.00	0.12	-0.24	0.18
ORG5	-0.07	0.07	0.46	0.11	**0.64**
ORG6	0.44	-0.18	0.04	0.26	**0.68**
ORG7	**0.63**	0.36	0.01	0.26	0.17
ORG8	0.14	0.20	0.02	-0.18	**0.75**
ORG9	0.35	0.15	0.40	0.38	0.29
ORG10	0.21	0.09	0.18	**0.77**	0.12
ORG11	**0.69**	0.34	0.33	0.37	-0.01
ORG12	-0.25	-0.45	0.21	0.53	-0.29
ORG13	-0.09	**-0.79**	0.25	0.12	-0.12
ORG14	0.00	**-0.87**	0.10	-0.12	0.05

Table 3: Results of the Principal Components Analysis with rotation for the "organization" type variables. Approximately 67% of the variation is explained by these five components. The coding scheme for all variables was 1 for the lowest values and 4 for the highest values.

The emergent factor structure is quite easily interpretable. We used a loading of 0.6 as a cutoff point. Before going into the details of each of the factors, it is worthwhile to note that variable ORG12 (the occurrence of major reorganization(s) or staff down-sizing) was removed from further analysis since it seemed to relate to a number of factors and its' interpretation was not obvious. Although the variable ORG9 did not load on one dominant factor, we retained this variable since a recent study identified it as an important determinant of SPI Success [23]. We termed this variable "INVOLVEMENT" since it captures the involvement of the technical staff in SPI. The remaining variables fall into one of the five components, and are interpreted below.

We used the emergent factors to construct composite variables. A composite was calculated as an unweighted sum of each of its component variables. This is a commonly used approach when working with subjective scales [22]. For each composite scale we calculate its Cronbach alpha coefficient [3], a measure commonly used to evaluate the reliability of subjective measurement scales [2]. The coefficient can vary from zero to one where one is perfect reliability and zero is maximum unreliability. Nunnally has suggested that for the early stages of research a Cronbach alpha coefficient approaching 0.7 are acceptable [18]. Given that we are at a formative stage in developing SPI theories, and as noted earlier our study was exploratory in nature, it would seem reasonable to use this as a general guideline to judge the reliability of the composite variables.[6]

3.2.1 Factor 1: COMMITMENT

The first factor is termed "Commitment". All the variables are concerned with the extent to which resources are made available for SPI and management's interest in SPI. These are considered as indicators of commitment to SPI. This four item measure of commitment had a Cronbach alpha coefficient of 0.718. The final composite variable was constructed to have higher values when commitment is high, and has a range from 4 to 16.

[6] It is well known that multi-item scales are more reliable than single item scales [25]. However, for some of the constructs that were measured in this study, only single item scales were used. This is the case because ours is a secondary analysis of data that was already existing, and therefore it was not possible to change the structure of the questionnaire. Also, note that the single item questions did not end up playing a role in the final set of results. This is perhaps due to their lower reliability.

3.2.2 Factor 2: TURNOVER

The two questions that make up this variable concern the turnover at the middle management and technical levels within the organization. Note that turnover in senior management (variable ORG8) does not load on the same factor, and seems to be measuring a different construct. The Cronbach alpha coefficient for this two item variable is 0.65. Even though this number is not high, it is actually quite good for a variable consisting of only two items. The final composite variable was constructed to have high values when turnover is high, and has a range from 2 to 8. Thus, for example, for turnover to have the maximum value, there would have to be "substantial" turnover in middle management and the technical staff.

3.2.3 Factor 3: POLITICS

The third factor is clearly measuring an underlying construct of "Politics". This is a general label for politically motivated activities and incentives that may promote or hinder SPI within an organization. The Cronbach alpha coefficient for this two item variable was 0.732. The composite variable was constructed to have high values the more politics, and has a range from 2 to 8.

3.2.4 Factor 4: RESPECT

The fourth factor consists of a single item, and has been labeled "RESPECT". This measures the extent to which individuals involved in SPI are respected within the organization. This variable was coded to have higher values the greater the respect, and has a range from 1 to 4.

3.2.5 Factor 5: FOCUS

The final factor measures the extent to which the organization is focused in its SPI efforts. Turnover in senior management detracts from this because new management often imposes new directions for the organization as a whole, with consequent effects on ongoing improvement initiatives. If staff feel that improvement initiatives (or the current one specifically) get in the way of the "real" work then the process improvement will be sacrificed when pressure builds up (e.g., deadlines). Finally, an organization cannot be focused in its SPI effort if its improvement goals are not clearly stated and understood. The Cronbach alpha coefficient for this variable was 0.62. The composite scale was coded so that higher values indicate greater focus, and has a range from 3 to 12.

3.3 Dependent Variable

We used a single survey question as the dependent variable for our analysis: "How successfully have the findings and recommendations of the assessment been addressed?" We dichotomized the answers at the median value of "moderate" success. Figure 2 shows the overall distribution of the dependent variable. The modal response is "limited" Success. Values at "moderate" or below are classified in the "Low" category; responses of "substantial" or "marked throughout the organization" are classified in the "high" success category.

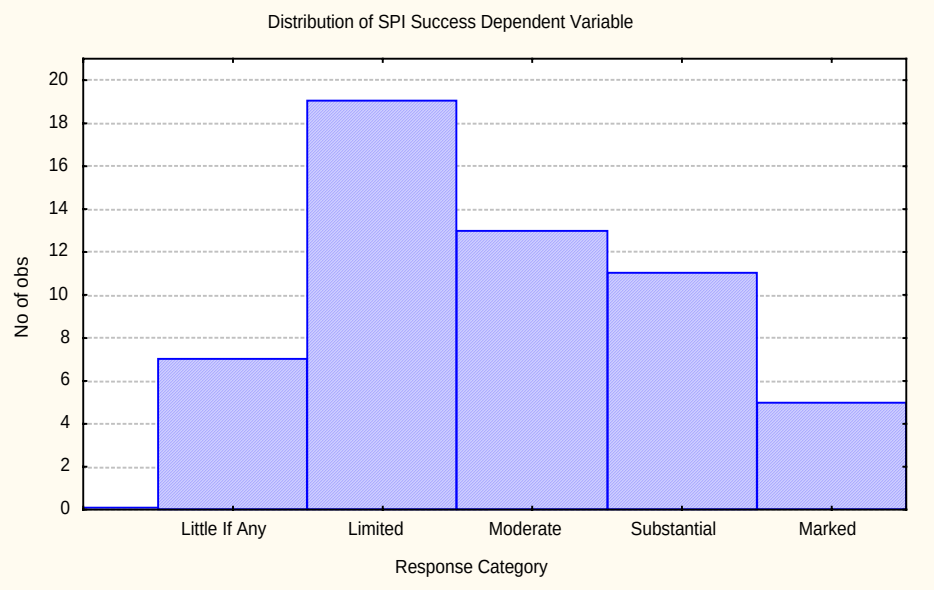

Figure 2: Distribution of the dependent variable "SPI Success".

3.4 Classification Tree

The final tree is shown in Figure 3. We first present the accuracy results, and then interpret the tree. Note that due to missing data, only 50 observations were used in constructing the tree[7].

The goodness of fit criterion (i.e., the resubstitution estimate of accuracy) for this tree was 92%, indicating a very good fit to the data. There are two ways in which we can calculate the cross-validation estimate of accuracy. As noted earlier, CART constructs many trees and then selects the best one. For selection of the best tree, the cross-validation estimate for each tree is used. This estimate for our tree is 84% accuracy. This too is quite high, and is a marked improvement over the chance probability of getting 50% correct classifications. The second approach is a "global" cross-validation whereby the entire analysis is replicated three times, each time with one third of the data set used as a test set. However, during each of these replications, these two-thirds of the data set are again split into three samples, and now only approximately 44% of the whole sample is used for constructing each of the trees that CART chooses from. With such small numbers, it is hardly surprising that not many trees are actually constructed and therefore the tree with the root node predominates. The tree with only a root node has an accuracy of 76%, which matches the base rate of 76% of organizations having "Low" SPI Success. Hence this second approach does not provide a reasonable estimate of the accuracy of the tree on unseen cases.

[7] We employed casewise deletion when there were missing data.

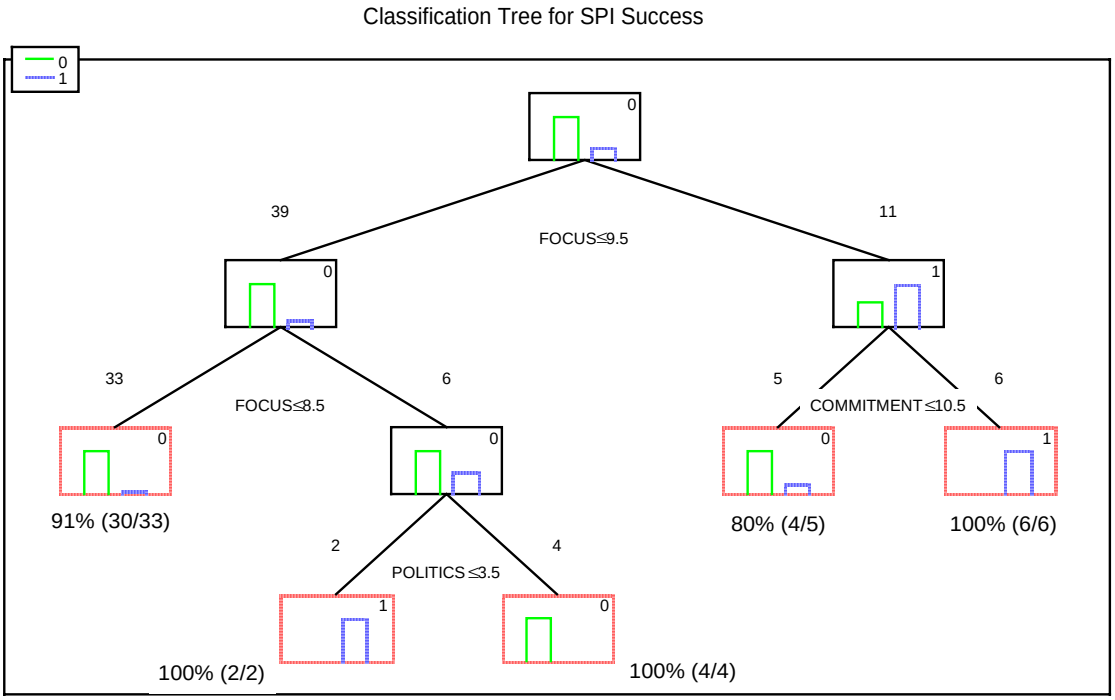

Figure 3: The CART generated classification tree. The number within each node is the predicted class (0 is LOW, and 1 is HIGH). The predicted class is the most frequent one within the node. The numbers on the edges are the number of observations. If the condition at a nonterminal node is true, then take the path on the left. The percentages next to each terminal node are the resubstitution estimates of accuracy for that particular terminal node.

All three composite variables do affect the probability that an organization will be successful in implementing process improvements based on the results of its assessment. However, the extent of such success depends on the interaction among the variables. Their effect is not interchangeable in an additive sense. High commitment and process focus pay off only when they are combined together. Organizational politics only are important when process focus is moderately high.

When process focus is low then an organization is likely to have little success in implementing process improvements based on the results of its assessment, regardless of the degree of organizational commitment or politics. Process improvement efforts are in fact more likely to be successful when there is a moderate amount of process focus, if the organization is not dominated by politics and turf guarding; otherwise the improvement effort will have little success. Implementation of process improvements is most likely to be successful if both process focus and organizational commitment are high. High amounts of process focus will not result in commensurate success when commitment remains low[8].

The tree highlights the importance of focus and commitment as the major determinants of SPI success. Also interesting is that none of the other variables were selected in the model, indicating that their relative influence in explaining success is small, at least for our current samples.

[8] To check for consistency across classification tree algorithms, we also constructed a tree using Quinlan's C4.5 [20]. C4.5 uses an entropy based measure to decide on which variable to make the split, and an approach using confidence intervals around the estimated misclassification rate to prune the tree. The final tree that we obtained was very similar to the one shown in Figure 4 with some minor exceptions. First, the splits on the continuous variables were different. However, this is to be expected as C4.5 uses an entropy based criterion to discretize the continuous variables, which is different from the one used in CART. Also, the branch with the POLITICS variable was not taken in the C4.5 tree. This is an indication that the entropy based splitting criterion used in C4.5 does not identify a gain in further partitioning the "medium FOCUS" node. However, the C4.5 tree had a 71% accuracy using the 3-fold cross validation estimate, which is less than the CART accuracy. We also found the C4.5 tree to be sensitive to settings of some of its parameters. By changing the minimal number of observations in a node, we obtained a tree that uses the variable PROC3 instead of COMMITMENT. This is consistent, however, as one can argue that the existence of an SEPG is a manifestation of an organization's commitment.

3.5 Sensitivity of the Tree

We investigated the sensitivity of the tree to two CART parameters: the SE rule and the minimal number of observations in a node. While some sensitivity to these parameters would be expected, the question is whether the impact is sufficient to question the stability of the results presented here.

The common default value for the SE rule that is used in selecting the optimal tree is one. Using this value means that the smallest tree within the standard error of the smallest misclassification rate. We varied this from 0.1 to 1.5 in 0.1 increments, and the same tree in Figure 3 was selected each time. This provides reassurance that this is the optimal tree for a wide variation in this parameter.

We also varied the minimal number of observations in a node for a split to occur. Increases in the minimal value result in a tree similar to the one in Figure 3 but that is pruned from the bottom. The reverse effect occurs when the minimal value is decreased. In both cases the cross-validation accuracy decreases, indicating that the tree in Figure 3 has the best accuracy for minimal node values around 5.

3.6 Variable Importance

During the tree growing process, surrogate splits are also considered. These are other splits that mimic the action of the primary splits that were actually chosen. This is evaluated using a measure of association between the alternative split and the primary split [1]. An association value of 1 indicates that the alternative split can predict perfectly the primary split.

For each variable when it appears as a surrogate, the improvements in the Gini index had that variable been selected for the primary split are summed up for all nodes. These summed improvements are scaled relative to the best performing variable such that the highest value is 100. This value is a measure of a variable's importance [1]. The importance score measures a variable's ability to mimic the chosen tree and to play a role as a stand-in for variables appearing in the primary splits.

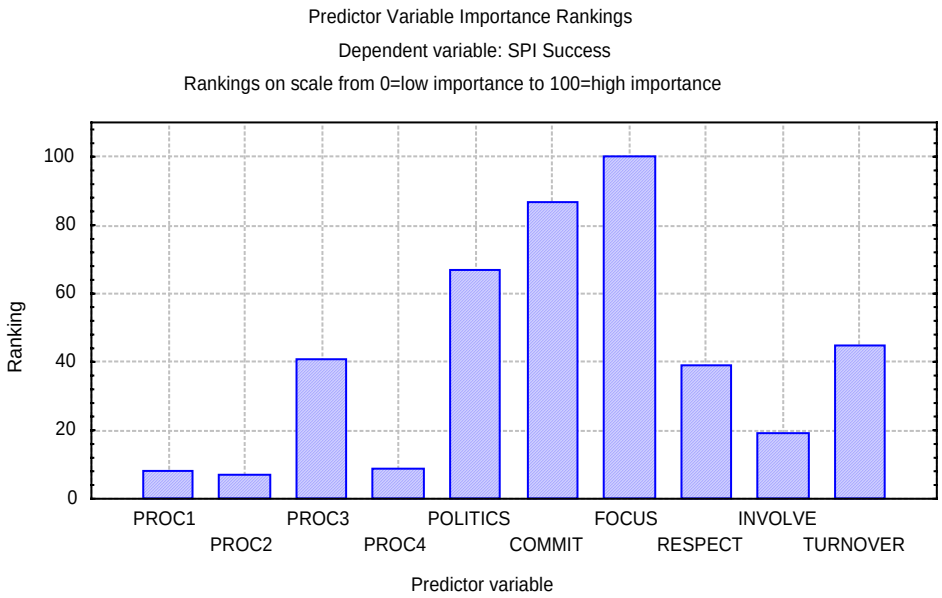

Figure 4: Variable importance for each of our independent variables.

As expected, the variables POLITICS, COMMITMENT, and FOCUS are the most important. However, the most interesting information in Figure 4 is that variables PROC1 (production of an action plan) , PROC2 (establishment of PATs), PROC4 (parent organization having an SEPG), and INVOLVEMENT have little relative importance in the context of the tree shown in Figure 3. This can be interpreted to mean that in the context of our tree, these four variables do not add much to explain SPI Success. Two points should be made about this assertion. First, this does not mean that these factors are not important for SPI, only that when you consider FOCUS, COMMITMENT, and POLITICS, they have *relatively* much less importance. Second, this assertion is limited to our current data set.

4 Conclusions and Next Steps

In this paper we have presented a model that explains the success of SPI efforts based on data on CMM based improvement. We found the model to have a good fit to our data, to have good estimated accuracy on unseen cases, and to remain stable to fluctuations in the parameters of the tree building algorithm that we have used. The model adds to the few multivariate, empirically grounded analyses of the conditions under which process improvement is likely to succeed or fail.

Of course all conclusions from a single study are tentative at best, and require confirmation through further research. More investigation is necessary if we are to provide strongly justifiable advice to organizations on how best to conduct their improvement efforts.

We will continue with further multivariate analyses on the current data set. Our initial focus is on identifying role differences among the managers, senior developers, and process champions who make up the overall sample. While role differences do not appear to affect the zero and first order analyses or our basic conclusions, we do have some tentative results from cluster analyses that may show common patterns of agreement by role over groups of similar variables.

In addition, we intend to do further multivariate comparisons of the impact on organizational performance (e.g., product quality, meeting schedule and budget targets, customer satisfaction) of both process improvement and organizational context (e.g., size, sector, and domain). And we will use log-linear and other modeling techniques to supplement and validate our current exploratory approach using principal components analysis and classification trees. Log-linear models are particularly useful for addressing interactions of categorical variables. While we have no current plans to collect new data in this area, we may do some additional comparisons using similar data sets from the ongoing international SPICE trials and other earlier work [23][4].

5 Acknowledgments

First of all, we once again thank the respondents to our survey, along with those who helped construct the sample. We are grateful to our many colleagues who helped with the original study. Particular thanks are due to David White and Michael Zuccher, without whom the data never would have been available for our analyses.

6 References

[1] L. Breiman, J. Friedman, R. Olshen, and C. Stone: *Classification and Regression Trees*. Wadsworth, 1984.

[2] E. Carmines and R. Zeller: *Reliability and Validity Assessment*. Sage Publications, 1979.

[3] L. Cronbach: “Coefficient Alpha and the Internal Structure of Tests”. In *Psychomterika*, 16(3):297-334, 1951.

[4] C. Deephouse, T. Mukhopadhyay, D. Goldenson, and M. Kellner: “Software Processes and Project Performance,” *Journal of Management Information Systems*, Vol. 12, No. 3 (Winter 1995-96), 187-205.

[5] P. Fowler and S. Rifkin: *Software Engineering Process Group Guide*. Software Engineering Institute Technical Report CMU/SEI-90-TR-24, 1990.

[6] D. Goldenson and J. Herbsleb: *After the Appraisal: A Systematic Survey of Process Improvement, its Benefits, and Factors that Influence Success*. Software Engineering Institute Technical Report CMU/SEI-95-TR-009, 1995.

[7] J. Herbsleb, D. Zubrow, D. Goldenson, W. Hayes, and M. Paulk: “Software Quality and the Capability Maturity Model,” *Communications of the ACM*, Vol. 40, No. 6 (June 1997), 30-40.

[8] J. Herbsleb, A. Carleton, J. Rozum, J. Siegel, and D. Zubrow: *Benefits of CMM-based Software Process Improvement: Initial Results*, Software Engineering Institute Technical Report CMU/SEI-94-TR-13, 1994.

[9] J. Herbsleb and D. Goldenson: “A Systematic Survey of CMM Experience and Results.” In *Proceedings of ICSE ’96*. March 1996, 25-30.

[10] W. Humphrey and W. Sweet: *A Method for Assessing the Software Engineering of Contractors*. Software Engineering Institute Technical Report CMU/SEI-87-TR-0023, 1987.

[11] J. Kim and C. Mueller: *Factor Analysis: Statistical Methods and Practical Issues*. Sage Publications, 1978.

[12] F. Lanubile and G. Visaggio: “Evaluating Predictive Quality Models Derived from Software Measures: Lessons Learned”. In *Journal of Systems and Software*, 38:225-234, 1997. Also appears as International Software Engineering Research Network, technical report ISERN-96-03, 1996.
[13] P. Lawlis, R. Flowe, and J. Thordahl: “A Correlational Study of the CMM and Software Development Performance,” CrossTalk, September 1995, 21-25.
[14] J. Maher and J. Gremba: “Organizational Barriers to SPI and Technology Transition”. In *Proceedings of the 1994 Software Engineering Symposium*, 1994.
[15] M. Miller and D. Goldenson: *Software Engineering Process Groups: Results of the 1992 SEPG Workshop and a First Report on SEPG Status*. Software Engineering Institute Special Report CMU/SEI-92-SR-13, 1992.
[16] B. McFeeley: *IDEAL: A User’s Guide for Software Process Improvement*. Software Engineering Institute, Handbook CMU/SEI-96-HB-001, 1996.
[17] J. Mingers: “An Empirical Comparison of Selection Measures for Decision-Tree Induction”. In *Machine Learning*, 3:319-342, 1989.
[18] J. Nunnally: *Psychometric Theory*, McGraw-Hill, 1978.
[19] J. Puffer: “Action Planning”. In *IEEE TCSE Software Process Newsletter*, No. 9, Spring 1997 (available from http://www.iese.fhg.de/SPN/process/spn.html).
[20] J. Quinlan: *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
[21] N. Schneidewind: “Validating Metrics for Ensuring Space Shuttle Flight Software Quality”. In *IEEE Computer*, pages 50-57, August 1994.
[22] P. Spector: *Summated Rating Scale Construction*. Sage Publications, 1992.
[23] The SPICE Project: *Phase 2 Interim Trials Report*. Project Report, March 1998.
[24] S. Weiss and C. Kulikowski: *Computer Systems that Learn*. Morgan Kaufmann, 1991.
[25] R. Zeller and E. Carmines: *Measurement in the Social Sciences*. Cambridge University Press, 1980