AMCIS 2007 Proceedings

Americas Conference on Information Systems (AMCIS)

December 2007

Estimating Risk in Information Technology Projects

Andrew Gemino
University of Oxford

Christopher Sauer
Simon Fraser University

Blaize Reich
The University of Tampa

Estimating Risk in Information Technology Projects

Andrew Gemino
Simon Fraser University
gemino@sfu.ca

Chris Sauer
Saïd Business School,
University of Oxford
chris.sauer@sbs.ox.ac.uk

Blaize Horner Reich
Simon Fraser University
breich@sfu.ca

Abstract

Few studies have focused on estimating the relative effects of risk factors on IT project performance This paper discusses three predictive modeling techniques, regression, neural networks and classification trees, that could be used for such an estimate. A comparison suggests that classification tree techniques may provide more effective estimates than other techniques. A data set (n=227) with 9 risk factors is used to develop a classification tree estimate. Results from this analysis provide a classification tree with 8 nodes that demonstrates the ease of interpretation of the classification tree results. The results are relatively easy to understand and provide actionable information. While results are only preliminary, we conclude that a risk assessment tool based on classification tree techniques could provide a new and effective tool in the management of IT project risk.

Keywords

Project Management, IT Project Risk, Predictive Modeling, Classification Tree Analysis

Introduction

A fundamental belief underlying much of the literature in systems analysis and design is that when requirements are not well defined an information technology (IT) project faces an increased risk of poor performance. While this assumption is held widely, little research has been done to estimate the relative effects of categories of risk on IT project performance (Zhang et. al., 2003; Tiwana and Keil, 2004; Wallace and Keil, 2004). Practitioners (Standish Group, 2003) and researchers (Barki et. al, 2001; Keil et. al. 1998; Nidumolu, 1995; Ropponen and Lyytinen, 1997; Wallace et. al, 2004(a)) have recognized that requirements uncertainty and other risks are important variables in considering IT project performance. Efforts have been made in understanding how to appropriately respond to requirements uncertainty and other risks (Barki et. al., 2001; Lyytinen et. al., 1998; Nidumolu, 1995; Ropponen and Lyytinen, 2000; Wallace et. al., 2004(b)). Research attention has now begun to focus on providing more effective tools to assess IT project risks (Tiwana and Keil, 2004; Zhang, et. al, 2004).

This paper discusses techniques for improving the assessment of IT project risks. Specifically, the focus of this paper is on the ability to estimate the impact of IT project risks on performance. Three generic estimation techniques for risk assessment are contrasted: regression, neural network and classification tree analysis. This comparison suggests that for risk assessment, classification trees may provide more effective estimates. A classification tree estimate is then developed using a sample data set of over 200 projects. Results from the analysis suggest that a classification tree can provide valuable and actionable information to address IT project risks. Based on this analysis, we conclude that a risk assessment tool based in classification tree technique could provide a new and effective tool in the management of IT project risk.

The paper is organized as follows. The next section provides background on previous research in risk management and some of the difficulties practitioners face in applying risk management techniques. A general discussion of three alternative estimation techniques is then provided. The techniques are then compared using several dimensions. This is followed by results from a classification tree analysis that estimates the influence of nine recognized risk factors on a binary dependent variable of IT project performance. An analysis of the results and conclusions from the analysis are discussed in the final section. .

Background

Research in risk management in IT projects has focused on early stages of risk identification and analysis (Keil et. al, 1998; Ropponen and Lyytinen,2000; Wallace and Keil, 2004). The focus on early stages of risk management is appropriate as more general theoretical insights are possible in the initiation and analysis phases of these projects. Once projects are past initiation and early planning, the management of risk naturally moves towards tracking, mitigation and response which requires deep organizational and project context to address. Current project management practices can provide effective tools for tracking, mitigating and responding to risk through project execution. What can be more difficult, however, is the early assessment of risk and estimating the impact that these risks might have on overall project performance.

Early risk assessment can be difficult for practitioners (Tiwana and Keil, 2004). For example, most risk assessments suggest the following four step technique: 1) identify risks (often with the help of a checklist); 2) estimate the likely impact of the risk on scope, budget and schedule (normally a subjective assessment in dollars); 3) estimate the likelihood of the risk occurring (normally a subjective probability estimate) and 4) create an overall risk assessment by multiplying impact by likelihood of occurrence and prioritize risks by sorting from highest score to lowest score. While step one and four are somewhat straightforward to develop, often the real value of the process remains in steps two and three. These steps require forecasts of both impact and likelihood of occurrence. These estimates are difficult to develop for several reasons. The data for these estimates is often not readily available, and even when available, the uncertainty that accompanies unique projects encourages estimates with large variances. In addition, the relationships between risks and project performance are arguably complex and nonlinear. The impact of risk is also likely to be contingent on the existence of other risks, and hence subject to interaction effects. These influences suggest the early estimation of risk impact and likelihood is far from a simple matter.

The lack of supporting data for these forecasts likely encourages assessments that reflect previous experience rather than actual risk. The risk assessments might therefore become a way of perpetuating risk assessments that either underestimate or overestimate risks in IT projects. There is also the additional relationship between risk assessment and project performance to consider. At what point do the risks drive a project to under-perform? Do projects under-perform because of inaccurate risk assessments that support “unattainable” project objectives, or is underperformance the result of unknown or poorly managed risks? These questions indicate that it is not only important to estimate the impact and likelihood of risks, but also to understand how the overall risk might be altered by addressing the risks that have been identified. In other words, it is not good enough to indicate a project is high risk; it is also important to identify how these risks might be altered to make successful outcomes more likely.

Techniques for Forecasting Risk Impacts

There are several classes of estimation techniques which can be used to estimate the impact of risk on project performance. The most common of these is regression and tools have been developed using regression techniques to assess the impact of project risk (e.g. Tiwana and Keil, 2004). Each estimation technique has advantages and disadvantages and it is important, in choosing an estimation model, to balance the nature of the data set, the type of dependent variable being estimated, and the utility of the results from the analysis. Our analysis suggests that regression may not provide the best method for assessing risks in projects. For this reason, we introduce two less common estimating alternatives – neural networks and classification tree analysis.

Regression

The use of regression techniques by Tiwana and Keil (2004) is an example of a tool focused on making risk assessment easier for practitioners. The tool provides a simple mechanism for estimating overall project risk based on relative weightings of six risks found to be important. The results suggest the quick assessment tool may be quite effective in identifying high risk IT projects.

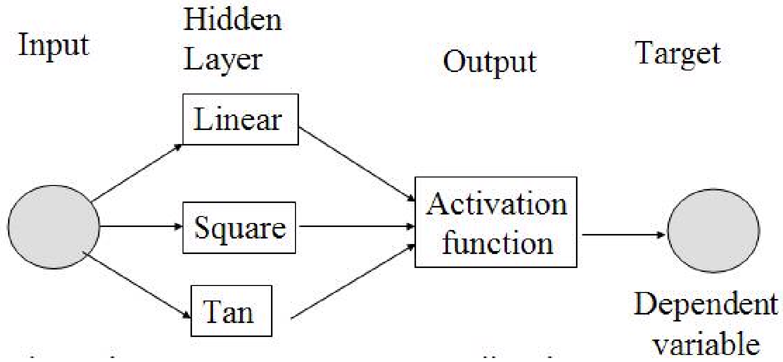
There are several reasons that regression is chosen for estimation: 1) its proven ability to provide robust estimates, 2) it provides minimum variance linear estimates, 3) the wide availability of statistical software and training for regression analysis, and 4) its ability to use polynomials (X^2, X^3, X^4 …) to approximate nonlinear relationships. However, there are limitations to how effective regressions are at estimating step functions and other discontinuous variables. Regression also requires full information (a case must include all dependent or independent data) which can be difficult to develop. Finally, managers often have difficulty with interpreting and choosing actions based on regression output.

Neural networks

Neural networks have been used to estimate the likelihood of IT project escalation (Zhang et. al., 2003). These networks develop purely data driven models through training. Unlike regression, neural networks do not depend on assumptions about functional form, probability distribution or smoothness. A neural network combines many simple computing elements (perceptrons) into a highly interconnected system to estimate a phenomenon (amore thorough analysis is provided in Bishop,1995). A perceptron is the mechanism used by the network to combine, or weight, several input variables into a predicted value for a dependent (target) variable. The set of weightings that creates this predicted value is called an activation function. The neural network is developed by entering input variables, one case at a time, and creating predicted values using the current activation function. The predicted values are compared with actual values of the dependent variable. When the predicted values are not close to actual values, the weightings of the various input variables are altered. The "training" method attempts to minimize the error in prediction by adjusting the weights associated with input variables in the activation function. The network continues refining until the changes in the activation function become small. At this point the neural network has stabilized and the activation function is ready for estimation.

Complex estimates in neural networks are created through hidden layers. Multilayer perceptrons use several hidden layers. Hidden layers represent transformations (usually non-linear) of the original input variables. These layers provide additional mechanisms for weighting resulting in a more complex, and perhaps more predictive, activation function. A basic visual of a multilayer perceptron is shown in figure 1 below. Three hidden layers are shown in the diagram, each using a different linear technique for weighting the input variable. The use of non linear functions enables the neural network to be extremely flexible in developing estimates for complex phenomena.

Figure 1: Multilayer Perceptron



The important elements of a neural network are that the activation function is developed and updated as a result of the learning that takes place from an initial data set. Once the activation function is found to perform well, new data values can be run through the established activation function to estimate the effect that the input variables will have on the dependent variable. It is important to note that the activation function is a collection of a complex set of weightings for each input variable. The activation function is often considered a "black box" for estimation. It is difficult to discern how the neural network develops the eventual estimate for the dependent variable. So while neural networks can provide accurate estimates for complex functions, it is difficult to provide an explanation for how the estimate itself is created. It is also difficult to estimate how much impact each input variable has on the dependent variable.

Classification Tree Analysis

In their survey of business application of data mining, Apte et al. (2002) identify risk management as a task well-suited to classification tree analysis. A classification tree is a segmentation of data that is created by applying a series of rules. The tree is constructed through a process of recursive partitioning (Breiman et al., 1984). The process begins with a target (dependent) variable that is to be estimated by several input variables. The input variables are first evaluated to find which of the variables provides the most "worthwhile" split in the data. There are many options for deciding splitting rules for the tree. Each rule assigns an observation to a segment based on the value of one input variable. One rule is applied, then another and another resulting in a hierarchy of segments. The hierarchy that is created is called a tree and each segment within the tree is a node. The final nodes are called "leaves". The tree is "trained" to make choices that maximize the contribution to the tree. The contribution is measured by worth which is defined as:

$\text{Worth} = I(\text{node}) - \Sigma P(b)*I(b)$

where

$I(\text{node})$ = entropy, Gini, or variance in node (for example $I(\text{node}) = \Sigma (Yi - Y)^2$)

$P(b)$ = proportion of observations in node b

$I(b)$ = entropy, Gini, or variance in node b

Criterion may be based on statistical significance, reduction in variance, or entropy. Once the "best" split is determined the first "split" of the data is implemented. This creates a new splitting problem, now with two data sets. The splitting process continues until no more useful splits can be developed and a stopping point is reached. At this time the splits in the data can be analyzed. The accuracy of estimation is provided for each node based on each nodes probability of accurately predicting the dependent variable

Once the tree has been developed, estimates for the target variable are created relatively easily. A tree analysis produces a model that may be represented in rules (if/then) logic statements. The results of these splitting rules provide the basis for estimating the dependent variable. The person requiring the estimate need only to find the node for which the currently unclassified example satisfies the conditions. Unlike the complex activation function in neural networks, or the sometimes confusing coefficients and t-stats in regression, the classification provides output in a tree form that is easy to read and understand. Results are quickly transparent to the reader and the tree also indicates actionable paths that might lead to lower risk. An example of classification tree output is provided in Figure 2. One of the primary advantages of classification trees is their ease of understanding associated with output from the technique.

Differences Between Estimation Approaches

The objective of a risk assessment/estimation is to establish IT projects that contain more risk, to understand what elements of risk are most important and to provide some feedback on what can be done to reduce the risk. When considering the three different estimating procedures, it should be recognized that there are two major categories of differences to consider. One category considers the technical advantages/disadvantages associated with each technique. These considerations include the ability to estimate nonlinear relationships and the impact of missing data points. The second category is the managerial advantages/disadvantages associated with each estimation technique. As noted above, accuracy of risk assessment is only one important managerial element. Knowledge of what might be causing the risk and how to act to reduce the risk are also important. The differences between the techniques are summarized in Table 1 below.

Table 1: Summarizing Differences Between Estimation Approaches

Category	Estimation Model		
	Regression	**Neural Network**	**Classification Tree**
Ability to Handle Complex Non Linear Relationships	**Weak** Polynomials provide complexity but are limited	**Good** Hidden layers provide complexity	**Good** Larger tree provides complexity
Ability to Handle Missing Data Points	**Poor** Must have full data	**Poor** Must have full data	**Good** Does not require full data
Accuracy of Estimation	**Best** for linear models	**Good** as long as training set is large	**Good** as long as training set is large
Ability to Explain Results	**Medium** Assumes independence among explanatory variables	**Poor** Activation function not simple to understand	**Good** Tree diagram relatively simple to understand.
Ability to Develop Actions to Reduce Risk	**Medium** Provides regression estimates and statistical significance	**Poor** Activation function not simple to understand	**Good** Tree diagram relatively simple to understand.

Our assumptions at the beginning of this comparison were that the relationships between risks and project performance are likely complex and non linear and that the important objectives beyond estimation accuracy are to provide results that are easy to understand and take action. Given these criteria, an analysis of Table 1 suggests that classification trees provide the best potential for estimation models that provide effective information for risk assessment and management. In the following section we develop a risk assessment using a classification tree technique to demonstrate their potential effectiveness.

Method

The data set was provided by 227 participants from registered readers of Computer Weekly, a popular UK-based, weekly newspaper for IT professionals. An initial email and follow-up request to participate in the study were sent to readers registered as project managers. Participants were asked to consider only their most recently completed (or abandoned) project in filling out the survey .Each survey point therefore represents the outcomes of a single project. The survey data was collected using web-based. forms. On average the participants reported high levels of experience with 17 years in the IT industry and 9 years as a project manager.

Variable Definitions

A dichotomous dependent variable (performing/ underperforming) was created using a cluster analysis of three performance measures: budget variance, schedule variance and scope variance. Five clusters were initially identified as noted in Sauer et .al. (2007). Of these clusters, two of them provided variances at or near original objectives on all three performance measures (budget, schedule and scope). This represented "Performing" projects, representing 64% of the sample. The remaining clusters showed project variances that were significantly above original estimates on one or more dimensions. These projects were labeled Underperforming".

Nine independent variables were used as input for the classification tree analysis. A summary of these variables is provided in Table 2. These nine variables represented a collection of risk factors, including size, requirements uncertainty, technical complexity, and experience). The first five of these factors measured project size. Project size has been noted as an indicator of project risk (McFarlan, 1982). Four of these factors measured actual project size, budget, effort (measured as person months), elapsed time (in months) and team size (effort divided by elapsed time). Each of these factors was collected using a continuous variable. For example, budget was measures in actual Pounds Sterling, and elapsed time was measured by the actual number of months. These continuous variables were then categorized to make the outcome of the analysis easier to interpret. The values of each category are provided in Table 2. A fifth factor measured the relative size of the project in relation to other projects in the organization. A 7 point Lickert type scale was used to measure this.

Several studies have indicated the importance of both requirements uncertainty and technical complexity (Barki et. al, 2001; Nidumolu, 1995; Ropponen and Lyytinen, 1997). The uncertainty of requirements and the technical complexity of the projects were estimated using 7 point Lickert scales. Values were assigned as shown in Table 2.

The level of experience of the project manager (PM) is also considered an important risk assessment factor. For example, the Standish Group lists PM experience it as the most important factor (Standish Group, 2003). The years of project management experience as well as the years of corporate experience of the respondent were collected and categorized as described in Table 2.

Table: 2: Summary of Independent Variables

		Values for Independent Variables								Std.
Risk	**n**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Mean**	**Dev.**
Budget Category	202	<100 K	1-500 K	0.5 - 1M	1- 5 M	5-10 M	>10 M		2.74	1.39
Person Months Category	202	< 12	12- 24	24 - 48	36-48	48- 96	> 96		2.73	1.77
Elapsed Time Category	222	< 3 month	3- 6	6- 12	12- 18	18- 24	>24		3.44	1.58
Team Size Category	208	< 5	5 - 20	> 20					1.43	0.73
Relative Size	227	Very Small	Small	Slightly Smaller	Average	Slightly Larger	Large	Very Large	3.22	1.00
Requirements Definition	227	Very Certain	Certain	Somewhat Certain	Average	Somewhat Uncertain	Uncertain	Very Uncertain	2.84	1.23
Technical Complexity	227	Very Low Complex	Low Complex	Slightly Complex	Average	Complex	Highly Complex	Very Complex	3.82	1.33
PM Years Experience	223	<5 yrs	5 - 15	> 15 yrs					2.35	0.76
Corporate Experience	223	<5 yrs	5 - 15	> 15 yrs					2.09	0.73

Splitting Algorithm

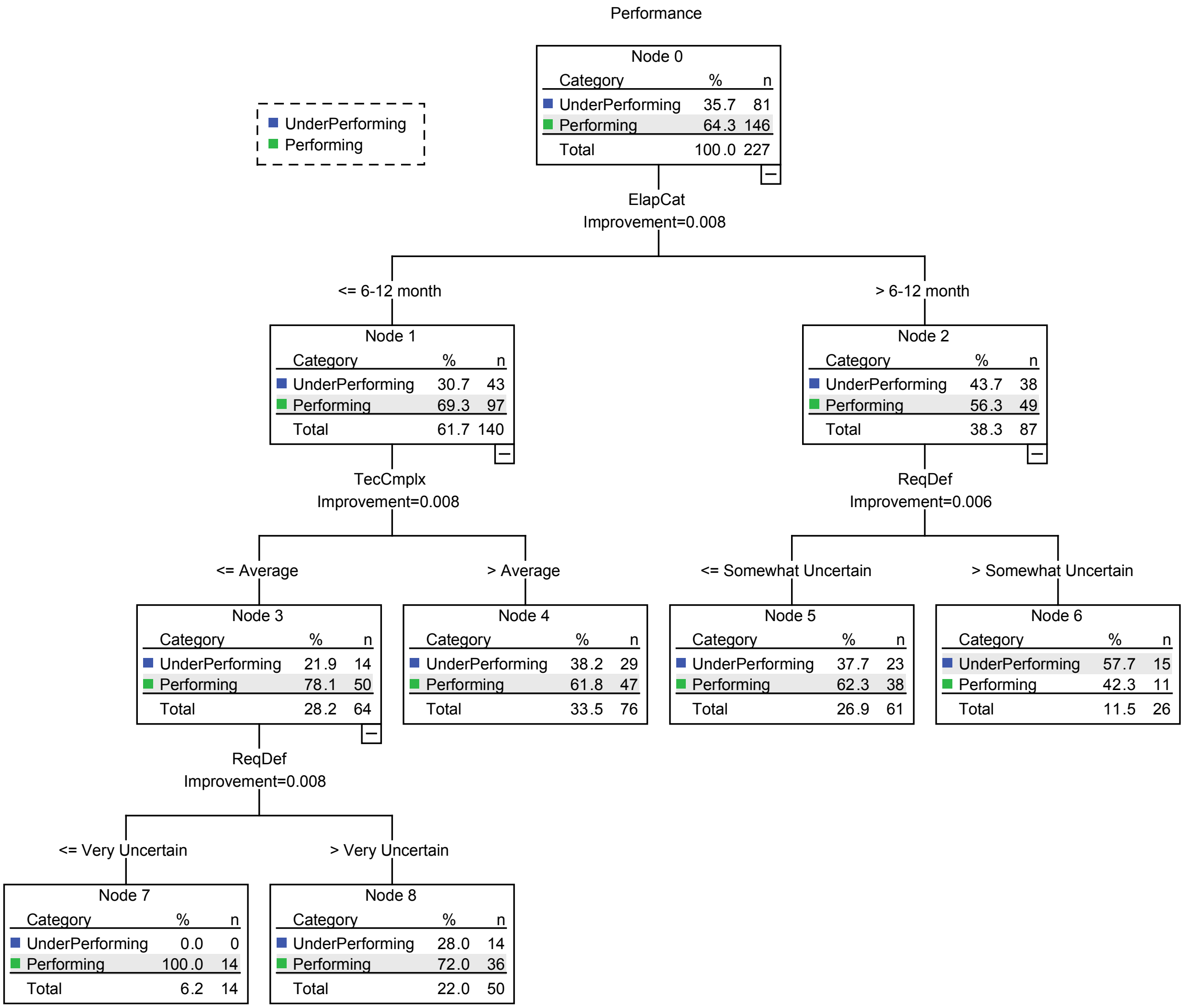
The TREE algorithm in SPSS version 15.0 was used to develop the estimate. The Classification and Regression Tree (CRT) growing method was used for the analysis. The CRT was preferred due to the binary nature of the dependent variable which eliminates the use of CHAID (Chi-squared Automatic Interaction Detection). The CRT splits the data into segments that are as homogeneous as possible with respect to the dependent variable. A node is homogenous when all cases have the same value for the dependent variable. An example of this would be "Node 7" in Figure 2.

Classification trees are data driven methods that require training and hence large data sets. The data set provided by the sample is relatively small. This has two important impacts. First, we have not created a holdback sample that would be used for testing. The data set was simply too small to partition. This means that the results from this analysis should be taken only as preliminary estimates. The second impact is that the number of cases in parent nodes was limited to 60 cases, and in child nodes 12 cases. These numbers are reasonable given the original data set. A larger data set will allow for larger numbers for both child and parent nodes and hence more certainty in regards to the stability of the model.

Results

The results for the CRT splitting algorithm are displayed in Figure 2 below. The entire data set (n = 227) is included in Node 0. Performing projects make up 64% of the data set. The first split is created on the Elapsed Time Category where projects that are less than or equal to 12 months in length are separated from projects greater than 12 months in length. Note the differences in estimated performance of the projects that are less than or equal to 12 months (Node 1, 69%) versus projects greater than 12months (Node 2, 56%). The results in Node 7 suggest that 100% of projects that are less than 12 months long, that are below average in technical complexity and are considered at least "Uncertain" (but not very uncertain) are classified as performing. While this result is not surprising, it does provide prima facie support for this type of estimating technique.

Figure 2: Results from the CRT Splitting Algorithm



Node 0

Category	%	n
🟦 UnderPerforming	35.7	81
🟩 Performing	64.3	146
Total	100.0	227

Node 1

Category	%	n
🟦 UnderPerforming	30.7	43
🟩 Performing	69.3	97
Total	61.7	140

Node 2

Category	%	n
🟦 UnderPerforming	43.7	38
🟩 Performing	56.3	49
Total	38.3	87

Node 3

Category	%	n
🟦 UnderPerforming	21.9	14
🟩 Performing	78.1	50
Total	28.2	64

Node 4

Category	%	n
🟦 UnderPerforming	38.2	29
🟩 Performing	61.8	47
Total	33.5	76

Node 5

Category	%	n
🟦 UnderPerforming	37.7	23
🟩 Performing	62.3	38
Total	26.9	61

Node 6

Category	%	n
🟦 UnderPerforming	57.7	15
🟩 Performing	42.3	11
Total	11.5	26

Node 7

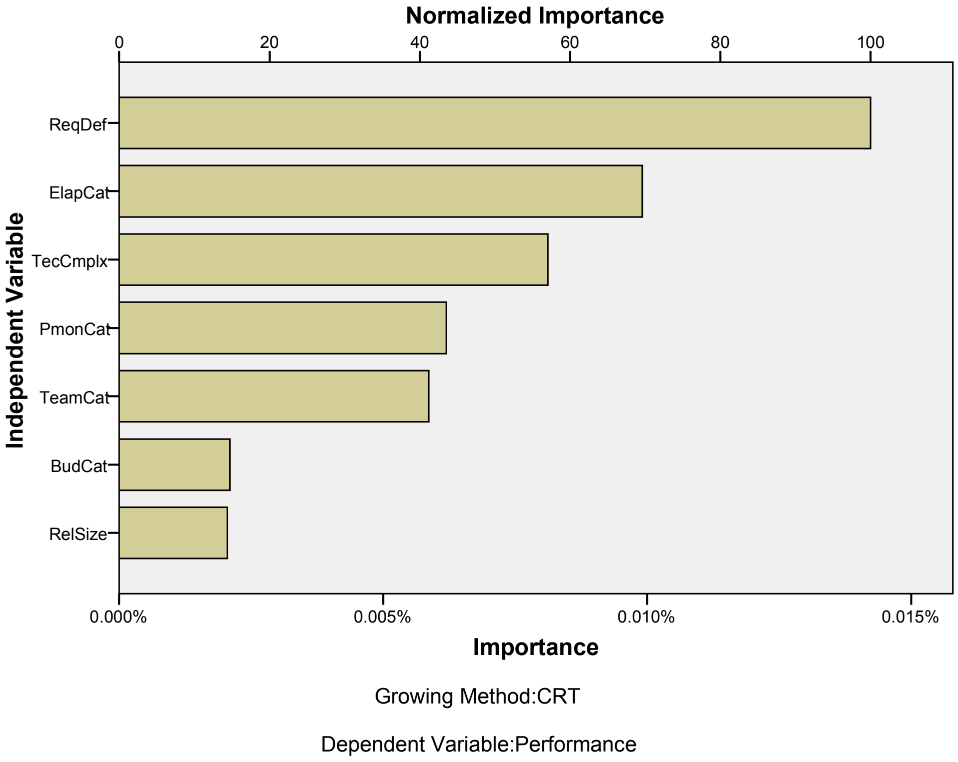
Category	%	n
🟦 UnderPerforming	0.0	0
🟩 Performing	100.0	14
Total	6.2	14

Node 8

Category	%	n
🟦 UnderPerforming	28.0	14
🟩 Performing	72.0	36
Total	22.0	50

Figure 2 demonstrates the effectiveness of the classification tree technique in providing easy to understand and actionable information for managers estimating project risk. The analysis suggests that the manager should pay attention to three important variables: Elapsed Time, Requirements Definition and Technical Complexity. One question that arises might be the relative importance of each of the included variables in the analysis. The output for the classification tree also provides an estimate for this. This is shown in Figure 3 below. Of these risks, the level of requirements uncertainty is identified as the most important. This results because the level of requirements uncertainty creates splits on both sides of the models (Node 2 and 3). This is an example of an actionable item. The classification tree shows that reducing the level of uncertainty will increase the likelihood of performance. Managers interested in lowering risk can therefore invest more time in developing requirements before beginning the project. The classification model also provides a probability estimate of how much lower risk can be expected. While the results are preliminary, this analysis provides some justification for the notion that properly defined requirements are a critical consideration in IT project risk assessment and overall project performance.

Figure 3: Relative Importance of Risk Factors on Estimating Performance/Under Performance



Conclusions

We began this paper by noting an assumption that when requirements are not well defined, an IT project faces an increased risk of poor performance. Researchers have established a set of IT projects risks and have provided some understanding of how these risks might impact performance. However, only a few papers have worked to estimate the effects of risks on performance (Tiwana and Keil, 2004; Zhang et. al., 2003). This paper has discussed three different estimating techniques to assess the impact of risk factors on IT project performance. These techniques included regression, neural networks and classification trees. A comparison of these techniques suggested that classification trees techniques have the potential to provide effective models that allowed managers to easily understand and act on the estimate outcomes.

To demonstrate the effectiveness of the classification tree results, a data set (n = 227) that included 9 risk factors as independent variables was used to develop an estimate of a binary (performing/underperforming) target variable. Results from this analysis provided a classification tree with 8 nodes. While the data set is too small to begin to develop validation of the model, the resulting tree demonstrates the ease of interpretation of the classification tree results. The results are clearly easier to understand than complex activation function in neural networks or the standard coefficients and t-statistics from regression. It is also easier to understand the interdependence of risk factors. For example, Node 3 and 4 in figure 2 suggest Technical Complexity becomes an issue in projects that are shorter (=< 6 months long) but is not a significant issue for longer projects. We have also shown how the level of requirements certainty provides an actionable item for reducing project risk. Based on this analysis, we conclude that a risk assessment tool based in classification tree technique as the potential to provide a new and effective tool in the management of IT project risk.

Limitations

We note several limitations to the results discussed in this study. The first is the relatively small sample size. This has the effect of limiting opportunities for useful validation samples as well as limiting the number of times in child and parent nodes. Increasing the sample size significantly should help to reduce the effect of small sample size. A further limitation was the small number of risk factors included in the analysis. A more complete analysis should include important factors such as the level of top management support, knowledge and experience of the project team, the level of organizational (user) support and others. The analysis presented in this paper demonstrates the effectiveness of the classification tree technique. More works needs to be done to develop a more comprehensive risk assessment model.

Suggestions for Future Research

This paper has suggested that classification tree techniques have the potential to provide more effective estimates of IT project risks assessments. The results from the classification tree techniques have demonstrated the techniques ability to provide information that is both easy to discern and actionable. Developing these techniques further will require larger data sets with more complete and sophisticated measures of IT project risks factors. Developing these large data sets should provide researchers with the opportunity to provide more effective early assessments of IT Project risk.

References

Apte, Chidanand, Liu , Bing, Pednault, P.D., Edwin and Smyth, Padhraic. "Business Applications of Data Mining." Communications of the ACM 45(8), 2002, 49-53.

Barki, H., Rivard, S., and Talbot, J. “An integrative contingency model of software project risk management”. Journal of Management Information Systems, 17, 4 (2001), 37-69.

Bishop, C.M., *Neural Networks for Pattern Recognition*, Oxford Press, 1995.

Breiman, L., Friedman, J., Olshen, R. and Stone, C., *Classification and regression trees*. Wadsworth International Group, Belmont, CA, 1984.

Haas, M.R. “Knowledge Gathering, Team Capabilities, and Project Performance in Challenging Work Environments”. Management Science, 52, 8 (2006), 1170-1184.

Keil, M., Cule, P.E., Lyytinen, K., and Schmidt, R.C. “A framework for identifying software project risks”. Association for Computing Machinery. Communications of the ACM, 41, 11 (1998), 76-83.

Lyytinen, K., Mathieson, L and Ropponen, J., “Attention Shaping and Software Risk: A Categorical Analysis of Four Classical Risk Management Approaches “, Information Systems Research, 9, 3, September, 1998, p. 233-255.

McFarlan, F.W. “Portfolio Approach to Information Systems”. Harvard Business Review, 59, 5 (1981), 142-150.

Nidumolu, S. “The effect of coordination and uncertainty on software project performance: Residual performance risk as an intervening variable”. Information Systems Research, 6, 3 (1995), 191-219.

Nidumolu, S.R. “A comparison of the structural contingency and risk-based perspectives on coordination in software-development projects”. Journal of Management Information Systems, 13, 2 (1996), 77-113.

Ropponen, J., and Lyytinen, K. “Can software risk management improve system development: An exploratory study”. European Journal of Information Systems, 6, 1 (1997), 41-50.

Ropponen, J., and Lyytinen, K. “Components of software development risk: How to address them? A project manager survey”. IEEE Transactions on Software Engineering, 26, 2 (2000), 98-112.

Sauer, C., Gemino, A., and Reich, B.H. “IT Project Performance: The Impact of Size and Volatility.” forthcoming in Communications of the ACM (2007).

Standish Group, 2003 “Latest Standish Group CHAOS Report Shows Project Success Rates Have Improved by 50%. Press release, March 25, 2003, http://www.standishgroup.com/press/article.php?id=2, Accessed: Feb 20, 2007..
.
Tiwana, A., and Keil, M. “The One-minute Risk Assessment Tool”. Association for Computing Machinery. Communications of the ACM, 47, 11 (2004), 73-77.

Wallace, L., and Keil, M. “Software Project Risks and Their Effect on Outcomes”. Association for Computing Machinery. Communications of the ACM, 47, 4 (2004), 68-73.

Wallace, Linda, Keil, Mark, and Rai, Arun (2004a),” Understanding software project risk: a cluster analysis”, Information and Management, Vol. 42, 2004, p. 115-125.

Wallace, Linda, Keil, Mark, and Rai, Arun (2004b), “How Software Project Risk Affects Project Performance: An Investigation of the Dimensions of Risk and an Exploratory Model”, Decision Sciences, Vol. 35, 2, Spring 2004, p. 289-321.

Zhang, Peter; Keil, Mark, Rai ,Arun and Mann, Joan “ Predicting information technology project escalation: A neural network approach”, European Journal of Operations Research, Vol. 146, 2003, p. 115-129.