Estimating Risk Management in Software Engineering Projects

Jaime Santos[1] and Orlando Belo[2]

[1] ISCTE/IUL, Portugal
[2] Algoritmi R&D Centre, University of Minho, Portugal

Abstract. Independently from the nature of a project, process management variables like cost, quality, schedule, and scope are critical decision factors for a good and successful execution of a project. In software engineering, project planning and execution are highly influenced by the creative nature of all the individuals involved with the project. Thus, managing the risks of different project stages is a key task with extreme importance for project managers (and sponsors) that should be focused on control and monitoring effectively the referred variables, as well as all the others concerned with their context. In this work, we used a small "cocktail" of data mining techniques and methods to explore potential correlations and influences contained in some of the most relevant parameters related to experience, complexity, organization maturity and project innovation in Software Engineering, developing in a model that could be deployed in any project management process, assisting project managers in planning and monitoring the state of one project (or program) under its supervision.

Keywords: Software Engineering, Project Management, Data Mining, Effort Estimation, Risk Management, and Project Success Classification.

1 Introduction

The lack of success has been a generic characteristic whenever they are related to new developments or just simple enhancements in information technology projects, particularly the ones related to software engineering. Is common in the majority of the projects, delivered all over the world, to closes affecting negatively one (or more) of the main project vectors: cost, duration, quality or scope**.** Several cases presented in studies and surveys, like the ones from KPMG in 1997 [1], the Standish Group in 1995 [2], or more recently in 2009, the ratios in what we call unsuccessful are very high. In a 2009 published survey, the Standish Group concludes that just 32% of projects ended within costs, time and delivering all functionalities required. We can accept that this kind of surveys generates some controversy, but the overall conclusions are always the same: there are higher rates of cancelled projects, over budgeting, and schedule failover.

The scope of this article is focused in project estimation process, since it yields some of the most important activities in project lifecycle, but normally, with low efficiency and highly neglected, being performed based on feeling, gusts or some other political factors. Since estimating should be based on a process, with quality

P. Perner (Ed.): ICDM 2013, LNAI 7987, pp. 85–98, 2013.

standards and a time consuming on benchmark analysis of the organization and market data, we easily understand why some project managers and there organizations neglected the process. Basically, this happens because the initial estimative represents one investment without consequent returns (e.g. proposals preparation), leading the organizations or IT departments to follow simplified procedures or eventually, skipped them, even when this is a subject highly referenced in project management methodologies, like *Project Management Institute* (PMI). PMI emphasizes this procedure as a main component to calculate cost, duration, and their relations to other components, like risk management [3]. It is important that the organizations introduce new procedures and models that are able to improve and facilitate the estimating process.

This paper presents and discusses a data mining application process addressing the effort estimating activities on software engineering projects, with the objective to reach a project classification model and a project effort estimation model. This paper is organized into 3 more sections, namely: section 2 that exposes some relevant issues in project management activities; section 3 that presents and discusses the entire data mining process carried out; and finally, section 4 that presents some final remarks and conclusions, as well as a few future research lines.

2 Augmenting Effectiveness on Project Management

Having the ability to capture information, predict the uncertainty, estimate dimension and there eventual impacts, planning all activities, time and resources and then, monitoring and controlling accordingly, are the most important tasks to manage a project aiming its success. To accomplish this task, the manager should have practice and knowledge in several relevant domains like: planning, risk management, relationship management and communications, giving him the ability to plan in an accurate manner, capture the project situation and then acting proactively to take corrective actions and mitigation plans. In order to monitor and control the project it's necessary to make some estimation. Usually, the first one happens at the planning stage, so during the execution phase it could be compared with reality and then, redoing or adjusting accordingly to the current situation of the project. This task is characterized by understanding and contextualizing the project scope and their characteristics as better as we can, producing a first estimation of effort, resources, duration, cost, defects, documents, and so on. Some other task that we highlight is the ability of the project manager and his team to capture and manage the scope.

The scope volatility related to a software engineering project, follows contours of higher complexity than those characterized by a repetitive nature. Thus, their unique and non-repeatable nature, along with the team and the project sponsors creativity, are major challenges and a serious risk in the project execution, since incrementing the scope will directly impact the other project vectors: cost, quality and time. The risk of error in the estimation process, the risk management framework and its amendments, or the unpredictability of actions for each stakeholder, has a direct impact on the cost, on the quality and on the time of the project. So, during a software implementation with a high degree of risk regarding is intangible and creative nature, along the fact that good governance is characterized by a proper risk management, with a tight control of variables, it is important to use concepts and methods for data collection

and retention on analytical processing systems, making the data available to apply data mining techniques, designed to develop models for estimating and forecasting, assisting in the planning and in the risk assessment. Data mining allows us to deep our knowledge about project management, helping us to extract behavioural evidences from historical data, while understanding relations between them. It is recognized that some characteristics affects the team productivity, so new knowledge will bring direct benefits in the estimation process, assisting us on understanding the impacts on productivity as well in the detection and quantification of risks. The acquisition of new knowledge, or the simple confirmation of some ideas taken for granted, can bring to us clear benefits to improve estimates or predict future events, enhancing the management of the inherent risk and the uncertainty present on those type of projects.

3 Mining Project Management Data

3.1 Overview

Project management is a highly complex activity that pushes several techniques and knowledge, focusing on extracting information from several objectives, deliverables and surrounded features, occurring on environments of constant uncertainty, being a continuous task of checking and acting. As already referred, this paper presents and discusses a data mining application process addressing the effort estimating on software engineering projects and also, methods to explore potential correlations and influences contained in some of the most relevant parameters related to experience, complexity, organization maturity or project innovation. Those methods could then be deployed in any project management process, assisting managers in planning and monitoring the state of one project or program under is supervision. Trying to assure a systematic approach for our work, we adopted and followed the CRISP-DM methodology [4]. First, we performed a research about the business related to the project management activities and to software engineering, attempting to understand the most important features, the environmental complexity and, trying to identify the most influencing characteristics on the project events and project risks across all phases, but specially, at planning and execution phases.

3.2 Data Sets, Acquisition and Preparation

We start data acquisition and preparation tasks extracting data from all the projects we considered with relevance to this study. Therefore, from 24000 projects available on our database, we selected only those whose purpose was related to application development or major enhancement applications development. It was further selected only those on a completed state, approved and available for metrics analysis. Thus, from the initial universe we extracted approximately 5000 projects. After this first step, we proceeded adding some more variables that arise from the junction with other tables presented in the database, such as type of industry, indicators of complexity, experience and project context, sizing, resources, etc. Later, other variables were incorporated, which resulted from the aggregation and transformation processes using some of the initial variables (totals for estimated values, indicators of failure, etc.). We explore the different variables, and by doing that, we detect that some of those (quantitative) variables did not show a normal distribution (figure 1). Thus, we opt for

the logarithm transformation. Three types of sizing were used on the projects, Function Points, that consists in a certified methodology for sizing application development, Lines of Code, representing the total number of lines coded to develop the application and "Others". Since "Others" are very diffuse, not quite understandable and not comparable, we decided to discard all projects having size calculated only with this type, resulting in the final dataset with approximate 4000 projects. We also detected missing values in several variables for some projects, what imposed several treatment acts.

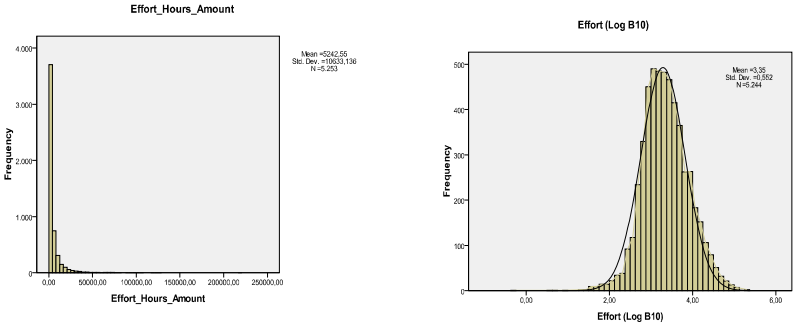

Fig. 1. Effort_Hours_Amount / *L*ogarithm histogram

The data set used contains a very broad representation in terms of geographic, technical and industry characterization. There are projects from more than 30 different countries, with major focuses in North America, followed by Australia and Europe (figure 2). The data has a hide industry representation (figure 3), from manufacturing, transport, energy, finance and even government entities. However, we can assist to the prevalence of manufacturing and finance. We also noticed that most projects were managed using some project management methodology, however, almost 20% did not use any formal methodology.

Country Region

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid		1213	20,6	20,6	20,6
	Australasia	791	13,4	13,4	34,0
	Central America	48	,8	,8	34,9
	Eastern Asia	17	,3	,3	35,1
	Eastern Europe	3	,1	,1	35,2
	North America	2786	47,3	47,3	82,5
	Northern Africa	10	,2	,2	82,7
	Northern Europe	429	7,3	7,3	90,0
	South America	158	2,7	2,7	92,7
	Southeast Asia	37	,6	,6	93,3
	Southern Africa	3	,1	,1	93,3
	Southern Asia	70	1,2	1,2	94,5
	Southern Europe	58	1,0	1,0	95,5
	Western Asia	2	,0	,0	95,5
	Western Europe	262	4,5	4,5	100,0
	Total	5887	100,0	100,0	

Fig. 2. Country Region frequency List

Industry Type

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Comms, Media & Entertainment	490	8,3	9,0	9,0
	Consumer Industries & Retail	31	,5	,6	9,6
	Internal	478	8,1	8,8	18,3
	Energy	14	,2	,3	18,6
	Financial Services	1151	19,6	21,1	39,7
	Government	471	8,0	8,6	48,3
	Healthcare	251	4,3	4,6	52,9
	Manufacturing	2134	36,2	39,1	92,1
	Multiple Industries	1	,0	,0	92,1
	Transportation	432	7,3	7,9	100,0
	Total	5453	92,6	100,0	
Missing	System	434	7,4		
Total		5887	100,0		

Fig. 3. Industry Type frequency List

During the preparation phase, and to better understand our data, we also explore some correlations between different variables; however, we didn't reach any significant correlation. We expected an immediate identification between sizing and effort, but ultimately the data showed very weak correlations, i.e., with Pearson correlation coefficient of 0.15 in relation to the functional size by 'function points' and 0.100 for the size in 'lines of code'. A wide representation of different programming languages can explain this fact. As is known, the relationship between language and effort is large, so, it is difficult to find correlations standing before such representative data. This is indeed the problem associated to the estimation, i.e. the existence of large dispersion and great amplitude in the factors that affect the productivity.

In a second attempt to demystify the foregoing, it was decided to perform an analysis on the correlation between effort and sizing for two types of programming language, having reached to values of significant Pearson correlations of 0,577 and 0, 564. Notwithstanding the foregoing, there are some interesting correlations detected, but not surprising, such as: strong correlation between the three variables of complexity classifiers - application innovation, technological innovation and BUS innovation; and also a strong correlation between some of the variables that classifies experience - computer experience, tools, language, methods and technology experience. There were a correlation of 0.988 between "FTE Amount" and "Effort Hours Amount". Being FTE (Full Time Equivalent) an expression used in business to summarize the total of man/months, then the correlation is completely acceptable. This conclusion allows us to reduce the dimensionality and complexity of the analysis by removing the variable "FTE Amount", not be considered in any following steps, in particular, at the mining tasks. It is more surprising to note the clear existence of a correlation between client complexity and the team complexity, holding a correlation of 0.388. This can show us that client as interference in how a manager constitutes his team, whether it is directly or inadvertently. Regardless the fact that we have not identified major surprises in the correlations, with the presence of very low rates, it is however possible to see differences in effort, for example, in the ratios between effort and sizing. To this end, we subdivide the data into three group types:

1) One group which sizing data was calculated according to FPA methodology.
2) Another group which sizing data was calculated according to methodology of lines of code.
3) A final group which sizing data was calculated using both methods.

Thus, we proceeded for an average comparison, according to a diverse set of deterministic variables. The OneWay ANOVA [5] method was applied to compare the means, separating each sample according to the experience, the complexity, the innovation and the maturity classifiers (e.g. table 1). The different populations were then defined according to a pre-existing data characterizing the level of each project.

Table 1. Populations characterized by experience in: project management, system, tools, programming language, methods, etc.

Code	Description
1	*Less Than 1 Year*
2	*1 - 3 Years*
3	*Greater Than 3 Years*

Before we go forward with the comparisons, there were some important preconditions to be verified for the feasibility of the chosen method. The first condition is to assure that the test variable is quantitative, which in our case the condition were guaranteed. Second, there must be a variable that defines the nominal groups. In our cases, all variables used are nominal and we can use the average function for each of its dimensions, thereby ensuring the suitability. In addition to the conditions set out above, it is still assumed that the variable under test follows a normal distribution, which was not the case. To be possible to go further, we decide to calculate a logarithm base 10 for both ratios, yielding so the tendency for the normal distribution, the desired condition for this test. Finally, to apply the OneWay ANOVA method is also assumed that there is an equality of variances in the different populations for the variable under study. For evaluating that condition we've performed a test of homogeneity of variances, which were defined by the following assumptions:

- H0: Variances are equal.
- H1: Some of the variance differs from other.
- $\alpha = 0.05$ (Alpha definition for the rejection of the null hypothesis).

Looking at the Table 2 we reject the null hypothesis, i.e. that there isn't equality of variances, since the value of Sigma is below the alpha (α).

Table 2. Test for the Homogeneity of variance

	Levene Statistic	df1	df2	Sig.
Ratio Effort by FP(LogB10)	9,894	6	2471	,000

Given this situation, generally occurred in all tests with the populations used, we were forced to abandon the ANOVA test, being resorted to a more robust equality of means test, which is the test of Brown-Forsythe and Welch [6].

Table 3. Robust Tests of Equality of Means[b]

		Statistic[a]	df1	df2	Sig.
Ratio Effort by FP(LogB10)	Welch	36,078	6	313,308	,000
	Brown-Forsythe	27,709	6	239,232	,000

a) Asymptotically F distributed.

Since in all cases our sigma (which indicates the variability) was zero, then we reject the null hypothesis, i.e. there is no equality in the average ratios. Thus, in terms of overall conclusion, the presentation of different mean values between different populations, leads us to conclude that the variables under study interfere effectively in productivity, and in turn, the effort required to produce one function point or one line of code, so, they had been considered very important variables to use in testing techniques, which would lead to the classification and estimation. As an example, in Figure 4 we can see that the mean has a tendency to increase as the complexity of the team increases, so the more complex the team is, the lower the productivity index by lines of code or by Function Points.

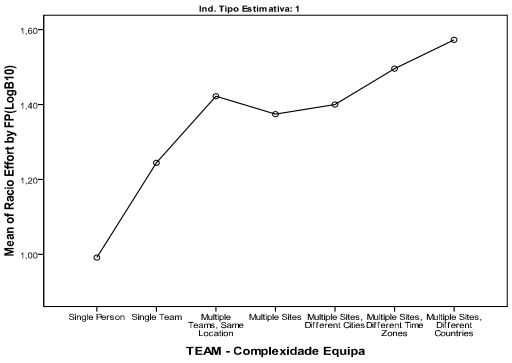

Fig. 4. Means comparison Chart: Team Complexity

Looking the variables related to innovation, the findings are as expected, i.e., there is a clear increase in the ratios as there is a higher rate of innovation in the project. However, the variable related to the application innovation presented a quite interesting behaviour, since the tendency is reversed to the initial expected, which means that the ratio decreases when the application innovation increases, so, less effort is need to produce the same size in a more innovative application. However, it's normally expected that teams struggle with some problems in innovative projects, negatively affecting their productivity. Perhaps this just occurs under the influence of programming languages, which modernization allows teams to deliver more functionality in less time! The trend found at the level of application innovation was also detected in the variables

characterizing the experience, no matter if occurs in the teams, about their system experience, language experience, or even in the project manager experience managing projects. Thus, this is a very important result, because it doesn't confirms the general knowledge that the experience as a positive impact in the productivity. What is certain is the fact that this conclusion may be itself as one of the most important in this work, since the normal thought is contrary to the one found here.

When we look to the ratios (Effort by FP or effort by LOC), separating the populations by organizational maturity (CMMI rating), we denote the benefit that a company can get by moving from a non-documented and possibly disorganized level (CMMI Level 1) to one with organizational evidences with use of standards. However, as the organization moves up the maturity, the impact in terms of productivity is achieved in a negative way. This may be caused by the existence of more bureaucracy, higher-level documentation. So, projects in organizations with higher maturity level turn out to be impacted by the amount and complexity of documents that have to bear, as well as the procedures for review and audit they are subjected.

3.3 Classification and Estimation Efforts

Considering that the target variable (Check_Sucess) was categorical binary and the fact that we are facing a world of mostly categorical or nominal data, we chose to apply two classification techniques: the C&RT classification and regression trees [7], and the C5.0 decision tree [8]. The initial universe of projects for the classification process integrated 3644 projects instances (after delete some cases to achieve balanced data set), containing a perfectly balanced data set of cases in which resulted in success or failure, remaining 1822 projects classified as failure instances. In order to ensure higher quality testing techniques, the data set was divided into three subsets, having respectively 35% for training data, 35% for test data, and 30% for validation data. Regarding the validation and for model quality evaluation purposes, we choose the misclassification error rate method [8]. In order to be able to perform the mining process, specifically the classification task, it was important to pre-characterize the target variable as to success or failure (see table 4).

Table 4. Previously classification of: Success vs. Failure

Pre-classify	Value
Success	1
Failure	0

With C&RT, the classification process was run in two modes: simple and advanced. In both cases, the results were the same, with a good ability to hit the projects targeted as success but with a high cost of misclassified cases for the ones considered with failure. Given the nature of the business, and expecting that this model help managers to anticipate risk situations in their projects, it is preferable a model that presents the best ability to classify failure to the detriment of those who had success. From the pre-classified cases of failure, the resulting output from this technique classified 1118 as success, corresponding to 61% of misclassification.

Regarding the poor results presented by this technique, it was no longer taken into account, not been used for any comparison step with other techniques. Next, we chose to train C5.0 decision tree using two modes, simple and advanced (as we did with C&RT). With this technique the situation was quite different, since there had been improvements in the classification rate, with the simple method achieving rates of 58% in the classification of cases of failure and 70% of success, but we continue to consider ineffective and with unsatisfactory results for the most important cases, the ones classified as failure. Alternatively, and after several training sessions, the execution of the advanced mode were performed with the option of 'pruning severity' equal to 50, the 'boosting' option enabled for a number of five attempts and the 'cross-validation' option also activated for a total of five folders with a minimum number of records by node of five. The advanced mode had a better ratio of good classifications, with a percentage of 65% accuracy on projects previously classified as failure, as we see on figure 5.

		$C-check_Sucesso	
check_Sucesso		0	1
0	Count	1176	646
	Row %	64.544	35.456
	Column %	65.116	35.147
1	Count	630	1192
	Row %	34.577	65.423
	Column %	34.884	64.853

Fig. 5. C5.0 Advance Mode: Classification Matrix

For the estimating task, looking to estimate the project effort, we used multiple regression [9], neuronal networks [10] and CHAID decision trees [11]. In the first trainings performed we used the complete data set, having projects whose sizing was calculated in 'functions points' or 'lines of code'. Since the results did not show a minimum quality required, we proceeded to split into two subsets, according to the method of sizing and due to time constraints it was decided to perform this task only for the universe of projects with the calculation of 'function points'. By using as first method a multiple regression, we just intended to verify if we can achieve some improvements in the final results, comparing it to more advanced techniques, such as neuronal networks.

Comparing $E-Effort_Hours_Amount with Effort_Hours_Amount

'Partition'	1 Training	2 Testing	3 Validation
Minimum Error	-31458,412	-60132,131	-79219,041
Maximum Error	71728,521	78644,846	60957,175
Mean Error	162,833	-301,841	-459,226
Mean Absolute Error	4696,258	4777,64	5065,872
Standard Deviation	8221,961	8908,419	9730,993
Linear Correlation	0,597	0,544	0,423
Occurrences	799	834	764

Fig. 6. Multiple regression – evaluation

To execute this technique, as the subsequent, the training was performed using the base variables, without any treatment, and then we always repeat the training using the size variables converted, in our case, to their logarithms or resulted from the min-max standardization method intending to achieve a normal distribution (since all

quantitative variables had a left skewed distribution). The multiple regression technique demonstrated to be incapable to result in any model when executed with the variables converted.

With neuronal networks we performed several training sessions, using several execution modes, several layers and neurons, presenting above the two with the best results obtained. In these two cases, the neuronal network was executed with the prune method [12], one with a simple mode and the other in advanced mode, using two layers, the first with three neurons and the second layer with seven neurons – Fig. 7 presents the most important variables used to estimate the effort. This figure has been extracted from the one reached with prune advance mode and it result in a simple model, with less number of variables, which is quite important for implementation purposes.

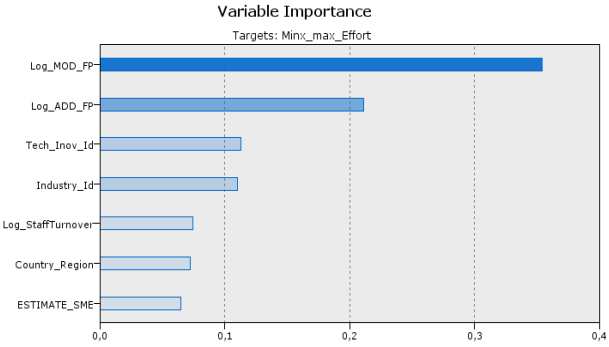

Fig. 7. Neuronal network with prune advanced mode - variable importance

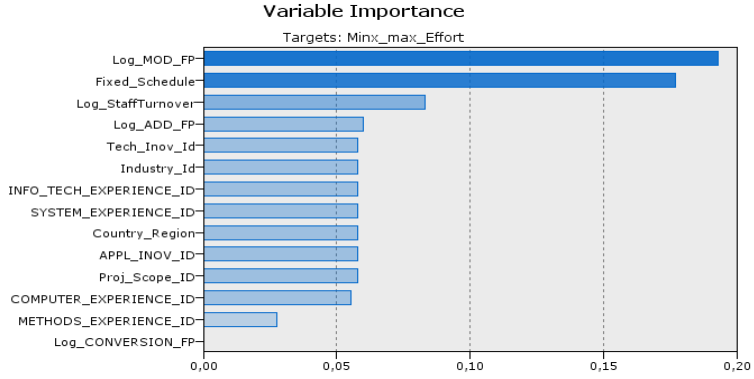

Fig. 8. CHAID decision tree - variable importance

Finally, we generated a decision tree with CHAID. It is possible to see that the values of 'sizing' hold a central importance, with CHAID method capturing some of the project context variables, those that could cause some variability in the productivity rates (figure 8).

3.4 Results Evaluation

Using the results obtained in each model, we done a small comparison process (Table 5), which allows us to evaluate the results according three indicators: overall error rate (OER), false positive rate (FPR), and false negative rate (FNR).

Table 5. Classification Task - results comparison

	OER	FPR	FNR
Decision Tree C5.0 – Simple	35,48%	29,75%	41,22%
Decision Tree C5.0 – Advanced	35,02%	34,58%	35,46%

We selected the most important variables in each model, and although there is a correspondence between models in order to the importance of the variables, we observe that: both models presented the 'client_participation', 'fixed_staff', and 'industry_id' as variables with explanatory power for failure; and the two models agree on the importance of variables related to experience, complexity and constraints of the project ('fixed team', 'fixed cost', etc.).

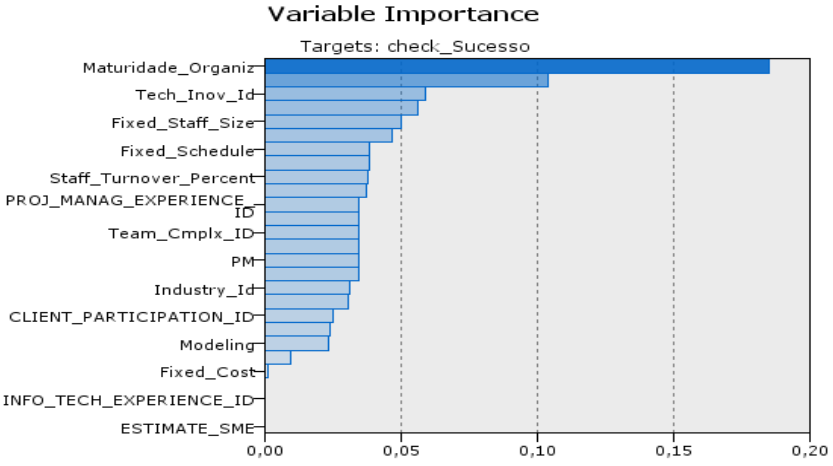

Fig. 9. C5.0 Simple - variable importance

We also observed that the tree generated by the algorithm C5.0, in advanced mode, has not only a lower overall error rate (35.02%) as a lower rate of error in the prediction of false negatives, that is, projects with failure misclassified as a success. Although the results did not reach the expected values, a good classification rate around 65% will allow the model to be implemented as a risk management tool.

In terms of the estimation tasks, we noticed the evident efficiency from all techniques comparing the results with the ones of multiple regression (Table 6) - all values were reached after applying the *min-max* normalization on the results extracted from the evaluative matrix.

Table 6. Estimation task - results comparison

	Minimum error	Maximum error	Average error	Mean Absolute Error	Standard Deviation
Multiple regression	-0,896*	0,670*	-0,015*	0,045*	0,097*
Neuronal Network – Prune Simple	-0,327	0,641	0,011	0,044	0,074
Neuronal Network– Prune Advanced	-0,358	0,639	0,003	0,051	0,090
Decision Tree – C&RT	-0,391	0,661	0,001	0,057	0,103
Decision Tree – CHAID	-0,320	0,658	-0,001	0,055	0,098

Looking at figures 7 and 8, we see that the different models agree about the variables regarded as the most explanatory. Multiple regression and C&RT decision tree are those with a simpler model, having concordance to the others in relation to the sizing variables, only. All models also feature that the sizing variables has high explanatory power for estimate effort, and there was agreement on the importance of the 'Staff Turnover' variable. There is also a widespread agreement among the various algorithms regarding the importance of variables related to experience, complexity and project constraints (team fixed and fixed cost, etc.). However, this agreement is substantially higher between the CHAID decision tree and neuronal network.

Finally, we can verify that a multiple regression showed good results in the estimation. However, it does not contain the variability resulting from the context in which a project is related. The same goes for the C&RT decision tree. It should be noted that the challenge presented in the work intend to capture this variability and thus, adjusting the estimate to be more effective, since the 20% deviations normally assumed, contains a high impact on the project's cost and the expected success. Thus, taking into account the variability of the above-mentioned issue and the results, we think that is appropriate to implement the model obtained by the decision tree CHAID.

4 Conclusions and Future Work

In this paper we presented two models that can be used in the software engineering projects management. The preliminary study of the source database and the data collection process resulted in one of the most challenging components and consuming effort of this work. The fact of having to use a relational database, with more than 200 tables, over any data mart or pre-prepared file, cause that this task became very time and resource consuming. However, this was a very important phase, because it allowed us to delve a little deeper on the business of managing software engineering projects, but mainly, it makes possible the data understanding, enabling the choice of tasks to perform, launching new challenges to the future.

This work helped us to detect within the database some important information to use in a mining process, but it was also detected some gaps and needs that should be addressed in a near future. Considering the available data, the source contains relevant information for the execution of any data mining task, not meaning that the database can't be further enriched, e.g., with detailed information about each stakeholder, like indicators of attitude, resistance to the project, level of communication, among others. The presence of many projects without the minimum quality for analysis was the major problem identified in the database, cases which have been considered by the database internal auditors. These situations were reflected in almost half of the initial data universe, and it is important to define actions aimed to improve the quality of the data. Another challenge that was put during the execution of the mining process, was due to the existence of a multitude of different programming languages, which direct influence the effort and cause data dispersion and a large deviation, so that, a segmentation task can bring clear benefits to future analytical works. It is important to note that was detected in the database some additional information related to documentation type and quantity produced, as well data about changes in the several project deliverables, which can make possible to conduct new mining processes, in particular, to estimate total number of pages of documentation or association tasks related to changes in the project. One of the most significant trends found, were, at the level of the innovation and the experience variables, for team and project manager. It was expected a trend towards an increase in productivity as well as experience increased, however this happens in reverse. We think that this occurs mainly because experience will make more positive impact in overall quality of the project (all deliverables: the software, the documentation, etc.) then it does to the productivity.

The paper demonstrates the complexity that involves a software engineering project, and although from a long time the sizing values allow us to estimate effort with a certain degree of confidence, this is not enough. On the one hand, an overvalued estimate puts at risk the victory in a competition for a project, as an undervalued estimate, causing 10% or more deviations in costs; it will direct impact on the organizations and their viability. Looking to the resulting models achieved by the tasks performed in this work, we think that the estimation model created can be implemented in various software engineering projects as an alternative tool to the techniques and methods commonly used, which representative spectrum confers a generic capabilities, while the results given a confidence in their applicability. Despite the total error rate of 35%, we think that the resulting model from the classification task can be incorporated into risk management procedures of any software engineering project, since the early detection of a disaster will allow making on time decisions and the necessary corrective actions. So, implementing this model into risk analysis, at the planning stage as well during project execution phase, will enable "what if" scenarios execution and test, enabling the manager to measure and validate several alternatives for correction or improvement, understanding how we can increase the chances of success. Another interesting mining process that can be done should aim a model resulted from stakeholder's segmentation or classification task. This could provide tools to the managers that allow him to manage each one at the most appropriate way, taking preventive actions that help to minimize impacts. This example is something that could be implemented in areas like communications management.

References

1. Whittaker, B.: What Went Wrong: Unsuccessful Information Technology Projects. KPMG Consulting, Toronto (1997)
2. Standish Group.: The Standish Group Report: Chaos (1995), http://www.projectsmart.co.uk/docs/chaos-report.pdf (acedido em January 17, 2011)
3. PMI.: A Guide to the Project Management Body of Knowledge: PMBOK Guide, 4th edn. Project Management Institute, Newton Square (2009)
4. Chapman, P., et al.: CRISP-DM 1.0: Step-by-step data mining guide. The CRISP-DM Consortium (2000)
5. Looney, S.: Biostatistical Methods, vol. 184. Humana Press, University of Louisville School of Medicine, Kentucky (2002)
6. Almeida, A., et al.: Modificações e alternativas aos testes de Levene e de Brown e Forsythe para igualdade de variâncias e médias. Revista Colombiana de Estatística 31, 241–260
7. Breiman, L., et al.: Classification and Regression Trees. Wadsworth, Belmont (1984)
8. Larose, D.T.: Discovering Knowledge in Data: An Introduction to Data Mining. John Wiley & Sons, Inc., New Jersey (2005)
9. Cohen, J., et al.: Applied multiple regression/correlation analysis for the behavioral sciences, 2nd edn. Lawrence Erlbaum Associates, Hillsdale (2003)
10. Ripley, B.D.: Pattern Recognition and Neural Networks. Cambridge University Press, Cambridge (1996)
11. Kass, G.V.: An Exploratory Technique for Investigating Large Quantities of Categorical Data. Applied Statistics 29(2), 119–127 (1980)
12. Cantú-Paz, E.: Prunnnig Neuronal Networks with distribution estimation algorithms. Center for Applied Scientific Computing. Lawrence Livermore National Laboratory, Livermore (2003)