A BBN based approach for improving a Telecommunication Software Estimation Process *

Stamatia Bibi[1], Ioannis Stamelos[1], George Gerolimos[2], Vangelis Kollias[2]
[1] *Department of Informatics, Aristotle University of Thessaloniki, 54124, Thessaloniki, Greece*
[2] *TELETEL SA, 124, Kifissias Ave., 11526 Athens, Greece*
[1]*{sbibi,stamelos@csd.auth.gr}*
[2]*{G.Gerolimos,V.Kollias@TELETEL.eu}*

Abstract

This paper describes analytically a methodology for improving the estimation process of a small-medium telecommunication (TLC) company. All the steps required for the generation of estimates such as data collection, data transformation, estimation model extraction and finally exploitation of the knowledge explored are described and demonstrated as a case study involving a Greek TLC company. Based on this knowledge certain interventions are suggested in the current process of the company under study in order to include formal estimation procedures in each development phase.

1. Introduction

Software Process Improvement (SPI) is widely acknowledged as one of the most important means for achieving competitive and effective software industry. In literature one can identify several Software Process models [4],[5],[6]. However the adoption of such models by a Small Medium Enterprise (SME), can be proved time consuming and often unreachable as it requires appropriate experience and knowledge on how to define and implement improvement actions at a reasonable cost [10].

Teletel SA is a TLC software company. In order to stay competitive they attempt to create a predictable well defined and structured software process that will lead to rapid application development. For this reason issues such as accurate project effort and duration estimation [3], potential components reuse and quality assurance are of high importance. When we interviewed company managers, they pointed out that it was crucial for them to adopt models that have both predictive and explanatory value, whose results can be intuitively confirmed.

Bayesian Belief Networks may provide such a formal framework, complying with the above requirements. BBN users are able to model software process at various levels of abstraction, visually present the activities performed and their dependencies, as suggested in [9]. As an example, BBNs can provide as an output the estimation of the effort, duration, size of team or code reuse required for the completion of the project.

In this paper we apply BBN both in the data of the company and also we suggest certain interventions in the development process of the company that involve information gathering and estimate generation.

The paper is organized as follows: Section 2 presents Bayesian Belief Networks, Section 3 describes the methodological approach used for the study. Section 4 discusses results derived from the company data, section 5 discusses the interventions in the process of the company. In section 6 we conclude the paper.

2. Bayesian Belief Networks

Bayesian Networks are Directed Acyclic Graphs (DAGs), which are causal networks that consist of a set of nodes and a set of directed links between them, in a way that they do not form a cycle [8]. Each node represents a random variable that can take discrete or continuous finite, mutually exclusive values according to a probability distribution, which can be different for each node. Each link expresses probabilistic cause-effect relations among the linked variables and is

* This work has been funded by the Greek Secretary of Research and Technology, action 4.3.1, DIERGASIA project.

978-0-7695-3323-0/08 $25.00 © 2008 IEEE
DOI 10.1109/PCI.2008.39

depicted by an arc starting from the influencing variable (parent node) and terminating on the influenced variable (child node). The presence of links in the graph may represent the existence of direct dependency relationships between the linked variables (that some times may be interpreted as causal influence or temporal precedence). The absence of some links means the existence of certain conditional independency relationships between the variables.

The strength of the dependencies is measured by means of numerical parameters such as conditional probabilities. Formally, the relation between the two nodes is based on Bayes' Rule:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

For each node A with parents B1, B2,…, Bn an NxM Node Probability Table (NPT) is attached, where N is the number of node states and M is the product of its cause-nodes states. In this table, each column represents a conditional probability distribution and its values sum up to 1.

A simple hypothetical BBN estimating software effort is the one presented in figure 1. Attached to the node of effort there is a node probability table that provides possible effort values according to the development platform and the language in which the project is developed.

In this study the extraction of BBN is achieved with an award winning algorithm [1].

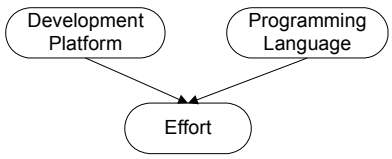

Figure 1. A BBN for software effort estimation

3. Research Approach

After interviewing the software quality team of the company we reached the conclusion that basic estimation issues in Teletel are in order of priority: estimation of the effort needed to complete a project, estimation of the duration of the project, software reuse of components of the particular project. Also the assessment of the size of the team that works on a certain project was an important issue for them.

The interviews were enlightening in order to identify the following:

-Which metrics can provide useful information for the company's software process?

-When and how these metrics will be collected?

-Which mathematical models will be used to analyze these metrics and create the process models?

We decided to collect data regarding process and computer metrics. Also among the mathematical models suggested to analyze the data, Bayesian analysis was the one that seemed more appropriate to the needs of the company. Bayesian Networks have been already used in the context of the estimation and process modeling [2], [11].

The next step of our research approach involved the extraction, deployment and evaluation of the proposed models on Teletel software lifecycle. We had to cooperate tightly with Teletel in order to come up with the appropriate BN model that would be able to fulfill their needs. The generalized process we followed is depicted in figure 2.

Initially we defined a complete set of process and product metrics that is recorded in literature and is considered for the benchmarking of a software project. Then, in cooperation with company managers, we selected those metrics that were most appropriate and relevant to the software that the company develops. Finally the metrics that participated in the analysis were further reduced to the ones that the company was able to collect and were derived either by the project manager of each project or by the limited information recorded at the time of the development of the project. The next step is to select the projects that will be used for the evaluation of the process models. The selected on-going projects had to be similar to the historical projects used to define the process model. Also for these projects all metrics defined in previous steps had also to be collected. After collecting all the appropriate metrics for the historical and on-going projects the parameters of the BN model had to be defined.

Certain important parameters that had to be defined and affect the structure and the estimation capability of the model are the following:

The initial order of the metrics. The metrics (variables) are represented as nodes in the BBN. Each node may be dependant on other nodes (an arc pointing towards this node) or may affect other nodes (an arc from this node towards other nodes). All variables before entering the model should be set in an order that will affect the direction of the arcs of the BBN. Each variable may depend only on the variables before it and can affect only variables after it. For example the first information we are aware of during the initialization of a software project is the business type of the project. Knowing this information we can define the platform on which the project will be developed. In that case the ordering of the two variables is 'Business type', 'Platform' and therefore the arc between the two nodes is directed towards 'Platform'.

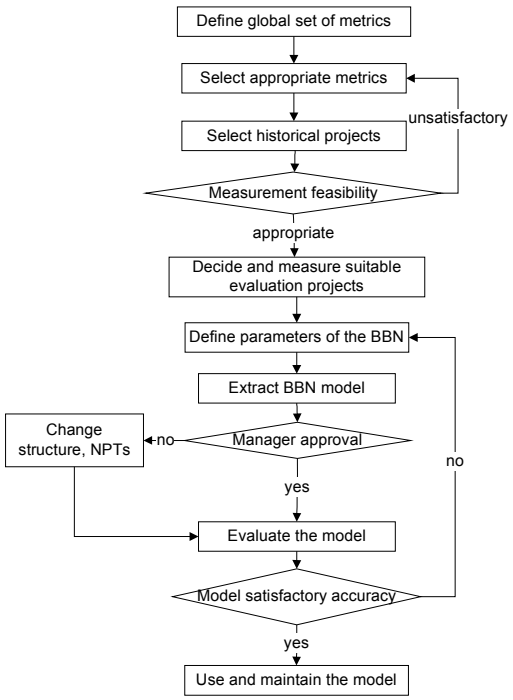

Figure 2. Methodology used for deriving the process model

Transformation of continuous variables into categorical. This transformation is necessary due to the nature of our BN approach and due to the fact that it is safer to produce estimation in intervals. The decision that has to be made at this point is the number of intervals and the way the projects will be allocated to an interval. There are many mathematical formulas that can provide the number of intervals. The splitting of the data into intervals is usually performed by one of the above methods: equal frequency binning, equal width binning, clustering.

The extraction of the BBN and the approval of it by the company is the next step. The initial BBN is a result of statistical analysis that uses the empirical distribution of the values of each variable in order to derive the structure of the BBN and the probabilities of the NPTs. The structure of the BBN should be approved by the managers of the company in order to be also verified and accepted intuitively. If the managers do not approve the structure of the BBN it is possible for them to add, delete or modify certain arcs and their direction that define the dependencies among the metrics. Also in certain NPTs the probabilities for the values of each variable may be close to each other, a fact that renders the estimation ability of the model weak. For this reason, refinement of the NPTs may be necessary, using information from managers in order to update probability values in the model.

Finally the evaluation of the BBN on the on-going projects is necessary. This evaluation will show the way that the BBN should be used and whether the estimations based on it are successful, and agree with the opinion of the managers. If the results of the evaluation of the BBN are not satisfying then certain steps are repeated. Possible problems can be the data used in the study (remedy: collection of more metrics, consideration of more projects), the definition of the parameters of the model (remedy: use different interval of effort, different order of variables) or unrealistic final BBN models (remedy: change of structure, NPTs).

When finally the BBN is extracted and is in agreement with the managers needs and successfully evaluated on more recent projects the next step is to ensure that the model is continuously updated. This can be achieved by entering new data to the model.

4. Data analysis

Fifteen projects were selected to participate in the analysis. More projects were available but we preferred to exclude some projects that were very different from the average company project (e.g. had only a small software component or were quite old). Ten out of these projects were completed and used as a training set. Five of these projects were used as a test set.

The process, product and implementation variables are presented in table 1. Teletel process is composed of two phases (P1 and P2).

The extracted BBN is presented in figure 3. During the initialization of a software project the first information given to a project manager is the business type of the application. It seems reasonable that the business type of the application may affect the reuse of code from past projects, because code that comes from similar business applications may be considered to be candidate for reuse. The causal relationships among the implementation attributes identified point out that BusinessType affects also the selection of Platform and the size of the project (LOC). On the other hand the platform affects the language with which the software is implemented.

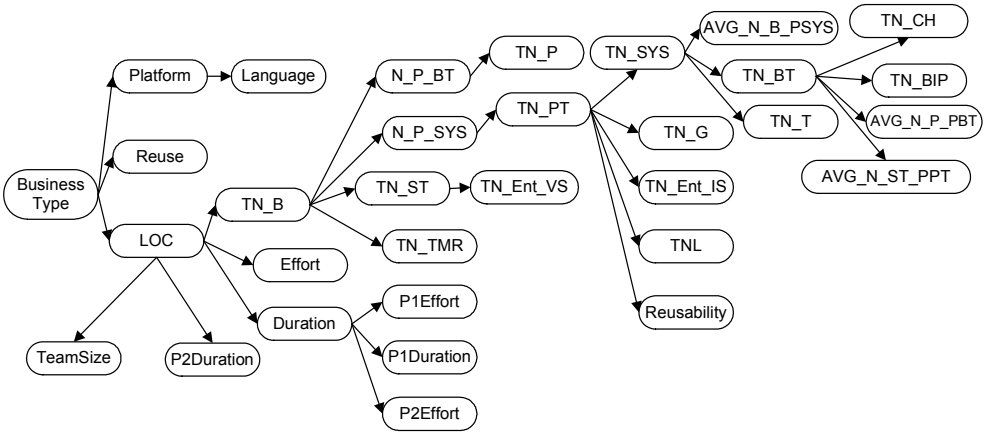

Figure 3. BBN for the estimation of all project attributes

Table 1. Process and product variables.

Variable
LOC (Lines of Code)
Duration (months)
Effort (months)
P1Duration (P1=analysis & design phase, man months)
P1Effort (man months)
P2Duration (P2=coding & testing phase, man months)
P2Effort (man months)
TeamSize (number of people in the project)
Reuse (% of reusage of previous project products)
Reusability(% of the project products reused)
TN_B (Total Number of Blocks)
N_P_BT(Number of Processes in a Block Type)
N_P_SYS (Number of Processes in a System Type)
TN_P (Total Number of Processes)
TN_ST (Total Number of States)
TN_PT (Total Number of Process Types)
TN_SYS (Total Number of Systems)
TN_TMR (Total Number of Timers)
TN_BT (Total Number of Block Types)
TN_T (Total Number of Data Types)
TN_G (Total Number of Gates)
TN_CH (Total Number of Channels)
TN_BIP (Total Number of Built in Procedures)
TN_Ent_VS (No. of SDL Entities with Valid Suffix)
TN_Ent_IS (No. of of SDL Entities with Invalid Suffix)
AVG_N_B_PSYS (Avg. No of Blocks per System)
AVG_N_P_PBT (Avg No of Processes per Block Type)
AVG_N_ST_PPT (Avg. No of States per Process Type)
Platform (WindowsNT&Linux, WindowsXP)
Language (C++, JAVA, SDL)
Date (2002- 2007)
BusinessType (TLC, TELEMEDICINE, DEFENSE))

The variable that connects implementation metrics to process and product metrics is BusinessType. BusinessType affects LOC that is the parent node of many process metrics and TN_B. It is reasonable that the size of a project affects the effort, duration and the size of the development team. Smaller projects require less effort, fewer people and last less months compared to bigger projects. The duration of the project affects the effort and duration of the 1st development phase and the effort of the 2nd phase, as suggested by managers. The analysis showed that projects that last 9.5 months or less require less than 5 man-months for analysis and design and 3.5 man-months or less for coding and testing. Also the duration of the 1st phase of these projects is 4.5 months or less.

For this BBN the process and product metrics are affected indirectly by the BusinessType of the application. TN_B, TN_PT, TN_SYS and TN_BT seem to affect the rest of the code metrics. TNL which is an indication of the size of an embedded project is mainly affected by TN_PT.

The estimation accuracy of the BBN of figure 3 is presented in table 2.

Table 2.Estimation results for process variables.

Variable	Hitrate
effort	80%
duration	60%
team size	60%
P1Effort	60%
P1Duration	80%
P2Effort	40%

5. Software Process Enhancement

In this study based on the metrics collected and the estimation results we suggest certain interventions in the current development phases of the company.

In the initial phase, 'Project Definition', we suggest the gathering of data relative to the project, such us the development type (Enhancement, Re-development, New Development), the application type

(TLc, Embedded, e.t.c), the organization type (Defense, Medical, etc).

During the 'Requirements Definition' the company can collect data relevant to the implementation, for example the development platform, the programming language, the DBMS, the CASE tools used and the development methodology. Also in this phase, initial estimations regarding the process metrics (effort, duration, team size) can be performed.

During the phase of the 'Design of the commercial and technical solution' metrics relative to the size of the software can be recorded such as number of modules, number of test modules and % of past projects code reusage. The initial estimates from the previous phase can be adopted and also some further estimates regarding code metrics (e.g SDL metrics) can be performed.

In the 'Product Development' phase the actual values of process and product metrics should be recorded and initial estimates of the quality of the software product (e.g no of defects) can be performed.

In the last two phases 'Delivery and Installation' and 'Maintenance and Technical Support' fault information data gathering is suggested.

6. Conclusions

In this paper we have analyzed recent process, product and implementation data coming from a small-medium TLC software company. We suggested a methodology to model the development process using Bayesian analysis to identify relationships among the project attributes.

The proposed methodology provides the means to a small-medium software company to start collecting, analyzing, evaluating its own data, albeit the fact that few projects have been implemented, in order to achieve a measurable, well-defined process. It can be used also to analyze cross-company data [7], providing alternative models to those coming from the company itself. In general, we suggest the use of BBNs in order to represent and use domain knowledge coming both from single and cross-company project data.

The results described in sections 4 and 5 and feedback from the company key personnel show that the analysis performed can be a useful tool in the hands of experts. BBNs can assist in estimating under uncertainty at the early stages of software development knowing just the initial information for a project. The model can be updated any time information from new projects is available. A project manager may also continuously apply the suggested model during all phases of software development and feed new information as the project evolves in order to refine his estimates.

Also the suggested model can be used for post mortem analysis in order to reach to some conclusions, for example which projects demand more effort, time, or personnel in order to be completed. This is particularly useful for someone who is forced to make an early estimation. Such analysis can provide indicators of potential success or risk for on-going projects.

One crucial observation is that the analysis performed considered a limited number of projects and consequently estimation outputs were limited to just two intervals. We are in the process of gathering data for on-going projects from the same company in order to be able to stress further the overall method and provide more elaborate estimates.

7. References

[1] Bayesian Belief Network Software, http://www.cs.ualberta.ca/~jcheng/bnpc.htm

[2] S. Bibi, I. Stamelos,: Sofware Process Modelling with Bayesian Belief Networks, Online Proceedings of the 10th IEEE International Conference on Software METRICS , September 2004, Chicago, USA.

[3] B.Boehm, R. Fairley, Software Estimation Perspectives, IEEE Software, pp.22-26, November/December 2000.

[4] M. Chrissis, M.Konrad and S. Shrum, CMMI: Guidilines for Process Integration and Product Improvement, Second Edition, Addison-Wesley, Nomber 2006.

[5] B. Curtis, M. Kellner, J. Over , " Process modeling ". Communications of the ACM 35, 9 (September 1992) 75-90.

[6] P. H. Feiler, W. S. Humphrey, "Software process development and enactment: Concepts and definition". Proceedings of the Second International Conference on Software Process (February 1993) 28-40.

[7] International Software Benchmarking Standards Group: www.isbsg.org

[8] F.Jensen, Bayesian Networks and Decision Graphs, Springer, 2002.

[9] M.I.Kellner, "Representation formalisms for Software Process Modeling", 4th International Software Process Workshop, Devon, UK 1989.

[10] R.G Pressman, "Software engineering, A Practitioner's Approach", McGraw-Hill Publishing Company, UK, 2000.

[11] I.Stamelos, L.Angelis, P.Dimou, E.Sakellaris, On the use of Bayesian belief networks for the prediction of software productivity, Information and Software Technology 45 (2003) 51-60.