Software maintenance project delays prediction using Bayesian Networks

Ana C.V. de Melo [a,*], Adilson J. Sanchez [b]

[a] *University of São Paulo (USP), Department of Computer Science, Rua do Matão, 1010,
Cidade Universitária, 05508 090 São Paulo, SP, Brazil*
[b] *Banco Itaú, Av do Estado, 5533 – Mooca, 03105 000, São Paulo, SP, Brazil*

Abstract

Managing software maintenance is rarely a precise task due to uncertainties concerned with resources and services descriptions. Even when a well-established maintenance process is followed, the risk of delaying tasks remains if the new services are not precisely described or when resources change during process execution. Also, the delay of a task at an early process stage may represent a different delay at the end of the process, depending on complexity or services reliability requirements. This paper presents a knowledge-based representation (Bayesian Networks) for maintenance project delays based on specialists experience and a corresponding tool to help in managing software maintenance projects.
© 2006 Elsevier Ltd. All rights reserved.

Keywords: Bayesian Belief Networks; Software management; Project delay estimation

1. Introduction

The advances in technologies and competitiveness of business products made computational support essential to have products in time and with a desirable quality to the market. Computational development became part of business strategy, instead of an auxiliary development support.

One of the reasons that makes computational support strategic to business products rests on the fact that business rules are continuously changing and this can be acquired by computational systems. Changing business rules and the corresponding system implies in adapting existing systems to accomplish new rules, the so called adaptive maintenance (Pressman, 2004). In this scenario, software maintenance process is essential to business products competitiveness to the market.

Despite of being widely studied and of interest to market, software maintenance is still a complex and costly task. Maintenance is pointed out as one of the most expensive tasks in software development and the difficulties inherent to its execution is the main cause of software high costs. Having no maintenance plans or, even when plans are established, having the plan schedule not achieved lead to cost-ineffective maintenance projects (Pressman, 2004).

That situation, however, does not come from no effort in trying to amend maintenance problems. Many people have employed a lot of work in developing new software development and management techniques to make software maintenance cost-effective. Even so, uncertainties in the maintenance process drives to unpredictable situations hard to be recovered during project execution (Ziv, 1997; Ziv, Richardson, & Klosch, 1996). Ordinary uncertainty situations in maintenance processes occur, such as:

- Maintenance plans represent only starting and end time of tasks, with no procedures to be applied as tasks are delayed in the middle of the project execution.

---

* Corresponding author. Tel.: +55 11 30919689; fax: +55 11 30916134.
E-mail addresses: acvm@ime.usp.br (A.C.V. de Melo), Adilson.Sanchez@itau.com.br (A.J. Sanchez).

- In general, information related to system restrictions and services are treated in different ways. Users want as many services as possible, even when they are not necessary to them. On the other hand, developers have to implement all services without a precise idea of their priorities.
- Time to complete tasks in the maintenance schedule is planned accordingly to a set of resources. Such resources may change through project development (the main project analyst is no longer with the company, for example), causing a delay in certain tasks.

Due to these uncertainties and others, project managers have to decide how to reconfigure the project plan as its schedule fails. One of the problems in replanning a project to achieve its schedule and services requirements rests on systems development information. What to do to finish the system in time? How much resource should be added in order to finish a delayed project in time? Can we implement the system in a shorter time without dropping its quality requirements? These and many other issues must be answered when a project is replanned and, in general, managers use their own experience to do that. Such an approach is very human-intensive and based on particular experiences – too subjective. A more objective approach can be supported by a decision system if experience of specialists and historical data of past projects are embedded in such a system.

Current project risk management processes have given a restricted focus on management of project uncertainties. The importance of treating uncertainties in project developments has already been recognised in the fields of project management (Chapman & Ward, 2000; Green, 2001; Ward & Chapman, 2003) and software development (Khodakarami, Fenton, & Neil, in press; Ziv et al., 1996). In both areas, uncertainties are pointed out as one of the causes of having delayed, unsuccessful and low-quality projects. In all those works, treating uncertainties and their main causes during project execution has been suggested as a way of having successful projects.

Several tools and methods have been proposed in order to support expert judgement. To produce such tools, a kind of knowledge about domains must be embedded, and Bayesian Belief Networks (Jensen, 1996; Pearl & Russell, 2003; van der Gaag, 1996) (Bayesian Networks for short) appear as a reasonable representation of uncertainties and knowledge. Roughly speaking, Bayesian Networks are cause-effect graphs, based on Bayes inference, that can model uncertainty and have been successfully deployed in certain domains such as medicine and politics. More recently, Bayesian Networks have also been applied to deal with uncertainties in software development. Stamelos, Angelis, Dimou, and Sakellaris (2003) show the construction of Bayesian Networks regarding technical and human factors in software development to estimate software productivity, by explicitly modelling uncertainties of factors influencing the productivity. Fan

and Yu (2004) present a scheme to incorporate Bayesian Networks in software project risk management to predict potential risks, identify sources of risks and advise dynamic resource allocation. Fenton, Krause, and Neil (2002), Fenton et al. (2004) and Fenton et al. (2007) show how quality metrics incorporated to predictive models (Bayesian Networks) can provide accurate prediction of software defects in a variety of development processes. Bayesian Networks have also been used to estimate costs of object oriented software, as in Bibi and Stamelos (2004). There, a set of Bayesian Networks are defined, concerned with the Rational Unified Process (RUP), to depict software process modelling along with software effort estimation. In Khodakarami et al. (in press), the application of Bayesian Networks enables the incorporation of risk, uncertainty and causality in project planning to identify specific risks and indicate their corresponding resources. Once resources of risk are identified, replanning projects (removing damaging resources) during execution can be performed. Ziv (1997) proposes a Bayesian Network to deal with the uncertainties of software testing. In the present work, a technique to support the prediction of software maintenance process delay during project execution is presented in order to help managers in replanning projects.

To help in managing maintenance project delays, this paper presents a technique to represent specialists knowledge about project development regarding delays. The main goal is to be able to calculate the project delay probability during its execution. Managers can use this information to replan a project in order to finish it in time. The forthcoming sections are as follows: Section 2 summarises Bayesian Belief Networks; Section 3 presents a particular Bayesian Network for the delay propagation problem; Section 5 gives guidelines of a tool to help managers to deal with maintenance project delays based on the Bayesian Network of the previous section; and the last section is about limitations of the present work and the effective use of this technique in software development.

2. Decision systems and Bayesian Networks

Decision systems are concerned with a way of embedding knowledge in order to make a mimic of human decisions (Jackson, 1990; Lucas & van der Gaag, 1991; Russell & Norvig, 2003). This involves how to represent knowledge and use such an information to make decisions. Those systems, the so called *expert systems*, are for solving problems of particular domains and the knowledge embedded in each domain is acquired from both domain specialists and systems development information. Expert systems have embedded a knowledge base and techniques to obtain (infer) new knowledge, or take decisions, from the existing base. One of the challenges in experts systems is how to acquire and represent knowledge as accurate as possible. If this is the case, the system can take decisions as close as a domain specialist.