Chapter 16
Software Process Model using Dynamic Bayesian Networks

Thomas Schulz  
Robert Bosch GmbH, Germany

Łukasz Radliński  
University of Szczecin, Poland

Thomas Gorges  
Robert Bosch GmbH, Germany

Wolfgang Rosenstiel  
University of Tübingen, Germany

ABSTRACT

This chapter describes a methodology to support the management of large scale software projects in optimizing product correction effort versus initial development costs over time. The Software Process Model (SPM) represents the implementation of this approach on a level of detail explicitly developed to meet project manager’s demands. The underlying technique used in this approach is based on Dynamic Bayesian Networks (DBNs). The use of Bayesian Networks (BNs) enables the representation of causal relationships among process and product key performance indicators elicited either by data or expert knowledge. DBNs provide an insight into various aspects of SPM over time to assess current as well as predicting future model states. The objective of this chapter is to describe the practical approach to establish SPM as state of the art decision support in an industrial environment.

INTRODUCTION

The most common way inserting defects into a software product is by applying changes to it. In every software release, hundreds of changes are applied to the software product. Every change has its unique characteristic based on process and product factors. With regard to these factors, every change has its own probability of injecting a defect. This chapter presents a methodology to estimate the amount of defects for a specific software product. It includes a goal oriented model design, detailed description of the process measurement framework including corresponding data mining.

DOI: 10.4018/978-1-60960-509-4.ch016
Software Process Model using Dynamic Bayesian Networks

concepts. The approach is based on process and product data combined with expert knowledge from one of the leading companies in the field of embedded applications in automotive industries. The resulting model is called the *Software Process Model* (SPM). SPM is built based on an extended version of the software process V-Model which itself is derived from the waterfall model. Process and product measurement frameworks combined with the knowledge of experts describe every phase of the V-Model and hereby enable the representation as a consistent model on the complete engineering process. SPM is mainly designed to optimize product quality and development effort over time. It enables to assess measures of the process influencing product quality. Due to the dynamic capabilities of the model, it further allows to analyze the impact of possible process and product changes over time. These capabilities give various views to project managers on the current and future software product that are discussed at the end of the chapter to highlight the practical benefit of the model. Details on automotive software engineering are part of the background section.

SPM is a Dynamic Bayesian Network (DBN) with capabilities to represent multiple change characteristics of a specific software release. It enables project managers to oversee the impact of every change made within this software release taking these individual change characteristics into account. Bayesian Networks (BNs), the base element for every DBN, are part of modern techniques of artificial intelligence (AI). Although a fundamental theorem on conditional probability was introduced by *The Reverend Thomas Bayes* back in XVIII century (Bayes, 1763), the term “Bayesian network” and its concepts were introduced in the 1980’s in pioneering work of Judea Pearl (Pearl, 1985; Pearl, 1988). Experienced researchers have recently put strong statements in the review of one of the books (Darwiche, 2009) on BNs: “Bayesian Networks are as important to AI Artificial Intelligence (AI) and Machine Learning as Boolean circuits are to computer science”, “Bayesian networks have revolutionized AI”. A BN can be perceived from two perspectives: (1) as a graph where the nodes represent random variables and the arcs represent the relationships between variables; (2) as a formal mathematical model definition where variables are expressed as conditional probability distributions. A DBN is a model forming a sequence of individual BNs where each individual BN reflects the state of the system at a specific point of time. BNs can be built by applying a learning algorithm to the dataset that enables learning a structure, parameters or both. The greatest potential of BNs lies in the ability to combine expert knowledge with empirical data. Other benefits of BNs include the ability to reflect causal relationships, to run the model with incomplete data, to forward and backward reasoning, explicit modeling of uncertainty as well as combining quantitative and qualitative data. An introduction to BNs and DBNs is given as part of the background section.

SPM’s structure and data as well as determination of usability, based on expert knowledge, are based on a certified industrial engineering process, focusing on high quality products. Databases on historical and current projects are used to validate the results. Process and project experts have assessed the resulting model for predictive accuracy and usability. All details on how SPM is built are part of the software process model section.

**BACKGROUND**

**Automotive Software Engineering**

The demand for software products in the automotive industry has been exponentially increasing over the last decades, making software engineering one of its critical success factors (Lee, 2007). The intensive use of *electronic control units* (ECUs) in vehicles today has already lead to an average of 20 ECUs in smaller and even more than 70
Related Content

Architecting Virtual Reality Systems
www.igi-global.com/chapter/architecting-virtual-reality-systems/8238?camid=4v1a

www.igi-global.com/chapter/estimation-and-evaluation-of-software-quality-at-a-particular-stage-of-software-development/189685?camid=4v1a

A Survey of Web Services Provision
An Liu, Hai Liu, Baoping Lin, Liusheng Huang, Naijie Gu and Qing Li (2010). International Journal of Systems and Service-Oriented Engineering (pp. 26-45).
www.igi-global.com/article/survey-web-services-provision/39097?camid=4v1a

Attribute Decoration of Attack–Defense Trees
www.igi-global.com/article/attribute-decoration-attack-defense-trees/66406?camid=4v1a