Int J Software Informatics, Volume 5, Issue 3 (2011), pp. 475–485
International Journal of Software and Informatics, ISSN 1673-7288

E-mail: ijsi@iscas.ac.cn
http://www.ijsi.org
Tel: +86-10-62661040

Environment Support for Improving Software Development Processes: A Vision Influenced by the Work of Barry W. Boehm

Lori A. Clarke

(Department of Computer Science, University of Massachusetts, USA)

Abstract Throughout his career, Barry Boehm has advocated the importance of understanding software development processes, measuring their performance, and using those measurements to guide the development of improved process models. In this paper, we describe PIE, a Process Improvement Environment, which supports that vision. PIE supports the definition of process models that can be analyzed and executed. The analysis is used to detect errors and vulnerabilities in the process models. Validated process models can then be simulated to detect inefficiencies and bottlenecks. Future work includes executing these process models, monitoring their performance, and then using that information to drive further process improvements.

Key words: software development processes; process improvement; process modeling; process analysis; requirement properties; model checking; fault-tree analysis; failure modes and effects analysis; human-intensive systems

Clarke LA. Environment support for improving software development processes: A vision influenced by the work of Barry W Boehm. *Int J Software Informatics*, **Vol.5, No.3 (2011): 475–485. http://www.ijsi.org/1673-7288/5/i95.htm**

1 Introduction

This paper describes environment support for improving software development processes. Much of this work can be traced back to and has built upon contributions made by Barry Boehm over the course of his career. Even in his early papers, Boehm argued for creating a model of the process under which software would be developed (e.g., [2]). He advocated that the process model to be followed in developing a software system was one of the first artifacts that needed to be explicitly articulated and agreed upon by all the stakeholders. Much of Boehm's career has been devoted to evaluating these process models. From the early waterfall model[25], to the various military standards that encoded this model in doctrine (e.g., [14]), to the more free-wheeling agile methods (e.g., [11, 19, 26]), and to the more thoughtful, risk-based approaches that Boehm has proposed, particularly the Spiral Model[4]

This material is based upon work supported by the National Science Foundation under Awards CCF-0820198, CCF-0905530 and IIS-0705772. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author and do not necessarily reflect the views of the NSF.
Corresponding author: Lori A. Clarke, Email: clarke@cs.umass.edu
Paper received on 2011-03-07; Revised paper received on 2011-03-28; Paper accepted on 2011-03-29; Published online 2011-04-08.

and the Incremental Commitment Model[6], he has undertaken careful assessments of these processes and, based on his experimental findings, reported on their strengths and weaknesses. Working with other pioneers (e.g., [5]), Boehm has argued for a scientific basis for evaluating and improving software development processes, and thus software development.

In this paper, we describe PIE, a Process Improvement Environment, being developed at the University of Massachusetts. This environment provides capabilities for modeling processes, for analyzing these processes for errors and vulnerabilities, and for providing execution and simulation support. We believe that this approach is a step toward supporting the vision that Boehm has consistently advocated throughout his career for explicitly understanding the processes that are being applied, for measuring their actual performance, and for using that information to develop improved process models.

2 Overview of the PIE

The Laboratory of Advanced Software Engineering (LASER) has been developing and evaluating technology to support the continuous improvement of processes. Specifically we have developed a modeling language, called Little-JIL[7], and a suite of analysis tools for evaluating Little-JIL process definitions. We have applied these technologies to various domains including software development environments[20], government processes, such as the conduct of elections[28] and on-line dispute resolution[17], and several life-critical medical processes, such as the administration of chemotherapy[9] and the transfusion of blood[16].

Creating an accurate and detailed process model is a labor-intensive activity that demands considerable time and effort. Our work has demonstrated, however, that this investment can be leveraged by applying a range of analyzers to the process models to uncover an assortment of errors, vulnerabilities, and inefficiencies. Figure 1 depicts how these analyzers are applied in PIE. The leftmost column in this figure shows the tools used to create a process definition (the Little-JIL editor) and to create properties (the PROPEL property elicitor[10;29]) that specify the requirements that are suppose to be upheld by this process definition. The second column shows the resulting process definition and property specifications and other information needed to support the analysis capabilities, which are shown in the third column. The fourth column shows the outputs from these analysis tools.

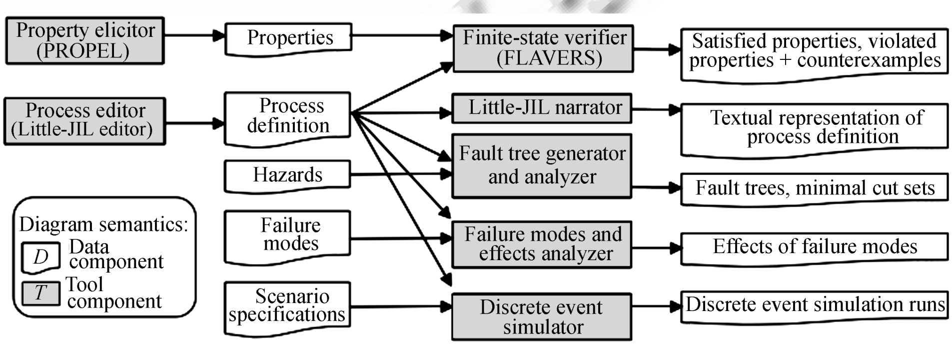

Figure 1. Architecture of the process improvement environment