

Keynote Address III

A System for Evaluating the Congruence of Software Process Models

Nazim H. Madhavji
School of Computer Science
McGill University
3480 University Street
Montreal, Quebec, CANADA H3A 2A7
email: madhavji@opus.cs.mcgill.ca
web: <http://www-se.cs.mcgill.ca>

Abstract:

It is increasingly being recognized that the design, customization, reuse and improvement of software processes are important for developing quality software within budget and on time. There is an urgent need for methods and tools to assist in these tasks. In this talk, we describe one of several works at McGill — a software system to address these needs.

The system uses an empirically derived “contingency model” that describes the degree to which the process/context relationships contribute to the success of a process. It uses this model along with the characteristics of a subject software process and its environment to: (i) evaluate how fit (or congruent) the process is in the given environment, and (ii) identify those relationships that are considered to lower process congruency.

Based on the feedback received from the system, the process engineer can change the values of appropriate process/context variables in order to improve process congruency.

We demonstrate the usage of the system by giving examples from three applications: (a) congruence evaluation and improvement of a process model, (b) selection of the fittest process model from among several models, and (c) process reuse. The data used in the examples was gathered in a field study of fifteen software projects, with particular focus on requirements engineering.

Acknowledgements:

This system was originally implemented by Graciela Perez, and recently re-implemented from scratch by Vivek Nanda, both as part of their Masters theses work. Khaled El Emam was involved in the research on contingency model as part of his Ph.D. thesis work.

Reference:

G. Perez, K. El Emam and N.H. Madhavji: Customising Software Process Models, Proceedings of the 4th European Workshop on Software Process Technology, Springer-Verlag, Leiden, Holland, pp. 70-78, April 1995.