Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 302 (2014) 95–109

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Semi-automated Tool Recommender for Software Development Processes

Marina Pilar, Jocelyn Simmonds and Hernán Astudillo

Departamento de Informática
Universidad Técnica Federico Santa María
Santiago, Chile
Email: {mpilar,jsimmond,hernan}@inf.utfsm.cl

Abstract

Application life-cycle management (ALM) tools are key for streamlining software development processes. However, small and medium development companies (SMBs) cannot afford to carry out time- and people-intensive tool evaluations for each project, and instead adopt fixed toolsets, thus losing flexibility. To simplify the tool selection process, this article proposes formalizing tool selection as a set of Multiple-Criteria Decision-Making (MCDM) problem, one for each ALM domain. Our domain-parametric recommender takes as inputs a domain, a process definition, and a set of tool evaluation criteria, and yields a ranked list of tools. The approach has been prototyped with the Testing domain and evaluated using a real process and project; the recommendations generated by our approach were quite similar to those of three Testing experts. Pending further evaluation, these results suggest that our approach can generate project-specific tool recommendations with results comparable to those of experts, but at a fraction of the cost.

Keywords: software development process; taxonomy; testing tools; multi-criteria decision-making

1 Introduction

The increased availability of IT tools supporting a wide range of activities in areas like government and health-care, has created a new problem for these organizations: they must now evaluate and compare an ever-growing set of tools – commercial, academic and open source – in order to determine which ones better suit their needs [7]. This issue is even more critical at software development companies, which have to take into account more criteria when evaluating tools, like their current technological ecosystem, tool integration capabilities, ease of use, user training, etc.; but most importantly, the tool must meet the needs of the project and/or organization.

This has made tool selection a complex and expensive task, one to which Small and Medium Businesses (SMBs) can only allocate limited resources to. As a result, these companies choose tools without much prior research [21, 26], and eventually, these tools either fail to meet the needs of the project and/or the development team

http://dx.doi.org/10.1016/j.entcs.2014.01.022

finds them too difficult or cumbersome to use. Others make ad-hoc use of non-specific tools like MS Office, for example, using a spreadsheet for bug-tracking, but this approach does not usually scale well.

One way to improve this process is to create and maintain a tool catalog that can be used by teams when deciding which tools to evaluate for future use. However, since many different criteria must be taken into account when deciding which tools to evaluate, and it is not clear (a-priori) if these criteria interact/interfere with each other, this approach does not completely solve the problem.

Thus, we propose using Multi-Criteria Decision-Making (MCDM) techniques [5] to semi-automate the tool recommendation process. Our framework takes as input a team's development process and their preference levels for various tool selection criteria and uses MCDM to produce a ranked list of tools that support the input process' tasks. These tools are selected from an extensible tool catalog, which is built on top of a set of tool and task taxonomies, one pair for each ALM domain.

The advantage of this approach is that any company that has formalized its software development processes can easily filter through a large amount of tools quite quickly (using a reduced set of criteria). Moreover, if a company evolves or tailors their development process [9], it is easy to check whether the same tools are recommended for the new process. Another advantage of this approach is that the tool and task taxonomies can be built incrementally.

In this article, we describe our framework as applied to the Testing domain. We have validated our prototype by using it to recommend tools for a real, previously documented process and project; the recommendations we obtained using our prototype were quite similar to those of three Testing experts. This article makes the following contributions: (1) We propose a domain-parametric, semi-automated tool recommendation framework that takes into account the project context and development process; (2) We have developed a testing tool catalog and its corresponding taxonomies; (3) We pose tool recommendation as a MCDM problem, which allows us to control the tool recommendation process through the specification of criteria preference levels.

The rest of this article is organized as follows. We give an overview MCDM and our approach in Sections 2 and 3. The Testing domain taxonomies are described in Section 4, and our expert study is presented in Section 5. After comparing our work with related approaches in Section 6, we conclude in Section 7 with a summary of the article and suggestions for future work.

2 Multicriteria Decision-Making

Decision making has become a mathematical science, where the various aspects involved in the decision making process have been formalized [5]. The key aspects in the decision making process are the problem definition, determining minimum requirements, specifying goals, defining selection criteria (tangible or intangible), as well as identifying possible alternatives. This process requires a significant amount of time, and we hope to reduce the amount of input required from the user. For