Dynamic Situational Adaptation of a Requirements Engineering Process

Graciela Dora Susana Hadad
Universidad Nacional del Oeste, Argentina & Universidad de Belgrano, Argentina

Jorge Horacio Doorn
Universidad Nacional del Oeste, Argentina & Universidad Nacional de La Matanza, Argentina

Viviana Alejandra Ledesma
Universidad Nacional del Oeste, Argentina & Universidad Nacional de La Matanza. Argentina

INTRODUCTION

The adaptation of any process to particular situations is considered a good practice in most fields. Literature shows that this practice is quite common in Software Engineering processes, such as the methodologies Rational Method Composer (Haumer, 2005) and OPEN Process Framework (Firesmith &Henderson-Sellers, 2002). However, Requirements Engineering (RE) approaches are seldom tailored to context or project situations (Potts, 1995; Leite, Hadad, Doorn, & Kaplan, 2000; Leffingwell & Widrig, 2003; Seyff et al., 2009). Nevertheless, sometimes the elicitation activity, as part of an RE process, is performed taking into account some environmental characteristics, such as number of information sources, user geographical distribution, user time availability, user experience, among others (Maiden & Rugg, 1996; Hickey & Davis, 2003; Coulin, 2007; Carrizo, Dieste, & Juristo, 2008). Recently, some proposals have appeared to design an RE process for a specific project by selecting existent RE techniques (Lauesen, 2002; Lobo & Arthur, 2005; Alexander & Beus-Dukic, 2009).

There are activities of the requirements process that are invariant regardless of situational factors, while others should be modified, removed or replaced. Not only activities may be adapted, models produced or used in the process may be also suited for the situation (Galster, Weyns, Tofan, Michalik & Avgeriou, 2014). This means that the RE process may be assembled like a flexible puzzle using interchangeable pieces depending on the situational factors identified.

Situational Method Engineering (SME) is advocated to build methods tailored to specific situations for the development of systems (Kumar & Welke, 1992). Following its principles, the adaptation of any software development process is based on indicators describing the situation (Khan, bin Mahrin & bt Chuprat, 2014). Part of the task is to compose such indicators based on observable factors, like degree of business processes reengineering, context complexity, developer expertise in the application domain, and project size, among others. Ideally, these situational factors should be taken into account before beginning the software process. However, there are factors not accurately known when initiating a software project, while other factors may change during the course of the project. Hence, a dynamic view of the adaptation of a software development process achieves a better effectiveness of the process itself. Considering that the production of requirements is the starting point of a software development, it should be necessary to pay more attention to factors impacting on the RE process.

A frequent question of practitioners is related with the obligation of performing all the process steps to reach requirements. *Is it possible to shorten the road or to follow a different one?* Under some

DOI: 10.4018/978-1-5225-2255-3.ch646

circumstances, there is an opportunity to reduce the RE process by deleting or simplifying activities; and sometimes different paths may be followed by choosing other techniques or even extending some activities. Project managers should make decisions depending mainly on the particular case.

Since problem domain knowledge is mostly expressed in natural language (NL), the use of an RE approach based on NL representations improves the commitment of customers and users to the project, increasing the probability of project success (Macaulay, 1993).

Therefore, in this chapter it is presented the tailoring of an RE process based on NL models, according to a particular set of situational factors. Additionally, a process for constructing this RE process, including the evaluation of such factors along software development life cycle, is proposed as an enhanced solution.

BACKGROUND

When working on the creation of an engineering product or system, it is important to have a process. This means having a predictable set of activities, techniques, inputs and outputs helps get a high quality outcome. Hence, the way the work is performed does not depend on individual criteria, allowing repeatability of costs, times and quality, and promoting the accumulation of knowledge about the process. As a consequence, the first activity of a process to develop a product consists in defining precisely the expected outcome. When the product is a software system, this initial activity is an RE process, whose outcome is a Software Requirements Specification (SRS). The RE process is particularly different from other activities of the software development process since it is the one that most interacts with people and their environment, while other activities are mainly carried out within the development team context (Carrizo, 2009). Besides, project decisions impose constraints, tools and methods to carry out those activities. Therefore, if the requirements process takes into account the particularities surrounding the application context and the project itself, then it will probably result both in a better SRS and in a more effective process.

Additionally, a requirements process needs appropriate and continuous communication to gain customers and users compromise. Good communication is achieved when all stakeholders use the same language. In RE, a proven way to accomplish this is by using the vocabulary of the application context (Leite, Doorn, Kaplan, Hadad, & Ridao, 2004). Communication occurs when stakeholders orally interact, and also when reports, documents and models are exhibited to customers. In this sense, NL models, such as glossaries, use cases and scenarios, stimulate stakeholders communication (Leite et al., 2004), and are the most frequently used in RE (Kaindl, 2000; Leffingwell & Widrig, 2003; Seyff et al., 2009; Antonelli, Rossi, Leite, & Oliveros, 2012).

Software development processes put into practice in real projects are often forced to be adjusted due to contingent circumstances, sometimes in a poorly controlled fashion while the project is ongoing. Hence, the evolving situation should be observed as the process goes forward in order to achieve a better tailoring. Possible adjustments to the process can be known in advance on the basis of certain characteristics, though they may change dynamically, i.e., settings are pre-planned but only implemented when an aspect of the situation changes (Rolland, 2008). In this regard, the process is defined as a set of blocks, having process blocks common to all situations and variant process blocks according to situational factors. Thus, the process is made up by assembling blocks for the particular situation (Henderson-Sellers & Ralyté, 2010). Method Engineering was specifically created to tackle this case, promoting the design, construction and adaptation of methods, techniques and tools in order to develop information systems (Brinkkemper, 1996). This discipline considers not only the creation of process blocks but also product blocks, and even blocks that assemble both process and product (Rolland, 2008; Henderson-Sellers &