Chapter 66
On Software Architecture Processes and Their Use in Practice

Perla Velasco-Elizondo
Autonomous University of Zacatecas, Mexico

Humberto Cervantes
Autonomous Metropolitan University, Mexico

ABSTRACT

Software architecture is a very important software artifact, as it describes a system's high-level structure and provides the basis for its development. Software architecture development is not a trivial task; to this end, a number of methods have been proposed to try to systematize their related processes to ensure predictability, repeatability, and high quality. In this chapter, the authors review some of these methods, discuss some specific problems that they believe complicate their adoption, and present one practical experience where the problems are addressed successfully.

1. INTRODUCTION

In recent years, software architecture has begun to permeate mainstream software development and, according to Shaw and Clements (Shaw & Clements, 2006), since the year 2000, architecture has entered a "popularization" period characterized by aspects such as increased attention to the role of the software architect and the introduction of software architecture processes into organizations. As part of this trend, a number of methods have appeared to try to systematize these processes to ensure predictability, repeatability, and high quality outcomes.

The software architecture of a software system is the structure (or structures) of this system, which comprises software elements, the externally visible properties of those elements, and the relationships among them (Clements et al., 2010). In this chapter, by software architecture development we refer to the activities that are typically performed early in a software development project, which contribute to creating the different structures that shape the architecture. Despite the availability of methods to support

DOI: 10.4018/978-1-5225-3923-0.ch066

the processes related to software architecture development, we consider that there is a set of specific problems that complicate the adoption of such methods in practice. A summary of these problems can be stated as follows:

1. **Selection of methods for the software architecture lifecycle:** Ideally, software architecture development should be carried out within the context of a software architecture lifecycle, which imposes a structure on the activities for developing it. Existing software architecture development methods typically focus only on a particular phase of the lifecycle and do not cover it completely. Thus, an appropriate combination of methods to cover the complete lifecycle must be chosen.
2. **Heterogeneity of the existing methods**: Many existing software architecture development methods have been defined by different authors "in isolation," i.e. independently of methods used in other lifecycle phases. This results in having them defined in terms of different activities, work products and terminology. This heterogeneity requires that, once a particular combination of methods is chosen, they must often be analyzed and modified to avoid mismatches, omissions or repetitions.
3. **No consideration of the software development process:** Software architecture development methods are typically defined independent of a particular software development process. Therefore, the introduction of architectural development methods into an organization often demands adapting both the organization development process and the architectural development methods to fit properly (Kazman, Nord, & Klein, 2003).
4. **Architectural design methods are decoupled from everyday practice:** To support the design of an architecture many methods use abstract concepts such as tactics and patterns. These concepts are frequently not the ones that software architects use the most in their day-to-day activities, as many architects tend to favor the selection of technologies such as software frameworks during design. Thus, it is necessary to find ways to include commonly used concepts into architectural design methods (Cervantes, Velasco-Elizondo, & Kazman, 2013).
5. **Difficulty of organizational deployment:** The introduction of architectural methods into an organization often involves costs related to process change, human resources training and technology investment. To promote the successful adoption of software architecture development methods in an organization it is necessary to follow a systematic deployment process.

Based on the problems listed above, in this chapter we propose some actions to address them and describe the observed benefits when implementing them in an industrial setting, specifically, in a large software development company in Mexico City, currently rated at CMMI-DEV level 5, which develops custom software for government and private customers.

This chapter is organized as follows. In Section 2, we introduce the notion of software architecture lifecycle and, within this context, review some well-known processes and methods to support it. Next, we discuss in more detail in section 3 the set of problems that we consider have complicated the adoption of these methods in practice. In Section 4, we describe a specific instance where these problems were addressed in practice. Section 5 presents a discussion. Finally, in the last section, we draw some conclusions and describe paths of future work.