Chapter V
Process Models of SDLCs: Comparison and Evolution

Laura C. Rodríguez
Autonomous University of Aguascalientes, Mexico

Manuel Mora
Autonomous University of Aguascalientes, Mexico

Miguel Vargas Martin
University of Ontario Institute of Technology, Canada

Rory O'Connor
Dublin City University, Ireland

Francisco Alvarez
Autonomous University of Aguascalientes, Mexico

ABSTRACT

The software engineering discipline has developed the concept of software process to guide development teams towards a high-quality end product to be delivered on-time and within the planned budget. Consequently, several software-systems development life-cycles (PM-SDLCs) have been theoretically formulated and empirically tested over the years. In this chapter, a conceptual research methodology is used to review the state of the art on the main PM-SDLCs formulated for software-intensive systems, with the aim to answer the following research questions: (a) What are the main characteristics that describe the PM-SDLCs?, (b) What are the common and unique characteristics of such PM-SDLCs?, and (c) What are the main benefits and limitations of PM-SDLCs from a viewpoint of a conceptual analysis? This research is motivated by a gap in the literature on comprehensive studies that describe and compare the main PM-SDLCs and organizes a view of the large variety of PM-SDLCs.

INTRODUCTION

In order for a product to be developed, a development (formal, semi-formal, or informal) process is required. For the specific case of software artifacts, a software (development) process is a method of producing such artifacts. This process is usually denoted as **the software-systems development life-cycle.** To guide its execution under different design conditions, a set of process models have been also proposed: **process model of systems development life cycles** (PM-SDLCs). In general, the aim of each single process is "to facilitate the engineer doing the job well rather than to prevent them from doing it badly" (Tyrrel, 2000).

In the software engineering discipline, the concept of a software *process* has been developed to guide the development team on constructing a high-quality end product that be delivered on-time and within the planned budget. Consequently, several PM-SDLCs have been theoretically formulated and empirically tested over the years, and in general many have been an evolution of previous models. In some cases, the evolution is originated as a result of a major advance in information and communications technologies (ICT), and in other cases, as a result of more planned changes in the organizations' settings and their business environments.

In this chapter, we use a conceptual research methodology (Glass, Vessey, & Ramesh, 2002; Mora, 2004) to review the state of the art on the main PM-SDLCs formulated for software-intensive systems, with the aim to answer the following research questions: (a) What are the main characteristics that describe the PM-SDLCs?, (b) What are the common and the unique characteristics of such PM-SDLCs?, and (c) What are the main benefits and limitations of PM-SDLCs from a viewpoint of a conceptual analysis?

The conceptual research approach is widely used in the software engineering domain (Glass et al., 2002). According to Cournellis' ideas (2000)—quoted by Mora (2004)—this research method studies concepts, ideas, or constructs on empirical objects. This chapter uses the research methodology process, described in Mora, 2004, that consists of the following phases: (1) formulation of the research problem; (2) analysis of related studies; (3) development of the conceptual artifact; and (4) validation of the conceptual artifact. The first phase and second phases are similar to other well-known research methods. In the third phase, two activities are conducted: the development of a general framework/model and the detailed development of this general framework/model. This third phase is a creativity-intensive process guided by the findings, contributions, and limitations found in the second phase and a set of preliminary proforms that are fixed through an iterative process (Andoh-Baidoo, White, & Kasper, 2004). Finally, in the last phase, the conceptual artifact's validation can be conducted using a panel of experts, a logical argument discourse, or/and a proof of concept developing a prototype or pilot survey. In this study, we used the first procedure with two internal academic experts and an expert practitioner in the development of SwE projects. Satisfactory average scores of 4.6 in a 5-point Likert scale of an instrument conceptual composed of eight items was achieved (Mora, 2004).

This research is motivated by the knowledge gap in the literature on comprehensive studies that describe and compare the main available PM-SDLCs. The research relevance can be considered high because the main objective of software engineering is the development of high-quality, on-time, and within budget software projects, which can only be delivered with the utilization of a systematic development process, as has been proven in other engineering disciplines. Therefore, this study contributes to organize the diverse and partial views of PM-SDLCs.

BACKGROUND

Software engineering, according to the *IEEE Standard Computer Dictionary* (1990) is the: " (1) Application of quantifiable approach, disciplined to the software development, operation and maintenance;