Chapter 5
The Incremental Commitment Spiral Model for Service-Intensive Projects

Supannika Koolmanojwong
University of Southern California, USA

Barry Boehm
University of Southern California, USA

Jo Ann Lane
University of Southern California, USA

ABSTRACT

To provide better service to customers and remain competitive in the business environment, a wide variety of ready-to-use software and technologies are available for one to "grab and go" in order to build up software systems at a rapid pace. Currently, a wide variety of Web services are available and ready to use for this purpose. Current software process models also support commercial-off-the-shelf (COTS)-based development processes. However, although COTS and Web Services are similar, they are different in many perspectives. On one hand, there are various software process models that support Web services development. Yet there is no process model that supports the project that uses services provided by others. This chapter introduces the Incremental Commitment Spiral Model (ICSM), a new generation process model that provides development guidelines, from exploring a Web service alternative to deployment and maintenance with case studies.

DOI: 10.4018/978-1-4666-2503-7.ch005

INTRODUCTION

The expeditious and fast growing of Internet and Web applications create a variety of available services over the Internet. Services or Web Services or Software as a service are software functionalities that are available online to be accessed by users using a web browser over the internet. Approximately 80% of the world economy provides services in various forms (CMMI 2009). Each day, there are roughly 3 new mashups or Web service extensions created and listed at programmableweb.com (Programmable 2009). The same increasing trend can be seen through the increasing demand for Web services usage in real-client software development projects for the graduate level software engineering course at University of Southern California (CSCI577ab 2011). New opportunities, such as using mashups or services, also come with new risks. The process user should not follow the same set of activities or use the default process model when the project has a new risk pattern (Boehm 2008). As a result, the Web services consumers need to know how to select the appropriate service and utilize the service properly. Various process models such as CMMI-SVC (CMMI 2009) provide great support for a Web-services development project and CMMI-COTS (Tyson et.al 2003) supports the COTS-based development. However, there is no process model for the software development project that uses services provided by others or service-intensive development. Although COTS and web services are both non-developmental items (NDI) and they are similar in numerous aspects, they are fairly different in detail. Moreover, based on our preliminary study (Koolmanojwong 10), a pure COTS-based development process does not fit well for services-intensive project. Hence, there is a need for a process model that could specifically support services-intensive or services-based development projects.

The ICSM (Boehm 1996) (Boehm and Lane 2007) is a new generation process model. ICSM covers the full system development life cycle, consisting of an exploration phase, valuation phase, foundations phase, development phase, and operations phase. It has been evaluated to be a reasonably robust framework for systems development. One of the main focuses of the ICSM is feasibility analysis; evidence must be provided by the developer and validated by independent experts. The ICSM combines the strengths of various current process models and limits their weaknesses. The ICSM, like the V-Model (V-Model 2009), emphasizes early verification and validation, but allows for multiple-incremental interpretation and alleviates sequential development. Compared to the Spiral Model (Boehm 1998), the ICSM also focuses on risk-driven activity prioritization, but offers an improvement by adding well-defined in-process milestones. While ICSM, RUP, and MBASE (Boehm 1996) perform concurrent engineering that stabilizes the process at anchor point milestones, ICSM also supports integrated hardware-software-human factors oriented development. Compared with Agile methods (Agile manifesto 2001), ICSM embraces adaptability to unexpected change perspective and concurrently allows scalability.

Based on the risk pattern, size, complexity, change rate, criticality, available NDI support, and organizational and personnel capability, projects could be categorized into eleven process patterns ranging from a very small scale project such as Use Single NDI to a very large scale multi-owner system of systems. This chapter focuses on one of the eleven process patterns called NDI-intensive process pattern by pinpointing to a services-intensive sub process pattern. Later, the chapter will layout the process decision framework and flow of activities in the service-intensive project development. Additionally, this chapter will discuss success case studies of e-services projects that follow services-intensive process pattern and how they improve project performance, especially in terms of faster-time-to-market and higher client satisfaction.