Chapter 16
Applying Software Engineering Design Principles to Agile Architecture

Chung-Yeung Pang
https://orcid.org/0000-0002-7925-4454
Seveco AG, Switzerland

ABSTRACT

Most enterprise IT systems are very complex with a combination of COBOL and Java programs running on multiple platforms. What is needed is a solid IT architecture that supports the operation and growth of a cross-platform IT system. It must enable the iterative and incremental development of applications that are foreseen in an agile development process. The design concept of such an architecture with its infrastructure and development tool is presented in this chapter. This design concept is based on the design principles and architectural patterns of software engineering. The architecture is a combination of layered, component-based, and service-oriented architectural patterns. The agile development process is based on a model-driven approach. The architecture and development approaches were first introduced in 2004. Since then, many applications have been developed on time and within budget.

INTRODUCTION

Despite advances in software engineering, software development remains a challenge for IT professionals. Statistics have shown that most software projects run too late and too expensive. The problem becomes even more apparent when a mix of legacy and modern applications in a complex IT system comes into play. In fact, IT enterprise systems typically go through a long development phase. Over the past decades, software projects have evolved into some of the most complex software systems. For large companies, mainframe applications programmed in COBOL often form the backbone of the IT structure. New applications and components are often developed using a language such as Java on a UNIX or LINUX platform. Often, business processes require collaboration between components on different platforms.

DOI: 10.4018/978-1-6684-3702-5.ch016

Maintaining and updating an IT system that combines legacy and modern platforms is one of the toughest challenges many companies are facing today. Modern businesses have high demands on IT systems to work in highly stable yet flexible and fast environments, and to introduce new features and processes to meet their steady growth. The agile software development process, which follows the evolutionary and iterative approach of software development and focuses on adapting to change (Ambler, 2010; Larman, 2003), seems to meet the requirements. However, the process alone does not guarantee the success of a software project. There are many other factors. Attempts to use only the agile approach to software development can still fail (Harlow, 2014; Ismail, 2017). Through years of evolution in software engineering, there are design principles and techniques that can help tackling the complexity involved in application development in an enterprise IT system. They must be incorporated into the agile development process.

A cross-platform enterprise IT system requires a software architecture that supports operations and growth. The architecture must enable the iterative and incremental development of applications. The purpose of this chapter is to introduce the design and implementation concept of such an architecture. The design concept is based on design principles and architectural patterns that have emerged from decades of research in software development. It is also based on years of hands-on experience of the author with an iterative incremental process and a continuous integration approach to software development.

The materials presented in this chapter focus on four areas: what, why, how and consequences. The first section covers the historical background of software engineering and the agile development process. The following section is the "what" topic that gives an introduction to the design principles that have resulted from software development and object-oriented programming, as well as the software architecture and agile development methodology. The next section discusses the "why" and explains the motivation behind the principles of software engineering, software architecture, and the agile development process. The section "how" develops agile architecture design and its use in the agile approach to enterprise application development. In the section "consequences" the applicability as well as the experiences from the practice are presented. The chapter ends with future research directions and conclusions.

BACKGROUND

As background, the following subsections introduce the disciplines of software engineering and the agile development approach.

Software Engineering

In the early days of software history, programmers tended to develop their programs without documentation in an ad-hoc style. The programs are usually not structured and organized. With the development with many new features, the underlying software becomes unmanageable and unmanageable. One result was the software crisis of the 1960s, 1970s and 1980s (Software Crisis, 2010).

Software engineering (2010) is a discipline that offers solutions to counteract the software crisis. It defines standards, disciplines, methodologies and processes for software development. In recent decades, many new programming languages, design principles and architectural patterns, and development paradigms have been developed. Programming styles such as structural programming (Yourdon & Constantine, 1979; Jackson, 1975), object-oriented programming (Booch, et al., 2007), etc. have been