Chapter 3
Ten Years of Experience with Agile and Model-Driven Software Development in a Legacy Platform

Chung-Yeung Pang
Seveco AG, Switzerland

ABSTRACT

In this chapter, a report containing the author's many years of experience in software development together with a discussion of software engineering are presented. The report begins with the software crisis and includes different projects following the traditional waterfall model with heavy documents. In a re-engineering project of a legacy IT system by modernizing COBOL applications, we established an agile and model driven approach to software development. This approach which has been successfully applied in 13 projects since 2004 is presented. The key factors required for our success will also be discussed. Both the good and bad experiences of the last ten years will be summarized. The chapter will be finalized with a vision of a new architecture for agile software development.

INTRODUCTION

Computer technology has evolved at a rapid rate. It was inconceivable that my current notebook has more computer power than the huge IBM 370 machine that ran my first program in my university days. It is a real joke when someone still uses hardware components from the good old days for today's business. On the other hand, what about software? Many COBOL programs developed in the 1970s are still in operation. In fact, enterprise IT systems usually undergo a long period of evolution. This decade long, the software projects evolved into some of the most complex software systems. For large corporations, mainframe applications programmed in COBOL often form the backbone of the IT structure. Despite their obsolescence, legacy systems continue to provide a competitive advantage through supporting unique business processes and containing invaluable knowledge and historical data.

DOI: 10.4018/978-1-5225-3422-8.ch003

Maintaining and upgrading legacy systems is one of the most difficult challenges many companies currently face. They struggle with the problem of modernizing these systems while keeping the day-to-day operation intact. As reported later in this chapter, many large corporations have tried but failed to re-build their legacy systems using object-oriented language like Java or Smalltalk. At the same time, new applications developed in Java do not seem to provide significant advantages in terms of performance or enhancement in business agility. On the other hand, with a proper approach to software development, one can build flexible, maintainable, agile applications in a legacy platform with a language like COBOL. The approach was first reported in 2012 (Cockburn & Highsmith, 2001; Pang, 2012) and its solution framework includes the following features:

- Modular and pluggable for architecture business and system component integration.
- Reuse in terms of design and code patterns.
- Use model driven approach and code generation in the development process.
- Test infrastructure for unit testing, component integration testing and service testing.
- Adapt agile development process.

The development approach is based on practical experience from a re-engineering project of a legacy IT system in a large corporation. It was first applied in 2004. During the last 10 years, the approach has been applied in 13 projects. Although some projects were under significant pressure with time and budget constraints, all projects were completed on time and within budget.

In this chapter, I report my many years of experience in software development, particularly the last ten years of using the development approach. The chapter is organized as follows. In the background section, my experience of the software crisis, software development process based on the waterfall model, challenges and success of software projects are elaborated. In the next section I discuss the agile development process with the agile manifesto and its critiques. Software architecture, which in my experience is crucial to the agile process for enterprise application development is also elaborated. In the following section the history and evolution of the agile architecture for our development approach are presented. The architecture, model driven approach, tools and framework are discussed. Following that is a section on project experience. Then a section on the experience with developers, development teams and managers is presented. Following is a section about overall experience and lessons learnt. The acceptance, positive and negative experience as well as the findings are presented in this section. The chapter is finalized with future research directions and conclusions.

BACKGROUND

As background I present my personal experience with the software crisis, various software development processes, as well as challenges and successes of software projects.

Software Crisis

In the early days of software history, programmers tended to develop their programs in an ad hoc style with no documentation. A result of this was the software crisis of the 1960s, 1970s and 1980s. Typical phenomena of the software crisis are (Crisis, 2010):