Chapter 72
Reuse in Agile Development Process

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

ABSTRACT

Reusability is a clear principle in software development. However, systematic reuse of software elements is not common in most organizations. Application programmers rarely design and create software elements for possible future reuse. In many agile software development processes, the project teams believe that the development of reusable software elements can slow down the project. This can be a misconception. This chapter examines various ways to reuse software. Three approaches to developing reusable software artifacts from 15 years of experience in the agile development process are presented. The first approach is to create generic programs or configurable frameworks that support similar solutions for a variety of use cases and environments. The reuse of patterns is the second approach presented. Another effective way is to use a model-driven approach with model patterns. These approaches help to speed deployment software. The final product is flexible and can easily be adapted to changes. This is one of the main goals of an agile approach.

INTRODUCTION

An experienced programmer would reuse his / her code written for another project whenever a similar function is needed. In fact, people always use the same solution for similar problems. That is why experience can be very valuable. When implementing a new module, a veteran programmer, after years of observation by the author, often cuts out code fragments from old modules that he / she has written and inserts them into the new module. He / she is usually more efficient and productive as he / she already has experience solving similar problems. Transforming this experience into a set of reusable elements that anyone can use would bring great benefits to overall software development. That's what software reuse is all about.

DOI: 10.4018/978-1-6684-3702-5.ch072

With advances of software technologies, there are many packages and libraries containing reusable software elements available commercially or open sources. Different business or system units from large organizations usually provide business domain specific or infrastructure related software components and functions (e.g. a function to retrieve customer data, etc.) for application development. Using these software elements in application development forms the common practice of reuse. With decades of experience working for many large organizations, the author finds that it is not a common practice for application programmers to design and build software elements for possible future. When developers have to meet deadlines and provide functionality, building software elements that can be reused is never a priority. In fact, often one can find programs with hard coded values and algorithm that prevent them from being reused in different context without code changes.

In recent years, the agile software development process has become very popular in the software industry (Ambler, 2010; Larman, 2003). In this approach, the emphasis is on providing working software with the highest business values as a measure of the progress of a software project. It prefers functioning software over documentation and customer collaboration over contract negotiation. As a result, so many developers are simply write ad hoc-style code, even though the agile approach does not exclude the value of analysis and design. For many agile development teams, designing and building reusable software elements are out of the question.

Although agile development practice has great advantages in software development, many projects are still failing (Harlow, 2014; Ismail, 2017). When a software element has high reuse potential through configuration and extension, it is flexible and easily adapt to changes. This actually fulfil the motivation of agile software approach that the end system should be flexible and easy to change. The purpose of this chapter to describe the techniques to design and build reusable software artefacts and demonstrate that they bring many benefits. The development process to provide working software can be speeded up. The end product is flexible and easily adaptable to changes, which is one of the main goals in using an agile approach.

The chapter gives a brief survey of software reuse. It presents the general reusable software artefacts with their motivation, implementation concept, consequences and applicability. Specific approaches to develop reusable software artefacts that evolved out of 15 years of experience of agile development practice are described. The approaches include those for "generic programming and configurable framework", "pattern based reuse" and "model driven approach". The chapter is organized with a first section on the background of agile software development and reusing software. It is followed by a section on reusable software elements. The three approaches mentioned before will be described in three different sections. The chapter ends with sections on the vision for the future and a final conclusion.

BACKGROUND

In the background, the first subsection deals with the history of software development and the agile approach. Following is a subsection on reuse based software engineering.

History of Software Development and Agile Approach

In the early days of software history, programmers tended to develop their programs without documentation in an ad-hoc style. As software systems evolved with features over the years, they were no longer