Chapter XVII
Agile Method Fragments and Construction Validation

Q.N.N. Tran
University of Technology, Sydney, Australia

B. Henderson-Sellers
University of Technology, Sydney, Australia

I. Hawryszkiewycz
University of Technology, Sydney, Australia

ABSTRACT

Method fragments for work units and workflows are identified for the support of agile methodologies. Using one such situational method engineering approach, the OPEN Process Framework, we show how the full set of these newly identified agile method fragments, each created from the relevant powertype pattern as standardized in the Australian Standard methodology metamodel of AS 4651, can be used to recreate four of the currently available agile methods: XP, Scrum, and two members of the Crystal family—thus providing an initial validation of the approach and the specifically proposed method fragments for agile software development.

INTRODUCTION

Situational method engineering (Welke & Kumar, 1991) is the subdiscipline of software engineering in which methodologies (a.k.a. methods) are envisaged as being constructed from parts called method fragments. These are identified from best practice and stored in a repository (or method base). In this chapter, we identify new fragments to support the work units and work flows needed to support agile software development—a set of method fragments that completes and complements those described in Tran, Henderson-Sellers, and Hawryszkiewycz (2007). These fragments are compliant with the metamodel of the Australian Standard AS4651 and extend the existing repository of the OPEN Process Framework (OPF: Firesmith & Henderson-Sellers, 2002; http://www.opfro.org), chosen on the basis of it having the most extensive content in its method base. Following this discussion of new fragments, we then use them to recreate a number of existing agile methods as validation of both the SME ap-

proach and the particular set of newly proposed agile method fragments.

AGILE SOFTWARE DEVELOPMENT AND THE AS4651 METAMODEL

As discussed in Tran et al. (2007), agile development adheres to the following fundamental values (Agile Manifesto, 2001):

- **Individuals and interactions** should be more important than processes and tools.
- **Working software** should be more important than comprehensive documentation.
- **Customer collaboration** should be more important than contract negotiation.
- **Responding to change** should be more important than following a plan and has a strong focus on teamwork.

Here, we use the basic understanding of agile software development to identify method fragments compatible with the OPEN Process Framework and the metamodel described in AS4651 (Standards Australia, 2004). The overall architecture of this metamodel is shown in Figure 1, using the notion of a powertype (Odell, 1994) (for full details see Tran et al., 2007).

We will now describe in more detail the metaclasses that are relevant to agile fragments but which were not covered in Tran et al. (2007), that is, work units (tasks and techniques) and workflows (a third kind of work unit)—the focus of this chapter.

WorkUnit-Related Metaclasses

A WorkUnit is a job performed within a project. A WorkUnitKind is a specific kind of work unit, characterized by its purpose within the project. It is specialized into TaskKind, TechniqueKind, and WorkFlowKind.

A task is a small-grained work unit that focuses on what must be done in order to achieve a given purpose. A TaskKind is a specific kind of task, characterized by its purpose within the project.

A technique is a small-grained work unit that focuses on how the given purpose may be achieved. A TechniqueKind is a specific kind of technique, characterized by its purpose within the project.

WorkFlow-Related Metaclasses

A WorkFlow is a large-grained work unit that operates within a given area of expertise. A WorkFlowKind is a specific kind of work flow, characterized by the area of expertise in which it occurs. It is specialized into ActivityKind and ProcessKind. An activity is a work flow that represents a continuous responsibility. An ActivityKind is a specific kind of activity, characterized by the area of expertise in which it occurs. A process is a work flow that represents a discrete job. A ProcessKind is a specific kind of process, characterized by the area of expertise in which it occurs.

Here, we first evaluate what current support is available for a range of agile methods for these two groups of metaclasses and their generated fragments. When the support is not available (in terms of a fragment held in the methodbase), we propose the addition of a new fragment, documented in the OPF standard style including alphabetical ordering (see Appendices).

NEWLY IDENTIFIED FRAGMENTS TO SUPPORT AGILE DEVELOPMENT

This study has identified a large number of new work unit fragments that could be considered for addition to the current OPF repository/methodbase. These are summarized in the following sections and are detailed in Appendixes A-C in terms of the metaclass from which they are generated.