Chapter 24

A Value-Based Framework for Software Evolutionary Testing

Du Zhang

California State University, USA

ABSTRACT

The fundamental objective in value-based software engineering is to integrate consistent stakeholder value propositions into the full extent of software engineering principles and practices so as to increase the value for software assets. In such a value-based setting, artifacts in software development such as requirement specifications, use cases, test cases, or defects, are not treated as equally important during the development process. Instead, they will be differentiated according to how much they are contributing, directly or indirectly, to the stakeholder value propositions. The higher the contributions, the more important the artifacts become. In turn, development activities involving more important artifacts should be given higher priorities and greater considerations in the development process. In this paper, a value-based framework is proposed for carrying out software evolutionary testing with a focus on test data generation through genetic algorithms. The proposed framework incorporates general principles in value-based software testing and makes it possible to prioritize testing decisions that are rooted in the stakeholder value propositions. It allows for a cost-effective way to fulfill most valuable testing objectives first and a graceful degradation when planned testing process has to be shortened.

1. INTRODUCTION

The fundamental objective in value-based software engineering is to integrate consistent stakeholder value propositions into the full extent of software engineering principles and practices so as to increase the value for software assets (Biffl et al., 2006; Boehm, 2006). In such a value-based setting, artifacts in software development such as requirement specifications, use cases, test cases, or defects, are not treated as equally important during the development process. Instead, they...
will be differentiated according to how much they are contributing, directly or indirectly, to the stakeholder value propositions. The higher the contributions, the more important the artifacts become. In turn, development activities involving more important artifacts should be given higher priorities and greater considerations in the development process. In the value-neutral setting of software engineering practices, the focus is primarily on technical issues and activities, and there is generally a lack of economic driving force behind the technical activities. The value-based approach broadens this narrow view and brings new perspectives into software engineering theory and practice. Developing successful software systems hinges not only on the body of software engineering knowledge, but also on knowledge from economics, management sciences, cognitive sciences, and humanities (Biffl et al., 2006). The value-based approach also makes it explicit that the ultimate goal for software systems to satisfy and conform to evolving human and organizational needs is to create value (Biffl et al., 2006).

Another emerging trend over the last two decades has been the application of machine learning to the development and maintenance of large and complex software systems. Though the types of machine learning applications to software engineering issues and tasks thus far are primarily restricted to the value-neutral setting (Zhang, 2000; Zhang & Tsai, 2003, 2005, 2007; Zhang, 2009), there does not exist any fundamental obstacle that would prevent machine learning applications from outgrowing into value-based setting. The objective in value-based software engineering is to make software development decisions that are better for value creation. The hallmark of machine learning is that it results in an improved ability to make better decisions. Therefore, we think machine learning has an active and important role to play with regard to the agenda items in value-based software engineering. In this paper, our focus is on the interplay of machine learning methods in value-based software engineering (Figure 1).

One of the machine learning methods is genetic algorithms. Genetic algorithms have been used in what is referred to as evolutionary testing for generating test data (McMinn, 2004). However, the state-of-the-practice in evolutionary testing has been largely confined to the value-neutral setting. In this paper, we propose a value-based framework for carrying out software evolutionary testing with a focus on test data generation through genetic algorithms. The proposed framework, which falls into the area of value-based verification and validation, incorporates general principles in value-based software testing and aligns stakeholder value propositions with the technical activities in the test data generation process. The framework also makes it possible to prioritize decisions in the testing process. It allows for a cost-effective way to fulfill most valuable testing objectives first and a graceful degradation when planned testing process has to be shortened.

The rest of the paper is organized as follows. Section 2 summarizes the state-of-the-practice of machine learning applications to software test data generation in a value-neutral setting. In Section 3, we provide definitions for several concepts that will be utilized in the framework. Section 4 describes the proposed value-based framework for software evolutionary testing. An illustrative example is given in Section 5 of how the proposed framework can be put to work. Finally in Section 6 we conclude the paper with remark on future work.