Chapter XII
Integrated Software Testing Learning Environment for Training Senior-Level Computer Science Students

Daniel Bolanos
Universidad Autonoma de Madrid, Spain

Almudena Sierra
Universidad Rey Juan Carlos, Spain

ABSTRACT

Due to the increasingly important role of software testing in software quality assurance, during the last several years, the utilization of automated testing tools, and particularly those belonging to the xUnit family, has proven to be invaluable. However, as the number of resources available continues increasing, the complexity derived from the selection and integration of the most relevant software testing principles, techniques and tools into an adequate learning environment for training computer science students in software testing, increases too. In this chapter we introduce a experience of teaching Software Testing for a senior-level course. In the elaboration of the course a wide variety of testing techniques, methodologies and tools have been selected and seamlessly integrated. An evaluation of students performance during the three academic years that the course has been held show that students' attitudes changed with a high or at least a positive statistical significance.

INTRODUCTION

In this chapter we present a complete methodology for software testing training in the context of a laboratory course for senior-level computer science students. The intent of this work is to provide educators with a set of guidelines to effectively instruct computer science students on software testing. The goal is not only to incorporate specific software testing skills into students' curricula,

but also to prepare the student with skills for independent lifelong learning on the topic. The designed course spans the whole software testing lifecycle, and includes teaching recommendations to address students' common difficulties and misconceptions, as well as techniques to evaluate Students' performance for every stage.

During three academic years (2003-2006, note that results for the ongoing academic year are not currently available) we have developed and improved a software testing learning environment that has been used to train senior-level students in the Department of Computer Science of Universidad Autonoma de Madrid (Spain). In this environment, students are instructed about the elaboration of the test plan, test cases design, testing automation by means of specific tools, reporting and interpreting test results and maintenance related issues. All of these tasks are carried out over a complete pre-existent software system that has been specifically developed for this purpose.

To evaluate the effectiveness of the approach we have carried out attitudinal surveys to students during the three years that the course has been offered. These surveys provided us with inestimable information about students' progress and perception on several aspects of the course. This information was used to find out which elements of the course were perceived by students as most useful, most difficult or most personally rewarding; and, of course, to improve the learning environment along the academic years. We have found that, thanks to their immersion in this testing environment, students understood the crucial importance of software testing across the software lifecycle. Also, they incorporated a complete testing methodology and a broad set of software testing tools into their previous knowledge.

The chapter is divided into the following sections: a background section in which previous work on the topic is discussed and compared to the proposed approach, a description of the software testing learning environment including teaching recommendations and a description of the students' performance evaluation method, an evaluation of the effectiveness of the approach and a final section with the conclusions and future work.

BACKGROUND

Due to the increasingly important role of software testing in software quality assurance, during the last years, the use of testing frameworks that assist the developer during the testing process, and particularly the use of those belonging to the xUnit family, has proven to be invaluable. The production of high-quality and bug-free software products and solutions has gained a crucial importance in the software development industry, always focused to meet the needs of its increasingly more demanding end-users. In the last few years, many software testing techniques and methodologies have emerged to address these challenges, some of them influenced by agile (Beck, K. et al., 2001) and particularly by Extreme Programming (XP) (Beck, K., 2000). These techniques provide a wide set of principles, practices and recommendations for all the tasks involved in the software testing process, from test case design to automation of functional tests. In this context, an overwhelming number of testing frameworks and tools have been developed and are available (many of them under open-source licenses) with the purpose of aiding the developer in testing every particular system aspect written in any programming language imaginable.

However, as the number of resources and techniques available continues increasing and demonstrating new benefits, the complexity derived from the selection and integration of the most relevant software testing principles, techniques and tools into an adequate learning environment for training computer science students in software testing, increases too. Though several interesting experiences have been reported, to