Using Web-Based Repository of Testing Tutorials (WReSTT) with a Cyber Learning Environment to Improve Testing Knowledge of Computer Science Students

Dr. Hakam Alomari, Miami University

Dr. Hakam W. Alomari is an Assistant Professor in the Department of Computer Science and Software Engineering at Miami University in Ohio, USA. Dr. Alomari’s research interests are centered on static program analysis, more specifically, software slicing. And he has authored a number of IEEE/ACM refereed publications, including a selected and invited paper to a special issue of the best papers at WCRE’12 of the (Wiley) Journal of Software: Evolution and Process.

Dr. James D Kiper, Miami University

James Kiper is Chair and Professor of the Department of Computer Science and Software Engineering at Miami University. He teaches a variety of courses across the CS and SE curricula. His research is in the areas of software testing, software risk assessment, design rationale, and computer science and software engineering teaching and learning.

Dr. Gursimran Singh Walia

Gursimran S. Walia is an associate professor of Computer Science at North Dakota State University. His main research interests include empirical software engineering, software engineering education, human factors in software engineering, and software quality. He is a member of the IEEE Computer Society. Contact him at gursimran.walia@ndsu.edu

Katharine Zaback
Using Web-Based Repository of Testing Tutorials (WReSTT) with a Cyber Learning Environment to Improve Testing Knowledge of Computer Science Students: An Empirical Investigation

Author 1, Author 2
Department of Computer Science
XYZ University, City, Country
1 Introduction

Software systems must be maintained to remain useful [Lehman '80], and testing the system after a particular maintenance task is a key aspect for any system. As systems grow, maintenance typically becomes more complicated and costly [Ramil and Lehman '00]. Thus, software developers must be well-trained in software testing to plan and carry out effective software testing activities. To achieve a better trained workforce in the area of software testing, curricula at academic institutions need to be integrated with software validation techniques, including software testing.

The problem of integrating software testing concepts and related techniques into programming and software engineering (SE) courses is still an open research problem. Over the past decade, there have been many investigations, specifically by members of both academic community and software industry, into ways to integrate software testing into computer science (CS) and SE curricula. For example, several approaches attempted to integrate testing concepts into CS1 and CS2 courses using test-driven learning (TDL) and test-driven development (TDD) methods [Frezza '02, Edwards '03, Janzen and Saiedian '06, Desai, Janzen et al. '09, Dvornik, Janzen et al. '11, Clarke, Pava et al. '12, Clarke, Davis et al. '14]. TDL is an approach that is used to teach computer programming that integrates automated unit tests throughout the CS and SE curriculum, and TDD is a disciplined development approach that is used with programmers or learners to write an automated test then the code to make the test pass. These tests are usually small, testing only one method, and written with a unit testing framework such as JUnit. Other findings [S. Elbaum, S. Person et al. '07, Desai, Janzen et al. '09, Schaub '09] propose similar techniques that all seek to familiarize students with the concepts and practices of software testing implicitly in curriculum. These approaches seek to integrate the concepts and practices of software testing into existing curriculum in the hopes of teaching students how to better develop software, rather than the typical trial-and-error approach that Edwards [Edwards '03] and others have frequently observed.

However, while these proposed techniques show promise there is still a lack of resources and tools available for instructors to adopt these approaches. In order for the integration of software testing into existing CS/SE curriculum to be effective, instructors must possess the necessary knowledge and skill. There is a need for appropriate assistive tools, course materials and other resources in order to achieve successful curriculum integration. WReSTT - Cyber-Learning Environment is a research project whose goals include providing this support. Web-Based Repository of Software Testing Tutorials (WReSTT) is web-based repository of learning objects and video tutorials for helping students learn about software testing techniques and tools. It includes support for team collaboration and some simple forms of gamification to improve student learning. A learning object is a self-contained collection of materials like text, video, and examples and some related assessments, e.g., quizzes, with the goal of achieving some students learning outcome. Learning objects can be used as a standard format to structure the learning materials to instructors and students who plan to use WReSTT as a future learning resource.

To the end of improved training in software testing for our students, this paper presents empirical studies conducted in multiple undergraduate CS/SE programming courses at two large public universities (University A and University B). The goal of our study is two-fold: 1) to evaluate the impact of using WReSTT on the undergraduate students’ acquisition of knowledge and proficiency of using testing techniques and tools, and 2) to demonstrate that using WReSTT resulted in improved learning for students on their programming assignments and their ability to use testing tools on their team projects.

1 http://wrestt.cis.fiu.edu/about-wrestt-com
We conducted two evaluations: 1) pre/posttest instrument used to illustrate the first objective, and 2) user survey used to address the second objective. The results indicate that using WReSTT helped students improve their understanding and use of software testing techniques and tools. Specifically, students indicated that learning objects and video tutorials in WReSTT helped them understand both software testing concepts and tools, and that tutorials in WReSTT helped students during the course assignments and earn a better grade in the class. Students also indicated that the current version of WReSTT was easy to use, that they are more likely to use testing tools during their team project, and that they would recommend its use to other undergraduate students.

The reminder of this paper is organized as follows. Section 2 describes the main features of WReSTT, along with the instructor and the student view of the tool. Section 3 describes the experiment design, the participating students, and the study procedure and data collected during the experiment run. Section 4 presents the analysis of the data organized around the study goals with a discussion of results including limitations and threats to the results' validity followed by concluding remarks in Section 5.

2 WReSTT: Web-based Repository of Software Testing Tutorials

In this section, we discuss how the WReSTT tool can be used both by instructors and students, in CS/SE courses. The following subsections provide an overview of the tool, the current design, and its main features that are available for both categories of users.

2.1 Overview, Current Design, and Main Features

WReSTT - version 1, was initially designed and developed by Clarke et al. in 2010 [Clarke, Allen et al. '10] as a testing aid to support the pedagogical needs of instructors and students by providing a set of testing tutorials and tools. Since then, WReSTT has evolved into a collaborative learning environment with social networking features (e.g., virtual points, leader board). This enhancement was based on the results of several studies and feedback from students and instructors [Peter J. Clarke, Jairo Pava et al. '11, Clarke, Pava et al. '12]. WReSTT - version 2, used in this paper, is now available for use online at http://wrestdt.cis.fiu.edu/. The current design of WReSTT- version 2 is discussed in following subsections.

The WReSTT tool is a web-based application that includes several components as shown in Figure 1, including:

- **Learning content** (testing concepts/tools) which provides the student with learning objects (related to testing concepts) and video tutorials on a variety of software testing tools and methodologies;
- **Course management** component which allow the instructor to monitor activity streams of each student, to generate student reports, and to allocate virtual points for students’ activities. Virtual points are given for individual activities, as well as team activities where all team members must complete the activity for the team. Although virtual points are independent of the course grade, it can be used by instructors as a small part of the course grade.
- **Social networking** features allow students to upload his/her user profile, post comments on the discussion board, gain virtual points (for completing a learning object), and monitor the activity of peers via course activity and leaderboards.
Instructor and Student View

WReSTT can be used in a variety of ways based on the level of testing that is required in a given course due to the variety of learning objects and tutorials available on WReSTT. For example, WReSTT may be introduced to both graduate and undergraduate students by having instructors assign students learning content with varying level of difficulty. Note that all students involved in the experiment described this paper were undergraduate students.

Instructor View - WReSTT allows the capability for an instructor to create a course module by enrolling students into the course and providing students with access credentials for using WReSTT. By creating a course and using the course management page, the instructor can (1) upload the class roster; (2) create unique login credential for the students; (3) assign students to virtual teams; (4) describe the rubric for the allocation of virtual points for different student activities; and (5) create student reports. There is help menu available on the WReSTT to assist instructors during the course set-up and subsequent management.

Student View: Students can create a user profile by uploading a profile picture (and gain some virtual points), browse the testing tutorials, complete assigned learning objects by passing with at least 80% on assigned quizzes (and gain virtual points), watch tutorial videos on the different testing tools (e.g., JUnit, JDepend, EMMA, CPPU, Cobertura), interact with other students in the class via testing based discussions (and gain virtual points for relevant discussions), and monitor the activity stream for whole class.

3 Empirical Study Design

Empirical studies were aimed at investigating the impact the WReSTT had on students' acquisition of knowledge, and proficiency of testing techniques and tools. Specifically, WReSTT was used at two large public universities in different semesters across multiple undergraduate programming courses with some programming content. This section describes research questions, subject selection, experimental design, and instrumentation.
3.1 Research Questions

We decided to evaluate the impact of using WReSTT through both objective (pre/posttest instrument) and subjective (user survey) evaluation. A pretest and posttest instrument was used at the beginning and at the conclusion of each course respectively.

Users in the treatment groups (that were exposed to WReSTT) completed the post-study survey and provided feedback on using WReSTT in CS/SE courses, on a variety of quality attributes using a 5-point Likert scale. The WReSTT survey questions focused on three primary areas: (1) usability questions focused on the three primary usability requirements: visibility (it should be obvious what components are used for), affordance (it should be obvious how a component is used), and feedback (it should be obvious when a component has been used). (2) feedback on the impact of learning testing concepts and tools on their programming assignments; and (3) feedback on the impact the social networking component had on students’ participation. Therefore, using the results of the pre/posttest and the user surveys, we endeavored to answer the following research questions:

- **RQ1**: Does WReSTT significantly impact the testing knowledge and testing tool proficiency gained by the students?
- **RQ2**: Has WReSTT resulted in improved learning as demonstrated in students’ feedback?
- **RQ3**: Does WReSTT meet usability requirements?

3.2 Subjects and Courses

The testing courses ranged from CS1 and CS2 (Introductory CS courses); Modern Software Development (200 level), System1 (200 level), Data Communications and Networks (200 level), Software Construction (200 level), and Systems Testing and Maintenance (200 level). Table 1 shows a summary of the classes that participated in the study from University A, and Table 2 shows a summary of the classes that participated in the study from University B.

As shown in Table 1, at University A, a total of 195 subjects participated, including three control groups; CG1 – CG3, containing 30, 49, and 14 students, respectively, and three treatment groups; TG1 – TG3, containing 21, 43, and 38 students, respectively. In University B, as shown in Table 2, a total of 158 subjects participated, including three control groups; CG1 – CG3, containing 31, 28, and 27 students, respectively, and three treatment groups; TG1 – TG3, containing 23, 26, and 23 students, respectively. The average percentage of participation across both universities was ~82%.

Please note that the participating subjects at both universities represent students who completed pre/post- tests and the post-study survey. All subjects enrolled in these courses were pre and post-tested using the same instrument. There was no difference in terms of demographics and course preparation between the subjects in the treatment groups and control groups. The only difference was the treatment group was exposed to WReSTT and the control group did not have a WReSTT support but were exposed to the same lecture and other course content.

Students in the treatment groups were additionally completed a survey at the end of the semester. As shown in Table 1, a total of 127 students out of 139 students participated in the survey from University A with an average percentage of participation equal to ~92%. Similarly, a total of 90 students out of 96 students participated in the WReSTT user experience survey from University B, as we can see in Table 2, with an average percentage of participation equal to ~94%.
### Table 1. Participating Subjects at University A.

<table>
<thead>
<tr>
<th>Group ID</th>
<th>Class</th>
<th>Enrollment</th>
<th>Pre/Posttest</th>
<th>WReSTT UX Survey</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Participation</td>
<td>%</td>
</tr>
<tr>
<td>TG1</td>
<td>CS1 - SP16</td>
<td>29</td>
<td>21</td>
<td>72.4</td>
</tr>
<tr>
<td>TG2</td>
<td>CS2 - SP16</td>
<td>58</td>
<td>43</td>
<td>74.1</td>
</tr>
<tr>
<td>TG3</td>
<td>MSD - Fa15</td>
<td>52</td>
<td>38</td>
<td>73.1</td>
</tr>
</tbody>
</table>

**Treatment Group:**

**Control Group:**

<table>
<thead>
<tr>
<th>Group ID</th>
<th>Class</th>
<th>Enrollment</th>
<th>Pre/Posttest</th>
<th>WReSTT UX Survey</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Participation</td>
<td>%</td>
</tr>
<tr>
<td>CG1</td>
<td>CS1 - Fa15</td>
<td>33</td>
<td>30</td>
<td>90.9</td>
</tr>
<tr>
<td>CG2</td>
<td>CS2 - Fa15</td>
<td>51</td>
<td>49</td>
<td>96.1</td>
</tr>
<tr>
<td>CG3</td>
<td>STM - SP16</td>
<td>16</td>
<td>14</td>
<td>87.5</td>
</tr>
</tbody>
</table>

Total: 239 195 81.6 127 91.4

MSD – Modern Software Development, STM – Software Testing & Maintenance
SP1x – Spring 201x; Fa1x – Fall 201x

### Table 2. Participating Subjects at University B.

<table>
<thead>
<tr>
<th>Group ID</th>
<th>Class</th>
<th>Enrollment</th>
<th>Pre/Posttest</th>
<th>WReSTT UX Survey</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Participation</td>
<td>%</td>
</tr>
<tr>
<td>TG1</td>
<td>CSE278a - SP16</td>
<td>31</td>
<td>23</td>
<td>74.2</td>
</tr>
<tr>
<td>TG2</td>
<td>CSE211a - SP16</td>
<td>35</td>
<td>26</td>
<td>74.3</td>
</tr>
<tr>
<td>TG3</td>
<td>CSE283a - SP15</td>
<td>30</td>
<td>23</td>
<td>76.7</td>
</tr>
</tbody>
</table>

**Treatment Group:**

**Control Group:**

<table>
<thead>
<tr>
<th>Group ID</th>
<th>Class</th>
<th>Enrollment</th>
<th>Pre/Posttest</th>
<th>WReSTT UX Survey</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Participation</td>
<td>%</td>
</tr>
<tr>
<td>CG1</td>
<td>CSE278b - SP16</td>
<td>36</td>
<td>31</td>
<td>86.1</td>
</tr>
<tr>
<td>CG2</td>
<td>CSE278c - SP16</td>
<td>32</td>
<td>28</td>
<td>87.5</td>
</tr>
<tr>
<td>CG3</td>
<td>CSE283b - SP15</td>
<td>30</td>
<td>27</td>
<td>90.0</td>
</tr>
</tbody>
</table>

Total: 194 158 81.4 90 93.8

CSE278 – System1; CSE211 – Software Construction; CSE283 – Data Communications & Networks
SP1x – Spring 201x

### 3.3 Study Procedure

The data used in this study was collected using a pre/posttest instrument and the WReSTT heuristic survey. The pre/posttest instrument was designed to determine students’ knowledge before and after being exposed to testing topic. The pre/posttest instrument used in this study is available at [http://wrestt.cis.fiu.edu/documentation](http://wrestt.cis.fiu.edu/documentation). It consists of eight questions that focus on the usage proficiency of the testing tools in addition to the conceptual testing knowledge. In order to simplify the evaluation of pre/posttest, 1 point was allocated for each correct answer.

The study procedures at both universities were similar. The pretest was administered at the beginning of the class (i.e., week 1), prior to the testing part of the course. Next, also prior to the testing part of the course, subjects in the treatment groups were introduced to WReSTT features, tutorials, post and view discussion threads, take self-assessment quizzes and learning objects related to different testing concepts. Additionally, subjects were trained on how to use the tool to complete their programming assignments by browsing through learning objects and video tutorials mostly focused on JUnit (unit testing framework) and EclEmma (code coverage tools). Near the
completion of the semester (e.g., week 12), subjects were post-tested using the same instrument (that they had been pretested) to evaluate their improvement in knowledge of testing concepts and tools. Students in the control groups were not exposed to WReSTT during these courses at all, however, those students still learned about testing.

Upon finishing the testing topic being taught in the course and finishing using the WReSTT tool, the participants in the treatment groups (CS1, CS2, and MSD courses from University A, and CSE278a, CSE211a, and CSE283a courses from University B) were asked to complete a WReSTT heuristic evaluation. The heuristic evaluation involves participants examining the WReSTT tool and our used approach of the tool and judge its impact on their learning and using testing tools during their team project and programming assignments.

The survey, also available at http://wrestt.cis.fiu.edu/documentation, consisted of 25 questions with each answer on a five point Likert scale from: 1 = strongly disagree, 2 = disagree, 3 = neither agree nor disagree, 4 = agree, to 5 = strongly agree. These questions are designed to measure the strength of users’ opinions by counting the number of responses at each point in the scale. The three main categories used to develop our survey’s questions are: (1) overall reaction-related questions to WReSTT website, (2) testing-related questions, and (3) social networking-related questions.

4 Results and Analysis

This section reports analysis of data collected during the study run to answer the three major research questions listed in Section 3.1.

4.1 Pre/posttest

Prior to the testing topic being taught in the class and at the end of each semester, students were administered a pretest and posttest, respectively. For each test, participants were given a score to indicate their performance. Table 3 and Table 4 show a summary of the results and analysis for the pre/posttest for control and treatment groups in University A and University B, respectively. Each table is divided into three sections: section (A) shows both the mean and standard deviation for the score on the pre/posttest, section (B) shows the main effects of the test using a 2 (test group) X 2 (test) repeated measures ANOVA [Michael H. Kutner, Christopher J. Nachtsheim et al. ’04], and section (C) shows the interaction between the test and the test group.

As shown in Table 3, students in the treatment group who used WReSTT performed significantly better on their posttests than students in the control groups (who did not use WReSTT). The same scenario is reported by University B, as shown in Table 4. However, the difference between students’ pretest scores is smaller than the corresponding difference reported by University A. As we can see from Table 3, there was no significant difference between students’ pretest scores in both universities, indicating that this difference was not due to differences in students’ knowledge before taking the course.

4.2 WReSTT User Experience Survey

At the end of the semester, students in the treatment groups were administered a user experience survey to assess the three heuristics categories mentioned above, again these categories are: (1) the ease of using the WReSTT website, (2) the usefulness of using WReSTT to better understand the
testing concepts and tools, and (3) the usefulness of using WReSTT as a collaborative learning environment to complete programming assignments and to achieve better grades in their courses.

Table 3. University A's overall results of Pretest and Posttest. Section (A) reveals students’ mean scores and standard deviations in the control and treatment groups. Section (B) presents the main effects of the test and group using a 2 (test group) X 2 (test) repeated measures ANOVA. Section (C) presents the interaction between the test group and the test.

<table>
<thead>
<tr>
<th>(A)</th>
<th>Group</th>
<th>N</th>
<th>Pretest</th>
<th>Posttest</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Mean</td>
<td>SD</td>
</tr>
<tr>
<td></td>
<td>Control</td>
<td>93</td>
<td>6.61</td>
<td>3.64</td>
</tr>
<tr>
<td></td>
<td>Treatment</td>
<td>102</td>
<td>7.75</td>
<td>4.21</td>
</tr>
</tbody>
</table>

(B) Measure: Main Effects
Test (2) F(1, 193) = 19.82, p < 0.01, students performed better on the posttest. Follow-up analysis: t(102) = -4.47, p < 0.01, shows significant difference between pre- and posttest scores in the treatment group, but not in the control group.

Group (2) F(1, 193) = 8.31, p < 0.01, students performed better in the treatment group

(C) Measures: Interaction
Group, Test F(1, 193) = 21.64, p < 0.01, significant interaction

Table 4. University B's overall results of Pretest and Posttest. Section (A) reveals students’ mean scores and standard deviations in the control and treatment groups. Section (B) presents the main effects of the test and group using a 2 (test group) X 2 (test) repeated measures ANOVA. Section (C) presents the interaction between the test group and the test.

<table>
<thead>
<tr>
<th>(A)</th>
<th>Group</th>
<th>N</th>
<th>Pretest</th>
<th>Posttest</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Mean</td>
<td>SD</td>
</tr>
<tr>
<td></td>
<td>Control</td>
<td>86</td>
<td>5.03</td>
<td>3.61</td>
</tr>
<tr>
<td></td>
<td>Treatment</td>
<td>72</td>
<td>4.51</td>
<td>3.73</td>
</tr>
</tbody>
</table>

(B) Measure: Main Effects
Test (2) F(1, 156) = 20.6, p < 0.01, students performed better on the posttest. Follow-up analysis: t(72) = -3.40, p < 0.01, shows significant difference between pre- and posttest scores in the treatment group, but not in the control group.

Group (2) F(1, 156) = 15.42, p < 0.01, students performed better in the treatment group

(C) Measures: Interaction
Group, Test F(1, 156) = 8.64, p < 0.01, significant interaction

We start by considering the response values for the questions. Specifically, each question’s mean value of responses for all participants. To assess the level of agreement, we calculate the mean value of responses for each given question by comparing successive categories. In general, if the differences in the average for all questions under a specific category are small, then we can consider that category trending towards “disagreement”, otherwise trending towards “agreement”.
Figure 2 shows the mean value of responses for each question grouped by the three categories, Category 1 – Category 3. As shown, generally speaking, students in both universities trend towards agreement in all categories. However, by focusing on the question itself in isolation, it appears that the mean value of responses by students in both universities to Q6 in Category 1 (the information, such as online help, on-page messages, and other documentation provided with the web site is clear) is low: 1.93 in University A and 2.87 in University B. Thus, participants in both universities agreed that there is no sufficient help documentation provided. Therefore, we recommend that we consider this question as a future improvement opportunity of the WReSTT website. Additionally, Question Q20 under Category 2, asking whether they would have used testing tools if WReSTT had not existed, indicated that most of them would not (with a mean value ~2). We consider this result as a positive reaction of the usefulness of using WReSTT.

The questions from Q1 – Q5 in Category 1, trend towards agreement by students in both universities. However, students in University B were more inclined to agree that WReSTT was easy to use and provided clear feedback without any visibility issues. In contrast, students in University A were more tending to agree of the usefulness of the tutorials available in WReSTT (Q15 through Q19 under Category 2), and that the use of a collaborative learning (aka social networking) environment in WReSTT was a motivating factor in their learning (Q21 through Q25 under Category 3).

These results can be explained in part due to the selection of samples for University A and University B. For example, most of the participants in University A come from an Introductory level courses CS I and CS II, and participants from University B come from 200 level courses. Therefore, participants in University A did better in Category 2 and Category 3, as they found WReSTT a good resource of learning testing concepts and tools and they benefit from it in enhancing their grades and their collaborative learning in their team project. However, participants in University B did better in Category 1 (still doing good in the other two categories), which means they are more comfortable using the tool and they found it usable and easy to use.

Figure 2. The students' mean scores of responses for each question in Category 1 (overall reaction to WReSTT website), Category 2 (Testing-related questions), and Category 3 (Collaborative learning-related questions) by University (A) and University (B) using a Likert scale from strongly disagree represented by 1, to strongly agree, represented by 5.
4.3 Discussion

In this section, we discuss our takeaways and highlights from the data. This includes our plans for future work intended to address any shortcomings and opportunities for improvement identified in our study. The pre/posttest results are used to illustrate the first research question (RQ1) and partially address the second research question (RQ2). Then, the survey’s results are used to fully address the second research question (RQ2) and the third research question (RQ3).

4.3.1 RQ1: Significant Impact

The results shown in Table 3 and Table 4 indicate that using WReSTT as a teaching tool for students learning software testing can significantly improve their understanding and use of software testing techniques and tools. This is clearly obvious using the results provided by the pre/posttest. The posttest scores were much better than pretest scores in both universities showing that WReSTT may have been one factor in helping students improve their testing conceptual knowledge. ANOVA test used to evaluate the significance of the improvement returned p values of < 0.05 in all cases showing that posttest scores were significantly better than pretest scores.

Interestingly, students in the treatment groups in University B reported significantly lower confidence of testing knowledge at the beginning of the semester than students in the control groups. Additionally, the same students reported lower pretest’s scores than corresponding students in University A. It is not clear why this is the case. Intuition would suggest that students in the treatment groups in the upper level courses (University B) should report higher scores in pretest than students in the control groups in the same university, and higher scores in pretest than students in introductory programming courses (University A). Contrary to this expectations, there was not an observable difference in baseline or improved performance in higher level course versus introductory computer programming courses (CS1 and CS2). These results raise some interesting questions for future research.

4.3.2 RQ2: Improved Learning

The user experience survey results, as shown in Figure 2 – Category 2, indicate that students do find WReSTT a useful resource for learning software testing concepts, techniques, and tools. Specifically, students indicated that the tutorials in WReSTT helped them understand both software testing concepts and tools, and that there are a sufficient number of tutorials in WReSTT. These findings are also supported by the reported pre/posttest results as shown in Table 3 and Table 4. Students in the treatment groups from both universities exhibited greater understanding and increased use of software testing tools on their posttests than students in the control groups, as well as a significant increase in knowledge from their pretests. Students also indicated, as shown in Figure 2 – Category 3, that the current version of WReSTT helped them complete course tasks and earn a better grade in the class.

4.3.3 RQ3: Usability Requirements

Regarding ease of use, students indicated, as shown in Figure 2 – Category 1, that the current version of WReSTT are easy to use and pleasant, and they would recommend use of the site to other future students. However, there is no help documentation provided, as reported by the participants from both universities as shown in Figure 2 – Category 1 – Q6.
4.4 Questions with regards to validity and experiment design

In this paper, we presented a large empirical study at two public universities aimed at improving testing knowledge of CS/SE students by using a web-based repository of testing tutorials (WReSTT). The dataset was obtained from a pretest and posttest instrument, and a user experience survey. The major threat to the result’s validity comes from studying the pre/posttest results as an overall result without looking more closely at the student achievement in specific knowledge area. For example, by looking back at the questions in the pre/posttest instrument, we can see that these questions can be distributed into three different knowledge areas, such as: testing concepts, tool usage, and tool proficiency level. This is a future topic we intend to explore.

Another threat to validity of these results are related mainly to the selection of the samples for the treatment and control groups. Table 1 and Table 2 show the sample size used in this study; we can see there is a lower participation for the treatment groups in both universities with an average equal to ~18% in University A and ~13 in University B. This lower participation can be explained through the observation that participation in this experiment was an “add-on” to the class and was not fully integrated into the course. Many of our students are motivated by grades and points. There was not a sufficient level of motivation through the small point value given for their participation.

5 Conclusion

In this paper, we described our approach to integrate software testing as a concept, techniques, and tools into CS/SE courses. In practice, we do not foresee this approach being a replacement for existing modules taught in CS/SE courses. Rather, instructors can use our approach to supplement/complement them by providing a collaborative learning environment that engages students and supports software testing learning activities. These learning materials are introduced using a Web-Based Repository of Testing Tutorials, WReSTT.

We described the current design to WReSTT and the results from the comprehensive study to determine the impact WReSTT has on testing pedagogy. The study was performed using seven computer science/software engineering classes at two large public universities during Fall and Spring 2015/2016 semesters. The results showed that (1) the use of WReSTT as a teaching tool can significantly improve the students' understanding and use of software testing techniques and tools; (2) students find WReSTT as a useful learning resource for learning software testing techniques and tools; and (3) students find using WReSTT as a collaborative learning environment motivates them to complete programming assignments and achieve a better grade in that course.

In the near future, we plan to perform additional studies at different academic institutions to further validate our results and experience. We will be organizing our approach of using WReSTT into learning objects. This will allow instructors and students to customize their use of WReSTT and provide a more structured approach to use WReSTT as a learning resource.

References


