Using WReSTT in SE Courses: An Empirical Study

Peter J. Clarke, Jairo Pava, Debra Davis and Frank Hernandez
School of Computing and Information Sciences
Florida International University
Miami, FL 33199, USA
{clarkep,jpava001,dledavis,fhern006}@cis.fiu.edu

Tariq M. King
Department of Computer Science
North Dakota State University
Fargo, ND 58108, USA
tariq.king@ndsu.edu

ABSTRACT

There continues to be a lack of adequate training for students in software testing techniques and tools at most academic institutions. Several educators and researchers have investigated innovative approaches that integrate testing into programming and software engineering (SE) courses with some success. The main problems are getting other educators to adopt their approaches and ensuring students continue to use the techniques they learned in previous courses.

In this paper we present a study that evaluates a non-intrusive approach to integrating software testing techniques and tools in SE courses. The study uses a *Web-Based Repository of Software Testing Tools* (WReSTT) that contains tutorials on software testing concepts and tools. The results of the study show that (1) students who use WReSTT in the classroom can improve their understanding and use of testing techniques and tools, (2) students find WReSTT a useful learning resource, and (3) the collaborative learning environment motivates students to complete assignments.

Categories and Subject Descriptors

K.3.2 [**Computer and Information Science Education**]: Miscellaneous

General Terms

Experimentation

Keywords

Software Testing, Unit testing, Repository, Collaborative Learning

1. INTRODUCTION

Integrating software testing in the curricula of many undergraduate computer science (CS) and information technology (IT) programs continues to be challenge. Although testing is one of the major approaches used to produce quality software [3], students get little or no training in the use of adequate testing techniques and tools. Several educators have investigated the integration of testing in CS1 and CS2 courses with success [5, 6, 7, 11] using various innovative techniques. However, anecdotal evidence suggests that subsequent courses in the curricula do not enforce the testing approaches taught in the earlier courses. This is one of the main reasons why we have elected to use a non-intrusive approach to integrating software testing techniques and tools in SE and upper-level programming courses.

There have also been several attempts to improve the testing content in the courses later in the curricula [10, 12, 13, 15]. This is of particular importance since software engineering and computer programming are expected to be two of the fastest growing occupations in the USA[1]. Although more academic institutions are offering software engineering (SE) courses, a great deal more needs to be done in order to adequately expose students to software testing and associated tools [14]. Using a non-intrusive approach to integrating software testing and testing tools into SE courses significantly reduces the need for the instructor to change the existing syllabus, assignments or projects. We expect our approach will lead to wider adoption of the use of testing tools in SE and upper-level programming courses.

The work presented in this paper builds on the work by Clarke et al. [1, 2] on the *Web-Based Repository of Software Testing Tools* - (WReSTT), an online repository of learning resources, to support software testing pedagogy. The initial presentation [1] describes the first version (V1) of WReSTT which primarily contains tutorials on how to setup and use software testing tools. Based on student feedback from WReSTT V1, a second version (V2) of WReSTT, was developed using a more user-friendly interface and included a component that utilized collaborative learning techniques [2]. Results from previous preliminary studies were used to guide the development of WReSTT V2.

In this paper we describe the recent improvements to WReSTT and the results of the first comprehensive study on the impact WReSTT has on students' conceptual understanding and use of testing techniques and testing tools. The main contributions of the paper are as follows:

1. Extensions of WReSTT, specifically the instructors' and administrators' interface for course management.
2. Results of a study to show: (1) the impact WReSTT has on students' knowledge of software testing and testing tools; (2) the usability of WReSTT as a learning resource; and (3) the impact the collaborative learning environment has on motivating students to complete tasks.

The remainder of the paper is organized as follows. Section 2 presents the work most closely related to studies on using online resources in software testing pedagogy. Section

SIGCSE'12, February 29–March 3, 2012, Raleigh, North Carolina, USA.
ACM 978-1-4503-1098-7/12/02.

[1] http://www.bls.gov/oco/ocos303.htm#outlook

3 presents a brief overview of WReSTT and recent updates. Section 4 describes the study and we conclude in Section 5.

2. RELATED WORK

There are several online resources to support teaching and learning software testing concepts, skills, and other aspects of SE course curricula. In this section, we provide information on other web-based software testing learning repositories and pedagogical approaches being used to integrate testing in SE and CS programming courses.

2.1 Online Resources

The Software Quality Engineering Research Group (SoftQual) at the University of Calgary have developed lab courseware that seeks to teach students practical software testing skills [10]. The SoftQual repository [20] is freely available online, and consists of several labs on various testing topics. In each lab, students are introduced to the objectives and given instructions that familiarize students with the activity to be performed [10]. Tools used to support the SoftQual labs include Bugzilla, JUnit, CodeCover, CodeLipse, Rational Functional Tester, and MuClipse [10, 20].

WebIDE [23] is a web-based development environment that seeks to introduce SE best practices at an early stage in a student's educational career. The current version of the tool focuses on Test-Driven Learning (TDL) through small iterative examples. Users can browse labs by programming language (C, Java). The WebIDE environment incorporates a lockstep mechanism that requires students to write examples and tests before solutions [6, 23]. Instructors can also write their own labs according to an SE approach of choice, build custom evaluators, and provide detailed error messages to students.

The Repository for Open Software Education (ROSE) [17] is an education-friendly, open-source repository designed for SE courses. Although the repository is general-purpose, it provides details on the test-related artifacts associated with the projects. These details include the availability and specific format of automated unit and acceptance tests, as well as the inclusion of test plans and requirements documentation [15, 17]. Other online resources are described by Clarke et al. [2].

There are some similarities between WReSTT and the aforementioned repositories, however the differences are: (1) WReSTT focuses on tutorials for testing tools, and (2) WReSTT supports a collaborative learning model where students perform team activities and obtain virtual points in a social networking-like environment.

2.2 Studies

Garousi [10] discusses the experience of using the SoftQual Testing Lab Courseware [20] in a fourth-year undergraduate Software Reliability and Testing Course. The class consisted of 27 students, and there was a single TA to help the instructor with grading the students' lab solutions. Garousi [10] states that the students found the lab exercises very useful and beneficial for learning. Students' comments on each lab are also documented in the experience report, and provide evidence that the SoftQual testing labs are effective for teaching software testing skills.

Dvornik [6] performed a pilot study on using WebIDE in two sections of a CS0 course, where the topic was Android Development. The following hypothesis was examined in the study: Students using WebIDE will perform better on programming tasks than those using the traditional static labs. Students were randomly assigned to control and experimental (WebIDE) groups and validation was performed to ensure that the programming experience of each group was equivalent. The results of the study showed that: (1) students were more likely to successfully complete their Android application with WebIDE than with a traditional development environment; and (2) among students with prior programming experience, students who used WebIDE wrote better automated unit tests than those who did not use WebIDE.

Meneely et al. [15] reported their lessons learned from integrating some project artifacts available in ROSE into an undergraduate SE course. The course included a 2-hour lab session once a week to provide students with hands-on experience through tutorials and other teaching activities. Use of the iTrust project [18] was successful, and the authors emphasized that the value of ROSE is in alleviating the difficulty of finding these types of education-friendly open-source projects for use in SE courses.

Clarke et el. [1, 2] preformed two preliminary studies related to the two implementations of WReSTT (V1 and V2). The first study [1] reported on attempts to integrate the use of testing tools into CS2 and SE courses using a non-intrusive approach. This study provided information that WReSTT V1 would have a greater impact on student learning in SE courses than in CS2 courses. In the CS2 course the instructor would have to change his syllabus and labs in order for students to participate in WreSTT V1. The results of the second study [2], although preliminary, showed that WReSTT V2 had an impact of team work and students' understanding on testing and testing tools.

3. WRESTT

In this section we provide a more detailed overview of WReSTT and describe recent updates made to improve the course management component.

3.1 Overview of WReSTT

As previously stated, WReSTT [24] provides students access to learning resources to support their pedagogical needs in the area of software testing. There are currently two versions of WReSTT used in the study presented in this paper. WReSTT V1 [1] initially contained mainly testing tool tutorials. These tutorials include: *JUnit* [9] - a unit testing framework, *SWAT* [22] - Simple Web Automation Toolkit, and *Cobertura* [4] - a code coverage tool, among others. Clarke et al. [1] provide a more complete description of WReSTT V1.

WReSTT V2 was developed to support collaborative learning by allowing students to work together in teams and earn virtual points. Unlike WReSTT V1, V2 includes tutorials and quizzes on the core concepts of software testing and no tutorials on testing tools. WReSTT V2 allows students to be grouped into virtual learning teams, which can then compete for points based on the collaborative completion of quizzes and other knowledge assessment tasks. Points translate to in-class extra credit at the discretion of the course instructor. In addition, the student interface was redesigned to provide students with the look and feel of a social networking application e.g., Facebook [8]. The social aspect of WReSTT V2 provides students with the ability to cre-

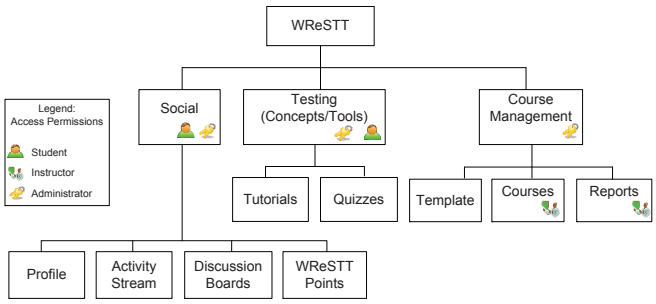

Figure 1: Partial block diagram of WReSTT V2.

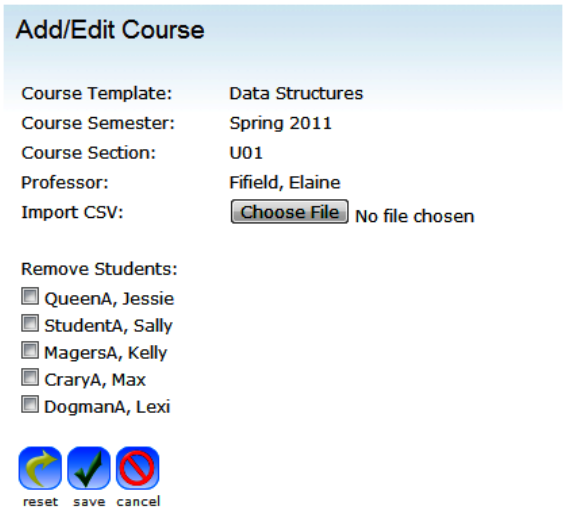

Figure 2: Instructor's interface to edit course.

ate a profile, monitor the learning progress of classmates, post comments to the discussion boards, and identify which students have acquired the most points. The partial block diagram in Figure 1 shows the social and testing components of WReSTT V2.

3.2 Course Management

During the 2011 spring semester a team of students in the Senior Project class worked on improving the features in WReSTT V2. The project team was led by *Jairo Pava* (second author of this paper) and two other students *Dionny Santiago* and *Yesenia Sosa*. The main objectives of the project were to (1) develop the functionality for an administrator to create and manage courses (2) create an interface for instructors to upload and manage students in their classes, and (3) to test the new and existing functionality of WReSTT V2. The new course management component of WReSTT V2 is shown on the right of Figure 1. The legend on the left shows the access permissions for each user. For example, the instructor can access individual courses and the reports for those courses. Note that the access permissions for a block are used in its sub-blocks in addition to any new permissions.

The course management component provides an administrator with the ability to create course templates which can then be used to assign an instructor to a course. Once an instructor is assigned a course, that instructor can upload students into the course using a comma separated value (csv) file, the interface is shown in Figure 2. After uploading the course roster the instructor can then assign students to the various virtual teams. During the course, the instructor can access statistics such as the time spent on WReSTT, time spent per tutorial, and time spent per quiz. The instructor can also download the number of virtual points per student for the semester.

4. EMPIRICAL STUDY

In this section we present the various components of the study including the objectives, the methods used, the results and a discussion of the results. The results of the study were based on data collected using a pre/posttest and a student survey.

4.1 Objectives of Sudy

Most of the work published on WReSTT included preliminary or informal studies on the impact WReSTT has on students in SE and programming classes. This is the first study that uses the appropriate statistical methods to determine how beneficial it is to use WReSTT in SE courses. The objectives of this study are as follows:

Objective 1: Determine if the use of the resources in WReSTT improve the students' knowledge of software testing and their knowledge of software testing tools.

Objective 2: Determine whether students find WReSTT a useful and easy to use learning resource.

Objective 3: Determine if the use of a collaborative learning environment in WReSTT motivates students to complete the assigned tasks.

4.2 Methods

Sample: The students who participated in the study were recruited from the Spring and Summer 2011 *CEN 4010 Software Engineering I* courses at Florida International University. The groups were taught by two different instructors following the same syllabus. A total of 48 subjects (23 from the Spring semester and 25 from the Summer semester) participated in the study. Students from the Spring semester were assigned to the control group and students from the Summer semester were the treatment group.

Measurement: Two instruments, a pre/posttest (test) and survey, were used to collect the data for the study. Test instruments given to both the Spring and Summer classes were used to measure the students' knowledge of testing concepts, online software testing resources and testing tools. The test instrument is shown in Appendix A and consists of eight questions. Questions 1 and 2 focused on the objective of program testing and the use of testing techniques to create input values to test a simple method. Questions 3 and 4 focused on the students' knowledge and proficiency in using software testing tools. Questions 5 and 6 were used to determine if the students were aware of online testing resources and the type of learning materials that are available in these resources. Questions 7 and 8 addressed the importance of testing in programming assignments.

The survey instrument consists of thirty questions divided into five sections. Section 1 consisted of one question to determine if the students knew of any other online resource besides WReSTT. Section 2 focused on comparing the students' experiences with using WReSTT V1 and WReSTT

Q	Overall Reaction to Websites	N	V1	V2
			M (SD)	M (SD)
2	Overall, I am satisfied with how easy it is to use the website	24	3.75 (1.1)	4.17 (0.9)
3	It is simple to use the website	24	3.71 (1.0)	4.04 (1.0)
4	I feel comfortable using the website	24	3.75 (1.0)	4.13 (0.8)
5	It was easy to learn to use the website	23	3.91 (1.0)	4.17 (1.0)
6	I believe I became productive quickly using the website.	23	3.83 (0.9)	3.87 (1.1)
7	The information (such as online help, on-page messages, and other documentation) provided with the web site is clear	22	3.59 (1.0)	3.86 (1.0)
8	It is easy to find the information I need	23	3.61 (1.0)	3.50 (0.8)
9	The information is effective in helping me complete the tasks and scenarios	23	3.83 (1.0)	4.00 (1.0)
10	The interface of the website is pleasant	24	3.54 (1.2)	4.17 (1.1)
11	I like using the interface of this website	24	3.42 (1.1)	3.96 (1.2)
12	The website has all the functions and capabilities I expect it to have	24	3.38 (1.0)	3.63 (1.2)
13	I believe that the website helped me earn a better grade	23	3.70 (0.9)	3.92 (.09)
14	I would recommend the website to fellow students	24	3.79 (0.9)	4.17 (0.9)
15	Overall I am satisfied with the website	24	3.63 (0.8)	4.13 (0.9)

Table 1: Students' mean scores (standard deviations) in Section 2 of the survey measuring their overall reactions to WReSTT versions 1 and 2.

V2 and consisted of 14 questions (Q2 - Q15), see Table 1. Section 3 consisted of 6 questions (Q16 - Q21) that focused on testing related questions in the context of using WReSTT, see Table 2. Section 4 focused on collaborative learning in the context of WReSTT V2 and consisted of 5 questions (Q22 - Q26), see Table 3. The questions in Section 2 of the survey were adapted from Tullis and Albert [21, pg. 140]. The following Likert scale was used in the survey: *1 = Strongly Disagree, 2 = Disagree, 3 = Neither agree nor Disagree, 4 = Agree, 5 = Strongly Agree.*

Design: The pre/posttest component of the study was conducted on two different groups from two different semesters (Spring and Summer). The pretest was administered prior to the testing part of the CEN 4010 course, i.e., prior to week 8 of the semester, and the posttest was administered at the end of the semester. The students were advised that the pre/posttests were optional and would not impact their grades.

The survey was administered only to the CEN 4010 group at the end of the Summer 2011 semester. The students participating in the study were required to register on WReSTT V1 prior to week 8 of the semester. The instructor then registered the students on WReSTT V2 in week 10 and assigned them to virtual teams, which were the same teams they were assigned to for the course project. The students were given

Q	Testing Related Questions	N	M (SD)
16	The tutorials in WReSTT helped me to better understand testing concepts	24	4.33 (0.7)
17	The tutorials in WReSTT helped me to better understand how to use unit testing tools.	24	4.17 (0.7)
18	The tutorials in WReSTT helped me to better understand how to use code coverage testing tools.	24	3.54 (1.0)
19	The tutorials in WReSTT helped me to better understand how to use functional testing tools.	21	3.33 (1.0)
20	The number of tutorials in WReSTT is adequate	24	4.33 (1.0)
21	I would have used testing tools in my project if WReSTT did not exist.	23	2.91 (1.2)

Table 2: Students' mean scores (standard deviations) in Section 3 of the survey measuring perception of the usefulness of testing tutorials in WReSTT.

Q	Collaborative Learning Related Questions	N	M (SD)
22	The use of virtual points in WReSTT V2 encouraged me to visit the website and complete the tasks	24	4.67 (0.8)
23	The use of virtual points in WReSTT V2 encouraged my team to visit the website and complete the tasks	24	4.71 (0.6)
24	The event stream showing the activities of the other members in the class encouraged me to complete my tasks in WReSTT V2	24	4.38 (0.9)
25	The event stream showing the activities of the other members in the class encouraged my team to complete my tasks in WReSTT V2	24	4.50 (0.8)
26	Our team devised a plan to get the maximum number of points in WReSTT V2.	23	3.96 (1.3)

Table 3: Students' mean scores (standard deviation) of the survey measuring usefulness of the collaborative learning environment in WReSTT

two weeks to become familiar with WReSTT V1 before they received access to V2. Prior to taking the survey, the students were made aware of the virtual points they earned by participating in the activities for WReSTT V2.

4.3 Results and Analysis

To evaluate the effects using WReSTT on students' knowledge and understanding software testing techniques and tools, students were administered a pretest, prior to software testing being covered in class, and a posttest at the end of each semester. For each assessment, students were given a score indicating their performance on the test. A 2 (semester) x 2 (test) ANOVA [16] revealed a main effect of the test ($F(1,46) = 13.56$, $p < .01$), with students performing better on the posttest ($M = 12.27$, $SD = 10.5$) than the pretest. ($M = 7.19$, $SD = 5.1$). Results also revealed a main effect of semester ($F(1,46) = 40.40$, $p < .01$), with students performing better in the Summer semester ($M = 13.78$, $SD = 7.5$) than the Spring semester. ($M = 5.33$, $SD = 4.9$).

Results also found a significant interaction between semester and test ($F(1,46) = 25.47$, $p < .01$). As can be seen in Table 4, students from the Summer term, but not the Spring term, tended to perform better on the posttest than the

		Pre-test		Post-test	
Semester	**N**	**Mean**	**SD**	**Mean**	**SD**
Spring	23	6.22	4.1	4.43	5.6
Summer	25	8.08	6.3	19.48	8.7

Table 4: Students' mean scores and standard deviations on the software testing pre- and post-tests in the spring and summer semesters.

pretest. Follow-up analyses revealed a significant difference between pre/posttest scores in the Summer ($t(24) = -5.01$, $p < .01$), but not in the Spring.

There was an increase in the number of students that were able to identify different testing tools (Appendix A, Q. 4), in the Summer semester from 7/25 to 22/25, there was no change in the Spring semester, 5/23. In addition, during the project presentations at the end of the Summer semester every student team demonstrated the use of at least one testing tool. No student team in the Spring semester demonstrated the use of any testing tool. There was also an increase in the number of students that were able to apply testing techniques (Appendix A, Q. 2) in the Summer semester from 1 to 12, versus the Spring semester from 1 to 3.

Students in the experimental group (Summer semester) were additionally administered a survey at the end of the semester designed to assess their perception of the ease of use and usefulness of WReSTT. A total of 24 students participated in the survey. In Section 1 of the survey 28% of students (N=18) indicated that they had previously used a learning resource other than WReSTT to learn about software testing, while 72% indicated no use of additional resources. Students' overall reactions to both versions of WReSTT were positive, with mean scores on questions in Section 2 of the survey above 3.38 for V1 of the website and above 3.50 for V2. Mean scores for each question in Section 2 for both versions can be found in Table 1.

To compare student reactions to WReSTT V1 and V2, students' responses to each question in Section 2 were analyzed using the Wilcoxon Signed-Ranks Test [19]. Results revealed, for those students whose responses varied, a significant difference between the students' reactions to the two versions of the website for questions 2 ($z=-2.23$, $p<.05$), 4 ($z=-2.07$, $p<.05$), 10 ($z=-2.82$, $p<.01$), 11 ($z=-2.70$, $p<.01$), and 15 ($z=-2.47$, $p<.01$), where students indicated a preference for WReSTT V2 over V1.

Students' perceptions of the usefulness of the tutorials available in WReSTT were also positive, with mean scores above 3.30 for all questions in Section 3. Students' responses to question 21, asking whether they would have used testing tools if WReSTT had not existed, indicated that many would not ($M = 2.91$, $SD = 1.2$). Mean scores for each question in Section 3 can be found in Table 2. Students further indicated that the use of a collaborative learning environment in WReSTT was a motivating factor in their learning, with mean scores above 3.95 on questions 22 through 25 in Section 3 of the survey, see Table 3.

4.4 Discussion

Objective 1. Results indicate that the use of WReSTT as a teaching tool for students learning software testing can significantly improve their understanding and use of software testing techniques and tools. Students in the Summer semester, who used WReSTT, performed significantly better on their post-tests than students in the Spring semester, who did not use WReSTT. There was no significant difference between their pre-test scores, indicating that this difference was not due to differences in students' knowledge before taking the course or the semester. One confounding variable, however, is a possible effect of the instructor. The Spring and Summer courses were taught by different instructors, which may have influenced the results. We plan to repeat the study using the same instructor for the control and treatment groups.

Objective 2. Results indicate that students do find WReSTT a useful resource for learning software testing 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. Students also indicated that both versions of WReSTT helped them complete course tasks and earn a better grade in the class. Regarding ease of use, students indicated that both versions of WReSTT are easy to use and pleasant, and they would recommend use of the sites to other students.

Objective 3. Results indicate that students do find that the use of a collaborative learning environment in WReSTT motivates them to complete assignments. Specifically, students reported that the use of virtual points and event streaming encouraged them and their fellow team members to complete tasks in WReSTT.

5. CONCLUDING REMARKS

In this paper we describe the recent updates made to the *Web-Based Repository of Software Testing Tools* (WReSTT) and the results from the first comprehensive study to determine the impact WReSTT has on testing pedagogy. The study was performed using two software engineering (SE) classes, Spring and Summer 2011. 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 assignments. We plan to perform additional studies at different academic institutions to further validate our results.

6. ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under grants DUE-0736833 (FIU) and DUE-0736771 (FAMU). We would like to thank Dionny Santiago and Yesenia Sosa for their work on WReSTT V2. We also appreciate the insightful comments provided by the reviewers on how to improve the paper.

7. REFERENCES

[1] P. J. Clarke, A. A. Allen, T. M. King, E. L. Jones, and P. Natesan. Using a web-based repository to integrate testing tools into programming courses. In *Proceedings of the ACM OOPSLA 2010 Companion*, SPLASH '10, pages 193–200, New York, NY, USA, 2010. ACM.

[2] P. J. Clarke, J. Pava, Y. Wu, and T. M. King. Collaborative web-based learning of testing tools in se

courses. In *Proceedings of the 42nd SIGCSE Conference*, pages 147–152, New York, NY, USA, 2011. ACM.

[3] CNSS. Software 2015: A national software strategy to ensure u.s. security and competitiveness. Technical report, Center for National Software Studies, 2005.

[4] Cobertura Team. Cobertura, May 2010. `http://cobertura.sourceforge.net/`.

[5] C. Desai, D. S. Janzen, and J. Clements. Implications of integrating test-driven development into cs1/cs2 curricula. *SIGCSE Bull.*, 41(1):148–152, 2009.

[6] T. Dvornik, D. Janzen, J. Clements, and O. Dekhtyar. Supporting introductory test-driven labs with WebIDE. In *In Proceedings of the 24th IEEE-CS CSEET Conference*, pages 51 –60, may 2011.

[7] S. H. Edwards. Using software testing to move students from trial-and-error to reflection-in-action. In *Proceedings of the 35th SIGCSE Conference*, pages 26–30, New York, NY, USA, 2004. ACM.

[8] Facebook Team. Facebook, 2011. `http://www.facebook.com/`.

[9] E. Gamma and K. Beck. JUnit, 2008. `http://www.junit.org/`.

[10] V. Garousi. An open modern software testing laboratory courseware - an experience report. In *In Proceedings of the 23rd IEEE CSEET Conference*, pages 177 –184, March 2010.

[11] D. S. Janzen and H. Saiedian. Test-driven learning: intrinsic integration of testing into the cs/se curriculum. *SIGCSE Bull.*, 38(1):254–258, 2006.

[12] E. L. Jones. Integrating testing into the curriculum — arsenic in small doses. *SIGCSE Bull.*, 33(1):337–341, 2001.

[13] C. Kaner. Center for software testing education and research (CSTER), 2010. `http://www.testingeducation.org/`.

[14] T. C. Lethbridge, J. Diaz-Herrera, R. J. J. LeBlanc, and J. B. Thompson. Improving software practice through education: Challenges and future trends. In *FOSE '07: 2007 Future of Software Engineering*, pages 12–28, Washington, DC, USA, 2007. IEEE Computer Society.

[15] A. Meneely, L. Williams, and E. F. Gehringer. Rose: a repository of education-friendly open-source projects. In *In Proceedings of the 13th ITiCSE Conference*, ITiCSE '08, pages 7–11, New York, NY, USA, 2008. ACM.

[16] J. Neter, W. Wasserman, and M. H. Kutner. *Applied linear statistical models*. Irwin Press, Boston, 1990.

[17] Realsearch Group. ROSE: Repository for Open Software Education, 2011. `http://realsearchgroup.com/rose/`.

[18] Realsearch Research Group. iTrust, 2011. `http://agile.csc.ncsu.edu/iTrust/`.

[19] S. Siegel and N. J. Castellan. *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill, New York, second edition, 1988.

[20] Software Quality Engineering Research Group (SoftQual). SoftQual Repository of Software Testing Laboratory Courseware, 2011. `http://www.softqual.ucalgary.ca/projects/testing_labs/`.

[21] T. Tullis and B. Albert. *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics*. Elsevier: Morgan Kaufmann, 2008.

[22] Ultimate Software. SWAT, 2009. `http://sourceforge.net/projects/ulti-swat/`.

[23] WebIDE Development Team. WebIDE Labs, 2011. `http://web-ide.org`.

[24] WReSTT Team. Web-based Repository for Software Testing Tools, 2010. `http://wrestt.cis.fiu.edu/`.

APPENDIX

A. PRETEST/POSTTEST

This test will NOT impact your course grade.

ID Number (DO NOT use your name):

1. What is the main objective of program testing?

2.(a) Identify all the testing techniques you have used to create test cases:

(b) Write test cases to test the code provided below. Identify the testing technique used to generate the test case.

```
//Method to withdraw money from a bank account
//The following variables are defined as follows:
//requiredMin = 50.0 and balance = 100;
public void withdraw (double amount) {
 if ((balance - amount) < requiredMin)
  System.out.println("Insufficient funds");
 else
  balance = balance - amount;}
```

Input Value	Expected Output Values			Testing Technique
amount	amount	balance	requiredMin	

3. Have you ever used tools to support testing of programs? Circle your answer: Yes No

4. If you have answered "Yes" to Question (3), answer the following questions:
   a. List the names of the tools you have used:
   b. List one tool in each of the following categories that you have used and indicate your level of proficiency corresponding to each tool on a scale of 1 - 5 with, *1 = barely competent* and *5 = extremely proficient*:

	Category	*Tool(s)*	*Proficiency*
i	Unit Testing		
ii	Functional Testing		
iii	Code Coverage		

5. Do you know of any online resources that provide information on software testing? Yes No

6. If you have answered "Yes" to Question (5), answer the following questions:
   a. State the names of online resources.
   b. State the type of materials (notes, lab exercises, tutorials etc.) found at each resource listed above.

7. How beneficial do you think it is to use tools to support testing of your programming assignments? Use a scale of 1 - 5 with *1 = Not beneficial* and *5 = Extremely beneficial*

8. If you answered 2 or above to Question (7), state one reason for your answer.