Collaborative Web-Based Learning of Testing Tools in SE Courses

Peter J. Clarke, Jairo Pava, Yali Wu
School of Computing and Information Sciences
Florida International University
Miami, FL 33199, USA
{clarkep, jpava001, ywu001}@cis.fiu.edu

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

ABSTRACT

One of the main concerns in the software industry continues to be the development of high quality software. This concern will be exacerbated as software systems become more complex. The training of software developers continues to grow in academia since more institutions are offering software engineering (SE) courses. However, the list of topics that are expected to be covered in this course leaves little or no time for topics that focus on developing quality software, such as software testing and the use of testing tools.

In this paper we describe an approach that non-intrusively integrates the use of software testing tools in SE courses. The cornerstone of our approach is the interaction students have with a *Web-Based Repository of Software Testing Tools* (WReSTT) that contains tutorials on testing concepts and testing tools. WReSTT employs both collaborative learning and social networking features that are attractive to students. We present the results of preliminary study performed in two SE courses that show how using the resources in WReSTT can potentially impact the students' understanding of software testing and the use of testing tools.

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

One of the main concerns in the software industry continues to be the development of high quality software. This concern will be exacerbated as the size and complexity of software systems increase. The training of software developers continues to grow in academia since more institutions are offering software engineering (SE) courses. The literature states that most software engineers in industry are not educated in key portions of the SE body of knowledge, including testing [2, 19].

The list of topics that are expected to be covered in SE courses leaves little or no time for topics that focus on developing quality software, such as software testing [2]. Testing continues to be the primary technique used to ensure the development of high quality software, but recent studies [6] indicate that major improvements in software testing are needed. Therefore it is important that innovative ways be developed to encourage students to learn about testing and testing tools without extending the length of the already crowded curriculum. Although we focus on SE courses in this paper, the problem of developing quality software should be addressed in courses earlier in the computing curriculum e.g., CS1-CS3. Clarke et al. [5] describe a three-phase project that supports the integration of testing into programming courses supported by a *Web-Based Repository of Testing Tools* (WReSTT). WReSTT contains tutorials on testing concepts and testing tools.

In this paper we describe an approach that non-intrusively integrates the use of software testing tools in SE courses. The cornerstone of our approach is the interaction students have with WReSTT. Unlike the original version of WReSTT describe in [5] the updated version (WReSTT V2) to employ the use of both collaborative learning and social networking features that are attractive to students. WReSTT V2 is currently in the beta stage and is still being updated to reflect the improvements recommended by the students in the study. We present the results of a preliminary study performed in two SE courses that use WReSTT V2 as a learning resource. The results show how using the resources in WReSTT can potentially impact the students' understanding of software testing and the use of testing tools. We acknowledge that a more comprehensive study is required before any major claims can be made regarding the impact of using WReSTT in the classroom.

The paper is organized as follows: Section 2 provides a review of the approaches and online resources that support the integration of testing into programming courses. Section 3 describes the structure of WReSTT and its use as a learning resource. Section 4 describes the study and presents the results. Section 5 presents the concluding remarks.

SIGCSE'11, March 9–12, 2011, Dallas, Texas, USA.

2. RELATED WORK

In this section we review some of the approaches used to integrate testing into courses that contain a significant portion of programming, more specifically SE courses. A brief overview for some of the common online resources that can support student learning in software testing is also presented.

2.1 Integrating Testing into Programming

Frezza [11] describes an approach that integrates testing into an introductory software design course. The approach was tried in several instances of the course and the students rated the approach as effective. However, the approach was not able to make up the deficiencies in student programming ability. Barbosa et al. [3] tackle two prominent challenges in software education, (1) making students recognize the relevance of the testing activity, and (2) motivating students to use testing ideas in their projects. The approach, used in an object-oriented development, course was based on the idea of test-driven development. The results of the work indicated that students generally approved of the testing early approach

Janzen et al. [16] present test-driven learning (TDL) as a pedagogical tool that can be incorporated into multiple levels of the CS and SE curricula. They propose that TDL can be applied early in the CS curriculum and should not compete with other approaches used to teach early programming courses. A survey was conducted using a CS2 course, undergraduate SE course, and a graduate SE course. The results indicated that less experienced students are more open to adopting a test-first approach.

Astigarraga et el. [2] present a study that evaluates software testing as an industry profession and discusses ongoing efforts to advance the status of software testing in academic curricula. The conclusion of the paper recommends changes to the curricula that focus more on test courseware, journals, societies and education opportunities for software test engineering. It is well known in academia that such changes will take a long time to be realized; therefore we propose integrating testing into SE courses by using available web-based resources, collaborative learning, and social networking to attract students to software testing.

2.2 Online Resources

There are several web-based repositories that provide access to learning materials on software testing. We provide a brief description of these repositories below.

CSTER [18]. CSTER, The Center for Software Testing Education and Research, provides materials to support the teaching and self-study of software testing, software reliability, and quality-related software metrics. CSTER has a large repository of content on software testing including video lectures, practice quizzes, drills, and other assessment materials.

OpenSeminar - Module 7 [26]. OpenSeminar is a web-based open courseware platform that enables instructors to collaborate on material for similar courses by sharing links to content. The testing component in OpenSeminar is located in model 7 of the software engineering component, and provides users with examples, lab exercises, lectures (slides) and readings.

Bug Hunt [8]. Bug Hunt is an online tutorial to instill good testing practices through lessons and challenges. Bug Hunt's tutorials provide instructors and students with several features including: immediate feedback, self-paced progression, configurability to accommodate instructor's requirements, and provision of an automatic assessment of student performance.

Ensemble [9]. Ensemble is a NSF NSDL Pathways project to support the distribution of computing education materials. Ensemble provides access to other repositories such as SWENET [20] and CITIDEL [24]. SWENET, The Network Community for Software Engineering Education, is a project to produce and organize high-quality materials supporting software engineering education. One of the Software Engineering Education Knowledge (SEEK) modules in SWENET is Software Verification and Validation (VAV). CITIDEL, the Computing and Information Technology Interactive Digital Educational Library is a resource to discover Computer Science education and research materials.

MERLOT [21]. MERLOT, Multimedia Educational Resource for Learning and Online Teaching, is an online, user-centered, and searchable collection of peer reviewed and selected higher education learning materials. MERLOT is not exclusive to software testing. However, testing material can be found by searching with the browse path "All>Science and Technology>Computer Science" and keywords "Software Testing".

3. USING WRESTT IN THE CLASSROOM

The *Web-Based Repository of Software Testing Tools* - (WReSTT) [27], is one of the components of a project consisting of three phases to integrate software testing into programming and software engineering courses. The three phases include: (1) developing an online portal of learning resources that supports pedagogy in the area of software testing, particularly the use of testing tools; (2) holding a series of annual workshops for instructors that introduce them to software testing techniques and tools available through WReSTT, and (3) performing studies to determine the effectiveness of WReSTT and the instructors workshop on improving student learning. Clarke et al. [5] describe WReSTT and report on the first instructor's workshop. In this paper we focus on integrating collaborative learning into WReSTT and report the preliminary results of a study performed in two SE courses.

3.1 Overview of WReSTT

WReSTT [27] was developed to support the pedagogical needs of students and instructors in programming and software engineering courses by providing access to a comprehensive and up-to-date set of learning material on software testing tools. Although WReSTT focuses on software testing tools, we also provide links to other repositories containing software testing materials. WReSTT was developed using Drupal [7], a content management system. WReSTT currently contains learning materials for several tools, including: *JUnit* [12] - a unit testing framework and *SWAT* [25] - Simple Web Automation Toolkit, among others. Clarke et al. [5] provide a more complete description of the first version of WReSTT.

3.2 Integrating Collaborative Learning

In Fall 2009 WReSTT was introduced to the students in the undergraduate SE course at Florida International University (FIU) to encourage them to use testing tools to

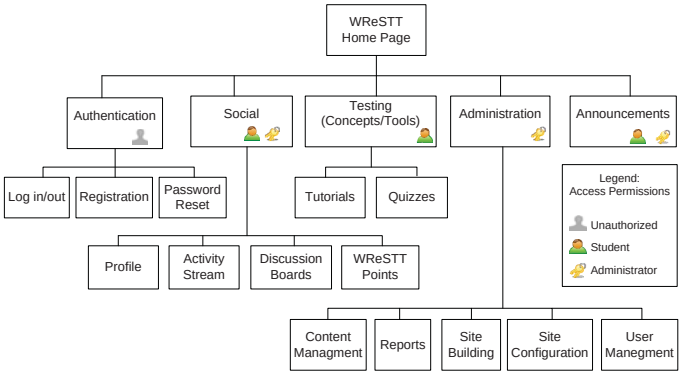

Figure 1: Block diagram of WReSTT version 2.

test the software developed for their class projects. Upon completion of the academic semester one of the students in the class, Jairo Pava (second author of this paper), recommended to the WReSTT team that students would be more willing to use WReSTT if (1) the interface to WReSTT was improved to have more of a social networking tool feel e.g., Facebook [10], and (2) there was a competition for bonus points between project teams to access and complete quizzes based on software testing tutorials. Jairo was invited to be part of the WReSTT team and assisted in the development of a new version of WReSTT that is still being beta tested.

The process of acquiring knowledge, or learning, is largely enhanced by social activity. If there is no interaction, students often feel isolated, overwhelmed, and reluctant to learn [14]. As a result, there has been a great rise in the use of Online Learning Communities (OLC) for instruction through academic and professional training programs on the Internet. An OLC describes a group of people who are dedicated to learning together in an online environment that encourages dialogue, feedback, and reflection. Members of an OLC may be students, professors, tutors, or domain experts who engage in meaningful conversations and work in teams towards a common purpose or objective [15].

A successful OLC uses positive reinforcement, presents new information in motivating ways, and encourages group collaboration. These characteristics correspond specifically to the behavioral, cognitive, and social learning theories which have been empirically proven to be successful for instruction [17]. To increase student motivation, participation, and retention, WReSTT was transformed into a collaborative OLC.

3.3 WReSTT V2 Design

The second version of WReSTT (WReSTT V2) is an OLC and incorporates features of common social networking tools. These improvements are expected to be more attractive to students and will encourage them to use the site more often and promote collaborative learning. Figure 1 shows a hierarchical block diagram of the components in WReSTT V2. The major components are as follows:

- *Authentication* - allows users to log in/off, register for an account, and reset their passwords.
- *Social* - allows users to create a profile, monitor the activity stream, post comments to the discussion boards, and monitor virtual points assigned to users or user teams.
- *Testing* - contains the tutorials and quizzes for testing concepts and the tools hosted in the repository.

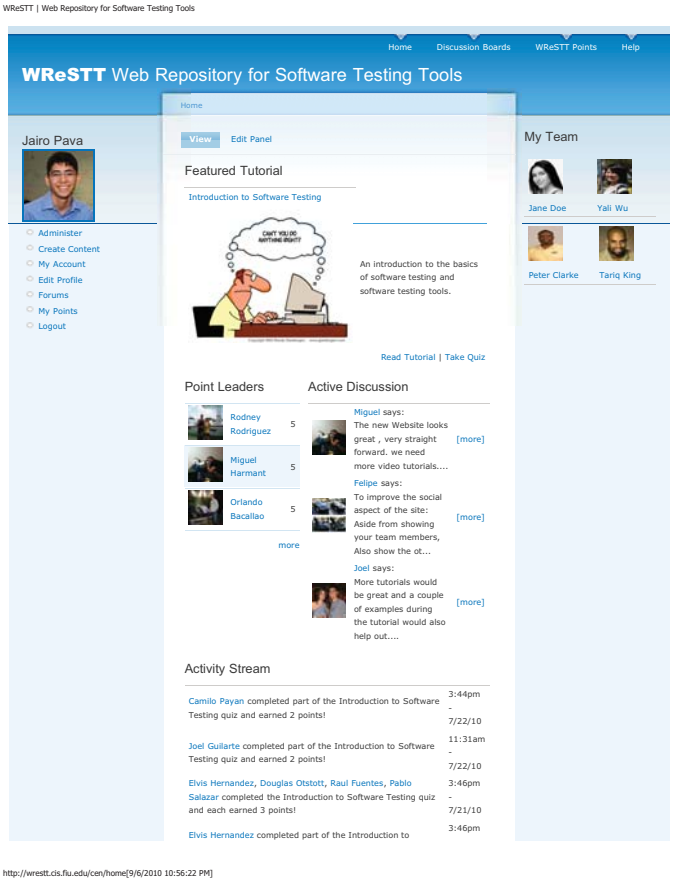

Figure 2: Student's homepage in WReSTT V2.

- *Administration* - provides the basic operations to maintain the site including: content management, generation of reports, extending the site, configuration of the site, and management of user accounts.
- *Announcements* - distribute announcements to the registered users of the site.

Building and configuring the Social and the Testing components are the innovative components of the site. Drupal [7], the content management system used to build the site, provides most of the functionality in the other components. Note, however, that these components still need to be configured correctly for WReSTT V2 to function properly.

Figure 2 shows the home page for one of the students currently registered on WReSTT V2. On the left side of the page is the picture of the student and the main menu. The right side of the page shows the student's team members and their pictures. The main block in the center of the page is divided into four parts (1) *Featured Tutorial* - the current tutorial available to the team, there are two links available: one to access the tutorial and the other to take the quiz; (2) *Point Leaders* - the students currently with the most points assigned by the WReSTT V2 system, (3) *Active Discussion* - comments posted by students, and (4) *Activity Stream* - posting showing the recent activities by the users of the system.

3.4 Using WReSTT in an SE Course

The process an instructor would follow to use WReSTT V2 is as follows. At the beginning of the academic semester, the instructor uploads the course roster onto the WReSTT website and places each of the students into virtual WReSTT

teams. Each team may contain up to six students. There is no limit to the amount of teams that participate on the WReSTT website. After uploading the roster and forming teams, accounts on the WReSTT website are created, and an e-mail with a WReSTT username and password is delivered to each student. The instructor may then make the tutorials on software testing tools available to the teams either all at once or as the course progresses through various testing topics.

Each tutorial is followed by a collaborative quiz which all of the students on a virtual team must complete with a grade of 80% or higher to receive virtual points. The quizzes are divided into as many individual parts as there are students in a team. Each student must complete a quiz part using his or her own WReSTT account. A student is rewarded with virtual points for completing an individual quiz part, but must work with his or her team members to ensure that everyone on the team passes their individual quizzes to receive the maximum amount of virtual points. The more tutorials a virtual team reads, the more quizzes it is allowed to complete, and the more opportunities each of its members has in accumulating virtual points. WReSTT maintains a count of each of the students' earned virtual points so that the instructor may use these points as extra-credit points towards the students' grades.

Quizzes on WReSTT are divided into two parts. The first part consists of multiple choice questions where the students may use a combination of HTML form elements such as radio buttons, checkboxes, and lists to submit answers that are automatically graded. Students receive immediate feedback on answers and are automatically assigned virtual points from multiple choice questions. The second part consists of short response questions where students are asked to reflect on a concept, and the responses are used to stimulate dialogue on the discussion boards. WReSTT maintains a count of the posts each student makes on the discussion boards and recommends to the instructor the amount of extra virtual points a student should earn.

As a virtual team successfully completes quizzes and is awarded points, other teams are notified of the team's accomplishments through an activity stream on the front page of WReSTT. The names and pictures of the students are posted on the front page along with the name of the tool they have now learned about and the amount of points they have earned as a result of successfully completing the quizzes. A points leader board on the front page is updated to show the name of the student in the class with the highest amount of points on WReSTT to promote a competitive social environment. The students benefit from the competitive environment as they now have more motivation to learn from software testing tutorials, take quizzes, and earn points.

4. PRELIMINARY STUDY

To determine the effectiveness of using WReSTT in the classroom we conducted a preliminary study that involved students in the *CEN 4010 - Software Engineering I* course for the spring 2010 and summer 2010 semesters. The students were exposed to both the WReSTT V1 implementation to access the tool tutorials and the WReSTT V2 implementation for collaborative learning. Based on the results of the study we expect to merge both versions of WReSTT. One of the main components in the SE course is the semester long project. To achieve the learning outcomes of the course students are required to work in project teams, which lends itself to collaborative learning.

Class of Data	**Spr. 2010**	**Sum. 2010**
Number of students that participated in the study	22 of 38 (58%)	32 of 34 (94%)
Students that received points	18 (82%)	30 (94%)
Students that received maximum points	2 (9%)	7 (22%)
Students that completed a quiz	22 (100%)	31 (97%)
Students that passed the quiz	17 (77%)	25 (78%)
Students that received maximum points on the quiz	9 (41%)	19 (59%)
Number of teams in class	7	7
All members of team that completed the quiz	1 (14%)	3 (43%)

Table 2: Data collected from WReSTT V2.

There were two parts to the study used to evaluate the improvement of students' understanding of software testing and use of testing tools. The first part focused on the impact WReSTT V1 had on the students' ability to use testing tools in their software projects and be able to demonstrate the use of the tools to the instructor. The second part of the study focused on the impact WReSTT V2 had on promoting collaborative learning and improving the students' conceptual understanding in the area of software testing.

4.1 Data Capture

The data for the study was collected from two sources. The first source was the pre/post test instruments given to the class prior to the testing component of the course being taught, and upon completion of the course, respectively. In addition to the pre/posttest instrument, there was also data collected from the grading rubric to evaluate the students' proficiency of testing tools during the demonstration of the software projects. The second source of data was from the WReSTT V2 web site.

The pre/posttest contained four classes of questions: Q(1)-(3) focused on program testing and the use of testing tools, Q(4)-(5) on online resources available to support testing, and Q(6)-(7) assessed the importance of tool support for software testing. The leftmost column of Table 1 shows the closed ended questions from the pre/post test instrument. During the demonstration of the project each student was asked to demonstrate their knowledge of a testing tool if they stated that one or more tools were used during testing.

The data collected from the WReSTT V2 was in the form of WReSTT virtual points. These points were awarded to students as follows: 2 points for completing an individual quiz part; 3 points for being on a team that had all of its members pass their individual quizzes; 1 point for uploading a profile picture; and 1 point for posting on the discussion board. Teams were also awarded bonus points based on the team with the highest point tally in the quizzes and the promptness in completing the quizzes.

4.2 Results

Table 1 shows a summary of the results for the pre/post test for the two semesters (spring and summer 2010). Column 1 shows the questions that required discrete answers. Questions 3.b.i, and 6 used a Likert scale ranging from 1 (low) to 5 (high). Row 1 of the tables shows the number of responses of the students that took both the pretest and posttest. In the spring semester 18 out of 38 (47%) took both tests and in the summer 24 of 34 (71%).

Question	**CEN 4010 - Spring 2010**		**CEN 4010 - Summer 2010**	
	Pretest	*Posttest*	*Pretest*	*Posttest*
Number of Responses:	18 out of 38 (47%)		24 out of 34 (71%)	
2. Have you ever used tools to support testing of programs?	N(15) Y(3)	N(6) Y(12)	N(20) Y(4)	N(6) Y(18)
3.b.i Unit Testing Tool Proficiency	Avg = 4 (1 response)	Avg = 1.7 (8 responses)	Avg = 2.5 (2 response)	Avg = 3.2 (13 responses)
3.b.ii Functional Testing Tool Proficiency	(0 response)	(0 responses)	Avg = 3 (1 response)	Avg = 2.7 (3 responses)
3.b.iii Code Coverage Tool Proficiency	(0 responses)	Avg = 3 (4 responses)	Avg = 3 (1 response)	Avg = 2.0 (6 responses)
4. Do you know of any online resources that provide information on software testing?	N(11) Y(7)	N(5) Y(13)	N(21) Y(3)	N(2) Y(22)
6. How beneficial do you think it is to use tools to support the testing of programs?	Avg = 4.3	Avg = 4.6	Avg = 4.4	Avg = 4.4

Table 1: Results for the closed ended questions in the pretest/posttest instrument. N – No; Y – Yes; Avg – average of scores are out of 5.

The data in Table 1 shows that there was a 50% inrease in the number of students who used testing tools during their software project from the spring to the summer semester. There was also an increase in the number of students using the different types of tools e.g., unit and functional testing tools and code coverage tools. Note however that the highest tool usage (54% of the students) was the unit testing tool at the end of the summer semester. There was a significant improvement in the number of students that became aware of online resources with information on software testing. This was expected since the students were introduced to WReSTT during the class. The last row of table 2 shows that students in both classes generally agreed that it is beneficial to use testing tools to test programs.

The data in the table with respect to the students' knowledge and skill of using testing tools was validated when the students demonstrated their SE project to the instructor. That is, unlike other SE classes taught in the past students were able to describe the process of using an automated tool during testing. The students understood concepts such as test setup/tear down, and why it was necessary to explicitly state the expected results of a test. One of the interesting aspects of the study was the variety of testing tools used by the teams. These included unit testing tools - FlexUnit [1], JUnit [12], MbUnit [23], PHPUnit [4], Visual Studio Team System 2008 [22]; functional testing tools - SWAT [25]; code coverage tools - NCover [13]. The most popular tools were JUnit, MbUnit and PHPUnit.

Table 2 contains the data captured from the part of the study that used WReSTT V2. Column 1 shows the class of the data collected, Columns 2 and 3 contain the data for the spring and summer semesters, respectively. The percentages presented in Rows 2-6 are based on the students that participated in the study. Of the 38 students in the spring class only 22 (58%) participated in the WReSTT V2 study, and 32 of the 34 (94%) students participated during summer. The students who participated in the study and performed tasks to be awarded virtual points were over 80% in both semesters. Similarly, 77% and 78% of the students passed the quiz in the spring and summer semesters, respectively. The number of teams that completed the quiz was 14% and 43% for the spring and summer, respectively. There was an increase in teams completing the quiz of 29% from spring to summer. Out of the seven teams in both semesters, one team completed the quiz in spring and three teams in the summer. Based on the results in Table 2, it can be deduced that the main hurdle to overcome is to get all students to participated in the study. If all students do not participate then some teams will not be able to get the maximum number of virtual points.

4.3 Discussion

The results shown in Tables 1, 2 and the anecdotal evidence from the SE project demonstrations, support the claim that using WReSTT in SE classes exposes students to testing tools and improves their practical skills in testing software systems. Prior to the studies related to WReSTT, students in the SE classes at FIU did not use any automated testing tools to test their software projects. One of the main concerns regarding the study is that not enough students participated. One reason for the low participation by students was that the study commenced two-thirds of the way into the semester, this is when testing is introduced into the SE class. Around this time some students were not attending class due to deadlines for projects and exams in other classes. In both the spring and summer classes, the percentage of student participation in the study was very similar to the percentage of the student attending class.

The results collected from the WReSTT V2 component of the study are very promising, particularly during the summer semester. The participation was high, over 80% in both classes, and almost all students took the quiz online. The precentage of the students that passed the quiz was also good, around 77%. Of the students who participated in the study, 41% of the students in the spring semester and 59% of the students in the summer semester obtained full points on the quiz. Note that the percentage of teams where all members completed the quiz was low (14% and 43%) due to the fact that some students did not participate in the study or did not complete the quiz in the allocated time. It is evident that using the collaborative learning approach in WReSTT is a factor that can impact students' conceptual understanding of software testing, particularly if bonus points are awarded based on the WReSTT virtual points.

Threats to Validity. The spring SE class was taught by a new instructor with little expertise in software testing and did not engage the students in using testing tools during the SE class. This may have also contributed to the low participation of the students in the study. This problem was rectified in the summer SE class with a more experienced SE instructor who is knowledgeable in the area of software testing. No control groups were used during the preliminary study since there was only one section of CEN 4010 being offered in each semester and it was important to expose all of the students in the course to the testing tools. We plan

to repeat the study in the coming semesters when WReSTT is fully implemented and using a control group.

5. CONCLUDING REMARKS

In this paper we describe how the learning resources in a *Web-Based Repository of software Testing Tools* (WReSTT) may be used to support the teaching of testing in software engineering (SE) courses. The learning resources in WReSTT include tutorials on software testing concepts and testing tools. The tutorials in WReSTT utilize a collaborative learning approach where students are grouped into virtual teams and each team is awarded virtual points based on the completion of specific tasks, such as quizzes. WReSTT contains social networking features that allow students to create profiles, monitor the activities of fellow students and identify who are the points leaders in their class.

A preliminary study was performed with two SE classes, spring and summer 2010, to evaluate the effectiveness of using WReSTT in the classroom. The results showed that (1) most students have never been exposed to an online resource containing software testing learning materials before using WReSTT; and (2) integrating collaborative learning into WReSTT, with the reward of virtual points, can potentially improve students' conceptual understanding of software testing and the use of testing tools during an SE course.

6. ACKNOWLEDGEMENTS

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 Dr. Edward L. Jones (FAMU) and the reviewers for their insightful comments on how to improve the paper.

7. REFERENCES

[1] Adobe Open Software. FlexUnit, 2010. http://docs.flexunit.org/index.php.

[2] T. Astigarraga, E. Dow, C. Lara, R. Prewitt, and M. Ward. The emerging role of software testing in curricula. In *Transforming Engineering Education: Creating Interdisciplinary Skills for Complex Global Environments, 2010 IEEE*, pages 1–26, apr. 2010.

[3] E. Barbosa, J. Maldonado, R. LeBlanc, and M. Guzdial. Introducing testing practices into objects and design course. In *Proceedings of CSEET 2003*, pages 279 – 286, mar. 2003.

[4] S. Bergmann. PHPUnit, 2010. http://www.phpunit.de/.

[5] 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.

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

[7] Drupal Community. Drupal, 2008. http://drupal.org/.

[8] S. Elbaum and K. Stolee. Bug Hunt , 2010. http://esquared.unl.edu/BugHunt.

[9] Ensemble Development Team. Ensemble - Connecting Computing Educators, 2010. http://www.computingportal.org/.

[10] Facebook Team. Facebook, 2010. http://www.facebook.com/.

[11] S. Frezza. Integrating testing and design methods for undergraduates: teaching software testing in the context of software design. In *Frontiers in Education, 2002. FIE 2002. 32nd Annual*, volume 3, pages S1G–1 – S1G–4 vol.3, nov. 2002.

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

[13] Gnoso. Ncover, 2010. http://www.ncover.com/.

[14] N. Hara and R. Kling. Students' Frustrations with a Web-based Distance Education Course, September 1999. http://www.slis.indiana.edu/CSI/wp99_01.html.

[15] S. Hiltz. Collective learning in asynchronous learning networks. In *Proceedings of WEB 1998*, Orlando, Florida, USA, 1998.

[16] 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.

[17] S. D. Johnson and S. R. Aragon. An instructional strategy framework for online learning environments. In *Education*, pages 31–43, 2002.

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

[19] 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.

[20] M. J. Lutz, W. M. McCracken, S. Mengel, M. Sebern, G. W. Hislop, and T. B. Hilburn. Swenet - seek category: Software verification and validation (vav), 2010. http://www.swenet.org/browseModules.aspx?categoryID=11.

[21] MERLOT Community. Multimedia educational resource for learning and online teaching (MERLOT), 2010. http://www.merlot.org/merlot/index.htm.

[22] Microsoft Corporation. Visual Studio Team System 2008, May 2010. http://msdn.microsoft.com/en-us/library/ee338734(v=VS.90).aspx.

[23] NUnit.org. MbUnit, May 2010. http://www.mbunit.com/.

[24] C. D. Team. CITIDEL - Computing and Information Technology Interactive Digital Educational Library, 2010. http://www.citidel.org/.

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

[26] L. Williams. Openseminar: Software engineering: Testing, 2010. http://openseminar.org/se/modules/7/index/screen.do.

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