ASEE's 123rd Annual • Conference & Exposition • New Orleans, LA • June 26-29, 2016

New Orleans JAZZED about Engineering Education

Paper ID #15885

Integrating Software Testing to CS Curriculum Using WRESTT-CyLE

Dr. Yujian Fu P.E., Alabama A&M University

Dr. Yujian Fu is an associate professor of computer science department at Alabama A&M University. Her research interests fall in formal verification of cyber physical systems, behavioral analysis of mobile security, software architecture and design analysis of safety-critical and mission-critical systems. Her projects are supported by NSF, Air Force and DoD. She have several publications regarding to the research and educational projects.

Dr. Peter J. Clarke, Florida International University

Peter J. Clarke received his B.Sc. degree in Computer Science and Mathematics from the University of the West Indies (Cave Hill) in 1987, M.S. degree from SUNY Binghamton University in 1996 and Ph.D. in Computer Science from Clemson University in 2003. His research interests are in the areas of software testing, software metrics, model-driven software development, domain-specific modeling languages, and computer science education. He is currently an associate professor in the School of Computing and Information Sciences at Florida International University. He is a member of the ACM (SIGSOFT, SIGCSE, and SIGAPP); IEEE Computer Society; and a member of the Association for Software Testing (AST).

Prof. Nelson Barnes Jr, Alabama A&M University

Prof. Barnes has performed a variety of technical projects including: Providing system specifications, hardware/software compatibility assessments, software design, software development and integration expertise for an assortment of systems; Developing goals and objectives for business process management and technology integration; Ensuring reliable software integration across multiple platforms for various programs; and Architecture development for an assortment of Service Oriented Architecture (SOA) and Web Application systems. His research interests are in the areas of Mobile Device Development, Software Engineering, Software Project Management, and Information Systems Analysis.

Integrating Software Testing into Computer Science Curriculum Using WReSTT-CyLE and Learning Objects

Yujian Fu[1], Nelson Barnes[1], and Peter Clarke[2]
[1]Alabama A&M University
4900 Meridian Street
Normal AL 35762
[2]Florida International University
11200 SW 8th Street
Miami, FL 33199
Telephone: (305) 348-2440

Abstract

A learning object repository is a digital library developed by a group of educators and researchers to store both context and/or content while sharing, managing and reusing this resource. This idea aims at making the knowledge units interchangeable with assessment forms in a standard way so that evaluating learning outcomes and teaching strategies results in greater educational benefits. WReSTT-CyLE is a cyber-enabled virtual learning environment that provides students and educators with information on software testing, supports various types of teaching materials in the form of learning objects (LOs), and facilitates social and media networking and peer study environments. Virtual Learning Environments has become a major field of interest in recent years, with the integration of multiple social and media networking applications and has dominated in current academic learning environments and peer learning support resources. In this paper, we present a computer science course study of multiple subjects using WReSTT-CyLE to teach software-testing concepts. Software testing is considered a high-level concept and is not widely offered in many computer science programs. WReSTT-CyLE is a learning resource that can be used by students and instructors to improve their knowledge of software testing techniques and testing tools. A study was performed at Alabama A&M University to determine the impact using WReSTT-CyLE had on students' knowledge of software testing.

Introduction and Motivation of This Project

Software testing is one of the key methodologies for quality assurance and is applied through the software development cycle [1]. With serious concerns surrounding the qualitative and quantitative analysis of large-scale software intensive systems, the development of effective software-testing techniques that cover various domain types has flourished, ranging from home surveillance systems to real-time monitoring systems, from service robotics to space craft systems. In comparison to the rapid growth in testing techniques, a question raised to educators is how can these new testing techniques be integrated into curricula so that students can have a better understanding of software testing concepts be able to apply different techniques to projects? Answering this question would help us build the next generation of technology-based worker that is both knowledgeable and pragmatically effective in the area of software development. Currently, there is a lack of software testing concepts in computer science education and curricula, based on a quick survey of several institutions' websites.

These issues create a challenge for instructors and educators who would like to incorporate software testing concepts into their computer science curriculums. A student-centered learning model is illustrated in Figure 1, in which both cyber-enabled learning and technology-based learning are integrated seamlessly through the use of the WReSTT-CyLE platform. WReSTT-CyLE is a cyber-enabled virtual learning environment that provides students and educators with information on software testing, supports various types of teaching materials in the form of learning objects (LOs), and facilitates social and media networking and peer study environments.

Motivational and student-centered teaching methods are effective innovations for improving student-learning outcomes based on research studies. From those students who are already mentally and self-motivated, we can easily observe results from any number of approaches. However, for many students, the process of motivation needs to be considered from the beginning of the learning process.

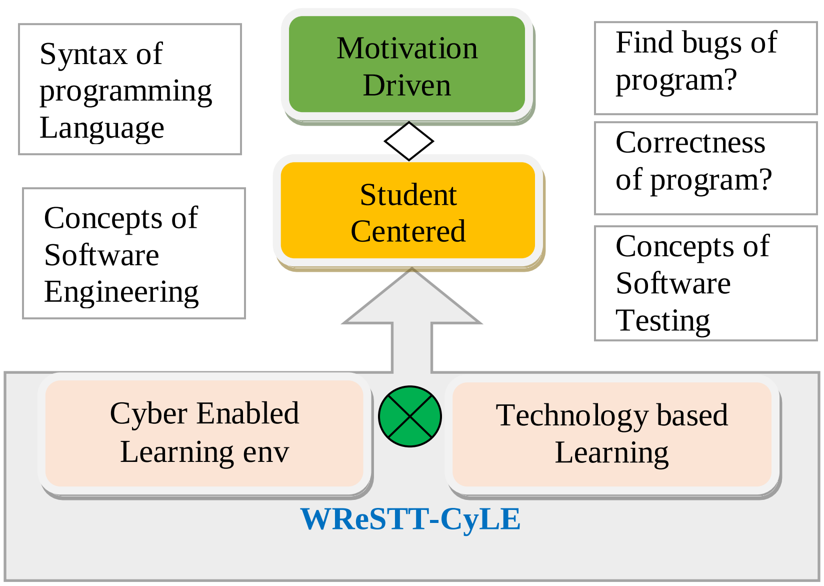

Figure 1. The Student-centered Integrated Learning Model

Few research studies focus on integrating software-testing concepts to computer science curricula. However, there are several examples of applying conceptual learning in software engineering classes within computer science programs [2]. As a specific area of software quality assurance, software-testing education is a little behind in its use of current technology and cyber-enabled environments. The study presented in this paper reports on how current technology and cyber-enabled environments can be used to support student learning in software testing.

Technology-Based Teaching

The Internet is the most influential technology in the century, and the most popular application throughout human history. Of course, several other technologies, such as networking and protocols (WAN and LAN techniques), memory, and web page design, are the key supporting components for the development and boom of the Internet. Aside from the Internet, mobile applications, apps, and robotics are all the latest emerging technologies that currently occupy classroom space. Today's technology-based teaching is not limited to the Internet but also encompasses all new emerging technologies that affect traditional teaching strategies. The definition of technology-based teaching (TBT) is changing from instilling digital technology to classroom teaching via ad hoc multiple techniques to a new cyber-learning environment. One of the key ideas behind the new era of teaching is to motivate students and improve the quality of educators and instructors.

First and foremost, we use technology-based teaching as the education strategy for content delivery via all electronic technology mediums, including the Internet, intranets, satellite broadcasts, audio and video tape, video and audio conferencing, Internet conferencing, chat rooms, e-bulletin boards, webcasts, computer-based instruction, and CD-ROM [3]. TBT also encompasses related terms, such as online learning and web-based technology that occurs via the Internet, and computer-based learning that is restricted to learning through the use of computers. E-learning is synonymous with TBT and has largely replaced "TBT" in scholarship and industry as the term of choice. In this paper we use these terms interchangeably. However, it is worth noting that there is difference in distance learning, sometimes referred to as *technology-delivered learning,* and TBT. TBT includes methodologies in which instructors and learners are in the same room or instruction is computer-based and there is no 'distance' involved. On the other hand, TBT is more narrowly defined in that it does not include text-based learning or courses conducted via written correspondence that would be covered by either distance learning or technology-delivered learning. Furthermore, technology-enhanced learning describes a methodology in which technology plays a subordinate role and serves to enrich a traditional face-to-face classroom. However, both TBT and digital learning aim at improving students' learning outcomes through increasing their motivation and curiosity.

Motivational challenges in cyber-enabled instruction include those that are characteristic of both distance learning in general and computer-based instruction if the web instruction includes tutorials. This means that students normally have to work for long periods without social reinforcement and have to be able to learn effectively from text and graphics. Many people require multi-modal interaction to learn effectively. That is, they like to hear the content and discuss it in addition to, or instead of, reading it.

Consequently, the motivational challenges of web instruction fall into three broad categories. This first is learning environment design. In web-based instruction, it is desirable to include all of the features that are known from previous research in computer-based instruction and other self-directed learning research. These include basic principles of instructional design, such as clearly describing the goals and content of the instruction and providing concrete examples and application exercises with feedback. Some form of assessment is especially important because it provides the only means for students to know if they are mastering the material.

From a motivational perspective, the learning environment must have features that both capture and sustain student attention. To "re-energize" the students from time to time, it is necessary to provide variation in the sequencing and types of activities and to include unexpected features, such as "pop-up" windows with interesting facts or anecdotes about the content. The relevance and confidence dimensions are also critical because of the isolation of the students and the students' temptations to avoid the lesson in favor of other, more immediate demands of their lives.

Cyber-Enabled Learning Environments

The recent developments in web-based technology have resulted in a bloom of web-based education and media supported peer-learning environments in academia and marketing. These new platforms present educators with an important opportunity to increase students' access to secondary and tertiary education. Meanwhile, a wider use of active learning strategies with broader development potential appears. While stimulating excitement among educational technology professionals, such development also stirs fears of yet another resource-hungry enterprise, draining finances from already under-funded academic institutions. The nascent development of learning object standards and repositories offers a productive response to these fears. Sharing high-quality learning objects across the Internet, developed by a few but used by many, enables cost-effective development and deployment of these expensive resources [4]. But how can educators be assured that the learning objects they find in online repositories are of high enough quality to fulfill their objectives?

A typical form of e-learning applications is exercise submission and assessment systems that allow students to work on their assignments whenever and where they want (i.e., dislocated, asynchronous work). Since the demand for new types of exercises arises based on the teaching environment, it is important that e-learning systems provide a flexible way to add new types of exercises, and that they are scalable to address the demands of different courses and lectures.

In undergraduate and basic computer science courses, programming exercises and assessments are widely used. These courses usually have a large number of participants, leading to several problems when the submitted exercises have to be graded by tutors. The main reason for this is that programming exercises, no matter which programming language is used, tend to have a large degree of freedom for learners. Thus, simply comparing the provided solutions with a sample solution does not produce a reasonable result that can be used for grading, since different, yet still correct, solutions for the same exercise exist. It is possible that the submitted solution still fulfills the required aspects of the exercise, but follows a completely different way to solve the problem than the given sample solution. Only a manual correction by an experienced tutor and a semantic comparison with a sample solution can lead to an acceptable form of correction. The tutor must be aware that there are many ways to solve one specific programming problem. Some programming languages offer a larger degree of freedom than others, but generally this characteristic is typical for a high-level programming language.

Learning Object Development

Resulting from the explosive growth of the Internet and the rise of technology, a new level of resource sharing has appeared and become available in various forms. The revolution of web-development technology synthesizing with education has been displayed in various types of formats, one of the latest of which spawned from the web and information technology and is the so-called *learning object*, the cyber equivalent of earlier shareable resources for education and training. The types of educational formats, such as lecture handouts, textbooks, and presentation slides, can all be considered as learning modules. A large variety of topics can be displayed and presented in learning objects (LOs). In addition, the interactive learning bound to LOs can be realized by programming for assignments, cases, models, virtual laboratory experiments, simulations, and many other electronic resources for education and training. Many thousands of learning objects are now freely available through online repositories that can be searched using metadata that is being standardized by international and national organizations.

A learning object (LO) is defined as *a knowledge element, learning resource, online material, or instructional component,* all of which are terms that have been used to refer to learning objects. NETg, a major eLearning provider, defines a learning object as a resource with three parts–a learning objective, a learning activity, and a learning assessment [5].

The Learning Technology Standards Committee (LTSC) [6] of the Institute of Electrical and Electronic Engineers (IEEE) is a standardized international organization that eventually took a broader perspective on learning objects and provided a uniformed specification. In addition, this standardization categorized the learning object into three groups: aggregation level, interactive type, and resource type. As a standard-setting body with representatives from a wide range of organizations that develop products and specifications for eLearning, LTSC defines a learning object as *any entity, digital, or non-digital, which can be used, reused, or referenced during technology supported learning* [7]. These properties and categories were approved in 2002, which are elements of the IEEE Learning Object Metadata (LOM) standard [8].

The pressing issue for educators and researchers are the design, development and production of large resources of learning objects so that learners and students can find the appropriate learning objects for their particular uses among the profusion of online materials. Since this issue poses a serious challenge within the educational and research domains, many researchers have conducted studies in order to partially solve this problem [13 -16, 20-22]. Perhaps, the critical defining characteristic of the learning object concept is the ongoing development of a set of related metadata standards and specifications that permit learning resources to be searched for and retrieved in convenient and effective ways. Some key works were completed by the standard organizations in order to ease the development and to support the interactive design online.

Drupal [17] is a free and open source PHP-based content management system. Content management systems traditionally allow for multiple end users in an organization to contribute content to a website, without the need to go through a single webmaster. Many web applications and systems can be redesigned using Drupal without any programming involved. Drupal is also an extensible framework that can be used to create web applications and social software as well [18]. Many popular sites on the web were built with Drupal. Using Drupal, the current author tried to create a blended and interactive learning environment that was learner-centered, not content-centered, and student-centered, not instructor-centered [19].

Developed by the Advanced Distributed Learning (ADL) Initiative [9], and in order to meet the eLearning needs of the US Department of Defense, the Shareable Content Reference Model (SCORM) combines a range of technical specifications and standards, including the IEEE LOM Standard. Moreover, conformance tests were developed by the ADL to verify that a learning object complies with specific aspects of SCORM, such as the metadata requirements. One can anticipate that SCORM will drive a substantial portion of eLearning providers to pack their current and future products with standard metadata. Notably absent from the IEEE LOM standard are metadata on the quality of learning objects as judged by users or third-party assessors—the type of metadata with which this article is primarily concerned. We believe that the development of tools and formats for quality evaluation of learning objects will be the next major advance in LOM and one that will have a powerful impact on the design of interactive cyber-enabled learning environments to facilitate teaching and education in various types of media formats in web-based online education and training.

Current Learning Objects Developed at Alabama A&M University

In order to populate WReSTT-CyLE with learning objects on software testing techniques a tree diagram was created to guide the development [10, 11]. In this tree diagram, the learning objects developed at AAMU are mainly in the category of white-box testing and black-box testing. White box testing includes graphic control structure, path testing, statement testing, etc. Black box testing includes boundary value testing and equivalence testing [1]. Table 1 includes a summary of current learning objects (LOs) in regard to the above testing categories that are either in the WReSTT-CyLE environment or being reviewed for WReSTT-CyLE.

Table 1. Learning Objects Developed at AAMU

Learning Objects	Description	Practice Question	Test Questions	Answer Key (Available)
LO#1. Control Flow Graph (CFC)	Give an introduction to basic concepts and step-by-step building control flow graph.	10	10	Yes
LO#2. DD Path Testing	Give an introduction to the definition and step-by-step development of DD-path.	10	8	Yes
LO#3. Basis Path Testing	Give an introduction to the definitions and step-by-step development of basis path testing.	10	8	Yes
LO#4. Boundary Value Testing	Give an introduction to the relevant definitions and step-by-step development of boundary value testing.	10	10	Yes
LO#5. All def Testing	Give an introduction to All Def testing.	7	5	Yes
LO#6. All uses Testing	Give an introduction to the definition of def and uses and how to perform all uses testing.	7	5	Yes
LO#7. All Def-Uses path testing	Give an introduction to the definition of def and uses and how to perform all uses testing.	7	5	No
LO#8. Statement Coverage	Give an introduction to the definition of coverage.	5	5	No
Others	Mobile app testing techniques and tools (Android and iOS).	10	10	No

Other than the developed learning objects (LOs) and those being developed, shown in Table 1, we plan to develop more learning objects for mobile app testing and robotics system testing. The testing of these types of platforms greatly differs from the existing forms.

Cyber-Enabled Learning Environments and WReSTT-CyLE

Development of the first WReSTT (*Web-based Repository of Software Testing Tools*) version was completed at Florida International University (FIU) in 2009 and was supported by the NSF program [12]. WReSTT, the initial repository, was an online software-testing repository that contained learning materials about software-testing tools only. Later, based on existing results, class studies and research on the learning objects, an integrated framework of a software-testing repository using learning objects was created in 2012. This project was a collaborative effort between FIU, Florida A&M University (FAMU), Miami University (MU) and North Dakota State University (NDSU) [13]. The NSF Course, Curriculum, and Laboratory Improvement (CCLI) Phase I project (first version of WReSTT) had the following objectives: (a) create learning materials on testing tools, (b) increase the number of students who have access to testing tool tutorials, and (c) train instructors on how to use testing tools and WReSTT in the classroom.

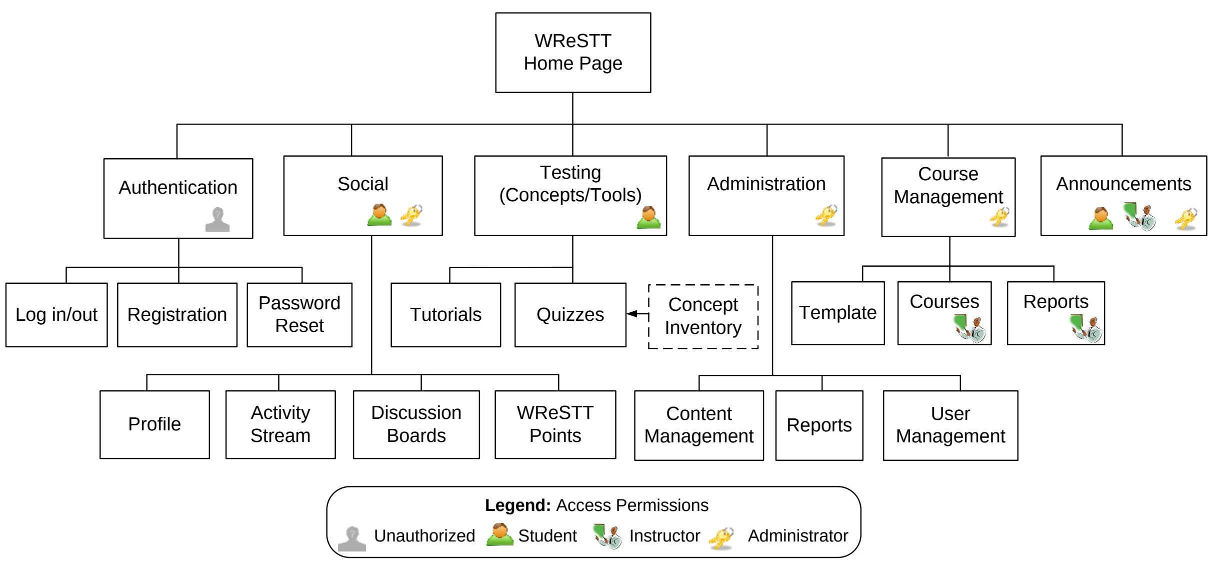

Figure 2. Block Diagram of WReSTT-CyLE

WReSTT-CyLE is the focus of an NSF Transforming Undergraduate Education in Science (TUES) II project that aims at providing a cyberlearning environment that facilitates the improvement of students' conceptual understanding and practical skills in software testing. The *main goals* of this project are to *create new learning materials* and *develop faculty expertise* to significantly increase the number of undergraduate students that are exposed to testing methodologies and tools in undergraduate courses with a programming component. The WReSTT-CyLE (*Web-based Repository of Software Testing Tutorials – a Cyberlearning Environment*) project is ported to Drupal 7 using PhP 5 or a higher version to create the repository [17]. The key features of the WReSTT-CyLE tool include the following:

Social Features: These allow students to set up profiles; access the activity streams of students in their classes; pose questions to the members of their virtual teams, other students in their classes or to the WReSTT community; and view the virtual point leaders of students in their classes. In addition, the students will be able to link other social networking accounts to newly created WReSTT-CyLE user groups and feeds from other testing user groups.

Learning Objects: These are important enhancements to the contents of WReSTT-CyLE for software-testing education and community. The LOs are made in response to the feedback of academic community users. These enhancements include the following: (1) presenting the material in the learning objects using varied formats (e.g., video, audio and text); (2) the addition of new learning objects on testing techniques for black-box, white-box and GUI testing; and (3) the addition of new learning objects for the testing tools based on varied IDE platforms (e.g., Eclipse and NetBeans).

The gamification design and mechanism is integrated in the social features and is developed with several elements that are connected to the project objectives. We will discuss these elements, the mechanism, the psychological characters and system implementation in the next section. An overview design and analysis is shown in Figure 2.

Current Situation of Software Testing in the AAMU Computer Science Curriculum

Table 2 includes a summary of the current situation of software-testing instruction within AAMU's computer science curriculum.

Table 2. All Computer Science Class Studies at AAMU

Course Num	Course Title	Semesters of Class Study Performed
CS/CMP 206	Visual Programming I (in Java)	Spring, Fall 2015
CS/CMP 328	Object oriented Design in UML	Spring 2015
CS/CMP 401	Software Engineering	Spring 2014, 2015
CS/CMP 403	Senior Design	Fall 2014, 2015
CS/CMP 521	Object oriented Programming and Design	Fall 2014, 2015
CS/CMP 561	Software Engineering Methodology	Spring 2014, 2015
CS/CMP 582	Wireless and Mobile Computing	Summer, Fall 2015

Class Studies within AAMU's Computer Science Curriculum

We have completed several class studies in the past three years, including CS/CMP 206, CS/CMP 328, CS/CMP 401 and CS/CMP 403, CS/CMP 521 and CS/CMP 561. A summary list of each course is given in Table 2, as previously mentioned. Table 3 summarizes the process that was adopted by each course during the class study at AAMU, followed by a short description of CS/CMP 401, 403 and 561.

Table 3. Process Units Adopted by All the Classes

Course Num	Pre-Test (Y/N)	Post-Test (Y/N)	# of LOs	Rewards Pts (Y/N)	Final Grade (Y/N)
CS/CMP 206	Y	Y	7	Y	Y
CS/CMP 328	Y	Y	7	Y	Y
CS/CMP 401	Y	Y	3	N	Y
CS/CMP 403	Y	Y	3	N	Y
CS/CMP 521	N	N	7	Y	Y
CS/CMP 561	Y	Y	7	Y	Y
Total	5	5	7	4	7

Pre-test provides a starting bar of student assessment, which can be enabled before releasing the LOs to students. To be valid, pre-test is used for most of the courses during study. In comparison with pre-test, post-test shows an improvement of understanding of software-testing concepts and test skills. This result varies from class to class. Currently, 200-level and graduate-level classes demonstrated a more efficient learning outcome than students in 300-level or higher classes. By observing this, we have performed another survey regarding the knowledge units. Through this survey, we found that some of the graduate students were exposed to the content of software testing in the undergraduate curriculum in relevant courses.

The last column of Table 3 indicates whether or not the points rewarded for the LOs in WReSTT-CyLE were included in the students' final grade. During the data collection of Spring 2014, all studied classes included the relevant questions. The main reason is based on our faculty's previous class study since there was a strong link between the final exam questions and the LOs in WReSTT-CyLE. If the questions on software testing were not included in the final exam, even students with high motivation and excellent grades in the class would have a hard time maintaining a high performance.

Our current computer science curriculum is designed to organize the senior design project into two semesters: Software Engineering (CS/CMP 401) and Senior Problem (CS/CMP 403). In this way, computer science students have enough time to complete a sophisticated project that is able to demonstrate integrated software development skills through the development cycle. In responding to this curriculum design, graduate-level courses include CS/CMP 521 and CS/CMP 561 for design & programming and software engineering. Short descriptions of the relevant courses are illustrated in the following subsections.

CS/CMP 401 – Software Engineering

This course covers the ideas involved in large-scale programming design. The software life cycle is covered along with design specifications, verification and validation and the use of various supporting CASE tools. The student is expected to design and document a software system of some kind and may be asked to code some of the design. Software-testing techniques and strategies for performing software testing on the system will be introduced. WReSTT-CyLE provides many meaningful LOs that introduce the testing concepts to the students.

In this course, students are placed within groups and given an application that they must create using the Software Development Life Cycle (SDLC) over the course of the year. During the creation of the application, students are required to test the application against the requirements to ensure that the application was designed and implemented correctly. In order to prepare them for this task, the students were provided access to several introductory learning objects (LOs). Each LO was designed to familiarize the students with topics in software testing that would assist them in understanding the various software testing techniques. This preparation should increase the students' chances of success in both creating testable requirements for their applications as well as creating adequate test cases.

CS/CMP 403 – Senior Problem

This is capstone course of Computer Science at AAMU and a continuation of CS/CMP 401 (software engineering). During this course, the student is expected to code a single, meaningful project started earlier in CS/CMP 401 and present the results of this project in class. This project must meet a set of standards for software design and documentation. Topics of professional ethics and responsibilities are discussed in the class. Software-testing techniques must be used in this course on the selected project to ensure the quality of the system. WReSTT-CyLE can be used in this course with seamless connection to the project content.

CS/CMP 561 – Software Engineering Methodology

This is a graduate-level course that explores the traditional approach to software construction, software crisis and software characteristics. The course covers various software engineering paradigms, and the fundamental concepts of analysis, design, coding, testing and maintenance. It also introduces various CASE tools. Software-testing techniques must be used in this course on the selected project to ensure the quality of the system. Several testing tools were introduced in this project, such as xUnit, Robotium and Selendroid, etc. As with CS/CMP 403, WReSTT-CyLE can be used in this course with seamless connection to the project content.

Data Collection

This report discusses the initial study of a software-testing repository using a WReSTT-CyLE tool. Depending on the courses and semesters, the class studies covered a different number of LOs. In Spring 2014, all courses only covered three learning objects (Table 1). In the Fall 2015, the classes ranged to seven learning objects, which increased the reliability of the results. Even though the number of learning objects was slightly different between semesters, the data collection was automatically done through WReSTT-CyLE. The following discussion introduces the data collection of the class study of Spring 2014 for the three learning objects.

The class study consisted of students completing a series of learning objects designed to introduce them to the principles of software testing. The LOs were comprised of three separate series covering introductory topics in software testing. Each student was ranked based on his or her individual scores and also based on the average score of the student's group. The team scores were then ordered from highest to lowest and compared to the number of testable requirements that the group was able to create. The team scores were also compared to each group's scores on their graded test cases to determine if a correlation could be found. During the project implementation of the past three years, five workshops involving a team were developed to disseminate the tool and the research study on using LOs in the computer science curriculum. Several study results have been published over the past few years [13, 14, 16].

Analysis

There are two ways to calculate the score from the WReSTT-CyLE tool kit. One considers the test score only. By doing this, the bias created by reward points is reduced. However, the disadvantage is that the motivation of students using other engagement strategies, e.g., gamification, cannot be identified and studied. Another calculation method includes the reward points. The benefit is an increased point of evaluation of student motivation and assessment regarding the use of gamification research on teaching. The drawbacks are obvious. First, the grade of test scores is skewed, as students with a top score without the reward points will be immediately taken out of the list due to the low total grade. Second, many students are blindly completing the test questions; therefore, their group can receive the highest score, but have the lowest level of knowledge of software-testing concepts and techniques. This can be reflected and fixed by questioning the final grade. The above facts are observed in the following courses: CS/CMP 206, CS/CMP 328, and CS/CMP 521, CS/CMP 561 and CS/CMP 582.

Although the results proved to be inconclusive in determining a relationship between the student's performance on the requirements and testing assignments, we did seem to find that there were situations in which a team's average LO score showed a slight improvement in the assignment score in the study of CS/CMP 401 and CS/CMP 403 classes. More specifically, LO3 appeared to have the greatest affect on the group assignment. The average group score on LO3 was a 74; this produced an average of 91 on the team requirement assignments and an average of 82 on the team test cases assignment. The LO with the least noticeable affect was LO2, which produced significantly lower averages on both the requirements and test case assignments while the average group score on LO2 was a 61.

Discussion of Results

From the class study of using learning objects in WReSTT-CyLE, we observe a consistent performance between a student's final grade in the class and the grade of learning objects in the test score. If the student is able to maintain the top level in this class, this student is able to main an ideal level of test grade in the learning objects of software-testing concepts and techniques and vice versa.

Thus far, we have observed both valid and false positive results of using rewards points in the learning process. To evaluate the false positive results of using rewards points, additional study is needed to reveal the truth and false factors in order to derive meaningful results [13,14,15,16].

Conclusions and Future Works

A study of software-testing learning objects in the computer science curriculum from sophomore classes to graduate courses is presented in this paper. This study of software-testing learning objects and integrating software-testing concepts into the computer science curriculum was conducted on and facilitated by a cyber-enabled learning environment WReSTT-CyLE, which has been supported by the NSF program. In addition, a small proportion of software-testing learning objects were developed by the authors of this paper. The results of the class studies have been presented, data were collected and analyzed and some research and pedagogical issues were identified. Future works mainly fall in two aspects. One is from the view of the development of more learning objects in support of software-testing education in computer science and IT curricula. The current WReSTT-CyLE platform needs to be extended to provide additional functionality including additional social features. The other is from the view of class studies, as more courses are added, especially CSI-level courses, the LOs in software testing need to be tailored to this level of student.

Acknowledgements

The authors would like to acknowledge the support of the NSF program under award No. DUE-1225654, DUE-1225742. The authors would like to thank the reviewers for their thoughtful and insightful reviews.

BIBLIOGRAPHY

[1] Ammann, P. and Offutt, J., Introduction to Software Testing, Cambridge University Press, 2008.

[2] Kahsai, T. and Liam, O'R., Teaching software engineering concepts through industrially used tools early in the undergraduate curriculum. In Proceedings of 122nd ASEE Annual Conference & Exposition. Seattle, WA June 14 – 17, 2015.

[3] ASTD's E-Learning Glossary. E-Learning Glossary. Retrieved on Jan 30th, 2016, from http://www.lupi.ch/Schools/astd/astd2.htm.

[4] Munkittrick, P., Building a foundation for connected learning, Technological Horizons In Education Journal, 27 (9), April 2000, 54–56.

[5] ADLNET, SCORM Users Guide for Instructional Designers. Version 8. 2011.

[6] Learning Technology Standards Committee (2002), *Draft Standard for Learning Object Metadata. IEEE Standard 1484.12.1* (PDF), New York: Institute of Electrical and Electronics Engineers, *retrieved 2008-04-29.*

[7] IEEE P1484.12 Learning Object Metadata Working Group. Scope and purpose. Retrieved July 2002. Available at http://ltsc.ieee.org/wg12/s_p.html.

[8] Draft Standard for Learning Object Metadata. IEEE P1484.12.1. Retrieved August 2002. Available at http://ltsc.ieee.org/doc/wg12/LOM_WD6_4.pdf.

[9] ADLNet. Retrieved August 2002. Available at http://www.adlnet.org.

[10] P. J. Clarke, D. Davis, T. M. King, J. Pava and E. L. Jones, " ACM Trans. Comput. Educ.," *Integrating Testing into Software Engineering Courses Supported by a Collaborative Learning Environment,* vol. 14, no. 3, p. 33, Oct 2014.

[11] P. J. Clarke, J. Pava, D. Davis and T. M. King, "Using WReSTT in SE Courses: An Empirical Study.," in *the 43rd ACM Technical Symposium on Computer Science Education (SIGCSE '12).*, Raleigh, NC, 2012.

[12] Transforming Undergraduate Education in STEM: Making and Measuring Impacts. 2011 CCLI/TUES Principal Investigators (PIs) Conference January 26-28, 2011 Washington, DC.

[13] P. J. Clarke, J. Pava, Y. Wu and T. M. King, "Collaborative Web-Based Learning of Testing Tools in SE Courses," in *the 42nd ACM Technical Symposium on Computer Science Education (SIGCSE '11).*, New York, NY, USA, 2011.

[14] P. J. Clarke, D. L. Davis, R. Chang-lau and T. King, "Observations on Student Use of Tools in an Undergraduate Testing Class," in *the 121st American Society for Engineering Education (ASEE) - Software Engineering Constituent*, 2014.

[15] Fu, Y. and Clarke, P., "Gamification based Cyber Enabled Learning Environment for Software Testing" submitted to *the 123rd American Society for Engineering Education (ASEE) - Software Engineering Constituent*, 2016.

[16] A. Goswami, G. S. Walia and S. Aburardeh, "Using a Web-Based Testing Tool Repository in Programming Course: An Empirical Study," in *the 2014 International Conference on Frontiers in Education: Computer Science and Computer Enginee,* 2014.

[17] Drupal Community. Drupal, 2016. http://drupal.org/.

[18] Mening, R., "Building a Simple Drupal Website", Websitesetup.org. November 2013. Available from http://websitesetup.org/build-website-drupal/.

[19] Rovai, A.P.. Blended learning and sense of community: A comparative analysis with traditional and fully online graduate courses. International Review of Research in Open and Distance Learning, 5(2), 2004

[20] Ghadeer W. Alharthi, Abdellatif E. Elgazzar, and Ahmed M. Nouby. Designing Learning Objects for a Computer Science Course: Are There Any Effects on Developing Practical Skills and Professional Interests among Students of Computer Department, Faculty of Education, University of Dammam. Open Journal of Social Sciences Vol.2 No.2, Pub. Date: January 9, 2014

[21] L. C. Begosso, L. R. Begosso, A. Ribeiro, R. M. dos Santos and R. H. Begosso, "The use of learning objects for teaching computer programming," Frontiers in Education Conference (FIE), 2015. 32614 2015. IEEE, El Paso, TX, 2015, pp. 1-6.

[22] A. Alharbi, F. Henskens, and M. Hannaford. Student-centered learning objects to support the self-regulated learning of computer science. Creat Educ 3:773–783, 2012.