WReSTT – Web-Based Repository of Software Testing Tools

Peter J. Clarke
School of Computing and Information Sciences
Florida International University
Miami, FL 33199, USA
clarkep@cis.fiu.edu

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

Edward L. Jones
Dept. of Computer and Information Sciences
Florida A&M University
Tallahassee, FL 32307, USA
ejones@cis.famu.edu

Abstract

The size and complexity of software systems continue to grow as software becomes more pervasive and ubiquitous. Ensuring the quality of such systems in the 21st Century will require changes to the development strategies and improvement to the pedagogy used to teach these strategies in academia. After more than five decades, software testing continues to be the primary approach used to ensure the development of high quality software. During this time, new testing tools and techniques have been developed to support software testing in industry. However, practitioners have indicated that more work is needed to train developers in the use of these testing tools and techniques, as on the job training will no longer suffice.

Any comprehensive approach to improving the quality of software systems developed in the future requires that academic institutions play a vital role in training students how to test software, and making them aware of the tools and techniques available to support software testing. Unfortunately, students and instructors continue to be frustrated by the lack of support provided when faced with the task of selecting appropriate testing tools and program analyzers to verify programs under development. In this paper we describe the initial version of a **W**eb-based **Re**pository of **S**oftware **T**esting **T**ools (WReSTT), and outline planned enhancements. WReSTT provides students and instructors with: (1) a vetted selection of software testing tools in a single location that can be used for pedagogical purposes, (2) easy access to tutorials on software testing and tutorials for testing tools, and (3) access to a moderated forum (discussion board) where they can share feedback on the usefulness of software testing tools and tutorials.

1. Introduction

The size and complexity of software systems continue to grow as software becomes more pervasive and ubiquitous. Ensuring the quality of such systems in the 21st Century will require changes to the development strategies and improvement to the pedagogy used to teach these strategies in academia. After more than five decades, software testing continues to be the primary approach used to ensure the development of high quality software. Recent studies [CNSS (2005), RTI (2002)] have indicated that there still needs to be major improvements in the way software is tested. The National Software Summit report states that there is a gap between the current state-of-the-art technologies and what is required in software tools to routinely develop error-free software. The NIST planning report [RTI (2002)] states that the potential cost reduction from feasible software testing infrastructure improvements (including new testing tools) is estimated at $22.2 billion annually. This urgent need to develop high quality software products has resulted in a plethora of software testing tools and program analyzers being developed in recent times [ATT(2007), Wikipedia(2010a)]. However, the development of testing tools alone cannot solve the problem of developing high quality software products.

Improving the quality of software systems developed in the future requires that academic institutions play a vital role in training students how to test software, and making them aware of the tools available to support software testing. This is particularly true in the Computer Science (CS) and Information Technology (IT) degree programs. Unfortunately, students and instructors continue to be frustrated by the lack of support provided when faced with the task of selecting the appropriate testing tools and program analyzers to verify programs under development. One approach to alleviate this problem is to provide students and instructors with guided access to a set of vetted software testing tools and program analyzers that will support the development of high quality software during their academic training. Although there are several repositories available that contain software testing tools, none target the pedagogical needs of students and instructors.

In this paper we describe the initial version of a **W**eb-based **Re**pository of **S**oftware **T**esting **T**ools (WReSTT) (http://wrestt.cis.fiu.edu/) and describe how WReSTT is currently being used. The full-scale implementation of WReSTT will provide several benefits to students and instructors in CS1-CS3 and software engineering courses. These benefits include: (1) the availability of a vetted selection of software testing tools in a single location that can be used for pedagogical purposes, (2) easy access to tutorials on software testing and tutorials for testing tools, and (3) access to a moderated forum (discussion board) where students and instructors can share feedback on the usefulness of the tools and tutorials in the repository.

WReSTT is part of a National Science Foundation (NSF) sponsored CCLI Phase I project, and is a collaboration between Florida International University (FIU) and Florida A&M University (FAMU). During this project we identified several enhancements that users requested and will describe these enhancements in this paper. The outline of the rest of the paper is as follows: Section 2 describes other web-based repositories to support testing in the classroom. Section 3 describes the structure of WReSTT and its contents. Section 4 discusses the enhancements that will be made to WReSTT in the future, and in Section 5 we conclude the paper.

2. Related Work

There are several web-based repositories that provide access to learning materials on software testing. These repositories can be classified as: (1) repositories sponsored by academicians, (2) repositories sponsored by companies, and (3) major academic initiatives sponsored by companies. In this section we will briefly describe some of the repositories and academic initiatives related to the work we are doing in WReSTT.

The repositories maintained by academicians are usually those initially sponsored by grants from federal agencies or companies. Some of these repositories are:

Center for Software Testing Education and Research (CSTER) [Kaner(2010)]. The mission of CSTER is to “Create effective, grounded, timely materials to support the teaching and self-study of software testing, software reliability, and quality-related software metrics.” The major support for CSTER was from the NSF under grant EIA-0113539 ITR/SY+PE "Improving the Education of Software Testers." CSTER has a large repository of content on software testing including video lectures, practice quizzes, drills, and other assessment materials. The content of CSTER is free and there is no registration to access the site. Since November 30th 2004 there have been 144,148 visitors to the CSTER site.

OpenSeminar: Software Engineering [Williams(2010a)]. 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 module 7 of the software engineering component [Williams(2010b)]. The major support for the OpenSeminar Software Engineering component includes the IBM Corporation, NSF, and North Carolina State University. Unlike CSTER, software testing is only one module in the Software Engineering component and does not provide much content on software testing. To gain access to the content to the testing module of OpenSeminar: Software Engineering, you may enter as a guest or register as an instructor or student.

The Network Community for Software Engineering Education (SWENET) [Lutz et al.(2010)]. SWENET is a project to produce and organize high-quality materials supporting software engineering education. SWENET is sponsored in part under the NSF grant EEC-0080502. The project builds on the ABET program criteria for software engineering education, and the IEEE Computer Science Software Engineering Body of Knowledge (SWEBOK). One of the Software Engineering Education Knowledge (SEEK) modules is Software Verification and Validation (VAV) [Lutz et al.(2010b)]. There are seven entries in the VAV category. It is worth noting that when all the SEEK categories are listed there are some testing topics not located in the VAV category that are contained in SWENET. There are 184 registered users on SWENET.

Bug Hunt [Elbaum and Stolee (2010)]. Bug Hunt is an online tutorial to instill good testing practices through lessons and challenges. Bug Hunt is supported in part by the Career Award 0347518 to the University of Nebraska-Lincoln, the Great Plains Software Technology Initiative, and the UCARE Project at the University of Nebraska-Lincoln. 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. Bug Hunt provides access to a demo that allows users to get a feel for the tutorial. To use the additional features of Bug Hunt, users need to create an account. Since October 29th 2007 there have been 6949 visits to the Bug Hunt website.

Multimedia Educational Resource for Learning and Online Teaching (MERLOT) [MERLOT (2010)]. “MERLOT is a leading edge, user-centered, searchable collection of peer reviewed and selected higher education, online learning materials, catalogued by registered members and a set of faculty development support services.” Merlot is supported by a consortium of higher education institutions, corporate partners and community partners. MERLOT is not exclusive to software testing but supports a wide cross-section of learning materials divided into seven collections, which are further divided into many disciplines. These collections include Arts, Business, Education, among others. A search for the term “Software Testing” returns 22 entries, but the majority of the search results seemed to not be related to testing. MERLOT contains over 21752 materials and has 79294 members.

Several companies provide access to software engineering educational materials through academic initiatives. The companies include IBM [IBM(2010)], Microsoft [Microsoft(2010)] and Sun Microsystems [Sun(2010)]. These academic initiatives all provide access to course materials and tools to support the programming courses at academic institutions.

3. WReSTT

WReSTT (http://wrestt.cis.fiu.edu/) was developed using a four-tier architectural model, and implemented using Drupal [Drupal2008] (See Figure 1). WReSTT currently contains learning materials and multimedia tutorials for several testing tools (TT), metrics tools (MT) and code coverage tools (CCT) including: Cobertura [Doliner(2009)] (CCT), CppUnit [CppWiki(2009)] (TT), EclEmma [Hoffmann(2009)] (CCT), JDepend [Clarkware(2010)] (MT), JUnit [Object Mentor(2009)] (TT), SWAT [SWAT(2009)] (TT) and Rational Functional Tester [IBM(2009)] (TT).

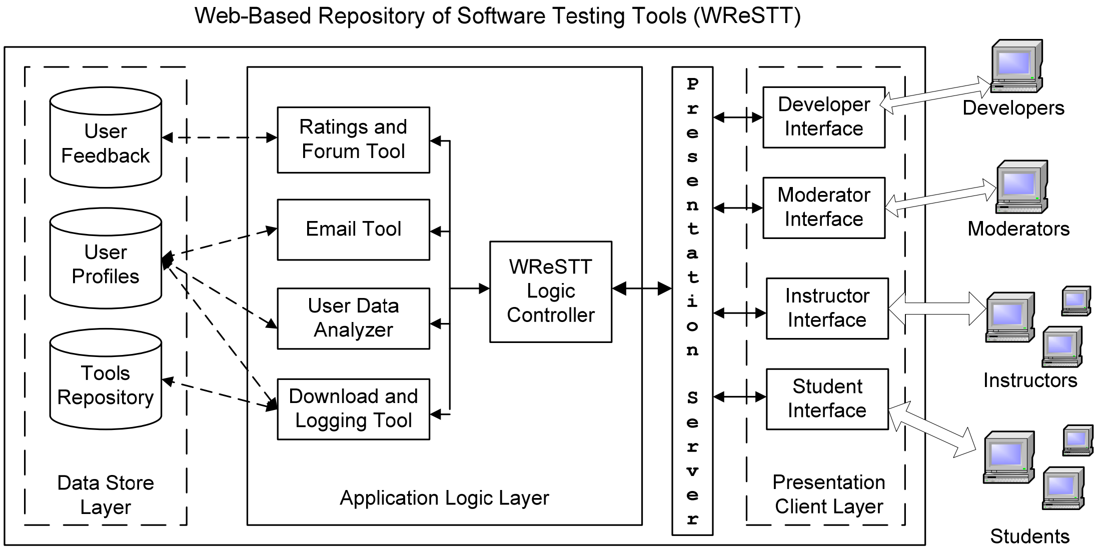

Figure 1: Four-tier architecture for the Web-Based Repository of Testing Tools (WReSTT)

WReSTT is designed to allow access to four user roles: developers, moderators, instructors, and students. Users of each category have access to different subsets of facilities in WReSTT: *developers* are provided administrative access to the systems; *moderators* respond to queries from users and moderate the forums that users can access; and *instructors* monitor the usage of WReSTT by the students in their classes (e.g., access scores obtained by students on a particular tutorial); and *students* have access to all the learning materials. Currently, there are over 520 users from 16 different academic institutions registered in WReSTT. WReSTT is currently accessible from MERLOT. Access to WReSTT from NDSL [NDSL (2009)] and Connexions [Connexions (2009)] is currently pending. WReSTT was denied a request to host it on OpenDOAR [SHERPA(2009)] due to the fact that WReSTT requires registration to access some of the material.

Students and instructors can browse the testing tools and tutorials in WReSTT in three ways: (1) *by category* – according to popular classifications such as code coverage, metrics, plug-ins to IDEs, test execution, and web; (2) *by language* – according to the programming language used to implement the system under test, e.g., C++, Java, and (3) *by test level* – according to the stage of the development process, e.g., unit, integration, system. WReSTT also provides an advanced search mechanism which allows users to filter testing tools using descriptive tags that are related to the aforementioned browsing methods. For example, a user can search for unit testing tools that can validate C++ programs, and are also plug-ins to IDEs. Students and instructors can use this *advanced filter* to quickly locate the tutorials that meet their needs, or recognize that such tools are not available and suggest that they be included in the repository.

4. Enhancements to WReSTT

We plan to enhance WReSTT in the coming months to include additional instructional materials (e.g., multimedia tool tutorials, syllabi, and labs). Feature enhancements to WReSTT will include incorporating a *social networking component for students and instructors*, and adding *new features for instructors*.

The student social networking component will support social learning [Lave & Wenger(1998)] by providing students with the opportunity to learn and think in groups. Students will be able to set up virtual groups and work on tutorials together, pose questions to each other in the group, and seek advice from instructors as a group. Similarly, the instructor social networking component will also allow instructors to create virtual groups, but for the purpose of exchanging ideas and experiences on how to use the resources in WReSTT to effectively support software testing instruction. Both student and instructor members of WReSTT, who are also members of the social networking utilities Facebook [Facebook(2010)] and Twitter [Twitter(2010)], will have the option of linking these social networking accounts to newly created WReSTT user groups and news feeds. The aforementioned groups and feeds will be available and accessible through the WReSTT portal and will also serve as a basis for expanding the user base of WReSTT through links back to the WReSTT website.

Additional instructor features will provide the ability to upload class rolls using a CSV file, and allow queries to the repository based on attributes related to class rolls. An example of one application of such features is to allow students to work through a tutorial, and then provide the instructor with access to feedback on each student’s performance. Such an application can be extremely beneficial when learning to use new testing tools. Finally, to enhance the overall student user experience, statistics on the most frequently accessed tutorials will be tracked using the existing monitoring features of WReSTT, and the content uploaded to the popular video sharing website YouTube [YouTube(2010)]. Mirroring this content to YouTube should facilitate faster download times for the multimedia tutorials that are most demanded by students. The YouTube versions of the video tutorials will be directly accessible and viewable through links embedded in the WReSTT portal.

5. Conclusions and Future Work

In this paper we introduced a **W**eb-based **Re**pository of **S**oftware **T**esting **T**ools (WReSTT) (http://wrestt.cis.fiu.edu/) and described the enhancements that will be made to WReSTT in the near future. WReSTT has been used in several classes at Florida International University (FIU) and Florida A & M University. Students have made several suggestions on how to improve the contents of WReSTT, including the addition of tutorials on tools to test programs written in different languages e.g., PHP, and different platforms, e.g., Netbeans. In the near future we plan to evaluate the effectiveness of WReSTT in several classes including CS2, Software Engineering I, and Software Testing at FIU and North Dakota State University.

References

[ATT(2010)] Applied Testing and Technology, Inc (2010). Software QA testing and test tool resources. http://www.aptest.com/resources.html.

[Clarkware(2010)] Clarkware Consulting, Inc. (2010). JDepend. http://clarkware.com/software/JDepend.html

[CNSS(2005)] CNSS. Software 2015: A National Software Strategy to Ensure U.S. Security and Competitiveness. Technical report, Center for National Software Studies (CNSS), April 2005.

[CppWiki(2009)] CppUnit Wiki. (2009) http://sourceforge.net/apps/mediawiki/cppunit/.

[Doliner(2009)] Doliner, M. (2009). Cobertura. http://cobertura.sourceforge.net/index.htm

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

[Facebook(2010)] Facebook (2010). http://www.facebook.com/

[Hoffmann(2009)] Hoffmann, M. R. (2009). EclEmma. http://www.eclemma.org/

[IBM(2009)] IBM (2009). Rational Functional Tester. http://www-01.ibm.com/software/awdtools/tester/functional/index.html

[IBM(2010)] IBM (2010). IBM Academic Initiative. http://www.ibm.com/developerworks/university/academicinitiative/

[Kaner(2010)] Kaner, C. (2010). Center for Software Testing Education and Research. http://www.testingeducation.org/.

[NDSL (2009)] The National Science Digital Library. (2009) http://nsdl.org/

[Object Mentor(2009)] Object Mentor. (2009). Junit.org. http://www.junit.org/

[RTI (2002)] RTI. The Economic Impacts of Inadequate Infrastructure for Software Testing. Technical Report 7007.011, National Institute of Standards and Technology NIST, May 2002.

[Lave & Wenger(1998)] Lave, J., & Wenger, E. (1998). Communities of Practice: Learning, Meaning, and Identity: Cambridge University Press.

[Lutz et al.(2010a)] Lutz, M. J., McCracken, W. M., Mengel, S., Sebern, M., Hislop, G. W., and Hilburn, T. B. (2010). SWENET - The Network Community for Software Engineering Education. http://www.swenet.org/

[Lutz et al.(2010b)] Lutz, M. J., McCracken, W. M., Mengel, S., Sebern, M., Hislop, G. W., and Hilburn, T. B. (2010). SWENET - SEEK Category: Software Verification and Validation (VAV). http://www.swenet.org/browseModules.aspx?categoryID=11

[MERLOT(2010)] MERLOT (2010). http://www.merlot.org/merlot/index.htm

[Microsoft(2010)] Microsoft (2010). The MSDN Academic Alliance. http://msdn.microsoft.com/en-us/academic/default.aspx

[Sun(2010)] Sun Microsystems (2010). Sun Academic Initiative. http://www.sun.com/solutions/landing/industry/education/sai/.

[SWAT(2009)] SWAT Development Team (2009). Simple Web Automation Tool Kit. http://sourceforge.net/projects/ulti-swat/

[Twitter(2010)] Twitter (2010). http://twitter.com/

[Wikipedia(2010a)] Wikipedia (2010a). List of unit testing frameworks. http://en.wikipedia.org/wiki/List_of_unit_testing_frameworks

[Williams(2010a)] William, L. (2010). OpenSeminar: Software Engineering. http://openseminar.org/se/

[Williams2010b] William, L. (2010). OpenSeminar: Software Engineering: Testing http://openseminar.org/se/modules/7/index/screen.do

[YouTube(2010)] YouTube LLC (2010). http://www.youtube.com/