Reduce, Reuse, Recycle, Recover: Techniques for Improved Regression Testing

Mary Jean Harrold
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280
harrold@cc.gatech.edu

Abstract

One of the most expensive activities that occurs as software is developed and maintained is the testing (or retesting) of the software after it has been modified. Studies suggest that a significant portion of development and maintenance costs go to this retesting, which is known as *regression testing*. Reports estimate that regression testing consumes as much as 80% of the overall testing budget and can consume up to 50% of the cost of software maintenance.

Rapidly changing software and computing environments present many challenges for effective and efficient regression testing in practice. Regression testing can be performed after changes are made to the software, such as after nightly or regular builds, before a new version of the software is released, every time the software is saved and compiled, such as in an agile development environment, or before patches, such as security patches, are released. Regardless of the environment or when it is performed, the goals of regression testing are the same: to improve confidence that the changes behave as intended and that they have not adversely affected unchanged parts of the software.

Because regression testing is important, but expensive, much research has been performed, both in industry and in academia, to develop techniques to make regression testing more effective and efficient. This research has also produced many tools and systems that have been used for empirical studies that investigate the effectiveness, scalability, and practicality of the techniques.

Researchers have developed techniques for addressing a number of issues related to regression testing, and, in this talk, I will discuss them in four areas. First, techniques attempt to *reduce* the regression testing time by creating effective regression test suites that test the changed part of the software, by identifying test cases in the regression test suite that do not need to be rerun on the changed software, and by identifying and removing obsolete test cases. Second, techniques can *reuse* test suites created for one version of the software by identifying those test cases that need to be rerun for testing subsequent versions of the software and by computing an effective ordering for running the test cases. Third, techniques can *recycle* test cases by monitoring executions to gather test inputs that can be used for retesting and by creating unit test cases from system test cases. Finally, techniques could *recover* test cases by identifying, manipulating, and transforming obsolete test cases, by generating new test cases from old ones, and by repairing test cases when the software changes.

In this talk, I will overview the research in testing of evolving software, and discuss achievements to date in managing regression testing by reducing, reusing, recycling, and recovering test cases. I will also present the state of the research and the state of the practice in regression testing. Finally, I will discuss the current trends in both academia and industry, the challenges for solving the difficult problems that exist, the promise for testing of evolving software in the future, and the important open challenges for regression testing in the next decade.

Biography

Mary Jean Harrold is the ADVANCE Professor of Computing and Associate Dean for Faculty Affairs in the College of Computing at Georgia Institute of Technology. She performs research in analysis and testing of large, evolving software, fault localization and failure identification using statistical analysis, machine learning, and visualization, monitoring deployed software to improve quality, and software self-awareness through real-time assessment and response. Her research has been consistently supported by government agencies, such as NSF and NASA, and industries, such as Boeing Aerospace, Tata Consultancy Services, Microsoft, and IBM. Professor Harrold received an NSF NYI Award and was named an ACM Fellow. She has served on the editorial boards of ACM TOSEM, ACM TOPLAS, IEEE TSE, and JSTVR, and on the organizing and program committees for the top conferences in software engineering. She currently serves on the Board of Directors for the Computing Research Association (CRA), and she is on the CRA Committee on the Status of Women in Computing (CRA-W). She received the Ph.D. from the University of Pittsburgh.

978-1-4244-4828-9/09/$25.00 2009 IEEE Proc. ICSM 2009, Edmonton, Canada