2011 Fourth International Conference on Software Testing, Verification and Validation Workshops

Empirically Evaluating Regression Testing Techniques: Challenges, Solutions, and a Potential Way Forward

Gregory M. Kapfhammer
Department of Computer Science
Allegheny College
gkapfham@allegheny.edu

***Abstract*—The published studies of regression testing methods often contain many of the hallmarks of high quality empirical research. Beyond features like clear descriptions of the methodology and the visualization and statistical analysis of the data sets, certain papers in this field also provide some of the artifacts used in and/or produced by the experiments. Yet, the limited industrial adoption of regression testing techniques is due in part to a lack of comprehensive empirical evaluations. Moreover, the regression testing community has not achieved a level of experimental reproducibility that would fully establish it as a science. After identifying the challenges associated with evaluating regression testing methods, this paper advocates a way forward involving a mutually beneficial increased sharing of the inputs, outputs, and procedures used in experiments.**

***Keywords*-regression testing, reproducible research**

I. Introduction

Software testing techniques establish a confidence in the correctness of and isolate defects within a program by running a collection of tests known as a test suite. These tests often operate by placing the program into a known state, executing one of the program's methods, capturing the output of this method under test, and comparing the method's actual output to the anticipated return value. When the actual output is the same as the expected output, then the test passes and a tester becomes more certain that the program is correct. Alternatively, different values of the actual and expected output signal a test case failure and suggest that there may be a fault in the program under test. Even though testing is conceptually simple and may be both expensive and error-prone, the field continues to attract considerable interest in both industry and academia. For instance, Bertolino notes that four out of the twelve research track sessions at the 28th International Conference on Software Engineering (ICSE 2006) had the theme of "Test and Analysis" [1].

Regression testing is one noteworthy testing method that involves repeatedly running a test suite whenever the program under test and/or the program's execution environment changes [2], [3]. Executing a regression test suite upon the introduction of either a defect fix or a new feature ensures that the modification of the program does not negatively impact the overall correctness of the software system. Previous reports from industry suggest both that software engineers frequently employ regression testing techniques [4] and that the use of regression testing methods often leads to a software application with high observed quality [5]. Furthermore, Yoo and Harman's survey of 159 papers in the field of regression testing, spanning the years 1977 to 2009, reveals that this growing and productive field of research comprises a wide variety of useful techniques for making the re-testing process more efficient and effective [3].

The rise of research in software testing in general, and of regression testing in particular, corresponds to a commensurate increase in interest for the field of empirical software engineering. For example, during the introduction of the first plenary session at the 31st International Conference on Software Engineering (ICSE 2009), Fickas noted that ICSE attendees self-identified themselves as most interested in the topic of "empirical software engineering" [6]. Further analysis of the ICSE 2009 attendee preferences uncovers the fact that the most popular pairs of interests are "analysis and testing" with "dependability" and "analysis and testing" with "empirical software engineering" [6]. Moreover, both Bertolino [1] and Harrold [7] identify the fundamental role that empirical studies must play in advancing the state of practice and enhancing the body of knowledge in testing.

Empirical studies currently play a role in software testing research, even though Juristo et al. determined that more than half of a surveyed body of knowledge about testing methods is influenced by "intuition, fashion, or market-speak" [8]. Interestingly, regression testing research often leads the testing community in the presentation of empirically supported results. For instance, Do et al. describe the software-artifact infrastructure repository (SIR) that furnishes programs and experiment designs often used for controlled experimentation in regression testing [9]. As two further illustrative examples, Do and Rothermel's study of mutation faults in test suite prioritization [10] and Li et al.'s examination of search techniques for test reordering [11] both exhibit characteristics of high quality empirical research: clear descriptions of the methodology, insightful visualizations, rigorous statistical analyses, and a discussion of the threats to experimental validity. In the case of Li et al.'s paper, a careful search on the Internet reveals that the authors have even provided the coverage reports used in the experiments, thus enabling a partial replication of the study.

978-0-7695-4345-1/11 $26.00 © 2011 IEEE
DOI 10.1109/ICSTW.2011.88

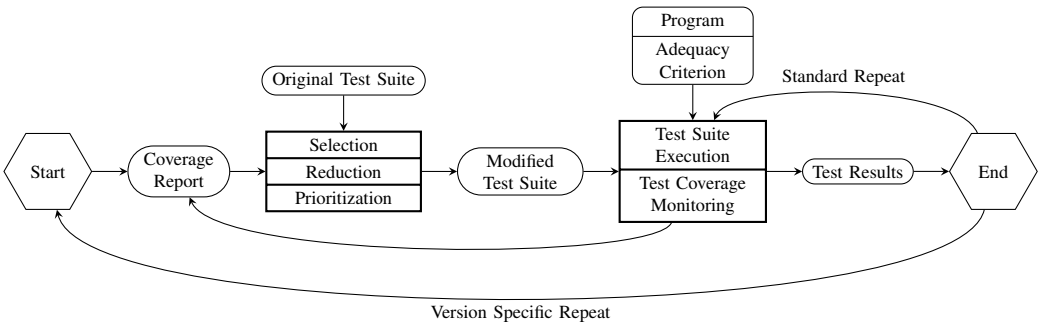

Figure 1. A Model for the Process of Regression Testing.

While acknowledging the noteworthy characteristics of [10] and [11] – and the many other articles like them – this paper asserts that the empirical study of regression testing methods must take several crucial steps forward in order to positively impact industry and truly advance this scientific discipline. From an industrial perspective, Ostrand and Weyuker note that few practitioners are willing to incorporate new testing methods into their development process because of a "lack of empirical studies" [12]. Furthermore, a scientist's vantage point reveals that the experimental assessment of regression testing techniques could potentially stagnate due to the relative dearth and general inaccessibility of suitable: (i) free and open source software (FOSS) tools to support regression testing, (ii) complete implementations of regression testing techniques, (iii) coverage reports for a wide variety of programs and test suites, (iv) full frameworks for conducting experiments, (v) data sets that describe both the efficiency and effectiveness of testing methods, (vi) routines for data visualization and statistical analysis, and (vii) cached copies of all the relevant intermediate results.

As a potential way to both improve industrial adoption and revitalize empirical research, this paper encourages the regression testing community to affirm the assertion made by Buckheit and Donoho: "for a field to qualify as a science, it is important first and foremost that published work be reproducible by others" [13]. Since it is often difficult to discern what qualifies as reproducible research, this paper advocates the adherence to the replication standard that King articulated in 1995 as "sufficient information exists with which to understand, evaluate, and build upon a prior work if a third party can replicate the results without any additional information from the author" [14]. Following the replication standard will have obvious benefits such as enabling the exact and differentiated replication of prior experiments [15] and lowering the barriers to entry for new researchers in the field [16]. The adoption of King's standard will enable researchers to more easily locate their own data sets and repeat their experiments [13], while often increasing the number of citations to papers with replication support [17].

With a firm foundation in empirical methods and a number of past papers that will enable partial replication, the regression testing community is poised to become a full-fledged scientific discipline that affects the practice of engineering software. In contrast to prior related work, this paper casts a vision for experimental studies that is both customized for regression testing and more ambitious than previous proposals like those from Barr et al. [15]. As an extension to Do et al., this paper advocates for the sharing of all experimental artifacts, instead of primarily focusing on "programs, versions, test cases, faults, and scripts" [9]. Unlike Nelson et al.'s proposal of a new framework for experimentation and a concentration on the sharing of data and experiments [16], this paper stresses the importance of using existing tools and further encourages the release of both the statistical analysis and visualization methods and the cached copies of all the relevant intermediate results. Finally, the references section of this paper forms a reading list that, to the best of this author's knowledge, has never been presented in either this community or the broader group of software engineering researchers and practitioners.

II. Challenges of Empirical Studies

In support of a concrete discussion of the challenges associated with experimentally studying regression testing techniques, Figure 1 furnishes one possible model for the process of regression testing. In this diagram, a box with rounded corners represents an input or output while a standard box denotes a procedure. For instance, a coverage report is an input to the regression testing procedure that commonly involves test suite selection, reduction, and/or prioritization [2], [3]. Even though tools for test suite execution are commonly available, an empirical study of regression testing initially confronts the challenge related to the lack of tool support for selecting, removing, and reordering the tests within real-world test suites. In fact, to the best of this author's knowledge, Smith and Kapfhammer were the first to describe, experimentally evaluate, and release a free and open source framework containing some of the commonly used regression testing algorithms (e.g., Harrold Gupta Soffa and delayed greedy for reduction and additional greedy and 2-optimal greedy for prioritization) [18].

Of course, the regression testing methods normally require as input a report furnishing the coverage or fault detection information on a per-test case basis [2]. Regrettably, most

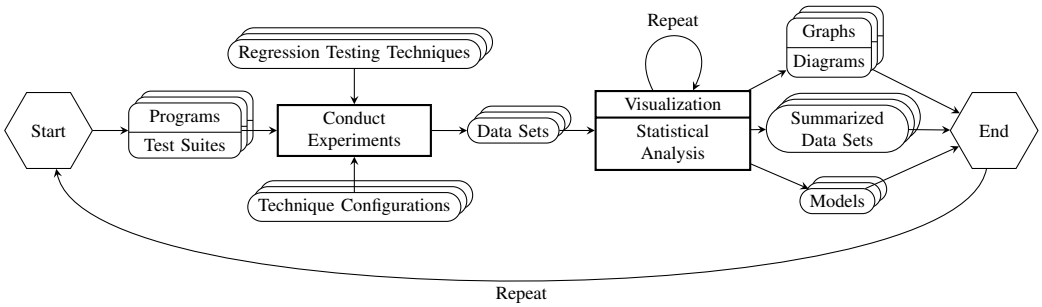

Figure 2. A Model for the Process of Experimentally Evaluating Regression Testing Methods.

coverage monitoring and mutation analysis tools, such as Cobertura and Javalanche, default to only storing this information at the granularity of the entire test suite. Moreover, with the exception of articles like Li et al. [11], many regression testing papers do not give the coverage reports, modified test suites, or testing results associated with the empirical study. Without the ability to create new inputs, the authors of 56% of the 159 papers surveyed by Yoo and Harman only used the programs available from the software-artifact infrastructure repository [3]. While the use of the SIR programs aids in the comparison between different techniques, ultimately it is a threat to the external validity of the experiments and thus a limit on the field's growth.

Figure 2 shows one representative way to empirically evaluate regression testing methods. Building on the graphical standards established in Figure 1, this diagram uses the same node shapes while also adding node replicas to indicate multiple inputs of the same type. For example, the process of conducting an experiment will normally take as input multiple regression testing techniques, programs, test suites, and tool configurations. Of course, the sheer number of input combinations makes it difficult and time-consuming to conduct an experiment in this field.

Yet, many experiments do not consider a sufficient number of configurations, conduct an appropriate number of trials, or create data sets containing information about both the efficiency and effectiveness of the studied methods. For instance, while Arcuri and Briand ask researchers to conduct 1,000 trials when using randomized algorithms [19], few papers have met this standard. Moreover, not many of the articles surveyed in [2] and [3] report on the efficiency of the regression testing methods, yielding an empirical picture that is incomplete. Finally, papers reporting on the experimental evaluation of a regression testing method do not release the framework used to conduct the experiments, thus severely limiting the ability of others to replicate the results [16].

As indicated in Figure 2, an experimental study normally involves the iterative visualization and statistical analysis of the data sets. While difficult to directly discern from reading the published literature, it seems that few researchers use data mining methods to explore the patterns in their data sets, thus suggesting that some important trends may never be reported. Moreover, certain papers contain analyses and visualizations created with commercial statistical software and the vast majority of papers do not release the source code used for this task. Although it would enable better adherence to the replication standard, few papers share the graphs, diagrams, summarized data sets, and models produced by the visualization and statistical analysis procedure.

III. A Potential Way Forward

While the regression testing community has made notable advances in experimental methodology, the evidence in Section II suggests that the field has not achieved the standards of scientific rigor and replication as respectively advocated by Buckheit and Donoho [13] and King [14]. Broadly speaking, this paper responds to this predicament by recommending that researchers share the procedures, inputs, and outputs depicted in the processes of Figures 1 and 2. While releasing these materials from a personal Web site is acceptable, whenever possible authors should deposit the deliverables from experiments in established infrastructures for data sharing, such as the Dataverse Network [20].

When considering the sharing of the artifacts used during experimentation, the costs to the individual researcher and the benefits to the overall community are obvious. While it will certainly take time for researchers to prepare all of the items in Figures 1 and 2, the use of proven tools such as the R language for statistical computing [21] and the Dataverse Network [20] may ease this burden. Beyond accelerating the pace of scientific innovation and enabling the replication of experiments, sharing will lower the barriers to entry for new researchers. However, distributing these items will also profit the individuals who elect to share by allowing them to better locate and expand upon the artifacts from their own prior work, ultimately enabling the investigation of new ideas sooner and faster. Additionally, sharing research data in other scientific and medical fields leads to an increase in the citation rate of the sharing paper [17]. Even though their findings are not tailored to regression testing research, Piwowar et al. remarkably observe that 48% of cancer microarray clinical trial papers with publicly available data received 85% of the aggregate citations, regardless of matters like the impact factor of the publishing journal and the chosen data sharing method [17].

IV. Practical Suggestions and Conclusion

Since sharing, in and of itself, will not solve all of the challenges mentioned in Section II, this paper gives a series of practical suggestions for improving the empirical assessment of regression testing methods. To start, researchers should consider using and contributing to the development of existing tools (e.g., [18], [22]) that select, reduce, and prioritize regression test suites. When developing new FOSS tools for regression testing, coverage monitoring, and test suite execution, community members should ensure integration with the popular xUnit framework and allow for the production and use of coverage and fault information on a per-test case basis. Developers should carefully engineer the tool support since the experience of the author and his colleagues suggests that this can make a real difference in both the efficiency and effectiveness of testing methods [23].

When implementing tools that conduct experiments and analyze results, researchers should consider using the R language for statistical computing since it provides, among many other features, advanced facilities for the analysis of software engineering data [19], easy to use data mining techniques that produce high quality visualizations [24], methods for sharing complex statistical models [25], and integration with a major infrastructure for data sharing [20]. Moreover, developers should always construct their experimentation frameworks to default to reporting and analyzing both the efficiency and effectiveness of the studied methods. During the data visualization and analysis phase, researchers should consider using data mining algorithms to make, for instance, hierarchical, non-parametric, easy-to-interpret tree models that enable the study of approaches to regression testing without making assumptions concerning the relationship between the explanatory and response variables [22].

In conclusion, the regression testing community has much to celebrate. Adherence to this paper's standards and suggestions may give further cause for celebration as the field becomes a full-fledged scientific discipline that truly impacts theory, experimentation, and practice. Yet, the advised potential way forward may seem daunting at first glance. Perhaps the best strategy is to proceed incrementally by picking *one* element from Figures 1 and 2 and deciding to share it as part of your next paper. Are you ready to usher in the scientific future of regression testing research through advances in sharing, useful tool support, and sophisticated data analysis? Your participation is welcomed and anticipated!

References

[1] A. Bertolino, "Software testing research: Achievements, challenges, dreams," in *Proc. of FOSE*, 2007.

[2] G. M. Kapfhammer, "Regression testing," in *The Encyclopedia of Software Engineering*. Taylor and Francis, 2010.

[3] S. Yoo and M. Harman, "Regression testing minimization, selection and prioritization: a survey," *Software Testing, Verification and Reliability*, 2010.

[4] A. K. Onoma, W.-T. Tsai, M. Poonawala, and H. Suganuma, "Regression testing in an industrial environment," *Communications of the ACM*, vol. 41, no. 5, 1998.

[5] M. Gittens, H. Lutfiyya, M. Bauer, D. Godwin, Y. W. Kim, and P. Gupta, "An empirical evaluation of system and regression testing," in *Proc. of CASCON*, 2002.

[6] S. Fickas, "Plenary session presentation," 31st International Conference on Software Engineering, 2009.

[7] M. J. Harrold, "Testing: a roadmap," in *Proc. of FOSE*, 2000.

[8] N. Juristo, A. M. Moreno, and S. Vegas, "Reviewing 25 years of testing technique experiments," *Empirical Software Engineering*, vol. 9, 2004.

[9] H. Do, S. G. Elbaum, and G. Rothermel, "Supporting controlled experimentation with testing techniques: An infrastructure and its potential impact." *Empirical Software Engineering*, vol. 10, no. 4, 2005.

[10] H. Do and G. Rothermel, "On the use of mutation faults in empirical assessments of test case prioritization techniques," *Transactions on Software Engineering*, vol. 32, no. 9, 2006.

[11] Z. Li, M. Harman, and R. M. Hierons, "Search algorithms for regression test case prioritization," *Transactions on Software Engineering*, vol. 33, no. 4, 2007.

[12] T. Ostrand and E. Weyuker, "Software testing research and software engineering education," in *Proc. of FoSER*, 2010.

[13] J. B. Buckheit and D. L. Donoho, "WaveLab and reproducible research," Department of Statistics, Stanford University, Tech. Rep. 474, 1995.

[14] G. King, "Replication, replication," *PS: Political Science and Politics*, vol. 28, no. 3, 1995.

[15] E. Barr, C. Bird, E. Hyatt, T. Menzies, and G. Robles, "On the shoulders of giants," in *Proc. of FoSER*, 2010.

[16] A. Nelson, T. Menzies, and G. Gay, "Sharing experiments using open-source software," *Software: Practice and Experience*, vol. 41, no. 3, 2011.

[17] H. A. Piwowar, R. S. Day, and D. B. Fridsma, "Sharing detailed research data is associated with increased citation rate," *PLoS ONE*, vol. 2, no. 3, 2007.

[18] A. M. Smith and G. M. Kapfhammer, "An empirical study of incorporating cost into test suite reduction and prioritization," in *Proc. of 24th SAC*, 2009.

[19] A. Arcuri and L. Briand, "A practical guide for using statistical tests to assess randomized algorithms in software engineering," in *Proc. 33rd ICSE*, 2011.

[20] G. King, "An introduction to the dataverse network as an infrastructure for data sharing," *Sociological Methods and Research*, vol. 36, no. 2, 2007.

[21] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, 2011, http://www.R-project.org.

[22] A. P. Conrad, R. S. Roos, and G. M. Kapfhammer, "Empirically studying the role of selection operators during search-based test suite prioritization," in *Proc. of 12th GECCO*, 2010.

[23] R. Just, G. M. Kapfhammer, and F. Schweiggert "Using conditional mutation to increase the efficiency of mutation analysis," in *Proc. of 6th AST*, 2011.

[24] G. J. Williams, "Rattle: A Data Mining GUI for R," *The R Journal*, vol. 1, no. 2, 2009.

[25] W.-C. L. Alex Guazzelli, Michael Zeller and G. J. Williams, "PMML: An Open Standard for Sharing Models," *The R Journal*, vol. 1, no. 1, 2009.