Assuring Software Quality Through the Use of Fuzzy Testing

David Jefts
SE625, Software Quality Assurance
Embry-Riddle Aeronautical University

***Abstract*—*THIS IS NOT MY ABSTRACT* Software engineering researchers solve problems of several different kinds. To do so, they produce several different kinds of results, and they should develop appropriate evidence to validate these results. They often report their research in conference papers. I analyzed the abstracts of research papers submitted to ICSE 2002 in order to identify the types of research reported in the submitted and accepted papers, and I observed the program committee discussions about which papers to accept. This report presents the research paradigms of the papers, common concerns of the program committee, and statistics on success rates. This information should help researchers design better research projects and write papers that present their results to best advantage.**

This paper makes the case for TaaS—automated software testing as a cloud-based service. We present three kinds of TaaS: a “programmer’s sidekick” enabling developers to thoroughly and promptly test their code with minimal upfront resource investment; a “home edition” on-demand testing service for consumers to verify the software they are about to install on their PC or mobile device; and a public “certification service,” akin to Underwriters Labs, that independently assesses the reliability, safety, and security of software. TaaS automatically tests software, without human involvement from the service user’s or provider’s side. This is unlike today’s “testing as a service” businesses, which employ humans to write tests. Our goal is to take recently proposed techniques for automated testing—even if usable only on toy programs—and make them practical by modifying them to harness the resources of compute clouds. Preliminary work suggests it is technically feasible to do so, and we find that TaaS is also compelling from a social and business point of view.

***Index Terms*—Fuzzy Testing, Fuzzing, Software Quality Assurance, Software Vulnerabilities, Testing, Automated Testing, Error Detection**

I. Introduction

ENSURING that software works properly before public (or private) release is important because software bugs, or in-code errors, can cost the developing company a lot of money. Software bugs cause program faults, and faults cause program failures. Maintenance and repair is 65-85% of system cost [1]. Naturally, the goal during software development is to limit these defects by testing the software before release. However, “software testing is a very labor intensive and costly task, [and] many software testing techniques to automate the process of software testing have been [designed and] reported in the literature” [2]. Reportedly, the most effective technique is called *Assertion-Based* [3]. This involves finding program inputs in which various assertions are violated. When these assertions are violated then there is a fault in the program [3]. One of these automated techniques is *Fuzzy Testing*.

Why I chose this topic

Goal of this paper

II. Fuzzy Testing

What is fuzzy testing?

Fuzzy testing techniques

Examples of use of fuzzy testing in industry

Explain the context of these comparisons

Analyze potential applications and best-case uses for fuzzy testing

III. Fuzzy Testing as Compared To Other Software Testing Techniques

Advantages and disadvantages of fuzzy testing

Fuzzy Testing vs Other Fuzzy Testing techniques

IV. Conclusion

Is fuzzy testing better or worse than other methods of testing and for which cases is it better?

Appendix A

Stuffity stuff stuff stuff

Appendix B

More stuffy things

References

[1] M. Towhidnejad, “Lecture 4-defects,” February 2019, in-class presentation.

[2] A. M. Alakeel, “Using fuzzy logic techniques for assertion-based software testing metrics,” *The Scientific World Journal*, vol. 2015, 2015, copyright - Copyright © 2015 Ali M. Alakeel. Ali M. Alakeel et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited; Last updated - 2018-10-09. [Online]. Available: http://search.proquest.com.ezproxy.libproxy.db.erau.edu/docview/1679

[3] B. Korel and A. M. Al-Yami, “Assertion-oriented automated test data generation,” in *Proceedings of IEEE 18th International Conference on Software Engineering*, March 1996, pp. 71–80.

[4] G. Candea, S. Bucur, and C. Zamfir, “Automated software testing as a service,” in *Proceedings of the 1st ACM Symposium on Cloud Computing*, ser. SoCC ’10. New York, NY, USA: ACM, 2010, pp. 155–160. [Online]. Available: http://doi.acm.org.ezproxy.libproxy.db.erau.edu/10.1145/1807128.1807

[5] K. K. F. Yuen and H. C. Lau, “A fuzzy group analytical hierarchy process approach for software quality assurance management: Fuzzy logarithmic least squares method,” *Expert Systems with Applications*, vol. 38, no. 8, pp. 10 292 – 10 302, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417411002636

[6] S. Bekrar, C. Bekrar, R. Groz, and L. Mounier, “Finding software vulnerabilities by smart fuzzing,” in *2011 Fourth IEEE International Conference on Software Testing, Verification and Validation*, March 2011, pp. 427–430.

[7] B. Arkin, S. Stender, and G. McGraw, “Software penetration testing,” *IEEE Security Privacy*, vol. 3, no. 1, pp. 84–87, Jan 2005.

[8] I. van Sprundel, “Fuzzing: Breaking software in an automated fashion,” Web, Chaos Computer Club, December 2005.