

## **Software Testing & Reliability**

Justin R. Pike

CSCI 315: Data Structures Analysis

Sean T. Hayes

September 19, 2022

## Software Testing & Reliability

In an increasingly interconnected world that is built on computers and software, the importance of requirement of software testing is indisputable. The current processes that exist within the industry, although improving, are expensive, unpredictable in their effectiveness and largely completed in an ad-hoc manner (Bertolino). Further clouding the picture is the question of the extent to which certain programs need to be tested from a business and services perspective. The impacts of not completing enough testing or testing too much must balance the risks and nature of software failures and the increased costs associated with exhaustive testing. In order to reach a determination of satisfactory performance and minimization of risk, a standardized framework of testing and evaluation should be adopted by the industry thereby removing the potential ethical dilemmas that inevitably occur when the question of money is introduced.

As technology professionals, one of our core responsibilities, according to the Institute of Electrical and Electronics Engineers, is to “hold paramount the safety, health, and welfare of the public” (“IEEE Code of Ethics”). There is little doubt that anyone would ever intentionally cause harm or intend for their whatever they are working on to intentionally fail. In fact, in perfect world, each of us would choose for our creations to work flawlessly as intended every single time. However, to reach this level of certainty in any product testing requires an extensive amount of testing. The general factors taken into account when testing is completed or being decided upon are the risks due to failure and costs associated with increased time and development (“Planning Report 02-3”). This cost benefit analysis must be examined carefully, and risks of failure must be fully understood to make the proper determination of the amount of testing. Is there a good enough scenario, when the risk to consumer is minimal or is the risk much more substantial and the software in question is now a high assurance item (Howden). In

the case of the THERAC-25 incidents the software in question had a high-risk factor due to potential harm to the individual person should there be a software failure. The most obvious solution would be to complete exhaustive testing in such cases; however, this additional testing comes with increased costs that are then passed on to the consumer who then may not be able to afford it or even live to see it if the time takes too long. Much like other approval processes such as the pharmaceutical industry, a framework of testing and approval processes could be potentially implemented in order to ensure proper functionality, reduce risk to the person and remove the ethical decision from those in charge. In other low risk cases, where the risk to the individual, again determined by outside organization, the amount of testing should be left up to the individual or company. The quality and reliability of this product, most likely a convenience type product, drives the sales of such an item and the company and will attempt to balance quality and costs.

While implementing a system of standardization and approval for software and the products that they support may be part of a solution to minimizing risk and reducing ethical questions, the implementation of a certification process for software engineers could also be an effective step towards improving software reliability and reducing costs when certain high-risk programs are being written. God has given everyone a variety of gifts and calls on them us to use them to serve one another. A certification process and professionalization of the software engineer field is a tangible way to ensure that serve others to the highest degree possible with our gifts (1 Pet. 4:10 New Living Translation) . Certification processes and requirements for non-software engineers, such as structural engineers, have been developed over the last century in the United States. Not only do these processes ensure that highly-skilled individuals are executing high-risk tasks, but they also hold these individuals accountable to their work and plans. The

Association for Computing Machinery directs computing professionals to “follow generally accepted best practices” ( “ACM Code of Ethics and Professional Conduct”). These best practices should be industry standards and requirements to be followed by all programmers instead of polite suggestions for use. Holding software engineers accountable will promote higher quality and more reliable work that will ultimately lead to less cost associated with testing.

The questions surrounding software will never be of a need for testing, only how much testing needs to be done. Software testing is completed to ensure the proper functionality of the software and its implementation with hardware and reduces the risks the consumer of the products but comes at the price of increased costs. Standardization of testing processes by governing bodies and certification of software engineers are two steps that could be taken to help improve the software testing timeliness and reliability as well as reduce ethical questions surrounding the amounts of testing.

## References

“ACM Code of Ethics and Professional Conduct.” *Code of Ethics*, <https://www.acm.org/code-of-ethics#h-1.-general-ethical-principles>.

Bertolino, A., "Software Testing Research: Achievements Challenges Dreams", *Future of Software Engineering (FOSE '07)*, pp. 85-103, 2007.

Howden W. H., "Good enough versus High Assurance Software Testing and Analysis Methods", *Proc. of IEEE HASE*, 1998.

“IEEE Code of Ethics”. <https://www.ieee.org/content/dam/ieee-org/ieee/web/org/about/corporate/ieee-code-of-ethics.pdf>.

*New Living Translation Bible (2013)*. Tyndale House Publishers, Inc. (Original work published 1996)

*Planning Report 02-3 The Economic Impacts of Inadequate Infrastructure for Software Testing - EPFL*. EPFL. (n.d.). Retrieved September 19, 2022, from [https://lara.epfl.ch/w/\\_media/misc/rti02economicimpactsinadequateinfrastructuresoftwaretesting.pdf](https://lara.epfl.ch/w/_media/misc/rti02economicimpactsinadequateinfrastructuresoftwaretesting.pdf)