

How Much Testing is Enough?

Alena Durel

CSCI 315

Dr. West

26 October 2021

The more testing that a system requires to achieve a higher level of reliability also causes the cost of the system to rise, which might cause it to become inaccessible to people who cannot afford it. A developer must consider the degree of risk, both financial and personal, the public faces. Hence, this begs the question, how much testing is necessary?

According to a Communications of the ACM article, “Greater numbers of life critical systems [...] are controlled by computers. [...] Consequently, lives, well-being, and quality-of-life are vulnerable to poor system design and failure” (Nissenbaum 74). Due to these systems being “life critical”, financial risk cannot be the only consequence observed, rather the reliability of the software is crucial because it affects the well-being of an individual should the software fail. In the IEEE Code of Ethics, one of the first values is “to hold paramount the safety, health, and welfare of the public, [...] and to disclose promptly factors that might endanger the public or the environment”. Therefore, it would be unethical for a developer to present a less reliable system. Less testing might cut costs, but it will also decrease the reliability of the system and as such, cause issues that threaten the welfare of the public.

Thus, testing a program is necessary. While there is no guarantee that a program will be perfect and the software is likely to have some minor issues even after several rounds of testing, to overestimate a program’s reliability or undertest it would be unethical and irresponsible. Deuteronomy 22:8 states, “when you build a new house, make a parapet around your roof so that you may not bring the guilt of bloodshed on your house if someone falls from the roof” (*NIV*). Consideration must go into the creation of software, to ensure the safety of the users. How the program will be tested, and what level of testing is necessary must be thought of beforehand.

Now, how do we determine how much testing is needed? The developer has an obligation to set testing goals at the beginning of the development process and meet them, warn the user

about untested areas of the software, and educate them about the limitations. Software should also not hold sole responsibility for maintaining the safety of the system. In a Communications of the ACM article, the author states that “life-preserving software should have backups in case of failure [...] This can mean physical backups, physical fail-safe features, computer backups, human monitoring, and so on” (Collins et al. 86). While these fail-safes might increase the cost of a system, they also add a level of safety to the product as the system is not relying solely on software.

Any type of product must be reliable before it reaches the end-user. Engineers must often complete a certification process to ensure that they can produce reliable products and it can be asked whether a similar process should apply to software engineers. Such a certification would be beneficial, especially as more “life critical” systems rely heavily on software. Just as engineers must prove that they can produce reliable products so that their work does not threaten the well-being of the public, a software engineer should have to do the same so that the systems they work on will not also become risks.

1 Corinthians 8:9 states, “but take care that this right of yours does not somehow become a stumbling block to the weak”, and later in 10:23 Paul writes “‘everything is permissible’ – but not everything is beneficial. ‘Everything is permissible’ – but not everything is constructive”. When a developer is given a system to work on, they have an obligation to make sure that it meets the standards the client is looking for and that it is a reliable piece of software. Such a certification could help to ensure that the programmers are held to a certain standard when working on these life critical systems and to ensure that the systems receive the proper testing to establish a reliable product.

Works Cited

Collins, W. Robert, et al. "How Good Is Good Enough?" *Communications of the ACM*, vol. 37, no. 1, 2 Jan. 1994, pp. 81–91., <https://doi.org/10.1145/175222.175229>.

"IEEE Code of Ethics." *Institute of Electrical and Electronics Engineers*, www.ieee.org/about/corporate/governance/p7-8.html.

New International Version. Biblica, 2011. *Bible Gateway*, www.biblegateway.com/versions/NewInternational-Version-NIV-Bible/

Nissenbaum, Helen. "Computing and Accountability." *Communications of the ACM*, vol. 37, no. 1, 2 Jan. 1994, pp. 72–80., <https://doi.org/10.1145/175222.175228>.