

It never fails, get a group of software engineers together and ask about testing. Eyes will roll and throats will grumble. Rarely will you get enthusiastic confirmation about this part of the development. Yet, testing critical software systems is perhaps the most important stage of development, and this aspect of the project is often given a back seat. Knowing these tendencies and understanding the necessity, many industries (ISO, IEEE, FAA, FDA, etc.) have identified and formalized software testing into two categories, Verification and Validation. Independence is required for safety-critical software; therefore, the complete activity is known as Independent Verification and Validation (IV&V).

Independent

It is fairly well known in the industry that software developers are poor testers of their own code. There are many reasons for this seemingly unjustifiable statement. Writing software is mostly a cerebral activity...software engineers read the requirements and then write the code to meet those requirements. Both the reading and the writing involve interpretation and judgement. How we interpret, and the judgements we make based on those interpretations, are open to scrutiny...and scrutiny clashes with egos!

Also, software developers who test their own systems, tend to "burn a path" through the code and forego other crucial paths. This gives a false sense of security and much of the software goes untested.

The best defense against shortfalls incurred by interpretation, judgement, ego and limited path testing is "independence." IV&V test engineers are trained to be objective. The perspective of the test engineer is to evaluate the code against the written requirements and provide objective evidence of the results. For certified systems, the evidence is prescribed by certification guidelines (e.g., DO-178C, IEC 60880, ISO 13485, IEC 62304). Good test engineers have been trained in the certification guidelines and have the requisite experience to provide unbiased evidence of the tested software.

Verification

The bottom line of testing is to verify that the software meets the requirements. Further, objective evidence is required to show that the tests and the testing are complete. IV&V test engineers develop robust test cases and completely test the many paths of the software. They have a deep understanding of the certification requirements (when testing for certification) and have experience with software testing tools. Whereas a typical software engineer will tend to test basic functionality, the IV&V test engineer will greatly expand the testing to include boundary tests, interface tests and other robustness checks.

Validation

Requirements form the foundation of good software development. How many times have you heard that a software development project was abandoned or delayed because the requirements were insufficient or poorly developed? Validation ensures that requirements are complete and are well defined. And, for certified systems (e.g., Avionics, Medical, Industrial, Transportation), validation ensures that the requirements meet the demands of the certification authority (e.g., FAA, FDA). For instance, safety-critical systems require that there is no unused, untested code in the system.

In Summary

Formalized, independent software testing provides assurance that the code was tested with the rigor and robustness required for the stated level of criticality. Further, IV&V reduces the cost of development by providing insightful knowledge of the completeness of the requirements (good requirements form the foundation of the software development process), and by providing broad spectrum test and coverage results of the code, uncovering issues before the code is deployed. This saves time and money, and assures high quality software and products.

GENERAL DIGITAL CORPORATION

8 Nutmeg Road South

South Windsor, Connecticut 06074

800.952.2535 | 860.282.2900

FOLLOW US ON SOCIAL MEDIA



Scan QR Code to
Download Tech Brief

