

Software Quality

by Chet Haibel, ASQ Certified Software Quality Engineer

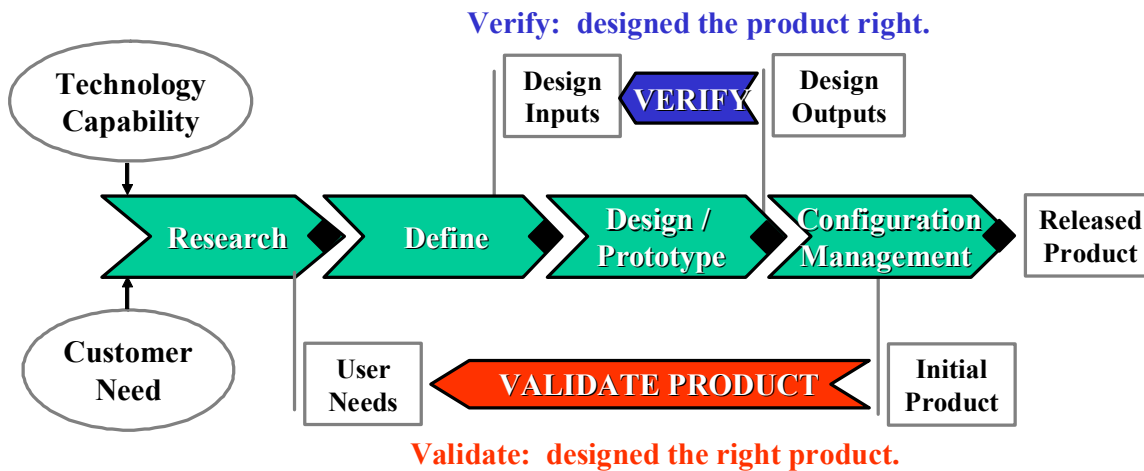
Low defect rates in software can only be achieved by using rigorous software engineering processes, greatly facilitated by writing software in modules suitable for general use, and then re-using successful modules at every opportunity.

For all but the most trivial software, it is impossible to thoroughly exercise all branches and paths. In fact only a small portion of branches and paths can be tested in a reasonable time. Therefore, attempting to “test in” software quality will be very expensive, frustrating, and ultimately unsuccessful. The consequence of this is **all complex software has defects**.

An appropriate use of testing is to characterize the quality of the software engineering process, carefully segregating the code which has been produced by type, author, complexity, etc. Testing thus becomes a way of sampling the software quality and should lead to acceptance and rejection of different portions of code. Ideally, the rejected code is recycled through the entire software engineering process, perhaps after simplifying a complex module into several less complex modules.

In spite of the foregoing comments, testing software programs is widely undertaken, some defects are indeed discovered and corrected, hopefully fewer new defects are introduced, and eventually imperfect software is released. Software maintenance continues this process as customers report defects and revisions are made throughout the entire life cycle of the software.

Software testing is often called Software V&V for verification and validation. There is a difference between the two, but the words are misused so often that even the definitions are not generally agreed upon. Simplistically, product design looks like the following picture, which at least serves to define the difference between verification and validation:



Ideally, software V&V revolves around unambiguous, complete, non-conflicting requirements. These requirements are usually refined as the software was generated, issues were discovered, and customers were presented with prototypes for comment. A reasonable model for this converging process is Barry Boehm’s Spiral Model. The Spiral model is an admission that a lot of learning happens when successively refined prototypes are shown to customers. This is in contrast to the apparently clean, generally accepted Waterfall Model, shown above.

At some point, requirements are frozen into the Design Inputs, typically called a Software Requirements Specification (SRS).

Not far behind the SRS in completion, a Verification Plan and the resulting test cases are completed and testing may begin. Actually, a lot of testing of low level modules has taken place all along and these test cases focus on fairly integrated code.

Given that testing will be undertaken, there are two classes of software modules that should be given focused attention to stretch software test resources as far as possible. The first class is made up of those modules with a high complexity as rated by McCabe’s Cyclomatic Index, Halstead’s measures, or similar complexity metric. The second class is made up of those modules used most frequently in typical operation.

It may take several iterations to get to a targeted level of Verification. Depending on the severity of the changes, Regression Testing focusing on what was changed may be sufficient. Sometimes a complete fresh start of all test cases is necessary because the effect of even a fairly minor software change can often propagate far and wide. Some sage has said there are no small software changes. At some point it is time to move on to fully integrating the product.

Now is the time to go back to the customers who helped define the User Needs for Validation. Often the same customers are not available. Often the same customers have seen something new and raised their expectations. Often the same customers have simply forgotten what they said or changed their mind about what they want. This phase is sometimes called Beta Testing and a whole bunch of new customers are brought into the process. Inevitably more changes are made, and after a certain amount of Regression Testing (both Verification and Validation) is completed, the software product is released.