Verification vs Validation

According to Sommerville (2011), the goodness of fit of a software system can be assessed with two concepts: verification and validation. Both verification and validation can help determine if we are producing a “good” system. As Sommerville puts it, with verification we are concerned with “are we building the product **right**”, whereas with validation the concern is “are we building the right **product**”. The former question can be answered by addressing whether the software adheres to its own specifications. The latter, is concerned with meeting the needs of the user. Both of these considerations are necessary in determining if the product we are building is correct.

A table I found particularly useful in contrasting verification and validation is reproduced here:

|  |  |  |
| --- | --- | --- |
| Criteria | Verification | Validation |
| Definition | The process of evaluating work-products (not the actual final product) of a development phase to determine whether they meet the specified requirements for that phase. | The process of evaluating software during or at the end of the development process to determine whether it satisfies specified business requirements. |
| Objective | To ensure that the product is being built according to the requirements and design specifications. In other words, to ensure that work products meet their specified requirements. | To ensure that the product actually meets the user’s needs, and that the specifications were correct in the first place. In fother words, to demonstrate that the product fulfills its intended use when placed in its intended environment. |
| Question | Are we building the product right? | Are we building the right product? |
| Evaluation Items | Plans, Requirement Specs, Design Specs, Code, Test Cases | The actual product/software. |
| Activities | Reviews  Walkthroughs  Inspections | Testing |

(Softwaretestingfundamentals.com, 2011).

Besides defining validation and verification, it is important to note that although a software product will pass one test, it may not pass the other (Softwaretestingfundamentals.com, 2011). Verifying software such that it meets its specifications is important, but even if it passes this criteria it may still not be “good” software. For example, If the product does not meet the customer’s needs, it doesn’t matter how functional the code is in itself. In this sense, validation is as important, if not more so, than verification and should be considered through the software development life-cycle.

References

Softwaretestingfundamentals.com (2011). “Verification vs Validation.” Retrieved from <http://softwaretestingfundamentals.com/verification-vs-validation>

Sommerville, I. (2011). “Chapter 8” Software Engineering, Ed. 9.