VERIFICATION vs VALIDATION

The terms ‘**Verification**‘ and ‘**Validation**‘ are frequently used in the software testing world but the meaning of these terms are mostly vague and debatable. You will encounter (or have encountered) all kinds of usage and interpretations of those terms, and it is our humble attempt here to distinguish between them as clearly as possible.

|  |  |  |
| --- | --- | --- |
| **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 other 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 |

It is entirely possible that a product passes when verified but fails when validated. This can happen when, say, a product is built as per the specifications but the specifications themselves fail to address the user’s needs.

* Trust but Verify.
* Verify but also Validate.