LAST UPDATED ON JUNE 14, 2018
PLUTORA BLOG - TEST CASE MANAGEMENT, TEST MANAGEMENT

Verification vs Validation: Do You know the Difference?

Reading time 6 minutes

confusion. While the distinction may seem trivial, the two fulfill very separate purposes.

Imagine being asked to do a verification on a certain project but hold off on the validation. Our first question might be how are they different? When would you start, and what would that work look like?

If the difference between them is a little confusing, you're not alone –countless development and testing professionals are in the same boat. So, whether you are completely confused or just fuzzy on the details, hopefully we'll make it crystal clear.

The differences between them are significant.

Verification

Software Engineering standards known as IEEE-STD-610 defines "Verification" as:

"A test of a system to prove that it meets all its specified requirements at a particular stage of its development."

The last phrase of the definition, "at a particular stage of its development" is the key part of verification. Before coding begins on any application, a set of specifications will have been outlined. The verification of development refers to checking application that is still being developed to ensure that it adheres to these specifications. These checks could be something as simple as reading the specifications and comparing them against the code logic to make sure they line up. The verification process will include activities like code reviews, walkthroughs, inspections but little, if any, actual testing.

be regularly checked and compared against the various landmarks along the route. For example, head west until you cross the river, turn north at the store and so forth.

With instructions like this, the driver is verifying the route against the directions that were provided.

Here's another example. During the development of a spreadsheet, the basic mathematical functions need to be verified that their individual calculations are accurate before they can be applied to more complex code and eventually formulas. This type of testing is done alongside of the development to ensure that each new step meets the predefined specifications. The value of verification testing is realized when development is complete, and the application functions as expected.

This type of testing helps to shift the identification and resolution of any bugs further left (earlier on in the application lifecycle). This means significant cost and time savings on the overall project. The reasoning is simple – it is far easier and more efficient to fix a small bug as it's created, than later on when hundreds of lines of code have to be searched to find the same issue.

Validation

Validation, on the other hand, is quite different and serves a very different purpose. The definition of Validation according to IEEE-STD-610 is:

"An activity that ensures that an end product stakeholder's true needs and expectations are met."

is performed upon the completion of a given module, or even the completion of the entire application. Validation focuses on ensuring that the stakeholder gets the product they wanted.

The Validation effort doesn't care how you got there, only that you have arrived, and that everything is as expected. Going back to our example of the driver: if your planned destination was the beach, to validate your arrival at this location, you might ask some questions:

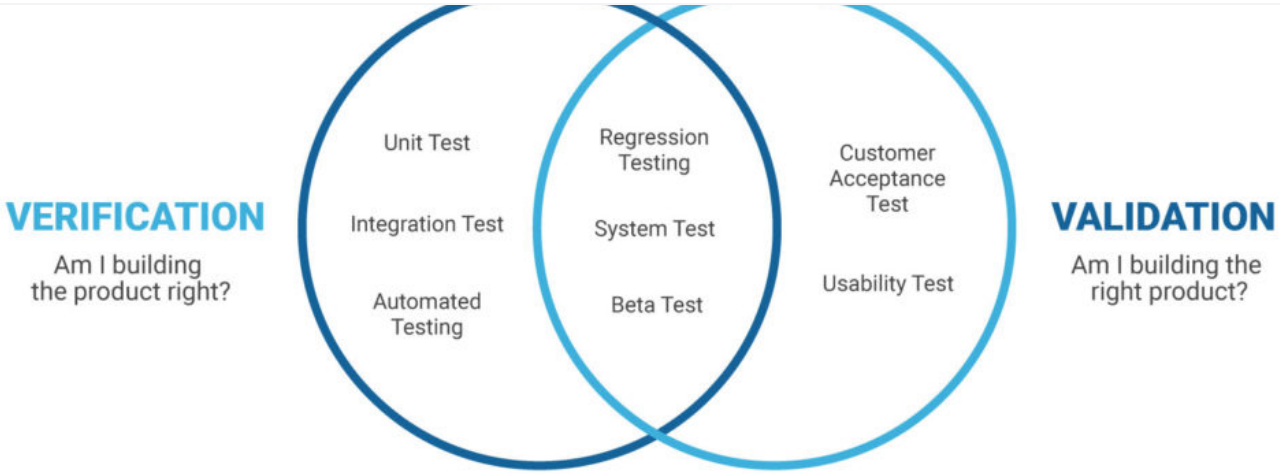
- Can I feel sand underfoot?
- Can I see the ocean and waves?
- Does this location meet my expectation of a beach?

These type of validation tests ensures only that your current location meets your expected criteria.

Using our example of creating a spreadsheet, once the development of the spreadsheet is complete, we would run validation tests to ensure that the finished product will meet the needs of the customer.

This is high level testing and typically consists of regression testing, user testing, performance testing and so on.

Summary



Now, let's go back to the original question. If you were asked to do a verification on a certain project but hold off on the validation, the answer now would be much clearer.

To start with, you would pull the original project specifications and then proceed to do a code review, walkthrough or code inspection to ensure that the pieces are being created as planned. Then, once the application development is completed, you would validate that the end product is in fact what the customer had requested.

This is one of those scenarios where words get easily confused because they look similar. So, to further assist in keeping them straight, we've created the chart below for a quick reference. Feel free to print it and pin it up over your desk.

	Verification	Validation
As per IEEE-STD-610:	"A test of a system to prove that it meets all its specified requirements at	"An activity that ensures that an end product stakeholder's true

The Process of:	Ensuring we are developing the product according to specifications.	Testing and validating the actual product to ensure we have developed it correctly.
Involves:	Little or no code execution	Code Execution
Activities Include:	Reviews, Walkthroughs, Inspections, Desk-checking, etc.	Black box testing, white box testing, non-functional testing, etc.
Activity Type:	Low-level	High-level
Method/Process Type:	A static method of checking documents and files	A dynamic process of testing the real product.
Target:	Application, software architecture, specifications, complete design, high level and database design, etc.	Actual product
Answers the Question:	Am I building the product right?	Am I building the right product?

Verifying or Validating?

Plutora is a continuous delivery management solution that will help you track test metrics across the entire enterprise. User information, versions, builds, test environments, test cases, requirements coverage, change management, defect management, automation, audit trails and even results and events from your favorite integrated tools – everything is captured to a data mart that decision makers can confidently use to release the product to production.

Learn more about the Plutora platform.

Dan Packer

Dan is an Industry Specialist at Plutora. Dan got his first taste of programming in high school, coding games in Basic. Since then, he has been directly involved with nearly every aspect of the Development and Release lifecycle — coding, testing, project management, team management, architecture, database, web & graphics designer, and much more. He has implemented development lifecycle methodologies for companies like Sears Financial, Novell, Sprint, Daimler-Benz Financial, Sabre, Centex and T-Mobile to name a few. In addition to his enterprise work, he has founded multiple companies, and continues to work as a business and technology advisor on various domestic and international projects. In total Dan has managed and orchestrated literally hundreds of deployments, development initiatives and thousands of iterative code enhancements.

Readers also check out

The Importance of Requirements Traceability Matrix

The Evolution of Test and the Enterprise Data Mart

Unit Testing in an CI/CD Enterprise Environment

The Regression Testing Solution for DevOps

Integration Testing in an Enterprise DevOps Environment

8 Things That Make a Test Management Tool Next Generation

Outcomes

Integrations

Optimize Continuous Delivery Pipelines

CIO

DevOps Manager

Release Manager

Environment Manager

Careers

Case studies

PLUTORA