

**G Suite. Free for 14 days**

All you need to do your best work, in one package that works seamlessly from any device.



OPEN

Verification vs Validation

VERIFICATION vs VALIDATION are hugely confused and debated terms in the software testing world. You will encounter (or have encountered) all kinds of usage and interpretations of these terms, and it is our humble attempt here to distinguish between them as clearly as possible.

Differences

Criteria	Verification	Validation
<i>Definition</i>	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.
<i>Objective</i>	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.
<i>Question</i>	Are we building the product <i>right</i> ?	Are we building the <i>right</i> product?

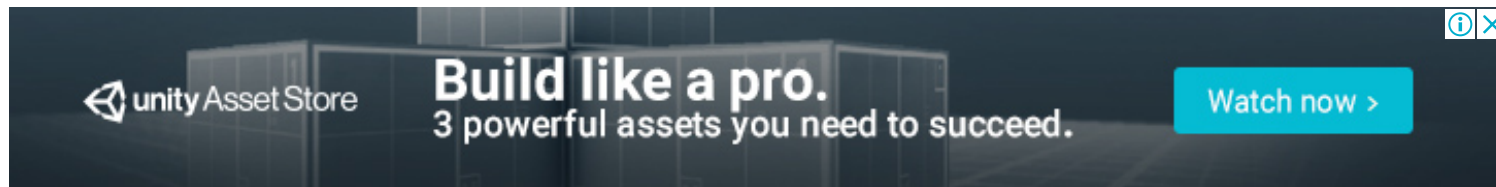


<i>Evaluation Items</i>	Plans, Requirement Specs, Design Specs, Code, Test Cases	The actual product/software.
-------------------------	--	------------------------------

<i>Activities</i>	<ul style="list-style-type: none">• Reviews• Walkthroughs• Inspections	<ul style="list-style-type: none">• Testing
-------------------	--	---

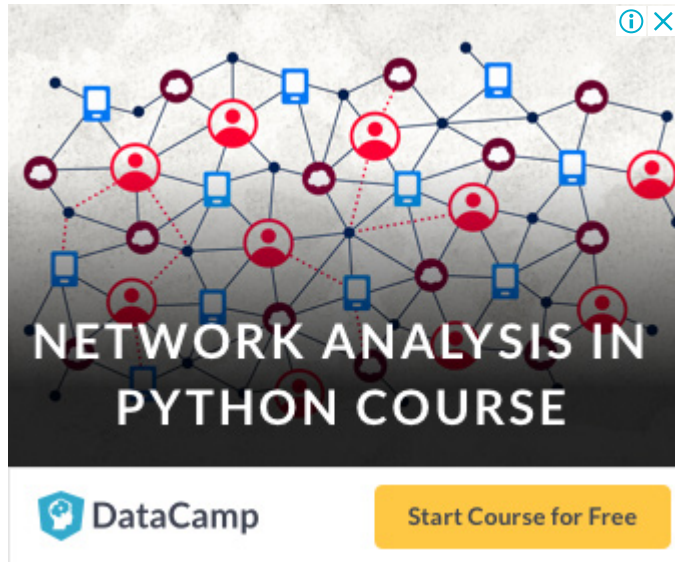
It is entirely possible that a product passes when verified but fails when validated. This can happen when, say, a product is built as per the specifications but the specifications themselves fail to address the user's needs.

- Trust but Verify.
- Verify but also Validate.

[Recommend 48](#)[Share](#)[G+ Share](#)[Tweet](#)[Stumble](#)**57**
Shares[◀ Defect Severity \(http://softwaretestingfundamentals.com/defect-severity/\)](#)[Defect Priority ▶ \(http://softwaretestingfundamentals.com/defect-priority/\)](#)

Search...





Enter your email address:

Subscribe

If you do not find the verification email, check your spam folder

Delivered by FeedBurner (<https://feedburner.google.com>)

Categories

📁 Artifacts (<http://softwaretestingfundamentals.com/category/artifacts/>)

📁 Basics (<http://softwaretestingfundamentals.com/category/basics/>)

📁 Career (<http://softwaretestingfundamentals.com/category/career/>)

📁 Defects (<http://softwaretestingfundamentals.com/category/defects/>)



📁 [Levels \(http://softwaretestingfundamentals.com/category/levels/\)](http://softwaretestingfundamentals.com/category/levels/)

📁 [Methods \(http://softwaretestingfundamentals.com/category/methods/\)](http://softwaretestingfundamentals.com/category/methods/)

📁 [Metrics \(http://softwaretestingfundamentals.com/category/metrics/\)](http://softwaretestingfundamentals.com/category/metrics/)

📁 [Resources \(http://softwaretestingfundamentals.com/category/resources/\)](http://softwaretestingfundamentals.com/category/resources/)

📁 [Types \(http://softwaretestingfundamentals.com/category/types/\)](http://softwaretestingfundamentals.com/category/types/)

STF

[Disclaimer \(http://softwaretestingfundamentals.com/disclaimer/\)](http://softwaretestingfundamentals.com/disclaimer/)

[Privacy Policy \(http://softwaretestingfundamentals.com/privacy-policy/\)](http://softwaretestingfundamentals.com/privacy-policy/)

[Contact \(http://softwaretestingfundamentals.com/contact/\)](http://softwaretestingfundamentals.com/contact/)

Social

[Facebook Page \(https://www.facebook.com/softwareTestingFundamentals/\)](https://www.facebook.com/softwareTestingFundamentals/)

[Facebook Group \(https://www.facebook.com/groups/259049327448031/\)](https://www.facebook.com/groups/259049327448031/)

[Twitter \(https://twitter.com/stestingf\)](https://twitter.com/stestingf)



Ponder

Weinberg's Second Law: If builders built buildings the way programmers wrote programs, then the first woodpecker that came along would have destroyed civilization.

Copyright - 2018 STF | Theme by Colorlib (<http://colorlib.com/wp/>) Powered by WordPress (<http://wordpress.org/>)