

V&V - Veridation or valification?

Odd NORDLAND
SINTEF Telecom and Informatics*
NO-7465 Trondheim, Norway

Abstract

There is a lot of confusion about the terms verification and validation, as is revealed by a look at the definitions used in various standards. In particular, the aspect of requirements' validation is consistently overlooked. The consequences of this are illustrated and new, clearer definitions are proposed and justified. The Euro-Interlocking Verification and Validation Plan is presented as an example.

Keywords

Definitions, Validation, Verification, Euro-Interlocking

Introduction

The title of this paper is not a misprint, it's a provocation! The fact is that there is a lot of confusion about the difference between verification and validation. Verification and validation (V&V) are activities that are performed in order to demonstrate that a development process will result or has resulted in a product that has the intended attributes. V&V is something everybody has a plan for, but nobody seems to understand the terms fully, so the plans have substantial shortcomings. This can lead to expensive blunders in a development project.

Standard definitions

Many standards contain definitions of the terms verification and validation, and comparing the various definitions reveals that what one standard calls verification is often called validation in another, and vice versa. Let's look

at some examples. We start with the definitions given in two related standards, the CENELEC railway application standards EN 50126 and EN 50128 (ref. [1] and [2]):

EN 50126:

Validation = "*Confirmation by examination and provision of objective evidence that the particular requirements for a specific intended use have been fulfilled*"

Verification = "*Confirmation by examination and provision of objective evidence that the specified requirements have been fulfilled*"

EN 50128:

Validation = "*activity of demonstration, by test and analysis, that the product meets in all respects its specified requirements*"

Verification = "*activity of determination, by analysis or test, that the output of each phase of the life-cycle fulfils the requirements of the previous phase*"

Now a definition is really just a special case of a specification, and like any good specification it should only say "what", but not "how" nor "when", "why" etc. So in order to extract the actual contents of the above definitions, we remove the superfluous frills, which makes the same definitions become:

* SINTEF has been reorganised since publication of this article, and the correct affiliation is now (2006) SINTEF Information and Communication Technology

EN 50126:

Validation = "*Confirmation ... that the ... requirements for a specific ... use have been fulfilled*"

Verification = "*Confirmation ... that the specified requirements have been fulfilled*"

EN 50128:

Validation = "*...demonstration ... that the product meets ... its specified requirements*"

Verification = "*...determination ... that the output of each phase ... fulfils the requirements ...*"

Now here we see that the EN 50126 definitions of verification and validation are effectively the same. It should be noted that the definitions in EN 50126 are identical to the definitions in IEC 61508, ref. [3].

More interestingly, the definition of verification in EN 50126 is essentially the same as the definition of validation in EN 50128! These two standards have both been adopted and are applicable, so the confusion is "official".

Meine van der Meulen (ref. [4]) has collected definitions from a number of "authoritative" sources, including many international standards (albeit neither of the above quoted CENELEC standards), and lists seven different definitions for validation and eight different definitions for verification from a total of eleven different sources. In addition to definitions for verification and validation separately, he also quotes two sources that have a definition for "Verification and validation" as a combined process. So what is the difference between the two terms? In order to answer that question, we must look at the "normal" English usage of the words.

Back to basics

Webster's Dictionary (American English), ref. [5], lists five different definitions of the word verification, depending on context, and a similar number for validation. The situation is

not much different with the Oxford Dictionary (British English), ref. [6], or the Macquarie Dictionary (Australasian English), ref. [7]. So to understand the terms, we must take a look at their origins as well as the contexts in which we will be using them.

Verification

Verification comes from Latin "veritas", which means truth, and is basically the activity of showing that something is true. In our context, that something is a claim that requirements are fulfilled, so we can use the following "definition":

"Verification is the activity of providing objective evidence that requirements truly are fulfilled."

We will see in the following that verification is an activity that contributes to validation, which at least partly explains the confusion.

Validation

Validation comes from valid, which comes from the Latin "validus" meaning strong or influential. In modern usage, the word has two basic meanings: a valid bus ticket, for example, is one that is applicable to a particular journey whilst a valid argument is one that is correct and therefore generally accepted as "true". Since validation is the act of showing that something is valid, we end up with two different "definitions", depending on what we are validating. In order to keep the difference clear, we shall use the expressions "product validation" and "requirements' validation".

Product validation

Product validation is validation in the first sense, i.e. showing that a product is applicable for some intended use. The word "intended" is important here, because the intended use is not necessarily identical with the specified use!

Nevertheless, on the assumption that the product's specification does indeed correctly reflect its intended use, showing that the product is suitable for its specified use can be

done by demonstrating that all its requirements are fulfilled.

And this is where the confusion comes in. Demonstrating that the requirements are fulfilled is, as we have just seen, verification of the requirements. So verifying that all the requirements are fulfilled is a validation of the product!

It is important to note the word "all": product validation is equivalent to verification that all requirements are fulfilled. This, however, was under the assumption that the requirements did indeed correctly reflect the intended use.

Requirements' validation

Requirements' validation is validation in the second sense, i.e. showing that the requirements are correct.

During the various phases of a development life cycle, in each phase (except the first one) there will be requirements coming from the previous phase. These requirements will be refined in the next phase, so it is usually considered to be sufficient to show that the refined requirements of one phase can be mapped back to the requirements of the previous one. However, this alone is not sufficient.

Firstly, the fact that all requirements at one phase can be mapped back to the previous phase does not necessarily mean that the converse is true. The refinement process will result in distributing the requirements amongst different modules (or units or whatever you prefer to call them) that together are supposed to fulfil the requirements coming from the previous phase. If this distribution is not done correctly, some of the requirements may get lost. On the other hand, the distribution of requirements amongst modules will result in additional interfacing requirements, which cannot be mapped to the previous phase. They must also be shown to be correct. This means demonstrating that they don't conflict with the other requirements, and that they will function in the intended way.

For the first phase of a development process we don't have a previous one to refer to, so how can

the top level requirements be validated? The simplest answer is "What the customer asked for is correct." This is usually wrong! Firstly, the customer does not normally know the implications of his wishes. Does he have requirements that contradict each other? Do they comply with legal prescriptions? Are there implicit requirements that haven't been stated? Answering these questions is the real content of requirements' validation, and it's usually overlooked. That can be expensive.

Invalid requirements can be verified: as stated above, verification is a process of demonstrating that requirements truly are fulfilled. It doesn't say anything about whether or not the requirements were correct. So the fact that requirements have been verified should be taken with caution. A few examples from real life illustrate this.

1. A homing-in torpedo system was required to destroy itself if it changed direction by 180°, because then it was assumed that it would be pointing back at the vessel that fired it. This was verified during an exercise, when the torpedo got stuck in the torpedo tube. Three hours later, when the ship turned to go home, the torpedo exploded. The self-destruction requirement was impressively verified, but evidently not quite correct. The new torpedo line was immediately decommissioned.
2. The first version of the flight software of a civil aircraft was required to keep the plane flying in the air. At the end of the first test flight, the pilot had to switch off the board computer and land manually, because the flight software refused to let him come down. The requirement was verified, but obviously not correct. The software had to be substantially modified.
3. A rail traffic control system was ordered that was not compliant with national law. The developer built it as specified, thinking it was for export purposes. It wasn't, and the customer was not allowed to use it. He had to order and pay for a complete redesign. He never ordered anything else.

In all the above cases, the requirements were all verified, whereby the products were apparently "validated". But because the top-level requirements had not been validated, expensive retrospective modifications were needed.

Requirements' validation activities

The above examples show that it is important to validate requirements before they are implemented, i.e. before there is any possibility of verifying them. Requirements' validation consists of demonstrating the three C's: Correctness, Completeness and Consistency. This applies to all phases of the development process, not just the first one.

Correctness: Demonstrating that the requirements are correct includes showing that they comply with laws, rules and regulations, so the first thing to do is to get your competitors' lawyers to find out how many laws you're breaking!

There will also be requirements derived from management decisions or policy. Such requirements will not necessarily have a technical justification, so the management decisions or policies must be documented. Then explicit confirmation by the relevant management that the decisions or policies apply will be evidence that the corresponding requirements are correct.

Then you can start analysing the requirements to see if they will successfully implement the intended attributes of the system. Functionality, i.e. the customer's wishes, is one attribute. The others are reliability, availability, maintainability and safety ("RAMS").

Demonstrating that the requirements, when implemented, will successfully result in the intended attributes will often be an iterative process, because at the start of a development the requirements will be fairly coarse. As development proceeds, the requirements will become more detailed, and the details must be examined to see if the specified attributes correspond to the intended attributes.

If the top-level requirements have been shown to be correct, then the subsequent refinements

will also be correct when it can be shown that they reflect the contents of the previous stage.

In addition to the various refinements of top-level requirements, interfacing requirements must also be shown to be correct. This will involve showing that the specified interface requirements will result in interfaces that will work together.

Completeness: Completeness is usually assumed when the requirements of one phase can all be mapped to the previous phase. This will not normally be the case, because there will be additional interfacing requirements that are not derived directly from the previous phase's requirements. Completeness for interfacing requirements means that each side of every interface is specified and that the interfaces correspond to each other.

At the top level you will have interfaces to the outside world. Your competitors' lawyers have already told you which legal requirements you'd overlooked, so you should be reasonably complete there. But you still need to check that you haven't missed out any functional or RAMS requirements.

Consistency: Many a modular system has been impossible to integrate because the individual modules had conflicting requirements. Demanding that all subsystems shall be artificially delayed to a uniform execution time may avoid synchronisation problems, but it can make achieving specified response times impossible. It is important to realise that requirements in completely different sub-systems can contradict each other, so it is not sufficient to check that the requirements of a given sub-system do not contain contradictions. Consistency must be shown across sub-system boundaries.

V&V Plans

Most V&V plans refer to applicable standards, and because the standards have such inaccurate and inconsistent definitions, the plans are equally inaccurate and inconsistent. In particular, the aspect of requirements' validation is usually overlooked. It is not mentioned in the

standards, because they tend to assume that what the customer wants is correct, so if the customer has approved a specification the specification will be correct. As we have seen, this is not necessarily the case.

It is not easy to substitute "official" definitions that are given in already adopted standards by new definitions, particularly if a project has decided to adopt the definitions from a given standard. But with the vast number of different and conflicting definitions in the standards, adopting the definitions of any one standard will automatically result in a conflict with another standard.

The solution is to show that the definitions that you use cover the contents of both verification and validation in any standard along with the aspect of requirements' validation that the standards overlook. This approach is adopted in the Euro-Interlocking Verification and Validation Plan, ref. [8], for example. Here, the definitions of verification and validation given in the pre-standard prEN 50129 (ref. [9]) had been adopted by the project, so the plan had to show that the V&V activities in the plan were not in conflict with the definitions in the pre-standard.

The Euro-Interlocking V&V Plan

Ref. [8] is a generic verification and validation plan that can be used as a starting point for more specific plans. The plan relates the V&V activities to the various phases of the development model that is used in the CENELEC railway application standards, ref's [1], [2] and [9].

The Euro-Interlocking project focuses on the first four phases of the development model, viz. Concept, System definition and application conditions, Risk analysis, System requirements, and the "final" phases System acceptance, Operation and maintenance, Decommissioning and disposal. These are the phases where generic aspects of an interlocking system are relevant.

For an actual implementation, the V&V activities will depend on project specific details

of the implementation, such as the technology that is used, the architecture of the system etc. Such activities cannot be specified in a generic plan, so a project specific supplement will be required.

In addition to identifying the V&V activities to be performed, ref. [8] provides a schedule for performing the activities, based on the phases of the development model as defined in ref. [1].

The processes and procedures that should be applied will depend on the details of a specific project and must be specified in the project specific supplement to ref. [8]. Ref. [8] contains generic descriptions that can be used as a foundation for specific descriptions.

Requirements on the qualification and independence of V&V personnel are also identified in the plan. Since the Euro-Interlocking project aims at systems with a high safety integrity level, the personnel requirements are correspondingly strict. However, if a complex system has both high and low safety integrity level components, the highest safety integrity level must be fulfilled, so the strict requirements will be applicable.

The results of the V&V activities must be documented in corresponding reports, and the plan identifies the requirements for such reports.

The plan was developed for the context of a railway interlocking system, so it demonstrates compliance with the CENELEC railway application standards, ref's [1], [2] and [9]. This is done by means of tables identifying the V&V related clauses in the CENELEC railway application standards and the sections of the plan that address those clauses.

Conclusion

The most important part of validation, namely requirements' validation, is usually overlooked in V&V plans. This is partly caused by the fact that the word validation is usually used alone, so that the different contexts get lost. In order to avoid this, the following terms and definitions are recommended:

Requirements' validation is the activity of providing objective evidence that the requirements are correct, complete and consistent.

Verification is the activity of providing objective evidence that requirements truly are fulfilled.

Product validation is the activity of providing objective evidence that the product is suitable for its intended use.

Invalid requirements can usually be verified after they've been implemented, but that doesn't make them correct. So requirements' validation must be the primary validation activity. Verification will be performed during implementation and integration, and product validation then boils down to showing that all the requirements have been validated and verified. If this is not stated clearly and comprehensibly in the applicable V&V plan, the result can be expensive modification and retrofit exercises.

The Euro-Interlocking Verification and Validation Plan is an example of a plan that pays due attention to the aspect of requirements' validation. It is a plan that is intended to be used for a railway application, but the concepts that are used are sufficiently generic, so that a generalisation to other application areas is not a major task.

References

- [1] EN 50126:1999; Railway Applications - The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS); CENELEC; 1999
- [2] EN 50128:2001; Railway Applications - Software for railway control and protection systems; CENELEC; 2001
- [3] IEC 61508-4; Functional safety of electrical/electronic/programmable electronic safety-related systems, Part 4 Definitions and abbreviations; IEC; 1998

- [4] M.J.P. van der Meulen: Definitions for Hardware/Software Reliability Engineers; Springer Verlag; 2000; ISBN 1852331755
- [5] Webster's Third New International Dictionary of the English Language; G.&C. Merriam Co.; 1976; ISBN 0-87779-103-1
- [6] Oxford Advanced Learner's Dictionary of Current English, 6th edition; Oxford University Press; 2000; ISBN 0 19 431 510 X
- [7] Macquarie Concise Dictionary, 3rd edition; The Macquarie Library Pty Ltd; 1998; ISBN 0 949757 96 9
- [8] O. Nordland: Euro-Interlocking Verification and Validation Plan; Euro-Interlocking; 2002
- [9] prEN 50129:2000; Railway applications - Safety related electronic systems for signalling; CENELEC; 2000