

Please note! This is a self-archived version of the original article.

Huom! Tämä on rinnakkaistalenne.

To cite this Article / Käytä viittauksessa alkuperäistä lähdettä:

Perttula, A. & Kukkamäki, J. (2020) Enabling Rapid Product Development through Improved Verification and Validation Processes. *Technology Innovation Management Review*, 2020:3, s. 24 - 35.

URL: <http://doi.org/10.22215/timreview/1334>

# Enabling rapid product development through improved verification and validation processes

Antti Perttula and Joni Kukkamäki

*“Requirements analysis is critical to the success or failure of a systems or software project.”*

Abran Moore and Bourque Dupuis,  
Editors of The Guide to the Software Engineering Body of Knowledge

Fierce competition in consumer electronics market has raised a lot of challenges for product development. Products now must enter to the market as fast as possible. The verification and validation (V&V) process is normally the most resource-demanding activity in product development (PD), and thus also has huge potential for improvement. The V&V process is traditionally executed near the end of the development process, and is one of the most critical activities because it identifies design errors. Error correction for a nearly complete product is often difficult and therefore can cause unexpected delays in product delivery. Performing V&V activities in early phases of the PD process and utilising V&V methods other than testing has proven to be a good approach to reduce risks associated with taking a long time to reach the market. However, V&V can only be carried out when verifiable requirements exist. This paper focuses on the importance of moving V&V activities to each phase of PD by defining the requirements for components and modules. In addition, we explore how some of product-level requirements can be verified before a product has been integrated, and explain the difference between definitions of verification and validation requirements. Finally, we present the idea of changing the focus of verification activities from being set-based in the early phase of development, to being point-based when the product is close to completion.

## 1. Introduction

Companies are facing increasing demands to improve their product development (PD) efficiency because of tough competition in the consumer product market. Newer and more innovative products and services must be brought to market earlier than competitors, which means a need for continuously shorter design cycles. In their studies, McGrath (1998) and Mahadevan (2009) claimed that bringing products to market earlier than competitors brings many benefits, including larger sales volumes and longer sales period. Current changes in the business environment, with a fast time to market and demands to decrease PD costs, have increased the importance of having an efficient product creation processes.

Verification and validation (V&V) are usually the most resource-demanding activities in PD, which makes it possible to find remarkable resource savings in them. The traditional way of carrying out PD has placed V&V as the last activity before product delivery to customers. According to Reinertsen (2009), this timing plan is

problematic because, typically, V&V are used to detect errors in product designs, which makes it a critical activity. Correcting errors can be a resource-demanding and time-consuming task, hence, it can lead to delayed delivery or delivery of products with low quality (Kelkar, 2012).

Several examples exist of PD projects that have failed because V&V was not carried out properly, for example, Motorola's Iridium Communication System (Millard, 2017) and the Hubble Space Telescope (Redd, 2017). Both projects caused the development organisation to incur billion dollar-level extra costs because V&V was done too late. The Iridium System development was performed according to plan and passed most verification tests. However, complete validation was not done until the system was finished. This meant that customers didn't adopt the system into actual use. In other words, validation failed, which led to the company's eventual bankruptcy (McIntyre, 2009). In the Hubble Space Telescope project, some requirements related with the primary mirror's surface were not met during the development phase, which caused blurred

## Enabling rapid product development through improved verification and validation processes *Antti Perttula and Joni Kukkamäki*

pictures (Goodwin, 1993). Later, image quality was improved by installing corrective optics to the telescope in space with estimated additional cost of one billion US dollars (Cohen, 2009). This paper describes both the purpose of V&V and also how to move their implementation to all phases of the development process, as a way to improve PD efficiency. In short, it means that requirements must be created for all the phases of development where V&V is to be carried out.

### 2. Definitions of Verification and Validation

Verification has been widely understood as a method to prove a product's compliance with specifications. These are not only user requirements for the finished product, but also requirements for components and subassemblies. Mooz, Forsberg and Cotterman (2003) noted that it is not well-known that verification can actually be determined, in addition to testing, inspection, demonstration, and analysis.

The aim of validation is to prove that users are satisfied with the end product. Validation answers the following question: "Is this product behaving as the customer anticipates?" Validation involves the evaluation of customer requirements against their needs and expectations in the most representative environment achievable. According to Stevens, Brook, Jackson, and Arnold (2000), "validation" is sometimes defined as an end-to-end verification process that aims to show that the whole system meets its requirements under operational conditions. The Institute of Electrical and Electronic Engineers Computer Society (2017) defined V&V as "processes that are used to determine whether the development products of a given activity conform with the requirements of that activity, and whether the product satisfies its intended use and user needs".

### 2.1 Product Requirements

Product requirements are the foundation of the whole PD process. Defining the product-level requirements is basically a straightforward task. First, we ask the customer or anticipate their wants and needs. Second, we transfer the customer's input into exact engineering requirements. Finally, we check with customers that the requirements are right. In addition to these, there are many requirements for products defined by legislation and type approval regulations. When the requirements are clear, we can then attempt to make a product that meets these requirements.

The major problem in PD is the time delay between the definition of requirements and the product's launch to market. During this time period, a customer's needs and wants may change. When the product is finally launched, however, it may happen that customers do not want the product anymore, signifying that product validation has failed. Some system development models, such as Extreme Programming (XP), have tried to minimise the risks associated with validation through the continuous task of customer input and requirements definition. However, according to Paulk (2001), these kinds of agile development models cannot be applied to all kinds of PD. Similarly, Cobb (2019) stated that it may be difficult to apply an agile method, such as Scrum, to large and complex projects.

As normally practiced, product verification can start after the requirements are defined and a product is integrated to market. In practice, this means a lot of work in the late development phase, when errors found at this point can cause unexpected delays in the product's launch. However, many of product-level requirements rather guide the component and module development than influence the final PD. Hence,

**Table 1.** Some of the product level requirement can be verified at earlier development phases

Product Requirement	<i>Level, where verification should be done and the method used</i>			
	Product level	Sub-system level	Component level	Material level
1) Lead-free				<i>Do not use lead in any materials</i>
2) One million key pressings		<i>Verify that a keyboard is functional after a million pressings</i>	<i>Select the right components</i>	
3) TA	<i>TA testing for the product as required by the authorities</i>	<i>TA testing for modules</i>		

## Enabling rapid product development through improved verification and validation processes *Antti Perttula and Joni Kukkamäki*

several product-level requirements can be verified at the component and module level. The following requirements shown in Table 1, for example, can be verified partly or completely before a product is integrated into market.

### 2.2. Technology Requirements

Because of the continuous demand to reach for a shorter time to market, sufficient time often cannot be spent on thorough verification at the product level. Product level requirements must therefore be verified as early as possible, as discussed in the previous chapter. If we make modular products and want to move even more of the verification process away from the product level, then we must be able to define the technology requirements, including components and module requirements. In this case, technology means any kind of technical solution for a complete product.

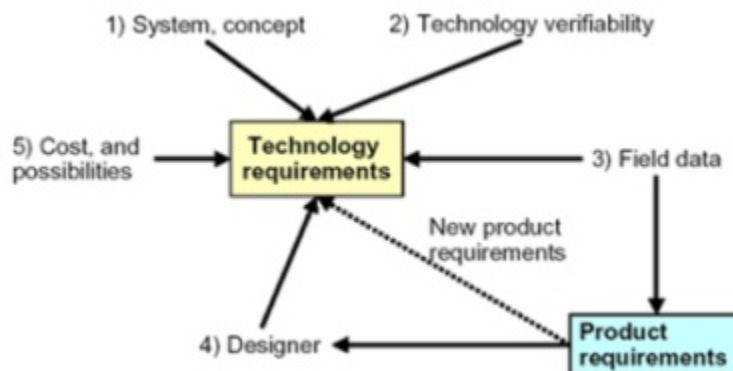
The definition of technology requirements is normally more complex than the definition of product requirements because customers are often not able to give input to them. In other words, technologies, modules, and components in a product are not always visible to the customers. In fact, there is a definite difference between product and technology requirements and the linkage between them is rarely straightforward. However, as usual, technology (including components and module) requirements are developed from product and customer requirements, which is a sensible approach. The importance of the designer's competence, along with existing technological possibilities, design rules, and guidelines when defining requirements, must not be forgotten.

In the simplest cases, subsystem and component

requirements can be copied from product requirements. For example, if a product must function normally in temperatures between 20°C and +40°C, then we can simply define the same requirements for its components and modules. However, there may be heat sources inside the product, which can increase the internal temperature. In some areas, the temperature can reach +80°C, which may require increasing the upper-temperature limit of the components accordingly. This may, however, be expensive to produce, while some technologies may not be functional at such high temperatures. In such a case, it would be beneficial to create a temperature simulation model for the product before the physical prototype is made, so that temperature-sensitive components can be placed in cooler locations.

The definition of technology requirements is often difficult because the linkage to product requirements is not always solid. For this reason, the designer plays an important role based on their competences, together with the existing technological possibilities. Requirements for the same module may also vary depending on the designer's competences and the existing technological possibilities. Combined, these can strongly affect the probability that the module will function properly under conditions of product exposure. Because of this, the technology requirements cannot alone guarantee that the module or technology will survive when being integrated into a product.

We have noticed that definitions of technology and subsystem requirements often include many different, and sometimes, unmeasurable inputs. Figure 1 describes the five main inputs to definitions of technology and subsystem requirements.



**Figure 1.** Inputs to definitions of technology requirements, (Perttula, 2007).

## Enabling rapid product development through improved verification and validation processes *Antti Perttula and Joni Kukkamäki*

Inputs to define technology requirements include the following:

- 1) The system and concept that is to be built roughly defines the level and set of requirements.
- 2) Like any other requirements, technology requirements must also be verifiable with the resources, equipment, and cost that producers can afford.
- 3) Field data is an important source of input both for technology requirements and product requirements. If we notice that similar technology is failing in the field, we must input this information into new requirements. Also, if a technology is used in other PD projects and seems to be risky, this should be included as input as well.
- 4) The designer's competence and possibilities are probably the most important inputs.
- 5) The last inputs are the costs and possibilities of a technology, which must be in line with each other.

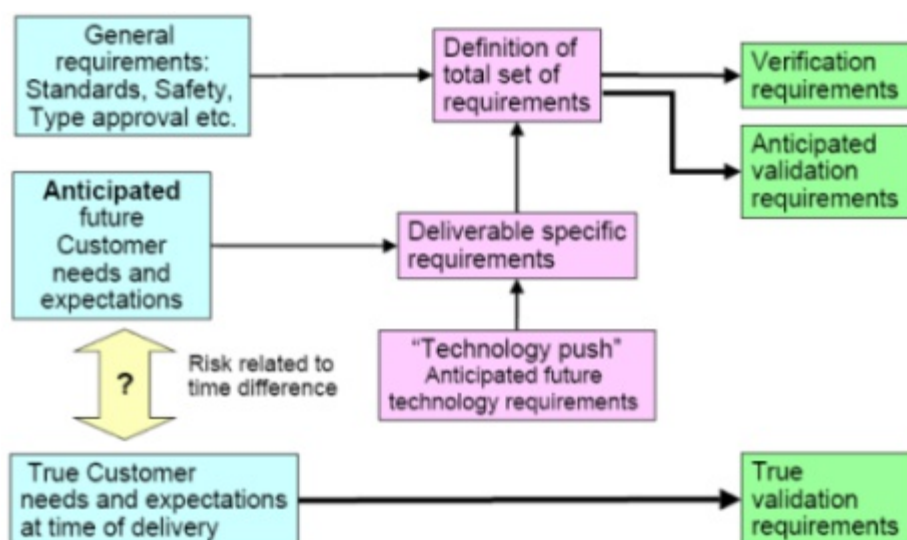
Although a solid link between the product and the technology requirements may not be present, information about developing new product requirements could come directly from defining a technology's requirements. Although defining a technology's requirements is not an easy task, it is crucial that it is completed in order to be able to carry out verification before the integrated product phase.

### 2.3. Definition of verification and validation requirements

Figure 2 describes the new process of defining V&V requirements. Inputs (on the left-hand side of the picture) are general requirements, such as standards and legislation, as well as anticipated and true future customer needs and expectations. The deliverable specific requirements inputs are anticipated future customer requirements, along with anticipated future technology requirements. The deliverable specific requirements together with general requirements form the total set of requirements. From this set, the true verification and anticipated validation requirements will be developed. Validation requirements cannot be accurately defined in the early stage because customer needs and expectations are not yet known. This means that there is always a risk in customer validation, with the level of risk being strongly correlated to PD time. Because of this, PD time should be decreased by all possible means.

### 3. Modified V-model in Platform-Based Product Development

The V-model created by Harold Mooz and Kevin Forsberg is widely used in PD projects to highlight V&V activities (Mooz & Forsberg, 1991). This model is used in both software and hardware development. Our ambition here is to help move relevant product level verification activities to earlier development phases in order to decrease the workload at the product level. This kind of approach can be called "incremental verification". We illustrate this method below using the idea of the V-



**Figure 2.** Defining the verification and validation requirements (Perttula, 2007).

## Enabling rapid product development through improved verification and validation processes *Antti Perttula and Joni Kukkamäki*

model.

The idea of incremental verification is to decompose the product requirements to smaller ‘subrequirements’, develop ‘subitems’, and verify them against these subrequirements separately. Suitable requirements for incremental verification often cover broad areas, such as those relating to product usability and type approval. By developing and verifying these subitems at an early stage, only some of the verification efforts will be needed at the product phase, which also means fewer risks. Typically, subitems are simulation models, modules, and components.

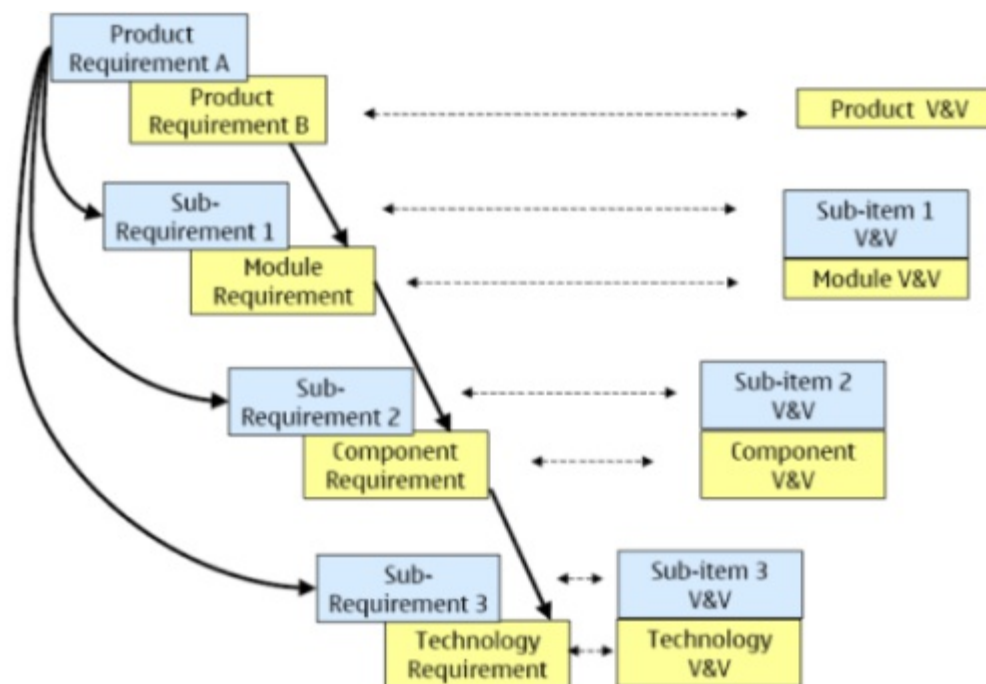
On the left-hand side of the V-model, product requirement A is decomposed into three sub-requirements. Based on these sub-requirements, three sub-items are developed and then verified separately. When these verifications are completed, then the product level requirement A will be verified. Before decomposition can start, product requirement A must be understood as being composed of lower-level requirements.

Our aim is to complete the technology, components, and module V&V before an integrated physical product is available. In practice, this means that we need to apply

other types of verification methods beyond only physical testing, as often as possible. For example, we can replace product level drop testing and thermal analysis for finished products by extensive simulations, before the physical samples are produced. By focusing on early V&V, we can save time and select the right technology for further product integration. This will reduce risks at the product phase, and ultimately improve PD efficiency.

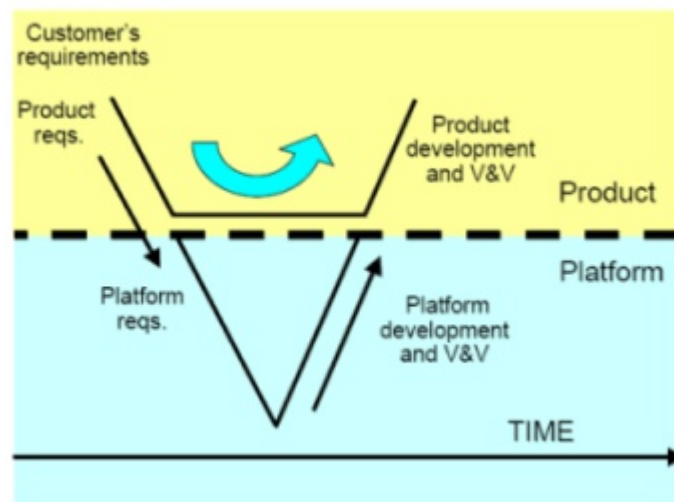
In platform-based development, products are made of modules and components, which are usually completed before product integration. The subsystems are then integrated into several future products. One of the major challenges with platform-based development is how to build platforms for future products. Figure 4 shows the traditional way of utilising platforms in PD.

Customer requirements flow through the product programme to platform development. Platforms are developed based on these requirements and are delivered to the product programme that makes use of them. This approach makes sense when there is no serious time pressure during the PD phase, and it is possible to wait until the platform is complete for product integration. PD projects in such cases must wait



**Figure 3.** Incremental V&V of some product-level requirements.

## Enabling rapid product development through improved verification and validation processes *Antti Perttula and Joni Kukkamäki*



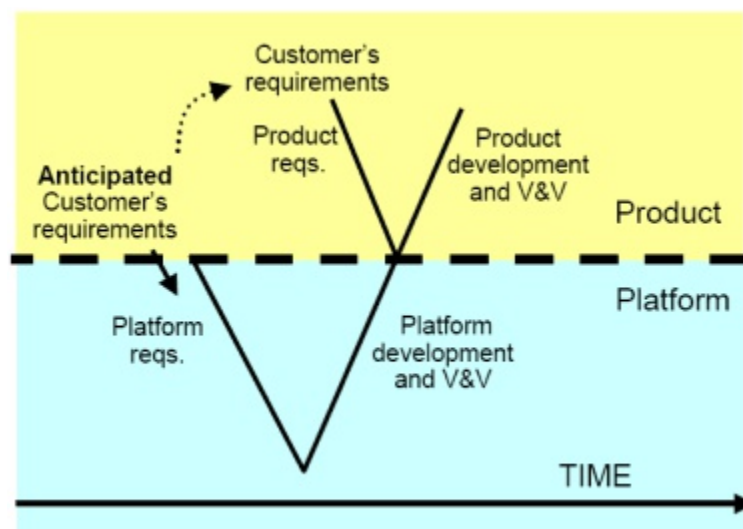
**Figure 4.** A typical way of utilising platforms in product development.

for platform development to be completed. The thick curved arrow in Figure 4 displays the waiting period. This can be reduced by starting platform development before actual PD. We have further elaborated on the idea of the V-model to describe platform-based PD when there is a time constraint during the PD phase. Figure 5 describes platform development taking place before the PD phase.

At the beginning of the platform development phase, customers' requirements for the product are not necessarily yet known. The technology development organisation must first anticipate these missing

requirements, develop the platform, and finally both verify and validate the platform before forwarding it to PD. The PD process shown in Figure 4 is straightforward; product requirements are developed from customer requirements, then the product is developed, and finally verified and validated.

Missing customer requirements at the early development phase present new challenges for platform creating organisations. Adequate knowledge and competence are needed to anticipate future customer requirements for a product, along with readiness to adjust the requirements over the course of time. In the



**Figure 5.** A modified V-model for platform-based product development.

## Enabling rapid product development through improved verification and validation processes *Antti Perttula and Joni Kukkamäki*

normal way of developing products, customer interface is managed during the production programmes, instead of platform development period.

As shown above, the major challenge of V&V for platform-based development is a difficulty in defining the requirements for technologies, modules, and components. It is often either not possible to create the exact requirements, or they continuously change over time. Because of this, we have started to change the focus of verification from error detection with a pass/fail indication or point-based verification activity, toward defining opportunities, that is, a set-based verification (SBV) process developed by Perttula (2007). In SBV, all relevant information about a design is collected as early as possible using normal measurement and analysis methods. This data can be referred to whenever needed, as a way to check whether a design still meets the changing requirements. This eliminates the need to continuously repeat physical measuring and testing. Using this approach, we have found that remarkable time and cost savings can be created.

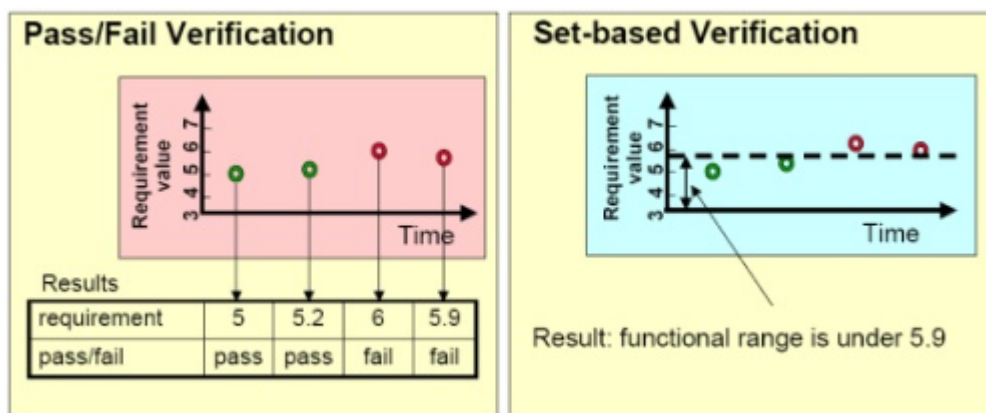
The left-hand side of Figure 6 displays the traditional, point-based way of carrying out pass/fail verification. The design in question meets the requirement with values of 5 and 5.2, but fails when the requirement is 5.9 or 6. Without repeating the verification, it is not known whether the design is working between 5.2 and 5.9. In set-based design (on the right-hand side in Figure 6), we did not verify the design against a current fixed requirement, but rather employed a ‘test-to-fail’ approach in order to understand what range the design could tolerate. If the initial requirement changed, we

would know instantly without new verification whether the design could meet the new requirement. By utilizing a set-based approach to verification, it became possible to remarkably improve product development speed in certain circumstances, with frequently changing requirements.

### 4. Verification & Validation (V&V) in Agile Projects

Constant changes in product requirements might cause several issues in PD projects, while proper reactions by those in development can lead to better products. Because Agile methods (for example, Scrum) are created specifically to take note of changes, they fit very well into development projects that involve a high degree of uncertainty. The Agile manifesto (Beck et al., 2001) pointed out that Agile projects should value rapid responses to changes instead of following a pre-set plan. Furthermore, the following is included along with the 12 principles of Agile software: “Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage” (Beck et al., 2001).

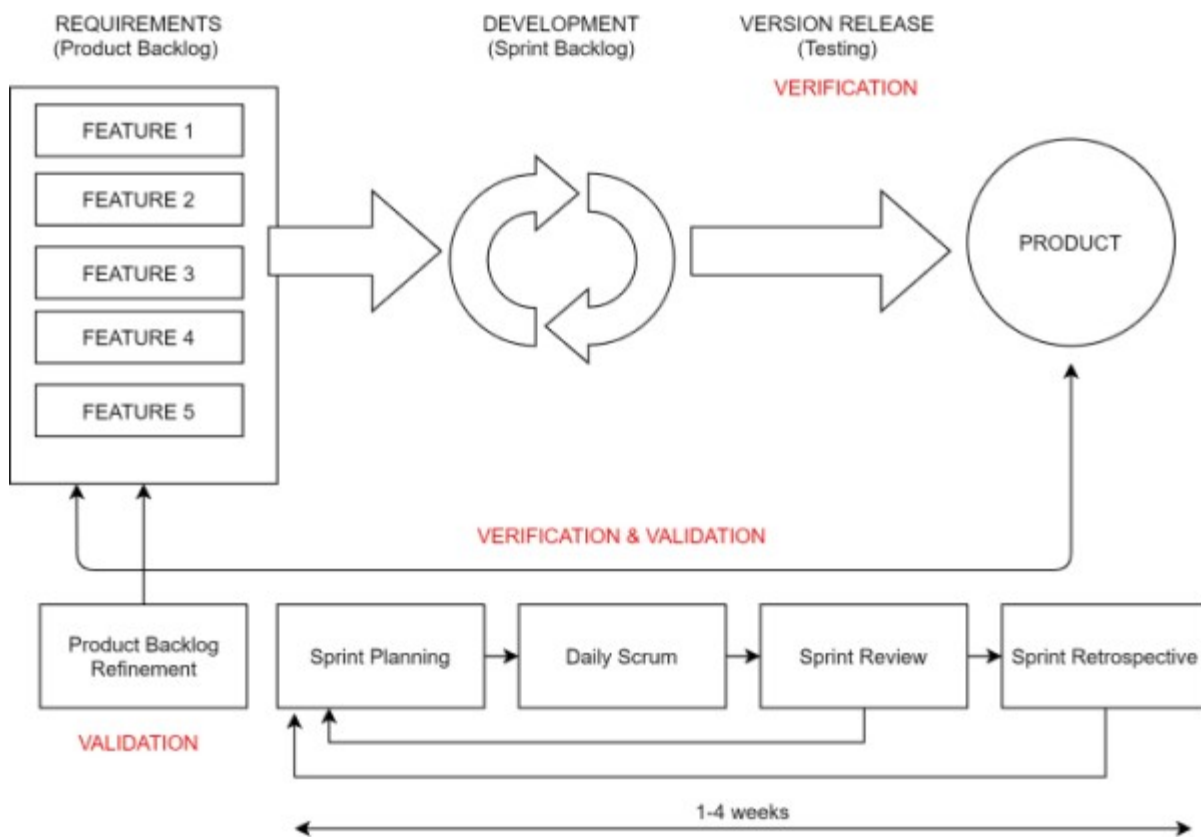
The Agile manifesto is written from the viewpoint of software development. Nevertheless, it includes principles that can be implemented for any kind of development process. It is good to understand and take note of the differences between software development and, for example, physical PD. Software development is immaterial or informational, data-based, while developing physical products involves materials, manufacturing, and logistics.



**Figure 6.** The Y-axis is a functional value, while the horizontal, dashed line is the maximum of this functional value that meets the requirement. Set-Based verification estimates design possibilities. (Camarda et al., 2019).



## Enabling rapid product development through improved verification and validation processes *Antti Perttula and Joni Kukkamäki*



**Figure 7.** Verification and validation processes implemented in each iteration of Agile development.

Agile methods are usually artefact-based and iterative processes. Everything starts with features, which are usually described as “user stories”. These describe who the users are, what they need and how they use a product, thus giving a strong user-centric way to developing products. According to Schön et al. (2017), when a company has compiled a comprehensive set user stories, created together with users, then a validation process can be applied to these stories.

Agile methods are iterative. Schwaber and Sutherland (2017) stated that in Scrum, one iteration is called a “sprint”. In one sprint, multiskilled teams take these features to be developed in a short, 1–4-week period. The aim is to produce deliverables that are fully ready to be implemented in a product, including user testing. At the beginning of a sprint, the team splits the features into small tasks that need to be done to achieve the goals of the sprint, in order to meet the feature requirements. This way, verification can be implemented in testing processes and done iteratively.

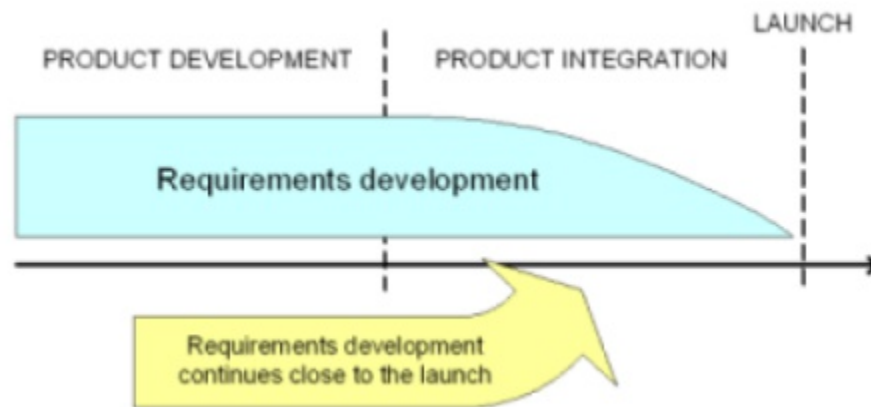
In an Agile project, the development team focuses only on the ongoing activities and features, so that changes can be done more easily at later stages while they are still open for changes. V&V processes can be implemented for each iteration, instead of at the end of a project, as illustrated in Figure 7.

Agile methods used for PD are based on short user stories or features, which are then developed and integrated into the product during 1-4 week long sprints. Each sprint or iteration includes planning, design, coding, unit testing (verification) and acceptance testing (validation).

### 5. Changing the Focus of Verification in Product Development

Sometimes, we may need to change the focus of a verification process between point-based and set-based, according to the phase of PD. It is important to ask what should be done to avoid repeating the verification process when the requirements are changing. The

## Enabling rapid product development through improved verification and validation processes *Antti Perttula and Joni Kukkamäki*



**Figure 8.** Requirements development continues close to the launch of the final product.

classical definitions of V&V do not give direct answers to this challenge. If we collect more information from the product than just enough to make a pass/fail justification, we may then be able to utilise it when a requirement has changed. On the other hand, when requirements are stable, the point-based verification is often enough, and is more economical to conduct than a set-based approach.

Some of the necessary components and modules for the product are usually acquired from external suppliers. These components and modules are the end products of their manufacturers. For example, a smartphone is a subsystem of a larger communication network. If we can divide the PD (end-product, module, or component) into technology development and product integration, then changing the focus of the verification process can be advantageous. In such situations, we can utilise both types of verification. At the beginning of the process, the requirements may not necessarily be well known or can even be erroneous. Thus, they will need to be clarified and changed as time passes. Figure 8 describes how the development of requirements continues almost to the end of the PD. In some development models, such as XP, the defined requirements actually continue almost with the same intensity to the end of PD.

Figure 9 shows how the focus of verification should change during different phases of PD. The requirements development occurs often at the early design phases. For this, the SBV approach is almost the only method suitable, whereas pass/fail verification plays a significant role in the later design phases.

At the later development phases, closer to product

launch, the requirements are typically stable, and defined by customers or based on legislation. These requirements are often related to type approvals or customer acceptance. Product verification against these requirements is often of the pass/fail type, while the most important goal is fast execution. However, because of possible variations in both production and components, during the production phase it is usually beneficial to carry out an additional set-based type of verification to see how close to the limit a design is, as shown in the right-hand side of Figure 10.

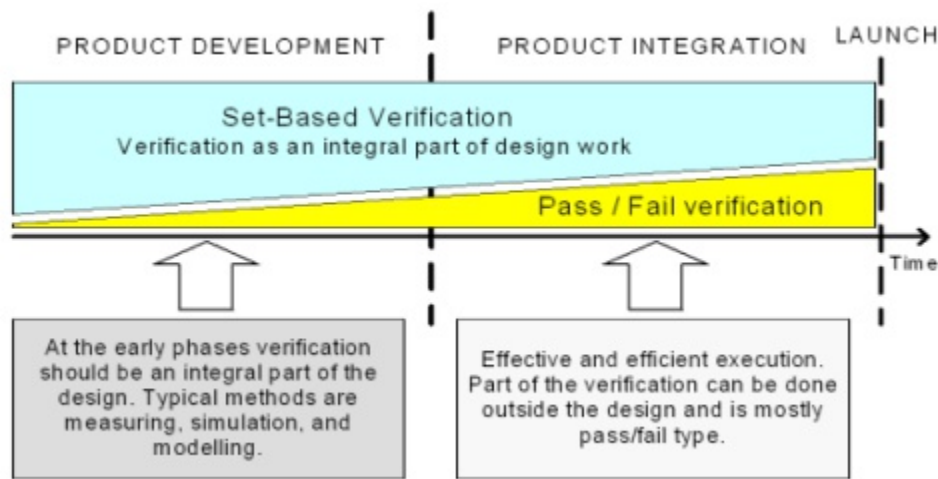
On the left-hand side, pass/fail verification only shows that in weeks 31 and 37 some samples did not meet the acceptance criteria, while SBV gives much more information. For example, we can see how stable the production process was and how far away samples were from the acceptance criteria limit.

### 6. Conclusion

This paper's goal was to study how improved definitions of V&V requirements can enable rapid PD, and in particular, to research what is required to distribute V&V actions over the whole PD process. The most important findings of this research are as follows. First, carrying out V&V at each phase of PD, instead of just close to product completion, can be very beneficial because error correction risks can be better managed this way.

Second, V&V activities can be carried out only when relevant requirements exist; hence, the need to create requirements for components and modules in addition to a complete product. However, the definition of these requirements is more complex than the definition of

# Enabling rapid product development through improved verification and validation processes *Antti Perttula and Joni Kukkamäki*

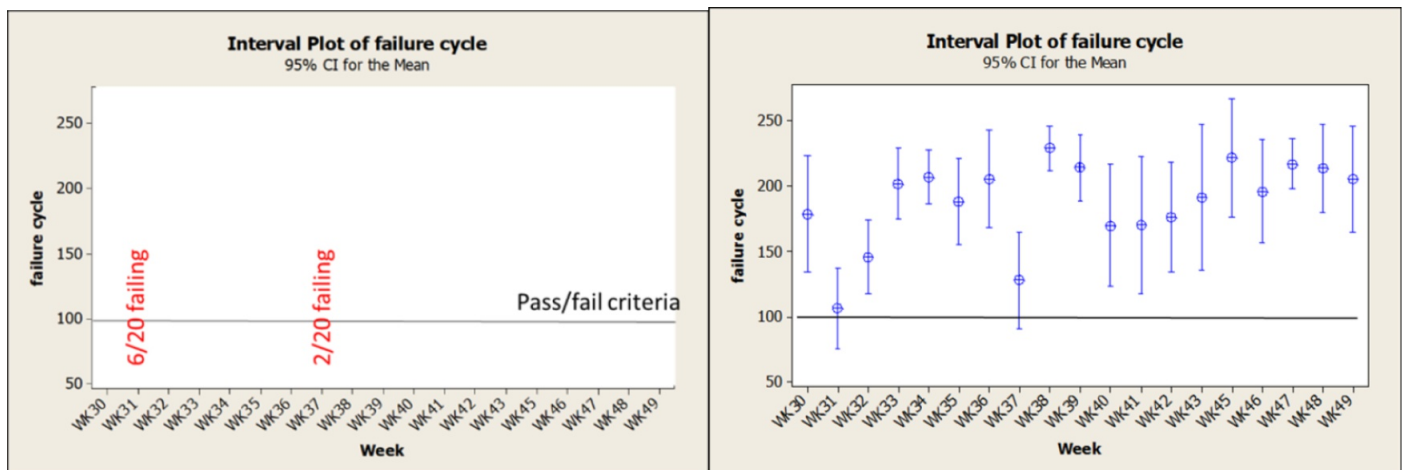


**Figure 9.** Changing the focus of the verification and validation in product development where technology development occurs separately before the product integration.

product requirements because customers often do not have visibility. Nevertheless, in some cases technology (including component and module) requirements can be developed from product and the customer requirements, or can be simply anticipated. The linkage between these requirements is usually not straightforward, and thus the designer’s role is key. Depending on the designer’s competences, existing technological possibilities, design rules, and guidelines, technology requirements may vary. Neglecting the designer’s possibilities can cause delays in introducing the latest new technology into products. Using product-level requirements as technology requirements, it is also very likely that a technology supplier may not be capable

of making a new technology meet these requirements on time, or in terms of affordable costs. In other words, the new technology may be so “fragile” that it just cannot meet the product requirements in the beginning, but instead requires further development that takes more time.

Third, PD requirements are often not fully set at the beginning because they are constantly changing, such as in Agile development projects. Thus, it is beneficial to apply SBV. When requirements are stable, the normal point-based verification that enables fast execution is the best and most economical approach. However, even in mass production, SBV can be useful. If we understand



**Figure 10.** Information received from point-based (left-hand side) and set-based verification in mass production.

## Enabling rapid product development through improved verification and validation processes *Antti Perttula and Joni Kukkamäki*

the stability of the production process, we can predict in advance whether samples are going to fail or pass in future acceptance testing. At the initial stage, carrying out SBV requires more time and effort than point-based because information about the functional areas of a design must be collected, not just pass/fail justification as a certain criterion. Nevertheless, we believe that SBV has advantages that make it preferable in certain circumstances, as highlighted in this paper.

### References

- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., et al. 2001. *Manifesto for agile software development*. <https://agilemanifesto.org/> Cited on December 20, 2019.
- Camarda, C.A., Scotti, S., Kunttu, I., Perttula, A. 2019. Rapid product development methods in practice - case studies from industrial production and technology development. Paper presented at *ISPIM Connects Ottawa*, Ottawa, Canada on 7-10 April, 2019.
- Cobb, C. 2019. *What Are the Advantages and Disadvantages of Agile and Scrum?* High Impact Project Management, Inc. <https://managedagile.com/what-are-the-advantages-and-disadvantages-of-agile-scrum/>. Cited on January 15, 2020.
- Cohen, M. 2009. *Recollections of the NASA STS-61 First Hubble Space Telescope Servicing Mission*. Company Seven. [http://www.company7.com/c7news/nasa\\_sts61.html](http://www.company7.com/c7news/nasa_sts61.html). Cited on February 14, 2020.
- Goodwin, I. 1993. Hubble Space Telescope Makers Pay for Costly Mirror Mistake. *Physics Today*, vol. 46, Issue 11, November 1, 1993.
- IEEE Computer Society. 2017. *1012-2016 - IEEE Standard for System, Software, and Hardware Verification and Validation*. 012-2016/Cor 1-2017
- Kelkar, S.A. 2012. *Software Project Management: A Concise Study*. PHI Learning Pvt. Ltd. 3rd edition. ISBN 978-812034702-1
- Mahadevan, B. 2009. *Operations Management, Theory and Practise*. Pearson Education, 2009, 4th. edition. ISBN 987-81-7758-564-9
- Millard, D. 2016. Iridium: story of a communications solution no one listened to. *New Scientist*. 3 August 2016. <https://www.newscientist.com/article/mg23130850-700-iridium-story-of-a-communications-solution-no-one-listened-to/>. Cited Feb 12, 2010.
- McGrath, M. E. 1996. *Setting the PACE in product development: a guide to product and cycle-time excellence*. Butterworth-Heinemann, 1996, revised edition. ISBN 0-7506-9789-X
- McIntyre, D. A. 2009. *The 10 Biggest Tech Failures of the Last Decade*. Time 24/7 Wall St. Thursday, May 14, 2009.
- Mooz, H., Forsberg, K. 1991. The Relationship of Systems Engineering to the Project Cycle. *Joint Conference of NCOSE and the American Society for Engineering Management*.
- Mooz, H., Forsberg, K., Cotterman, H. 2003. *Communication project management the integrated vocabulary of project management and systems engineering*. Hoboken, NJ: John Wiley & Sons Inc.

# Enabling rapid product development through improved verification and validation processes *Antti Perttula and Joni Kukkamäki*

ISBN 0-471-26924-7

- Paulk, N. C. 2001. Extreme programming from a CMM perspective. *IEEE Software*, Vol. 18, Issue 6, November-December: 19–26.
- Perttula, A. 2004. Holistic view of verification and validation in system development models. *Proceedings of CSER04*, University of Southern California, Los Angeles, USA. April 15-16, 2004.
- Perttula, A. 2007. Challenges and Improvements of Verification and Validation Activities in High Volume Electronics Product Development. *PhD Dissertation*, Tampere, Finland: Tampere University of Technology. ISBN 978-952-15-1713-6, ISSN 1459-2045
- Redd, N., T. 2017. Hubble Space Telescope: Pictures, Facts & History. Space.com. <https://www.space.com/15892-hubble-space-telescope.html>. Cited Feb 12, 2020.
- Reinertsen, D. 2009. *The Principles of Product Development Flow Second Generation Lean Product Development*. Celeritas Publishing, Redondo Beach, California, USA. ISBN 978-1-935401-00-1 ISSN 1-935401-00-9
- Schwaber, K. & Sutherland J. 2017. *The definitive guide to scrum: The rules of the game*. <https://www.scrumguides.org/> Cited on December 20, 2019.
- Schön, E-M., Thomaschewski, J. & Escalona, M J. 2017. Agile Requirements Engineering: A systematic literature review. *Computer Standards & Interfaces*, Volume 49: 79-91.
- Stevens, R. Peter Brook, P., Jackson K., Arnold, S. 2000. *Systems engineering: Coping with complexity*. Prentice Hall. ISBN: 0-13-095085-8

## About the Author

Dr. Antti Perttula is Principal Lecturer of Systems Engineering and Head of Aircraft Engineering Education at TAMK. In 2019, he was a Visiting Professor at the München University of Applied Sciences. Before academia, Perttula held several senior management positions in R&D organisations in the industry, military and civil aviation. Currently, Perttula works in rapid PD and in drone technology, including autonomous flying. He has worked in several countries in Europe, Asia and Africa.

Joni Kukkamäki acts as a Research Manager in the HAMK Smart Research Unit. He runs the Digital Solutions and Platforms team whose role is to provide technology-based research and development activities for the needs of HAMK research and industry. Focus areas for the team are software development, the Internet of Things, Game Engines and data analytics. Kukkamäki is a Bachelor of Business Administration (IT) graduate and is currently running his MSc (Computer Science) at Tampere University.

Citation: Perttula, A. and Joni Kukkamäki, J. 2020. Enabling Rapid Product Development through Improved Verification and Validation Processes. *Technology Innovation Management Review*, 10(3): 24–35. <http://doi.org/10.22215/timreview/1334>



Keywords: Verification, validation, requirements, set-based design, agile