Appendix **B**

The V-Model

Today, the Systems Engineering Vee is omnipresent in almost every systems development environment. It has an iconic status. The V-Model, as the Systems Engineering Vee is also called, emphasizes a rather natural problem-solving approach. Starting on coarse grain level partitioning the problem to manageable chunks. From the fine grain level the final solution integrates up to the initial level. On each level, one can compare the solution or part of it with the problem or the related part of it. The simplicity of this model (see Figure B.1) permits various projections and results in many interpretations.

B.1 A BRIEF HISTORY OF THE V-MODEL OR THE SYSTEMS ENGINEERING VEE

The V-Model emerged probably in the 1960s, though there seem to be no public citations available. The citations hereafter suggest that the V-Model emerged from more than one source independently. The designation of the model varies depending on the sources. Hereafter, we cite the designation as used in the referenced documents.

Model-Based System Architecture, First Edition.
Tim Weilkiens, Jesko G. Lamm, Stephan Roth, and Markus Walker.

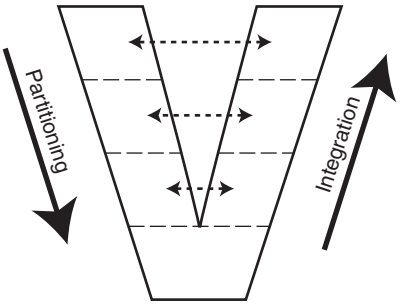

Figure B.1. A basic V-Model.

In 1979, Barry W. Boehm published a paper [15] that was built up on the Vee. He used the Vee in the context of software engineering to emphasize the importance of verification and validation. Boehm made a distinction between an upper part of the Vee for validation and a lower part of the Vee for verification and linked these processes to the related requirements and specifications, respectively. The multilevel nature of systems depicted in the Vee is not further elaborated. Boehm attributed the "V-Chart", as he named it, to personal communication from J.B. Munson, System Development Corporation in 1977.

In a systems context, the "V-Chart" was presented at the first annual conference of NCOSE in 1991 [40]. This conference was the predecessor of today's INCOSE International Symposium. Kevin Forsberg and Harold Mooz introduced the "V-Chart" to clarify the role and responsibility of system and design engineering within the cycle of a project. Other than Boehm's paper that focused on V&V processes in software engineering, Forsberg and Mooz focused on projects realizing systems. Consequently, the multilevel nature of systems is a major issue within this paper. They presented a three-dimensional "Vee-Model" intending to explain the relation between project management and engineering. It acknowledges the iterative and incremental nature of engineering and hence encourages concurrent engineering. Forsberg and Mooz mention a major contribution to the "V-Chart" by Richard Roy.

In 1992, Germany published a standard for governmental IT projects. Apparently, the standard originated from the Ministry of Defense and was there established in the very early 1990s or even earlier. This comprehensive standard is a process description designated as "V-Modell®". The Vee does not stand for the graphical representation of the process model. "V-Modell®" is an abbreviation for the German

"Vorgehensmodell" what could be translated as "process model". Nevertheless, the described processes are depicted in a V-shape. The 1992 edition covered only SW development. It was made available to the public by Bröhl and Dröschel in "Das V-Modell" [18] in 1993. A further developed edition of the "V-Modell®", published in 1997, included systems aspects with references to the system definition by ISO 12207. The standard evolved and became the "V-Modell® XT" with the XT standing for extendable or extreme tailoring. The 1997 edition of the "V-Modell®" as well as the current edition of the "V-Modell® XT" are accessible on a dedicated web site [27], in German and partly in English.

At the INCOSE International Symposium 2013, Dieter Scheithauer and Kevin Forsberg presented the paper "V-Model Views" [124]. This paper collects experiences and improvements from the preceding two decades. The authors extend the scope of the V-Model from the development process to the life cycle of the system, reaching from stakeholders' needs to stakeholders' satisfaction. Compared to the paper of 1991, it splits the overall view into four distinct views: Basic-V, Development-V, Assurance-V, and Dynamic-V. It transforms the horizontal dimension from a time or maturity sequence to a logical sequence of the value stream. Scheithauer and Forsberg emphasize the inductive design of left side of the Vee in opposition to a deductive decomposition imposed by a waterfall-like process. And finally, it exemplifies that validation does not only apply for the finally deployed system but also to each artifact along the life cycle of the system. That is, stakeholder requirements, system requirements, architecture on each level, and integrated into the whole system architecture as well as the operating system need to be validated, meaning checked for their fitness for purpose. The authors state in the paper that the V-Model had been introduced twice, in the 1980's by NASA and with paper by Forsberg and Mooz in 1991.

B.2 A HANDY ILLUSTRATION BUT NO COMPREHENSIVE PROCESS DESCRIPTION

The Systems Engineering Vee is an illustration that depicts only some aspects of the development process of a system. Apart from the German "V-Modell®", the Systems Engineering Vee is no comprehensive process model or process description. Probably the most important aspect depicted in the Vee is the multilevel nature of systems. Based on the

definition, a system needs to comprise at least two levels. Therefore, the simplest Vee considering the system only would comprise two levels, the system level and the system element level.

The levels of the Vee correlate with the system levels. This may include also logical levels. The designation of these levels varies depending on the context of application. But each consecutive level pair represents a system in its own right. The lowest level is a bit special, as parts or components represented by this level will not further be partitioned from the viewpoint of the depicted Vee. This imposes that elements represented by the lowest level can be acquired, produced, or harvested. The composition or aggregation of these elements does not matter for the development system of interest. This does not preclude the lowest level elements to be also systems viewed from a different viewpoint. As with each model, the extent of a V-Model should follow a purpose. It typically includes the levels for which a certain team or organization is responsible plus the adjacent lower level and sometimes the adjacent upper level.

Depicting development-related life cycle processes of systems within the Vee, each level needs to comprise instances of the same processes. On each level, requirements engineering, architecting, integration, and verification needs to be performed. The concrete naming of these processes will vary depending on the organization applying these processes. As a consequence, artifacts resulting from these processes will exhibit the Zig-Zag pattern as described in Section 7.1. Figure B.2 depicts a Vee with the basic development processes creating artifacts as named in Figure 7.4. It displays that the Zig-Zag pattern not only exists on the left side of the Vee but proceeds up the right side as well as the system being integrated.

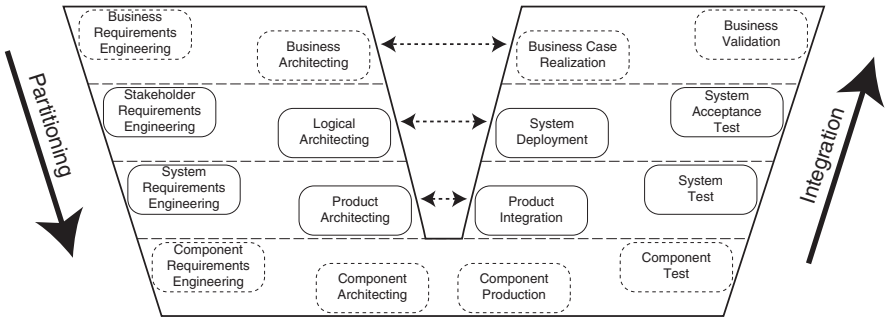

Figure B.2. V-Model with exemplary named basic development processes.

The dotted lines in Figure B.2 designate processes not in the responsibility of the assumed development. The system levels addressed in the Vee are from the top:

- The business as a system to earn money
- A stakeholder process gaining benefit from using the product to be developed
- The product to be developed
- Components from which the product is built.

Figure B.2 does not serve as a development process description. Many important elements are elided. It does not depict any control or object flows. It only maps processes to system levels. The figure designates a requirements engineering process but does not make any distinction between stakeholder and system requirements. In the absence of any object flows it makes no distinction between allocated requirements and elicited requirements. The integration processes are shown with a variety of names and verification and validation processes are mixed and partly designated as test. Figure B.2 follows a widely used, but rather suboptimal naming. Test is used as a synonym for verification and validation, imposing verification or validation are only effective by testing. This neglects other, often more efficient and less costly methods such as inspection, analysis, or demonstration. Finally, validation is reduced to an activity at the very end, though it should start during the stakeholder requirements definition process. Validation happens virtually always within other processes, though sometimes as an isolated task.

The rather simple illustration depicted above neglects a further issue. The number of elements in a lower level is higher than in levels above. Reusing elements in modular systems will degrade the increase of element kinds in subsequent levels. Each of these elements has its own life cycle. Their development or integration sequence depends on availability of data from the adjacent elements. For a comprehensive illustration representing the multiplicity of element kinds, a three-dimensional Vee need to be drawn. This third dimension would consider the interaction between the processes related to different elements on the same level. One can easily imagine, that an illustration with a three-dimensional Vee as depicted in Figure B.3 enriched with processes as displayed above becomes difficult to read and understand. Considering the elided items mentioned before would make it even worse.

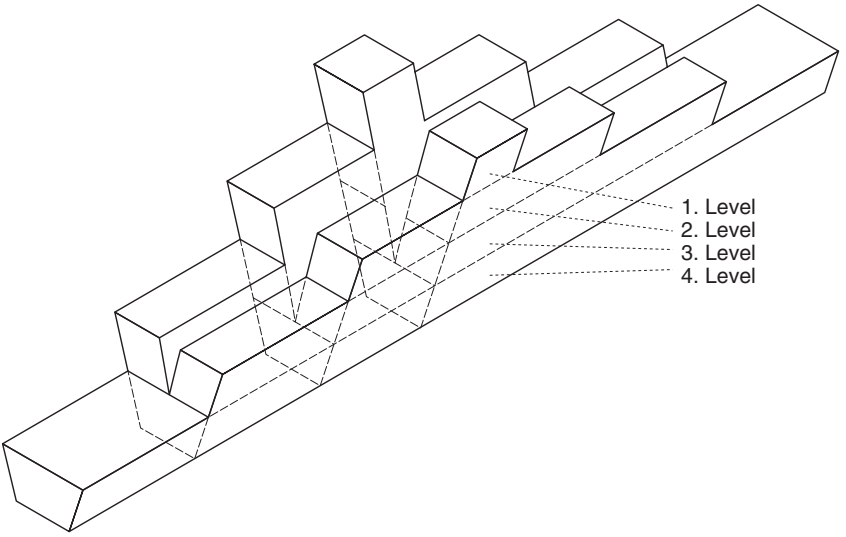

Figure B.3. V-Model considering discrete levels and numbers of elements.

B.3 CRITICAL CONSIDERATIONS

The V-Model received many interpretations. Many of them did not consider the original purpose and criticized the model for not fitting another purpose. The world evolved during the decades since the introduction of the V-Model. New or refined engineering methods came up, new technology and tools provide possibilities to improve effectivity and efficiency of our work. The V-Model can still help to illustrate the multilevel nature of systems and dependencies during the development process. But this can hardly be illustrated in a single view. Dieter Scheithauer and Kevin Forsberg provided with their paper "V-Model Views" [124] a very good explanation and summary for an appropriate application of the V-Model. We address hereafter some issues that frequently lead to discussions.

B.3.1 The V-Model as Process Description

The Systems Engineering Vee, unlike the German "V-Modell®", was never intended as a comprehensive process description. It is intended to depict the multiple levels of a system and only some specific process aspects related to such levels and their dependencies. The V-Model can assist in explaining that the life cycle processes need to be applied at each level for each element belonging to these levels.

B.3.2 The V-Model does not Impose a Waterfall Process

Interpreting the V-Model as equal or similar as a waterfall approach, neglects the meaning of levels within the V-Model. In a waterfall approach, each process, requirements engineering, architecting, implementation, verification, and final validation is considered on its own level. A waterfall process is intended to flow from the top (requirements engineering) down to the bottom (final validation). When comparing such approach with the V-Model, the second half of that cascade is bended up to shape a Vee. This brings requirements engineering and final validation to the top level, followed by architecting and verification on the intermediate level and implementation at the tip of the Vee. This view neglects the meaning of the levels to represent the levels of the system rather than some artificial hierarchy of processes. The horizontal dimension of the Vee represents a logical sequence in the value stream and not a time line of the development process. The inevitably required communication when allocating requirements to lower levels and subsequently validating them requires and ensures a bidirectional exchange of data. Considering the life cycle of each level and hence applying recursively the life cycle processes at each level results in a distinction between allocated requirements and other stakeholder requirements. The upper level becomes one and only one among other stakeholders of the lower level. Unlike with a waterfall approach, where all requirements are supposed to be available up front, the V-Model permits and requires induction of stakeholder requirements at lower levels.

B.3.3 The V-Model Accommodates Iterations

Maintaining a consistent configuration baseline on each level enables iterations in several forms. Iterations are possible on system element level on either side of the Vee. They can involve one or more levels. But also the big iterations including both sides over one or more levels are possible. Especially, the small iteration loops build up on early validation, such as checking requirements allocation and design of system elements for their fitness to contribute to the overall goal. Virtual integration of such system elements in the system model permits validation of each system level and each system element before their implementation starts. Iterative approach in the development of systems is not new and had been addressed already by the invention of Scrum [133]. Forsberg and Mooz promoted iterative development in their 1991-paper [40]. The same applies for early validation of artifacts. Though not described in

detail, Boehm mentioned in his 1979-paper [15] as example requirements validation, design validation, and validation tests for the final software. He did already emphasize the benefits of early validation.

B.3.4 The V-Model Permits Incremental Development

Using system models and applying iterations as explained above permits incremental approaches. The use of system models permits simulations and demonstration of the virtual integrated system. This permits early validation of the system with the related stakeholders. Incremental development was already promoted by Forsberg and Mooz in 1991 [40].

B.3.5 The V-Model and Concurrent Engineering

Since the V-Model emphasizes system levels, each with a number of system elements with their life cycle, it can be used to explain the impact of concurrent engineering. Maintaining a consistent configuration baseline and defining increments for each iteration enables concurrent engineering.

B.3.6 The V-Model Accommodates Change

An incremental and iterative approach and the maintenance of a consistent configuration baseline permits to predict impact of injected changes. The V-Model can assist in explaining where change can be expected. And it can visualize where these changes impact the development. Sources of change include discovering of new stakeholders, changing stakeholder needs, each validation or verification step.

B.3.7 The V-Model Permits Early Verification Planning

An incremental and iterative approach and the maintenance of a consistent configuration baseline permits early verification planning. Consequently, optimization of the verification processes and related infrastructure can be achieved. This includes combining verifications of different levels or combining interface verification during integration with system or system element verification.

B.3.8 The V-Model Shows where to Prevent Defects

The V-Model can display where validation should be performed. Validation does not only apply to the top-rightmost part of the Systems Engineering Vee. Each artifact created during a development should

be validated. That is, evidence should be provided that the considered artifact contributes to satisfy stakeholders. Early validation ensures that the right problem is solved. Literature does often not very explicitly mention such early validation. The standard ISO/IEC 15288:2008 [60] mentions a number of artifacts to be validated, though only in notes and not in the normative text. The INCOSE Systems Engineering Handbook [56] designates such early validation as "in-process validation" or "continuous validation". The standard ISO/IEC/IEEE 29148:2011 [63] emphasizes the necessity of requirements validation. Scheithauer and Forsberg make the execution of validation very explicit in their Assurance-V [124]. Validation, and hence defect prevention starts at the top-leftmost part of the V-Model and proceeds all the way down and up again, even into the operation phase of the system of interest. Six kinds of validations can be identified along a system life cycle:

1. Validation of stakeholder requirements
2. Validation of allocated requirements
3. Validation of system element definitions
4. Validation of the virtually integrated system
5. Validation of the operational system deployed into its environment
6. Validation of the in-service system.

B.4 READING INSTRUCTION FOR A MODERN SYSTEMS ENGINEERING VEE

To summarize this chapter, we provide a reading instruction for V-Models. As mentioned earlier, a single view can hardly depict each aspect. Views have to be designed to frame concern of a dedicated audience and elide other aspects. Following the seven rules stated hereafter will support a consistent understanding.

B.4.1 The Vertical Dimension

The vertical dimension of the V-Model denotes the multiple levels of the system of interest. The topmost level represents the system context in which the system operates. The bottommost level represents the parts that can be obtained and hence suffice to be defined in a black-box-view. The levels in between represent physical or logical levels of the system. Therefore, the V-Model should comprise of at least three levels: system context, system, and system elements.

B.4.2 The Horizontal Dimension

The horizontal dimension of the V-Model denotes a logical sequence of the value stream. This does not impose that each requirement needs to be frozen upfront. An increment in development may impose to develop some parts before defining the remaining requirements on top level.

B.4.3 The Left Side

The left side of the V-Model denotes the general direction of the top-down development. This is only the general direction. Especially, interface-related issue require to push parts of the development to very low levels before proceeding with the remaining parts of the upper levels.

B.4.4 The Right Side

The right side of the V-Model denotes the general direction of the bottom-up integration. A model-based approach enables virtual integration. This results in a number of instances of the right side. Such representations are sometimes called Y-Model as such virtual integration can be depicted with a branch starting in the middle of the left side and heading upward in parallel to the right side of the V.

B.4.5 The Levels

The Levels represent the system levels. Life cycle processes are applied on each level. Levels get requirements allocated from the next level above, elicit requirements from its specific stakeholders and allocate requirements to the next level below. Levels receive verified system elements (or parts of) from the next level below and provide the integrated and verified system (or parts of) to the next level above.

B.4.6 Life Cycle Processes

The life cycle processes such as stakeholder requirements definition, requirements analysis, architectural design, integration, verification, validation, and the related artifacts appear on each level.

B.4.7 The Third Dimension

The third dimension of the V-Model can be used for different purposes. For instance to display the multitude of system elements per level resulting in parallel application of the life cycle processes.