

The “V” Model Reloaded

Casey Medina, CSEP

CVM Design, Inc.

casey@cvmdesigninc.com

10548 Kipling Pl.

Westminster, CO 80021

Copyright © 2015 Casey Medina. Permission granted to INCOSE to publish and use.

Abstract: Systems engineers can provide valuable input and support to product development efforts. In order to provide that support, the systems engineer must first convince his or her team that systems engineering methodologies are valuable. This paper presents a new view of the systems engineering “V” model that aims to make the model accessible and understandable to teams with little to no systems engineering experience. In doing so, it opens the door for broader adoption of systems engineering.

Introduction

The “V” model has become one of the most ubiquitous symbols of systems engineering. It is revered by seasoned and novice systems engineers alike. The problem, however, is that we often find ourselves alone in our love for the “V.” While many development teams may listen to the systems engineer discuss all the reasons they should consider employing the model in their development effort, not all of those teams are likely to embrace it fully. Teams that do attempt to employ the model in their processes often fall short as they perceive the model to represent a cumbersome, lengthy approach to development when they “just need to get a product out.” This paper presents a different view of the system “V,” a view that will help even the most resistant teams understand its purpose and the beauty behind its elegant form.

When we first learn about the “V” model, we are usually presented with an image like this one (Blanchard, 1998):

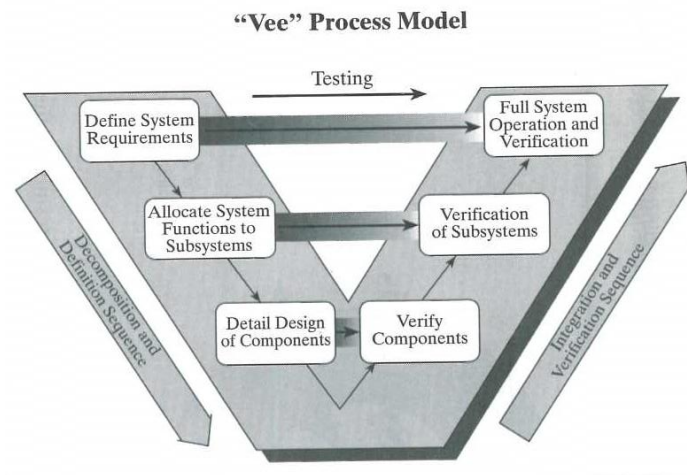


Figure 1. Classical representation of the “V” model.

Similarly, we may see this image from the SE Handbook (Haskins, 2011):

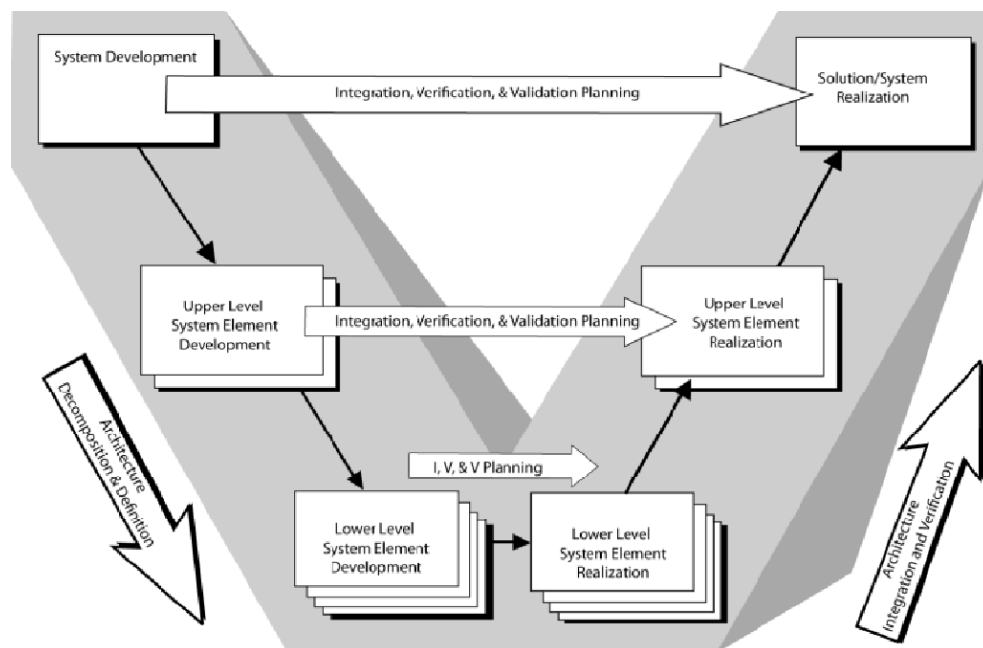


Figure 2. “V” model as shown in the SE Handbook.

If we began our careers in medical devices, we likely favor the waterfall diagram preferred by the United States Food and Drug Administration (CDRH, 1997).

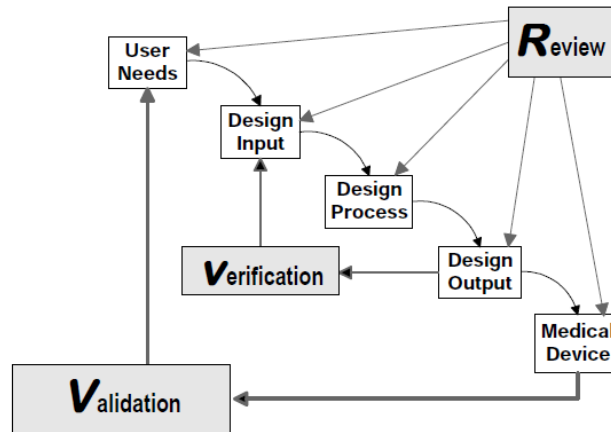


Figure 3. FDA Waterfall model of product development

As systems engineers, we know these models work. We understand them. We also, in the course of a development effort, witness our teams become so focused on tactical implementation details that they lose sight of the strategy embodied by the model. The following series of figures are not merely a different way of describing the ubiquitous “V” model; they are a fresh look at what it represents in a manner that is accessible to everyone on our project teams, regardless of their technical backgrounds. The author acknowledges representations of the “V” model in literature are more complex (Sheithauer and Forsberg, 2013) and varied (Forsberg and Mooz, 2001) in order to accurately identify the details of the lifecycle phase and concepts being described. Moreover, many versions of the “V” model exist in the context of the organization or environment in which the model will be employed. This can create inconsistencies in the communication of key systems engineering activities (Sheithauer and Forsberg, 2013). The “V” model herein serves to simplify these representations in a manner that fosters willing adoption by organizations new to systems engineering by describing the purpose of the systems engineering activities rather than focusing on the activities themselves.

Building the Foundation

To truly understand its beauty, we must begin with the model’s fundamental construct: its form. In Figure 4, we see the model’s simple, elegant shape.

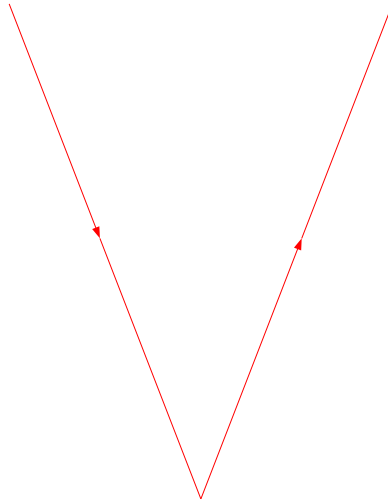


Figure 4. Study of the “V” model form.

It is obvious that the shape exists in two dimensions. The vertical dimension represents depth of knowledge. The horizontal dimension represents time. (Forsberg and Mooz, 1991) As we progress through time, we achieve greater depths of understanding of the solution space in details that are actionable until we reach the point where engineers and developers have sufficient information about the solution space to generate a detailed design. From here we progress upward through those levels of understanding to ensure that the detailed design does, in fact, exhibit the necessary behavior and performance of the defined solution space.

This information seems elementary. In theory, perhaps it is. In practice, it is quite the opposite. Schedule and budget pressure and a general, underlying desire to “just get the product out” can cloud our implementation of the “V” model. Additionally, teams may feel the “V” model is not representative of agile development as the model exists in a sequential hierarchy (Sheithauer and Forsberg, 2013). By redirecting our focus to the model’s shape and what it represents, we enable our projects to progress purposefully and properly. The shape becomes more than a development process model. It becomes a representation of how we, as humans, solve problems.

Asking the Right Questions

The design process answers questions. Sometimes, we ask the questions before we begin designing. This is ideal. Sometimes, we begin designing and ask the questions later. This is not ideal but often viewed as “practical” by sales-driven organizations. Sometimes, we design and then try to find a question our solution will answer. This is trouble. It’s also a great way to assume a lot of financial and general business risk. If we ask the questions at the right time, how do we know the correct questions are being asked to ensure a great product will be developed in a timely manner that provides value to its stakeholders?

Fortunately, the “V” model helps teams focus their efforts to provide high value to stakeholders. By viewing the model through a different window, we bring light to key, fundamental questions cloaked by more technical representations of the model. In other

words, we help teams understand the motivations driving the systems engineering activities described in more traditional views of the model. Figure 5 illustrates the questions that are necessary to gain a key understanding of the problem definition process.

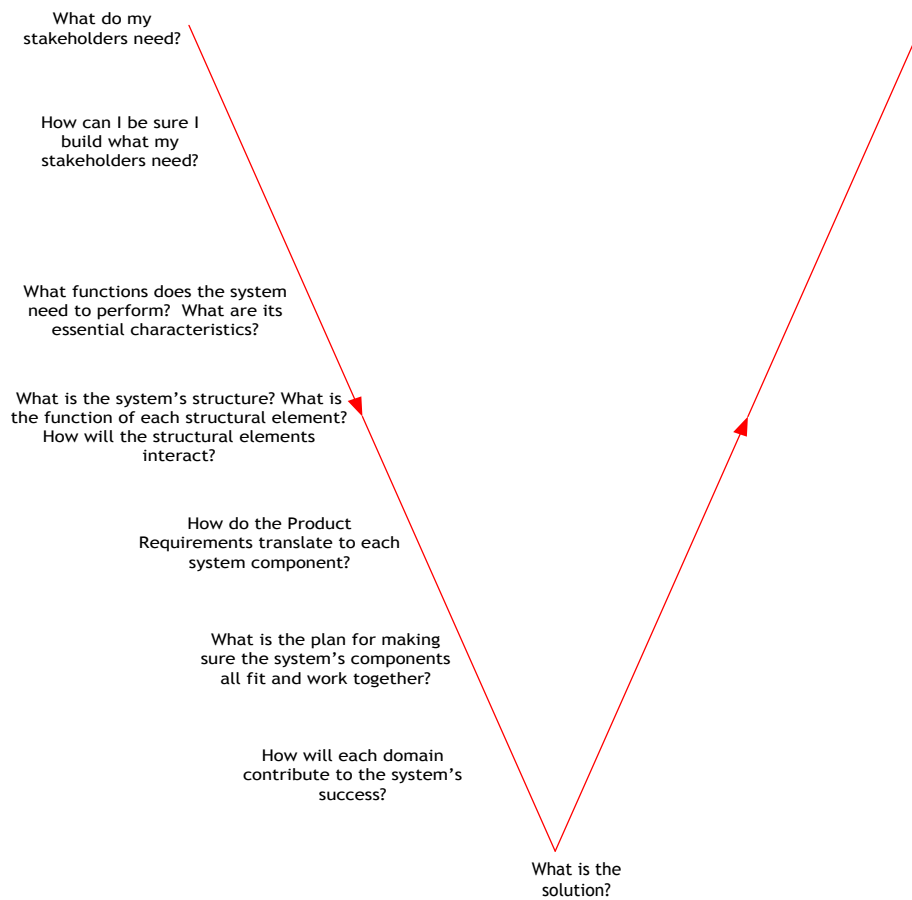


Figure 5. Understanding the problem space.

The first question in Figure 5 encourages teams to obtain a thorough understanding of the stakeholders and the stakeholders' wants and desires. The second question requires teams to translate the stakeholder research into actionable engineering terms that will be captured as a concept of operations and stakeholder requirements or, as the FDA refers to them, "user needs" (CDRH, 1997). The next question helps direct the team's focus to describing the "white box" behavior of the system. This is followed by a description of the logical organization of the intended system, i.e. logical architecture. The next question asks the team to describe the behaviors and performance of each logical group. Teams are then prompted to develop a plan for integrating the system. Integration can be an unfamiliar concept to many development teams. The assumption is that, if they put all the pieces together and it works, then they did their job. This assumption leads to poorly-understood interfaces and a higher likelihood that "unexplained" issues will surface in the hands of the customer (Armstrong, 2014). The final step before detailed design is to describe how each discipline will contribute to the success of the system.

Systems engineers recognize these efforts as requirements engineering, system architecting and integration planning. Why not just say that? Simply stated, we don't use those systems engineering terms because many teams don't understand them and many don't want to understand them. We must help them understand the development process in terms of what it accomplishes. Only after this understanding is achieved can we begin to assign more technical terms. We always need to be aware of our audience and present the material in a manner that suits them.

Figure 5 shows the detailed design effort located at the bottom of the "V." Some models show the "V" with a "flat" bottom, as in Figure 2. If we return to the description of the model's form and the acknowledgement of the vertical dimension as our depth of understanding of the problem space, we see that the "point" at the bottom of the "V" is consistent with this acknowledgement. Our depth of understanding reaches a maximum with our understanding of the implemented design.

As we begin to ensure our design solves the problem statement, we ask pointed questions that remind teams of the purpose of verification and validation (V&V). Figure 6 illustrates the questions that expand traditional descriptions of V&V, popularly-stated as "Did I build the *thing right*?" and "Did I build the *right thing*?" We provide our teams with an understanding of the activities that put these two ubiquitous, broad questions into perspective in the context of the development effort and the system.

Progressing up the left side of the "V" in Figure 6, we begin to objectively evaluate our design. The first question, "Did I satisfy my domain requirements?" leads teams to understand the scope of verification. Verification happens at all levels of system hierarchy; it is not just performed on the final, fully-integrated system (Haskins, 2011). The next question, "How do the pieces fit together?" helps teams realize the integration process is more than just assembly of the system elements. The next two questions represent iterations of integration and corresponding verification. The author acknowledges that many more integration/verification iterations may exist than are identified in Figure 6. The iterations represented in Figure 6 are intended to help teams understand the concept of integration and verification. The model then prompts teams to step through system verification and design validation. The final question, "How does the market respond to my design?" encourages our teams to consider post-market monitoring of their systems.

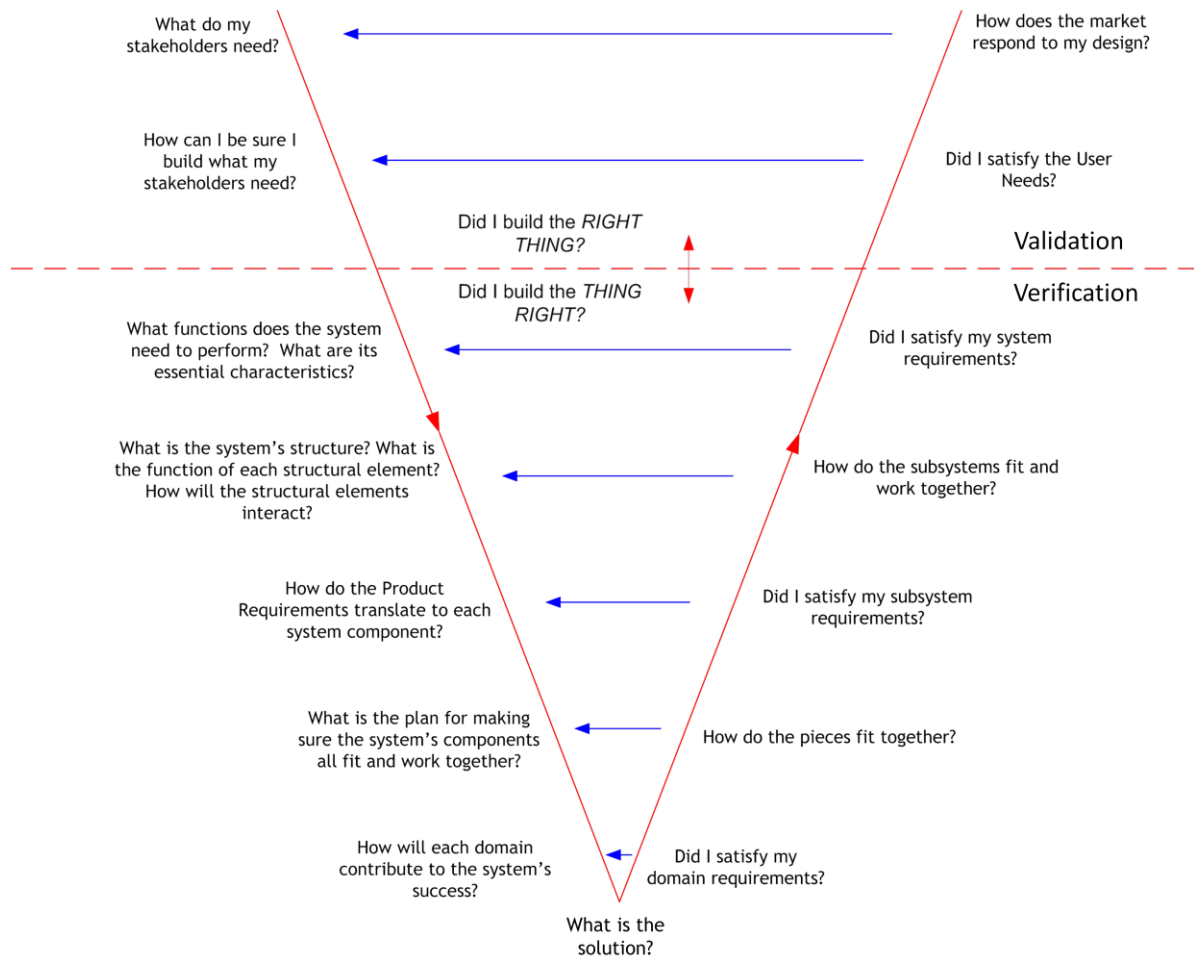


Figure 6. The “V” model demonstrating verification and validation questions provide evidence that the implemented design answered the problem definition questions.

Note that in Figure 6, the arrows are pointing from the left side of the “V” to the right side. This is done to help teams understand that V&V activities consider the design in terms of the problem it was intended to solve. In a 2011 report, the FDA identified that more than one third of all medical device recalls between 2003 and 2009 were attributed to design (FDA, 2011). In that same report, the FDA claims that a likely cause for the recall rate is the difficulty companies experience “in designing medical products for actual, and not merely intended, use” (FDA, 2011). Other causes included lack of focus on reliability engineering and poor software quality due to ineffective integration and verification (FDA, 2011). By examining the “V” model in Figure 6, we are reminded that the verification of a system or system element is directly related to the quality of the requirements to be verified. Again, examining the “V” model, we see that a solid understanding of the stakeholder needs is required to enable teams to adequately answer all of the following questions in Figure 6. Therefore, it is necessary to help our teams perform adequate analysis of the stakeholder needs to enable a directed, focused design effort.

As systems engineers, we recognize that all of our requirements, design, and V&V work needs to trace to a solid understanding of the stakeholder’s motivations and needs driving development of the intended system. If we fail to ensure that all requirements and design

work is in support of achieving a valuable mission, we have little chance of honestly answering the question “Did I satisfy the user needs?” with a confident “YES!” We have, in effect, set our project up to fail. Organizations with established systems engineering teams will likely capture a validated description of the mission and purpose of the intended system in a concept of operations (ConOps) document. Some industries, like the medical device industry, that have not fully embraced systems engineering may not write a ConOps at all. In this case, it is difficult to answer the questions on the left side of the “V” with any certainty as a cohesive description of the intended system’s behavior in its intended use environments does not always exist. If this description does exist, it may or may not be validated with the stakeholders. This allows the system description to change with the perceived ebb and flow of a particular market segment, often resulting in “scope creep.” It forces teams to assume a reactionary posture rather than proactively defining a system with high market value. The following questions are invaluable in helping development teams understand and tell the story of their intended system in accessible language. These questions encourage teams to consider the lifecycle of the intended system so that teams are encouraged to think holistically about their projects to focus stakeholder needs research and subsequent stakeholder requirements elicitation. By answering these questions, teams capture the essential information contained in a ConOps without the perceived burden of creating the document per the IEEE 1362:1998 standard.

Question 1: Why is my idea important? This is, perhaps, the simplest question to ask and one that is often not answered until the development effort is largely complete. The answer to this question helps lay the foundation for the development of an effective mission statement.

Question 2: How will my idea help? This question helps the team understand the value proposition at the beginning of the project so that the organization can evaluate its attractiveness to target markets.

Question 3: Why will my idea be used? The answer to this question gives a glimpse into the minds of stakeholders. It also helps identify stakeholders that may have not been considered otherwise. In answering this question, it is important to encourage teams to focus on use case descriptions rather than creating detailed analyses of the step-by-step use of the intended system. This is an overview that will be used when planning and performing stakeholder research.

Question 4: Why is my idea better than other ideas currently available? This question further refines the value proposition to help the organization decide if the project should receive funding. It also identifies opportunities for innovation in the market space to create enhanced stakeholder value.

Question 5: Who will like my idea? Why? On the surface, these two questions seem elementary. Looking more deeply, we find these questions actually encourage teams to consider stakeholders and users early. They create a basis on which teams can further tailor stakeholder research and identify unmet needs.

Question 6: Who won't like my idea? Why? Understanding which stakeholders won't like the intended system and their motivation for disliking it is every bit as valuable as understanding who will like the intended system and why. The class of stakeholders identified by answering these questions presents threats to market acceptance as well as operational threats to the deployed system. The motivations identified here may not be malicious threats. They may be the result of bad experiences with similar products or an unwillingness to adopt new technologies. A nurse who has had difficulty adjusting drug delivery of an infusion pump that incorporated a touch screen is likely to distrust any new infusion pumps with touch screens until they have proven to be easier to use than current models.

Question 7: How will someone acquire my idea? One of the key contributions of systems engineering is the consideration of the lifecycle of a system during its planning and development. This question encourages teams to understand the value of considering the beginning of the system's lifecycle. It will help with product placement, pricing strategy, and logistics planning. An effective answer to this question will also help teams develop good requirements for the acquisition of the system including shipping and packaging requirements.

Question 8: What is a "day in the life" of my idea? This is the team's opportunity to describe how they envision the intended system will be used. Initially, this may be a set of assumptions or even "best guesses" based on the current understanding of the operational environments. Capturing this information early allows teams to validate their assumptions with stakeholder research. Once validated, the answer to Question 8 provides a foundation upon which teams can begin early proof-of-concept prototyping and even human factors analyses.

Question 9: How will someone discard my idea when they are done using it? This question requires teams to consider the complete lifecycle of the intended system. It will likely become a source for the invocation of standards and regulations. Question 9 also encourages teams to consider product roadmaps. The strategy for disposal and phase out will enable the organization to begin planning for a follow-up product or products to guarantee a seamless product offering to its customers.

The nine questions above represent the minimum body of information necessary for a team to formulate a solid mission statement and tell a cohesive story to their organization and stakeholders. These questions are not targeted at a specific industry. It is advantageous to refine the answers as new information becomes available to ensure they remain valid. If new information results in a change to any answer rather than a refinement of it, the change should be validated and its impact should be assessed to all of the items tracing to it. By employing the "V" model, understanding traces from the ConOps through requirements, design, and V&V becomes less daunting. These nine questions may be answered in a group forum, by survey, or by individual interview. They are intended to be non-intrusive and, asked correctly, help individuals and teams see the systems engineering effort in a positive light. They feel invested in the process.

Asking the Questions Right

The questions presented in this paper are stated in accessible, first-person language to allow them to be easily understood, stress the elegant simplicity of the question set, and to remove the stigma that a solid systems engineering effort adds unnecessary project delays. Asking the questions in the first person, whenever possible, places personal, individual ownership on the questions and their associated answers. In this manner we create an environment where individuals feel engaged; they feel self-obligated and accountable for answering the stated questions completely and correctly. They feel their input is both valued and valuable. We have created a safe environment in which individuals can share ideas.

Asking questions is one of the most effective methods to facilitate great discussion among teams and between teams and stakeholders. The savvy systems engineer must be careful to ensure that the questions are not delivered in a confrontational or accusational manner. Questions are asked to find facts, to discover unmet needs, and to help teams reach consensus and move forward. The right question delivered poorly at a critical moment can derail the effort and foster animosity. Well-delivered questions successfully place the team's attention on the problem at hand and preserve relationships.

As teams answer each question, they gain a deeper understanding of the problem space and they begin to understand the flow of information. The purpose of the systems engineering effort is to ensure that all members of the development team are working together with a common understanding of the system's mission to satisfy stakeholders (Haskins, 2011). As systems engineers, it is our responsibility to bring that clarity and understanding to our teams. If we can achieve that mission using standard systems engineering methodologies and tools with a willing team, great. In many cases, we find teams resistant to classical approaches to systems engineering. It is unrealistic to expect that a team or organization new to systems engineering will openly embrace the entirety of the art. We need to make it accessible, even if that means we have to present simpler alternatives to our technical processes – i.e. we need to “ask the questions right” for our particular team culture. The body of systems engineering knowledge is absolutely useless if we are unsuccessful in helping our teams implement good systems engineering practices.

Figure 6 makes the systems engineering effort accessible by asking questions easily understood by all members of the development team. By distilling the development process into a series of questions, we help our teams understand the flow of information and the importance of defining the problem space to wisely invest development effort and financial resources in the creation of the right solution. Rather than describing the decomposition of the problem space in terms of requirements engineering and logical architecture development, we state the purpose of those activities. Likewise, many teams confuse system verification and system validation. By relating verification and validation to the questions used to define the problem space, we remove the confusion surrounding those activities. This makes the model less intimidating to our teams and allows them to more readily adopt the use of systems engineering tools and methodologies to help answer the necessary questions. By focusing on these questions, every member of the development team can more easily

understand how his or her role contributes to a successful description of the solution space. This translates to better, more complete, requirements earlier in the development cycle, more informed logical architecture solutions, and more thorough V&V efforts.

Don't Forget, the Process is Iterative

Often, the “V” model is incorrectly interpreted as a strictly-linear approach to product development. It is true that there is a logical and natural flow of information and activities in a development effort. It is also true that the information and activities can be refined or repeated. The “V” model is representative of large and small projects and is as equally-applicable to segmented development efforts as it is to the entire effort. Scheithauer and Forsberg describe several views of the “V” model showing the iterative nature of development and how that is represented by the “V” (Sheithauer and Forsberg, 2013). Perhaps the most challenging factor in adoption of the “V” model and any structured process derived from it is the perception that iteration is not allowed. Organizational pressure encourages a “one chance to get it right” approach to development. Mistakes are not tolerated and iteration is viewed as a mistake. Beasley, Nolan, and Pickard describe the dangers of pressuring teams to “get it right the first time” and some tactics to better plan for iteration (Beasley, 2014).

Overcoming Barriers to Acceptance

Systems engineers possess the ability to think holistically and strategically about problems. What is obvious to a systems engineer is not necessarily obvious to others on our teams. As a result we must exhibit confidence in our use of tools and methodologies and embody patience when teaching those tools and methodologies to our teams. In the author's experience, one of the largest barriers to acceptance for any systems engineering effort is the perception of increased workload with little to no actual product return. This issue is compounded by a strong affinity for legacy product development processes that do not incorporate systems engineering approaches as the legacy processes feel simpler. This paper presents an accessible description of the “V” model of product development that will help systems engineers gain acceptance and foster adoption. By replacing the technical “systems speak” from our descriptions of system development activities with more accessible language, we enable our teams to better understand how systems engineering and the “V” model enable good product development processes. Teams more clearly understand how requirements, design and V&V are linked to the overall objective and value proposition of the intended system. When we help our teams achieve this level of understanding, we enable them to experience greater success for themselves, for their companies, and for their stakeholders.

Conclusion

The “V” Model is widely accepted and utilized by systems engineers across industries (Haskins, 2011). The “V” Model described in this paper was utilized to implement a new product development process at a major medical device manufacturer. Due to the proprietary nature of the projects, only anecdotal evidence can be described to indicate the success of the

“V” model in fostering adoption for the new process. The new process is structured around and based on the “V” model. As a result, the company was able to identify gaps in the previous process and understand the relationships that would be improved by introducing some new processes and revising current processes. Processes that were introduced included a system architecture process, a human factors engineering process, and a combined integration/verification process. The “V” model was instrumental in showing teams how these tasks fit into the system development lifecycle and how they had a positive impact on the design process and corresponding sustainability efforts.

The requirements definition process was enhanced with a better understanding of the mission of the intended system. This led to increased focus on traceability and decomposition from stakeholder requirements to low-level system element requirements. Teams are now able to demonstrate the value each requirement brings in the context of the mission. Verification and validation activities became more focused and resulted in better coverage as requirements development improved and interfaces became better defined.

Prior to the implementation of the “V” model, only software requirements were decomposed and allocated below the system requirement level and subsequently verified. Consequently, only system-level verification was explicitly covered by formal verification and validation. Additional verifications of non-software system elements were performed at the recommendation of a senior engineer based on “tribal knowledge” and experience. Following implementation of the “V” model, system element requirements are being written and are inclusive of previously-ignored elements such as labeling, localization and maintenance. As a result of more complete requirements, the verification efforts offer more complete coverage. The enhanced focus on the system story and mission statement has resulted in a better understanding of the concept of validation.

The “V” model depicted in this paper provides an accessible, meaningful explanation of the systems engineering process. It reduces complex, abstract topics to an understandable set of questions that communicate the motivation behind systems engineering processes. The “V” model has resulted in adoption of basic systems engineering principles and resulted in generation of better work products in the cases tested.

References

- Armstrong, J. 2014. Systems Integration: He Who Hesitates is Lost. Proceedings of the 24th International Symposium of the International Council on Systems Engineering, 2014.
- Beasley, R., Andy Nolan, and Andrew Pickard. 2014. When “Yes” is the Wrong Answer. *Proceedings of the 24th International Symposium of the International Council on Systems Engineering, 2014.*
- Blanchard, B. and Wolter Fabrycky. 1998. *Systems Engineering and Analysis*. 3rd ed. Upper Saddle River, NJ: Prentice Hall.

- “Design Control Guidance for Medical Device Manufacturers.” 1997. United States Food and Drug Administration Center for Devices and Radiological Health (CDRH). <http://www.fda.gov/RegulatoryInformation/Guidances/ucm070627.htm>
- Forsberg, K., and H. Mooz. 1991. The Relationship of System Engineering to the Project Cycle. *Proceedings of the 1st Annual Conference of NCOSE, 1991.*
- Haskins, C. ed., SE Handbook Working Group, *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. Version 3.2.2. Revised by K. Forsberg and M. Krueger. San Diego, CA (US): International Council on Systems Engineering, 2011.
- IEEE. 1362 – 1998 (R2007). *IEEE Guide for Information Technology – System Definition – Concept of Operations (ConOps) Document*. December 5, 2007.
- “Understanding Barriers to Medical Device Quality.” 2011. United States Food and Drug Administration (FDA). <http://www.fda.gov/downloads/AboutFDA/CentersOffices/CDRH/CDRHReports/UCM277323.pdf>
- Scheithauer, D. and K. Forsberg. 2013. V-Model Views. *Proceedings of the 23rd International Symposium of the International Council on Systems Engineering, 2013.*

Biography



Casey Medina, CSEP - Casey currently practices systems engineering in the medical device industry. He has developed products in the areas of automated stem cell growth, patient monitoring, pulse-oximetry, blood collections and therapeutics. Casey holds multiple patents in the medical device sector. Professionally, he is focused on developing the art of systems engineering in a manner that fosters adoption and acceptance by organizations resistant to change. He currently applies systems engineering practices and principles to enable rapid medical device development and is working to enhance the use of MBSE as an enabler for usability and human factors analyses.