System Lifecycle Process Models: Vee

Lead Authors: *Dick Fairley, Kevin Forsberg,* ***Contributing Author:*** *Ray Madachy, Phyllis Marbach*

There are a large number of life cycle process models. As discussed in the System Life Cycle Process Drivers and Choices article, those models described fall into three major categories: (1) primarily pre-specified single-step or multistep, also known as traditional or sequential processes; (2) evolutionary sequential (or the Vee Model) and (3) evolutionary opportunistic and evolutionary concurrent (or incremental agile). The concurrent processes are known by many names: the agile unified process (formerly the Rational Unified Process), the spiral models) and include some that are primarily interpersonal and unconstrained processes (e.g., agile development, Scrum, extreme programming (XP), the dynamic system development method, and innovation-based processes).

This article specifically focuses on the Vee Model as the primary example of pre-specified and sequential processes. In this discussion, it is important to note that the Vee model, and variations of the Vee model, all address the same basic set of systems engineering (SE) activities. The key difference between these models is the way in which they group and represent the aforementioned SE activities.

General implications of using the Vee model for system design and development are discussed below; for a more specific understanding of how this life cycle model impacts systems engineering activities, please see the other knowledge areas (KAs) in Part 3.

Contents

A Primarily Pre-specified and Sequential Process Model: The Vee Model

The sequential version of the Vee Model is shown in Figure 1. Its core involves a sequential progression of plans, specifications, and products that are baselined and put under configuration management. The vertical, two-headed arrow enables projects to perform concurrent opportunity and risk analyses, as well as continuous in-process validation. The Vee Model encompasses the first two life cycle stages listed in the "Generic Life Cycle Stages their purposes, and decision gate options" table of the INCOSE *Systems Engineering Handbook*: concept, and development (INCOSE 2015).

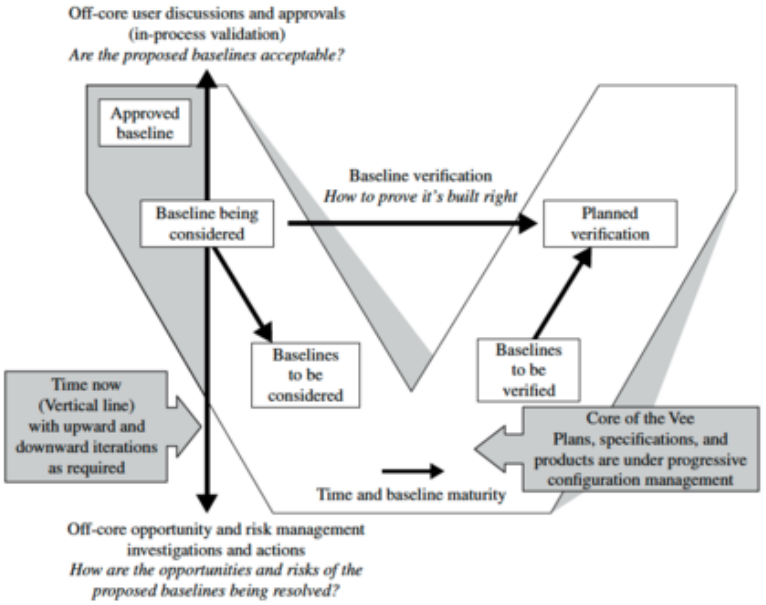

Figure 1. Left Side of the Sequential Vee Model (INCOSE 2015, adapted from Forsberg, Mooz, and Cotterman 2005, Reprinted with permission of John Wiley & Sons Inc. All other rights are reserved by the copyright owner.

The Vee Model endorses the INCOSE *Systems Engineering Handbook* (INCOSE 2015) definition of life cycle stages and their purposes or activities, as shown in Figure 2 below. Replace Figure 2 with the updated figure that removes the first Exploratory stage.

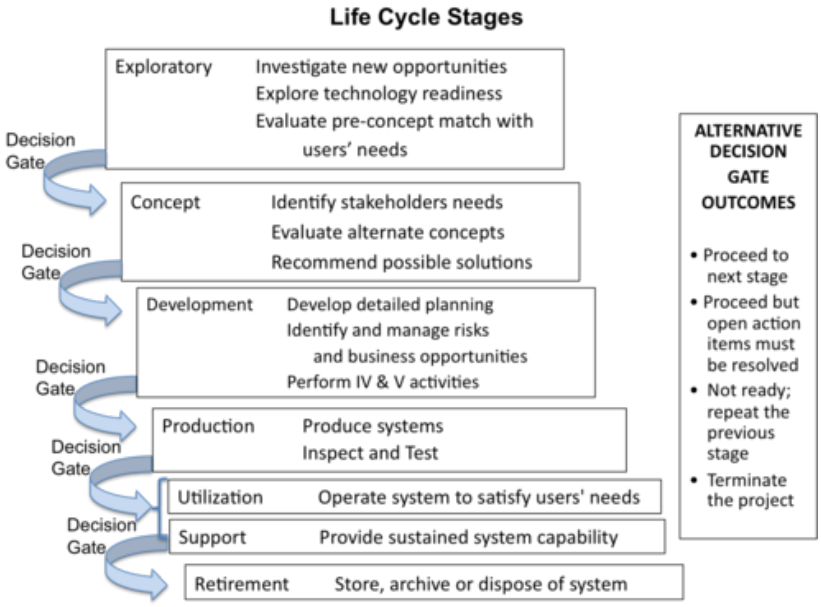

Figure 2. An Example of Stages, Their Purposes and Major Decision Gates. (SEBoK Original)

A more detailed version of the Vee diagram incorporates life cycle activities into the more generic Vee model. This Vee diagram, developed at the U.S. Defense Acquisition University (DAU), can be seen in Figure 3 below.

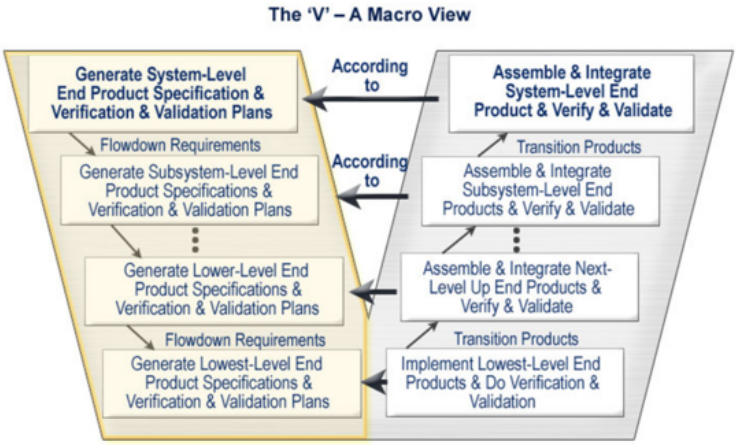

Figure 3. The Vee Activity Diagram (Prosnik 2010). Released by the Defense Acquisition University (DAU)/U.S. Department of Defense (DoD).

Application of the Vee Model

Lawson (Lawson 2010) elaborates on the activities in each life cycle stage and notes that it is useful to consider the structure of a generic life cycle stage model for any type of system-of-interest (SoI) as portrayed in Figure 4. This **(T)** model indicates that one or more definition stages precede a production stage(s) where the implementation (acquisition, provisioning, or development) of two or more system elements has been accomplished.

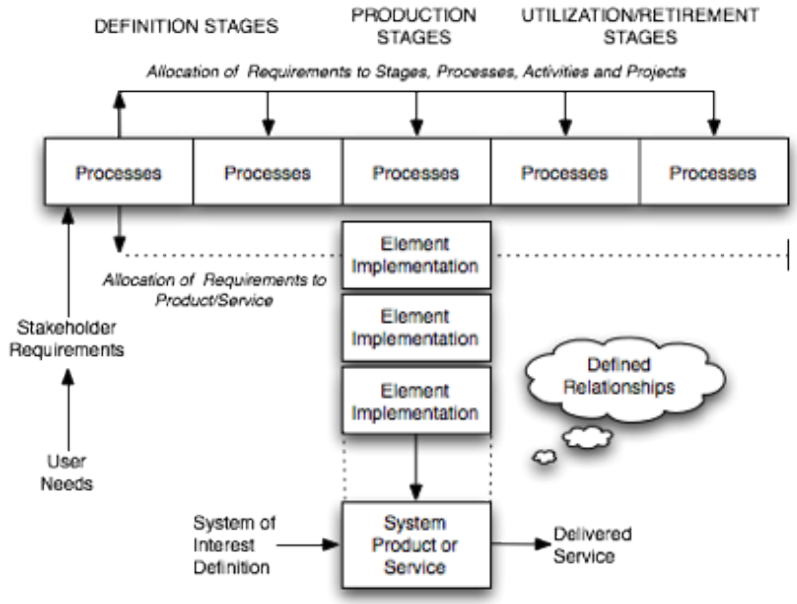

Figure 4. Generic (T) Stage Structure of System Life Cycle Models (Lawson 2010). Reprinted with permission of Harold Lawson. All other rights are reserved by the copyright owner.

Figure 5 shows the generic life cycle stages for a variety of stakeholders, from a standards organization (ISO/IEC)

to commercial and government organizations. Although these stages differ in detail, they all have a similar sequential format that emphasizes the core activities as noted in Table 1 (concept, production, and utilization/retirement).

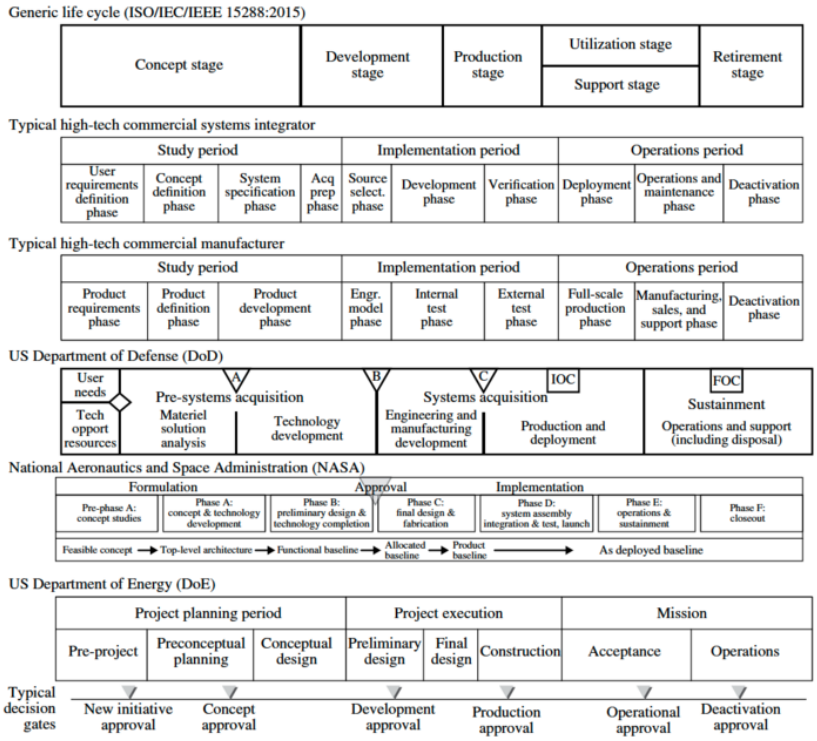

Figure 5. Comparisons of Life Cycle Models (Forsberg, Mooz, and Cotterman 2005). Reprinted with permission of John Wiley & Sons. All other rights are reserved by the copyright owner.

It is important to note that many of the activities throughout the life cycle are iterated. This is an example of recursion as discussed in the Part 3 Introduction.

Fundamentals of Life Cycle Stages and Program Management Phase

For this discussion, it is important to note that:

- The term **stage** refers to the different states of a system during its life cycle; some stages may overlap in time, such as the utilization stage and the support stage. The term “stage” is used in ISO/IEC/IEEE 15288.
- The term **phase** refers to the different steps of the program that support and manage the life of the system; the phases usually do not overlap. The term “phase” is used in many well-established models as an equivalent to the term “stage.”

Program management employs phases, milestones, and

decision gates which are used to assess the evolution of a system through its various stages. The stages contain the activities performed to achieve goals and serve to control and manage the sequence of stages and the transitions between each stage. For each project, it is essential to define and publish the terms and related definitions used on respective projects to minimize confusion.

A typical program is composed of the following phases:

- The **feasibility or study phase** consists of studying the feasibility of alternative concepts to reach a second decision gate before initiating the execution stage. During the feasibility phase, stakeholders' requirements and system requirements are identified, viable solutions are identified and studied, and virtual prototypes (glossary) can be implemented. During this phase, the decision to move forward is based on:
  - whether a concept is feasible and is considered able to counter an identified threat or exploit an opportunity;
  - whether a concept is sufficiently mature to warrant continued development of a new product or line of products; and
  - whether to approve a proposal generated in response to a request for proposal.
- The **execution phase** includes activities related to four stages of the system life cycle: *development, production, utilization, and support*. Typically, there are two decision gates and two milestones associated with execution activities. The first milestone provides the opportunity for management to review the plans for execution before giving the go-ahead. The second milestone provides the opportunity to review progress before the decision is made to initiate production. The decision gates during execution can be used to determine whether to produce the developed SoI and whether to improve it or retire it.

These program management views apply not only to the SoI, but also to its elements and structure.

Life Cycle Stages

Variations of the Vee model deal with the same general stages of a life cycle:

- New projects typically begin with an exploratory research phase which generally includes the activities of concept definition, specifically the topics of business or mission analysis and the understanding of stakeholder needs and requirements. These mature as the project goes from the exploratory stage to the concept stage to the development stage.
- The production phase includes the activities of system definition and system realization, as well as the development of the system requirements (glossary) and architecture (glossary) through verification and validation.
- The utilization phase includes the activities of system deployment and system operation.
- The support phase includes the activities of system maintenance, logistics, and product and service life management, which may include activities such as service life extension or capability updates, upgrades, and modernization.
- The retirement phase includes the activities of disposal and retirement, though in some models, activities such as service life extension or capability updates, upgrades, and modernization are grouped into the "retirement" phase.

Additional information on each of these stages can be found in the sections below (see links to additional Part 3 articles above for further detail). It is important to note that these life cycle stages, and the activities in each stage, are supported by a set of systems engineering management processes.

Concept Stage

User requirements analysis and agreement is part of the concept stage and is critical to the development of successful systems. Without proper understanding of the user needs, any system runs the risk of being built to solve the wrong problems. The first step in the concept stage is to define the user (and stakeholder) requirements and constraints. A key part of this process is to establish the feasibility of meeting the user requirements, including technology readiness assessment. As with many SE activities this is often done iteratively, and stakeholder needs and requirements are revisited as new information becomes available.

A recent study by the National Research Council

(National Research Council 2008) focused on reducing the development time for US Air Force projects. The report notes that, “simply stated, systems engineering is the translation of a user’s needs into a definition of a system and its architecture through an iterative process that results in an effective system design.” The iterative involvement with stakeholders is critical to the project success.

Except for the first and last decision gates of a project, the gates are performed simultaneously. See Figure 6 below.

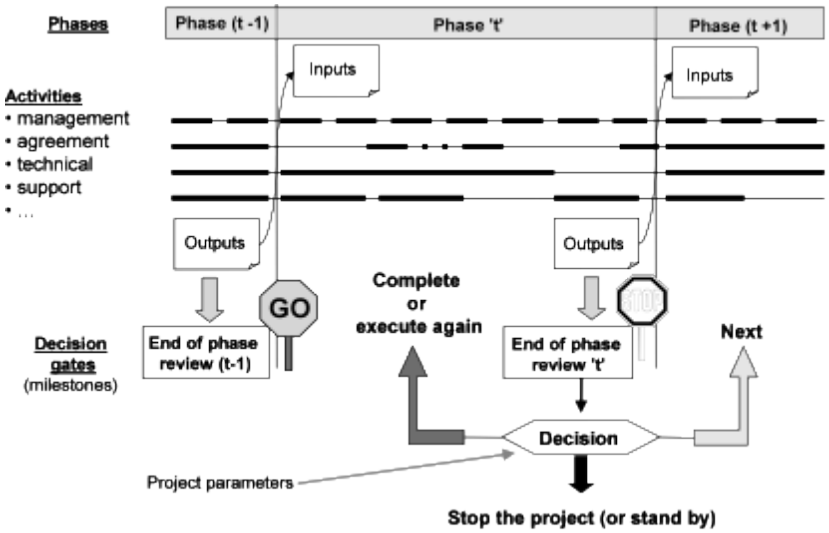

Figure 6. Scheduling the Development Phases. (SEBoK Original)

During the concept stage, alternate concepts are created to determine the best approach to meet stakeholder needs. By envisioning alternatives and creating models, including appropriate prototypes, stakeholder needs will be clarified and the driving issues highlighted. This may lead to an incremental or evolutionary approach to system development. Several different concepts may be explored in parallel.

Development Stage

The selected concept(s) identified in the concept stage are elaborated in detail down to the lowest level to produce the solution that meets the stakeholder requirements. Throughout this stage, it is vital to continue with user involvement through in-process validation (the upward arrow on the Vee models). On hardware, this is done with frequent program reviews and a customer resident representative(s) (if appropriate). In agile development, the practice is to have the customer representative integrated into the

development team.

Production Stage

The production stage is where the SoI is built or manufactured. Product modifications may be required to resolve production problems, to reduce production costs, or to enhance product or SoI capabilities. Any of these modifications may influence system requirements and may require system re-qualification, re-verification, or re-validation. All such changes require SE assessment before changes are approved.

Utilization Stage

A significant aspect of product life cycle management is the provisioning of supporting systems which are vital in sustaining operation of the product. While the supplied product or service may be seen as the narrow system-of-interest (NSOI) for an acquirer, the acquirer also must incorporate the supporting systems into a wider system-of-interest (WSOI). These supporting systems should be seen as system assets that, when needed, are activated in response to a situation that has emerged in respect to the operation of the NSOI. The collective name for the set of supporting systems is the integrated logistics support (ILS) system.

It is vital to have a holistic view when defining, producing, and operating system products and services. In Figure 7, the relationship between system design and development and the ILS requirements is portrayed.

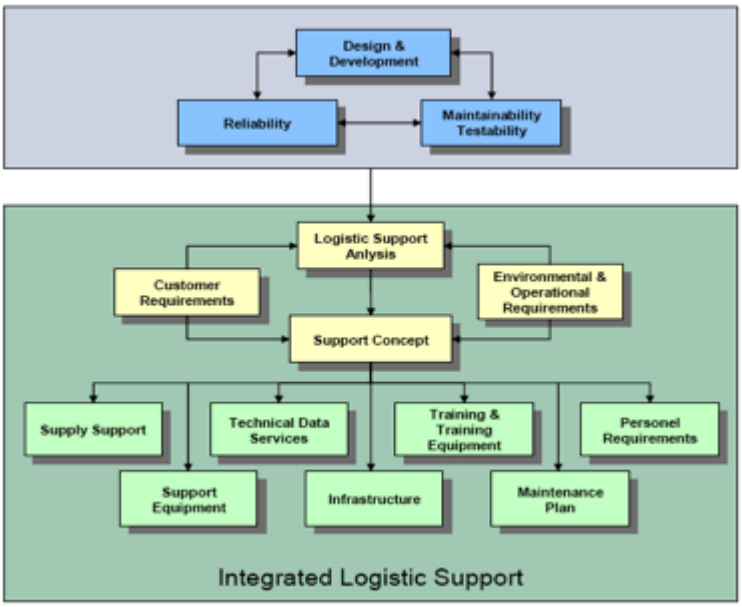

Figure 7. Relating ILS to the System Life Cycle (Eichmueller and Foreman 2009). Reprinted with permission of of ASD/AIA S3000L Steering Committee. All other rights are reserved by the copyright owner.

The requirements for reliability, resulting in the need of maintainability and testability, are driving factors.

Support Stage

In the support stage, the SoI is provided services that enable continued operation. Modifications may be proposed to resolve supportability problems, to reduce operational costs, or to extend the life of a system. These changes require SE assessment to avoid loss of system capabilities while under operation. The corresponding technical process is the maintenance process.

Retirement Stage

In the retirement stage, the SoI and its related services are removed from operation. SE activities in this stage are primarily focused on ensuring that disposal requirements are satisfied. In fact, planning for disposal is part of the system definition during the concept stage. Experiences in the 20th century repeatedly demonstrated the consequences when system retirement and disposal was not considered from the outset. Early in the 21st century, many countries have changed their laws to hold the creator of a SoI accountable for proper end-of-life disposal of the system.

Life Cycle Reviews

To control the progress of a project, different types of reviews are planned. The most commonly used are listed as follows, although the names are not universal:

- The **system requirements review** (SRR) is planned to verify and validate the set of system requirements before starting the detailed design activities.
- The preliminary design review (PDR) is planned to verify and validate the set of system requirements, the design artifacts, and justification elements at the end of the first engineering loop (also known as the "design-to" gate).
- The critical design review (CDR) is planned to verify and validate the set of system requirements, the design artifacts, and justification elements at the end of the last engineering loop (the “build-to” and “code-to” designs are released after this review).
- The integration, verification, and validation reviews are planned as the components are assembled into higher level subsystems and elements. A sequence of reviews is held to ensure that everything integrates properly and that there is objective evidence that all requirements have been met. There should also be an in-process validation that the system, as it is evolving, will meet the stakeholders’ requirements (see Figure 7).
- The final validation review is carried out at the end of the integration phase.
- Other management related reviews can be planned and conducted in order to control the correct progress of work, based on the type of system and the associated risks.

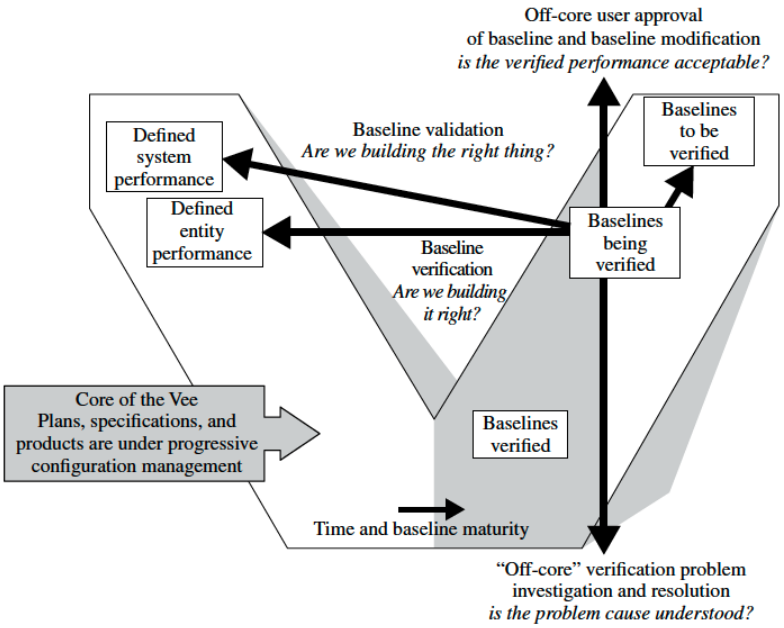

Figure 8. Right Side of the Vee Model (Forsberg, Mooz, and Cotterman 2005). Reprinted with permission of John Wiley & Sons Inc. All other rights are reserved by the copyright owner.

References

Works Cited

Eichmueller, P. and B. Foreman. 2010. *S3000LTM*. Brussels, Belgium: Aerospace and Defence Industries Association of Europe (ASD)/Aerospace Industries Association (AIA).

Faisandier, A. 2012. *Systems Architecture and Design*. Belberaud, France: Sinergy'Com.

Forsberg, K., H. Mooz, and H. Cotterman. 2005. *Visualizing Project Management,* 3rd ed. New York, NY, USA: J. Wiley & Sons.

INCOSE. 2012. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities,* version 3.2.2. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.2.

Lawson, H. 2010. *A Journey Through the Systems Landscape.* London, UK: College Publications, Kings College, UK.

Primary References

Beedle, M., et al. 2009. "The Agile Manifesto: Principles

behind the Agile Manifesto". in *The Agile Manifesto* [database online]. Accessed December 04 2014 at www.agilemanifesto.org/principles.html

Boehm, B. and R. Turner. 2004. *Balancing Agility and Discipline.* New York, NY, USA: Addison-Wesley.

Fairley, R. 2009. *Managing and Leading Software Projects.* New York, NY, USA: J. Wiley & Sons.

Forsberg, K., H. Mooz, and H. Cotterman. 2005. *Visualizing Project Management.* 3rd ed. New York, NY, USA: J. Wiley & Sons.

INCOSE. 2012. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 3.2.2. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.2.

Lawson, H. 2010. *A Journey Through the Systems Landscape.* Kings College, UK: College Publications.

Pew, R., and A. Mavor (eds.) 2007. *Human-System Integration in the System Development Process: A New Look*. Washington, DC, USA: The National Academies Press.

Royce, W.E. 1998. *Software Project Management: A Unified Framework*. New York, NY, USA: Addison Wesley.

Additional References

Anderson, D. 2010. *Kanban*. Sequim, WA, USA: Blue Hole Press.

Baldwin, C. and K. Clark. 2000. *Design Rules: The Power of Modularity.* Cambridge, MA, USA: MIT Press.

Beck, K. 1999. *Extreme Programming Explained*. New York, NY, USA: Addison Wesley.

Beedle, M., et al. 2009. "The Agile Manifesto: Principles behind the Agile Manifesto". in The Agile Manifesto [database online]. Accessed 2010. Available at: www.agilemanifesto.org/principles.html

Biffl, S., A. Aurum, B. Boehm, H. Erdogmus, and P. Gruenbacher (eds.). 2005. *Value-Based Software Engineering*. New York, NY, USA: Springer.

Boehm, B. 1988. “A Spiral Model of Software

Development." IEEE *Computer* 21(5): 61-72.

Boehm, B. 2006. "Some Future Trends and Implications for Systems and Software Engineering Processes." *Systems Engineering.* 9(1): 1-19.

Boehm, B., A. Egyed, J. Kwan, D. Port, A. Shah, and R. Madachy. 1998. "Using the WinWin Spiral Model: A Case Study." IEEE *Computer.* 31(7): 33-44.

Boehm, B., R. Turner, J. Lane, S. Koolmanojwong. 2014 (in press). *Embracing the Spiral Model: Creating Successful Systems with the Incremental Commitment Spiral Model*. Boston, MA, USA: Addison Wesley.

Castellano, D.R. 2004. "Top Five Quality Software Projects." *CrossTalk.* 17(7) (July 2004): 4-19. Available at: http://www.crosstalkonline.org/storage/issue-archives/2004/200407/200407-0-Issue.pdf

Checkland, P. 1981. *Systems Thinking, Systems Practice*. New York, NY, USA: Wiley.

Crosson, S. and B. Boehm. 2009. "Adjusting Software Life cycle Anchorpoints: Lessons Learned in a System of Systems Context." Proceedings of the Systems and Software Technology Conference, 20-23 April 2009, Salt Lake City, UT, USA.

Dingsoyr, T., T. Dyba. and N. Moe (eds.). 2010. "Agile Software Development: Current Research and Future Directions." Chapter in B. Boehm, J. Lane, S. Koolmanjwong, and R. Turner, *Architected Agile Solutions for Software-Reliant Systems.* New York, NY, USA: Springer.

Dorner, D. 1996. *The Logic of Failure*. New York, NY, USA: Basic Books.

Forsberg, K. 1995. "'If I Could Do That, Then I Could...' System Engineering in a Research and Development Environment." Proceedings of the Fifth Annual International Council on Systems Engineering (INCOSE) International Symposium. 22-26 July 1995. St. Louis, MO, USA.

Forsberg, K. 2010. "Projects Don't Begin With Requirements." Proceedings of the IEEE Systems Conference, 5-8 April 2010, San Diego, CA, USA.

Gilb, T. 2005. *Competitive Engineering*. Maryland Heights, MO, USA: Elsevier Butterworth Heinemann.

Goldratt, E. 1984. *The Goal*. Great Barrington, MA, USA: North River Press.

Hitchins, D. 2007. *Systems Engineering: A 21st Century Systems Methodology.* New York, NY, USA: Wiley.

Holland, J. 1998. *Emergence*. New York, NY, USA: Perseus Books.

ISO/IEC. 2010. Systems and Software Engineering, Part 1: Guide for Life Cycle Management. Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC), ISO/IEC 24748-1:2010.

ISO/IEC/IEEE. 2015. *Systems and Software Engineering -- System Life Cycle Processes.* Geneva, Switzerland: International Organisation for Standardisation / International Electrotechnical Commissions / Institute of Electrical and Electronics Engineers. ISO/IEC/IEEE 15288:2015.

ISO/IEC. 2003. *Systems Engineering — A Guide for The Application of ISO/IEC 15288 System Life Cycle Processes*. Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC), ISO/IEC 19760:2003 (E).

Jarzombek, J. 2003. "Top Five Quality Software Projects." *CrossTalk.* 16(7) (July 2003): 4-19. Available at: http://www.crosstalkonline.org/storage/issue-archives/2003/200307/200307-0-Issue.pdf.

Kruchten, P. 1999. *The Rational Unified Process.* New York, NY, USA: Addison Wesley.

Landis, T. R. 2010. *Lockheed Blackbird Family (A-12, YF-12, D-21/M-21 & SR-71).* North Branch, MN, USA: Specialty Press.

Madachy, R. 2008. *Software Process Dynamics*. Hoboken, NJ, USA: Wiley.

Maranzano, J.F., S.A. Rozsypal, G.H. Zimmerman, G.W. Warnken, P.E. Wirth, D.W. Weiss. 2005. "Architecture Reviews: Practice and Experience." IEEE *Software*. 22(2): 34-43.

National Research Council of the National Academies (USA). 2008. *Pre-Milestone A and Early-Phase Systems Engineering*. Washington, DC, USA: The National

Academies Press.

Osterweil, L. 1987. “Software Processes are Software Too.” Proceedings of the SEFM 2011: 9th International Conference on Software Engineering. Monterey, CA, USA.

Poppendeick, M. and T. Poppendeick. 2003. *Lean Software Development: an Agile Toolkit.* New York, NY, USA: Addison Wesley.

Rechtin, E. 1991. *System Architecting: Creating and Building Complex Systems.* Upper Saddle River, NY, USA: Prentice-Hall.

Rechtin, E., and M. Maier. 1997. *The Art of System Architecting*. Boca Raton, FL, USA: CRC Press.

Schwaber, K. and M. Beedle. 2002. *Agile Software Development with Scrum*. Upper Saddle River, NY, USA: Prentice Hall.

Spruill, N. 2002. “Top Five Quality Software Projects.” *CrossTalk.* 15(1) (January 2002): 4-19. Available at: http://www.crosstalkonline.org/storage/issue-archives/2002/200201/200201-0-Issue.pdf.

Stauder, T. 2005. “Top Five Department of Defense Program Awards.” *CrossTalk.* 18(9) (September 2005): 4-13. Available at http://www.crosstalkonline.org/storage/issue-archives/2005/200509/200509-0-Issue.pdf.

Warfield, J. 1976. *Societal Systems: Planning, Policy, and Complexity*. New York, NY, USA: Wiley.

Womack, J. and D. Jones. 1996. *Lean Thinking.* New York, NY, USA: Simon and Schuster.

Relevant Videos

- Basic Introduction of Systems Engineering (V-method) [Part 1 of 2]
- Basic Introduction to Systems Engineering (V-Method) Part 2 of 2

< Previous Article | Parent Article | Next Article >
SEBoK v. 2.6, released 20 May 2022

---

ess_Models:_Vee&oldid=65754"
