# Generic Life Cycle Model

Generic Life Cycle Model



---

***Lead Authors:*** *Kevin Forsberg, Richard Turner,* ***Contributing Author:*** *Rick Adcock*

---

As discussed in the generic life cycle paradigm in System Life Cycle Approaches, each system-of-interest (SoI) has an associated life cycle model. The generic life cycle model below applies to a single SoI. SE must generally be synchronized across a number of tailored instances of such life cycle models to fully satisfy stakeholder needs. More complex life cycle models which address this are described in Life Cycle Models.

☐

## Contents

A Generic System Life Cycle Model
Applying the Life Cycle Model
References
    Works Cited
    Primary References
    Additional References

## A Generic System Life Cycle Model

There is no single “one-size-fits-all” system life cycle model that can provide specific guidance for all project situations. Figure 1, adapted from (Lawson 2010, ISO 2015, and ISO 2010), provides a generic life cycle model that forms a starting point for the most common versions of pre-specified, evolutionary, sequential, opportunistic, and concurrent life cycle processes. The model is defined as a set of stages, within which technical and management activities are performed. The stages are terminated by decision gates, where the key stakeholders decide whether to proceed into the next stage, to remain in the current stage, or to terminate or re-scope related projects.


**Figure 1. A Generic Life Cycle Model.** (SEBoK Original)

Elaborated definitions of these stages are provided in the glossary below and in various other ways in subsequent articles.

The **Concept Definition** stage begins with a decision by a protagonist (individual or organization) to invest resources in a new or improved engineered system. Inception begins with a set of stakeholders agreeing to the need for change to an engineered system context and exploring whether new or modified SoI can be developed, in which the life cycle benefits are worth the investments in the life cycle costs. Activities include: developing the concept of operations and business case; determining the key stakeholders and their desired capabilities; negotiating the stakeholder requirements among the key stakeholders and selecting the system's non-developmental items (NDIs).

The **System Definition** stage begins when the key stakeholders decide that the business needs and stakeholder requirements are sufficiently well defined to justify committing the resources necessary to define a solution options in sufficient detail to answer the life cycle cost question identified in concept definition and provide a basis of system realization if appropriate. Activities include developing system architectures; defining and agreeing upon levels of system requirements; developing systems-level life cycle plans and performing system analysis in order to illustrate the compatibility and feasibility of the resulting system definition. The transition into the system realization stage can lead to either single-pass or multiple-pass development.

It should be noted that the concept and system definition activities above describe activities performed by *systems engineers* when performing *systems engineering*. There is a very strong concurrency between proposing a problem situation or opportunity and describing one or more possible system solutions, as discussed in Systems Approach Applied to Engineered Systems. Other related definition activities include: prototyping or actual development of high-risk items to show evidence of system feasibility; collaboration with business analysts or performing mission effectiveness analyses to provide a viable business case for proceeding into realization; and modifications to realized systems to improve their production, support or utilization. These activities will generally happen through the system life cycle to handle system evolution, especially under multiple-pass development. This is discussed in more detail in the Life Cycle Models knowledge area.

The **System Realization** stage begins when the key stakeholders decide that the SoI architecture and feasibility evidence are sufficiently low-risk to justify committing the resources necessary to develop and sustain the initial operational capability (IOC) or the single-pass development of the full operational capability (FOC). Activities include: construction of the developmental elements; integration of these elements with each other and with the non-developmental item (NDI) elements; verification and validation of the elements and their integration as it proceeds; and preparing for the concurrent production, support, and utilization activities.

The **System Production, Support, and Utilization (PSU)** stages begin when the key stakeholders decide that the SoI life-cycle feasibility and safety are at a sufficiently low-risk level that justifies committing the resources necessary to produce, field, support, and utilize the system over its expected lifetime. The lifetimes of production, support, and utilization are likely to be different. After market support will generally continue after production is complete and users will often continue to use unsupported systems.

**System Production** involves the fabrication of instances or versions of an SoI and of associated after-market spare parts. It also includes production quality monitoring and improvement; product or service acceptance activities; and continuous production process improvement. It may include low-rate initial production (LRIP) to mature the production process or to promote the continued preservation of the production capability for future spikes in demand.

**Systems Support** includes various classes of maintenance: corrective (for defects), adaptive (for interoperability with independently evolving co-dependent systems), and perfective (for enhancement of performance, usability, or other key performance parameters). It also includes hot lines and responders for user or emergency support and the provisioning of needed consumables (gas, water, power, etc.). Its boundaries include some gray areas, such as the boundary between small system enhancements and the development of larger complementary new additions, and the boundary between rework/maintenance of earlier fielded increments in incremental or evolutionary development. *Systems Support* usually continues after *System Production* is terminated.

**System Utilization** includes the use of the SoI in its context by operators, administrators, the general public, or systems above it in the system-of-interest hierarchy. It usually continues after *Systems Support* is terminated.

The **System Retirement** stage is often executed incrementally as system versions or elements become obsolete or are no longer economical to support and therefore undergo disposal or recycling of their content. Increasingly affordable considerations make system re-purposing an attractive alternative.

## Applying the Life Cycle Model

Figure 1 shows just the single-step approach for proceeding through the stages of a SoI life cycle. In the Life Cycle Models knowledge area, we discuss examples of real-world enterprises and their drivers, both technical and organizational. These have led to a number of documented approaches for sequencing the life cycle stages to deal with some of the issues raised. The Life Cycle Models KA summarizes a number of incremental and evolutionary life cycle models, including their main strengths and weaknesses, and also discusses criteria for choosing the best-fit approach.

In Figure 1, we have listed key technical and management activities critical to successful completion of each stage. This is a useful way to illustrate the goals of each stage and gives an indication of how processes align with these stages. This can be important when considering how to plan for resources, milestones, etc. However, it is important to observe that the execution of process activities is not compartmentalized to particular life cycle stages (Lawson 2010). In Applying Life Cycle Processes, we discuss a number of views on the nature of the inter-relationships between process activities within a life cycle model. In general, the technical and management activities are applied in accordance with the principles of concurrency, iteration and recursion described in the generic life cycle paradigm.

# References

## Works Cited

ISO/IEC/IEEE. 2015.*Systems and Software Engineering - System Life Cycle Processes*. Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)/Institute of Electrical and Electronics Engineers. ISO/IEC 15288:2015.

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.

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

## Primary References

Forsberg, K., H. Mooz, H. Cotterman. 2005. *Visualizing Project Management*, 3rd Ed. Hoboken, NJ: J. Wiley & Sons.

INCOSE. 2015. 'Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities', version 4.0. Hoboken, NJ, USA: John Wiley and Sons, Inc, ISBN: 978-1-118-99940-0.

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

## Additional References

None.

---

< Previous Article | Parent Article | Next Article >
**SEBoK v. 2.11, released 25 November 2024**

ldid=72617"