

There are a few industry standards related to process improvement models we should mention briefly. For you as a beginner, it is enough to know they exist. However, if you start working for large corporations, you will find that many will follow one or the other of these standards.

### Capability Maturity Model Integration

The Capability Maturity Model Integration (CMMI) is one of the leading models and based on best practice. Independent assessments grade organizations on how well they follow their defined processes, not on the quality of those processes or the software produced. CMMI has replaced CMM.

### ISO 9000

ISO 9000 describes standards for a formally organized process to manufacture a product and the methods of managing and monitoring progress. Although the standard was originally created for the manufacturing sector, ISO 9000 standards have been applied to software development as well. Like CMMI, certification with ISO 9000 does not guarantee the quality of the end result, only that formalized business processes have been followed.

### ISO 15504

ISO 15504, also known as Software Process Improvement Capability Determination (SPICE), is a "framework for the assessment of software processes". This standard is aimed at setting out a clear model for process comparison. SPICE is used much like CMMI. It models processes to manage, control, guide and monitor software development. This model is then used to measure what a development organization or project team actually does during software development. This information is analyzed to identify weaknesses and drive improvement. It also identifies strengths that can be continued or integrated into common practice for that organization or team.

### External Links

---

- CMMI Official Website (<http://www.sei.cmu.edu/cmmi>)
- Capability Maturity Model ([http://www.dmoz.org/Computers/Programming/Methodologies/Capability\\_Maturity\\_Model/](http://www.dmoz.org/Computers/Programming/Methodologies/Capability_Maturity_Model/)) at DMOZ
- ISO 9000 ([http://www.dmoz.org/Science/Reference/Standards/Individual\\_Standards/ISO/ISO\\_9000/](http://www.dmoz.org/Science/Reference/Standards/Individual_Standards/ISO/ISO_9000/)) at DMOZ
- Introduction to ISO 9000 and ISO 14000 ([http://www.iso.org/iso/iso\\_catalogue/management\\_standards/iso\\_9000\\_iso\\_14000.htm](http://www.iso.org/iso/iso_catalogue/management_standards/iso_9000_iso_14000.htm))
- ISO 15504 News (isospice) (<http://www.isospice.com>)
- Automotive SPICE (<http://www.automotivespice.com/>)

### Life Cycle

---

The **Systems Development Life Cycle (SDLC)**, or *Software Development Life Cycle* in systems engineering, information systems and software engineering, is the process of creating or altering systems, and the models and methodologies that people use to develop these systems. The concept generally refers to computer or information systems.

In software engineering the SDLC concept underpins many kinds of software development methodologies. These methodologies form the framework for planning and controlling the creation of an information system<sup>[1]</sup>: the software development process.

### Overview

---

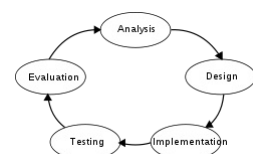
Systems Development Life Cycle (SDLC) is a process used by a systems analyst to develop an information system, including requirements, validation, training, and user (stakeholder) ownership. Any SDLC should result in a high quality system that meets or exceeds customer expectations, reaches completion within time and cost estimates, works effectively and efficiently in the current and planned Information Technology infrastructure, and is inexpensive to maintain and cost-effective to enhance.<sup>[2]</sup>

Computer systems are complex and often (especially with the recent rise of Service-Oriented Architecture) link multiple traditional systems potentially supplied by different software vendors. To manage this level of complexity, a number of SDLC models have been created: "waterfall"; "fountain"; "spiral"; "build and fix"; "rapid prototyping"; "incremental"; and "synchronize and stabilize".<sup>[3]</sup>

SDLC models can be described along a spectrum of agile to iterative to sequential. Agile methodologies, such as XP and Scrum, focus on light-weight processes which allow for rapid changes along the development cycle. Iterative methodologies, such as Rational Unified Process and Dynamic Systems Development Method, focus on limited project scopes and expanding or improving products by multiple iterations. Sequential or big-design-up-front (BDUF) models, such as Waterfall, focus on complete and correct planning to guide large projects and risks to successful and predictable results<sup>[citation needed]</sup>. Other models, such as Anamorphic Development, tend to focus on a form of development that is guided by project scope and adaptive iterations of feature development.



Model of the Systems Development Life Cycle



Model of the Systems Development Life Cycle