Software Development Life Cycle

by Sophia

WHAT'S COVERED

When someone has an idea for a new function to be performed by a computer, how does that idea become reality? If a company wants to implement a new business process and needs new hardware or software to support it, how does it go about making it happen? Recall that software is created via programming, the process of creating a set of logical instructions for a digital device to follow using a programming language. While programming is critical to software development, it only represents one phase of the overall process. In this tutorial, we will discuss the software development life cycle, and the tools used to support the software development process.

Our discussion will break down as follows:

1. Software Development Life Cycle

The process of developing good software is usually not as simple as sitting down and writing some code. True, sometimes a programmer can quickly write a short program to solve a need. But most of the time, the creation of software is a resource-intensive process that often times involves several different groups of people in an organization. For this reason, various software development methodologies exist. The term **software development life cycle (SDLC)** is used to describe the process steps employed by each methodology to develop and deploy software. The table below lists several of the most utilized methodologies. The methodologies will be covered in detail in a later tutorial.

SDLC Methodology	Description	Pros	Cons
Rapid Application Development	Process that focuses on quickly building a working model of the software, getting feedback from users	• Most projects are completed on time • Businesses and project managers can better control risks • More quality applications	• Potential for poor design to occur • Projects cannot be scaled • Utilizes a great deal of resources
		• Encourages	

Agile	Process that utilizes incremental changes with a focus on quality and attention to detail	collaboration across the team • Encourages clients to get involved with the development process • Easier to manage costs of development	• Costs are less predictable • Does not work well for small design projects • Relies heavily on client input
Lean	Process that focuses on taking an initial idea and developing a multiple minimum viable product (MVP)	• Eliminates wasted time and resources • Encourages development team members to make decisions	• Relies too much on organized team • Relies on advanced technical skills
Cascade	Linear method in which development occurs sequentially through phases	• Saves time; easy for project managers • Easy to stay on budget • Testing is made easy	• Long delivery time • Owners need a way to contribute • Not scalable
Prototyping	Process that seeks to develop multiple prototypes of the software program being developed	• Easy to refine data • Feedback is provided at every stage • Money and time saved during the development process	• Deliverable can be altered due to excessive client interaction • Costs • The potential for changes to the product can occur at every phase of development

WATCH

Learn about the SDLC methodologies in the video below.

TERM TO KNOW

Software Development Life Cycle (SDLC)

The general process steps used to develop and deploy software.

2. Phases of the Software Development Life

Cycle

Regardless of the specific development methodology to be used, the software development life cycle (SLDC) provides the general process steps, or phases, for software developers to follow in the development of software. Each phase is characterized by a specific deliverable that should be developed by the end of the phase. The table below lists the general phases of the software development life cycle.

Software Development Life Cycle Phase	Description
Requirement Analysis	--In this phase, a review is done of the request. Is creating a solution possible? What alternatives exist? What is currently being done about it? Is this project a good fit for our organization? --A key part of this step is a feasibility analysis, which includes an analysis of the technical feasibility (is it possible to create this?), the economic feasibility (can we afford to do this?), and the legal feasibility (are we allowed to do this?). --This step is important in determining if the project should even get started. --In this phase, one or more system analysts work with different stakeholder groups to determine the specific requirements for the new system. --No programming is done in this step. Instead, procedures are documented, key players are interviewed, and data requirements are developed in order to get an overall picture of exactly what the system is supposed to do. --The result of this phase is a system-requirements document.
System Design	--In this phase, a designer takes the system-requirements document created in the previous phase, and develops the specific technical details required for the system. --It is in this phase that the business requirements are translated into specific technical requirements. --This document will have everything a programmer will need to actually create the system.
Programming	--The code finally gets written in the programming phase. Using the system-design document as a guide, a programmer (or team of programmers) develops the program. --The result of this phase is an initial working program that meets the requirements laid out in the system-analysis phase and the design developed in the system-design phase.
Testing	--In the testing phase, the software program developed in the previous phase is put through a series of structured tests. --The first is a unit test, which tests individual parts of the code for errors or bugs. --Next is a system test, where the different components of the system are tested to ensure that they work together properly. --Finally, the user-acceptance test allows those that will be using the software to test the system to ensure that it meets their standards. --Any bugs, errors, or problems found during testing are addressed and then tested again.
Deployment	--Once the new system is developed and tested, it has to be implemented in the organization. --This phase includes training the users, providing documentation, and conversion from any

	previous system to the new system. --This is typically the final stage of initial development, in which the system in put into production for the organization/user to use.
Maintenance	--This phase takes place after deployment. --In this phase, the system has a structured support process in place: reported bugs are fixed and requests for new features are evaluated and implemented; system updates and backups are performed on a regular basis. --This typically involves a continuous evaluation of the software's performance.

3. Software Development Management Tools

Regardless of which methodology is utilized, the software development life cycle is an incredibly complex process. Various management tools exist to aid in the overall management of the software development process. Listed below are several software development management tools.

Management Tool	Description
MS Project	Project management tool that aids in overall planning of software project, disseminating project resources, and managing budgets and employee workloads. Can also be used to create project budget.
Visio	Software tool that allows for the creation of charts, diagrams, flowcharts, and network diagrams. Can be used to map out a process.
Visual Studio	Integrated development environment used for writing computer programs, developing websites, and web applications. Supports a wide array of programming languages such as C, C++, C#, Visual basic.net, etc.
Zoho Project	Free online project management tool that allows users to plan, collaborate, and track the progress of large projects by breaking them down into smaller tasks.
Primavera	Software development project management tool that provides users with the ability to collaborate on a development project while also offering control over each defined phase of the project. For example, a project manager can control who has access to certain aspects of a project. Furthermore, Primavera can be easily integrated into a business's enterprise software/information system, such as Oracle's and SAP's enterprise resource planning system.

SUMMARY

Software development is a large and often complex process through which multiple people collaboratively work. Due to the complexity, businesses and organizations wishing to develop applications often have a difficult time settling on a particular **software development life cycle** methodology, as it can be very challenging to implement a particular SDLC based on the scope of the project. To accomplish the task of developing software, various types of tools are available to manage the process. In this tutorial, we discussed what the software development life cycle is, and the **tools** used to aid in managing the process.

Source: Derived from Chapter 10 of “Information Systems for Business and Beyond” by David T. Bourgeois. Some sections removed for brevity. https://www.saylor.org/site/textbooks/Information%20Systems%20for%20Business%20and%20Beyond/Textbook.html

TERMS TO KNOW

Software Development Life Cycle (SDLC)
The general process of steps used to develop and deploy software.