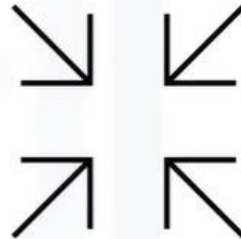


What is the Software Development Life Cycle?

Software development life cycle (SDLC)

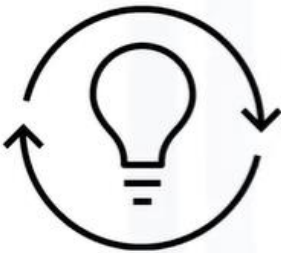
- Systematic process to develop high-quality software
- Aims to produce software that meets requirements
- Defined phases with their own processes and deliverables
- Cycle of planning, design, and development
- Minimizes development risks and costs



IBM Developer

SKILLS NETWORK 

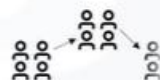
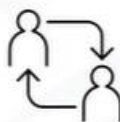
History of the SDLC



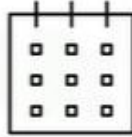
- Conceived of in the mid-1960s
- A deliberate approach needed to manage complex projects
- Initially used the waterfall method
- Adapted to use iterative methods

Advantages of the SDLC (1 of 2)

- Improves efficiency and reduce risks
- Team members know what they should be working on and when
- Facilitates communication among stakeholders
- Team members know when development can move to the next phase



Advantages of the SDLC (2 of 2)



- Respond to changing requirements
- Solve problems early in the process
- Reduces overlapping responsibilities

Phases of the SDLC



The original SDLC was conceived as a traditional waterfall method where the phases are linear, but have since been adapted to introduce iteration so that shifting requirements can be accommodated.

Phase 1: Planning



As part of the planning process, labor and material costs are estimated and weighed against time constraints. Also, project teams are identified, and roles of each team member are proposed. If stakeholders are struggling to define requirements, often the development team may produce prototypes during the planning stage to tease out those requirements. A prototype is a small-scale replica of the end product used to get stakeholder feedback and establish requirements. A prototype is used to test basic design ideas. Though prototyping usually occurs during the planning stage, prototyping can occur at various phases of the SDLC whenever requirements need to be reconsidered or clarified as the project develops.

Prototyping



- Small-scale replica to clarify requirements
- Tests design ideas
- Can be developed at various stages of the SDLC

After requirements have been gathered, they are combined into a document called a software requirements specification, or SRS, document. The SRS needs to be clearly understood and approved by all stakeholders. The developers are also involved at this stage so they can gain a clear understanding of these requirements. Requirements and the SRS will be discussed in more detail in a later video. In the design phase, the requirements gathered from the SRS are used to

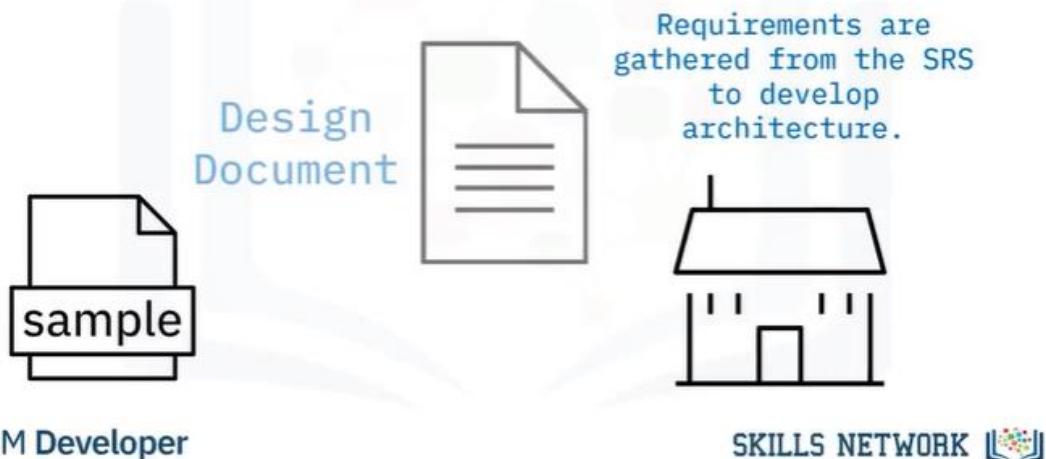
develop the software architecture.

Software requirements specification



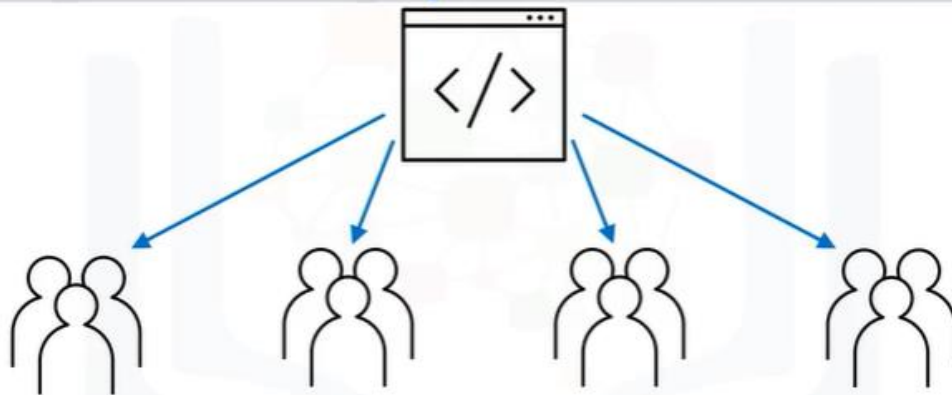
- Requirements are documented in the SRS
- All stakeholders must agree

Phase 2: Design



The document created in this phase is called a design document, and is used by developers during the next phase, which is the development phase. The development phase, sometimes called the “building” phase or the “implementation” phase, is when the developers start the coding process once the design document is completed. The project planners use the design document to determine and assign coding tasks. This phase often requires the use of programming tools, different programming languages, and software stacks. Organizations may also have standards or guidelines that need to be followed.

Phase 3: Development



The testing phase is next in the process once the coding is complete. And some large projects have dedicated testing teams. Code needs to be thoroughly tested to ensure it is stable, secure, and meets the requirements outlined in the SRS. Testing can be manual, automated, or a hybrid of both. Product bugs are reported, tracked, and fixed, and code is retested until the software is stable. Some common levels of testing include unit testing, integration testing, system testing, and acceptance testing. Each of these testing levels will be described in further detail in another video. The deployment phase is where the application is released into the production environment and made available to users. This can also happen in stages— first, it is released onto a user acceptance testing, also called UAT, platform and once the customer signs off on the functionality, it is released to production. This approach can be used for making software available on a website, mobile device app store, or a software distribution server on a corporate network.

Phase 4: Testing



Code is tested to ensure stability, security, and that it meets requirements from the SRS



Bugs reported, tracked, fixed, and retested

Phase 5: Deployment



Phase 6: Maintenance



This phase helps to find any other bugs, identify user interface issues, or UI for short, and identify other requirements that may not have been listed in the SRS. Code enhancements can also be identified at this stage. If bugs are discovered in this phase that were missed during testing, these errors may need to be fixed for high-priority issues or incorporated into the requirements as part of a future software release and the process can start over again.