

Software Development Life Cycle (SDLC): A Structured Approach to Building Reliable Software Systems

Vivek Sharma*

Department of Computer Science and Engineering, Indian Institute of Technology Delhi, India

Editorial

Received: 01-Dec-2025, Manuscript No. JET-25-187988; **Editor assigned:** 04-Dec-2025, Pre-QC No. JET-25-187988 (PQ); **Reviewed:** 22-Dec-2025, QC No. JET-25-187988; **Revised:** 26-Dec-2025, Manuscript No. JET-25-187988 (R); **Published:** 31-Dec-2025, DOI: 10.4172/jet.14.018

*For Correspondence

Vivek Sharma, Department of Computer Science and Engineering, Indian Institute of Technology Delhi, India

E-mail: vivek.sharma@iitd.ac.in

Citation: Vivek Sharma, Software Development Life Cycle (SDLC): A Structured Approach to Building Reliable Software Systems. RRJ Eng Techno. 2025.14.018.

Copyright: © 2025 Vivek Sharma, this is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

ABSTRACT

The Software Development Life Cycle (SDLC) is a systematic process used to design, develop, test, and deploy software applications. It provides a structured framework that ensures high-quality software development while minimizing risks and costs. SDLC encompasses various phases, including planning, analysis, design, implementation, testing, and maintenance. Different SDLC models, such as Waterfall, Agile, and Spiral, offer flexible approaches to managing software projects based on requirements and complexity. This article explores the concepts, phases, models, applications, advantages, and challenges of SDLC. It highlights the importance of adopting appropriate methodologies and tools to improve efficiency, collaboration, and product quality. The integration of modern practices such as DevOps and automation is also discussed as a key factor in enhancing the SDLC process.

Keywords

SDLC, Software Development, Agile, Waterfall Model, Devops, Software Engineering, Project Management

INTRODUCTION

The Software Development Life Cycle (SDLC) is a framework that defines the steps involved in developing software systems. It provides a structured approach to ensure that software is developed efficiently, meets user requirements, and maintains high quality. SDLC is widely used in software engineering to manage complex projects and ensure consistency in development processes.

The increasing complexity of software systems and the demand for rapid development have made SDLC an essential part of modern software engineering. By following a defined process, organizations can reduce errors, improve produc-

tivity, and deliver reliable software products. SDLC is not a single methodology but a collection of models and practices that guide the development process. These models vary in their approach and flexibility, allowing organizations to choose the most suitable one based on project requirements ^[1].

PHASES OF SOFTWARE DEVELOPMENT LIFE CYCLE

The SDLC consists of several phases that guide the development process from initial concept to final deployment. The first phase is planning, where project goals, scope, and feasibility are defined. This phase involves identifying requirements, resources, and timelines. The analysis phase focuses on gathering detailed requirements from stakeholders and understanding system needs. This information is used to create a clear specification for the software. The design phase involves creating the architecture and design of the system. This includes defining system components, data structures, and interfaces.

The implementation phase, also known as coding, is where the actual development of the software takes place. Developers write code based on the design specifications. The testing phase ensures that the software is free from defects and meets quality standards. Various testing methods are used to identify and fix issues. The final phase is maintenance, where the software is updated and improved over time to meet changing requirements and fix any issues that arise ^[2].

SDLC MODELS AND APPROACHES

There are several SDLC models that provide different approaches to software development. The Waterfall model is one of the earliest and simplest models, where each phase is completed before moving to the next. It is suitable for projects with well-defined requirements. The Agile model is a flexible and iterative approach that emphasizes collaboration, continuous feedback, and rapid delivery. It is widely used in modern software development due to its adaptability.

The Spiral model combines elements of both iterative and risk-driven approaches. It focuses on risk assessment and continuous improvement throughout the development process. The V-model is an extension of the Waterfall model that emphasizes testing at each stage of development. It ensures that quality is maintained throughout the process. Each model has its advantages and limitations, and the choice of model depends on factors such as project size, complexity, and requirements ^[3].

APPLICATIONS AND IMPORTANCE OF SDLC

SDLC is used in a wide range of applications, from small software projects to large enterprise systems. It provides a structured approach that helps organizations manage development processes effectively. One of the key benefits of SDLC is improved project management. By defining clear phases and deliverables, it ensures that projects are completed on time and within budget. SDLC also enhances software quality by incorporating testing and validation at various stages. This reduces the risk of defects and improves user satisfaction.

Another important aspect is documentation, which ensures that all aspects of the software are well-documented and easy to maintain. This is particularly important for large and complex systems. SDLC also supports collaboration among team members, including developers, testers, and stakeholders. This improves communication and ensures that requirements are met ^[4].

ADVANTAGES AND CHALLENGES OF SDLC

Another advantage is the ability to identify and address issues early in the development process. This reduces the cost and effort required to fix problems later. However, SDLC also has some challenges. One of the main challenges is the rigidity of certain models, such as the Waterfall model, which may not be suitable for projects with changing requirements.

Another challenge is the need for skilled professionals and effective communication among team members. Poor communication can lead to misunderstandings and project delays. Additionally, the implementation of SDLC can be time-consuming and may require significant resources, particularly for large projects. The integration of modern practices such as DevOps and automation is helping to address these challenges by improving efficiency and reducing development time ^[5].

CONCLUSION

The Software Development Life Cycle (SDLC) is a fundamental framework in software engineering that ensures the systematic development of high-quality software. By following structured phases and adopting appropriate models, organizations can improve efficiency, reduce risks, and deliver reliable software products. Despite challenges related to flexibility and resource requirements, advancements in technology and methodologies are enhancing the effectiveness of SDLC. As software systems continue to evolve, SDLC will remain a critical tool for managing development processes and achieving successful outcomes.

ACKNOWLEDGEMENT

None.

CONFLICT OF INTEREST

None.

REFERENCES

1. Roadmap Epigenomics Consortium Integrative analysis of 111 reference human epigenomes. Nature.2015.
2. Bird A. Perceptions of epigenetics. Nature.2007.
3. Jones A. Functions of DNA methylation: islands, start sites, gene bodies and beyond. Nature Reviews Genetics.2012.
4. Kundaje A. Integrative analysis of epigenomic data. Nature.2015.
5. Feinberg P. The key role of epigenetics in human disease prevention and mitigation. New England Journal of Medicine.2018.