

# **MODULE 5 : ACQUIRING INFORMATION SYSTEMS AND APPLICATIONS**

## **Approach of SDLC**

# Meet The Group

---



T.M.Nithyashree  
23JUCS116



C. Rajakumari  
23JUCS286



A.shiny  
sandhya  
23JUCS125



S. Madhumitha  
23JUCS171



M cherin  
Christina  
23JUCS317

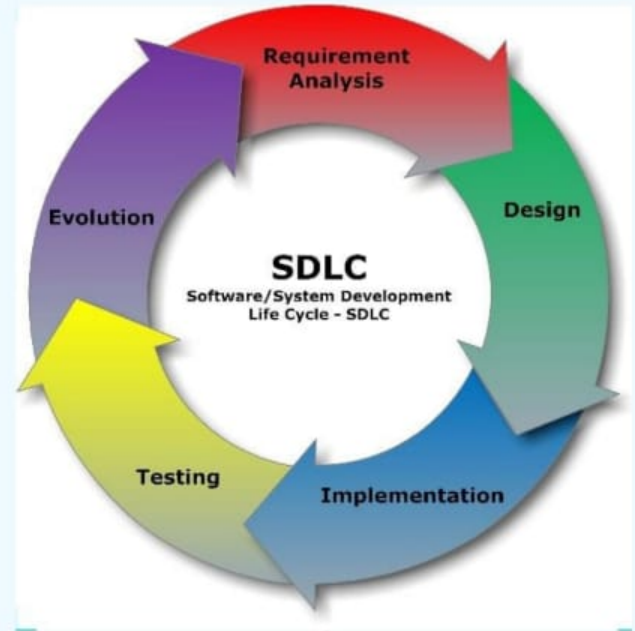


# Index

- 1 Introduction
- 2 Phrases
- 3 Types of model
- 4 Tools and Technology

# Introduction

The Software Development Life Cycle (SDLC) is a systematic process used in software engineering to design, develop, test, and deploy software applications. It typically involves stages such as planning, analysis, design, implementation, testing, deployment, and maintenance. Each phase contributes to ensuring the software meets quality standards and user requirements. The approach aims to enhance efficiency, minimize errors, and deliver a robust, reliable product. Adhering to SDLC methodologies like Agile or Waterfall helps manage projects effectively, fostering collaboration among teams and ensuring a structured progression from concept to final product.







# phrases



- 1 Planning
  - 2 Design
  - 3 Implement
  - 4 Testing & Integration
  - 5 Deployment
  - 6 Maintenance
- 

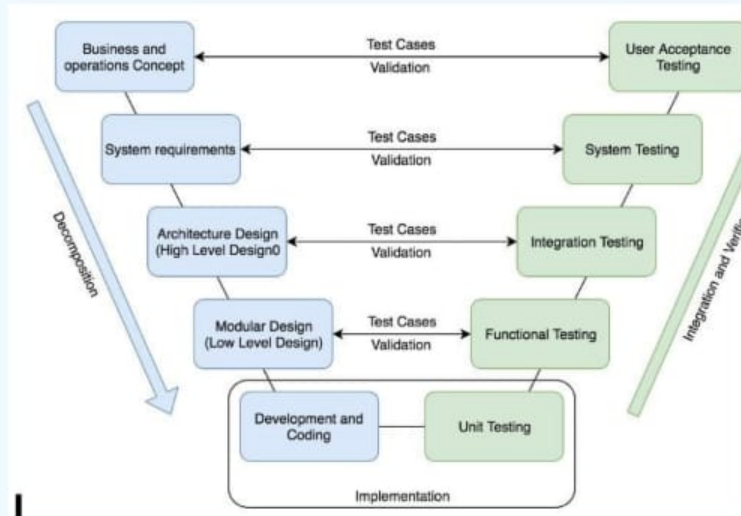
# 1.Planning & analysis:

In Software Development Life Cycle (SDLC), planning and analysis are crucial phases. Planning involves defining project scope, timelines, and resources. Analysis focuses on gathering requirements and understanding user needs for effective system design.

## 2.Design:

In the Software Development Life Cycle (SDLC), the design phase involves creating detailed plans for how the software will be built. This includes system architecture, database design, user interface design, and more. Design is crucial for guiding the development process and ensuring that the final product meets requirements effectively. we design our project is web development so we design the jpr university college website. source code and other details about website is in zipper file

# V-Model

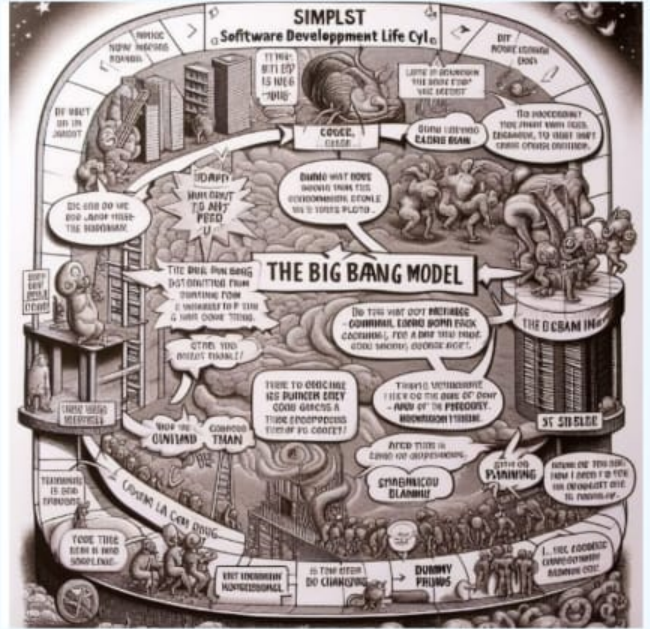


The V-model is a software development model that emphasizes a sequential and phased approach. It consists of stages on the left side (Requirements, Design, Implementation, Testing) and corresponding validation phases on the right side (Unit Testing, Integration Testing, System Testing, Acceptance Testing), forming a V shape. This model highlights the relationship between each development stage and its associated testing phase.



# The Big Bang

The Big Bang model is a software development approach where the entire system is developed at once, without following specific phases or processes. It involves extensive planning and analysis upfront, followed by a simultaneous implementation of all components. Testing is typically done after the completion of the entire system. While it allows for flexibility, it may pose challenges in terms of managing dependencies and identifying issues early in the development process.





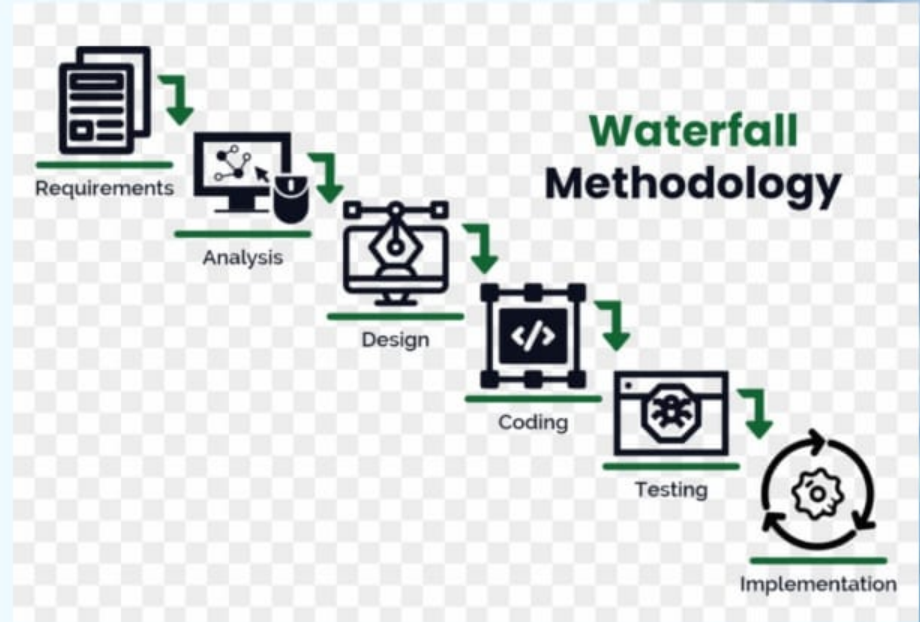
# Agile Model



Agile is an iterative and incremental software development methodology that prioritizes flexibility and collaboration. It involves breaking the development process into small. Teams work on short-term goals, continually adapting and responding to changes in requirements. Agile methodologies, such as Scrum or Kanban, promote customer feedback, collaboration, and the delivery of a minimum viable product (MVP) at the end of each iteration, allowing for quick adjustments based on user input.

# Waterfall

1. Requirements: Gather and document the project's requirements in detail.
2. Design: Create a system design based on the gathered requirements.
3. Implementation: Code the system based on the design specifications.
4. Testing: Conduct various levels of testing to ensure the software meets requirements.
5. Deployment: Deploy the software for use in the intended environment.
6. Maintenance: Address issues, make updates, and provide ongoing support.





# Iterative model



1. Planning: Define the overall project goals and plan the first iteration.
2. Design: Develop a partial system design based on the defined goals for the current iteration.
3. Implementation: Code a portion of the system based on the design from the current iteration.
4. Testing: Test the implemented features and gather feedback.
5. Evaluation: Assess the results of the iteration, learn from them, and plan the next iteration.
6. Repeat: Repeat the process, refining and expanding the software in subsequent iterations until the complete system is developed.



# Spiral model

1. Planning: Identify goals, constraints, and alternatives. Plan the next iteration.
2. Risk Analysis: Evaluate potential risks and uncertainties in the project.
3. Engineering: Develop the product incrementally, starting with the most critical features.
4. Evaluation: Review the results and assess project progress. Decide whether to proceed with the next iteration or terminate the project.
5. Repeat: Repeat the process, gradually expanding the scope and functionality of the software in each iteration.



# Tools & Technology



## 1. *Project Management Tools:*

- Jira: Often used for agile methodologies, Jira is a popular project management tool that helps in planning, tracking, and managing the entire software development process. It facilitates collaboration among team members and ensures a streamlined workflow.

## 2. *Version Control Systems:*

- Git: Git is a widely used distributed version control system. It allows multiple developers to collaborate on a project by managing changes to the source code. Platforms like GitHub or GitLab are often used in conjunction with Git for hosting repositories and enabling collaboration.



**THANK YOU**