

SDLC

Sujata Batra - sujatabatra@hotmail.com

- IT Trainer Since 2000
- More than 50+ Corporate Clients

Definition

The Software Development Life Cycle (SDLC) model is

- An approach to have a linear sequence of steps to develop a system or software product
- To execute the process from start to finish without revisiting any previous step
- One of the oldest systems development models and is still the most commonly used



SDLC

The systems development life cycle (SDLC) is a term used in:

Systems
Engineering



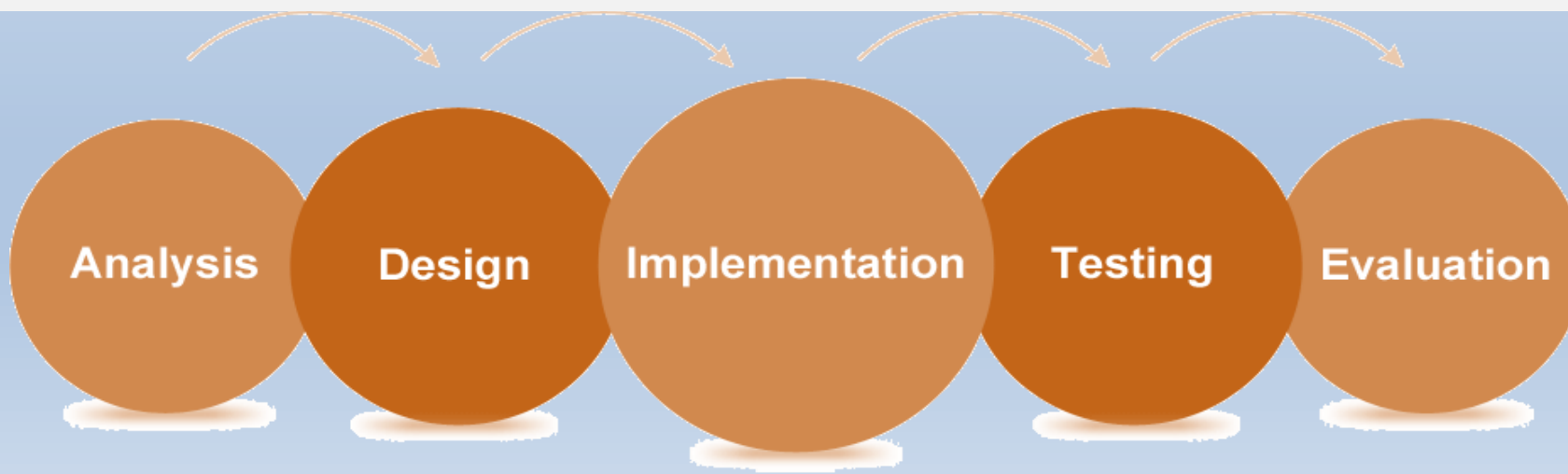
Information
Systems



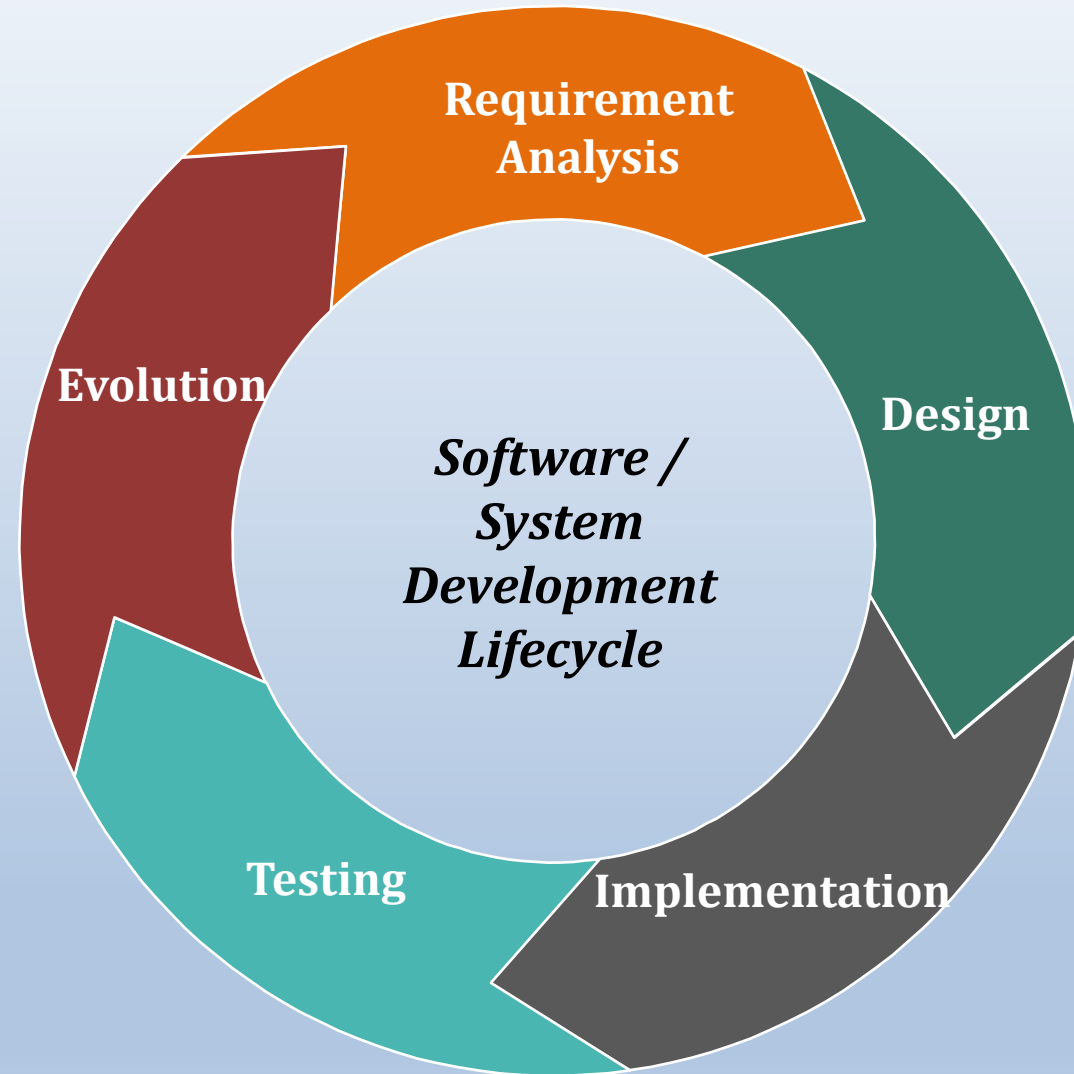
Software
Engineering



Also called application development life-cycle.

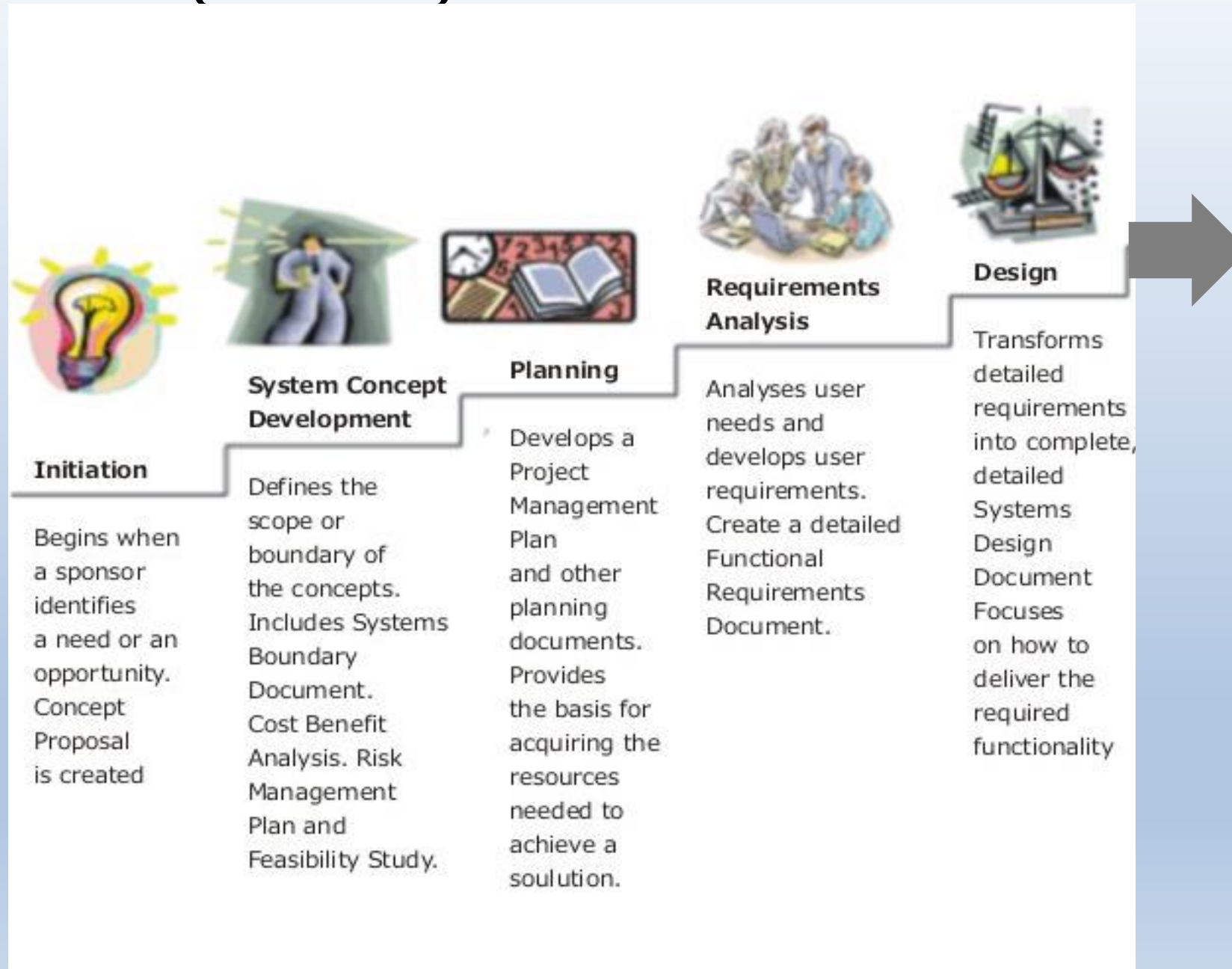


SDLC

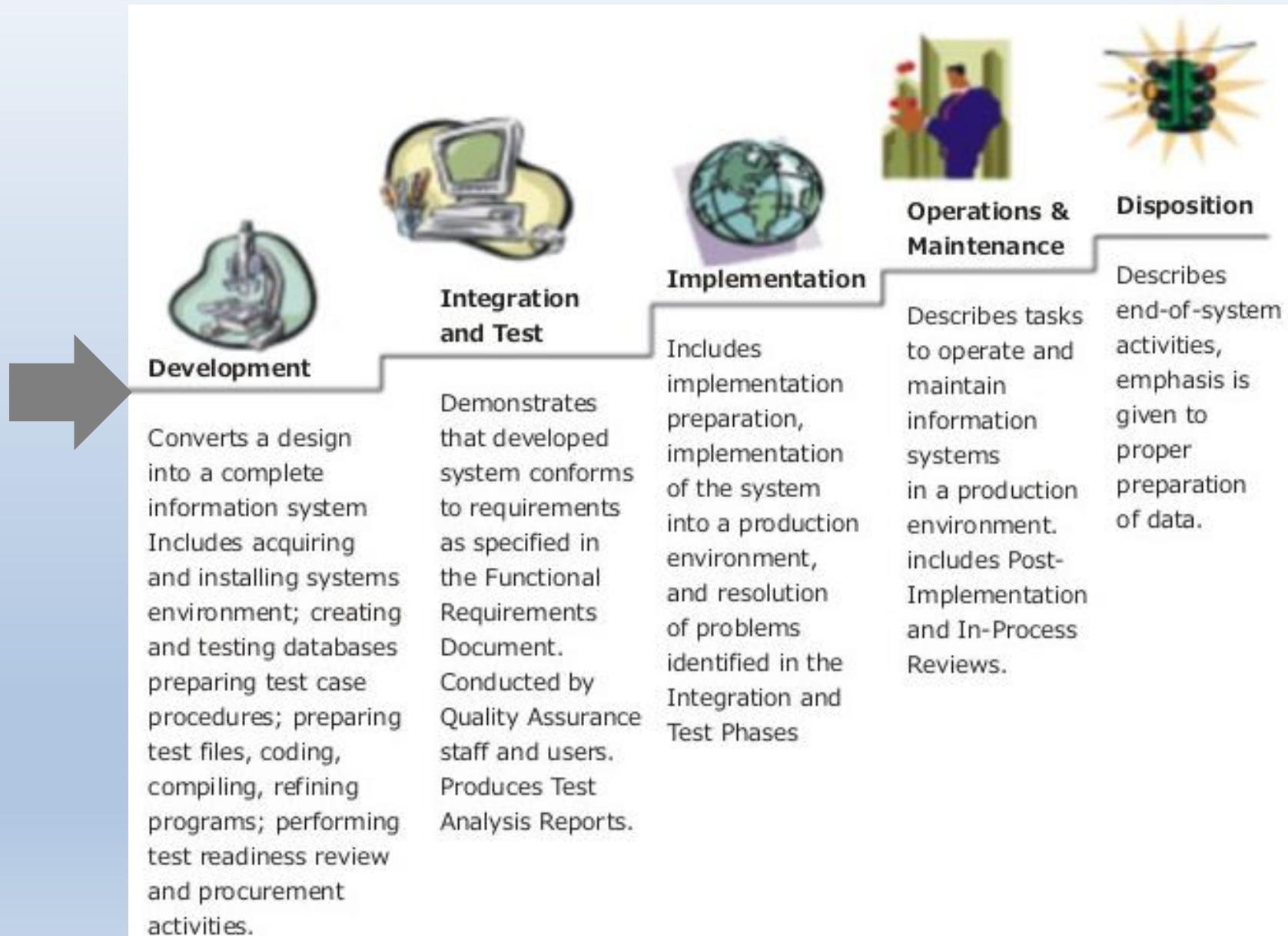


SDLC is the motherboard for all kinds of project developments!

SDLC Phases (Part 1 of 2)



SDLC Phases (Part 2 of 2)



Project Initiation Phase

This is the 1st phase in the Project Life Cycle, as it involves starting up a new project.

A project is started by defining its :

- Objectives
- Scope
- Purpose
- Deliverables

Also in this phase we hire the project team, setup the Project Office and review the project, to gain approval to begin the next phase.

The purpose of the Initiation Phase is to start the project.

Concepts Development Phase

The Concept Development Phase may begin after the approval of the completion of the Initiation project status review, and the approval to proceed to the Concept Development Phase.

The focus of the phase is two-fold:

- 1) Evaluate feasibility of alternatives and
- 2) Clearly define and approve project scope, including the system, all deliverables, and all required activities.

Planning

Project Planning – Determines the project's goals and results in a high-level view of the potential project.

Proper comprehensive project planning is essential to a successful IT project, and incomplete project planning and analysis are frequently root causes of project failure.

The purpose of the Planning Phase is to plan all project processes and activities required to ensure project success and to create a comprehensive set of plans, known as the Project Management Plan (PMP), to manage the project from this phase until project termination.

Requirement Analysis

The Requirements Analysis Phase begins when the previous phase objectives have been achieved.

Documentation related to user requirements from the Concept Development Phase and the Planning Phase shall be used as the basis for further user needs analysis and the development of detailed requirements.

The purpose of the Requirements Analysis Phase is to transform the needs and high-level requirements specified in earlier phases into unambiguous (measurable and testable), traceable, complete, consistent, and stakeholder-approved requirements.

Design

During the Design Phase, the system is designed to satisfy the requirements identified in the previous phases.

The requirements identified in the Requirements Analysis Phase are transformed into a System Design Document that accurately describes the design of the system and that can be used as an input to system development in the next phase.

The purpose of the Design Phase is to transform the requirements into complete and detailed system design specifications. Once the design is approved, the Development Team begins the Development Phase.

Development

The Development Phase features a key step in the project: system construction.

The previous phases lay the foundation for system development; the following phases ensure that the product functions as required.

To complete the Development Phase successfully, two elements are required:

- 1) 1) A complete set of design specifications
- 2) 2) Proper processes, standards, and tools.

The purpose of the Development Phase is to convert the system design prototyped in the Design Phase into a working information system that addresses all documented system requirements. At the end of this phase, the working system will enter the Test Phase.

Testing

The Test Phase focuses on an empirical investigation in which the results describe the quality of the system: testing cannot confirm a system functions properly under all conditions but can establish that it fails under specific conditions.

In the Test Phase, testing of the system proves that the system meets all requirements, including those for performance and security.

The purpose of the Test Phase is to guarantee that the system successfully built and tested in the Development Phase meets all requirements and design parameters. After being tested and accepted, the system moves to the Implementation Phase.

Implementation

The Implementation Phase has one key activity:

Deploying the new system in its target environment. Supporting actions include training end-users and preparing to turn the system over to maintenance personnel.

The purpose of the Implementation Phase is to deploy and enable operations of the new information system in the production environment.

Operation and Maintenance

During the Operations and Maintenance Phase, the information system's availability and performance in executing the work for which it was designed is maintained.

System operations continue until the system's termination date, when the next phase, Disposition, begins.

The purpose of the Operations and Maintenance Phase is to ensure the information system is fully functional and performs optimally until the system reaches its end of life.

Disposition

The Disposition Phase is the end of an information system's life cycle. The information system is formally retired according to organizational needs, laws and regulations, and the Disposition Plan.

The disposition activities ensure that the information system is terminated in an orderly manner and that vital information about the system is preserved according to applicable records management regulations and policies for future access.

The decision to proceed with the Disposition Phase is based on recommendations and approvals from an In-Process Review during the Operations and Maintenance Phase.

The purpose of the Disposition Phase is to shut down the operational information system in a controlled manner.

Why SDLC ?

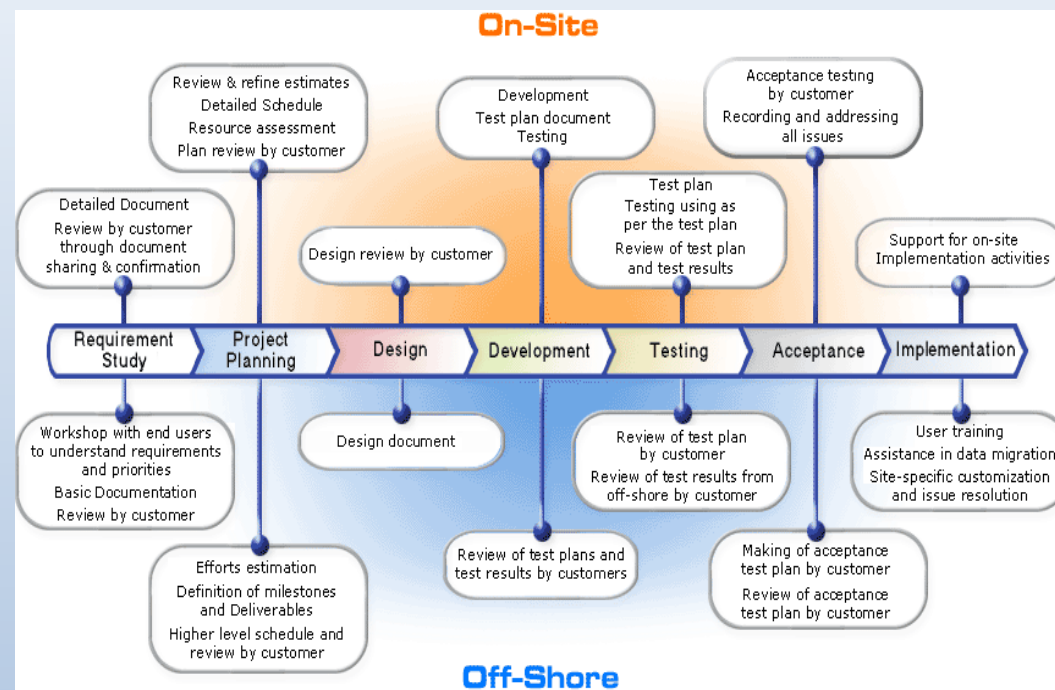
We need to follow SDLC

- To execute projects with proven frame work
- To define and focus roles and responsibilities
- To enforce planning and control
- To have consistency among deliverables
- To increase productivity by executing the project in systematic manner
- To reduce the rework effort during project execution



Types of SDLC

- Waterfall Model
- Prototyping Model
- Incremental Model
- Spiral Model
- V – Model
- Rapid Application Development Model (RAD)

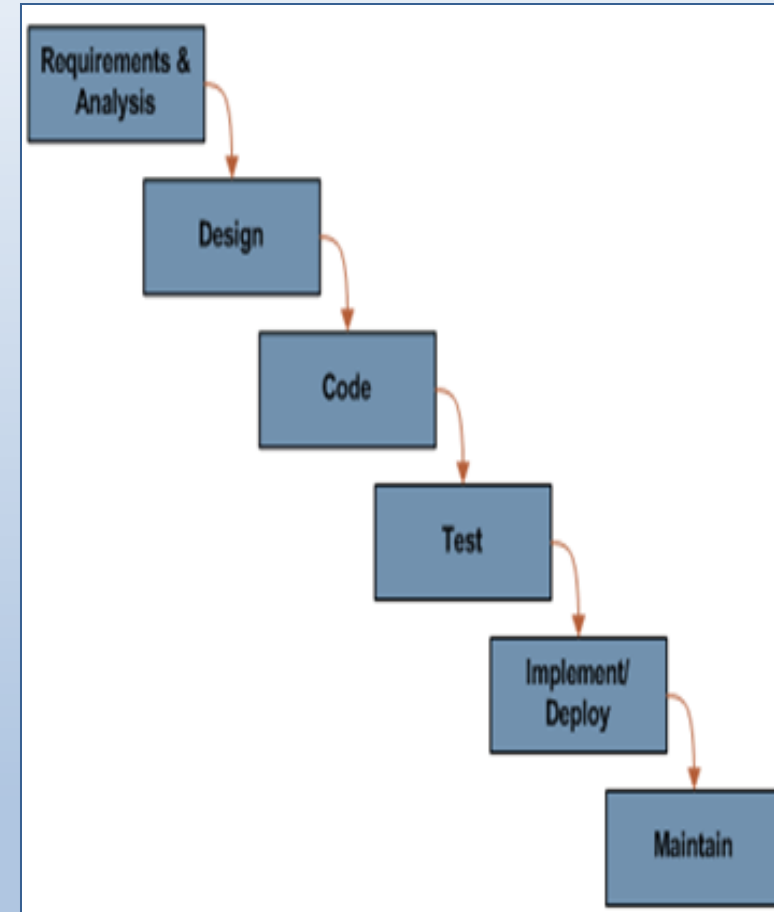


Waterfall Model

Waterfall model is the base Model of SDLC

Main features are:

- Whole process of software development is divided into separate phases
- Derives from its name, giving cascading effect from one phase to another phase
- Each phase has well defined starting and ending point with identifiable deliveries to the next phase
- Most commonly used model



Waterfall Model Advantages

1. The most commonly used model because it is easy to use and understand.
2. No phase is considered to be complete until it is documented and verified
3. Provides means for making structured and stable development process, fostering the creation of high quality deliverables
4. Milestones are well – defined and understood

Waterfall Model Disadvantages

1. Requirements must be fully defined at the beginning itself.
2. It is difficult to get early feedback either on requirements feasibility or implementation approach
3. A working version of the software will not be available until late in the cycle
4. The customer and developer interaction is less during the development of the product

Prototyping Model

Prototyping is the process of quickly putting together a working model.

This model

- Provides proof of concept
- Gives users an idea of what the final system looks like
- Increases the system development speed
- Helps to identify any problems with earlier design
- Enables users to give quicker feedback on the approach
- Is Cost effective
- Is not the final product and suitable especially for the analysis phase



Prototype Model Advantages

1. Sponsors can view steady progress.
2. This model can be used if the requirements change frequently
3. Communication between the developers and customers can be improved
4. Offers more satisfaction to users.
5. Prototype can be used as a marketing tool.

Prototype Model Disadvantages

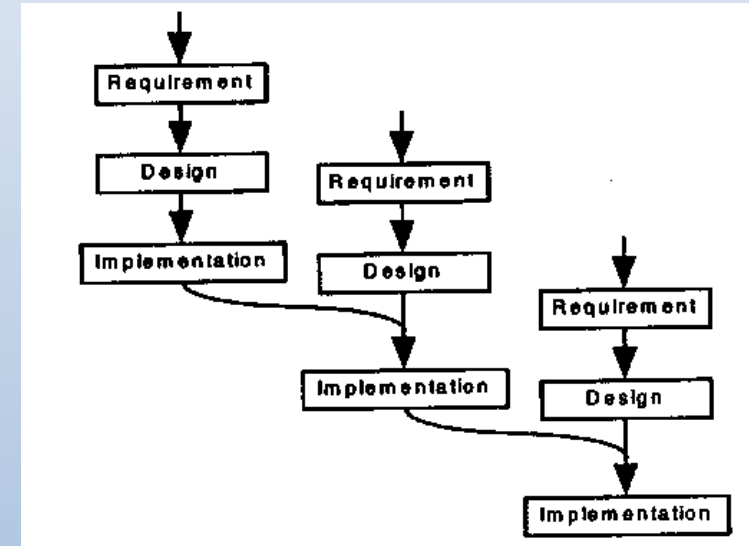
1. Difficult to plan the entire project at once, leading to difficulties and inaccuracies with estimating..
2. May encourage an excess of change requests.
3. User can get too involved whereas the program can not be to a high standard
4. Structure of system can be damaged since many changes could be made
5. Not suitable for large applications

Incremental Model

Incremental development is a scheduling and staging strategy in which the various parts of the system are developed at different times and integrated as they are completed.

Main features of incremental model are

- Increments may be built serially or in parallel
- Each increment adds additional or improved functionality to system
- Requires small group of developers
- Clients can see the system and provide feedback from time to time



Incremental Model Advantages

1. This model gives an opportunity to incorporate user refinements, resulting from experience with earlier releases, into subsequent release
2. At each release an operational product is delivered
3. More flexible - less costly to change scope and requirements.
4. Easier to manage risk because risky pieces are identified and handled during its iteration
5. Generates working software quickly and early during the software life cycle.

Incremental Model Disadvantages

1. Needs good planning, design and careful partitioning of the product
2. Need planned and well – defined interface between increments, especially if they will be developed in parallel.
3. Problems may arise pertaining to system architecture because not all requirements are gathered up front for the entire software life cycle

Spiral Model

Spiral Model is combined approach of prototype and waterfall model.

In this model

- Each phase is originated with alternative specifications and risk analysis
- Strengths are evaluated and the necessary amount of testing is proposed for each prototype
- The above process is iterated until customer is satisfied with that prototype
- Final system is constructed based on the refined prototype



Spiral Model Advantages

1. Flexible and easy to accommodate changes
2. New technologies and architectures can be integrated easily
3. Ability to react to risk at each evolutionary level
4. Each iteration of the spiral can be customized to suit the needs of the project
5. The final product or the software is produced early in the software life cycle

Spiral Model Disadvantages

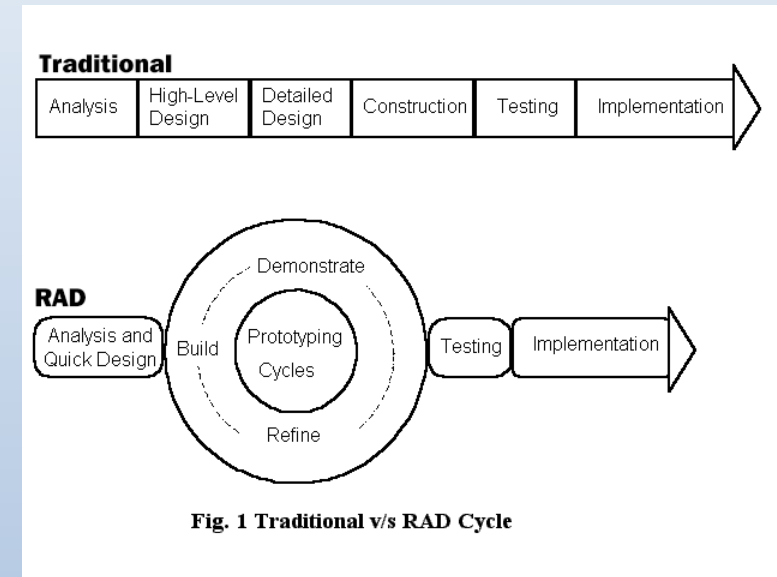
1. More complex planning and management processes
2. Cost may be high
3. It doesn't suit or work well with small projects
4. Highly customized limiting re-usability
5. Risk of not meeting schedule

Rapid Application Development (RAD)

RAD (Rapid Application Development) is a concept with which the products can be developed faster and of higher quality.

The approach focuses on

- Using workshops to gather requirements in fast manner
- Combining the best available techniques in proper sequence to make them effective
- following Prototyping techniques
- Using appropriate tools
- Re-using of software components / modules
- A rigidly paced schedule that defers design improvements to the next product version



RAD Advantages

1. Cycle time is reduced
2. Customer involvement is high throughout the complete cycle
3. Not achieving customer satisfaction can be minimized
4. Very few resources are needed
5. Modeling concepts are used for capturing the business, data and process

RAD Weaknesses

1. Difficult to implement in legacy systems
2. It requires a system that can be modularized
3. Developers & customers must be prepared for rapid-fire activities in an abbreviated time frame.
4. It may be difficult for many important users to commit the time required for success of the RAD process.
5. This method may not be useful for large, unique or highly complex projects