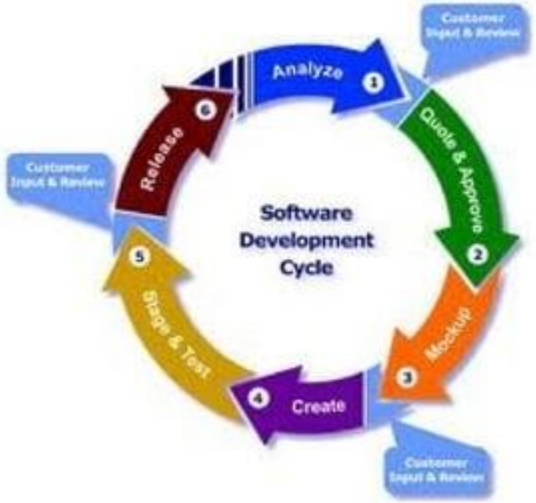
Software Development Life Cycle(SDLC)



- Overview
- Objective
- Development Phases
- Life cycle Model
- Agile SDLC
- Strength & weakness
- conclusion

1.Overview

- It is a **process** used to develop information systems and user ownership
- A **framework** that describes the activities performed at each stage of a software development project
- **High quality** system
- Reaches completion with cost and time
- Have various model like **Waterfall,spiral,RAD,Agile**

2.Objective

Sdlc has Three primary objectives:

- -Ensure the delivery of high quality systems
- -Provide strong management control
- -Maximize productivity

3.Development Phase

The development phase in SDLC are

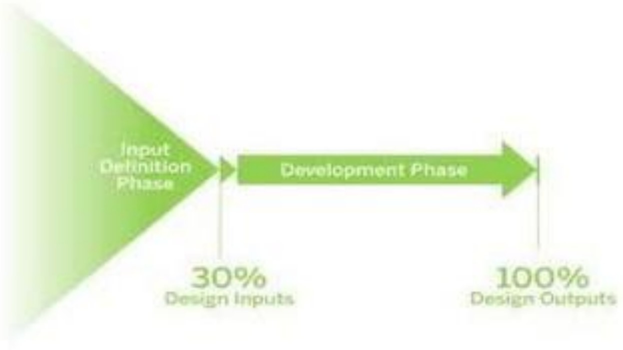
Requirement analysis

Design

Coding

Testing

Operation & Maintainance



Requirement Analysis

- It invovles 'breaking down' the system for
  * analysis of situation
  * analysis of project goals
- It can be done by individuals or team members

Design

- It takes the initial input
- For each requirements design elements will be produced
- It **describes** the software features and includes **hierarchy diagrams,screen layout diagrams**
- The **output** of ths stage describe the new system as a collection of modules or subsytems

Coding

- Modular & subsystem programming code will be accomplished during this stage
- It is interlinked with the testing stage
- Here overall coding will be tested

Testing

- Here the code are tested at various levels
- Most common testing are unit,system and user acceptance.
- Types of testing are

  # White box testing

  # Black box testing

  # Regression testing

Testing(cont.)

Performance testing

Integration testing

Data set testing

Operation & Maintenance

- The deployment includes changes and enhancements
- Maintaining is the important aspect of SDLC

4.Life cycle models

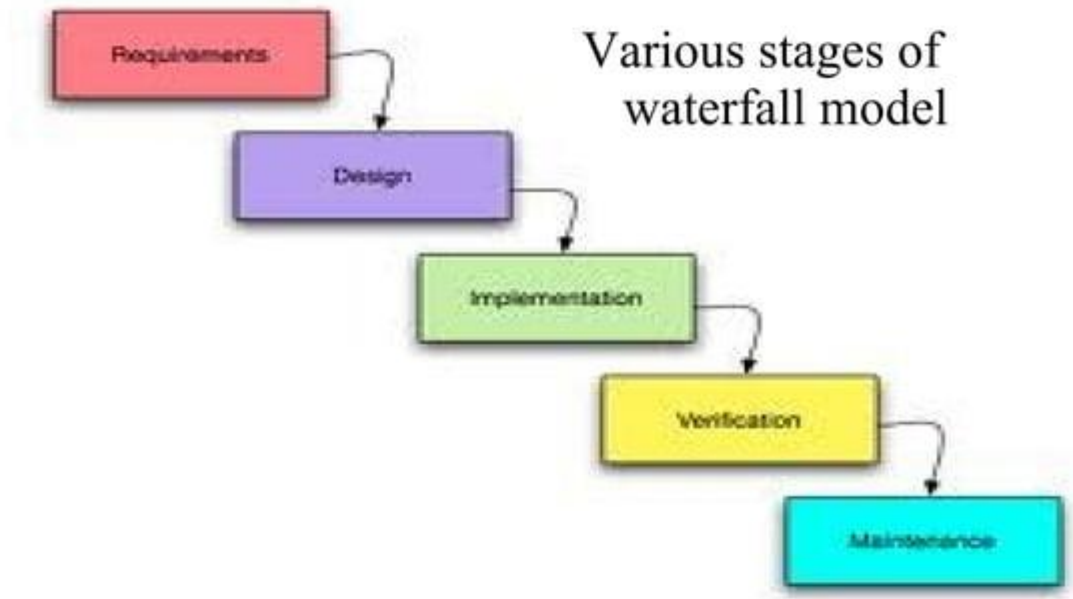
Different types of life cycle model available are

- Waterfall model
- Prototyping model
- Rapid Application Development(RAD)
- Spiral model

5.Waterfall model

- It is the classical system development model
- Requirements-defines needed information,function,behaviour,performance and interface
- Design-data structures,software architedtures,interface representations,algorithmic details
- Implementation-source code,database,documentation,testing

waterfall(cont.)



Various stages of waterfall model

Waterfall(cont.)

Strength

- Minimizes planning overhaed
- Structure minimizes wasted effort
- Works well for technically weak or inexperinced staff

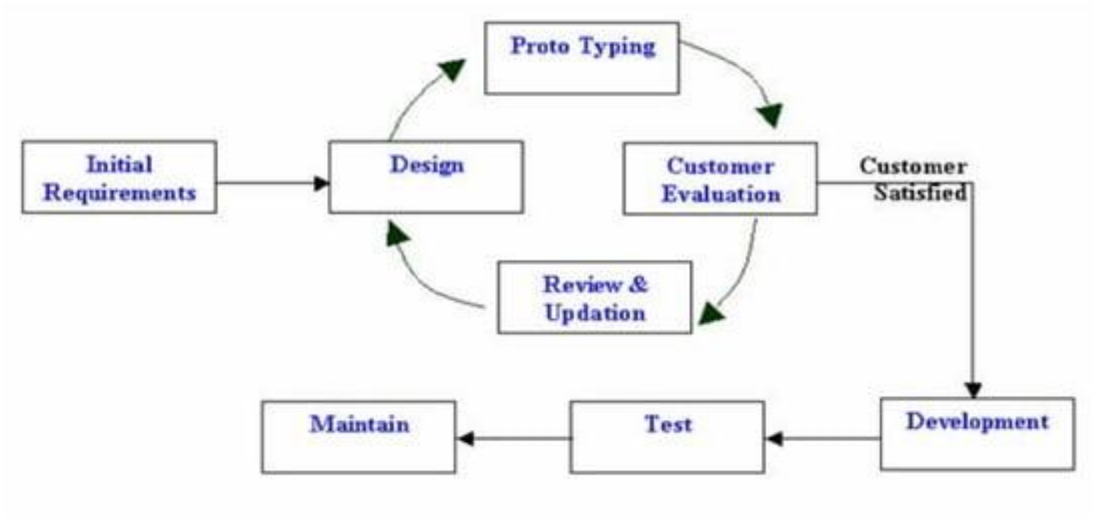
Weakness

- Inflexible
- Only final stage produces documentation
- Backing up to address mistake is difficult

6.Prototyping Model

- It uses multiple iterations or requirement,analysis,design
- After each iteration,the result is evaluted by the customer
- When the user is satisfied,the prototype code is brought up to the standards needed for afinal product.

Prototype(cont.)



Proto Type Model

Prototyping(cont.)

Strength

- Customers can see steady progress
- This is useful when requirements are changing rapidly

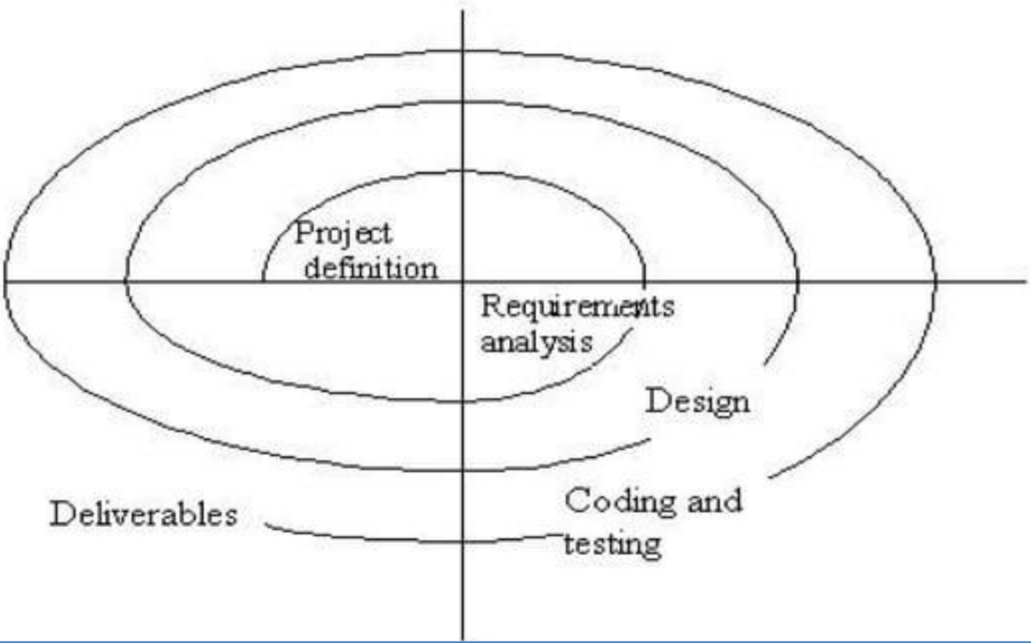
Weakness

- It is impossible to know how long it will take
- There is no way to know the no.of iterations will be required

7.Spiral Model

- It is risk-reduction oriented model
- It breaks the whole projects into mini projects
- For projects with risky elements,its beneficial.
- Each cycle invovles the same sequence as the steps as the waterfall process model

Spiral(cont.)


Project definition
Requirements analysis
Design
Coding and testing
Deliverables

Spiral(cont.)

Strength

- Early iterations of the project are cheapset
- Risk decreases
- All iterations meets the project needs

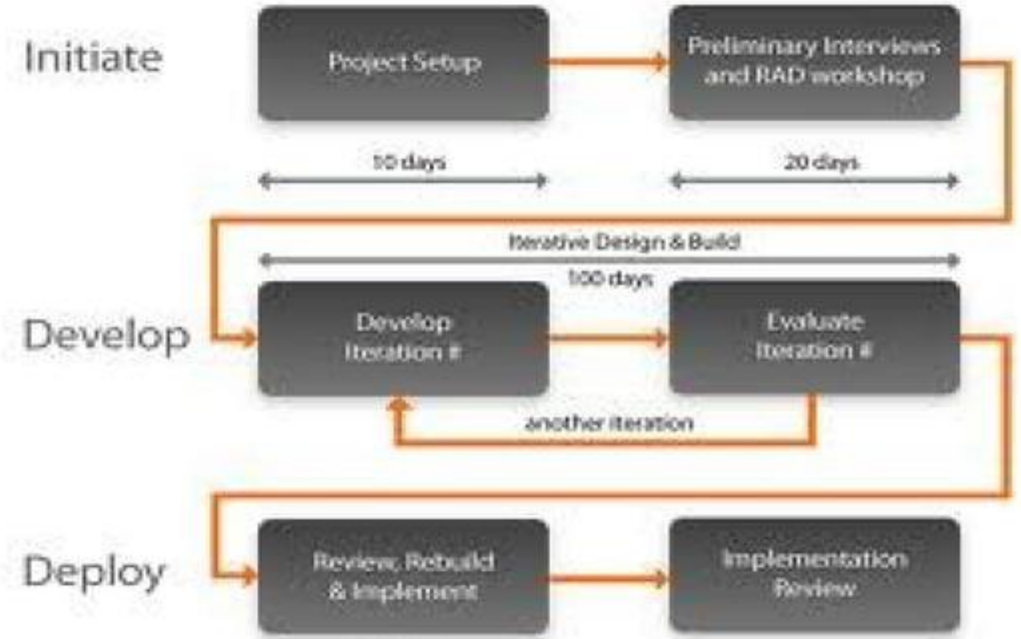
Weakness

- Complicated
- Require attentive & knowledgable management

8.RAD model

- RAD is a concept that products can be developed faster and higher quality through:
  - Gathering requirements using workshops
  - Prototyping and early,reiterative user testing of designs
  - the re-use of software components

RAD(cont.)


Initiate
Project Setup
Preliminary Interviews and RAD workshop
10 days
20 days
Iterative Design & Build
100 days
Develop
Develop Iteration #
Evaluate Iteration #
another iteration
Deploy
Review, Rebuild & Implement
Implementation Review

RAD(cont.)

Strength

- Reduces the development time
- Reusability
- Speed
- Easy to work with

Weakness

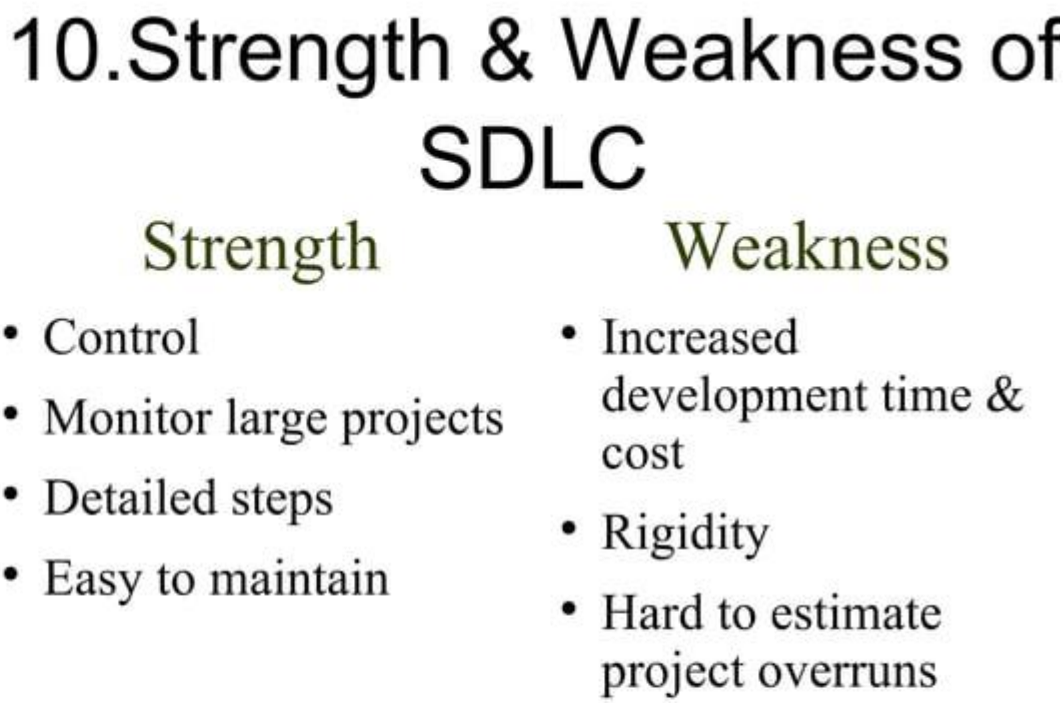
- Require higly skilled engineers
- Both the customer & developer should be commited to complete
- If it is difficult to modularize,its not work well

9.Agile SDLC

- Speed up or bypass on one or more life cycle phases
- Used for time critical application
- Usually less formal and reduced scope
- Used in organizations that employ disciplined methods

Some Agile Methods

- Adaptive software development(ASD)
- Feature driven development(FDD)
- Crystal clear
- Extreme programming(XP)
- Scrum
- RAD



10.Strength & Weakness of SDLC

Strength	Weakness
• Control	• Increased development time & cost
• Monitor large projects	• Rigidity
• Detailed steps	• Hard to estimate project overruns
• Easy to maintain	

My conclusion

- **RAD model** can be used in **mashups** as a life cycle development model because:

  # **Speed** process

  # **customer** can be **involved** upto delivery of projects

  # user **requirements can be added or modified** at any time during the project

conclusion(cont.)

#It **reduces** the development **time**

work can be **modularized**

can **support** multi platform like **PHP**,Python,Perl..

So **RAD** may be the right option to work with **PHP** for **Mashups**