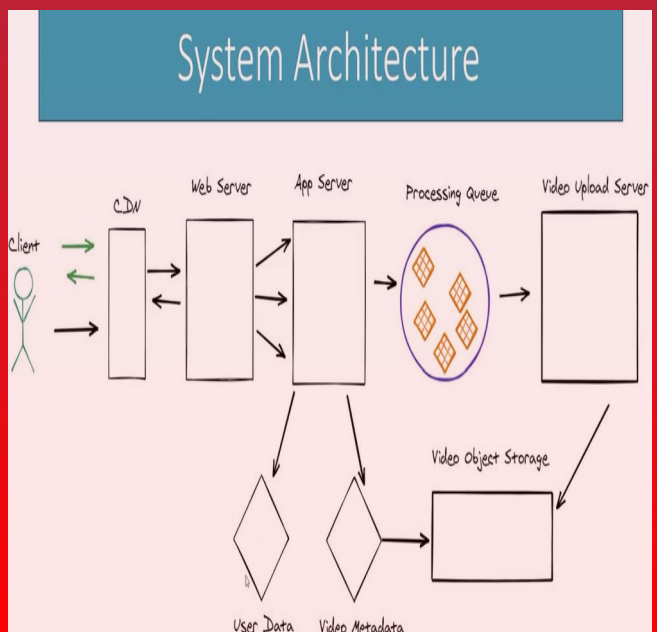
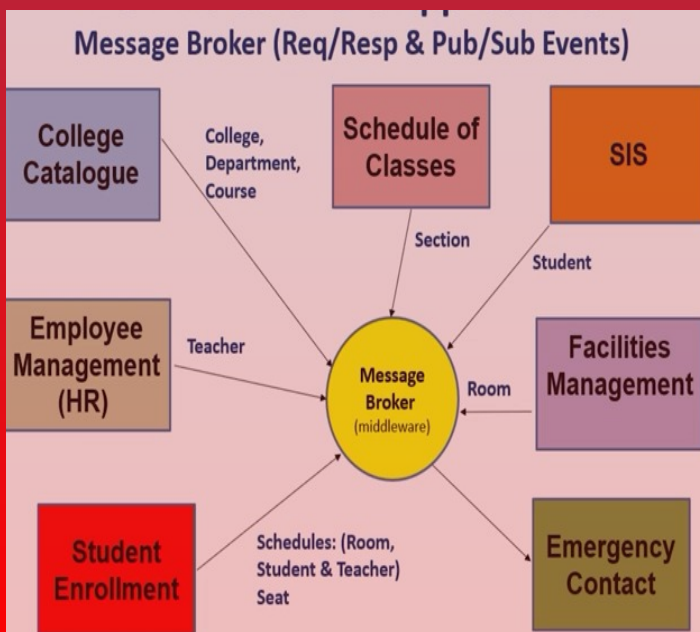
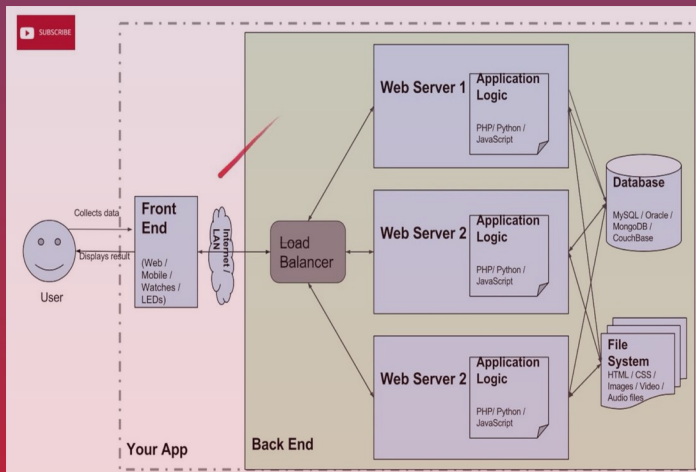


System Architecture

- Blueprints
- System and Domain Analysis
 - ~overview of the system and how the system or solution is going to work.
 - ~Gather, Analyze and Validate domain knowledge.
 - ~Understanding bussiness, procees and software needs.
- Requirements Analysis
 - ~Define requirements and prototype for the system
 - ~High level skeleton of components and how they interact
 - ~Define system Behaviour and communication / coordination Among components.
- Market and Risks analysis
 - ~Make estimates of cost security integration computing resources constraints
- Compose sub-systems into a larger system
- Evaluate other alternative solutions and prioritize requirements
- Examine Requirements and enhance the system goal
- Createa Software Requirements Specification(SRS) which specifies software,hardware,functional,and networking requiremnts of the system

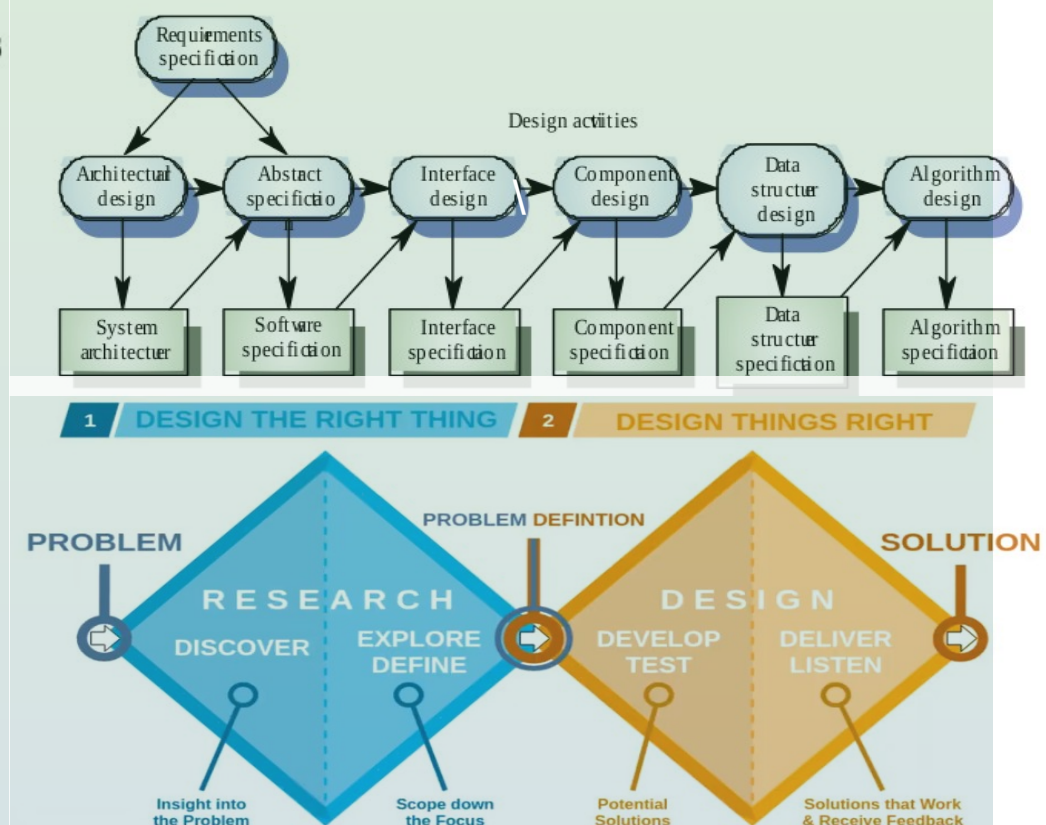


System Design

- Requires a document (SRS) that describes the behavior and state of the system during interactions with a User, Hardware and other components of the systems.
- Identify design goals from SRS and transform it into a logical structure.
- Design and delve into individual elements, functions, modules, classes and components of the system.
- Proceeds with design synthesis and system validation identifying boundary conditions.
- Design initial sub system decomposition.
- Map sub systems to processes and components.
- Design system controls, security, access control policies and control flow mechanisms.
- Design SYSTEMS INTERFACES (api, shells, backend, integration buses), STORAGE/DATABASE, DATA STRUCTURE, ALGORITHMS, NETWORK, HARDWARE, UI, RESOURCES.
- Design contingency, training, maintenance, testing and operational plans.
- Reviewing proposed designs

Design Considerations

- Compatibility
- Extensibility
- Fault-tolerance
- Maintainability
- Modularity
- Reliability
- Reusability
- Robustness
- Security
- Usability
- Performance
- Portability
- Scalability



System Development

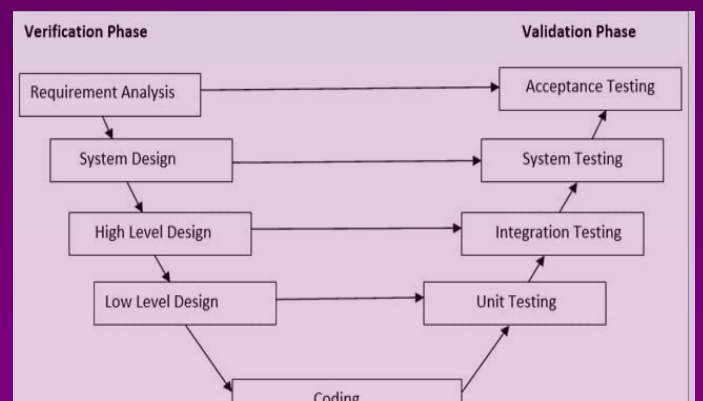
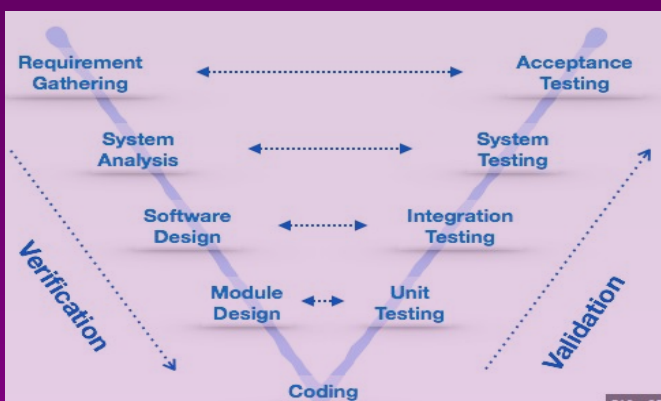
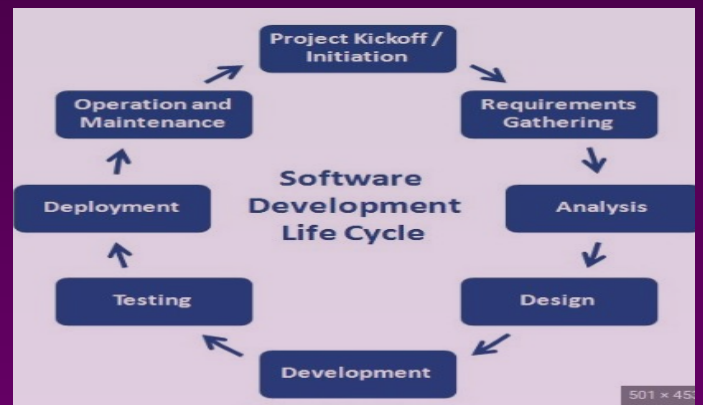
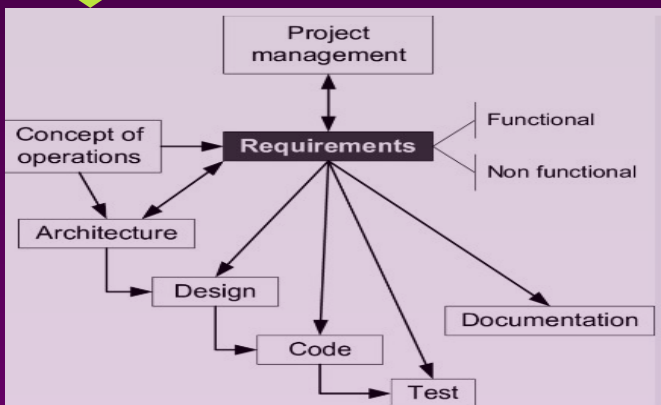
- System implementation
- Implement design details into source codes
- Documentation
- Integrates all modules, libraries, and source code into an environment to be used for testing, validation and deployment
- Coding, testing, QA, deploying, maintenance and support, Evolution

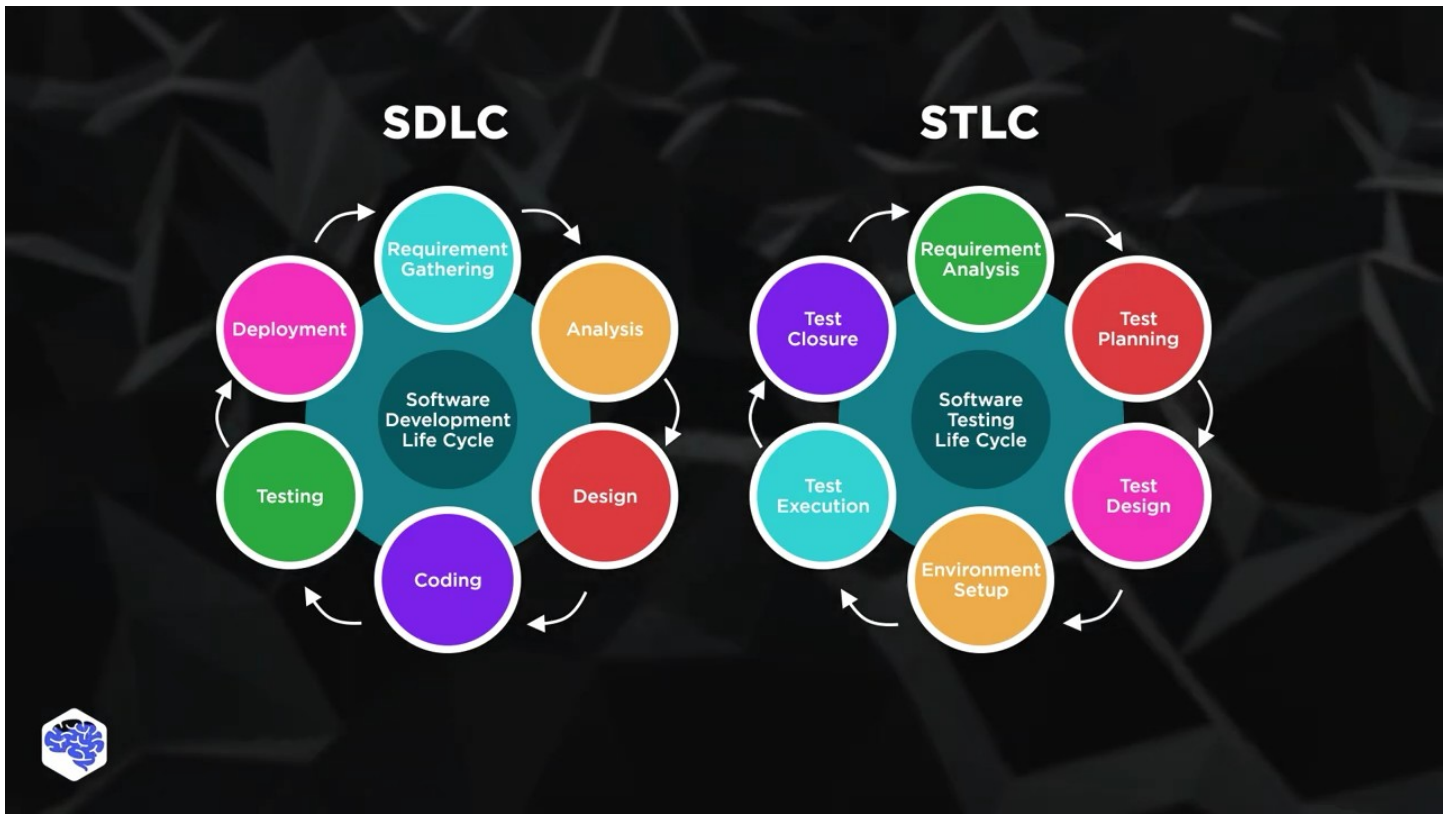
- Coding
- Documentation

- Testing
- Validation
- Integration
- Deployment

- Maintenance
- Support
- Evolution

Everything Together





IDEATION

Brainstorming ideas that solve a particular problem faced by target users.

REQUIREMENTS

Interacting with stakeholders and users to collect and document project requirements.

DESIGN

Creating the architecture of a software system and its elements.

DEVELOPMENT

Building the software using a programming language by the development team.

TESTING

Evaluating the quality of software with the aim of finding and fixing defects.

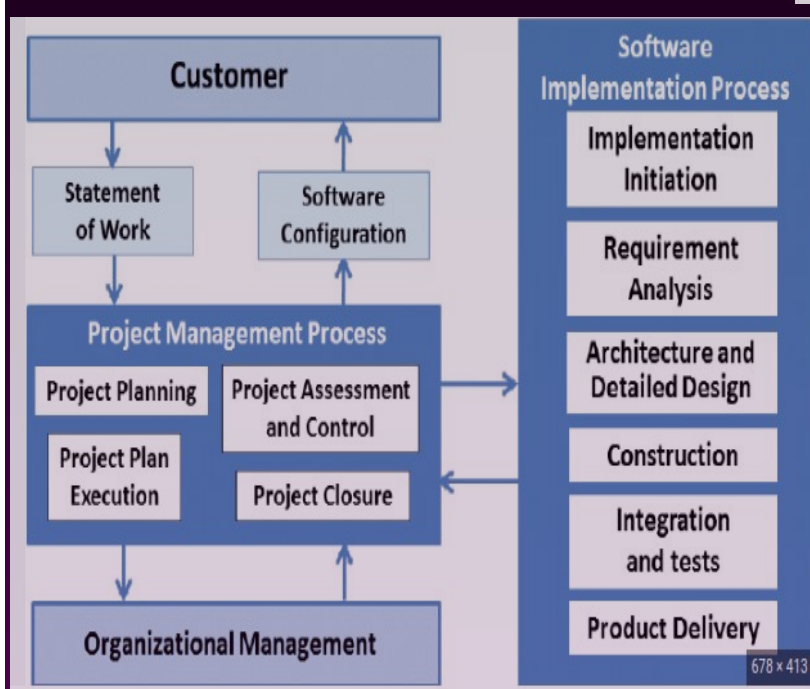
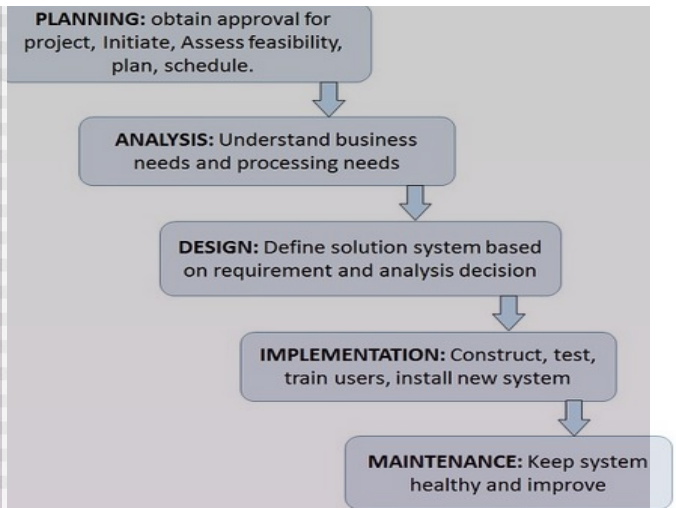
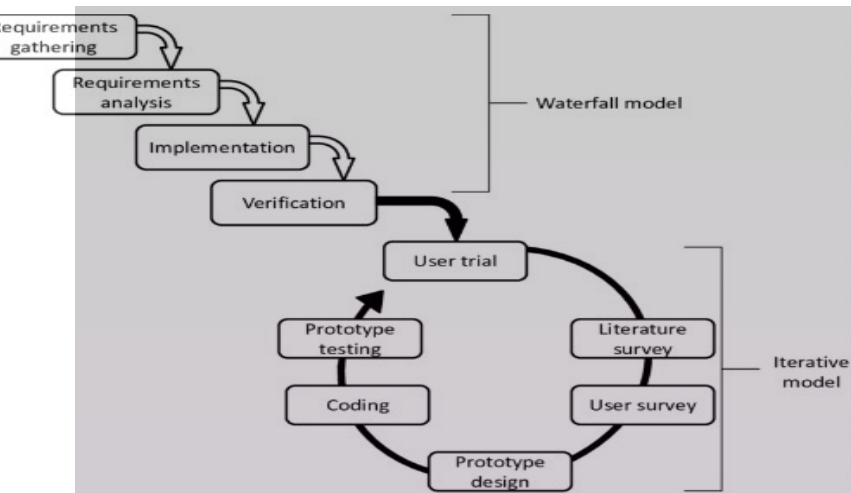
Deployment

Preparing the software to run and operate in a specific environment.

MAINTENANCE

Updating and supporting the software after it has been delivered to the market.

Software Development Life Cycle



More coming soon!!!!