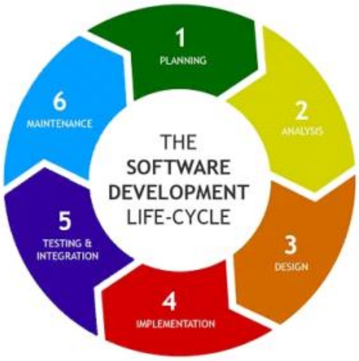
Models

Tuesday, September 17, 2024 11:37 AM

The SDLC is a framework that describes the activities performed at each stage of a software development project. SDLC process is used by the software industry to design, develop and test high quality software. It aims to produce the quality software that meets or exceeds customer expectations, reaches completion within time and budget.



SDLC Phases

1. Planning and Requirements Analysis
2. Defining Requirements
3. Designing the Software
4. Building or Developing the Software
5. Testing the Software
6. Deployment and Maintenance

SDLC- Software Development Life Cycle, a structured process that helps developers plan, design, develop, test, and deploy software. It's a systematic approach that helps ensure software is developed efficiently and consistently.

Types of Model:-

Waterfall Model -

- characterized by a structured, sequential approach to project management and software development.

Iterative Model -

The Iterative Waterfall Model is a software development approach that combines the sequential steps of the traditional Waterfall Model with the flexibility of iterative design.

Spiral Model -

- combination of the waterfall model and the iterative model.
- Risk Handling.
  Each loop of the spiral is called a phase of the software development process.

V-model -

- where the process executes sequentially in a V-shape
- Verification and Validation model.
- The development of each step is directly associated with the testing phase.

SDLC- Software Development Life Cycle, a structured process that helps developers plan, design, develop, test, and deploy software. It's a systematic approach that helps ensure software is developed efficiently and consistently.

- **Waterfall Model:**
  - **Sequential Process:** Changes in later stages can be difficult and expensive to implement.
  - **Late Testing:** Testing happens at the end, which can lead to high defect rates and rework.
  - **Limited Flexibility:** No room for adjusting project scope after initial stages, making it unsuitable for projects where requirements may evolve.
- **V-Model (Verification and Validation):**
  - **Rigid Structure:** Like the Waterfall model, it is sequential and lacks flexibility.
  - **Late Working Software:** Working software is available only at the end of the development cycle.
  - **High Risk for Changing Requirements:** If the project undergoes requirement changes, adapting is difficult.
- **Iterative Model:**
  - **Incomplete Requirements:** Initial iterations may lack clear requirements, leading to potential rework.

    - **Frequent Feedback:** May result in scope creep (continuous change in scope) if feedback is not managed properly.
    - **Resource Management:** More resource-heavy due to continuous iteration and testing.
- **Spiral Model:**
    - **Complexity:** Managing risks and iterations can become complex, requiring advanced expertise.
    - **Costly for Small Projects:** Best suited for large, complex projects but not cost-effective for small projects.
    - **Risk Assessment:** Inaccurate risk analysis can lead to project failure.
- **Agile Model:**
    - **Communication Overload:** Constant communication is required, which may slow down progress if not well managed.
    - **Scope Creep:** Lack of a defined scope can lead to uncontrolled changes, resulting in delays or additional costs.
    - **Requires Skilled Team:** Agile requires a high level of collaboration, flexibility, and experience from team members.