**1. What is SDLC and What Are Its Steps?**

**SDLC** stands for **Software Development Life Cycle**. It's a process used by software industry professionals to design, develop, test, and deliver high-quality software. It helps ensure that the software meets the customer's needs and requirements effectively. The SDLC process typically follows a series of phases that guide the team through software development.

The **steps of SDLC** include:

1. **Requirement Analysis**: Gathering the requirements from stakeholders, defining what the software needs to do.
2. **Planning**: Creating a project plan, determining resources, timelines, and estimating costs.
3. **Design**: Designing the system architecture and technical details, which includes the user interface, data models, etc.
4. **Implementation (Coding/Development)**: Writing the actual code to create the software application.
5. **Testing**: Testing the developed software to ensure that it meets the requirements and is free of bugs.
6. **Deployment**: Releasing the software to users or moving it into production.
7. **Maintenance**: Handling post-deployment updates, bug fixes, and adding enhancements.

**2. List and Describe Different SDLC Types**

There are several SDLC models that differ based on the process flow and project needs:

1. **Waterfall Model**:
   * A linear and sequential model.
   * Each phase must be completed before moving to the next one.
   * Suitable for projects with well-defined requirements.
   * Easy to manage but lacks flexibility in accommodating changes.
2. **Agile Model**:
   * Iterative and incremental model.
   * Involves continuous customer feedback and adaptive planning.
   * The project is developed in small parts (iterations), allowing flexibility.
   * Ideal for projects where requirements evolve or are uncertain.
3. **V-Model (Validation and Verification)**:
   * Extension of the Waterfall Model but emphasizes validation and verification.
   * Testing phases are planned in parallel with corresponding development stages.
   * Ensures early detection of defects, making it reliable for critical systems.
4. **Spiral Model**:
   * Combines iterative development with the systematic aspects of the Waterfall Model.
   * Emphasizes risk analysis.
   * Each phase begins with planning and ends with a customer evaluation.
   * Suitable for large projects with high-risk elements.
5. **Incremental Model**:
   * Develops software in increments, adding functionality in small pieces.
   * Each increment delivers part of the required functionality until the final product is complete.
   * Useful when the complete requirements are not clear at the beginning.
6. **RAD (Rapid Application Development) Model**:
   * Focuses on rapid prototyping and quick feedback over long development and testing cycles.
   * Promotes quick iterations and reduces development time.
   * Ideal for projects that need to be delivered quickly.
7. **DevOps**:
   * Focuses on integrating development and operations teams to automate and streamline the process.
   * Emphasizes continuous integration and continuous delivery (CI/CD).
   * Enables frequent updates and improved collaboration between teams.