**1) SDLC :** It stands for Software Development Life Cycle.

**Phases of SDLC**

1. Requirement analysis
2. Planning
3. Architectural Design & Prototype
4. Coding i.e. Software Development
5. Testing
6. Deployment
7. Maintenance

**SDLC Models**

1. **Waterfall Model**

Use cases for the Waterfall SDLC model:

* The requirements are precisely documented
* Product definition is stable
* The technologies stack is predefined which makes it not dynamic
* No ambiguous requirements
* The project is short

1. **V-Shaped Model**

Use cases for the V-shaped model:

* For the projects where an accurate product testing is required
* For the small and mid-sized projects, where requirements are strictly predefined
* The engineers of the required qualification, especially testers, are within easy reach.

1. **Iterative Model**

Use cases for the Iteration model:

* The requirements to the final product are strictly predefined
* Applied to the large-scale projects
* The main task is predefined, but the details may advance with the time

1. **Spiral Model**

Use cases for the Spiral model:

* Customer isn’t sure about the requirements
* Major edits are expected during the development cycle
* The projects with mid or high-level risk, where it is important to prevent these risks
* The new product that should be released in a few stages to have enough of clients feedback

1. **Big Bang Model**

A bit of an anomaly among SDLC methodologies, the Big Bang model follows no specific process, and very little time is spent on planning. The majority of resources are thrown toward development, and even the client may not have a solid grasp of the requirements. This is one of the SDLC methodologies typically used for small projects with only one or two software engineers.

Big Bang is not recommended for large or complex projects, as it’s a high-risk model; if the requirements are misunderstood in the beginning, you could get to the end and realize the project may have to be started all over again.

1. **Agile Model**

Use cases for the Agile model:

* The users’ needs change dynamically
* Less price for the changes implemented because of the many iterations
* Unlike the Waterfall model, it requires only initial planning to start the project