**DAY 1**

**SDLC**

Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality software’s. It consists of a detailed plan describing how to develop, maintain, replace, and alter or enhance specific software.

**WaterFall**

Waterfall methodology is a linear project management approach, where stakeholder and customer requirements are gathered at the beginning of the project, and then a sequential project plan is created to accommodate those requirements. There are five phases of waterfall method Requirements, Design, Implementation, Verification, Maintenance. Advantages of the method are it is easy to understand, fewer production issues, better budget management.

**AGILE**

Agile software development refers to a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams. Agile methods or Agile processes generally promote a disciplined project management process that encourages more frequent inspection and adaptation, a leadership philosophy that encourages teamwork, self-organization and accountability, a set of engineering best practices intended to allow for rapid delivery of high-quality software, and a business approach that aligns development with customer needs and company goals.

Scrum is a subset of Agile. It is a lightweight process framework for agile development, and the most widely-used one. The process is lightweight Scrum is a subset of Agile. It is a lightweight process framework for agile development, and the most widely-used one. It breaks down the agile method into several smaller incremental releases of the products called sprints, until the product is complete.

**Scrum Master**

The name was initially intended to indicate someone who is an expert at Scrum and can therefore coach others. The Scrum Master is also responsible for improving interactions between the Scrum team and the organization to maximize the productivity of the Scrum team. Finally, the Scrum Master is responsible to arranges and facilitates the team’s meetings.

**Product backlog**

The product backlog is compiled of all the things that must be done to complete the whole project. But it’s not just a simple list. An effective product backlog breaks down each of the items on the list into a series of steps that helps the development team.

Sprint Backlog is a subset of Product Backlog it contains only that item, or those items, that can be completed during each sprint.

Velocity

**Velocity** is a measure of the amount of work a Team can tackle during a single Sprint and is the key metric in **Scrum**. **Velocity** is calculated at the end of the Sprint by totaling the Points for all fully completed User Stories.

**Epic**

Are large bodies of work that can be broken down into a number of smaller tasks, Epics are a helpful way to organize your work and to create a hierarchy. Stories also called “user stories,” are short requirements or requests written from the perspective of an end user.

A sprint employs four different scrum ceremonies to ensure proper execution: sprint planning, daily scrum, sprint review and sprint retrospective.

1) Sprint Planning: This is where the team meets and decides what they need to complete in the coming sprint

2) Daily Scrum: This is a standup meeting, or a very short – 15-minute mini-meeting – for the team to make sure they’re all on the same page.

3) Sprint Review: This is another type of meeting, but one in which the team demos what they shipped in the sprint.

4) Sprint Retrospective: This is when the team reviews their work, identifying what they did well and what didn’t go as planned, so they can make the next sprint better.

**Grooming**

Grooming is when the product owner and some, or all, of the rest of the team review items on the backlog to ensure the backlog contains the appropriate items, that they are prioritized, and that the items at the top of the backlog are ready for delivery. This activity occurs on a regular basis and may be an officially scheduled meeting or an ongoing activity.

**How is Jira Usefull**

Jira Scrum board is designed so teams can organize their work around the Sprint timeframe. Jira Scrum Boards provide transparency into the team's work by slicing work into stages and utilizing burndown and velocity reports.

**Waterfall and SCRUM**

The waterfall development model or traditional software development life cycle. Its working approach is linear and sequential, it completes one activity before starting the other activity. Waterfall’s working style break up the work into the requirement, analysis, design, coding and testing, and term that phases. This works well with smaller projects

This is a member of the agile family. Scrum puts the focus on the management and development of the project. Scrum process is used to manage, develop and deliver the project on time. Scrum works best for complex projects and innovative solutions are delivered.

**Product Owner Responsibilities**

Defining the vision, Managing the product backlog, Prioritizing needs, Overseeing development stages, Anticipating client needs, Evaluating product progress at each iteration.

**Day 2**

**World Wide Web**

A Web server is software or hardware that uses HTTP (Hypertext Transfer Protocol) and other protocols to respond to client requests made over the World Wide Web (WWW).

**Characteristics of a client server computing.**

The client server computing works with a system of request and response. The client sends a request to the server and the server responds with the desired information.

The client and server should follow a common communication protocol so they can easily interact with each other. All the communication protocols are available at the application layer.

A server can only accommodate a limited number of client requests at a time. So it uses a system based to priority to respond to the requests.

Denial of Service attacks hinder a servers ability to respond to authentic client requests by inundating it with false requests.

**Presentation layer** is concerned with the representation of information as data, A major function of this is also making sure that data going in can be used by the local node, and that data going out can be used by the remote node.

A **database server** is a server which uses a database application that provides database services to other computer programs or to computers.

A **super-server** starts other servers when needed, normally with access to them checked by a TCP wrapper. It uses very few resources when in idle state. This can be ideal for workstations used for local web development, client/server development or low-traffic daemons with occasional usage.

A **2-tier** architecture is a software architecture in which a presentation layer or interface runs on a client, and a data layer or data structure gets stored on a server.

A **3-tier** architecture is a type of software architecture which is composed of three “tiers” or “layers” of logical computing. They are often used in applications as a specific type of client-server system. 3-tier architectures provide many benefits for production and development environments by modularizing the user interface, business logic, and data storage layers.

**File Server** The term server highlights the role of the machine in the traditional client–server scheme, where the clients are the workstations using the storage. A file server does not normally perform computational tasks or run programs on behalf of its client workstations.

**SOA & MicroServices**

Service-oriented architecture (SOA) enables increased business agility, improved business workflows, extensible architecture, enhanced reuse, and a longer life span of applications.

**reusability Service** is a design principle, applied within the service-orientation design paradigm, to create services that can be reused across a business. These reusable services are designed so that their solution logic is independent of any business process or technology.

**Differences between Web services and SOA**. Web services define a web technology that can be used to build applications that can send /receive messages using SOPA over HTTP. However, SOA is an architectural model for implementing loosely coupled service based applications. Web services can be used to implement SOA applications.

**Disadvantages of SOA**

High Bandwidth Server – As therefore net service sends and receives messages and knowledge often times so it simply reaches high requests per day. So it involves a high-speed server with plenty of information measure to run an internet service.

Extra Overload – In SOA, all inputs square measures its validity before it’s sent to the service. If you are victimization multiple services then it’ll overload your system with further computation.

High Cost – It is expensive in terms of human resource, development, and technology.

An **enterprise service bus** (**ESB** for short) refers to software architecture that allows for the integration of enterprise applications and services, such as middleware infrastructure platforms.

No in **SOA** we don’t need build a system from scratch, if we need to integrate any existing system you just can loosely couple wrappers which help in wrapping all customer services and expose all functionalities in a generic manner.

**Cultural**. SOA does require people to think of business and technology differently. Instead of thinking of technology first (e.g., If we implement this system, what kinds of things can we do with it?), practitioners must first think in terms of business functions, or services (e.g., My company does these business functions, so how can I set up my IT system to do those things for me most efficiently?).It is expected that adoption of SOA will change business IT departments, creating service-oriented (instead of technology-oriented) IT organizations.

Benefits of **Micro Services** Splits up complexity, Reusable components, Easy to scale, Reduces deployment time