

# **NJC LABS ACCELERATOR PROGRAM**

## **SESSION 1 NOTES**

### **1) Explain SDLC at a high level**

Software Development Lifecycle, widely referred to as SDLC, is a framework defining the general steps taken to develop a software in a software organization. It defines the several tasks that are performed by a team and responsibilities for team members at each steps of phases. It contains a detailed plan describing how to develop, maintain, alter, or enhance a software. These steps may overlap in few situations, but generally define the phases of the project. If some steps are not successful, the project may fall back to an earlier step.

SDLC is broadly divided into seven phases:

1. **Formation** – This is the most important and fundamental stage in SDLC, performed by senior members of the team with inputs from customers, sales department, market surveys and domain experts in industry. The information collected is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational, and technical areas. This stage is considered important as the planning for quality assurance requirements and risk identification associated with the project is done at this stage to implement the project successfully with minimum risks.
2. **Requirement/Planning** – Requirements of the product are clearly defined and documented at this stage through a Software Requirement Specification (SRS) document and get them approved by customers or market analysts. A Project Management Plan (Charter) is put into place for a smooth implementation of the project.
3. **Design** – Based on the requirements specified in SRS, more than one design approach is proposed and documented. SRS is considered as a reference by the product architects to come out with best architecture for the product to be developed. All the documentation is reviewed by the stakeholders and based on various parameters the best design approach is selected for the product.
4. **Construct** – The actual development starts, and the product is built at this stage. The programming code is generated as per the design documentation prepared at previous phase. The developers must follow the guidelines set by the organization and programming tools. A lot of Code Review happens at this stage to minimise product failure at later stages.
5. **Test** – A product goes through several types of testing throughout its development. However, at this stage a lot of testing activities are performed on the product where product defects or bugs are reported, tracked, fixed, and documented until the product reaches the quality standards defined in SRS.
6. **Product Release** – Once the product is tested and is ready for deployment, it is formally released in the market after operational acceptance by the stakeholders. The stakeholders sign an Acceptance document which is a formal handover of all the documentation and approval that the product meets the requirements and quality standards.
7. **Post Implementation** – After the product is released in the market, its maintenance is done for customer base.

## 2) What is waterfall and why it is still relevant?

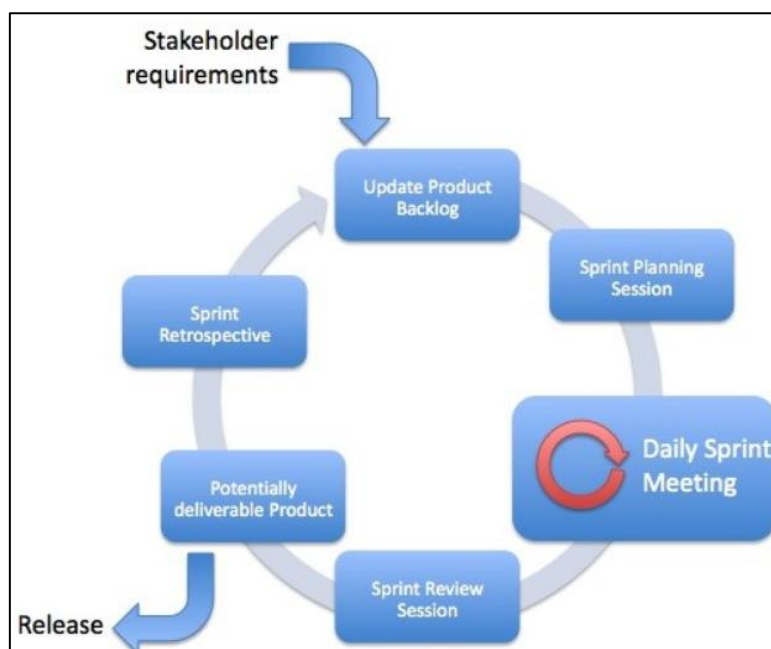
A Waterfall model is a model of software development which is a linear model in which all the steps/stages are performed sequentially and follows a top-down approach. Only after first stage in software development lifecycle is completed successfully, does the team move on to later stages.

Waterfall model has several pros and cons; however, it is still relevant to use this model for certain projects. The sequential approach allows to define clear goals and direction for the project during the initialisation. The progress of the project can be easily measured, and it is clear as to when to move to the next stage. There are clear milestones which indicate the progress of the project which saves time and money. Waterfall model can be followed when the scope of the project is known, and it is certain that no major changes can be expected during the development of the project. It is easier to follow this approach when the team has worked on similar projects in past and are bound to work in a strict budget for a project.

## 3) Explain Agile Model with a use case and the role of SCRUM in that.

Agile model is a software development approach where software development is an iterative process in which the feedback from other phases are considered as input to earlier phases to develop a product with desired requirements and quality standards.

SCRUM is an agile framework which divides a project into iterations or sprints, implements each iteration sequentially following all the software development stages for every sprint/iteration. The feedback received for a previous iteration is taken as input for next iteration. This allows to identify and rectify defects in the project at early stages.



Consider a use case of building a house. The main actors(stakeholders) of this use case are the owner of the house who wishes to build the house, the chief architect/engineer, the project manager. The owner discusses their requirements with the architect using which the architect develops a plan for the house. The architect gets it approved by the owner and promises to show the development in the building after every two weeks. This can be considered as the sprint time for this project. The architect discusses with the project manager about the plan and divides the building into smaller chunks for the development (sprint in case of SCRUM). The architects, project manager and workers discuss the

progress of the construction every day, which is similar to the daily sprint meeting. Whenever the owner visits the construction (Sprint Review), the owner gives their feedback to the architect based on which any further changes are made in the construction. After the construction is completed successfully with several visits (sprint review sessions) and the owner's feedback, the project is finally completed and is ready for handover (Release). SCRUM can be realised in many real time projects. In this scenario, the project manager is the Scrum Master who is responsible for gluing everything together and ensuring that the construction (scrum) is being done correctly. The chief architect, the Product owner, understands the stakeholder requirements, maintains a log of the work to be done, prepares a plan for development and plans the release of the project. The workers are the development team of the project in terms of SCRUM.

#### **4) Who is Scrum Master?**

SCRUM Master is the role responsible for gluing everything together and ensuring that the scrum is being done well. In practical terms, that means they help the product owner define value, the development team deliver the value, and the scrum team to get to get better. The scrum master is a servant leader which not only describes a supportive style of leadership but describes what they do on a day-to-day basis.

They serve the product owner by helping them better understand and communicate value, to manage the backlog, help them plan the work with the team and break down that work to deliver the most effective learning. The scrum master also serves the organization at large, helping them understand what scrum is and create an environment that supports scrum.

#### **5) Differentiate between Product/Sprint Backlog.**

A Product Backlog is compiled of all the things that must be done to complete the whole project. An effective product backlog breaks down each of the items on the list into a series of steps that helps the development team. There must be a duration, so the team knows when to start the task and how long they have until they must finish it. Product backlog is flexible. Although it is planned, it may be changed as the development team starts to work. The product backlog shrinks when the task is completed, as it should be removed from the product backlog list or new tasks are added as the project grows.

Sprint Backlog is a subset of the Product Backlog. The sprint backlog comes from the product backlog, but it contains only that item, or those items, that can be completed during each sprint. The complexity of the project will determine the sprint backlog, but overall the idea is to dedicate the team only to those tasks that can be completed during the sprint. Of course, if it is a complex project then sprint backlog can also grow in complexity and length.

Unlike the product backlog, though, the sprint backlog is unchanged during the period of the sprint. It can be changed, but only during the sprint planning meeting. Once agreed upon, the items and steps to complete them are frozen for the length of the sprint. If there are items left unfinished by the end of the sprint, they will be added back to the product backlog and addressed during the next sprint.

#### **6) What is Epic & Story?**

Epic is a bigger portion of a project which needs to be realized to successfully complete a project. An Epic can be further divided into user stories and tasks based on the needs or requirement of the customer or end users. Epics are helpful way to organize the work and create a hierarchy. The idea is to breakdown a project into deliverable pieces, so that large projects can actually get done and we can continue to ship value to the customers on regular basis. For example, in an online shopping

project, a customer might want to have a wishlist to come back and buy products. This can be considered as an epic.

A User story is the smallest unit of work in an agile framework. It is an end goal, not a feature, expressed from the software user's perspective. The purpose of a user story is articulate how a piece of work will deliver a particular value back to the customer. User stories are building blocks of larger agile frameworks like epic. User stories are a few sentences in simple language that outline the desired outcome. They don't go into detail.

## **7) What is called Velocity in SCRUM?**

Velocity is an extremely simple, powerful method for accurately measuring the rate at which scrum development teams consistently deliver business value. It is an indication of the average amount of Product Backlog turned into an Increment of product during a Sprint by a Scrum Team, tracked by the Development Team for use within the Scrum Team. Thus, to calculate velocity of an agile team, the estimates of the features, user stories, requirements or backlog items successfully delivered in an iteration are simply added up.

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 summing up the Points for all fully completed User Stories. Points from partially completed or incomplete stories should not be counted in calculating velocity.

There are some simple guidelines for estimating initial velocity of a Scrum team prior to completing the first few iterations, but after that point a team could use proven, historical measures of velocity estimation for sprint planning. Based on a series of past sprints, the estimation of velocity typically stabilizes and provides a more reliable basis for improving the accuracy of both short-term and longer-term planning of the Scrum projects.

## **8) Explain the SCRUM ceremonies.**

Successful SCRUM implementations involve a handful of important ceremonies. These ceremonies provide the framework for teams to get work done in a structured manner, help to set expectations, empower the team to collaborate effectively, and ultimately drive results. If they are not managed appropriately, however, they can overwhelm calendars and drown out the value they are intended to provide.

The following are the several events or ceremonies that are followed in SCRUM:

- Organize the backlog – Understand the requirements of the project and identify all the items that need to be fulfilled to successfully complete the project. Each item is broken down into steps to be followed, clearly defining the duration of each step.
- Sprint Planning – This ceremony happens at the beginning of a new sprint and is designed for the Product Owner and Development Team to meet and review the prioritized Product Backlog. Through a series of discussions and negotiations, the team ultimately decides on a sprint backlog that contains all items they are committing to deliver at the end of sprint.
- Sprints – This is the duration in which a team decides to implement all the items in a sprint backlog.
- Sprint Execution – This is the duration in which the team works to complete the work. The development team is involved majorly in the sprint execution.
- Daily Scrum/Stand Up - The Daily Scrum is the team's chance to get together, define a plan for the day's work, and identify any blockers. This ceremony provides a frequent opportunity for

the team to get together and communicate individual progress toward the sprint goal. It is not a status update. Instead, it should illuminate any impediments the team is having. The Scrum Master is responsible for clearing these roadblocks for the Development Team so they can focus on delivering the work identified in Sprint Planning.

- **Sprint Review** - The Sprint Review is the scrum ceremony where all work completed during the sprint can be showcased the stakeholders. At the conclusion of each sprint, the Sprint Review provides a platform for the Development Team to showcase all the work that has been completed. This allows stakeholders to see things sooner than later and inspect or adapt the product as it emerges.
- **Sprint Retrospective** - The Sprint Retrospective is the final scrum ceremony in the sequence that allows the team to look back on the work that was just completed and identify items that could be improved.

### **9) What is grooming?**

Grooming (or refinement) is a meeting of the Scrum team in which the product backlog items are discussed, and the next sprint planning is prepared. Product grooming is critical in product management because it means keeping the backlog up to date and getting backlog items ready for upcoming sprints.

Backlog grooming is often named pre-planning. Product owner and team representatives arrange it in the mid-sprint time. In this case, planning and refinement meetings alternate but happen on the same day each week. That provides an effective rhythm for the entire team.

The grooming involves splitting big items into smaller ones, rewriting backlog items to be more expressive, deleting obsolete or no more need items, and so on.

### **10) How Jira board is effective in SCRUM?**

A Jira board helps in managing the sprint effectively and efficiently with its simple UI and easy to use options. Jira Scrum Boards provide transparency into the team's work by slicing work into stages and utilizing burndown and velocity reports. A board displays issues from one or more projects, giving the team a flexible way of viewing, managing, and reporting on work in progress.

A JIRA Scrum board allows a team to create a product backlog and add sprints or issues related to it. The sprints are defined with details like sprint name, sprint duration, sprint start date, end date and sprint goal. The team can also add sub tasks to the sprints which are the smaller blocks of the sprints. The board depicts the active sprint and the issues the team is handling. The JIRA board allows the team to conduct daily meetings, report and log updates or issues, document the update and assign tasks to team members. JIRA board allows the entire team to maintain and communicate the progress of the project effectively.

### **11) Differentiate between SCRUM & Waterfall.**

Waterfall model is a software development approach which is a linear model in which all the stages are performed sequentially in a top-down approach. Only after the completion of one activity, the next activity is performed.

SCRUM is an agile framework in which the project is divided into smaller iterations/scrums and each stage of the software development are performed iteratively for each iteration. Scrum process is used to manage, develop, and deliver the project on time. Scrum works best for complex and large projects.

	SCRUM	Waterfall
Customer involvement	Involves customer and stakeholders at each phase	Keeps customers at bay until the release
Time and Money	Saves time and money by regular sprint review	May take extra time as reviewing is done at the end. If result is not successful, the process is back to stage 1
Work distribution	Work is divided into teams as individual responsibility	Work is divided into phases. Team works closely
Suitability	Works well with large and complex projects	Works well with smaller projects
Stages	No defined stages	Clear and defined stages to work on project
Change of scope	Welcomes changes at early and later stages	Welcomes changes only at initial stages. No liberty of changes at later stages
Product Delivery	Working software is delivered at every iteration	Complete software is delivered together at the end

## 12) Explain the responsibilities of Product Owner.

Agile teams are, by design, flexible and responsive, and it is the responsibility of the product owner to ensure that they are delivering the most value. The business is represented by the product owner who tells the development what is important to deliver. Trust between these two roles is crucial.

The product owner should not only understand the customer, but also have a vision for the value the scrum team is delivering to the customer. The product owner also balances the needs of other stakeholders in the organization.

The responsibilities of a product owner can broadly be defined as follows:

- **Managing Scrum Backlog** - The product owner is responsible for the product backlog that the development team pulls to deliver from. However, the product owner is not the only responsible to add or delete items in backlog. This means that the product owner should be aware of everything that happens in product backlog. The other team members need to communicate with the product owner while adding an item to backlog.
- **Release Management** – The scrum teams deliver frequently throughout the development allowing the customers to review. The teams may not be able to deliver continuously all the time. The product owner is responsible to decide the release times and know when things can and should be released.
- **Stakeholder Management** - Any product will have many stakeholders involved ranging from users, customers, governance, and organizational leadership. The product owner will have to work with all these people to effectively ensure that the development team is delivering value. That can mean a large amount of stakeholder management and communication.