River Island - Head Office, Chelsea House, West Gate, EALING, W5 1DR

Scrum is an iterative and incremental agile software development framework for managing product development. E.g Scrum process framework requires the use of development cycles called Sprints, Test driven programming.

Scrum is most often used to manage complex software and product development, Scrum significantly increases productivity coampared to classic “waterfall” processes. Scrum processes enable organizations to adjust smoothly to rapidly-changing requirements, and produce a product that meets evolving business goals.

An agile Scrum process benefits the organization by helping it to

•Increase the quality of the deliverables

•Cope better with change (and expect the changes)

•Provide better estimates while spending less time creating them

•Be more in control of the project schedule and state

Benefit of Agile

**Benefits to Customer**

Customers find that the vendor is more responsive to development requests. High-value features are developed and delivered more quickly with short cycles, than with the longer cycles favored by classic “waterfall” processes.

**Benefits to Vendors**

Vendors reduce wastage by focusing development effort on high-value features, and reduce time-to-market relative to waterfall processes due to decreased overhead and increased efficiency. Improved customer satisfaction translates to better customer retention and more positive customer references

**Benefits to Development Teams**

Team members enjoy development work, and like to see their work used and valued. Team members also know their work is valued, because requirements are chosen to maximize value to customers. Scrum benefits Team members by reducing non-productive work e.g unnecessary documentation.

The **Product Owner** is responsible for continuously communicating the vision and priorities to the development team. Product Owner is the interface between the business, the customers, and their product related needs on one side, and the Team on the other. He maintains the Product Backlog (which is the repository for all of this information), keeping it up to date and at the level of detail and quality the Team requires.

**Scrum Master**: The Scrum Master acts as a facilitator for the Product Owner and the team. The Scrum Master works to remove any impediments that are obstructing the team from achieving its sprint goals. •Improve the productivity of the development Team in any way possible. •Keep information about the Team’s progress up to date and visible to all parties.

**Scrum Team** : Each member in the team should be self-organized, dedicated and responsible for high quality of the work. Team members decide how to break work(user stories) into tasks, and how to allocate tasks to individuals, throughout the Sprint. The Team size should be kept in the range from five to nine people,

A User Story describes a desired feature (functional requirement) in narrative form. User Stories are usually written by the Product Owner

User Story is of the format : **Name, short description, Screen or external documents, How to test**

Not all requirements for new development represent user-facing features, but do represent significant work that must be done. We call these non-functional requirements “Technical Stories.” Technical Stories have the same elements as User Stories, but need not be cast into narrative form if there is no benefit in doing so. Technical Stories are usually written by Team members, and are added to the Product Backlog.

Defect

A Defect, or bug report, is a description of a failure of the product to behave in the expected fashion. Defects are stored in a bug-tracking system, which may or may not be physically the same system used to store the Product Backlog.

***Product backlog -> Spring backlog -> Sprint few week (e.g 30 days sprint) -> daily scrum calls > what is been done, what will be doing next, what challenges and issues facing currently, -> end of sprint and feature delivered***

A sprint (or iteration) is the basic unit of development in scrum. The sprint is a timeboxed effort; that is, it is restricted to a specific duration. The duration is fixed in advance for each sprint and is normally between one week and one month.

The product backlog comprises an ordered list of requirements that a scrum team maintains for a product. It consists of features, bug fixes, non-functional requirements, etc. The product owner orders the product backlog items (PBIs) based on considerations such as risk, business value, dependencies, and date needed.

Typically, the product owner and the scrum team come together and write down everything that must be prioritized, and this becomes content for the first sprint

The sprint backlog is the list of work the development team must address during the next sprint. The list is derived by the scrum team selecting product backlog items from the top of the product backlog until the development team feels it has enough work to fill the sprint.

Sprint burn-down chart : The sprint burndown chart is a public displayed chart showing remaining work in the sprint backlog.

Release burn-down chart : Feature level progress of completed product backlog items in the product backlog.

**Sprint** :A time period (typically 1–4 weeks) in which development occurs on a set of backlog items that the team has committed to—commonly referred to as a time-box or iteration

**Spike** : A time boxed period used to research a concept or create a simple prototype. Spikes can either be planned to take place in between sprints or, for larger teams, a spike might be accepted as one of many sprint delivery objectives. Spikes are often introduced before the delivery of large or complex product backlog items in order to secure budget, expand knowledge, or produce a proof of concept. The duration and objective(s) of a spike is agreed between product owner and development team before the start. Unlike sprint commitments, spikes may or may not deliver tangible, shippable, valuable functionality. For example, the objective of a spike might be to successfully reach a decision on a course of action.

**Tracer bullet** : The tracer bullet is a spike with the current architecture, current technology set, current set of best practices that results in production quality code. It might just be a very narrow implementation of the functionality but is not throw away code. It is of production quality, and the rest of the iterations can build on this code.

**Tasks** : Work items added to the sprint backlog at the beginning of a sprint and broken down into hours. Each task should not exceed 12 hours (or two days).

**Definition of done (DoD)** : The exit-criteria to determine whether a product backlog item is complete. In many cases the DoD requires that all regression tests should be successful.

**Velocity** : The total effort a team is capable of in a sprint. The number is derived by evaluating the work (typically in user story points) completed from the last sprints backlog items. The collection of historical velocity data is a guideline for assisting the team in understanding how much work they can do in a future sprint.

**ScrumBut** : ScrumBut (or Scrum but) is a term to describe the approach of a team who have adapted the scrum process to their own needs in some way contradictory to supposed pure scrum.

**Impediment** : Anything that prevents a team member from performing work as efficiently as possible.

**Sashimi** : A term used to describe one or more user stories, indicating that they are thin slices of a product feature or capability.

**Burn-down** chart represents the remaining work in a project.

**Burn-up** charts represent how much work has been completed

Modification of the code without changing its functionality to improve the performance is called **re-factoring**.

**Iterative**: Iterative method is a continuous process of software development where the software development cycles are repeated (Sprint & Releases) till the final product is achieved.

Release 1: Sprint 1, 2… n

Release n: Sprint 1, 2….n

**Incremental**: Incremental development segregates the system functionality into increments or portions. In each increment, each segment of functionality is delivered through cross-discipline work, from the requirements to the deployment.

**User Stories**:User Stories defines the actual business requirement. Generally created by Business owner.

**Task**: To accomplish the business requirements development team create tasks.

**Epic**: A group of related user stories is called an Epic.

**How do you deal when requirements change frequently?**

Write generic test plans and test cases which focus on the intent of the requirement rather than its exact details

Work very closely with the product owners or business analysts to understand the scope of change so testing can be updated

Make sure the team understands the risks involved in changing requirements especially towards the end of sprint

If you’re going to automate this feature, it is best to wait until the feature is stable and requirements are finalized

Negotiate to see if the changes can be kept to a minimum and/or implement the changes in next sprint.

**Task board** is dash board which shows progress of the project. It contains:

User Story: which has the actual business requirement.

To Do: Tasks that can be worked on.

In Progress: Tasks in progress.

To Verify: Tasks pending for verification or testing

Done: Completed tasks.

**Some Agile quality strategies are**: Re-factoring, Small feedback cycles, Dynamic code analysis, Iteration

**Agile manifesto** defines an iterative and people-centric approach to software development. It has basically 4 key values and 12 principals.

30) Mention what is the difference between Scrum and Agile?

Scrum: In the scrum, a sprint is a basic unit of development. Each sprint is followed by a planning meeting, where the tasks for the sprint are identified and estimated. During each sprint, the team creates finished portion of a product

Agile: In Agile, each iteration involves a team working through a full software development cycle, including planning, design, coding, requirement analysis, unit testing, and acceptance testing when a product is demonstrated to stakeholders

In simple words, Agile is the practice and scrum is the process to following this practice.

**31) Mention what are the challenges involved in AGILE software development?**

Challenges involved in Agile Software development includes

It requires more testing and customers involvement

It impacts management more than developers

Each feature needs to be completed before moving on to the next

All the code has to work fine to ensure application is in working state

More planning is required

**32) When not to use Agile?**

* Before using Agile methodology, you must ask following questions
* Is functionality split-able
* Is customer available
* Are requirements flexible
* Is it really time constrained
* Is team skilled enough

**33) Explain how can you implement scrum in an easy way to your project?**

These are the tips which can be helpful to implement scrum in your project

Get your backlog in order

Get an idea of the size of your product backlog items

Clarify sprint requirement and duration to complete the sprint backlog

Calculate the team sprint budget and then break requirements into tasks

Collaborate workspace- a center of all team discussion, which includes plans, roadmaps, key dates, sketches of functionality, issues, log, status reports, etc.

Sprint- Make sure you complete one feature at a time before moving on to the next. A sprint should not be abort unless if there is no other option

Attend a daily stand-up meeting: In meeting you need to mention, what have been achieved since the last meeting, what will they achieve before the next meeting and is anything holding up their progress

Use burndown chart to track daily progress. From the burndown chart, you can estimate whether you are on track, or you are running behind

Complete each features well before moving on to the next

At the end of the sprint- hold a sprint review meeting, mention what is achieved or delivered in the sprint.

**34) Explain what does it mean by product roadmap?**

A product roadmap is referred for the holistic view of product features that create the product vision.