**Faculty Reference**

**Document**

**(Version 2012\_1.2)**

**Instructions for Faculty (About this book):**

1. Content specified in ‘RED’ is available only in this faculty workbook/guide. These portions are not there in the student workbook. These are either guidelines for faculty or extra points and examples which a faculty should use to instruct the class. Faculty should be familiar with both the Faculty Workbook and Student Workbook.
2. There are some fill in the blanks inside the content of Student Workbook. In this Faculty Workbook, answers to fill in the blanks are given in ‘BLUE’ color.
3. Important concepts are highlighted in ‘Yellow’. Faculty should emphasize on these portions and ask the students to highlight the same.

[Agile & Scrum Overview 1](#_Toc340270413)

[Agile Defined 1](#_Toc340270414)

[Adaptive Project Management 2](#_Toc340270415)

[Understanding the Manifesto 3](#_Toc340270416)

[Principles behind the Agile Manifesto 4](#_Toc340270417)

[Agile Methods 7](#_Toc340270418)

[Scrum Overview 9](#_Toc340270419)

[Scrum Principles 9](#_Toc340270420)

[Advantages of using Scrum 12](#_Toc340270421)

[Chapter Quiz 13](#_Toc340270422)

[Scrum Roles 16](#_Toc340270423)

[Overview 16](#_Toc340270424)

[Product Owner 17](#_Toc340270425)

[Scrum Master 18](#_Toc340270426)

[Scrum Team 19](#_Toc340270427)

[Scrum Team characteristics: 19](#_Toc340270428)

[Team Selection 20](#_Toc340270429)

[Chapter Quiz 21](#_Toc340270430)

[Planning in Scrum 23](#_Toc340270431)

[Scrum Flow 23](#_Toc340270432)

[Pre-Project Meeting: 25](#_Toc340270433)

[Product Backlog 27](#_Toc340270434)

[Product Backlog Overview 27](#_Toc340270435)

[User Stories 28](#_Toc340270436)

[Creating User Stories 28](#_Toc340270437)

[Release Planning 30](#_Toc340270438)

[Chapter Quiz 31](#_Toc340270439)

[Sprint Planning 33](#_Toc340270440)

[Sprint Planning Meeting 33](#_Toc340270441)

[Estimation 36](#_Toc340270442)

[Product Backlog Grooming 38](#_Toc340270443)

[Done Definition 40](#_Toc340270444)

[Chapter Quiz 41](#_Toc340270445)

[Daily Scrum 43](#_Toc340270446)

[Meeting: Daily Stand-up/Daily Scrum 43](#_Toc340270447)

[Sprint Review Meeting 46](#_Toc340270448)

[Sprint Retrospective Meeting 48](#_Toc340270449)

[Chapter Quiz 49](#_Toc340270450)

[Implementation 51](#_Toc340270451)

[Scrum for Large Projects 51](#_Toc340270452)

[Transition to Scrum 52](#_Toc340270453)

[Mapping Traditional Roles to Scrum 53](#_Toc340270454)

[Distributed teams in Scrum 53](#_Toc340270455)

[Maintaining Stakeholder Involvement 54](#_Toc340270456)

[Importance of Executive Support 54](#_Toc340270457)

[Chapter Quiz 55](#_Toc340270458)

**INTRODUCTORY GUIDELINES FOR FACULTY**

Give a brief introduction of yourself, and ask delegates to do the same. If the class size is large (more than 8-10 people), then divide the delegates into groups and let them introduce themselves in their groups.

Provide delegates a set of Post-it notes, and instruct them to write down their queries and expectations and stick the notes to the questions backlog/expectations chart.

**Class Logistics and Preparation**

* Faculty sets up a story-board with their estimates of all the topics they expect will be discussed throughout the two days. As a topic is dealt with, it is moved from the in-progress section to the done section, and then the next topic is moved to the in-progress section.
* Faculty creates a corresponding burndown chart and updates it regularly (maybe halfway through the day, each day). It might be a good practice to let one of the delegates update the burndown chart and explain the progress to the team; he or she could discuss team velocity and how the group needs to progress to meet the deadline.
* Faculty sets up a questions backlog story-board where the delegates can post their queries and expectations for the course.
* Faculty is prepared with a timer to ensure all events in the course (especially breaks and exercises) are time-boxed.

**Ground Rules/Working Agreements**

* The course duration is two days. Class starts at 9:00 a.m. and will continue until 5:00 p.m. each day. There will be breaks in between sessions as instructed by faculty.
* Participants should be in the class on time and return from breaks as instructed by the faculty.
* Cell phones should be mandatorily switched off or kept in silent mode.
* Full participation is expected from all students. Please join in the activities and exercises as instructed by the faculty. Warning—you may actually have fun in this class!
* The study materials and resources that the participants will be using are copyrighted and wholly owned by SCRUMstudy. These should not be photocopied, distributed, or shared, and may only be used by students who have enrolled in the SCRUMstudy classroom training.

**Course Objectives**

* To provide an understanding of the philosophy and principles in the Scrum framework.
* To provide a practical working knowledge of Scrum— including roles, meetings, and artifacts.
* To prepare the participants to be comfortable in implementing Scrum in their organizations as well as to manage common issues and roadblocks.

**Course Outcomes**

* This course aims to familiarize delegates with the concepts, advantages, and challenges of the Scrum methodology.
* Students will be equipped with the knowledge needed to play the role of a Scrum Master in their organizations and help their organizations adopt Scrum methodology. Furthermore, delegates will develop an understanding of the other roles in Scrum.
* Students are led through simulated case scenarios during which they carry out a Scrum project.
* Students have knowledge pertaining to and can anticipate issues related to the practical implementation of scrum.
* Students will achieve certification, and be armed with the proper tools to address, resolve, and take the lead on Scrum issues in their organizations.

**Course Methodology**

* We promise a highly engaging course that ensures higher retention of concepts and theories.
* Delegates are encouraged to work through the concepts rather than just listen to the concepts. This provides a better internalization of concepts.
* Role-plays are conducted and practical implementation issues are discussed for all parts of the Scrum flow.
* A central case study is integrated into the course outline to allow delegates to simulate a product development exercise using Scrum methodology

# Agile & Scrum Overview

All of you must have heard of terms like Agile approach, Agile methodology, Agile teams, and so on; and, some of you may also have been part of projects using this methodology.

But what is Agile? Ask the class for opinions.

Jim Highsmith, a popular Agile guru who has written several books on Agile/Adaptive methods, defines it as follows.

## Agile Defined

So, Agile is being nimble. Agile is being adaptive to change. Agile is not just for software development; it can be used for any complex project. Highsmith uses mountaineering analogies to explain key concepts in his books.

### The Need for Agile

Before we delve into what Agile is all about, let’s understand why the development of Agile methods became necessary. The smartphone market can be taken as an example in which the following factors are prominent.

It is in such rapidly changing environments that predictive methods do not succeed and we look at adaptive methods for project management.

## 

## Adaptive Project Management

Agile was not developed in one day. People across the globe felt the need for Adaptive Project Management as they saw that the traditional Waterfall methods were lacking flexibility. Some of the methods are:

Notes: In case faculty is required to explain any of the technologies briefly.

**Crystal:** The Crystal family of methodologies focuses on efficiency and habitability as components of project safety. Crystal Clear focuses on people, not processes or artifacts.

Crystal Clear requires the following properties: Frequent delivery of usable code to users, Reflective improvement, and Osmotic communication.

**Dynamic Systems Development Method**: The Dynamic Systems Development Method is an Agile project delivery framework. DSDM is an iterative and incremental approach that embraces principles of Agile development, including continuous user/customer involvement. DSDM fixes cost, quality, and time at the outset and uses the MoSCoW prioritization of scope into musts, shoulds, coulds, and won't haves to adjust the project deliverable to meet the stated time constraint.

**Extreme Programming:** Extreme Programming is a software development methodology that is intended to improve software quality and responsiveness to changing customer requirements. Other elements of Extreme Programming include programming in pairs or doing extensive code review, unit testing of all code, avoiding programming of features until they are actually needed, a flat management structure, simplicity and clarity in code, expecting changes in the customer's requirements as time passes and the problem is better understood, and frequent communication with the customer and among programmers.

**Feature Driven Development:** FDD is a model-driven, short-iteration process that consists of five basic activities: Develop Overall Model, Build Feature List, Plan by Feature, Design by Feature, and Build by Feature.

**Adaptive Software Development:** ASD consists of a repeating series of speculate, collaborate, and learn cycles. This dynamic cycle provides continuous learning and adaptation to the emergent state of the project. The characteristics of an ASD life cycle are that it is mission focused, feature based, iterative, timeboxed, risk driven, and change tolerant.

## Understanding the Manifesto

Have the students highlight the bolded and highlighted manifesto statements to emphasize the concepts.

Applying any Agile method needs a clear understanding of the four values given below. While every project will have both sets of items, it is important to focus on the following 4 points, for a project to be Agile.

1. **Individuals and Interactions** over Processes and Tools

Although processes and tools help in successfully completing a project/product, it is people who undertake, participate in, and implement a project. The key actors in any project/product are the people, and the emphasis should be on them and their interactions.

1. **Working Software** over Comprehensive Documentation

While documentation is necessary and useful for any project, many teams focus on the interim deliverables although the real value is delivered to the customer primarily in the form of working software. Therefore, the focus is on delivering working software in increments throughout the lifecycle of the product.

1. **Customer Collaboration** over Contract Negotiation

Traditionally, customers were seen as outside players who are involved mainly at the start and end of the product lifecycle and whose relationships were based on contracts and their fulfillment. Agile believes in a shared value approach where customers are seen as collaborators. The team and customer work together to evolve and develop the product.

1. **Responding to Change** over Following a Plan

In the current market scenario, in which the customer requirements, the technologies available, and the patterns of business keep changing, it is essential to approach product development in an adaptive manner that enables change incorporation and fast product lifecycles rather than place emphasis on following plans formed on potentially outdated data.

## Principles behind the Agile Manifesto

[Ask the class participants to read them one by one in turn.]

Extra reading—Participants can be asked to read the next three pages as homework (i.e. the Traditional and Agile triangle through Agile methods.)

### What has changed?

**Agile Triangle**

**Traditional Iron Triangle**

Agile methodologies require a change in mindset from the traditional methods. While Waterfall methods focused on the Scope and used it to determine the Cost and Schedule, in Agile the focus is on Business Value and it is used to determine the Quality and Constraints of Development.

While the Waterfall model is very suitable for ordered/predictable scenarios, Agile methods are more successful in scenarios with chaos as they are Chaordic [chaos+order] methods. While Waterfall models believe in a rational agent model, Agile methods tend to be based on shared value approaches. Agile methods are based on inspect-adapt cycles as opposed to the command and control structures of the Waterfall method.

A comparison of Agile and Traditional methods:

|  |  |  |
| --- | --- | --- |
| **Approach** | **Agile** | **Waterfall** |
| **Emphasis** | People | Process |
| **Size** | Small/Creative | Large |
| **Domain** | Unpredictable/ Exploratory | Predictable |
| **Documentation** | Minimal | Comprehensive |
| **Process style** | Iterative | Linear |
| **Upfront Planning** | Low | High |
| **Perspective to Change** | Adaptability | Sustainability |
| **Management Style** | Decentralized | Autocratic |
| **Leadership** | Collaborative | Command-Control |
| **Performance Measurement** | Business Value | Plan Conformity |
| **Returns on Investment** | Early/throughout project life | End of project life |

3rd Sprint

1st Sprint

Analyzing

Designing

Testing

2nd Sprint

Release

5th Sprint

Analysis

Design

Test and Release

Implement

4th Sprint

**Time**

**Scrum Project Management**

**Traditional Waterfall method**

**Time**

**Functionality not Done**

As explained in the above diagram, Scrum projects are completed in an iterative manner wherein the functionalities with the highest business value are completed with top priority. Various cross-functional teams work in parallel across Sprints to deliver potentially shippable solutions at the end of every Sprint.

Because every Sprint results in an end-solution (which is a part of the overall product), there is a measurable objective that the team has to accomplish and this ensures that the team is progressing accordingly, and the project will be completed on time. Traditional methods do not present such timely checks and, therefore, result in situations in which the team might get off schedule and end up with a lot of work towards the end.

As the customer regularly interacts with the team, the work completed is regularly reviewed; thus, there is an assurance that the progress is as per customer specifications. However, in Waterfall, there is no such interaction as the work is carried out in silos and until the end of the project, there is no presentable functionality.

In complex projects, in which the customer is not clear of what the end product will be like and, therefore, the end product functionality keeps changing, the iterative model is more flexible in ensuring that these changes can be included and taken into consideration while working. The Waterfall method struggles to accommodate such changing functionalities.

However, in the case of simple projects with well-defined functionalities, and if the team has experience with previously completing such projects (and, therefore, estimation would be accurate), then the Waterfall method is successful.

## Agile Methods

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Core Values** | **Features** | **Roles** |
| **SCRUM** | Commitment  Focus  Openness  Respect  Courage | Iterative time-boxed model with self-organizing teams. Sprint Planning, Review, and Retrospective are part of the iteration. Daily Scrums for open communication. | Product Owner  Scrum Master  Scrum Team |
| **Feature Driven Development** | Plan, Design, and Build by feature | Individual class ownership, domain object modeling, feature teams, initial modeling, model storming configuration management, and regular builds. | Project Manager  Chief Architect  Development Manager Chief Programmer Class Owner  Domain Expert |
| **Extreme Programming** | Simplicity  Communication  Feedback  Courage  Respect | Focus on good practices in software development. Pair programming, test-driven development, collective code ownership, and refactoring. | XP Coach  XP Customer  XP Programmer  XP Administrator  XP Tracker  XP Tester |
| **Dynamic Systems Development Method** | Focus on the business need  Deliver on time  Collaborate  Never compromise quality  Develop iteratively  Build incrementally from firm foundations  Communicate continuously and clearly  Demonstrate control | Pre-project phase, Project life-cycle phase, and Post-project phase. Feasibility study, Business study,  Functional model iteration, Design and Build iteration, and Implementation. | Advisor User  Ambassador User  Analyst  Any Role  Architect  Developer  Executive Sponsor  Project Manager  Stakeholder  Tester  Visionary |

### Other Agile Methods

For faculty reference: To be discussed if inquired

1. **Lean product development (LPD):** LPD is the application of lean principles to product development; a cross-functional activity that seeks to uncover product knowledge hidden within the end-to-end production flow, typically in the hand-over points between functional units.
2. **Kanban Development:** Kanban is a method for developing software products and processes with an emphasis on just-in-time delivery while not overloading the software developers. It emphasizes that developers pull work from a queue, and the process, from definition of a task to its delivery to the customer, is displayed for participants to see.
3. **Agile Unified Process:** It describes a simple approach to develop business application software using Agile techniques and concepts yet still remain true to the Rational Unified Process. The AUP applies Agile techniques including test-driven development (TDD), Agile Modeling, Agile change management, and database refactoring to improve productivity.
4. **Essential Unified Process:** The Essential Unified Process for software development, or EssUP, was invented by Ivar Jacobson as an improvement on the Rational Unified Process. It identifies practices, such as use cases, iterative development, architecture-driven development, team practices, and process practices, which are borrowed from RUP, CMMI, and Agile development. The idea is that you can pick the practices that are applicable to your situation and combine them with your own process. This is considered an improvement with respect to RUP, because with RUP the practices are all intertwined and cannot be taken in isolation. EssUP is supported by a set of playing cards, each card describing a practice.
5. **Open Unified Process:** OpenUP preserves the essential characteristics of RUP/Unified Process, which include iterative development, use cases and scenarios driving development, risk management, and architecture-centric approach. Most optional parts of RUP have been excluded, and many elements have been merged. The result is a much simpler process that is still true to RUP principles. OpenUP targets small and collocated teams interested in Agile and iterative development. Small projects constitute teams of three to six people and involve three to six months of development effort.

## Scrum Overview

Hirotaka Takekuchi and Ikujiro Nonaka developed a new approach to product development that they called the holistic or rugby approach as it involved cross functional teams “going the distance as a unit, passing the ball back and forth.” They based their approach on manufacturing case studies from various industries.

Ken Schwaber, Jeff Sutherland, and others used similar methods for software product development and were the first to use the term Scrum in the early 1990s. They presented their approach at a conference held in 1995 in Austin, Texas. Scrum is the method of restarting play in rugby after the game has been stopped.

Scrum is an adaptive and iterative framework for managing software projects and application development. It was formulated as a faster and a more flexible approach to product development designed to deliver the greatest value in the least amount of time.

Scrum is the most popular of the Agile methods, and about half of all Agile projects use Scrum.

## Scrum Principles

1. **Empirical Process Control**

Scrum believes in making decisions based on observation and experimentation rather than detailed upfront planning.

There are two ways to control any process—defined process control and empirical process control. Empirical process control is based on the three main ideas of transparency, inspection, and adaptation. This approach is more appropriate for processes that generate unrepeatable and unpredictable outputs.

1. **Self-Organization**

As opposed to the traditional command and control style of management, Scrum believes that today’s workers have much more knowledge to offer than just their technical expertise and deliver greater value when self-organized.

It is a process in which global order arises due to the local interactions of the components of a system in disorder.

1. **Collaboration**

Scrum believes that product development is a shared value creation process that needs all the stakeholders working and interacting together to deliver the greatest value.

Collaboration is working together to achieve a common goal.

1. **Prioritization**

Delivering the greatest value in the shortest amount of time requires prioritization and dividing what will be done from what needs to be done.

Given the various constraints any team faces, prioritization becomes an important factor if the value delivered has to be maximized.

1. **Time-boxing**

Time is treated as a limiting constraint and time-boxing is used as the rhythm to which all work and contribute.

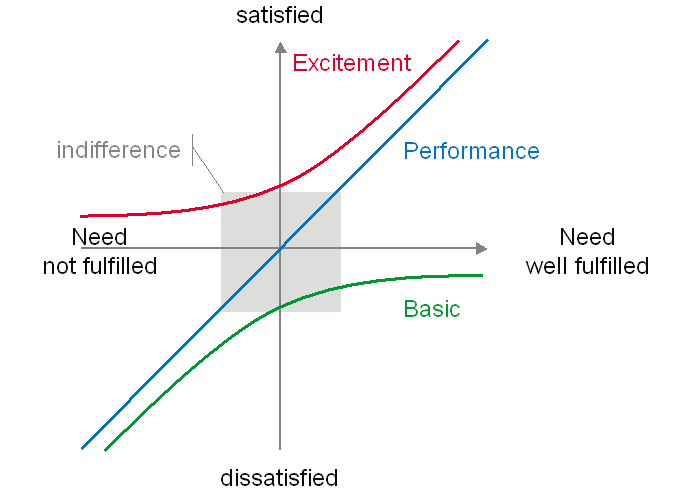
Scrum takes time as its primary constraint and, therefore, each process/activity of Scrum is allotted a fixed period of time.

A key principle behind Scrum is that the exact long term requirements regarding any project cannot be fully understood or defined especially in fast-changing volatile markets, and the focus is on making the team flexible enough to incorporate changing requirements. The traditional predictive methods of development (like Waterfall) cannot deal with such scenarios. Thus, Scrum is especially helpful for complex projects with greater uncertainty in which using long-term forecasting would be high risk. Scrum guides you through transparency, inspection, and adaptation to the most valuable business outcome.

**Kano Model**

Scrum derives its value perception based on the Kano model, which was developed in 1980s by Noriaki Kano, a Japanese professor of quality management at the Tokyo University of Science. It is used to map what is valued by the customer and tries to classify product attributes into Basic, Performance, and Excitement categories. It can be used to determine the minimally viable product that a customer will feel satisfies his basic requirements. The following graph is associated with the Kano model.

<graph from Wikipedia>



### Scrum Characteristics

* Scrum is one of the Agile processes of product development
* It is a framework structured to support complex product development
* Scrum consists of cross-functional, self-organized, and empowered teams
* Ensures transparency in communication across the organization
* Work is divided into a set of Sprints
* Emphasizes a working deliverable at the end of every Sprint
* Is only a framework and not the end solution

### Scrum Summary

**Scrum is most useful in projects involving…**

1. Development of cutting-edge technology products

In scenarios in which the product is innovative and any long-term estimation regarding the product requirements is uncertain, Scrum provides a more adaptive environment in which feedback can be incorporated in every new Sprint.

1. Highly qualified and dedicated cross-functional teams

Cross-functional teams are the essence of Scrum. Each Development Team is cross-functional to ensure a faster turnaround.

1. Development of products in hyper-competitive environments
2. Frequent and untimely change requirements

The iterative nature of Scrum helps handle projects in which the requirements might change frequently.

1. A regular need for feedback

Every Sprint has a Review Meeting and a Retrospective Meeting wherein the Product Owner provides feedback to the team.

It must be understood that Scrum is not the best alternative in all scenarios. For example, in high safety and critical systems in which all the variables are mapped out and such projects have been implemented before, the Waterfall method would be more appropriate.

## Advantages of using Scrum

* **Transparency:** Scrum ensures complete transparency across all its stakeholders. The customer is constantly aware of the product progress and the team members are also aware of their roles and expectations.
* **Collective accountability:** The Development Team is collectively responsible for ensuring that the work agreed upon for a Sprint is completed in a timely manner.
* **Adaptable:** Scrum provides an adaptive mechanism for product development wherein any change in requirements can be accommodated without impacting the overall project progress significantly.
* **Prioritized delivery:** Features delivering maximum business value can be prioritized on the Product Backlog.
* **Continuous progress:** Scrum ensures that the product functionalities are delivered on a continual basis, thereby ensuring steady progress.

## Chapter Quiz

1. Which of the following is not a common feature of adaptive project management
   1. Iterative product development
   2. High amounts of up-front planning
   3. Reduced time-to-market
   4. Flexible product delivery

Ans. B. High amount of up-front planning

Justification: Adaptive project management focuses on less up-front planning unlike waterfall methods. This creates a fairly adaptable environment for teams where they focus on only the immediate tasks at hand, complete them and then move to the next tasks. If there is any change in requirements, it is easily catered to over here.

1. You are the CEO of a company undergoing four different projects. Select the projects which you would implement the Agile philosophy
   1. Construction of a 5 floor residential block with 6 flats per floor
   2. Developing a new smartphone technology for a pre-defined market segment
   3. Developing a software technology for a customer for a change management exercise involving identifying the current as-is practices and developing a road-map for the to-be process depending on the vision of the management
   4. Constructing an automobile in a factory based on the developed prototype

Ans. b and c

Justification: The key to identifying the solution is to predict the amount of risk and uncertainty involved in the project. Routine projects with very specific requirements and lower degree of risk can be operated using the waterfall technique. Projects involving a high degree of complexity and those where there is a good chance of the requirements getting changed should be implemented using Agile and Scrum.

1. Which of the following are part of the values of the Agile manifesto
   1. Promoting Individuals over Processes
   2. Responding to change rather than long term planning
   3. Specialized teams rather than cross-functional teams
   4. Working software over comprehensive documentation

Ans. a, b and d

Justification: Adaptive Project Development focuses on the cross-functional nature of the teams. It is important to have cross-functional teams for clear and transparent communication and smooth project progress

1. As a Project Manager employing Agile practices in your organization, which of the following principles would employ.
   1. Business and technical employees working together
   2. It is always essential to provide attention to every minute detail of a software before sanctioning its release
   3. Maintain co-located teams thereby promoting face-to-face communication
   4. Welcome changing requirements, even late in development

Ans. a, c, d

Justification: Even though it is a good practice to try and deliver every possible detail, it is important to provide an acceptable level of detail in the given time (since Agile process are time-bound). Most of the times, the additional details consume a lot of time while providing very little additional benefits and delayed release of the product causes more harm

1. Identify the incorrect statement regarding Agile methodology
   1. Emphasis is on people rather than process
   2. Agile promotes minimal documentation contrary to the waterfall technique
   3. Agile recommends a Command based Leadership style
   4. Agile technique focuses on adaptability of the Project

Ans. c

Justification: Agile recommends Collaborative leadership style

1. Empirical Process Control is a Scrum Principle which
   1. Is used in cases where the inputs are clearly defined
   2. Focuses on providing control through frequent inspections and adaptations of processes that are imperfectly defined
   3. Is employed in situations where the processes generate unpredictable and unrepeatable outputs
   4. Is useful when a particular input always provides a specific output

Ans. b, c

1. All the following are part of the Scrum principles except
   1. Self-organization
   2. Time-boxing
   3. Task-prioritization
   4. Defined Process Control

Ans. d

1. Scrum projects are applicable in which of the following environments
   1. Developing cutting-edge products
   2. Frequently changing requirement
   3. Volatile and hyper-competitive markets
   4. Projects involving frequent need for feedback

Ans. All four are correct

1. Product development using SCRUM results in which of the following
   1. Transparency across the team and shareholders
   2. Adaptive product development environment
   3. Features offering the maximum business value are prioritized
   4. Cross—functional teams are used so that the work done during Sprints would not require client feedback

Ans. a, b, c

Justification: Scrum is designed in a manner that provides a transparent environment where the client can observe the work done every Sprint and continuously provide his/her feedback.

1. Prioritized delivery in SCRUM is
   1. A feature which is the simplest and would not require much involvement from the team will be completed first
   2. A feature with the least/no interdependencies will get completed first to ensure smooth and un-interrupted delivery
   3. A feature where tasks that provide maximum business value will be completed the earliest
   4. A feature where the tasks are completed on a first in first out basis

Ans. c

# Scrum Roles

## Overview

There are core and ancillary roles in every team. Deciding who is core and who is ancillary is not always easy. There is an often repeated story that will help you decide who is a core and who is ancillary.

The story is of a chicken and pig who decided to open a restaurant. The chicken was of the opinion that the restaurant should be called “Ham and Eggs.” The pig said, “In that case, no thank you. Because I would be committed while you would only be involved.”

There are three core roles in Scrum: The Product Owner, The Scrum Master, and the Development Team (also called the Scrum Team). These are the people responsible for completing the project objectives. The roles and responsibilities of the traditional project manager are divided among all the core roles.

1. **The Product Owner** is responsible for achieving maximum business value. The Product Owner is the voice of the customer and is generally from the customer side. The Product Owner is responsible for the delivery of the product and drives design, priorities, and reprioritization of the Product Backlog.
2. **The Development Team** is responsible for completing the product deliverable for the Product Owner. Scrum teams are cross-functional and self-organizing. The team decides the amount of work they will commit to in a Sprint. The Scrum team can consist of team members from different stages: development, quality assurance, business analysts, and so on.
3. **The Scrum Master** is a guardian and a facilitator who ensures that the team members are provided with the best environment to successfully complete the product development. The Scrum Master helps the team apply Scrum practices and protects the team from external impediments and helps maintain team effectiveness.

## Product Owner

The Product Owner (PO) represents the stakeholders and is responsible for ensuring that the team delivers value. The Product Owner drives the writing of User Stories with input from the team members and manages the Product Backlog. (In some cases, the team members write the User Stories under the supervision of the Product Owner.) The Product Owner is also responsible for ensuring clear communication of product functionalities to the Development Team and, therefore, is commonly called “the voice of the customer.”

This is perhaps the most misunderstood role in Scrum. The PO needs to understand the concept of the Sprint. Though the PO is not part of the development, he must be made to feel as though he is a part of the Scrum team and not alienated because of his business role.

### Voice of Customer (VOC)

* The customer is the most important stakeholder for any company, therefore it is extremely important to understand the customer’s needs and aspirations. Voice of the Customer refers to the stated or unstated needs of the customer, which should be understood very well before designing any product or service. Generally, in a Scrum environment, the Product Owner represents the Voice of Customer.
* For any Scrum project, the customers may be:
  + Internal (i.e. within the same organization)
  + External (i.e. outside the organization)

## Scrum Master

The Scrum Master is the person responsible for ensuring that the product development is progressing smoothly and the Development Team members have all the tools necessary to get the work done. The Scrum Master monitors the Release Planning and sets up and oversees meetings. The role of the Scrum Master is based on the concept of servant leadership in which leaders achieve results by giving attention to the needs of those they serve.

The Scrum Master also instructs everyone concerned about the values and methods of Scrum. This role is much more important and critical earlier on when transitioning to Scrum methods.

## 

## Scrum Team

## Scrum Team characteristics:

### Self-organized:

* Scrum promotes the concept of a self-organizing team, wherein the team has the collective ownership of a project.
* Team members are involved in all decisions related to the delivery of the project and choose how best to accomplish work with no interference from the outside world.

### Cross-functional:

* The Scrum team is cross-functional so that all skill sets and other requirements needed to accomplish the work are internally available without depending on any one who is not part of the team.
* The team model in Scrum is designed to optimize flexibility and productivity.
* A cross-functional team is more focused on a common goal.

### Colocation and Face-to-Face Communication:

* Scrum enables the creation of self-organizing teams by encouraging colocation of all team members and recommends face-to-face communication between all team members.
* A co-located team comprised of functional experts who collaborate to reach a common goal will succeed far more quickly than a team separated by function.

### Iterative Product Delivery:

* Scrum Teams deliver products iteratively and incrementally, maximizing opportunities for feedback.
* Incremental deliveries of “Done” product ensure that a potentially shippable version of a working product is always available.

## Team Selection

The Scrum Team (Development Team) is the core for any Scrum project implementation and getting the team selection right is important for maintaining the performance of the whole team. Scrum team members are generalist-specialists. (Have a general idea of various fields and be an expert in at least one of the fields—but beyond the subject-matter expertise, it is the people skills of team members that define the success of self-organizing teams.)

Ideal members of the Development Team should be independent, self-motivated, customer-focused, responsible, and collaborative. The team should be able to foster an environment of independent thinking and group decision-making in order to extract the best benefits of the structure.

When selecting teams, another important aspect is to create backups for every person. This ensures that there is no major loss of productivity due to losses of critical members. Every member of the team will back up a “specialist” member from one field, therefore enhancing the skill sets of team members.

### Advantages of Cross-functional Teams

Scrum uses a cross-functional team approach as it provides the following advantages:

* **Faster Decision Making:** A cross-functional team is small in size and consists of functional experts who can reach a common goal faster than the traditional function-oriented teams.
* **Improved Communication:** Cross-functional teams are generally co-located, which enables a smooth line of face-to-face communication among members of the team. Thus, the team interacts regularly leading to a smoother knowledge-sharing process.
* **Objective Orientation:** Cross-functional teams in Scrum are extremely focused on the desired result. The Scrum Team has a defined set of objectives for a \_\_ Sprint \_and is flexible to focus on a different set of objectives in the next \_\_\_\_ Sprint.
* **Collective Ownership:** The team as a whole is responsible for the delivery of results and any success or failure is at the team level.
* **Continuous Innovation**: The use of experts from various fields as part of a cross-functional team enables exchange of ideas, thereby fostering creative thinking.

## Chapter Quiz

1. Which of the following are core Scrum roles?
   1. Product Owner
   2. Stake holder
   3. Scrum Master
   4. Project Manager

Ans. a, c

1. Who amongst the following is responsible for providing the Scrum team with a favorable environment for the product development
   1. Scrum Master
   2. Product Owner
   3. Development Team
   4. External Stakeholders

Ans. a

1. Who amongst the following is responsible for creating an awareness regarding implementation of Scrum practices amongst the team members
   1. Scrum Master
   2. Product Owner
   3. Development Team
   4. Training department

Ans. a

1. Who amongst the following is responsible for achieving the maximum business value of the project
   1. Scrum Master
   2. Product Owner
   3. Development Team
   4. External Stakeholders

Ans. b

1. Who amongst the following is responsible for completion of work in a Sprint
   1. Scrum Master
   2. Product Owner
   3. Development Team leader
   4. None of the above

Ans. d

Justification: The scrum team is supposed to be a group of self-motivated individuals working together to reach the common objective. Scrum promotes team work and collective responsibility as well as discouraging hierarchy amongst development teams. Hence, the development team as a whole is responsible for completion of work

1. The advantages of using a cross-functional team are
   1. Improved communication between members of the team
   2. Faster decision making takes place due to the presence of members from different aspects of the Product Development in the same team
   3. Individual responsibility can be assigned to every member depending on their expertise
   4. Cross functional teams reduce dependency on an individual member of the team to complete the task from a specific function

Ans. a, b, d

Justification: Since Scrum focuses on collective ownership, the primary advantage of cross-functional teams is that the members of the team can develop an understanding of the other functions. However, the responsibility of delivery is with the whole team and not individuals

1. Who amongst the following is responsible for deciding on the acceptance criteria for various tasks
   1. Scrum Master
   2. Product Owner
   3. Development Team leader
   4. Development Team

Ans. b

1. Who amongst the following represents the voice of the customer
   1. Scrum Master
   2. Product Owner
   3. Development Team leader
   4. None of the above

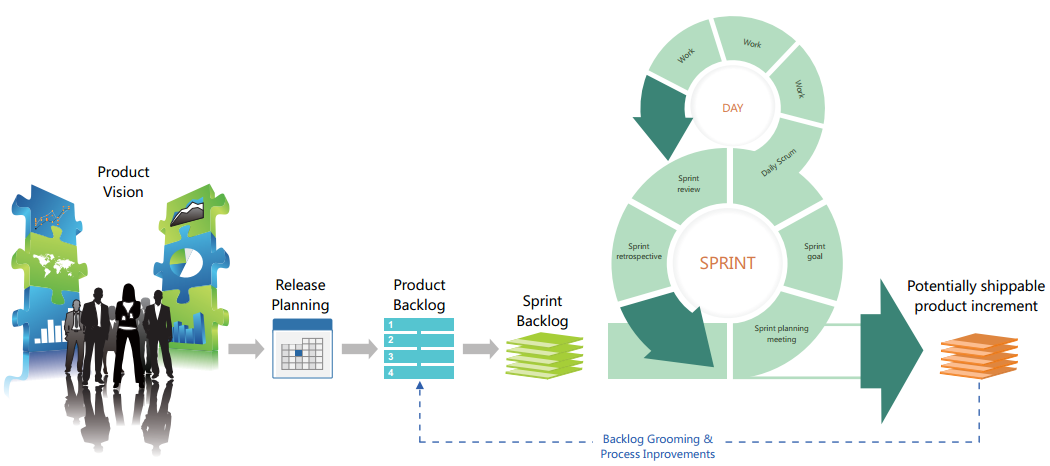
Ans. b

1. Who amongst the following is responsible for resolving conflicts among the development team members
   1. Scrum Master
   2. Product Owner
   3. Development Team leader
   4. Development team

Ans. d

# Planning in Scrum

## Scrum Flow



#### Note: The Daily Stand-up is more of a team activity, whereas the Product Owner is the main character in the Review Meeting.

## Pre-Project Meeting:

*A Scrum project starts with a Kick-Off Meeting during which the vision of the product is decided upon by the Product Owner and the management funding the project.*

The Product Owner is responsible for achieving maximum business value and takes input from management, and discusses the vision of the project and the intended deliverable. Then, the Product Owner develops the initial Product Backlog for the product. This Product Backlog consists of the list of requirements that have to be completed during the multiple Sprints. Until this stage, the Scrum Master and Team members were not involved. (Thus, the Kick-Off meeting includes primarily the PO —the team may not even be identified by this time—and the project sponsors).

(Note: This is not an official meeting per se, therefore at times is not included as a part of the Scrum flow. However, this does precede the actual Scrum; therefore, it has been included in this workbook).

Note:

These requirements are prepared by the Product Owner at this stage. Later on, the team provides its input and the feedback collected may be slightly altered or reprioritized accordingly. The timelines related to these items are also discussed with the team as part of the Sprint Planning Meeting.

After this process, the Scrum teams are identified and the Scrum process commences. The Product Owner meets the Scrum teams and defines the Product Vision and Project Objectives.

How do you write a Product Vision?

* The vision statement should not be too specific
* The vision should have room for flexibility (It is possible that the current understanding of the project may be based on some wrong technical assumptions that can change as we progress. It is important that the vision is flexible enough to accommodate these technicalities.)
* The Product Vision should focus on the problem rather than the solution

**Exercise to follow, refer case study**

## Product Backlog

*The Product Backlog is a list of requirements that, when turned into potentially shippable product functionality, will deliver the product vision.*

* The Product Backlog contains the list of work needed to accomplish the project. It is prioritized and the most valuable items (those offering maximum business value) are assigned top priority
* Only a single Product Backlog exists; this means the Product Owner is required to make prioritization decisions across the entire spectrum of work that could be done
* The Product Backlog generally contains User Stories but at times also contains information regarding key findings, bugs reported, functional and non-functional requirements, and so on
* Note: The priorities and groupings usually changes the moment the project starts and throughout the course of the project as the business requirements change
* A typical Product Backlog will contain the items to be worked upon (or User Stories), the initial time estimates and revised estimates based on task complexity, status of other higher priority tasks, and change in business requirements
* The items in the Product Backlog are allotted with points to indicate the relative time that could be taken to complete them
* The tasks requiring more time are further broken down into smaller ones during the Sprint Planning Meetings

It is the responsibility of the Product Owner to come up with the initial Product Backlog of requirements based on the Pre-project meeting. Based on the Product Vision statement, create a Product Backlog containing the User Stories (and a high-level estimation of the same). Note that this backlog will be further broken down into tasks in the Sprint Backlogs.

## Product Backlog Overview

## User Stories

A User Story is a statement (or a group of statements) that expresses the desired end user functionality. It usually is broken down into a sequential block of tasks. User Stories are developed by the Product Owner as part of the Kick-Off meeting. Like the Product Backlog, User Stories are refined with time.

* User Stories help enhance communication among stake holders. The requirements expressed in the User Stories are easy to comprehend and therefore reduce confusion regarding the requirements and also enable better estimations.
* User Stories also remove the need for the traditional long documents recording customer requirements. They are written in business domain language.
* User Stories are generally simple, short, and easy to implement. Longer User Stories are further broken down into multiple User Stories.
* The Product Owner has the responsibility of ensuring the creation of User Stories.
* User Stories are not necessary for implementing Scrum; instead of a well-defined User Story, it is also possible to just include simple sentences as Product Backlog items.

## Creating User Stories

* As mentioned before, User Story creation is the responsibility of the Product Owner. (Note: Generally the Product Owner creates the User Stories, but in some cases they are also developed by the Development Team in consultation with the Product Owner.)
* The Product Owner has to ensure that the business requirements from the customer are converted to functional requirements that the developer can understand and implement with ease.
* Generally, User Stories are written at the start of the project when all the requirements are defined broadly. Generally, these User Stories are too big to fit in a single Sprint. Such stories are called Epics.
* Once these Epics (large, un-refined Product Backlog items) come up in the Product Backlog as a high-priority task, they are then broken down into smaller User Stories in the Product Backlog Grooming exercise. Based on requirement changes, new User Stories are also continuously added.
* The User Stories are in the following format: As a \_\_\_\_ (user/Product Owner/ Developer) I want to be able to \_\_\_\_\_, so that I can \_\_\_\_\_\_.

Epic: An Epic is a large User Story, typically one that is too big to fit in a single Sprint. Epics need to be disaggregated into smaller User Stories at some point before implementation. Large story units are generally of low priority and will be broken down once they go up the Product Backlog. At times, the word “epic” is not identified by a few teams and it is basically an unrefined Product Backlog item.

Epic is not a standard definition and in a few cases, the word “epic” is not used. Teams just mention it as un-refined Product Backlog Items.

Theme: A Theme is a collection of related User Stories. Themes can contain one or more Epics. Multiple Themes can be associated with a product, but a Theme cannot be associated with more than one product at a time.

**Refer to Case study for exercise**

## 

## Release Planning

Before iterations (Sprints) can begin, a high level initial release plan needs to be defined.

The product should be released as early as possible but there must be enough value in the product in order to actually release it. The product owner defines a minimum marketable Feature set (MMF), also called First Feasible Delivery (FFD), the minimum functionality necessary in order to release the product for the first time. If the initial product backlog is prioritized correctly the MMP includes the top N features n the backlog.

The team provides high level estimates for the various User Stories. The length of the sprints is defined and also a target timeline leading to the first release.

The length of the sprints will influence the duration of later meetings (Sprint Planning, Sprint Review and Sprint Retrospective)

**Team velocity** is an important criterion in deciding timelines for the Release. Once the length of the Sprint is decided and estimation is done for the high-priority tasks, historical team velocity is used to identify how much work the team can complete in a particular Sprint. In case of a first-time project, the team provides a best guess of how much can be completed in a single sprint. Team velocity is adjusted throughout the project based upon experience gained in the project.

The planning can be done for the first release but also for potential further releases. If further releases are include the planning needs to be very minimal and high level in order to avoid waste.

### Guidelines for Release

The following guidelines should be followed while deciding on the Release output for each Sprint:

* Each iteration should have a potentially shippable functionality as its output. Each Release should provide significant value to the customer.
* Features with higher business value should be given higher priority in the Product Backlog and be developed first. This ensures that customers get the highest output from Scrum per their business needs.
* Interdependencies between features need to be considered in the planning. Also other preconditions for backlog items need to be considered and are included in the backlog as non-functional items. This ensures that every output received by the customer can be immediately used.

## Chapter Quiz

1. The prioritized set of work to be done is known as:
   1. User story
   2. Product Backlog
   3. Burndown chart
   4. Sprint Planning

Ans. b

1. Which is a user story:
   1. A statement describing the product vision
   2. A document describing the priority of tasks to be completed in the project
   3. A statement that expresses the desired end user functionality
   4. Stories providing information on similar tasks completed in previous Scrum implementation projects

Ans. c

1. How is the Product Backlog items arranged?
   1. Smallest items at the top
   2. Items with least interdependencies at the top
   3. Most complex items at the top
   4. Items with highest business value at the top

Ans. d

1. Who is responsible for creation of User stories?
   1. Scrum Master
   2. Product Owner
   3. Development Team
   4. Stakeholders

Ans. b

1. Epics are \_\_\_\_
   1. Un-refined, large user stories
   2. User stories with high business value
   3. Most important user-stories
   4. None of the above

Ans. a

Justification: Epics have to be refined and broken down into user stories as and when required

1. The responsibility for estimation of Product Backlog items lies with:
   1. The development team
   2. Scrum Master
   3. Product Owner
   4. External team with considerable experience

Ans. a

1. How often are the Product Backlog priorities changed?
   1. Never
   2. Whenever the Product Owner decides that an item has to be assigned higher priority
   3. Whenever the Scrum Master believes that an item has to be added
   4. When the senior management of the organization feels an item has to be added

Ans. b

Justification: The Product Owner has the finals say regarding any items that have to be added to the product backlog.

1. How frequently is release planning conducted?
   1. Prior to each release cycle
   2. Every Sprint
   3. After every Daily stand-up
   4. At the start of the project

Ans. b

1. Which of the following statements is not true?
   1. Team velocity is a measure of the amount of work that a team can complete in a Sprint
   2. Historical team velocity is used as an indicator when assigning tasks for any Sprint
   3. Team velocity is independent of team composition
   4. Team velocity is used in deciding release deadlines

Ans. c

Justification: Team Velocity is unique for a given team compositions. Even changing a person for any role could change the team velocity

# Sprint Planning

## Sprint Planning Meeting

Each Sprint starts with a Sprint Planning Meeting that is time-boxed to eight hours for a one-month Sprint. For shorter Sprints, the event is proportionately shorter.

The Sprint Planning Meeting is divided into two parts:

1. Objective definition
2. Tasks estimation

### Objective Definition:

* The Product Owner explains the top items in the Product Backlog to the teams.

Note: The Product Owner may also provide details from stakeholders about why these goals have been prioritized.

* The Scrum Team decides on the number of items selected from the Product Backlog for the Sprint Backlog in consultation with the Product Owner.

Note: Even though the Product Owner is consulted, the final decision regarding the work that the team can take on for the next Sprint lies with the team.

Based on team velocity, the Product Backlog Items for the next Sprint are selected.

Note: If this is not the first Sprint, then the team relies on the velocity known from previous Sprints. In case this is the first Sprint in the product development, the team relies on similar experiences from the past. If similar experiences are not available, then team members take their best guesses. These estimates become a benchmark for future Sprints.

* At the end of this section, the team commits to the Product Backlog Items to be done during the Sprint.

Note: Depending on the project characteristics it may be helpful to have some discussions close to the end of the previous Sprint. This will allow the Product Owner to address and communicate issues that can help the team understand and include high-priority backlog items that may have cropped up.

### Tasks estimation

* In this part, the team breaks down the selected Product Backlog Items into tasks.
* Then, the tasks are assigned estimates by the team members according to their complexity, risk involved, potential time required, and so on.

Note: Teams may prefer not to have the Product Owner as part of the estimate discussions, especially during the transition phase to Scrum. However, it is a good idea to have the Product Owner accessible to the team in case there are any doubts regarding the Product Backlog Items. Team resistance fades away over time. In some cases, the Product Owner is available at the meetings to ensure that he/she understands the estimates made by the team.

* These tasks are estimated using estimation exercises wherein the whole team is involved.
* The tasks are included in a Sprint Backlog that lists the work to be done by the team during the Sprint.

The output of the Sprint Planning Meeting is the Sprint Backlog.

### Sprint Backlog

* The Sprint Backlog consists of the tasks that a team selects for the current Sprint from the Product Backlog. The goal of each Sprint is to get the selected Product Backlog Items Done according to the defined Done criteria.

Note: Remind students that the results of a Sprint have to be potentially shippable.

* Each member of the Scrum team picks the tasks from the Sprint Backlog he/she wants to work on during the next day.

Note: Make it very clear that tasks are not assigned; neither by the Scrum Master nor by other team members.

* During a Sprint, it is possible that new tasks may be discovered that are required for developing the intended functionality. It is also possible that previously identified tasks may not be required. The Scrum team modifies the Sprint Backlog accordingly.

Note: Though some include a few additional tasks as back-up tasks during a Sprint, which can be taken up if the team completes the tasks allotted and has free capacity, it is a good practice to approach the Product Owner and discuss which additional Product Backlog Items can and will be added to the active Sprint.

Note: What happens when the Product Owner wants to make changes in the middle of a Sprint? The scope of the Sprint cannot be changed. If the change that needs to be made is so important that the results of the Sprint would be worthless without making the required changes, the Sprint should be cancelled. If not, they are pushed to a later Sprint.

### Sprint Planning Meeting Overview

## Estimation

Accurately estimating timelines and manpower requirements for delivery of solutions is a critical aspect for any company to be able to ensure a great project delivery experience for a client. However, estimates, by definition, are not accurate and the accuracy of the estimates vary with project complexity, team size, team availability, team priority, requirements volatility, etc. Scrum tries to ensure better estimation by using methods that involve the team and use relative comparison methods.

### Planning Game

Scrum Poker is a consensus-based technique for estimating, used to estimate effort or relative size of User Stories.

Note: Also called Planning Poker, it is a form of the estimation technique known as Wideband Delphi. It derives from the [Delphi method](http://en.wikipedia.org/wiki/Delphi_method) which was developed in the 1950-1960s at the [RAND Corporation](http://en.wikipedia.org/wiki/RAND_Corporation) as a forecasting tool. [Barry Boehm](http://en.wikipedia.org/wiki/Barry_Boehm) and [John A. Farquhar](http://en.wikipedia.org/w/index.php?title=John_A._Farquhar&action=edit&redlink=1) originated the Wideband variant of the [Delphi method](http://en.wikipedia.org/wiki/Delphi_method) in the 1970s. They called it "wideband," because the new method involved greater interaction and more communication between those participating.

Onepotential issue with consensus-based techniques for estimating is that they could lead to “Group think” wherein one or a few members of the team can influence the decision for the team. Scrum Poker overcomes this by forcing team members to think independently.

Each team member is assigned a deck of cards and each card is numbered in a sequence [for example Fibonacci sequence – 1, 2, 3, 5, 8, 13, 21, 34]. The numbers are used to represent the time complexity of the problem as estimated by the team member. For calibration, a task that is well known and understood by the whole team is first estimated and the below cycle is followed.

**Refer to Case Study for exercise**

## Product Backlog Grooming

In order to ensure that the prioritized items on the Product Backlog (for the next two to three Sprints) are refined into User Stories, it is recommended that from 7-10% of each Sprint must be used for refining the backlog. Product Backlog Grooming is mainly the responsibility of the Product Owner. He/she is responsible for adding/changing Product Backlog Items due to changed requirements and is responsible for providing more detailed User Stories that are usable for the next Sprint.

Grooming helps ensure that the refining is done well before the Sprint Planning Meetings so that the team has a well-analyzed and clearly defined set of stories that can be easily broken into tasks and subsequently estimated.

### Why Groom?

* Based on learning from an ongoing Sprint, there may be changes in the requirements or there may be a reprioritization that can be easily incorporated into the next Sprints.
* It enables the flexibility of the Scrum model by incorporating the latest business and technical insights for product development in the next Sprint.

### The Process:

**Analyzing customer requirements and feedback from the current Sprint**:

* Because customer requirements and expectations for complex projects are unclear, the Product Backlog is continuously changing. Changing customer requirements have to be analyzed before incorporation.
* Feedback from current software functionality also is used as an input while grooming the Product Backlog for future Sprints.

**Prioritizing the Product Backlog:**

* Based on input from the customer, external market changes, and learning from current and previous Sprints, the Product Backlog is reprioritized.

**Breaking down large tasks and Epics:**

* Based on priority in the Product Backlog, User Stories are created for the next Sprint. Larger items are broken into multiple User Stories so that they can be estimated.
* User Stories describe the functionality from a user’s point of view, are easy to understand, and can be reliably estimated.

Scrum does not fix time duration or slot for grooming exercises. Product Backlog Grooming is a continuous task for the Product Owner. To make changes to the backlog only right before the next Sprint Planning Meeting will guarantee bad User Stories and unclear requirements and priorities.

An important point to note is that any item on the Product Backlog is always open for re-estimation until it is accepted into a particular sprint. Thus, changes can be made until before the Sprint Planning Meeting if required.

### Product Backlog Grooming Overview

## Done Definition

* During the Product Backlog Grooming process, the epics are broken into User Stories that are worked on until they have presentable functionality. While working on the User Stories, the Product Owner defines what he/she considers an acceptance test criteria, which are the success criteria for each User Story.

* Thus, for any User Story to be considered as potentially shippable, it has to meet the acceptance test criteria of the Product Owner. Only then is the functionality considered “Done.”

The acceptance criteria of the Product Owner may include/imply the need to meet certain standards like CMMI or process criteria like ISO9001, TL9000, and so on, or some generic engineering practices.

* A clear definition of Done is critical as it helps remove ambiguity from the requirements and also helps the team adhere to the required quality norms.

Although the founders of Scrum intentionally refrained from prescribing engineering practices, many Scrum Teams have experienced success by pairing the framework with engineering practices from the eXtreme Programming movement (XP).

## Chapter Quiz

1. The advantages of Product Backlog Grooming are:
   1. Learning from a previous Sprint is incorporated for future Sprints
   2. Incorporates latest business and technical insights for next Sprint
   3. Ensure Product Backlog is refined before the Sprint Planning meeting so that the team has a better idea regarding the requirements prior to the meeting
   4. Removes requirement for a Sprint Retrospective meeting

Ans. a, b, c

1. The responsibility for Product Backlog Grooming lies with
   1. Product Owner
   2. Scrum Master
   3. Development Team
   4. External Stakeholders

Ans. a

1. What are the two primary tasks involved in a Sprint Planning meeting?
   1. Tasks estimation
   2. Product Backlog Grooming
   3. Objective definition
   4. Re-prioritization of tasks

Ans. a, c

1. Which of the following is not an activity in the Sprint Planning Meeting
   1. Product owner explains tasks to the team
   2. The development team, in consultation with the Product Owner, estimates the tasks for a given Sprint
   3. Based on the estimation, the team commits to a few tasks to be completed in the next Sprint
   4. The team gets feedback from the stakeholders

Ans. d

1. You are a member of the development team and you are instructed by the GM of your vertical to work on a very urgent task which is not part of the current Sprint. What will you do?
   1. Take up the task and tell the Product Owner to postpone the deadline for the current Sprint
   2. Talk to the Scrum Master and tell him/her to re-allot your tasks to someone else
   3. Talk to the Product Owner and tell him/her to re-allot your tasks to someone else
   4. Inform the Product Owner of the situation and let him/her take up the discussion with the GM
   5. Inform the Scrum Master of the situation and let him/her take up the discussion with the GM

Ans. d

1. During a Sprint, the team discovers an additional task critical to the current software deliverable
   1. The task is added to the product backlog and carried out in the next Sprint, the current deliverable is presented minus that portion
   2. Since the task is extremely critical to the software delivery, the team completes the other tasks and adds this item as the first one to be completed in the next Sprint. Upon completion of the said task, the deliverable is presented to the Product Owner in the next Sprint Review Meeting.
   3. The task is added to the current Sprint backlog as it is critical to the current release and it is compulsory for the team to provide implementable product functionality at the end of every Sprint. (To cater to such issues in the future, a small portion of the Sprint is assigned for such contingencies)
   4. None of the above

Ans. c

Justification: At the end of every Sprint, a presentable and working functionality has to be presented and if this task is critical to the functionality, it is to be taken up in the additional time available and at the cost of some less important functionality if required.

1. For any ongoing Sprint, when can new tasks be added?
   1. When the Scrum Master adds it to the Sprint Backlog and allots the responsibility to a team member
   2. Upon approval from the Product Owner
   3. Upon approval from the Development Team
   4. Never

Ans. d

Justification: The tasks of a current Sprint are never changed. If the requirement is extremely urgent, the current Sprint tasks are cancelled and a new Sprint begins. However, if the team is able to complete tasks in a particular Sprint ahead of time, it keeps a list of additional tasks which it can then take up.

1. Acceptance Criteria
   1. Reduces ambiguity regarding requirements
   2. Helps the team adhere to the quality norms regarding the functionality
   3. Is decided by the Scrum Master in consultation with the team members
   4. Is decided upon by the Scrum Master in consultation with product owner

Ans. a, b

1. What is the duration of a Sprint
   1. 1 month
   2. 2 months
   3. 1 week
   4. Generally between a week to a month (varies according to the project, the team size, team velocity and availability of the team members)

Ans. d

# Daily Scrum

## Meeting: Daily Stand-up/Daily Scrum

The Daily Scrum meeting is a short, time-boxed meeting (generally 15 minutes) for the team members to communicate their work statuses and their plans for the next 24 hours to the rest of the team. This is done by inspecting the work completed since the last Daily Scrum and forecasting the work that could be done before the next one.

One by one, each member of the team reports three (and only three) things to the other members of the team:

1. What have I completed yesterday?
2. What will I complete today?
3. Are there any impediments in the way?

The Scrum Master helps team members resolve issues or impediments. The Scrum Master is only a facilitator. The Scrum Master does not run the meeting. The Daily Scrum is a forum for information exchange only. Any discussion or issue resolution, if required, takes place after the meeting.

### Daily Stand-up Overview

### Artifact: Sprint Burndown Chart

The Sprint Burndown Chart is a Scrum artifact and a tool used by Scrum teams to keep track of their progress in the sprint. It also provides the stakeholders a snapshot of the progress during a Sprint.

* It shows the amount of work (as estimated) remaining (on Y-axis) across time (on X-axis).
* At the end of every day, the Scrum Team deducts the amount of work completed as they complete their current tasks in the Sprint Backlog.
* The Sprint Burndown Chart is a good indicator of team velocity in the active Sprint and allows a projection of whether or not the team will be able to complete all committed User Stories.
* The blue line indicates an ideal trend, and the red line plots the actual trend. The region above the blue line is the danger zone indicating a lag in task completion.
* Note: A burndown chart is prepared for a single Sprint (Sprint Burndown Chart) but can be prepared at the project level also (Release Burndown Chart).

Sample burndown chart

Note: The blue line does not need to be a straight line. In practice there will be vacation and other factors that influence the availability of team members on a daily base. If known at the beginning of the Sprint, they should be factored in for the amount of work the team commits to for the Sprint and also be reflected in the blue line (planned work remaining).

## 

## Sprint Review Meeting

* At the end of the Sprint, a Sprint Review Meeting is held to review the work completed during the Sprint.
* The purpose of the Sprint review is for the Team to present to the Product Owner the end-deliverable of the Sprint.
* This is done to demonstrate and validate that the deliverable is Done (according to the defined acceptance criteria).
* It also helps the team to get feedback from the stakeholders.
* The outcome of the Sprint review is either the acceptance or rejection of Done backlog items by the Product Owner.
* Items not accepted remain in the Product Backlog.

Note: It is the responsibility of the Scrum Master to ensure that the Product Owner does not change requirements or acceptance criteria during the Sprint review and rejects a Done backlog item because it does not meet the changed requirements. If the requirements have changed a Product Backlog Item needs to be created to address the changed requirements in a future Sprint.

Only the product functionalities that adhere to the definition of Done can be presented, WIP functionalities are included in a future Sprint. Team members present the work completed during the Sprint, answer stakeholder questions, and note their suggested changes.

### 

### Sprint Review Meeting Overview

## 

## Sprint Retrospective Meeting

Following the Sprint Review, the team gets together for the Sprint Retrospective Meeting. This is an opportunity for the Scrum Team to take a step back to introspect the previous Sprint and identify potential improvements in the process. These improvements can result in a simple agreement between the team members to do things differently going forward, but also in defining non-functional items in the Product Backlog.

* Note: Non-functional requirements can be discussed briefly. The Retrospective Meeting is attended by the Scrum Master and the team members. The product owner can attend, but does not have to.
* The team discusses what went well during the past Sprint and what did not.
* The team tries to raise any issues they faced during the past Sprint, and how they can address those issues.
* The team identifies potential improvements to its functioning.
* The team also reviews and tries to improve the definition of ‘Done’.

Note: This is basically the scrum version of a “lessons learned.” It is important to identify not only the areas for improvement but also what went well. The latter is not only important for team spirit but also ensures that best practices are not abandoned.

## Chapter Quiz

1. Which of the following is NOT discussed in a Daily Stand-up
   1. What was done by a member since the last meeting
   2. What the member plans to do till the next meeting
   3. What are the learning that the member has gained from his/her work
   4. Any impediments faced by the member

Ans. c

1. What is the duration of a typical Stand-up meeting
   1. 5 minutes
   2. 1 hour
   3. 15 minutes
   4. Stand-up meetings aren’t time boxed

Ans. c

Justification: It is time-boxed to 15 minutes for the general recommended team size (5-9 members) but might vary proportionate to team size

1. Conducting the Daily Stand-up is a responsibility of
   1. Scrum Master
   2. Product Owner
   3. Development team lead
   4. Collective responsibility of the group

Ans. a

1. Sprint Burndown Chart is a tool used by teams to
   1. Measure the work completed by them during a Sprint
   2. Measure the work left to be completed during the Sprint
   3. Calculate team velocity
   4. Identify the low performing team members

Ans. b, c

Justification: The burndown chart measures the work left to be completed rather than work completed. The amount of work done in a Sprint can be used to calculate the team velocity

1. The meeting in which the team discusses the tasks to be completed in the next Sprint is known as
   1. Product vision meeting
   2. Sprint Planning meeting
   3. Sprint Review meeting
   4. Sprint retrospective meeting

Ans. b

1. The meeting in which the team reviews the work done by the team presents the end functionality of the previous Sprint to the Product Owner is known as
   1. Product vision meeting
   2. Sprint Planning meeting
   3. Sprint Review meeting
   4. Sprint retrospective meeting

Ans. c

1. The meeting in which the team analyses the previous Sprint and identifies potential process improvements is known as
   1. Product vision meeting
   2. Sprint Planning meeting
   3. Sprint Review meeting
   4. Sprint retrospective meeting

Ans. d

1. In the Sprint Review meeting
   1. The team presents the end product to the Scrum Master
   2. The Scrum Master can accept/reject the end product
   3. The team presents the end product to the Product Owner
   4. The Product Owner can accept/reject the end product

Ans. c, d

1. Which of the following activity is not a part of the Sprint Retrospective meeting
   1. The team discussed the positives and negatives from the previous Sprint
   2. The team also identifies issues faced in the previous Sprint and discusses how to mitigate these issues for future Sprints
   3. The team reviews and identifies potential improvements to their functioning
   4. Based on the new suggested changes, the team proceeds to re-prioritize the Product backlog if required

Ans. d

# Implementation

## Scrum for Large Projects

Large projects require multiple Scrum Teams to work in sync to ensure project progress. Because the teams work in parallel, ensuring that all teams progress uniformly is critical with respect to deadlines. At times, future tasks of one team may depend on timely delivery of a previous task by another team. Also, there has to be a communication mechanism to ensure that the teams are in sync and any findings from one of the teams related to any part of the project is communicated across the teams effectively.

This is typically done through the Scrum of Scrums. This is mostly analogous to the Daily Scrum. This meeting is facilitated by the Chief Scrum Master. The Chief Scrum Master could also be the one to address impediments that impact more than one team.

There are four questions to be asked:

* + What has your team been working on since the last meeting?
  + What is your team going to do until the next meeting?
  + What impediments is your team facing?
  + Have you made any decisions that could impact other teams?

Immediately following the Scrum of Scrums, the opportunity to have discussions about problems that arise should be used. It is typically difficult to get all such people together at one time. All interested parties should stay after the meeting to discuss. Who should stay should be agreed upon as a consensus of the team. It is typically useful to hold a Scrum of Scrums following the teams’ retrospectives to discuss process improvements that may be useful to other teams.

### The Chief Product Owner

Generally for large projects, the client has a Chief Product Owner (generally a VP level person) who is effectively the Product Owner of the whole product. The Chief Product Owner sets the product vision, monitors the strategic objectives, and is responsible for resource allocation.

Note: The responsibility for the resource allocation could also be with another stakeholder (e.g. VP of engineering).

The Chief Product Owner’s responsibilities to product delivery are similar to those of the Product Owner for their respective Scrum Teams. The Chief Product Owner is responsible for the success or failure of the whole product and owns the overall Product Backlog.

The Chief Product Owner sets the priorities for the entire project. These Product Backlog Items are then divided among the various teams in consultation with the Product Owners. The Chief Product Owner (CPO) establishes the minimum Done Criteria to which all teams must adhere. Single teams can add anything they deem important but must not omit anything. The overall Product Backlog is divided into sub-parts for each team. This person has the final authority on prioritization of Product Backlog Items.

## 

## Transition to Scrum

The two basic methods of transitioning to Scrum are top down and bottom up. In top down, the transition is widely communicated. There is an effort to provide education about the change to everyone involved. This communication can be a source of change resistance. The other possibility is to change things more gradually within the organizational culture. Then, the transition to Scrum will be a more incremental one.

Note: With a top down introduction there may be communication issues. In one scenario, the head of engineering pushed Agile into his organization without buy in from product management or sales. This led to major issues with the Product Owner role itself, as well as Product Owner tasks such as Product Backlog creation.

Another aspect of transition to consider is how much of the organization to change to a Scrum methodology. The entire organization could be transitioned at the same time. However, this method is more susceptible to problems that may result in interruptions to profit generating activities. Therefore, it is more advisable to transition different divisions in an iterative fashion in order to reduce risk and to provide lessons learned for further iterations.

* Keep a backlog of Scrum elements to be introduced and prioritized
* Which ones give you the biggest bang for the buck?
* Which ones can be easily introduced?
  + Can you have cross-functional feature teams?
  + Can you have increments to get to your release?
  + Can you start with one or a few co-located teams?
  + If you cannot do continuous integration, can you increase the frequency of your builds?

#### Resistance to Change

As with any change, there will be some resistance to the transition to a new methodology. Middle-managers will typically fear their loss of role or loss of authority. They may not understand their new role or how they contribute to the success of the team. People who are invested in the old methodology will also typically resist the change. These people may be Systems Engineers or other related positions that fear their loss of role as a result of changing to Scrum.

## Mapping Traditional Roles to Scrum

The traditional role of Project Manager is divided among three roles in Scrum: Product Owner, Scrum Master, and Team Members. The Product Owner is the outward facing role. The Product Owner communicates with stakeholders and sets the priorities for the project. The Scrum Master performs the duties of the Project Manager of removing impediments to the progress of the team. The Scrum Master, like the Project Manager, is responsible for ensuring that the process is followed. The third role of Team Member also performs some of the traditional roles of the Project Manager in that they manage themselves and their own time. This allows the teams to gain a sense of ownership for their own success.

Note: In some companies the traditional role to ensure adherence to processes is within a separate quality assurance organization or department.

## Distributed teams in Scrum

Scrum requires a high level of communication between members of the Development Team and the Product Owners and Scrum Masters and, thus, prefers teams to be co-located. Practically, many organizations work on an offshore model wherein the Development Team is not onsite with the Product Owner.

* Appropriate infrastructure has to be made available to ensure that regular communications take place amongst the team members.
* Whenever possible, a distributed team should have an in-person kick-off meeting at a common location. This will help establish the basis for future communications.
* It is also a best practice to have different team members relocate to another site to act as cultural liaisons to ensure proper communication throughout the project. This can be the same person for the length of the project, or the role may be rotated among team members. In any case, frequent communication should be facilitated whenever possible.
* There can also be situations in which the Development Team is distributed geographically and this leads to difficulties in synchronization and communication. In such a situation make sure of the following:
* Keep the Scrum Teams to the proposed size of six to ten.
* Do daily team meetings. As Daily Stand-ups are not possible, they need to be done electronically. Agree upon an acceptable time for everybody (e.g. early morning for America, mid-day for Europe, and evening for Asia Pacific).
* Make sure to have the communications infrastructure to support such meetings.
* Have a Product Owner at each site to avoid feelings of abandonment and to improve communication.
* Have a Scrum Coach (a real Scrum Expert) at each site (can be a Scrum Master of one of the teams).

## Maintaining Stakeholder Involvement

An important aspect of Scrum is that it requires complete support from the stakeholders on a project. The following steps are recommended as a means for maintaining involvement:

1. Form a working agreement among all stakeholders to promote effective collaboration and participation of stakeholders on project activities.
2. Continually assess the changes in the project that affect the stakeholder landscape in order to ensure that new stakeholders on the project are appropriately engaged.
3. Communicate team progress and development capabilities in order to help the business stakeholders make informed decision about Scope, Time, and Cost.
4. Manage stakeholders’ expectations around minimal/most likely/optimal project outcomes, balancing accuracy and precision, so stakeholders have assurance that those outcomes will help them meet their business objectives.

## Importance of Executive Support

Executives are the people funding the project and, thus, for any Scrum project to be successful, it requires the support and backing of the executives. The following are key to maintaining executive support:

1. Communicate regularly with executives.
2. Keep executives aware of the latest progress.
3. Inform executives of any issues and potential lags in project delivery to minimize shocks.

It is important that the executives funding the project have clarity over the following issues:

1. What is the benefit of implementing the Scrum initiative?
   1. How does this initiative create benefits/prevents losses for the organization?
   2. How is being adaptive essential in the current business climate?
2. What are the deadlines and costs around the initiative?
   1. What is the expected completion time and cost for the initiative (best case/optimum/ worst case estimates)?
   2. What are the potential milestones on the way to completion?
   3. How frequently are executives going to meet regarding project progress?
3. What are the risks involved in the initiative?
   1. What are the potential roadblocks or risks in the implementation?
   2. What is the additional cost and time required to mitigate each of these risks?

## Chapter Quiz

1. What are the challenges with implementing Scrum in Large projects?
   1. Multiple teams required to work in sync
   2. Lack of an overall product backlog to track progress
   3. Additional level of hierarchy in the form of the Chief Product Owner
   4. Task interdependencies between teams

Ans. a, c, and d

Justification: There is an overall product backlog which is maintained by the Chief Product Owner to track task progress. Work is divided among the teams from this backlog.

1. Which one of the following is discussed as part of the Scrum of Scrum meetings?
   1. Overview of the tasks to be completed in the Sprint
   2. How any of your decisions could impact other teams
   3. Discussions around problems that may have come up for any of the Scrum teams
   4. Resolving impediments that may have come up

Ans. b

Justification: The Scrum of Scrum meetings is just like the Daily Stand-up meetings. Teams discuss the tasks completed post the previous such meeting and before the next such meeting. There is no discussion around tasks to be completed in next Sprint, nor are the issues discussed at length or impediments resolved as part of the meeting. The only topic discussed is if any of the decisions taken by a team are likely to affect other teams and if so, how.

1. Which of the following statements is/are false regarding Scrum in large teams
   1. Large teams have Chief Scrum Masters and Chief Product Owners to monitor progress of the project
   2. Chief Product Owner is responsible for resource allocation
   3. Chief Product Owner facilitates the Scrum of Scrum meetings
   4. Chief Scrum Master is responsible for the success or failure of the project

Ans. c and d

Justification: The Chief Scrum Master facilitates the Scrum of Scrum meetings while the Chief Product Owner is responsible for the success or failure of the project

1. Which of the following does not fall under the responsibilities of the Chief Product Owner
   1. Communicating the Sprint goals and objectives to members of the Scrum teams
   2. Deciding on the product vision and strategic objectives
   3. Establishing the minimum done criteria for the teams
   4. Maintaining and prioritizing the Product Backlog

Ans. a

Justification: Communicating Sprint goals and objectives among Scrum teams is the responsibility of the individual Product Owners and not the CPO