NAME:HASHAM UDDIN

CLASS: BSSE

SECTION: A

ROLL NO:21

SUBJECT: ISE(INTRODUCTION TO SOFTWARE ENGINEERING)

AGILE METHODOLOGY

Q1. State the 10 other things other than PLAN on which Project Management depends on?

Ans. Followings are the other ten factors other than PLAN in which Project Management depends on:

- Project Integration Management
- Project Scope Management
- Project Time Management
- Project Cost Management
- Project Quality Management
- Project Human Resource Management
- Project Communications Management
- Project Risk Management
- Project Procurement Management
- Project Stakeholder Management

1. Project Integration Management

Project integration management is the umbrella that covers all your other project management knowledge areas. It knits together all your individual processes and tasks into one project with defined goals and deliverables.

If you're looking at the big picture and how your project fits into your larger organization, this is the project management knowledge area you need.

Because this is the broadest area, you may want to save it for last or at least revisit it at the end of your project plan.

2. Project Scope Management

How many times have you started a project just to have extraneous tasks slipped in, making your completion times creep? This is why project scope must be well-defined and defended throughout the process.

As you complete your scope process groups, you'll create a management plan that defines, validates, and controls scope. These processes will ensure you stay on task and that everyone, including the project requester, understands what tasks will be included in the project to prevent frustrating changes and unmet expectations.

3. Project Time Management

Nearly all projects rely on several different timelines and the schedules of multiple people. Some team members may overestimate how much time it will take to complete a project in order to leave a cushion and not feel hurried. Others may underestimate their time. And, of course, unexpected problems will throw off your timeline as well.

But, these variables are exactly why effective time management is so critical.Your plans will determine which tasks can be adjusted and

how the team’s resources will be allocated and managed throughout the project. When those tricky problems surface, you’ll be glad to have a plan to refer back to and quell the panic.

4. **Project Cost Management**

With or without a budget, your project will cost money. Keeping costs low or at least at an expected or reasonable level is a fundamental part of showing ROI on a project. After all, if you can’t definitively lay out how much a project will cost, how will you be able to quantify if you’ve made any money?

Your role in cost management isn’t just a one-and-done task of creating a budget. You’ll need to continuously evaluate your costs to avoid any surprises at the end of a project.

5. Project Quality Management

In project management, quality isn’t the same as perfection. It’s not practical to spend the time and resources to take a project to perfection; and in many cases, that’s not even attainable. The goal of project quality management is to achieve consistency across your projects.

If you know and understand the expectations of your stakeholders and have created reasonable agreements with them and your team, quality control will ensure you’re delivering great work every time.

If you notice projects aren’t meeting results, you can adjust course and implement changes to the process or product to get back on track.

6. **Project Human Resource Management**

Working with people is part of the reason you signed up for project management, right?

One of the most rewarding parts of this process is creating teams that click and helping individual team members grow and learn new tasks. That's why this project management knowledge area is more than just setting schedules and assigning tasks.

Effective resource management requires you to know and work with the bandwidth of your team, identify their individual strengths and weaknesses, and their synergy with other team members.

And, back to that part about helping team members grow. You should also identify knowledge gaps and opportunities for continued training for individual team members and the entire team based on current and upcoming projects. You'll set your team up for success and increase commitment as you invest in their skills and growth.

7. **Project Communications Management**

How many times have you heard the phrase, “Keep me in the loop?” And yet, when changes happen, maybe important stakeholders were left out? There is a fine line between under and over communication and your communications management plan is crucial to identifying who needs to know what and when before your project starts.

8. Project Risk Management

The truth is that no project goes off without a hitch, and it’s unrealistic to look at a project and assume that everything will go smoothly.

If you can manage your firefighting by identifying major project risks and the mitigation plans associated with them, your team and project requesters will be prepared and more forgiving when issues in a project come up.

As an added bonus, you’ll have the benefits of time and energy upfront rather than trying to

troubleshoot at the eleventh hour when your team is stressed and up against a deadline.

9. Project Procurement Management

In some cases or areas of a project, you won't have the resources or team members in house to complete a task. If you hire contractors or vendors to take on certain tasks, you'll want them to be seamlessly integrated into the team.

This project management area gives the blueprint for which tasks or services will be completed by outside contractors. It also builds and plans the legal paperwork and coordination process ahead of time.

This may not be a project management knowledge area you use every time or even very often, but it's incredibly valuable when you do need it.

10. Project Stakeholder Management

Ultimately, the success or failure of a project depends on the delivery of your project to the stakeholders. But, who are your stakeholders?

Stakeholders include not only the project requester, but also team members who have worked on the project, contractors, suppliers, customers or the public, and many other people internal and external to the organization.

Not all stakeholders are equal in the eyes of the project. Identifying who is a stakeholder in a project and how they are involved in the process will make sure everyone gets the information they need to know—no more, no less.

Techniques of managing projects will vary depending upon the kind of stakeholders for the projects. In case a project has multiple stakeholders from different backgrounds, there is a possibility of disagreement between

them. In such cases, project management becomes extremely challenging as you cannot afford to have unhappy stakeholders and clients. Great convincing and negotiation skills are required in such cases to reach a consensus. It can be time consuming and hence the actual time dedicated to resources will reduce. The project manager needs to adopt tactful approaches in such cases and get the work done.

Q2)DIFFERENCE BETWEEN MICRO MANAGEMENT AND TRIPLE M?

ANS) micro-management is a management style whereby the manager very closely observes the work of employees, often scrutinising work or going over it with a fine-tooth comb. In general, micro-management is frowned upon, and businesses have put specific measurements in place to stop this from happening. The controlling aspect of this management style can cause great levels of distress to employees.

On the other hand, macro-management is where the boss takes a more “hands-off” approach. Though they’re still in regular contact with their employees about their work, employees are free to go about tasks with much less supervision than would be the case with a micro-manager. More businesses lean towards the macro-management approach, though there are still a few who do have micro-managers

Q3. List different agile methodology?

The most widely-used Agile methodologies include:

- Agile Scrum Methodology
- Lean Software Development
- Kanban
- Extreme Programming (XP)
- Crystal
- Dynamic Systems Development Method (DSDM)
- Feature Driven Development (FDD)
- Test Driven Development (TDD)

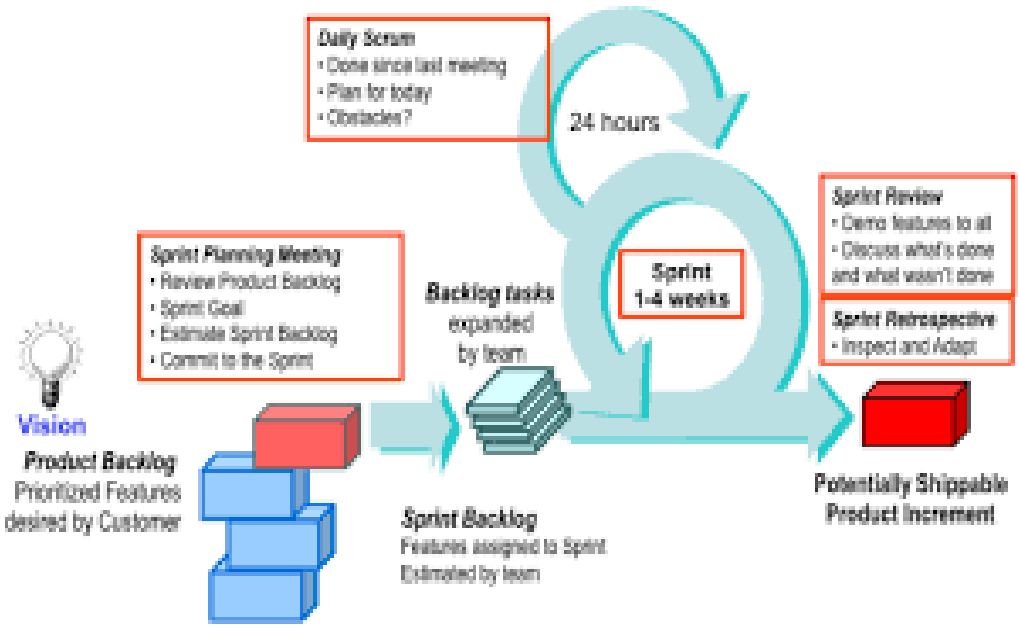
- ❖ Continuous integration/continuous delivery (CI/CD)

AGILE SCRUM METHODOLOGY:

Scrum is a lightweight Agile project management framework that can be used to manage iterative and incremental projects of all types. It has gained increasing popularity over the years due to its simplicity, proven productivity, and ability to incorporate various overarching practices promoted by other Agile models.

In Scrum, the Product Owner works closely with their team to identify and prioritize system functionality by creating a Product Backlog. The Product Backlog consists of whatever needs to be done to successfully deliver a working software system, including features, bug fixes, non-functional requirements, etc. Once priorities have been established, cross-functional teams will estimate and sign-up to deliver “potentially shippable increments” of software during successive Sprints, typically lasting 30 days. Once a Sprint’s Product

Backlog is committed, no additional



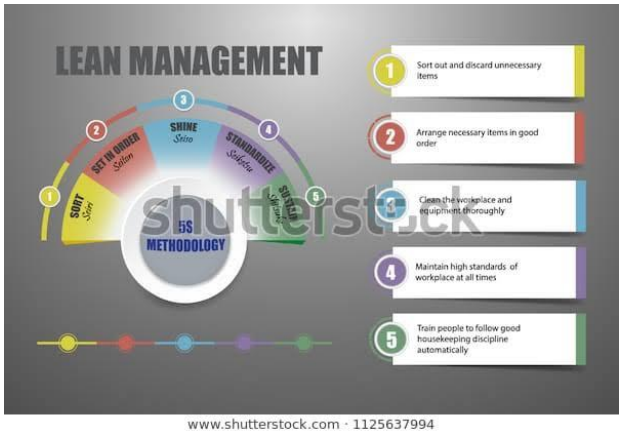
functionality can be added to the Sprint except by the team. Once a Sprint has been delivered, the Product Backlog is analyzed and reprioritized, if necessary, and the next set of deliverables is selected for the next Sprint.

Lean Software Development:

Lean Software Development is an iterative Agile methodology that

focuses the team on delivering value to the customer through effective value stream mapping. It is a highly flexible, evolving methodology without rigid guidelines, rules, or methods.

Principles:



The main principles of the Lean methodology include:

- ❖ Eliminating Waste
- ❖ Amplifying Learning
- ❖ Deciding as Late as Possible
- ❖ Delivering as Fast as Possible
- ❖ Empowering the Team
- ❖ Building Integrity In
- ❖ Seeing the Whole

Lean development eliminates waste by asking users to select only the truly valuable features for a system, prioritize those features, and then work to deliver them in small batches. It relies on rapid and reliable feedback between programmers and

customers, emphasizing the speed and efficiency of development workflows. Lean uses the idea of a work product being “pulled” via customer request. It gives decision-making authority to individuals and small teams since this has been proven to be a faster and more efficient method than a hierarchical flow of control. Lean also concentrates on the efficient use of team resources, trying to ensure that everyone is as productive as possible for the maximum amount of time. It strongly recommends that automated unit tests be written at the same time the code is written.

Kanban:

Kanban is a highly visual workflow management method that is popular among Lean teams. In fact, 83% of teams practicing Lean use Kanban to visualize and actively manage the creation of products with an emphasis on continual delivery, while not adding more stress to the software development life cycle. Like Scrum, Kanban is a

process designed to help teams work together more effectively.

Principle:

Kanban is based on 3 basic principles:

- Visualize what you'll do today (workflow automation): Seeing all the items within the context of each other can be very informative
- Limit the amount of work in progress (WIP): This helps balance the flow-based approach so teams don't start and commit to too much work at once
- Enhance flow: When something is finished, the next highest priority item from the backlog is pulled into play

Kanban promotes continuous collaboration and encourages active, ongoing learning and improvement by defining the best possible team workflow.

Extreme Programming (XP):

Extreme Programming (XP), originally described by Kent Beck, has emerged as one of the most popular and controversial Agile models. XP is a disciplined approach for high-quality agile software development, focused on speed and continuous delivery. It is intended to improve software quality and responsiveness in the face of changing customer requirements. It promotes high customer involvement, rapid feedback loops, continuous testing, continuous planning, and close teamwork to deliver working software at very frequent intervals, typically every 1-3 weeks.

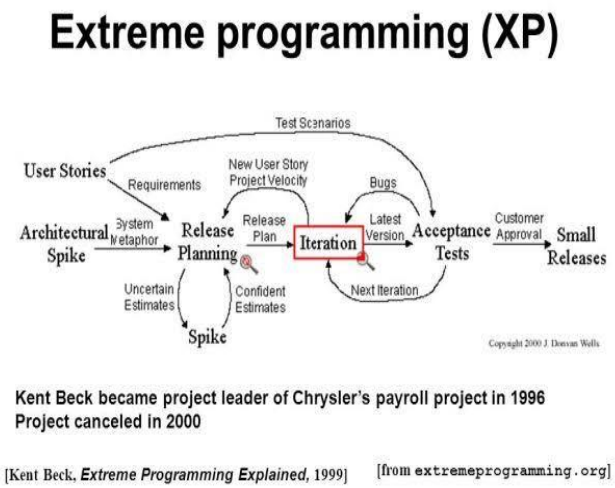
The methodology takes its name from the idea that the beneficial elements of traditional software engineering practices are taken to “extreme” levels. As an example, code reviews are considered a beneficial practice. Taken to the extreme, code can be reviewed continuously through the practice of pair programming.

The original XP method is based on four simple values: simplicity, communication, feedback, and courage.

SUPPORTING PRACTICES:

It also has twelve supporting practices:

- Planning Game
- Small Releases
- Customer Acceptance Tests
- Simple Design
- Pair Programming
- Test-Driven Development
- Refactoring
- Continuous Integration
- Collective Code Ownership
- Coding Standards
- Metaphor
- Sustainable Pace



In XP, the customer works closely with the development team to define and prioritize user stories. The development team estimates, plans, and delivers the highest priority user stories in the

form of working, tested software on an iteration-by-iteration basis. In order to maximize productivity, the practices provide a supportive, lightweight framework to guide a team and ensure high-quality enterprise software.

Crystal:

The Crystal methodology is one of the most lightweight, adaptable approaches to software development.

Crystal is actually comprised of a family of Agile process models, including Crystal Clear, Crystal Yellow, Crystal Orange and others. Each has unique characteristics driven by several factors, such as team size, system criticality, and project priorities. This Crystal family addresses the realization that each project may require a slightly tailored set of policies, practices, and processes in order to meet the product ‘s unique characteristics.

Introduced by Alistair Cockburn, Crystal focuses primarily on people and the interaction among them while they work on an agile software development project. There is also a focus on

business-criticality and business-priority of the system under development.

Unlike traditional development methods, Crystal doesn’t try to fix the tools and techniques of development but keeps people and processes at the core of the process. However, it is not only the people or the processes that are important, rather the interaction between them that is most important.

Several key tenets of Crystal include teamwork, communication, and simplicity, as well as reflection to frequently adjust and improve the process. Like other Agile frameworks, Crystal promotes early, frequent delivery of working software, high user involvement, adaptability, and the removal of bureaucracy or distractions.

Dynamic Systems Development Method (DSDM):

The Dynamic Systems Development Method (DSDM) is an Agile approach that grew out of the need to provide a common industry framework for rapid software delivery. Since 1994, the DSDM methodology has evolved to provide a comprehensive foundation for planning, managing, executing, and scaling Agile process and iterative software development projects.

DSDM is based on eight key principles that direct the team and create a mindset to deliver on time and within budget. These agile principles primarily revolve around business needs/value, active user involvement, empowered teams, frequent delivery, integrated testing, and stakeholder collaboration. DSDM specifically calls out “fitness for business purpose” as the primary criteria for delivery and acceptance of a system, focusing on the useful 80% of the system that can be deployed in 20% of the time.

Compromising any of the following principles undermines the philosophy of DSDM and

introduces risk to the successful outcome of the project.

PRINCIPLES:

DSDM's Eight Key Principles:

- Focus on the business need
- Deliver on time
- Collaborate
- Never compromise quality
- Build incrementally from firm foundations
- Develop iteratively
- Communicate continuously and clearly
- Demonstrate control

Business Requirements are baselined at a high level early on in the project. Rework is built into the process, and all development changes must be reversible. System requirements are planned and delivered in short, fixed-length time-boxes – also known as sprints or iterations – and prioritized using MoSCoW Rules.

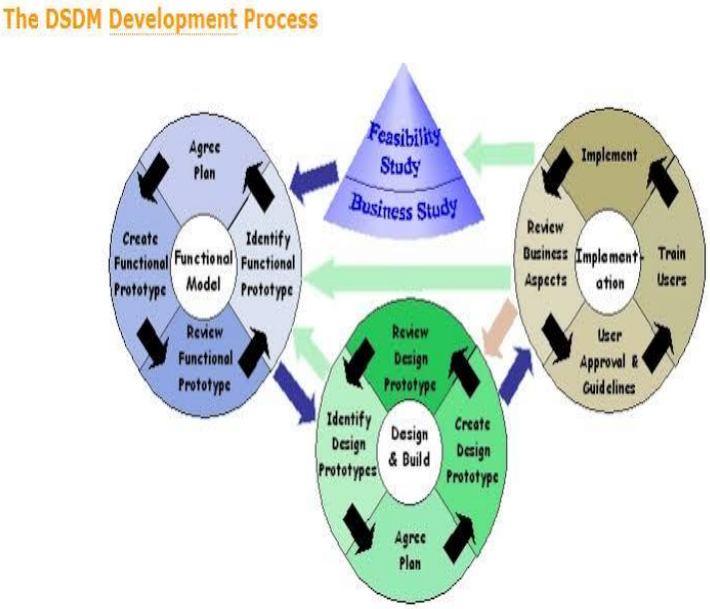
MoSCoW Rules:

M – Must have requirements

S – Should have if at all possible

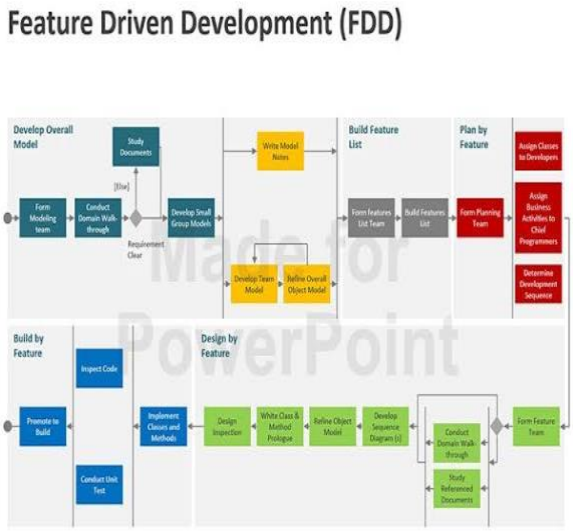
C – Could have but not critical

W – Won‘t have this time, but potentially later



All critical work must be completed in a DSDM project’s defined time-box. It is also important that not every requirement in a project or time-box is considered critical. Within each time-box, less critical items are also included so that they can be removed to keep from impacting higher priority requirements on the schedule.

Feature Driven Development (FDD)



Feature Driven Development is a model-driven, short-iteration process that was built around software engineering best practices such as domain object modeling, developing by feature, and code ownership. The blending of these practices that resulted in a cohesive whole is the best characteristic of FDD.

ACTIVITIES:

Feature Driven Development consists of five basic activities:

- Development of an overall model
- Building a feature list
- Planning by feature
- Designing by feature
- Building by feature

FDD begins by establishing an overall model shape, which will result in a feature list. It then continues with a series of two-week “plan by feature, design by feature, build by feature” iterations. The features are small, “useful in the eyes of the client” results. If they will take more than two weeks to build, then they will have to be broken down into smaller features.

FDD’s main purpose is to deliver tangible, working software in a timely manner, repeatedly. The advantage of using FDD is that it is scalable even to large teams due to the concept of ‘just enough design initially’ (JEDI). Because of its feature-centric process, FDD is a great solution to maintain control for incremental and inherently complex Agile project management.

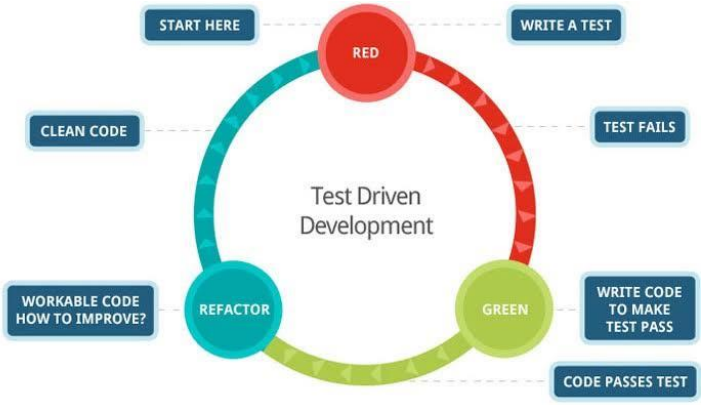
Test-Driven Development:

Test-Driven Development (TDD) is a special case of test-first programming that adds the element of continuous design . With TDD, the system design is not constrained by a paper design document. Instead you allow the process of writing tests and production code to steer the

design as you go. Every few minutes, you refactor to simplify and clarify. If you can easily imagine a clearer, cleaner method, class, or entire object model, you refactor in that direction, protected the entire time by a solid suite of unit tests. The presumption behind TDD is that you cannot really tell what design will serve you best until you have your arms elbow-deep in the code. As you learn about what actually works and what does not, you are in the best possible position to apply those insights, while they are still fresh in your mind. And all of this activity is protected by your suites of automated unit tests.

You might begin with a fair amount of up front design, though it is more typical to start with fairly simple code design; some white-board UML sketches often suffice in the Extreme Programming world. But how much design you start with matters less, with TDD, than how much you allow that design to diverge from its starting point as you go. You might not make sweeping architectural changes, but you might refactor the object model to a large extent, if that seems like

the wisest thing to do. Some shops have more political latitude to implement true TDD than others.



Continuous integration/continuous delivery (CI/CD)

Continuous integration (CI) is a software engineering practice where members of a team integrate their work with increasing frequency. In keeping with CI practice, teams strive to integrate at least daily and even hourly, approaching integration that occurs “continuous-ly.

Continuous delivery (CD) is to packaging and deployment what CI is to build and test. Teams practicing CD can build, configure, and package software and orchestrate its deployment in such a way that it can be released to production in a

software-defined manner (low cost, high automation) at any time.

High-functioning CI/CD practices directly facilitate agile development because software change reaches production more frequently. As a result, customers have more opportunities to experience and provide feedback on change.

Q4 : What is the scope, focus, values and roles in scrum? And it's importance? And its difference from focus, role and values?

Ans : **Scrum Management:**

Scrum is an agile project management methodology or framework used primarily for software development projects with the goal of delivering new software capability every 2-4 weeks.

Project Management In Scrum:

Scrum is a sub-group of agile:

Agile is a set of values and principles that describe a group's day-to-day interactions and activities. Agile itself is not prescriptive or specific.

The Scrum methodology follows the values and principles of agile, but includes further definitions and specifications, especially regarding certain software development practices.

Although developed for agile software development, agile Scrum became the preferred framework for agile project management in general and is sometimes simply referred to as Scrum project management or Scrum development.

Focus and Values of Scrum Methodology:

- Scrum focus on “Done” increment at least once by the end of every sprint.
- The Scrum events and artefacts help create focus on inspecting progress and new information, so we can adapt at frequent enough intervals.
- Each role has a distinct accountability which helps individuals know what to focus on as their

priority. That ultimately contributes to team outcomes.

- The Product Backlog is an ordered list, and that creates focus on what is most important thing to do next.
- Time-boxed events create a sense of urgency and help us focus on the purpose of the event.

Roles in Scrum Methodology:

In contrast to classical project management methods, Scrum doesn't have and doesn't need a product manager, a task manager or a team leader. The most important three roles of Scrum are:

- Product Owner
- Scrum Master
- Development Team
- Stakeholders

Product Owner:

(The person with the product vision)

One of the most important things for the success of scrum is the role of the Product Owner, who

serves as an interface between the team and other involved parties (stakeholders). It can be said that in companies that use scrum, the tasks and responsibilities of the particular Product Owner are never the same. Starting with the choice of that person provided with the proper and necessary skills, make them take specific trainings, up to the responsibility they take

Scrum Master:

(The Scrum expert who helps the team build the product according to the Scrum framework)

The most obvious difference between a team leader and a Scrum Master is represented by the name itself though. Whereas one is leading the team and sets the tasks, the other one is in charge of observing that the team obeys the rules and realizes the method of Scrum entirely. The Scrum Master does not interfere into the decisions of the team regarding specifically the development, but rather is there for the team as an advisor.

The SCRUM Master can ensure that nobody is blocked for more than a day, can see if a task is taking longer than expected, and can take action.

If some feature is going to take longer, a conversation with the Product Owner about whether some features or tasks should be removed from the sprint can be had. Ultimately the team wants to meet the sprint goal

Development Team:

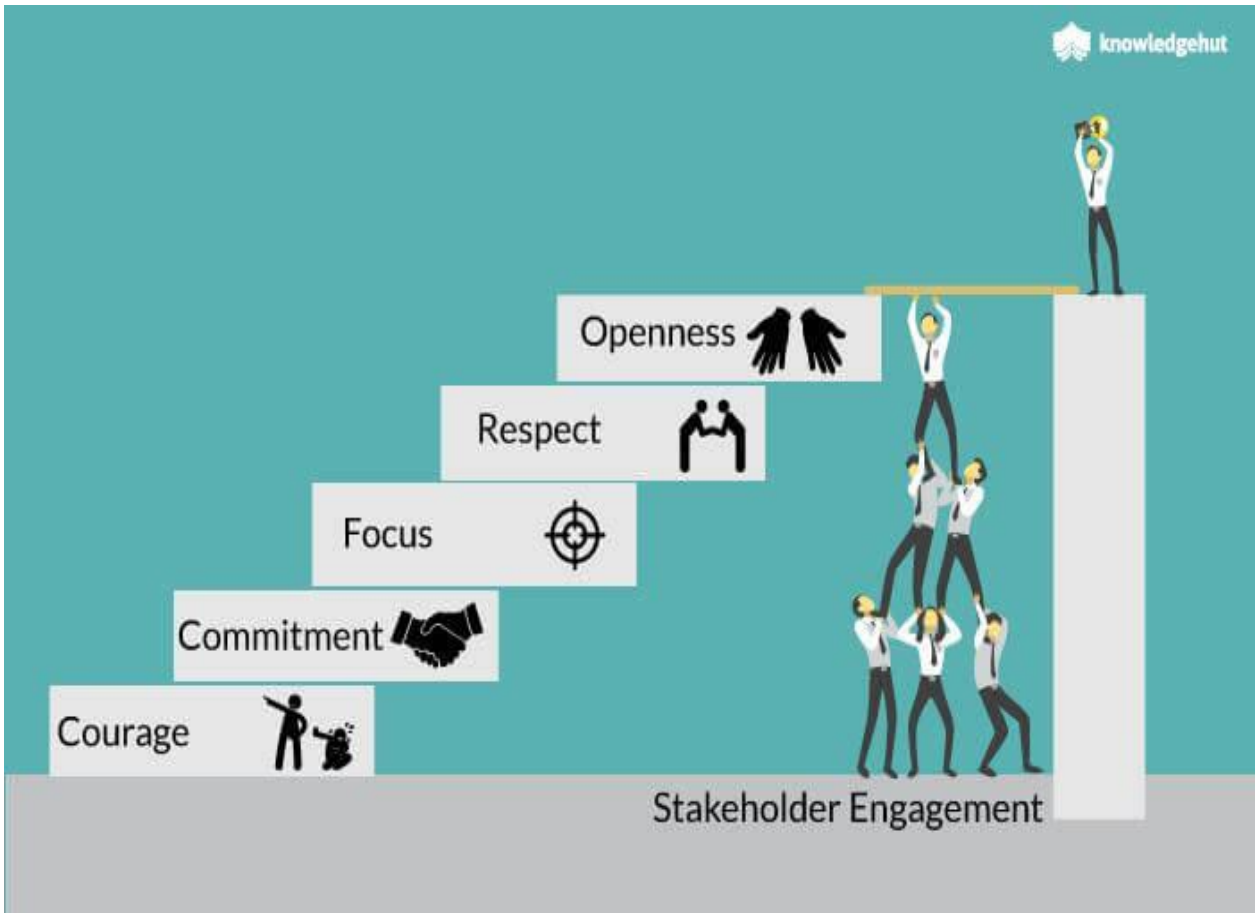
(The team members who execute the work)

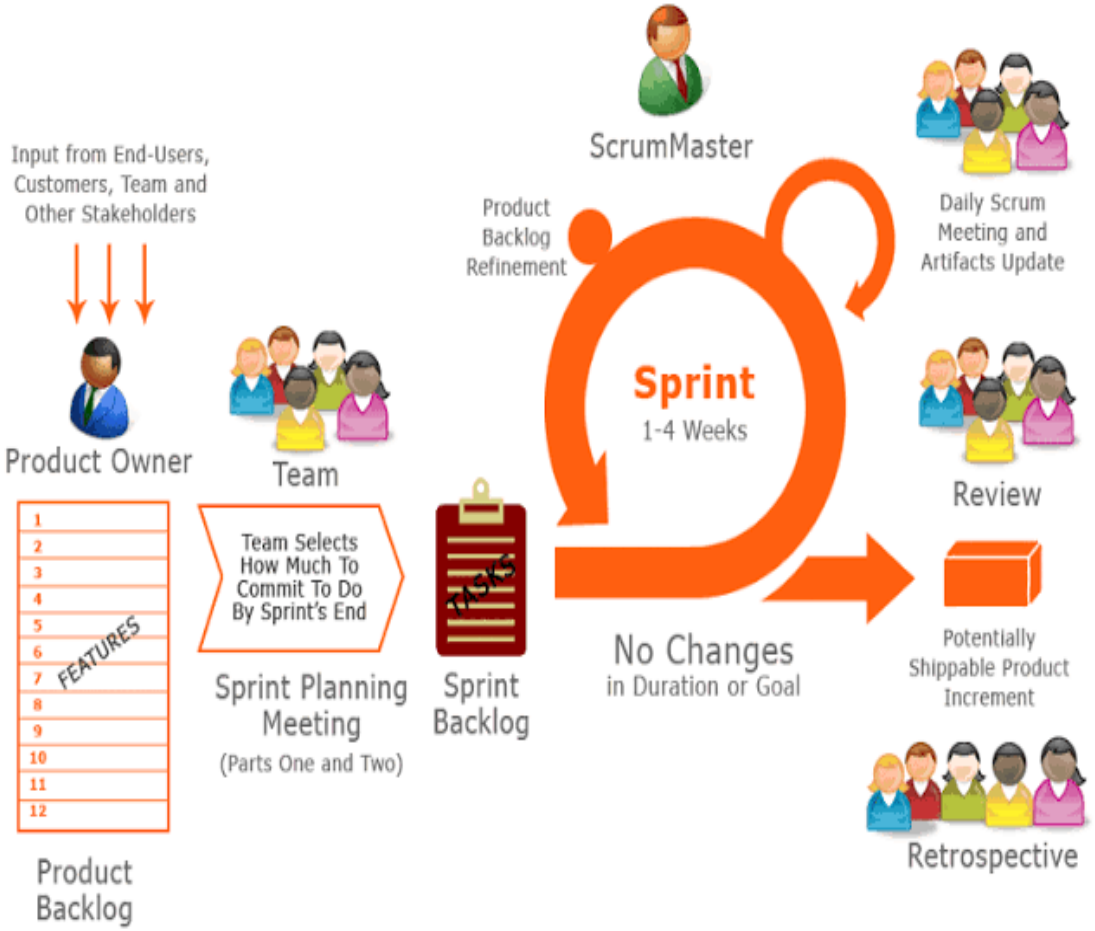
Different from other methods, in Scrum a team is not just the executive organ that receives its tasks from the project leader, it rather decides self dependent, which requirements or User Stories it can accomplish in one sprint. It constructs the tasks and is responsible for the permutation of those – the team becomes a manager. This new self-conception of the team and the therewith aligned tasks and responsibilities necessarily change the role of the team leader/project leader. The Scrum Master does not need to delegate all the work and to plan the project, he rather takes care that the team meets all conditions in order to reach the self-made goals

Stakeholders:.

The people, who have genuine interest in the Product, keep reviewing the team's products and progress and keep providing continual feedback.

These individuals share different tasks and responsibilities related to the delivery of the product. Scrum describes this as a self-organizing and cross-functional team.




Input from End-Users, Customers, Team and Other Stakeholders
Product Owner
Team
1
2
3
4
5
6
7
8
9
10
11
12
FEATURES
Product Backlog
Team Selects How Much To Commit To Do By Sprint's End
Sprint Planning Meeting
(Parts One and Two)
TASKS
Sprint Backlog
ScrumMaster
Product Backlog Refinement
Sprint
1-4 Weeks
No Changes
in Duration or Goal
Daily Scrum Meeting and Artifacts Update
Review
Potentially Shippable Product Increment
Retrospective

The Five Scrum Values are:

- Commitment.
- Focus.
- Openness.
- Respect.
- Courage.

1. Commitment:

Scrum teams must be committed to progress and willing to have practical objectives and stick to them. This is a team activity where you are a part of a team, and you are accountable to work together and to conform to your commitments.

This value can be expressed in three ways-

Sprint-based commitment- Sprint reflects the realistic goals and a short time duration to achieve these goals. Thus, the team has to be committed to their tasks in order to achieve the Sprint goals.

Commitment as a team- As a team, you need to welcome the changes and represent the adaptability. As a team, you can achieve the project goals in smaller chunks. In case of any issue raised, the team can gather and discuss with each other to come up with a concrete solution to tackle the issue.

Commitment as an individual- Being an individual in a team, you should contribute as much as you can to achieve the Sprint goal. This represents

your commitment as an individual. As an individual, you can exhibit a commitment to learning, collaboration, following Scrum values, working software, and so on.

2. Focus:

An iterative-incremental approach and timely delivery in Scrum helps to keep us stay focussed towards the project goal. Once the requirements are clear and the goal is set, the most effective way to attain the goal is to be goal-oriented. This motivates you for delivering faster, better and yield more. By focussing more on a goal, you can avoid resource wastage and deliver on time. This Scrum value leverages lower rate of risks and provides ample time to improve and deliver the needful.

3. Openness:

The Scrum induction requires transparency and openness. We need to investigate reality with a specific end goal to make sensible adjustments. Team members should be open about their work,

progress, what they learned and the issues they are facing. Also, you should be open to working with colleagues, recognizing individuals to be individuals, and not assets, robots or replaceable pieces of hardware.

You should always be open to team up with the stakeholders in terms of orders and abilities. You should be open to work together with team members in any extensive condition. As a team member, you should be open towards sharing a feedback and gaining from each other.

4. Respect:

As a part of the Scrum team, you should respect colleagues, their decisions, and their experience. As an efficient team member, you should also respect diversity. You should respect your stakeholders by not building anything in which people are not interested. You need to equally respect your users by resolving their problems. As a responsible team member, you need to completely adhere to the Scrum framework and the associated Scrum roles. As a part of the Scrum

team, you should respect each other’s skills, knowledge, and insights.

5. Courage:

Adaptability to change forms the bedrock of any Scrum project and to accept a change, courage is needed.

Scrum is all about taking risks and finding out an optimized solution. The Scrum team is allowed to think of different approaches to workshop the best and most appropriate solutions. In order to implement new things to the project, we need to explain these new ideas to the team.

Importance:

By adopting these 5 Scrum values in the project, eventually, you are making your team follow the basic Scrum tenets. Following these values helps in harmonizing the team so they cooperate with each other to create unique ideas for enhanced results.

DIFFERENCE OF SCRUM ROLE FROM OTHER:

The roles in Scrum are quite different from the traditional software methods. Clearly defined roles and expectations help individuals perform their tasks efficiently. In Scrum, there are three roles: Product owner, Development Team, and Scrum master. Together these are known as the scrum team.

DIFFERENCE OF SCRUM VALUES FROM OTHERS:

1. Understanding

Understanding is one of the less obvious values of Kanban. I read it into the first foundational principle, "*Start with what you do now*". Understand the thing you're changing, whether it's the nitty-gritty details of a process, the way a process performs under conditions of stress, or something as abstract as your organisation's overall approach to change. While this is not in scrum.

2. Agreement

Agreement is right there in the second foundational principle, "*Agree to pursue incremental, evolutionary change*". I like to turn this around: would you reasonably expect to be successful in implementing change without it? Could it be that it's lack of agreement that's limiting your progress? While this is not in scrum.

4. Leadership

Leadership features in most stories of success but it was only in 2012 that it was added as a foundational principle, in the

form “*Encourage acts of leadership at all levels in your organization – from individual contributor to senior management*”.

DIFFERENCE OF SCRUM FOCUS FROM OTHERS:

Agile methodologies are a tool, like a JavaScript library or an IDE, and should be used as such. With this thinking we can see that all agile tools have the same tool kit, the '5 whys', Continuous feedback loops, customer focus, Transparency, to name a few. Focusing on the idea of why we do agile, customer focus, better deliverables, reducing waste, faster delivery, we can pick the best tool for the job, and not be limited

Q5. What are different events in scrum?

Events in scrum:

There are Five events in Scrum:

i. Sprint planning.

ii. The sprint

iii. Daily Scrum

iv. Sprint reviews

v. Sprint retrospective

1. **Sprint Planning**

The process of selecting which items to work on during the sprint is done during the sprint planning meeting. Sprint planning is time-boxed to a maximum of 8 hours.

My personal experience is that 2-3 hours is a good length of time for sprint planning.

Sprint planning is focused on answering the following questions:

What can be delivered in the upcoming sprint?

How will we achieve the work needed to deliver these items?

The Product Owner begins the planning session by discussing the objective of the sprint (the sprint goal) and which items, if delivered, would result in the goal being met.

The team discusses how it will deliver these backlog items. This is typically done by breaking up the items into smaller tasks and estimating the length of time needed. By crafting a plan for how it will deliver these items, the team will then

calculate how many items it can confidently deliver.

Only the team can decide how many items are chosen for the sprint (i.e., accepted into the sprint backlog) in this way. No manager or Product Owner can tell the team what it must achieve. The team decides what it can achieve.

The Product Owner will discuss these items with the team during planning, answer questions, and often make trade-offs. For example, perhaps a feature cannot fit into the sprint but part of the feature could be done. The Product Owner would decide whether that is a good approach. At the end of the meeting, the team agrees and commits to achieving the sprint goal through delivering a set of sprint backlog items via a plan.

Note that once a set of items is chosen as the sprint backlog, then the Product Owner should not change priorities or change the sprint goal. Doing so would invalidate the team planning. Although scope may be clarified and re-negotiated as the team learns more during the sprint (e.g., a certain task is more complicated

than expected), the idea is to keep the priority and focus constant.

If there are substantial changes in scope, it is possible to cancel a sprint but this rarely happens in practice.

2. The Sprint:

The sprint is the heart of SCRUM.

A sprint is usually 2 or 3 weeks in duration (although 1-week or 4-week sprints are sometimes seen). The team will attempt to begin, finish, and release work during this time. Once a sprint length is set, it cannot be changed. If your team is doing 2-week sprints, then the sprint must last exactly 2 weeks.

CHARACTERISTICS OF SPRINT:

Some characteristics of sprints:

- Each sprint lasts for the same fixed duration (e.g., 2 weeks).
- The next sprint begins immediately after the previous one.

- The team attempts to begin, work on, complete, and release shippable features.
- Some items are selected from the product backlog that the team will attempt to complete for a sprint. These items become your sprint backlog.

3. Daily Stand-up

The daily stand-up is a daily meeting of no more than 15 minutes where each member of the team gives a brief update (ideally less than one minute) of how their work is progressing.

The daily stand-up is sometimes known as the daily scrum, like a rugby scrum, where many players come together!

It is more common to have this meeting in the morning, although some teams working across multiple time zones may schedule the meeting for later in the day.

It is imperative that the meeting last no more than 15 minutes. Physically standing up helps keep the meeting short (It's easier to take a long time if you're comfortably seated).

The goal of this meeting is to let each team member know what the others are doing and plan to do. Team members can also use this time to indicate if they are stuck or blocked from progressing on their tasks.

The SCRUM Master is fully in control of this meeting. He will ask each of the members:

- What did you do yesterday?
- What you are going to do today?
- Do you have any obstacles or impediments?

A SCRUM team member will often refer to an impediment as a "blocker."

It is important that each member listens to the update of the person speaking - no side conversations are allowed! Meetings should be held in the same place and at the same time. The meeting can be held in front of a progress board if one is used. That way members can update the board and see the status of the whole sprint in one place.

A daily stand-up often happens at a progress/status board!

Benefits of Daily Stand-up Meetings

Although the benefits of keeping each team member up-to-date may seem obvious, it is worth looking at how the daily stand-up can positively impact the team’s performance:

Each team member will know what each other member of the team is doing.

Team members will know if someone is working on something that might be affected by the tasks they are currently working on.

Team members can give input on other team members' tasks, helping them if they are blocked or stuck.

Each team member can surface blockers which can quickly be remedied.

A self-managing culture is born from the team working together, solving problems together, and unblocking each other. You may hear conversations such as “if you pick up that task, then we can get this done” or “I can help with this if you're stuck or if the team needs it."

The SCRUM Master can ensure that nobody is blocked for more than a day, can see if a task is taking longer than expected, and can take action. If some feature is going to take longer, a conversation with the Product Owner about whether some features or tasks should be removed from the sprint can be had. Ultimately the team wants to meet the sprint goal.

Pigs and Chickens

The members of the SCRUM team are often called pigs and non-members of the SCRUM team are called chickens. The explanation is that the chickens lay eggs but the pigs can be eaten as pork. In this sense the chickens are involved but the pigs are committed!

Thus, the non-team members of course care about the sprint but it is the members of the SCRUM team who are completely invested! There is a rule during the daily stand-up that pigs (team members) can talk but chickens (if they are present) cannot.

4. Sprint review: A time-boxed event holds at the end of the Sprint to inspect the Increment and adapt the Product Backlog if needed.

Sprint Demo:

At the end of a sprint, the SCRUM Master will organize a sprint demo where the items of the sprint that have just ended can be shown to key stakeholders. Many organizations choose to have an open invitation for sprint demos. Anyone in the company who is interested can turn up and watch the team demonstrate what they have built.

Advantages:

Doing a demo at the end of the sprint has many great advantages:

- It gives the team confidence to show their work and receive positive feedback from the rest of the organization.
- Key stakeholders know they have a moment when they will see built features before they are released. This gives them an opportunity to give input if they see something seriously

wrong or misunderstood, before customers are affected.

- Feedback can come from many sources if the whole of the organization is invited. Sometimes someone from the support team, marketing team, etc. can deliver an incredibly valuable piece of insight from their domain.
- When the tech team knows they have an upcoming demo with a large audience, they have an added motivation to deliver a great demonstration.

5. Sprint Retrospective

SCRUM teams will hold a retrospective meeting at the end of each sprint so that all team members can give input on what is working well and what is not working well. With well run teams, it shouldn't require more than 30 minutes (with 2 hours being an absolute max)**.**

A time-boxed event for providing an opportunity for the Scrum Team to inspect itself and create a plan for improvements to be enacted during the next Sprint.

Scrum artifacts: Scrum's artifacts represent work or value to provide transparency and opportunities for inspection and adaptation. Artifacts defined by Scrum are specifically designed to maximize transparency of key information so that everybody has the same understanding of the artifact.

Increment: The sum of all the Product Backlog items completed during a Sprint and the value of the increments of all previous Sprints.

Product Backlog: An ordered list of everything that might be needed in the product which is the single source of requirements for any changes to be made to the product. The Product Owner is responsible for the Product Backlog, including its content, availability, and ordering.

Sprint Backlog: The set of Product Backlog items selected for the Sprint, plus a plan for delivering the product increment and realizing the Sprint Goal.

Product Backlog Refinement: Ongoing Product Backlog Refinement is normally taken place within

each Sprint to refine items to be ready for future Sprints.

Q6)COMPARE AS EXCEL SHEET BETWEEN SCRUM AND AGILE?

Waterfall	Agile (Scrum)
Feasibility evaluation takes a long phase and is done in advance to avoid reworking in the next project phases.	Feasibility test takes a shorter while considerably. Clients are engaged in the early project phase to get the buy-in and refine the needs in the long run.
Project planning is done at the beginning of the project and is not open to any changes later on.	The plan is not given the foremost priority and is done during sprint planning. Modifications are welcome except during an active sprint.
Project progress gets monitored according to the project plan.	The development gets tallied in each sprint.
Only the project managers communicate and carry out progress review meetings weekly/ monthly.	Communication is frequent, face-to-face, and clients also participate throughout the project.
Roles are not interchangeable once distributed among project team members.	You can switch roles quickly, and the team can work in cycles.
Documentation gets a lot of emphases and that is pretty comprehensive.	There's a need to file requirements, build designs, and write test plans to promote working software delivery.