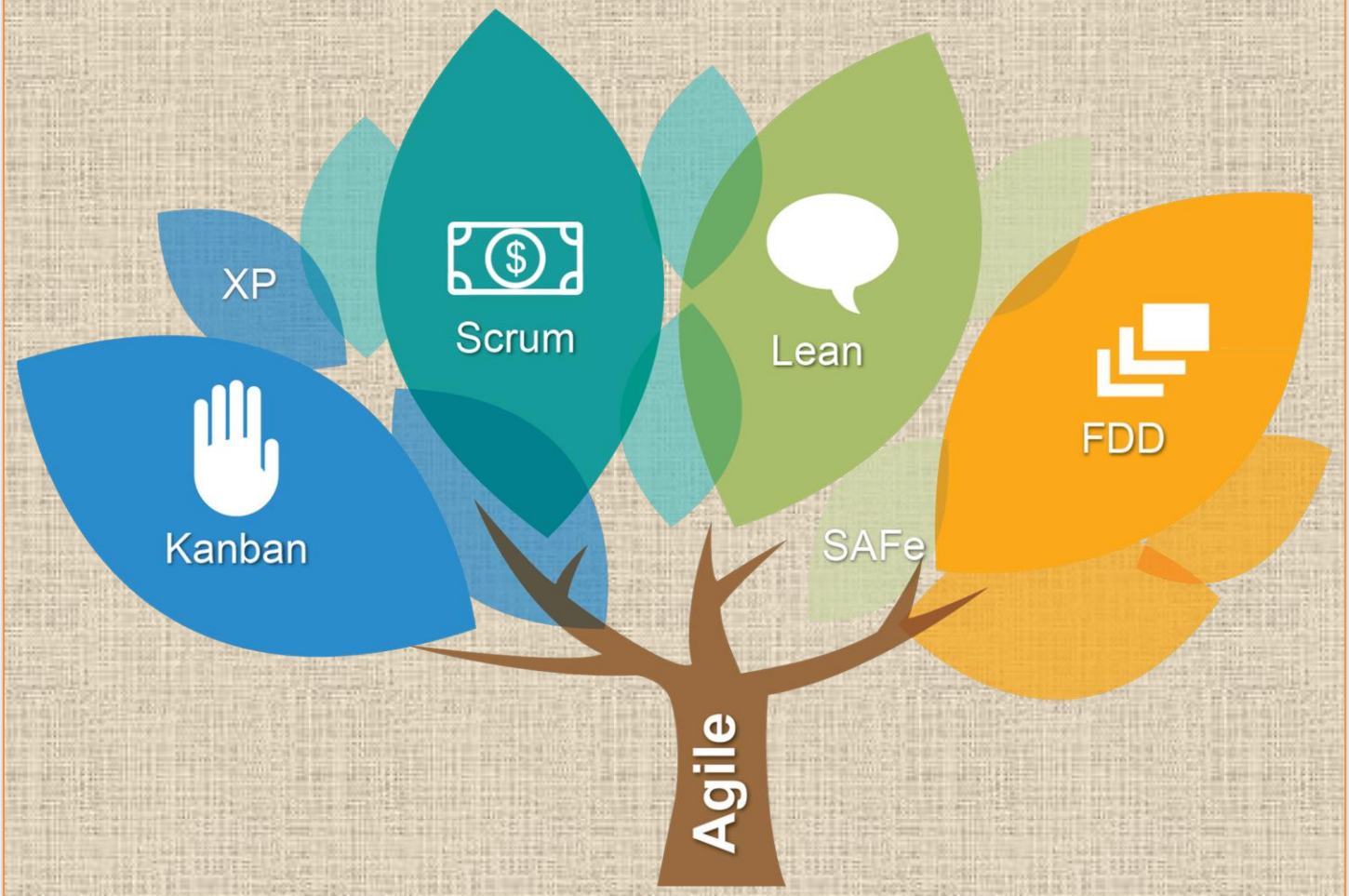
Scrum Master Supreme Action Guide



BY

SRIRAM

Preface

I have been involved in IT Software development since 1997. I have a unique combination of process, technical and industrial skills. As a Certified Scrum Professional (CSP), ICAgile Certified Professional in Agile Coaching (ICP-ACC), SAFe Program Consultant (SPC 4) I have expert level of knowledge in agile and technology practices such as Bigdata-Hadoop, Java, SharePoint & .Net with this combination I can help process and technology people, understand the world. Worked in India, USA, and UK which creates a global experience and awarded as a best agile coach in MNC. Dedicated "Scrum Master Supreme Action Guide" book to my Agile Guru's, my family members, friends and Scrum professionals across the world. Scrum Supreme Action Guide made handy and recollect everything at one shot.

Organization of this Book

Scrum Master Supreme Action Guide is designed to make you to success in the Scrum Master interview by providing valuable discussion on Scrum Master, Agile Coach, Agile Transformations, Scrum Exercises & videos. Practice Certification test to achieve Certified Scrum Master Certification. The progressive elaboration of Scrum Master knowledge to an Agile Coach is awesome. Enjoy Reading!

Table of Contents

Scrum Gurus

Ken Schwaber

Jeff Sutherland

Mike Cohn

Nanda Lankalapalli

Sriram Balasubramanian

Lesson 1 Scrum Master Discussions

Tell me about your introduction before proceed with?

- Worked as an Scrum Coach | Agile coach| SAFe coach in MNC with 16 years of IT experience out of which 8 Years have been dedicated to Agile, Scrum & SAFe practices at enterprise level
- My Agile Certifications are - ICAgile Certified Professional in Agile Coaching (ICP-ACC) & SAFe Program Consultant (SPC 4), Certified Scrum Professional (CSP)
- I am expertise in Scrum Coaching, Agile Coaching, training and mentoring & Lead a proposal with a big win to generate more business value, trained more than 1000+ Professionals on Agile Principles & Practices, XP, Scrum, Kanban, Scaled Agile framework by making the organization as an Agile Organization, build agile culture across 3 continents as an agile coach for distributed team
- Review how teams are conducting Sprint Planning, Scrum Daily calls, Sprint Review and Sprint Retrospective ceremonies and provide feedback., track team velocity for teams after each sprints and plan to Increase it each quarter using Graphs like CFD (Cumulative Frequency Diagram)
- Perform Project Management duties like Client Interaction, Estimations, Resource Handling and delivering on time.
- My agile websites and books are released by Syntel CEO & President -Rakesh Khanna & Temenos CEO - Susan Gibson. I have published my agile books in amazon such as Handy Agile, Agile A Key of Success, Scrum Alliance Professional, Agile Coaching, SAFe Q&A
- Having good experience in technology such as Java, Big data, SharePoint & .Net projects to support the development team

What are the good qualities of Scrum Master?

The good qualities of Scrum Master are: -

- Responsible
- Humble
- Collaborative
- Committed
- Influential
- Knowledgeable

Scrum Coach Sample Resume

Scrum Coach.docx

1.1 Introduction to Scrum

What is Agile Manifesto? (Agile 4 Values)

Meeting at Snowbird resort by 17 software pundits and light weight methodologists, February 2001. Created "Agile Manifesto"

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value. i.e., Agile Values

- **Individuals and interactions over processes and tools**
- **Working software over comprehensive documentation**
- **Customer collaboration over contract negotiation**
- **Responding to change over following a plan**

That is, while there is value in the items on the right, we value the items on the left more.

Author: - Kent Beck, Mike Beedle, Arie Van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Greening, Jim Highsmith, Andrew Hunt, Ron Jerries'', Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas

Agile Manifesto In detail

Individuals and interactions over processes and tools

- **Individuals and interactions are most important**
- **Processes and tools will be needed on projects**
- **Projects are completed by people not processes and tools**
- **Agile projects are people driven**

Working software over comprehensive documentation

- **Agile project need to deliver value**
- **Value is about the purpose or business need the project aims to deliver**
- **Documentation is barely sufficient**
- **Documentation is done just in time –as the last responsible moment**
- **Documentation might also be just because**
    - **Industry requirements | Organizational requirements**

Customer collaboration over contract negotiation

- Agile is flexible, accommodating, and willing to change
- Contracts are often rigid and uncooperative
- Agile contracts must accommodate change
- There's a difference between being right and doing the right thing

Responding to change over following a plan

- Agile welcomes change
- Predictive projects plan everything in advance
- Agile projects have lots and lots of many changes
- Agile projects have uncertainty up front

What are Agile Principles? (Agile 12 Principles)

No	Principle	Shortened Version
1	Our highest priority is to satisfy the customers through early and continuous delivery of valuable software	Customer Satisfaction
2	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage	Welcome Changes
3	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale	Deliver Frequently
4	Business people and developers must work together daily throughout the project	Work with business
5	Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done	Motivated People
6	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	Face to Face Communication

7	Working software is the primary measure of progress	Measure Software Done
8	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely	Maintain Sustainable Pace
9	Continuous attention to technical excellence and good design enhances agility	Maintain Design
10	Simplicity –the art of maximizing the amount of work not done is essential	Keep it Simple
11	The best architectures, requirements, and designs emerge from self-organizing teams	Team creates Architecture
12	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly	Reflect and Adjust

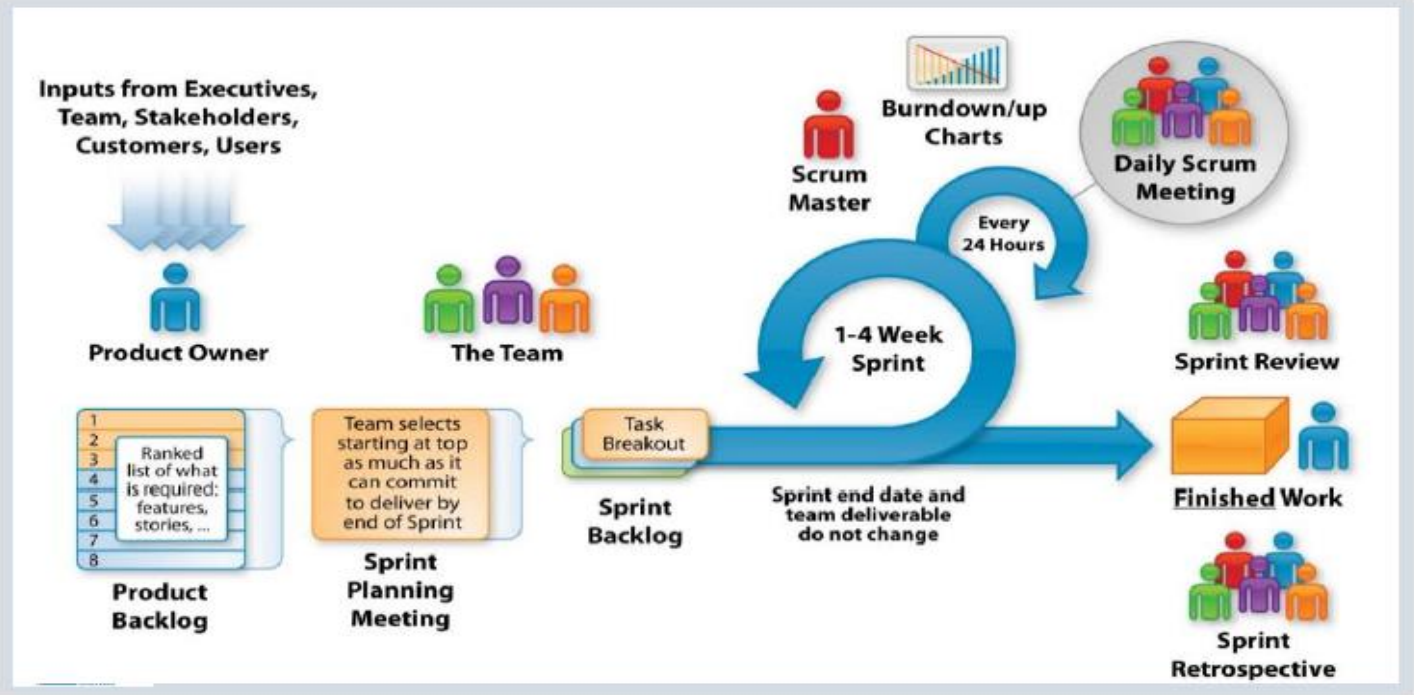
What are methodologies available in Agile?

Methodologies	Information
Scrum	Most popular agile methods Strongly codified set of ceremonies, roles and artifacts
XP	Foremost of agile methodologies & Strong set of technical practices
Lean Kanban	Lean - Set of principles evolved from manufacturing to eliminate waste Kanban literally means a “signboard” or “billboard” and it espouses the use of visual aids to assist and track production. Lean Kanban integrates the use of the visualization methods as prescribed by Kanban along with the principles of Lean creating a visual incremental evolutionary process management system.
DSDM	Offshoot of Rapid Application Development Methodology Cost | Quality/time fixed and requirements prioritized as per MOSCOW

Crystal	Principles are categorized according to criticality and size of the project. Critical Levels: Comfort (C)| Discretionary Money | Essential Money (E) | Life (L)
Feature Driven Development	Plan | Develop and build by feature
Adaptive Software Development (ASD)	ASD are constant adaptation of processes to the work at hand, provision of solutions to problems surfacing in large projects, and iterative, incremental development with continuous prototyping.
Agile Unified Process (AUP)	AUP combines industry-tried-and-tested Agile techniques such as Test-Driven Development (TDD), Agile Modelling, agile change management, and database refactoring, to deliver a working product of the best quality
Domain-Driven Design (DDD)	Domain-driven design is an Agile development approach meant for handling complex designs with implementation linked to an evolving model.
Test Driven Development	Test Driven Development is a software development method that involves writing automated test code first and developing the least amount of code necessary to pass that test later.

Definition of Scrum?

A framework within which people can address complex and adaptive problems while productively and creatively delivering products of the highest possible value.



Scrum uses a methodology called the scrum framework. Scrum is a framework for developing and sustaining complex products. Scrum is a framework with agile principles and values. Scrum was discovered from the Toyota Car Product Implementation by Jeff Sutherland and Ken Schwaber. Scrum was also taken from Rugby game.

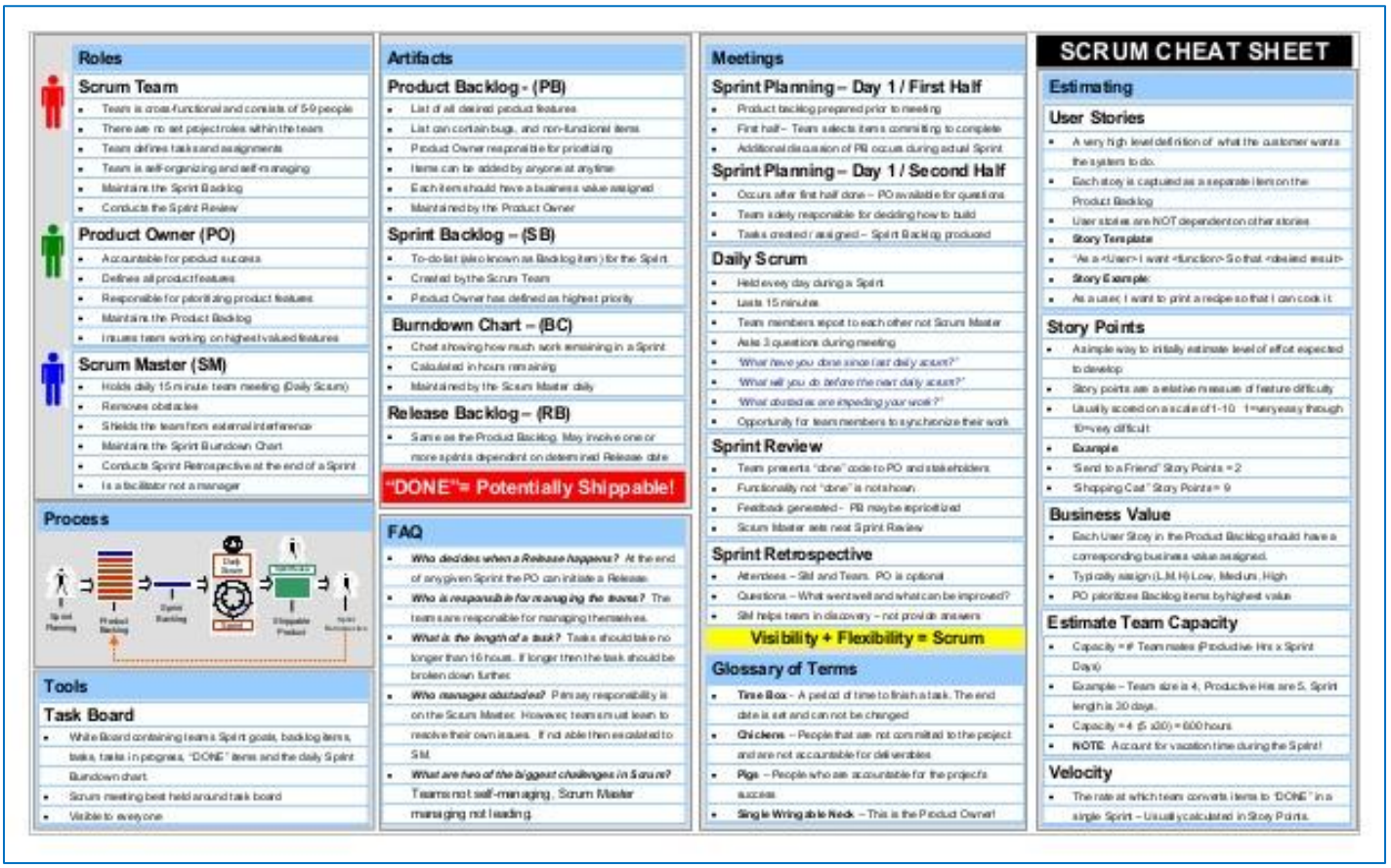
Scrum is light-weight, makes easier to understand but hard to master. Scrum is flexible and allows other agile practices like XP and Lean to plug-in.

The scrum framework is a set of practices, roles and responsibilities, events, artifacts, and rules

Scrum processes enable organizations to adjust smoothly to rapidly-changing requirements, and produce a product that meets evolving business goals. An agile Scrum process benefits the organization by helping it to

- Increase the quality of the deliverable's
- Cope better with change (and expect the changes)
- Provide better estimates while spending less time creating them
- Be more in control of the project schedule and state

Explain Scrum in a Page?



SCRUM CHEAT SHEET

Roles

Scrum Team
- Team is cross-functional and consists of 5-9 people
- There are no set project roles within the team
- Team defines tasks and assignments
- Team is self-organizing and self-managing
- Maintains the Sprint Backlog
- Conducts the Sprint Review

Product Owner (PO)
- Accountable for product success
- Defines all product features
- Responsible for prioritizing product features
- Maintains the Product Backlog
- Ensures team working on highest valued features

Scrum Master (SM)
- Holds daily 15 minute team meeting (Daily Scrum)
- Removes obstacles
- Shields the team from external interference
- Maintains the Sprint Burndown Chart
- Conducts Sprint Retrospective at the end of a Sprint
- Is a facilitator not a manager

Process

Tools

Task Board
- White Board containing team's Sprint goals, backlog items, tasks, tasks in progress, "DONE" items and the daily Sprint Burndown chart
- Scrum meeting best held around task board
- Visible to everyone

Artifacts

Product Backlog - (PB)
- List of all desired product features
- List can contain bugs, and non-functional items
- Product Owner responsible for prioritizing
- Items can be added by anyone at anytime
- Each item should have a business value assigned
- Maintained by the Product Owner

Sprint Backlog – (SB)
- To-do list (also known as Backlog item) for the Sprint
- Created by the Scrum Team
- Product Owner has defined as highest priority

Burndown Chart – (BC)
- Chart showing how much work remaining in a Sprint
- Calculated in hours remaining
- Maintained by the Scum Master daily

Release Backlog – (RB)
- Same as the Product Backlog. May involve one or more sprints dependent on determined Release date

"DONE"= Potentially Shippable!

FAQ
- ***Who decides when a Release happens?*** At the end of any given Sprint the PO can initiate a Release.
- ***Who is responsible for managing the teams?*** The team sare responsible for managing themselves.
- ***What is the length of a task?*** Tasks should take no longer than 16 hours. If longer then the task should be broken down further.
- ***Who manages obstacles?*** Primary responsibility is on the Scum Master. However, teams must learn to resolve their own issues. If not able then escalated to SM.
- ***What are two of the biggest challenges in Scrum?*** Teams not self-managing. Scrum Master managing not leading.

Meetings

Sprint Planning – Day 1 / First Half
- Product backlog prepared prior to meeting
- First half – Team selects items committing to complete
- Additional discussion of PB occurs during actual Sprint

Sprint Planning – Day 1 / Second Half
- Occurs after first half done – PO available for questions
- Team solely responsible for deciding how to build
- Tasks created / assigned – Sprint Backlog produced

Daily Scrum
- Held every day during a Sprint
- Lasts 15 minutes
- Team members report to each other not Scrum Master
- Asks 3 questions during meeting
- *'What have you done since last daily scrum?'*
- *'What will you do before the next daily scrum?'*
- *'What obstacles are impeding your work?'*
- Opportunity for team members to synchronize their work

Sprint Review
- Team presents "done" code to PO and stakeholders
- Functionality not "done" is not shown
- Feedback generated – PB may be reprioritized
- Scum Master sets next Sprint Review

Sprint Retrospective
- Attendees – SM and Team. PO is optional
- Questions – What went well and what can be improved?
- SM helps team in discovery – not provide answers

Visibility + Flexibility = Scrum

Glossary of Terms
- **Time Box** – A period of time to finish a task. The end date is set and can not be changed
- **Chickens** – People that are not committed to the project and are not accountable for deliverables
- **Pigs** – People who are accountable for the project's success
- **Single Wringable Neck** – This is the Product Owner!

Estimating

User Stories
- A very high level definition of what the customer wants the system to do.
- Each story is captured as a separate item on the Product Backlog
- User stories are NOT dependent on other stories
- **Story Template**
- "As a <User> I want <function> So that <desired result>
- **Story Example:**
- As a user, I want to print a recipe so that I can cook it

Story Points
- A simple way to initially estimate level of effort expected to develop
- Story points are a relative measure of feature difficulty
- Usually scored on a scale of 1-10. 1=very easy through 10=very difficult
- **Example**
- "Send to a Friend" Story Points = 2
- "Shopping Cart" Story Points = 9

Business Value
- Each User Story in the Product Backlog should have a corresponding business value assigned.
- Typically assign (L,M,H) Low, Medium, High
- PO prioritizes Backlog items by highest value

Estimate Team Capacity
- Capacity = # Team mates (Productive Hrs x Sprint Days)
- Example – Team size is 4, Productive Hrs are 5, Sprint length is 30 days.
- Capacity = 4 (5 x30) = 600 hours
- **NOTE**: Account for vacation time during the Sprint!

Velocity
- The rate at which team converts items to "DONE" in a single Sprint – Usually calculated in Story Points.

Even though you have lot of frameworks in agile why Use Scrum framework?

Some of the key benefits of using SCRUM in any project are: -

1. Adaptability—Empirical process control and iterative delivery make projects adaptable and open to incorporating change.

2. Transparency—All information radiators like a Scrum board and Sprint Burn down Chart are shared, leading to an open work environment.

3. Continuous Feedback—Continuous feedback is provided through the Conduct Daily Stand-up, and Demonstrate and Validate Sprint processes.

4. Continuous Improvement—The deliverables are improved progressively Sprint by Sprint, through the Groom Prioritized Product Backlog process.

5. Continuous Delivery of Value—Iterative processes enable the continuous delivery of value through the Ship Deliverable's process as frequently as the customer requires.

6. Sustainable Pace—Scrum processes are designed such that the people involved can work at a sustainable pace that they can, in theory, continue indefinitely.

7. Early Delivery of High Value—The Create Prioritized Product Backlog process ensures that the highest value requirements of the customer are satisfied first.

8. Efficient Development Process—Time-boxing and minimizing non-essential work leads to higher efficiency levels.

9. Motivation—The Conduct Daily Stand-up and Retrospect Sprint processes lead to greater levels of motivation among employees.

10. Faster Problem Resolution—Collaboration and colocation of cross-functional teams lead to faster problem solving.

11. Effective Deliverable's—The Create Prioritized Product Backlog process and regular reviews after creating deliverable ensures effective deliverables to the customer.

12. Customer Centric—Emphasis on business value and having a collaborative approach to stakeholders ensures a customer-oriented framework

13. High Trust Environment—Conduct Daily Stand-up and Retrospect Sprint processes promote transparency and collaboration, leading to a high trust work environment ensuring low friction among employees.

14. Collective Ownership—The Approve, Estimate, and Commit User Stories process allows team members to take ownership of the project and their work leading to better quality.

15. High Velocity—A collaborative framework enables highly skilled cross-functional teams to achieve their full potential and high velocity.

16. Innovative Environment—The Retrospect Sprint and Retrospect Project processes create an environment of introspection, learning, and adaptability leading to an innovative and creative work environment.

What are the characteristics of Scrum?

The characteristics of Scrum are: -

- The most popular 'Agile Processes' in Agile software development is Scrum
- A project management/execution process framework
- Well suited for projects that require Empirical process control
- Focuses on self-organizing teams
- Requirements are captured in a prioritized list (Product Backlog)
- Product progresses in a series of month-long "sprints"
- No specific engineering practices prescribed
- Uses generative rules to create an agile environment for delivering projects

What is Scrum theory?

Scrum is founded on empirical process control theory, or empiricism. Empiricism asserts that knowledge comes from experience and making decisions based on what is known. Scrum employs an iterative, incremental approach to optimize predictability and control risk. Three pillars uphold every implementation of empirical process control: transparency, inspection, and adaptation.

What is Defined & Empirical Process?

Defined Process

A Defined process defines all steps in advance, same output is expected every time the process is followed, best suits for "Simple" and "Complicated" problem domains.

- Follows pre-defined steps to achieve an Output.
- Suitable when the output is well defined.
- Same output is expected every time the process is followed.
- Best suits for problems those fall into "Simple" and "Complicated" problem domains.

Empirical Process

An Empirical process are interactive, incremental, change often, adapt, and pass through the reviews, Empirical processes are change-driven

As software products and requirements cannot be 100% confirmed, fixed at the beginning, the best way to build the winning product is to continuously inspect, and adapt at regular intervals, effectively and efficiently. Empirical Process is based on such inspect and adapt cycle.

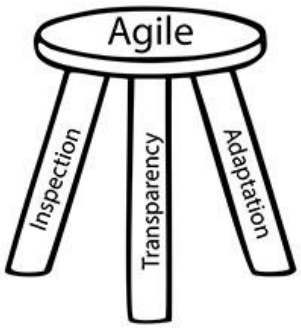
An Empirical Process is: -

- **Built based on the series of experiments**
- **Experience based decision making**
- **Suitable when the output can't be well defined**
- **Definition of output is refined based on the result of experiments**
- **Steps in the process are adjusted based on the feedback from the experiments**
- **Deming Wheel – Plan – DO Inspect -Adapt**

In order to build the winning products and deliver value SCRUM has various feedback loops so that product and process are inspected, adapted and transparent.

Three legs | pillars of SCRUM – Inspection, Adaptation & Transparency: -

- **Inspection – Frequent Inspection of artefacts helps stakeholders to make any changes to achieve the goal**
- **Adaptation – Continuous improvement by adjusting the process based on the inspection**
- **Transparency – All artefacts of the process are visible to all the stakeholders. This helps stakeholders to inspect the current stake and take any required action**



How would you say Scrum is based on empiricism?

Scrum is based in empiricism because: -

- All Artifacts should be transparent to all stakeholders
- All SCRUM roles are empowered to do the job right
- All SCRUM meetings allow collaboration and opportunities for inspection ad adaptation
- In SCRUM, the process is constantly adjusted if needed based on the short and continuous feedback loops at iteration levels.

Why Agile methodology is empirical in nature?

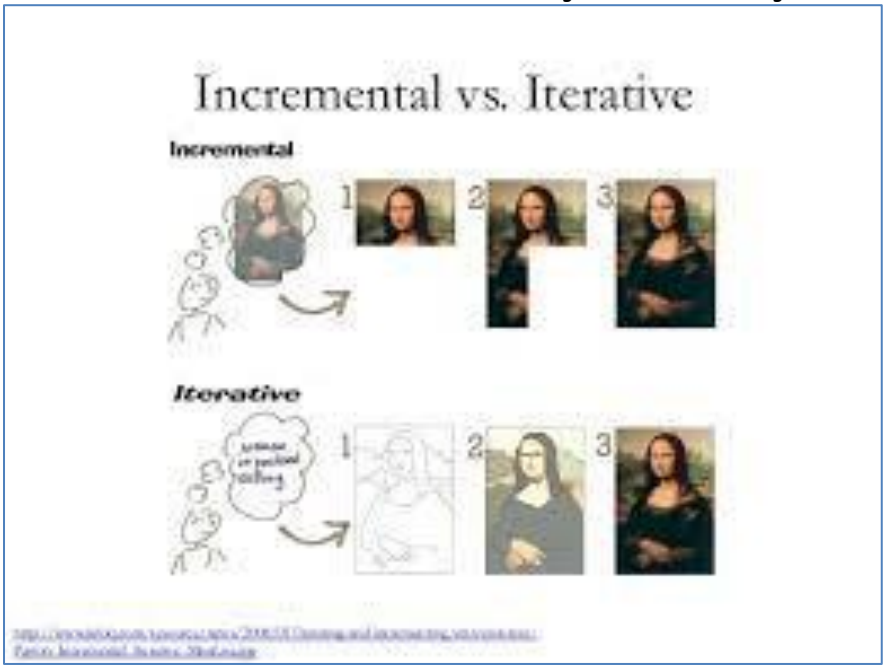
- Course correction at frequent intervals
- Regular customer feedback
- Failure detected early and hence early adoption of corrective measures
- Status is visible to all stakeholders in a consistent way

Tell me Scrum in 100 words?

- Scrum is an agile process that allows us to focus on delivering the highest business value in the shortest time.
- It allows us too rapidly and repeatedly and repeatedly inspect actual working software in every two weeks.
- The business set the priorities. Team self-organize to determine the best way to deliver the highest priority features.
- Every two weeks to a month anyone can see the real working software and decide to release it as is or continue to enhance it for another sprint.

How would you say Scrum is incremental and iterative?

Scrum team delivers value incrementally and Iteratively



Incremental Development

Incremental development is to build small increment of a full fledges product. Each increment adds more software value – like Adding package to a Software Product. After lot of increments, you have got a big Software Product.

Benefits

- **Reduce risk during development**
- **Early discovery and mitigation of risks**
- **Accommodates changes early**
- **Manageable Complexity**
- **Higher confidence and satisfaction from early repeated successful delivery**
- **Early and continuously visibility of product increment**
- **Better predictability and progress**
- **Higher quality and lower defects**
- **Final product close to customer's desire**
- **Early and regular process improvement**
- **Continuous collaboration ad engagement with customers**
- **Effective and efficient**
- **Usable product at any time**
- **Sustainable pace of development**

Iterative Development

Iterative development is to build something, to get some feedback, then refine it to make better, keep doing that until the product is good enough.

Benefits

- **Focus on high value and good Return on Investment (ROI)**
- **Reduce rarely used features, maximize frequently used features**
- **Usable product at any time**
- **Quality Focus**
- **Effective and efficient**
- **Usable product at any time**
- **Sustainable pace of development**

What are the SCRUM values?

All work performed in SCRUM needs a set of values as the foundation for the team's processes and interactions. And by embracing these five values, the team makes them more instrumental to its health and success.

Focus

Because we focus on only a few things at a time, we work well together and produce excellent work. We deliver valuable items sooner.

Courage

Because we work as a team, we feel supported and have more resources at our disposal. This gives us the courage to undertake greater challenges.

Openness

As we work together, we express how we're doing, what's in our way, and our concerns so they can be addressed.

Commitment

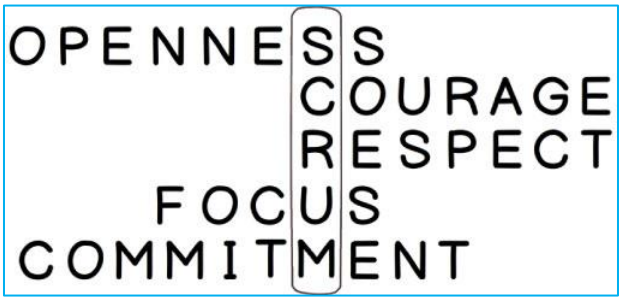
Because we have great control over our own destiny, we are more committed to success.

Respect

As we work together, sharing successes and failures, we come to respect each other and to help each other become worthy of respect.

As an organization applies Scrum it discovers its benefits. At the same time, it sees how these values inherently contribute to the success of Scrum and understands why they are both needed, and bolstered, by Scrum.

Coin a SCRUM Word from the SCRUM values?

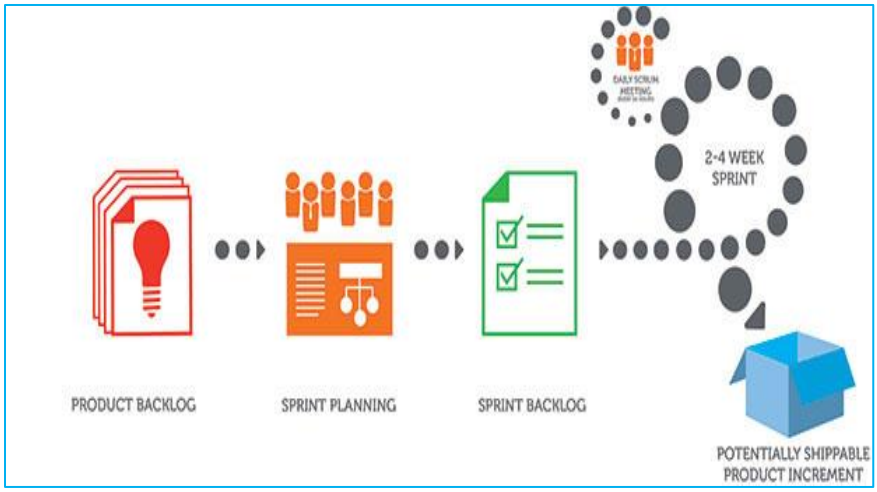


Tell me SCRUM Framework in short?

Already we have seen the Agile has 4 Values & 12 Principles

SCRUM is a simple process framework. SCRUM has

3 Legs	:	**Inspect, Adapt, Transparent**
3 Roles	:	**Product Owner | Scrum Master | Development Team**
3 Artifacts	:	**Product Backlog | Sprint Backlog | Product Increment**
4 Meetings	:	**Sprint Planning | Daily SCRUM | Sprint Review | Sprint Retrospective**
1 Activity	:	**Product Backlog Refinement**
5 Values	:	**Focus | Courage | Openness | Commitment | Respect**



Product Backlog - Ordered list of items to be worked on for the product

Sprint Backlog - Product backlog items selected to work in the Sprint and the work plan to complete those items

Product Increment - Completed product backlog items in a sprint, which are ready to be delivered to the customer

Product Backlog Refinement - A meeting to get the product backlog items ready for the next few sprints

Sprint Planning - A meeting to create the sprint goal and plan the work for the sprint

Daily Scrum - A daily 15-minute time boxed event for the Development Team to synchronize activities and create a plan for the next 24 hours

Sprint Review - A meeting to inspect the product increment and adapt the product backlog if needed

Sprint Retrospective - A meeting for the scrum team to inspect and adapt the process, people and tools

What are the Principles of Scrum?

SCRUM principles are the core guidelines for applying the Scrum framework and should mandatory be used in all Scrum projects. The six Scrum principles are: -

- Empirical Process Control
- Self-organization
- Collaboration
- Value-based Prioritization
- Time-boxing
- Iterative Development

----------Discussions----------

Compare Waterfall Vs Agile? How Scrum is different from waterfall?

In Waterfall

- Plan driven process, predictive, fixed scope, adjust schedule to preserve scope
- Long development cycle, linear, organize work into major phases, delivers value at project completion

In Agile

- Agile value driven Process, Adaptive, Fixed Schedule, Adjustable scope to preserve schedule
- Short development cycle 2-4 weeks, cyclic, organizes work into small deliverables, delivers values incrementally over time

The major differences are: -

- The feedback from customer is received at an early stage in Scrum than waterfall, whereas the feedback from customer is received towards the end of development cycle.
- To accommodate the new or changed requirement in scrum is easier than waterfall.
- Scrum focuses on collaborative development than waterfall where the entire development cycle is divided into phases.
- At any point of time we can roll back the changes in scrum than in the waterfall.
- Test is considered as phase in waterfall unlike scrum.

When should you use Waterfall over Scrum?

Use waterfall if the requirements are simple, predictable, fully defined and understood, and will not change.

How is Scrum different from Iterative model?

Scrum is a type of iterative model only but it is iterative + incremental model.

Do you know any other agile methodology apart from Scrum?

Other Agile methodology include – Kanban, XP, Lean

Explain Agile in 30 seconds?

Agile is a framework of approaches and behaviors that encourage "just-in-time" production that enables customers to receive quality software sooner.

Tell me one big advantage of using scrum?

The major advantage which I feel is – Early feedback and producing the Minimal Viable Product (MVP) to the stakeholders.

What is the advantage of doing Scrum?

The advantage of doing scrum is that while performing the test

- **It minimizes the risk in response to changes made to the system**
- **It increases ROI (Return of Investment)**
- **It improves the process continuously**
- **It repeatedly and rapidly looks into actual working software**
- **Anyone can see real working software product and continue their enhancement for another iteration**

List out the dis-advantages of Scrum?

- **It makes all dysfunction visible & It requires significant change**
- **It will be a tricky job for a scrum master to plan, organize and structure a project that lacks a clear goal**
- **Daily scrum meeting requires frequent reviews and substantial resources**
- **A successful project relies on the maturity and dedication of all the team members**
- **Uncertainty regarding the product, frequent changes and frequent product delivery remains during the scrum cycle**

Where you found the disadvantage of using scrum?

The problems mainly arise when the scrum team do not either understand the values and principles of scrum or are not flexible enough to change.

Do you think scrum can be implemented in all the software development process?

Scrum is used mainly for

- **complex kind of project**
- **Projects which have early and strict deadlines.**
- **When we are developing any software from scratch.**

Where does automation fit into scrum?

Automation plays a vital role in Scrum. In order to have continuous feedback and ensure a quality deliverable we should try to implement TDD, BDD and ATDD approach during our development. Automation in scrum is not only related to testing but it is for all aspect of software development.

As I said before introducing TDD, BDD and ATDD will speed up our development process along with maintaining the quality standards; automating the build and deployment process will also speed up the feature availability in different environment – QA to production. As far as testing is concerned, regression testing should be the one that will have most attention. With progress of every sprint, the regression suit keeps on increasing and it becomes practically very challenging to execute the regression suit manually for every sprint. Because we have the sprint duration of 2 – 4 weeks, automating it would be imperial.

Can you give an example of where scrum cannot be implemented? In that case what do you suggest?

Scrum can be implemented in all kinds of projects. It is not only applicable to software but is also implemented successfully in mechanical and engineering projects.

What are the most important components of Agile?

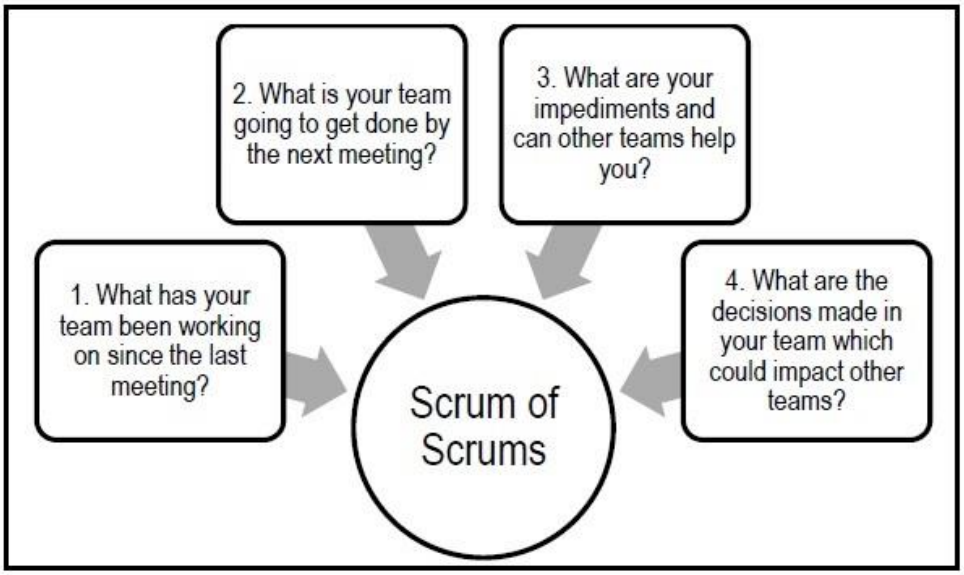
The key feature of agile are:

- **Daily stand-up meetings.**
- **CRC (Class Responsibilities and Collaborators) cards**
- **timeboxed task boards.**
- **TDD (Test Driven Development), Continuous Integration, regular code reviews, pair programming, automated builds, continuous deployment and delivery, etc.**
- **You have iteration planning meetings and carry out iterative development.**

Explain what is Scrum of Scrum? How will you work with Large Scrum Team?

Scrum of scrum is used to refer the meeting after the daily scrum. The responsible person from each team attends the meeting and discuss their work and answer the questions like

- **Since the last meeting, what is the progress of the team?**
- **What your team is expected to do or should accomplish, before the next meeting?**
- **What are the obstacles your team faced while completing the task?**
- **Were you going to allot any of your work to the following team?**
- **Four questions are answered:**
  - **What has your team done since we last met?**
  - **What will your team do before our next meeting?**
  - **Are there any roadblocks in your team's way?**
  - **Will your team put anything in another team's way?**



1.2 The Scrum Team

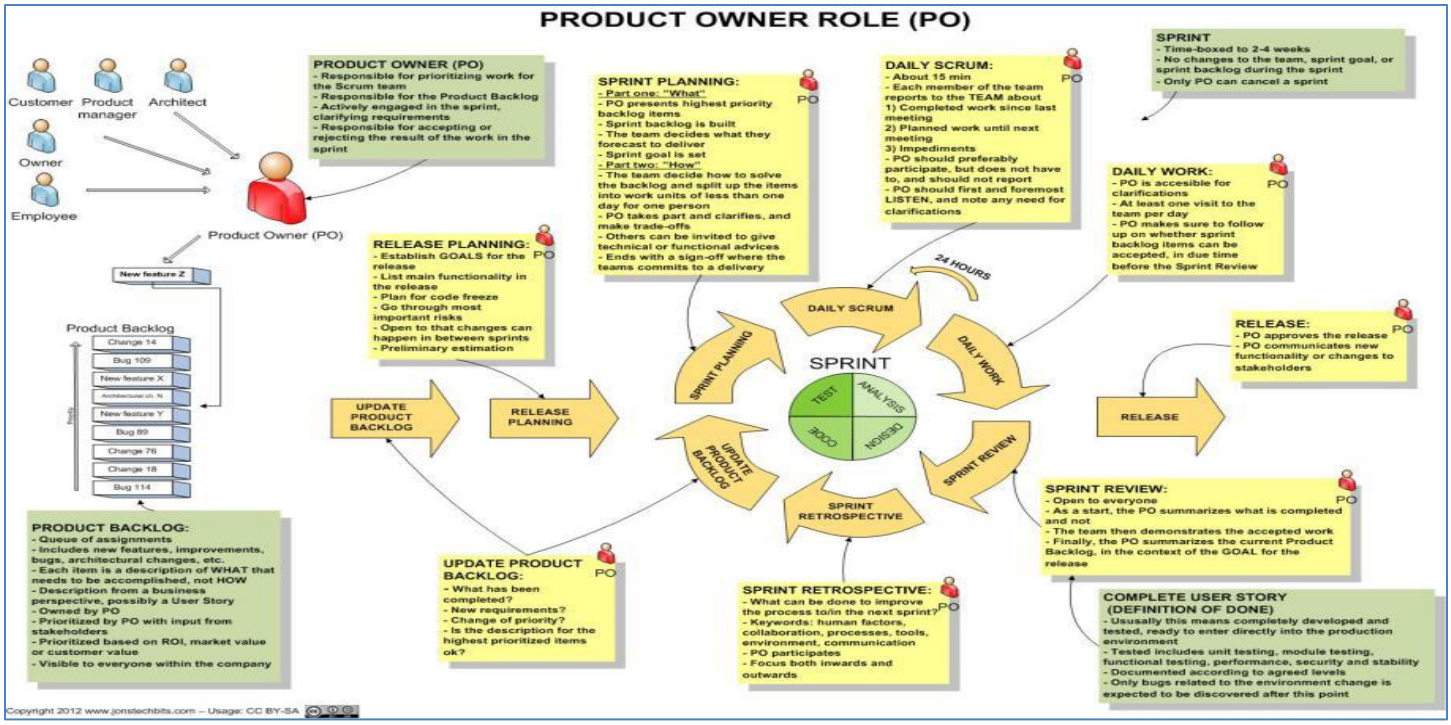
Who are all involved in the Scrum Team?

The Scrum Team consists of a Product Owner, the Development Team, and a Scrum Master. Scrum Teams are self-organizing and cross-functional. Self-organizing teams choose how best to accomplish their work, rather than being directed by others outside the team. Cross-functional teams have all competencies needed to accomplish the work without depending on others not part of the team. The team model in Scrum is designed to optimize flexibility, creativity, and productivity.

Scrum Teams deliver products iteratively and incrementally, maximizing opportunities for feedback. Incremental deliveries of "Done" product ensure a potentially useful version of working product is always available.

What are the responsibilities of The Product Owner?

The Product Owner is responsible for maximizing the value of the product and the work of the Development Team. How this is done may vary widely across organizations, Scrum Teams, and individuals.



The Product Owner is the sole person responsible for managing the Product Backlog. Product Backlog management includes: -

- Clearly expressing Product Backlog items;
- Ordering the items in the Product Backlog to best achieve goals and missions;
- Optimizing the value of the work the Development Team performs;
- Ensuring that the Product Backlog is visible, transparent, and clear to all, and shows what the Scrum Team will work on next; and,
- Ensuring the Development Team understands items in the Product Backlog to the level needed.

The Product Owner may do the above work, or have the Development Team do it. However, the Product Owner remains accountable. The Product Owner is one person, not a committee. The Product Owner may represent the desires of a committee in the Product Backlog, but those wanting to change a Product Backlog item's priority must address the Product Owner.

For the Product Owner to succeed, the entire organization must respect his or her decisions. The Product Owner's decisions are visible in the content and ordering of the Product Backlog. No one is allowed to tell the Development Team to work from a different set of requirements, and the Development Team isn't allowed to act on what anyone else says.

Product Owner's responsibility to Stakeholders

Product owner collaborates with stakeholders of the product to understand their vision and market conditions and regularly update them on the progress of the product development. Product Owner

- Creates the Product vision if it does not exist
- Creates and manages Product Backlog to fulfil the Product vision
- Reviews the product with stakeholders in Sprint Review meetings to solicit feedback
- Updates stakeholders about progress of the product development typically every sprint
- Represents single voice of customers
- Prioritizes product backlog items based on business needs with the consensus of all stakeholders
- Orders the product backlog items to maximize the delivery value

Product Owner responsibility to the team

- Establishes the shared vision between stakeholders and the team
- Details out the product backlog items as appropriate. Make sure that high value items are ready for implementation in upcoming sprints
- Helps team estimate by clarifying any questions
- Available to the team to answer any questions regarding the product domain during the sprint
- Optimizes the value of the work the development team performs
- keeps the product backlog transparent, clear and visible to all
- Attend Scrum Meetings
    - Sprint Planning: Helps the team plan to sprint
    - Sprint Review: Reviews the product increment with stakeholders
    - Sprint Retrospective: Contributes to come up with improvement ideas

Other responsibilities of a Product Owner

- Responsible for deciding whether to release the product increment at the end of the sprint
- Creates and manages the release plans
- Tracks the release progress

Product Owner: Summary

- Defines the feature of the product
- Decide on release date and content
- Be responsible for the profitability of the product (ROI)
- Prioritize features according to the market value
- Adjust features and priority every iteration, as needed
- Accept or reject work results. Accepted Sprints only considered for velocity.
- Maintains and grooms the Product Backlog
- An effective product owner is Committed, Responsible, Authorized, Collaborative, and Knowledgeable (CRACK)
- The Product Owner to be effective, will be available to the team 24/7, is accountable for the product, empowered to make decisions and has knowledge in the product domain preferably

What are the responsibilities of The Development team?

The Development Team consists of professionals who do the work of delivering a potentially releasable Increment of "Done" product at the end of each Sprint. Only members of the Development Team create the Increment.

Development Teams are structured and empowered by the organization to organize and manage their own work. The resulting synergy optimizes the Development Team's overall efficiency and effectiveness.

Development Teams have the following characteristics: -

- They are self-organizing. No one (not even the Scrum Master) tells the Development Team how to turn Product Backlog into Increments of potentially releasable functionality;
- Development Teams are cross-functional, with all of the skills as a team necessary to create a product Increment;
- Scrum recognizes no titles for Development Team members other than Developer, regardless of the work being performed by the person; there are no exceptions to this rule;
- Scrum recognizes no sub-teams in the Development Team, regardless of particular domains that need to be addressed like testing or business analysis; there are no exceptions to this rule; and,
- Team has common goal. Individual Development Team members may have specialized skills and areas of focus, but accountability belongs to the Development Team as a whole.

Development Team Size: Optimum Team Size 4 -9, Members excluding Scrum Master & PO

Team's responsibilities include but not limited to:

- Quality of the product increment
- Create the product increment
- Participate in all scrum meetings
- Implement good engineering practices like Continuous Integration, Continuous Delivery, Automation, Collective Ownership, Clean code, Simple design & effectively run the Test-Driven Development & Behavior Driven Development
- Create and manage the product backlog
- Identify and eliminate the "Technical Debt"
- Track progress of the sprint
- Helps the product owner in backlog management by explaining technical constraints
- Estimates Product Backlog items

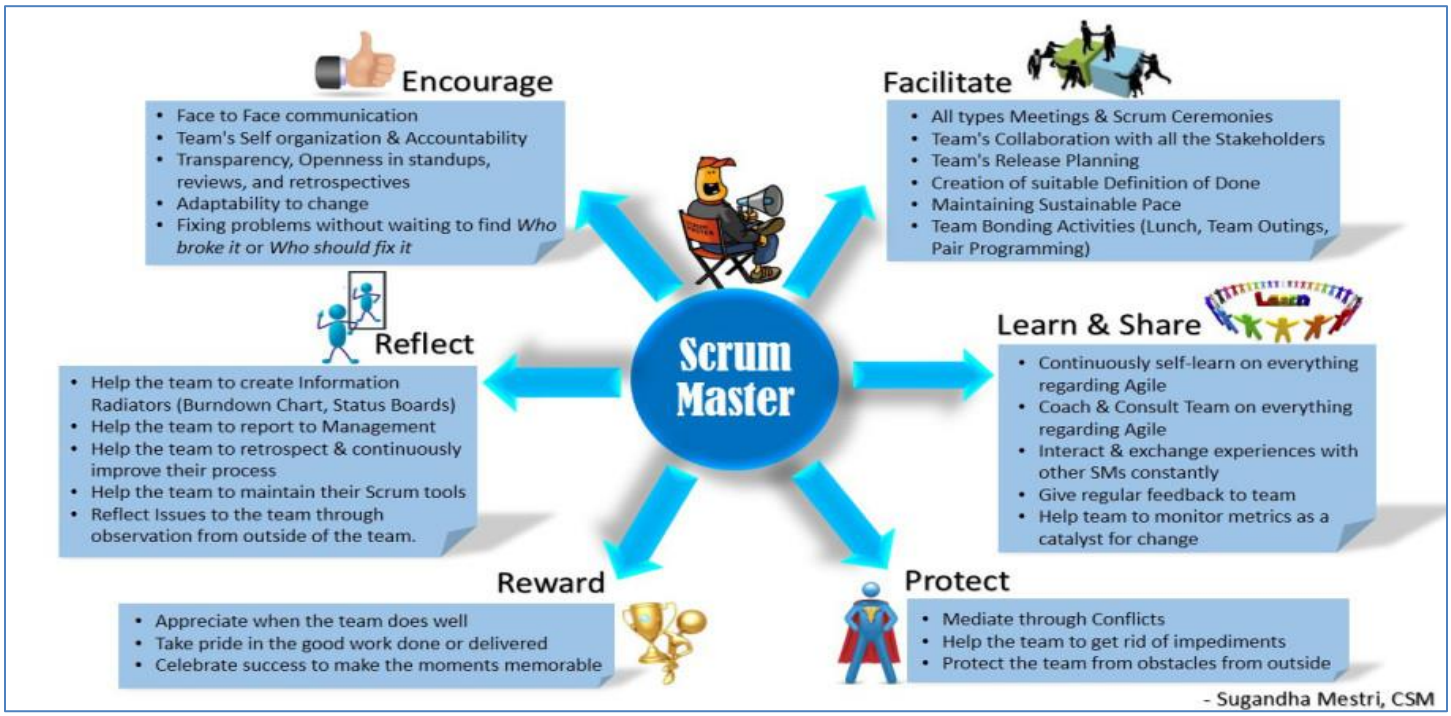
Development Team: Summary

- Typically, 4 – 9 people. Ideally 7+/- 2 (Note: Scrum Master & Product Owner Excluded)
- **Cross functional** – Programmers, testers, user experience designers, etc.,
- Members should be full-time – May be exceptions for DBA's
- Teams are **self-organizing** (No titles)
- Membership should change only between sprints

What are the responsibilities of Scrum Master? How Scrum Master services to PO, DT & Organization?

The Scrum Master is the process champion and responsible for ensuring Scrum is understood and enacted. Scrum Masters do this by ensuring that the Scrum Team adheres to Scrum theory, practices, and rules.

The Scrum Master is a servant-leader for the Scrum Team. The Scrum Master helps those outside the Scrum Team understand which of their interactions with the Scrum Team are helpful and which aren't. The Scrum Master helps everyone change these interactions to maximize the value created by the Scrum Team.



General Responsibilities of Scrum Master

- **Teaches Scrum to the team and the rest of the organization**
- **Ensures that scrum is understood and enacted**
- **Facilitate all scrum meetings as needed and requested. Makes sure that meetings are happening the scrum team. Coaches the team on how to complete the tasks on time within the timebox for that meeting**
- **Responsible for building the product fast by eliminating the waste**
- **A Servant – Leader for the Scrum Team**
- **Act as a change agent that increases the productivity of the Scrum team**

Scrum Master Service to the Product Owner

The Scrum Master serves the Product Owner in several ways, including: -

- **Helps Product Owner to prioritize the work. Teaches PO and Stakeholders value based prioritization**
- **Finding techniques for effective Product Backlog management;**
- **Helping the Scrum Team understand the need for clear and concise Product Backlog items;**
- **Understanding product planning in an empirical environment;**
- **Ensuring the Product Owner knows how to arrange the Product Backlog to maximize value;**
- **Understanding and practicing agility; and,**
- **Facilitating Scrum events as requested or needed.**

Scrum Master Service to the Development Team

The Scrum Master serves the Development Team in several ways, including: -

- **Identifies the impediments those are blocking the team to progress and help resolve them as quickly as possible. Scrum Master goal is to make sure the team is highly productive**
- **Coaching the Development Team in self-organization and cross-functionality;**
- **Helping the Development Team to create high-value products;**
- **Removing impediments to the Development Team's progress;**
- **Facilitating Scrum events as requested or needed; and,**
- **Coaching the Development Team in organizational environments in which Scrum is not yet fully adopted and understood.**

Scrum Master Service to the Organization

The Scrum Master serves the organization in several ways, including: -

- **Assess the readiness of the organization to implement the team**
- **Leading and coaching the organization in its Scrum adoption;**
- **Planning Scrum implementations within the organization;**
- **Helping employees and stakeholders understand and enact Scrum and empirical product development;**
- **Causing change that increases the productivity of the Scrum Team; and,**
- **Working with other Scrum Masters to increase the effectiveness of the application of Scrum in the organization.**

Scrum Master: Summary

- Represents management to the project
- Responsible for enacting Scrum rules, values and practices and censure team members and stakeholders not adhering to these rules ad norms
- Removes impediments
- Ensure that the team is fully functional and productive
- Enable close co-operation across all roles and functions
- Shield the team from external interferences
- Practices "Servant Leadership" – Facilitator and enabler rather than a Manager

What are the qualities of an effective Scrum Master?

The Scrum Master to be effective, there various important skills to acquire. Here some of those skills are: - Facilitation | Coaching | Servant Leadership

I Facilitation

Facilitation is the process of designing and conducting a successful meeting or event that has a particular objective. Facilitation serves the needs of any group, who are meeting with a common purpose. The person who facilitates the meeting is called a "Facilitator"

Characteristics of Facilitator

- Facilitator does not stand in front of the group and lecture
- Facilitator is an active unbiased member of the learning process
- Facilitator has to skilfully assist the group to understand their common objectives, and to help them to achieve these objectives without taking sides in any arguments
- Facilitator guides and helps to achieve consensus

Basic Skills of a Facilitator

- Follows a good meeting practice
- Time keeping
- Run the meeting on agenda
- Assisting the group to brainstorm and problem solve
- Ability to intervene in a way that adds creativity to a discussion rather than leading the discussion
- Ability to understand the group dynamics include:
    - Who is dominating the group? How to stop him/her
    - Who is withdrawn? How to involve him/her
    - Who looks bored? How to get their attention

Good Facilitation techniques should

- Help the participants to be comfortable with each other
- Create a fun and interesting learning development
- Boost the energy levels of workshop participants
- Organize interesting and productive group work activities
- Use participatory activities which enable dynamic reviews of what has been learnt?
- Increase group activity so that workshop participants can expand on the new

II Coaching

According to Tim Gallwey **"Coaching is unlocking a person's potential to maximize their own performance. It's helping them to learn rather than teaching them"**

Coaching is a useful way of developing people's skills and abilities, and of boosting performance. It can also help deal with issues and challenges before they become major problems.

A coaching session will typically take place as a conversation between the coach and coachee (person being coached), and it focusses on helping the coachee discover answers for themselves

Occasionally, coaching may mean an informal relationship between two people, of whom one has more experience and expertise than the other and offers advice and guidance as latter learns

Qualities of a Coach

- Coaches help to find solutions
- Coaching helps the organizations to change for the better
- Coaching is long term and sustainable results
- Coaching give progressive elaboration
- Coaching helps you to review and refine your own solution to make it work

III Servant Leadership

You must be the change you wish to see in the world. You are the change agent for your organization.

According to Robert K Greenleaf. The Servant-leader is servant first… it begins with the natural feeling that one wants to serve, to serve first. The conscious choice brings one to aspire to lead. That person is sharply different from one who is leader first, perhaps because of the need to assuage an unusual power drive or to acquire material possessions. The leader-first and servant-first are two extreme types. Between them there are shadings and blends that are part of the infinite variety of human nature.

On character and Servant Leadership

- Listening
- Empathy
- Healing
- Awareness
- Persuasion
- Conceptualization
- Foresight
- Stewardship
- Commitment to the growth of people
- Building community

----------Discussions----------

Do you know the Three Amigos in Scrum?

The three Amigos are – The product Owner, The Scrum Master and the Development Team.

What do you think should be the ideal size of a Scrum team?

The ideal size is 7 to9 with +/- 2

How long does a scrum cycle last? Who are involved in Scrum cycle?

Scrum cycle depends on the type of project the team is working on, usually, it ranges about 2-4 weeks to about a month. In scrum cycle, it includes a

- Scrum master
- Product owner
- Team

What are the roles in Scrum?

Scrum prescribes only three roles: The Product Owner, Scrum Master, and the Delivery Team.

These roles should ideally be cross-functional and not shared among other projects. Many Scrum Masters have not had the opportunity to work with a team that was cross-functional or dedicated due to the organization's resistance or inability to allow for what some refer to as a "luxury." This question may lead the interviewer to ask how you would handle working with a team that did not have a designer or tester within the team or how you would handle a team that was not dedicated. Other roles such as PM, Architect, BA, UI Designer, DBA, Sysadmin.

What are the roles of a Scrum Master and Product owner?

Scrum Master – Acts as a servant Leader for the scrum team. He presides over all the scrum ceremonies and coaches the team to understand and implement scrum values and principals.
Product Owner – Is the Point of contact for a scrum team. He/she is the one who work closest to the business. The main responsibility of a product owner is to identify and refine the product backlog items.

Explain what is "Sashimi" and "Impediments"?

- **Sashimi:** This term is analogous to "done", it is used to define the specific task when it is completed. The term used by different team to refer their completed task status may differ, but should remain same within one team.
- **Impediments:** Any obstacle that prevent the team members from performing their work is referred as impediments

Mention in brief, what is the role of scrum master in Scrum?

- Removes any obstacles that the team faces during the pursuit of its sprint goals
- Maximizing the productivity of the team
- Making sure that the scripting language used for system testing and unit testing is written in the same language
- Guides the team and product owner to improve the effectiveness of their practices
- Makes sure that all standard scrum practices are followed

So, in scrum which entity is responsible for deliverable? Scrum master or Product owner?

Neither the scrum master, nor the product owner. It's the responsibility of the team who owns the deliverable.

Describe a time when your Delivery team members didn't seem to be getting along. How did you handle this?

A little bit of conflict is always good, but your interviewer is looking for your ability to be an effective leader. Reflect on a time where you had a few team members that just never seems to able to work things out. How did you encourage those team members to work together? Was it a team building exercise? Did you make sure that they had a common goal? State the problem you had, how you addressed it, and the outcome.

How many Scrum teams have you managed at one time?

This is a popular question. Don't offer that Scrum guidelines state only one Scrum Master per team as your answer! In this new role, you may be required to lead more than one team. Notice the use of the word "managed" versus "led." Scrum Masters do not manage, they lead teams—so be sure to use this word in your response. Your interviewer is likely to be listening very closely! Ideally Scrum Master can manage two teams with maximum of 10 members in each team, because he has to handle the daily status and reports. Agile Coach can manage 2-4 teams.

Is it okay if someone wants to change a requirement?

Yes. Agile encourages frequent feedback from customers and stakeholders so that the product can be improved. We need to be able to embrace change. Approach Product Owner for suggesting some changes. Based on the Product needs it will be considered.

How much time should a person expect to spend on Scrum Master activities?

A Scrum Master should make this role their top priority to focus on benefits of the overall team. Their load will vary from sprint to sprint depending on what impediments and issues the team is dealing with. Newly formed teams typically take more Scrum Master time; 50%-100%, while experienced Scrum Masters with established well-functioning teams might spend 50% or less time on the Scrum Master role

What qualities should a good Agile tester have?

1. **Agile tester should be able to understand the requirements quickly.**
2. **They should know Agile concepts and principals.**
3. **As requirements keep changing, testers should understand the risk involved in it.**
4. **Agile tester should be able to prioritize the work based on the requirements.**
5. **Communication is must for an agile tester as it requires constant communication with developers and business associates.**

How QA can add value to an agile team?

- **QA can provide value addition by thinking differently about the various scenarios to test a story. They can provide quick feedback to the developers whether new functionality is working fine or not.**
- **QA is not a separate silo but is part of a cross-functional project team. It is included in the project from the beginning, and the whole team works together on user stories using the**

same tracking tools. The Director of the QA team works closely with the executive management team to identify technology and staffing needs in relation to project pipelines.

- Quality Assurance is empowered to support projects and add value in whatever way the situation requires. Examples include: design reviews, requirements assessments, browser and device support, process, tools, risk assessments, and helping to determine "Definition of Ready" and "Definition of Done."
- QA sits with the project team whenever possible, allowing for increased conversation and problem solving in real time. The QA team attends and contributes to all relevant planning meetings and sprint ceremonies and also work directly with clients on quality and testing processes.
- Members of QA teams always learn as individuals, as project team members, and as representatives of a skilled discipline within the organization. Our process and approach to testing evolves to keep up with advances in technology and the changing needs of clients. What works for one client or project might differ radically from another. Flexibility is the key.

Explain when Scrum cannot be useful?

Ideally scrum is useful to monitor work with 5 to 10 people, who are committed to achieving the sprint goal. It does not go well with huge groups or team having more responsibilities. For larger team, scrum can be applied by splitting the team into small groups and practice scrum.

Would you recommend automated testing for your project?

Scrum encourages the use of automated performance or regression testing so that you can continuously deliver software as quickly as possible. Offer examples of any automated testing tools that your team may have used.

How does agile testing (development) methodology differ from other testing (development) methodologies?

The testers (developers) ensure that the whole process of testing (development) is broken into small steps as possible, and just a small unit of code is tested (developed) in each of these steps. The team of testers (developers) consistently communicates the results of their work, and changes the short-term strategy and even the development plan on the go, based on the results of agile testing. Agile methodology encourages flexible and rapid response to change, which should lead to better end results.

Tell me the Scrum Ownership responsibilities?

Item	Development Team	Product Owner	Scrum Master
Estimates	✓ DT		
Backlog Priorities		✓ PO	
Agile Coaching			✓ SM
Velocity Predictions	✓ DT		
Definition of Done | Sprint Planning	✓ DT	✓ PO	✓ SM
Process Adherence			✓ SM
Technical Decision	✓ DT		

1.3 Product Vision

What is Product vision?

Product vision statement is few sentences that talks about the motivation behind the product. Since the product owner is responsible for success of the product, he/she leads the creation of Product Vision.

For Creating the Product Vision

- ✓ **Describe the Motivation behind the Product**
- ✓ **Look beyond the Product**
- ✓ **Distinguish between Vision and Product Strategy**
- ✓ **Employ a Shared Vision**
- ✓ **Choose an Inspiring Vision**
- ✓ **Think Big**
- ✓ **Keep your Vision Short and Sweet**
- ✓ **Use the Vision to Guide your Decisions**

----------Discussions----------

Who is responsible for creating the Product Vision?

Product owner is responsible for creating the Product Vision.

How will you distinguish between Product Vision & Product Strategy?

The strategy is a plan, the tactics are how the plan will be executed and the Vision is the end-result.

To overcome the product vision if fails,

1. **Set realistic goals**
2. **Stop overanalysing**
3. **Accept blame**
4. **Embrace learnings**

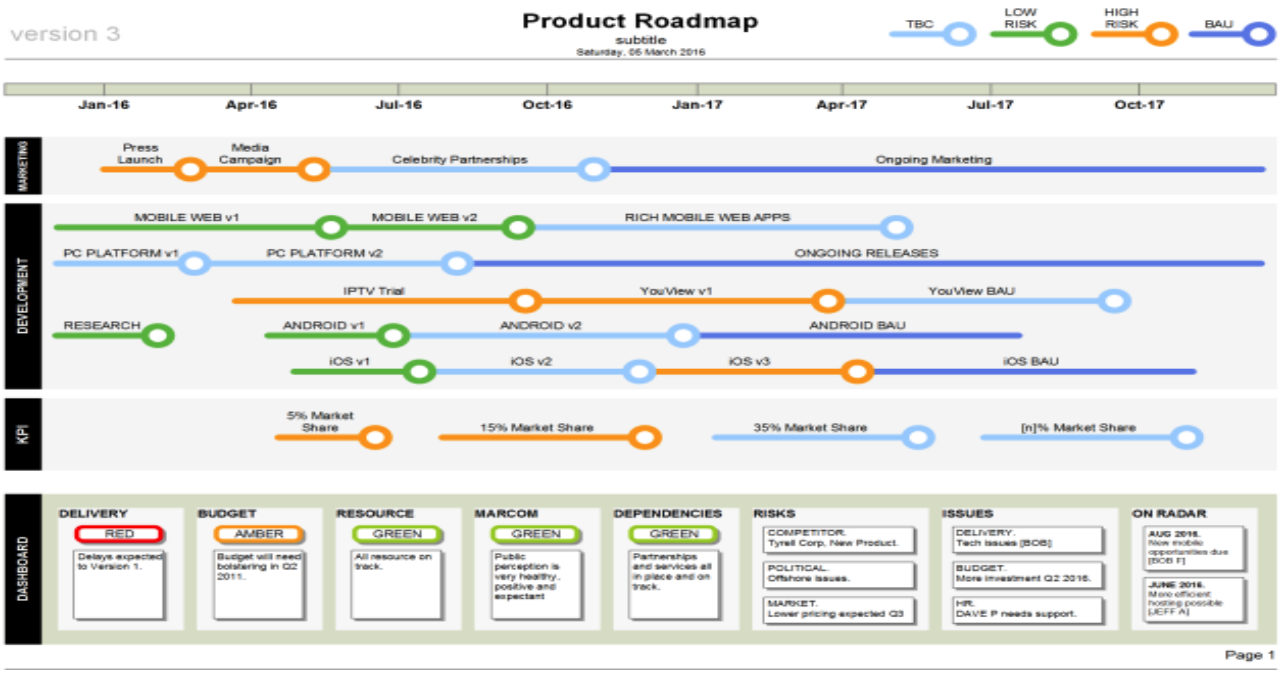
How to create a Product Vision with help of customer support?

If you are working on something exciting that you really care about, you don't have to be pushed. The vision pulls you. – Steve Jobs

Our vision is to be earth's most customer centric company; to build a place where people can come to find and discover anything they might want to buy online – Amazon.com

What is Product Roadmap?

- **The visualization of product features**
- **The product roadmap equates to the product division as a whole**
- **This is done and owned by the product owner**



1.4 Scrum Artifacts

What is 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.

List out Scrum Artifacts

- **Product Backlog**
- **Sprint Backlog**

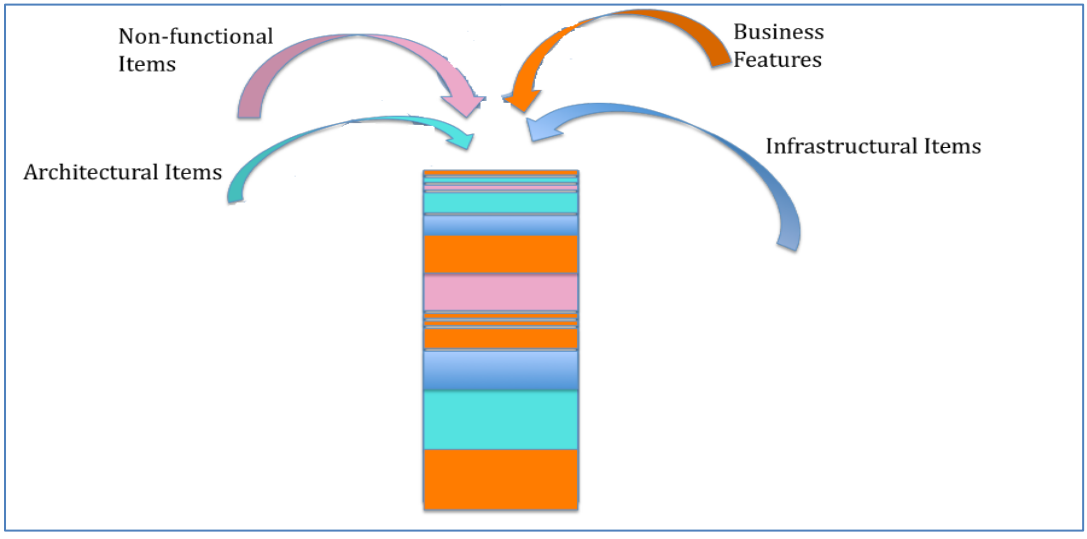
What is Product Backlog?

The Product Backlog is an ordered list of everything that might be needed in the product and 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. Items in the product backlog are called "Product Backlog Items".

Each product backlog item would have:

- **Description – Details of the item**
- **Value – What business value this item would provide**
- **Estimate – Effort estimate to build this item**
- **Order – The order in which the items should be worked in**

Product Backlog contains all items required to accomplish the product vision.



A Product Backlog is never complete. The earliest development of it only lays out the initially known and best-understood requirements. The Product Backlog evolves as the product and the environment in which it will be used evolves. The Product Backlog is dynamic; it constantly changes to identify what the product needs to be appropriate, competitive, and useful. As long as a product exists, its Product Backlog also exists.

The Product Backlog lists all features, functions, requirements, enhancements, and fixes that constitute the changes to be made to the product in future releases. Product Backlog items have the attributes of a description, order, estimate and value.

As a product is used and gains value, and the marketplace provides feedback, the Product Backlog becomes a larger and more exhaustive list. Requirements never stop changing, so a Product Backlog is a living artifact. Changes in business requirements, market conditions, or technology may cause changes in the Product Backlog.

Multiple Scrum Teams often work together on the same product. One Product Backlog is used to describe the upcoming work on the product. A Product Backlog attribute that groups items may then be employed.

Product Backlog refinement is the act of adding detail, estimates, and order to items in the Product Backlog. This is an ongoing process in which the Product Owner and the Development Team collaborate on the details of Product Backlog items. During Product Backlog refinement, items are reviewed and revised. The Scrum Team decides how and when refinement is done. Refinement usually consumes no more than 10% of the capacity of the Development Team. However, Product Backlog items can be updated at any time by the Product Owner or at the Product Owner's discretion.

Higher ordered Product Backlog items are usually clearer and more detailed than lower ordered ones. More precise estimates are made based on the greater clarity and increased detail; the lower the order, the less detail. Product Backlog items that will occupy the Development Team for the upcoming Sprint are refined so that any one item can reasonably be "Done" within the Sprint time-box. Product Backlog items that can be "Done" by the Development Team within one Sprint are deemed "Ready" for selection in a Sprint Planning. Product Backlog items usually acquire this degree of transparency through the above described refining activities.

The Development Team is responsible for all estimates. The Product Owner may influence the Development Team by helping it understand and select trade-offs, but the people who will perform the work make the final estimate.

How would we say the Product Backlog is DEEP?

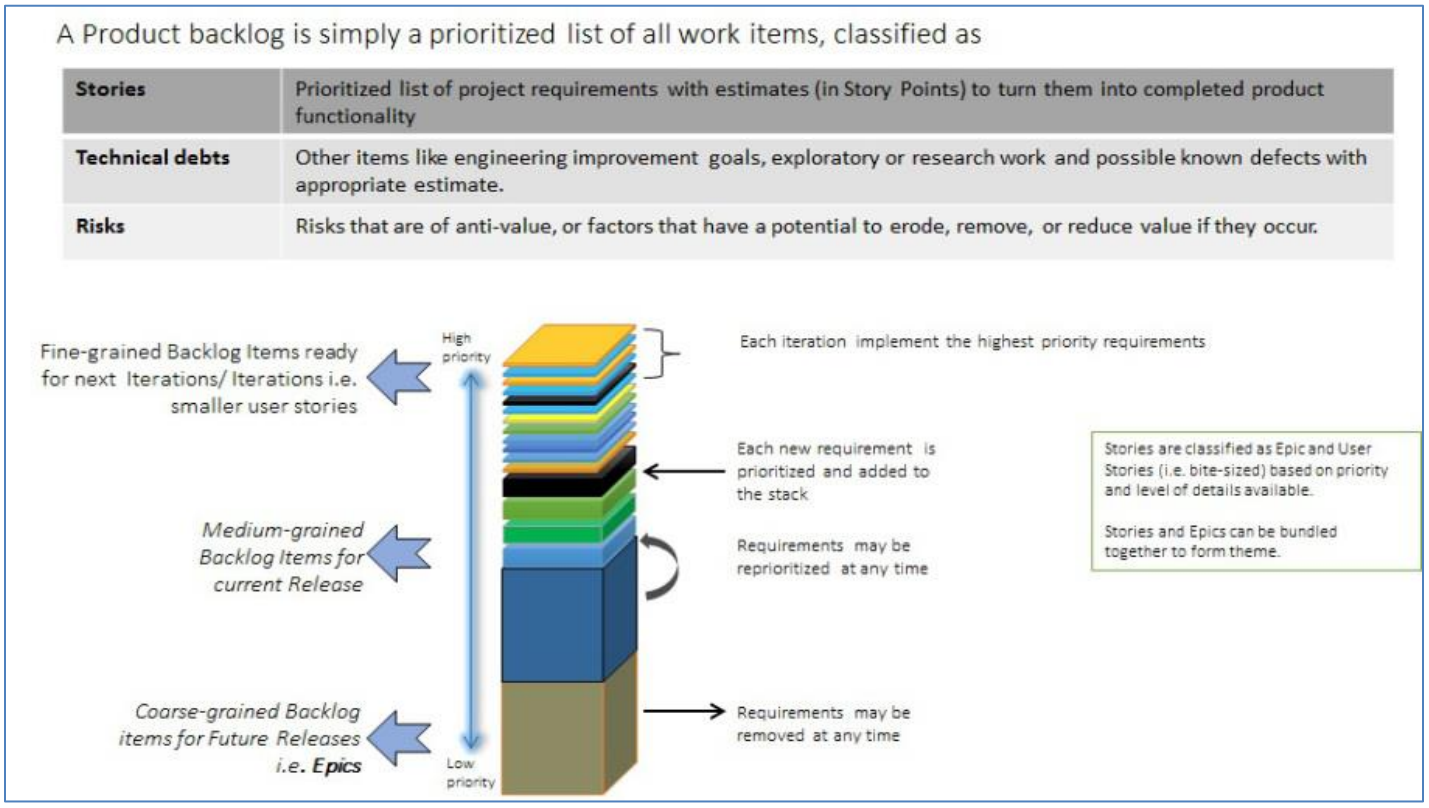
A Product Backlog is best described as DEEP

Detailed Appropriately: Higher order items are more detailed and well understood compared to lower order items

Estimated: Product backlog items are estimated i relative size by the development team. Product owner orders the items based on the value and the cost

Emergent: Product Backlog is a living artifact. It is always updated for details, estimates and order. The life of the product backlog is same as life of the product itself

Prioritized: Product backlog items are ordered based on the priority. The order is force ranking (1,2,3) so that there are no completing priorities



How do we determine the priority?

Customer value prioritization is concerned with working on the items that yield the highest value to the customer as soon as possible.

MoSCoW is a technique used to prioritize stories into four distinct categories:

- **M – MUST have this**
  Requirements that are fundamental to the system; without which system will not work and have no value and have to be included in the current delivery time box

- **S – SHOULD have this**
  Requirements that is important for project success; Important as MUST have but not as time-critical or have a work around's. In other words, not necessary for delivery in the current delivery time box

- **C – COULD have this**
  Requirements not necessary; can include if it increases customer satisfaction for little development cost

- **W – WON'T have this time, but WOULD like in the future**
  Alternatively WANT – No to this release

How do we Monitoring Progress Toward a Goal?

At any point in time, the total work remaining to reach a goal can be summed. The Product Owner tracks this total work remaining at least every Sprint Review. The Product Owner compares this amount with work remaining at previous Sprint Reviews to assess progress toward completing projected work by the desired time for the goal. This information is made transparent to all stakeholders.

Various projective practices upon trending have been used to forecast progress, like burn-downs, burn-ups, or cumulative flows. These have proven useful. However, these do not replace the importance of empiricism. In complex environments, what will happen is unknown. Only what has happened may be used for forward-looking decision-making.

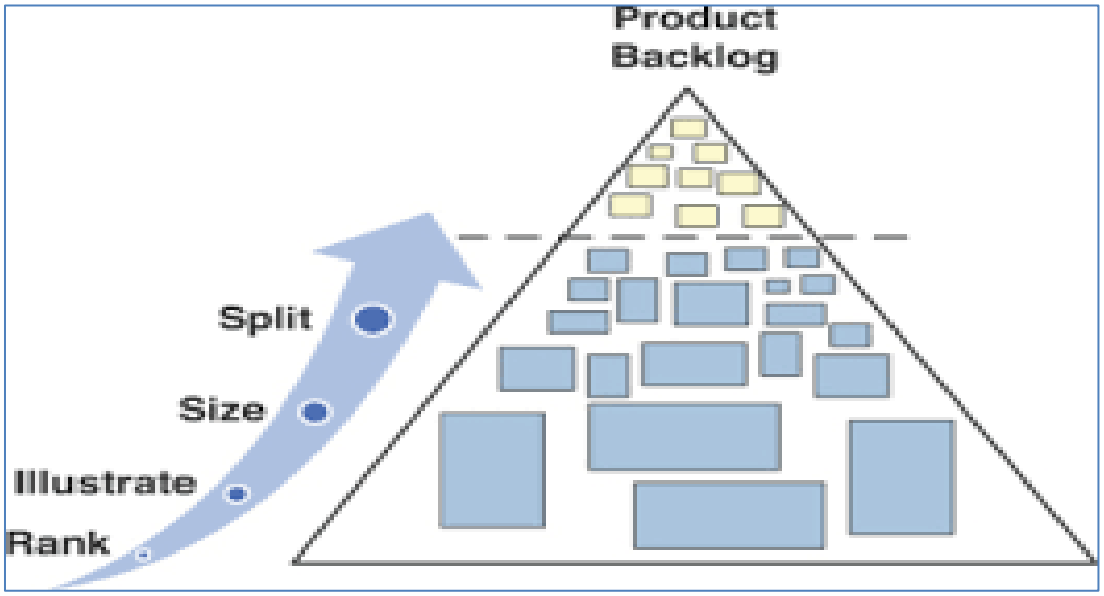
Product Backlog: Summary

- List of things that needs to be done to make the product come into existence
- The product backlog is the source for all product requirements
- The product owner sorts and prioritizes the backlog items
- The development team always works on the most important items based on the prioritized items in the product backlog
- The backlog is always prioritized before the current sprint
- Backlog refinement is done by both the product owner and the development team working in harmony
- The team estimates their capacity to attack the items in the product backlog

What is Product Backlog Refinement?

Product backlog refinement is the process through which product backlog items are reviewed by the Scrum team and revised, providing more detail and ensuring that there is greater clarity in the requirements for that item.

When product backlog is initially created it would have items of various sizes, clarity and value. But a scrum team needs clarity on few most important to get started. Backlog could be depicted as the following picture. Items at the top are important right now and should be smaller in size and more details so that team could start implementing them in upcoming sprints. As you go lower the product backlog, the items are less important ad less detailed. Product owner elaborates them as they become important.

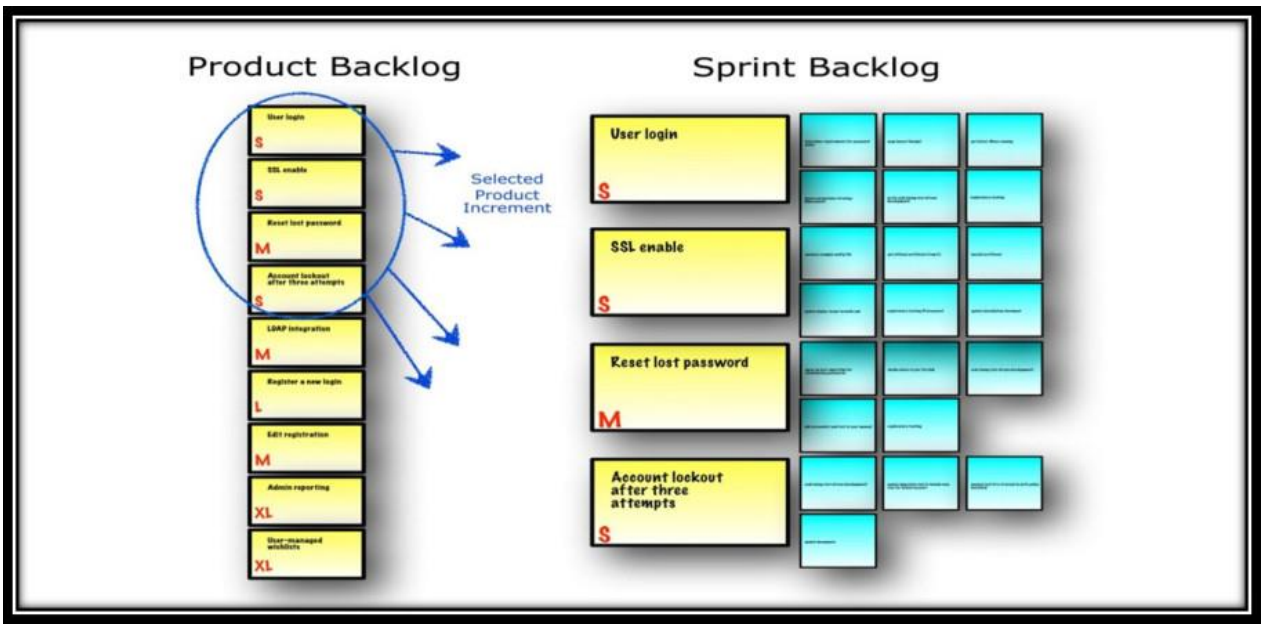


Scrum has an activity called "Product Backlog Refinement" to progressively elaborate the product backlog

- **Primary goal of product backlog refinement is to get ready with few items for upcoming one or two sprints**
- **Product Backlog items that are refined are deemed "Ready" for selection in a Sprint Planning**
- **Product Backlog Refinement is the act of adding detail, estimates and order to items in the Product Backlog**
- **Product Backlog Refinement is an ongoing activity throughout a Scrum project**
- **Team and PO decide the frequency and duration of backlog refinement meeting. However, it is time boxed at 10% of total available time.**

What is Sprint Backlog?

The Sprint Backlog is the set of Product Backlog items selected for the Sprint, plus a plan for delivering the product Increment and realizing the Sprint Goal. The Sprint Backlog is a forecast by the Development Team about what functionality will be in the next Increment and the work needed to deliver that functionality into a "Done" Increment. The Sprint Backlog makes visible all of the work that the Development Team identifies as necessary to meet the Sprint Goal.



The Sprint Backlog is a plan with enough detail that changes in progress can be understood in the Daily Scrum. The Development Team modifies the Sprint Backlog throughout the Sprint, and the Sprint Backlog emerges during the Sprint. This emergence occurs as the Development Team works through the plan and learns more about the work needed to achieve the Sprint Goal.

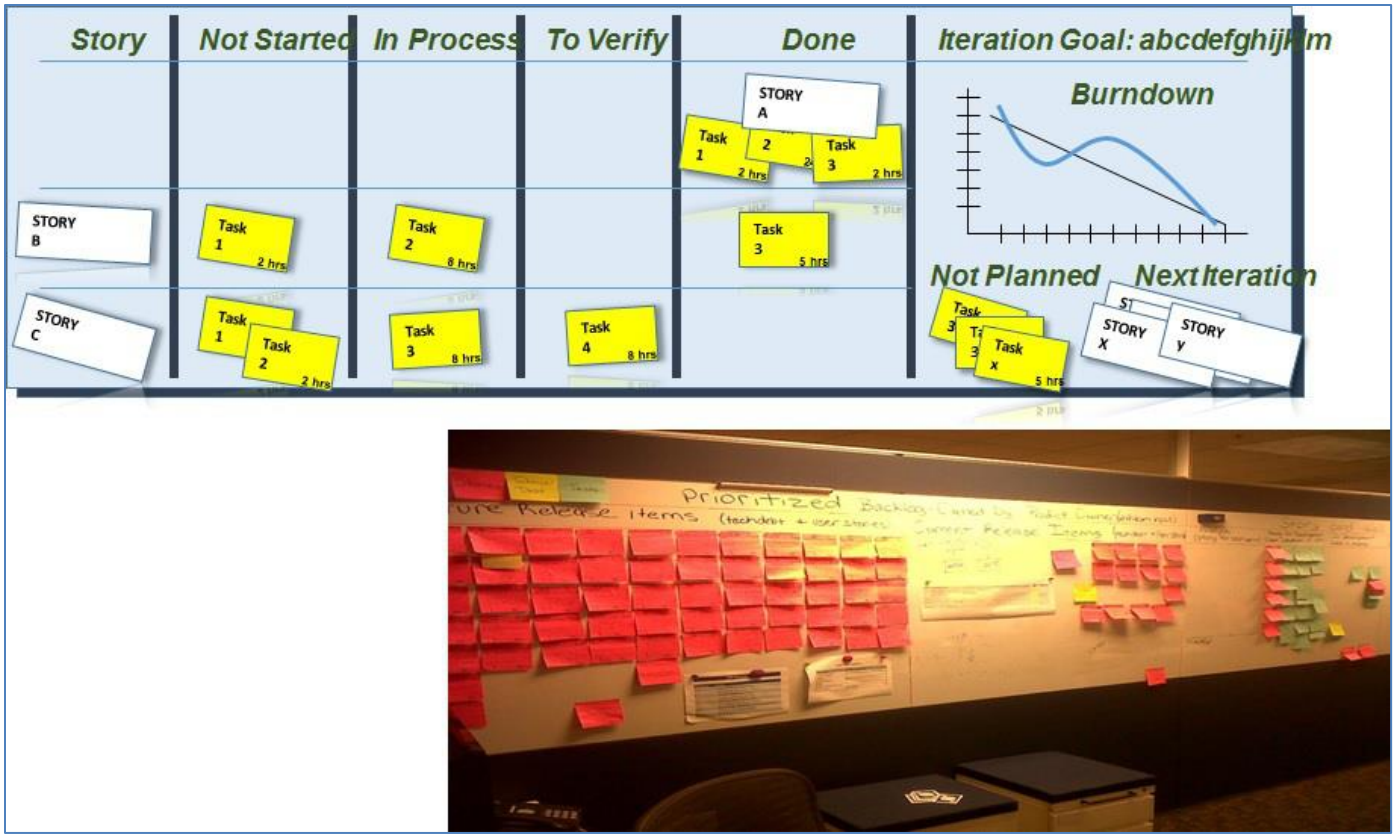
As new work is required, the Development Team adds it to the Sprint Backlog. As work is performed or completed, the estimated remaining work is updated. When elements of the plan are deemed unnecessary, they are removed. Only the Development Team can change its Sprint Backlog during a Sprint. The Sprint Backlog is a highly visible, real-time picture of the work that the Development Team plans to accomplish during the Sprint, and it belongs solely to the Development Team.

Sprint backlog should be transparent and visible to all the stakeholders. Many teams use physical task boards which are very visible all the time. However, distributed teams use some kind of electronic tool to share the sprint backlog. However, it is preferable to have a physical task board over electronic tool for the better information radiation.

Sprint Backlog: Summary

- Highest priority items take into the sprint for implementation and the plan to deliver those items is sprint backlog
- Sprint backlog is created during sprint planning
- Helps team see the total work involved in achieving the sprint goal
- Development team creates and manages the sprint backlog, this includes updating the time left of each task, create any forgotten tasks, update the status of each task etc.
- Team should keep the status of the items up to date so that the sprint progress is transparent
- All items should be updated at least once a day
- Team uses Daily SCRUM meeting to inspect and adapt the Sprint backlog

Visual Management of Sprint Backlog



How do we Monitoring Sprint Progress?

At any point in time in a Sprint, the total work remaining in the Sprint Backlog can be summed. The Development Team tracks this total work remaining at least for every Daily Scrum to project the likelihood of achieving the Sprint Goal. By tracking the remaining work throughout the Sprint, the Development Team can manage its progress.

How do we conclude the Increment is Done?

The Increment is the sum of all the Product Backlog items completed during a Sprint and the value of the increments of all previous Sprints. At the end of a Sprint, the new Increment must be "Done," which means it must be in useable condition and meet the Scrum Team's definition of "Done." It must be in useable condition regardless of whether the Product Owner decides to actually release it.

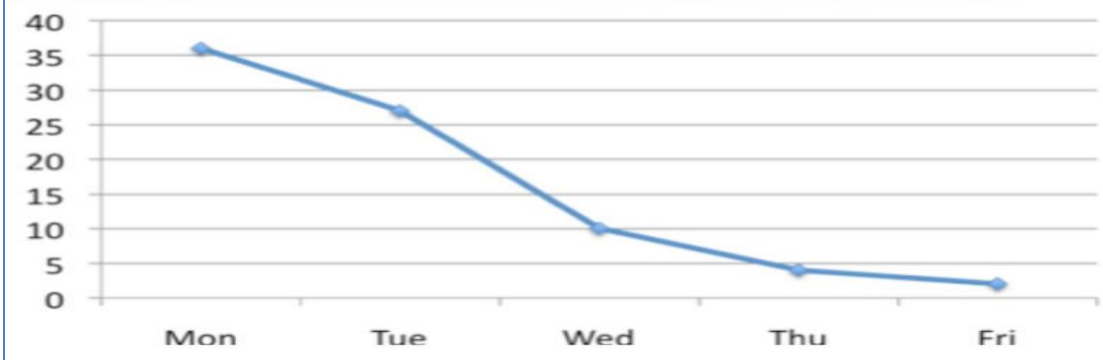
How will you track the Progress in Sprint?

Task boards are great information radiators. It is preferred to have a physical board over an electronic tool. Task boards give better feel for the progress of the sprint.

Explain Sprint Burn down Chart?

- **Primary method of tracking progress**
- **A Burn down chart shows how much work is left as of a date**
- **Scrum Master encourages team to use the Burn down chart as guidance in managing the sprint work**

Tasks	Mon	Tue	Wed	Thu	Fri
Design & Code UI	8	6	2		
Code Middle Tier	12	4			
Test Middle tier	12	16	8	4	2
Documentation	4	1			
Total work left	**36**	**27**	**10**	**4**	**2**



----------Discussions----------

What are the different Artifacts in scrum?

There are two Artifacts maintained in Scrum:

- Product Backlog – Containing the prioritized list of business requirements
- Sprint Backlog – Contains the user stories to be done by the scrum team for a sprint.

What is MVP in scrum?

A Minimum Viable Product is a product which has just the bare minimum required feature which can be demonstrated to the stakeholders and is eligible to be shipped to production.

Explain what is a product backlog in Scrum?

Before the scrum sprint initiates, product owner reviews the list of all new features, change requests, enhancements and bug reports and determines the priority. If the project is new, it includes new features that the new system must provide- this list of items is referred as Product Backlog. The items that are kept on sprint are referred as Sprint Backlog.

Explain what is Scrum Sprint?

Scrum project is developed in a series of "sprint". It is a repeatable and regular work cycle in scrum methodology during which work is accomplished and kept ready for review.

What is Scrum Sprint?

A Scrum Sprint is a regular, repeated work cycle in scrum methodology during which work is completed and made ready for review. Scrum sprints are basic units of development in the scrum methodology. Generally, scrum sprints are less than 30 days long.
Sprints contain and consist of the Sprint Planning, Daily Scrums, the development work, the Sprint Review, and the Sprint Retrospective.

During the Sprint:

- No changes are made that would endanger the Sprint Goal
- Quality goals do not decrease, and
- Scope may be clarified and re-negotiated between the Product Owner and Development Team as more is learned.

What are the Artifacts of Scrum process?

Scrum process Artifacts include

- **Sprint backlog** – The Sprint Backlog is the set of Product Backlog items selected for the Sprint, plus a plan for delivering the product Increment and realizing the Sprint Goal. The Sprint Backlog is a forecast by the Development Team about what functionality will be in the next Increment and the work needed to deliver that functionality into a "Done" Increment.
- **Product backlog** – The Product Backlog is an ordered list of everything that might be needed in the product and 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.
- **Velocity chart**- A velocity chart shows the sum of estimates of the work delivered across all iterations. Typically, velocity will stabilize through the life of a project unless the project team make-up varies widely or the length of the iteration changes.
- **Burn-down chart** – It is a chart that shows how quickly you and your team are burning through your customer's user stories. It shows the total effort against the amount of work we deliver on each iteration.

Explain what is a product backlog in Scrum?

Before the scrum sprint initiates, product owner reviews the list of all new features, change requests, enhancements and bug reports and determines which ones are of high priorities. If the project is new it includes new features that the new system must provide, this list of items is referred as Product Backlog. The items that are kept on sprint are referred as Sprint Backlog.

Explain the term "Increment"?

The term "Increment" is used to refer the total number of the product backlog items completed during the sprint and all previous sprints. At the end of the sprint, increment should be in done status; also, it must be in re-useable condition regardless of whether the product owner is willing to actually release a product or not.

1.5 User Stories Estimation

What is User Story?

User stories are well suited as product backlog items. User story is not a concept from Scrum but commonly used in Scrum projects to manage requirements. User Story is

As any user, I want to create my account

- **Token for the feature**
- **Usually written on a small index card**
- **Captures intent of the failure**
- **Captures conditions of satisfaction or acceptance test criteria**

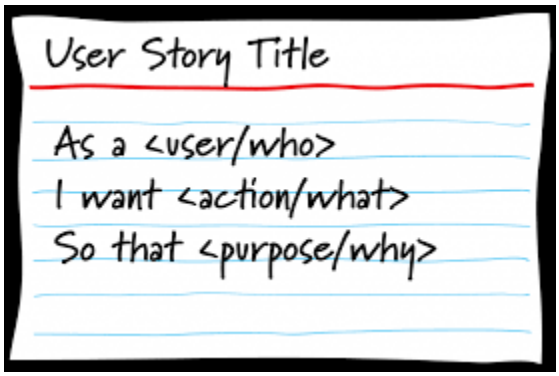
User story is written in which format?

As a <who> I want to <what> So that < why>

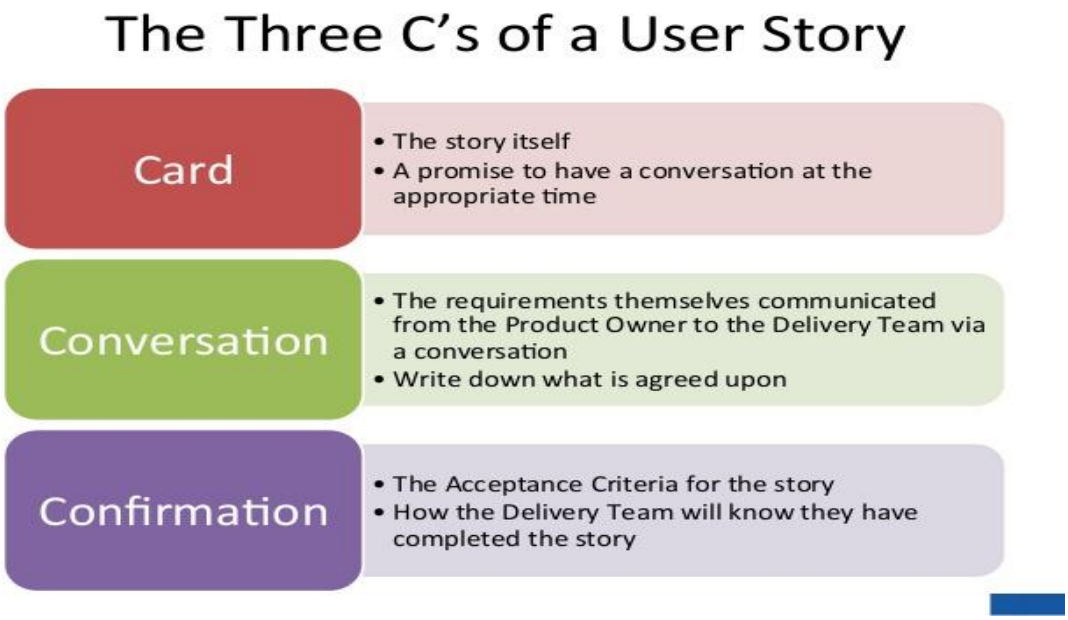
Who: User of the product

What: Activity user wants to perform using the feature

Why: Purpose the feature is going to serve.



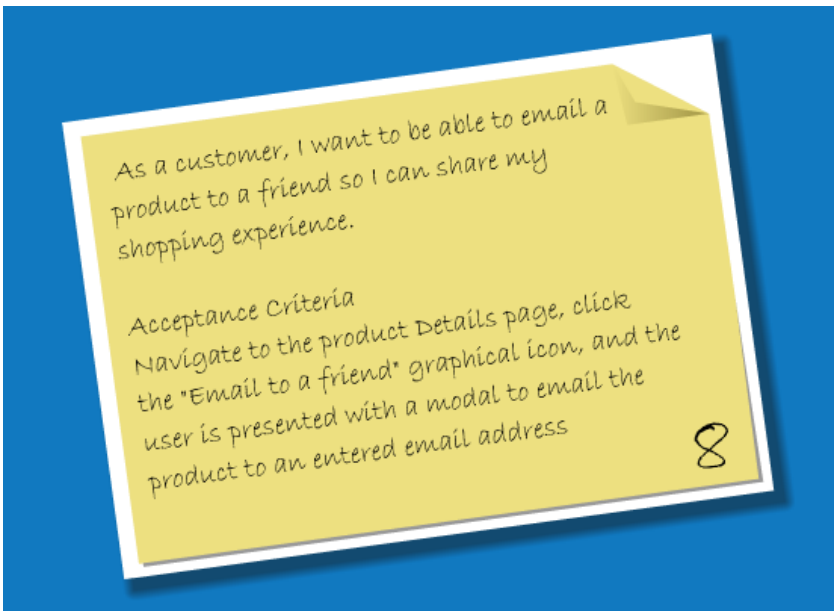
What 3C's of User story?



Acceptance Criteria of User Story?

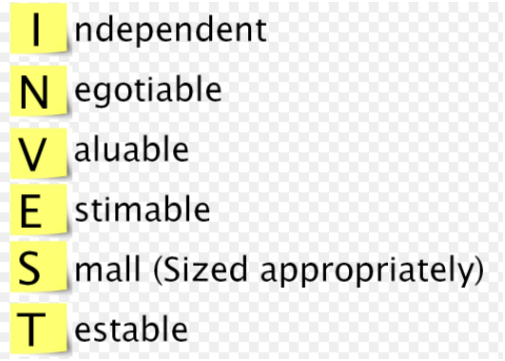
Acceptance Criteria is the criteria if the product satisfies the user story is considered done.

Acceptance Criteria is basis for high level test cases.



How to write good user stories

It is essential to write user stories, which are small and independent enough to be worked on in each sprint. For a good product backlog, the team should INVEST in good user stories:



User Story should have the INVEST characteristics:

Independent
One user story should be independent of another (as much as possible).

Negotiable
A user story is negotiable. The "Card" of the story is just a short description of the story which do not include details. The details are worked out during the "Conversation" phase.

Valuable
Each story has to be of value to the customer (either the user or the purchaser).

Estimateable
The developers need to be able to estimate (at a ballpark even) a user story to allow prioritization and planning of the story.

Small
A good story should be small in effort, typically representing no more, than 2-3 person days of effort. A story which is more than that in effort can have more errors associated with scoping and estimation.

Testable
A story needs to be testable for the "Confirmation" to take place.

----------Discussions----------

Explain what is user stories in Scrum?

In scrum, user stories are short, one sentence definitions of a feature or functionality.

How requirements are defined in a scrum?

Requirements are termed as "User Stories" in Scrum.

How do you define a user story? What type of requirements did you use for your teams?

Requirements in Scrum are written as user stories using a standard

The user stories are defined in the format of

As a <User / type of user> I want to <action / feature to implement> So that < objective>

As a Scrum Master, you don't necessarily write user stories, but you would assist the Product Owner to ensure that user stories are written, prioritized, and ready for the sprint.

How do you measure the complexity or effort in a sprint? Is there a way to determine and represent it?

Complexity and effort is measured through "Story Points".

In scrum it's recommended to use Fibonacci series to represent it.

How do you calculate a story point?

A Story point is calculated by taking into the consideration the development effort+ testing effort + resolving dependencies and other factors that would require to complete a story.

Is it possible that you come across different story point for development and testing efforts? In that case how do you resolve this conflict?

Yes, this is a very common scenario. There may be a chance that the story point given by the development team is, say 3 but the tester gives it 5. In that case both the developer and tester have to justify their story point, have discussion in the meeting and collaborate to conclude a common story point.

You are in the middle of a sprint and suddenly the product owner comes with a new requirement, what will you do?

In ideal case, the requirement becomes a story and moves to the backlog. Then based on the priority, team can take it up in the next sprint. But if the priority of the requirement is really high, then the team will have to accommodate it in the sprint but it has to very well communicated to the stakeholder that incorporating a story in the middle of the sprint may result in spilling over few stories to the next sprint.

In case you receive a story at the last day of the sprint to test and you find there are defects, what will you do? Will you mark the story to done?

A story is done only when it is development complete + QA complete + acceptance criteria is met + it is eligible to be shipped into production. In this case if there are defects, the story is partially done and not completely done, so I will spill it over to next sprint.

What are Epics?

Epics are equivocal user stories or we can say these are the user stories which are not defined and are kept for future sprints

What is difference between Epic, User stories & Tasks?

Epic is a group of related user stories.

User Stories define the actual business requirement. Generally created by the business owner.

Tasks is to accomplish the business requirements; development team create tasks.

Explain what is scrum poker or Planning Poker?

Scrum poker or planning poker is a technique to estimate the relative size of development goals in software development. It is a way to determine sprint item durations by playing number cards face down the table, instead of speaking them aloud.

Explain what is a Story Point in Scrum?

Each feature in scrum is Story. Story point is an arbitrary measure used by Scrum teams, and it is a metric used by agile teams to determine the difficulty of implementing a given story.

How long were your sprints?

An ideal sprint length is between 1 and 4 four weeks, with a 2-week sprint being the most widely used.

1.6 Definition of Done

What is Definition of Done (DoD)?

When a Product Backlog item or an Increment is described as "Done", everyone must understand what "Done" means. Although this varies significantly per Scrum Team, members must have a shared understanding of what it means for work to be complete, to ensure transparency. This is the definition of "Done" for the Scrum Team and is used to assess when work is complete on the product Increment.

The same definition guides the Development Team in knowing how many Product Backlog items it can select during a Sprint Planning. The purpose of each Sprint is to deliver Increments of potentially releasable functionality that adhere to the Scrum Team's current definition of "Done."

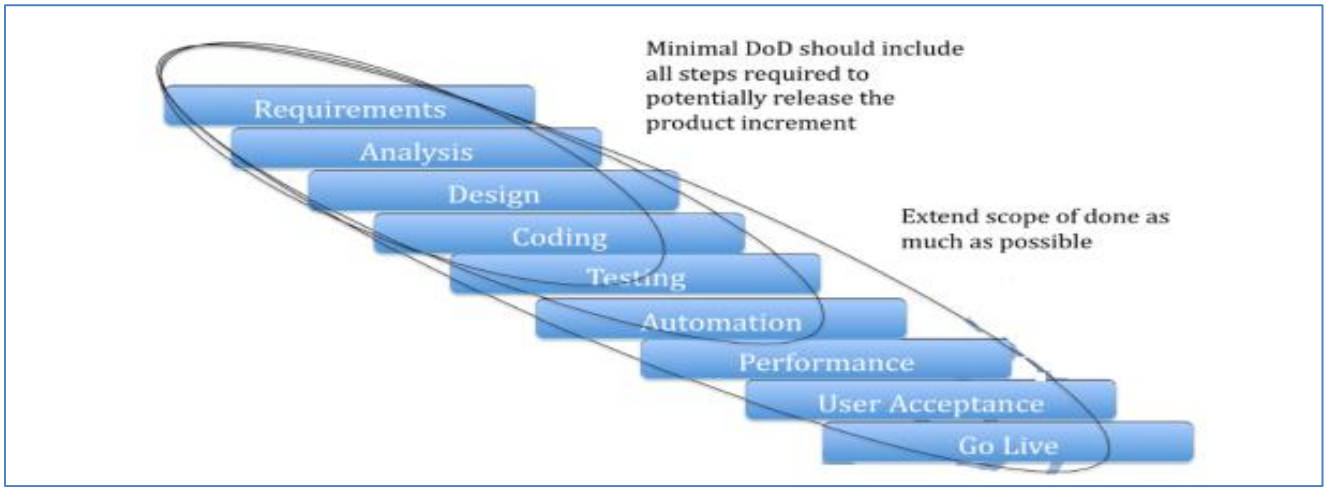
Development Teams deliver an Increment of product functionality every Sprint. This Increment is useable, so a Product Owner may choose to immediately release it. If the definition of "done" for an increment is part of the conventions, standards or guidelines of the development organization, all Scrum Teams must follow it as a minimum. If "done" for an increment is not a convention of the development organization, the Development Team of the Scrum Team must define a definition of "done" appropriate for the product. If there are multiple Scrum Teams working on the system or product release, the development teams on all of the Scrum Teams must mutually define the definition of "Done."

Each Increment is additive to all prior Increments and thoroughly tested, ensuring that all Increments work together.

As Scrum Teams mature, it is expected that their definitions of "Done" will expand to include more stringent criteria for higher quality. Any one product or system should have a definition of "Done" that is a standard for any work done on it.

- **Definition of Done is a list of attractive activities agreed by the Product Owner and the Development Team to call a backlog item is done**
- **Definition of Done consist of activities needed for functional and quality requirements**
- **Team comes up with the DOD and adheres to it while creating the product increment**

- Different teams may have different DOD but all teams should follow minimal DOD that includes all critical activities required
- If there are standards at organizational level, a common DOD can capture those and the teams should have separate DOD in addition to one at the organizational level
- A Stronger DOD leads to higher quality product: -
    - Code Complete
    - Unit tests Written
    - Code Review
    - Manual Functional Testing
    - Automation
    - Updated Documents
    - User Acceptance Testing
    - Successful Deployment
- Suppose if the DOD is missing essential activities, it is called a Weak DOD. For E.g. Load testing may not be done for every sprint and is deferred to later time.
- A weak DOD causes unfinished work in every sprint. The unfinished work is added back to product backlog. This increased the risk. If a major bug is found during the load testing, that could risk the release
- Weak DOD also increases the Technical Debt. This might include the automation or code reviews



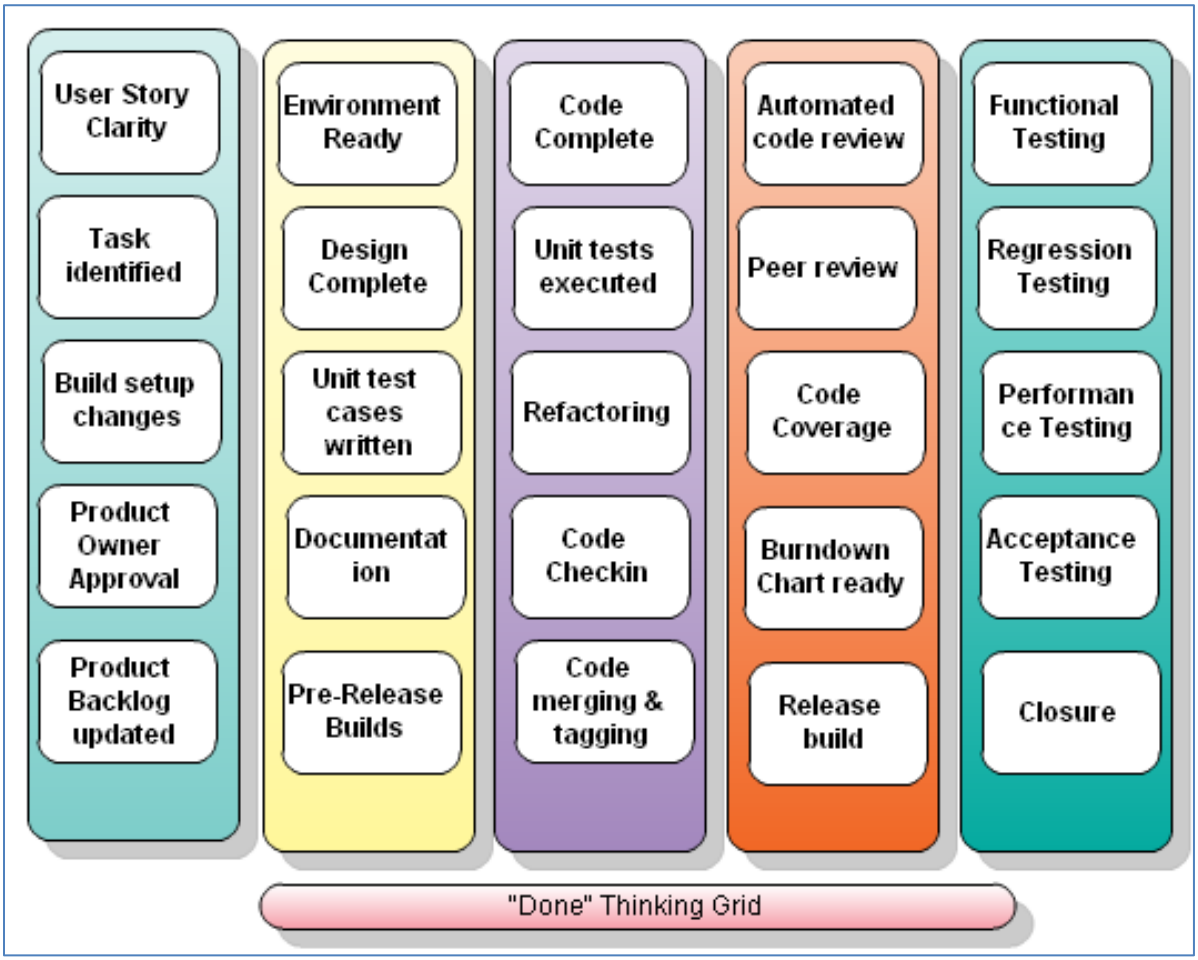
What is DoD? How is this achieved?

DoD stands for Definition of done. It is achieved when

- The story is development complete,
- QA is complete,
- The story meets and satisfy the acceptance criteria
- Regression around the story is complete

Then feature is eligible to be shipped / deployed in production

Explain DOD in detail for senior management?



User Story Clarity
User stories selected for the sprint are complete with respect to product theme, understood by the team, and have been validated by the detailed acceptance criteria.

Tasks Identified
Tasks for selected user stories have been identified and estimated by the team.

These first two topics are typically covered in the sprint planning meeting but are mentioned as done criteria for the sake of verification.

Build and package changes
Build and package changes have been communicated to the build master. These changes have been implemented, tested and documented to ensure that they cover the features of the sprint.

Product owner approval
Each finished user story has been passed through UAT (User Acceptance Testing) and signed off as meeting requirements

Updating Product Backlog
All features not done during the sprint are added back to the product backlog. All incidents/defects not handled during the sprint are added to the product backlog.

Environment ready
- **Development environment is ready with all third-party tools configured.**
- **Staging environment is ready.**
- **Continuous integration framework is in place. The build engine is configured to schedule hourly, nightly, and release builds.**
- **Desired build automation is in place. Why "desired"? Because there is no end to build automation**
- **Test data for the selected features has been created**

Design complete
Design analysis is complete as per the user story or theme. UML diagrams are either created or updated for the feature under development.

You might need to prototype various components to ensure that they work together. Wireframes and prototype have been created and approved by the respective stakeholders.

Unit test cases written
Unit test cases have been written for the features to be developed.

Documentation Ready
Documentation (Just enough or whatever the team agrees to) to support the sprint demo is ready

Pre-Release builds
Pre-release builds (hourly/nightly) have been happening and nightly build reports have been published on regular basis. The following could/should be part of pre-release builds:

- **Compile and execute unit test cases (mandatory)**
- **Creation of cross reference of source code**
- **Execution of automated code reviews for verification of coding rules**
- **Code coverage reports are generated**
- **Detection of duplicate source code**

- Dependency analysis and generation of design quality matrix (static analysis, cyclomatic complexity)
- Auto deployment in staging environment

It comes down to build automation; there is no end to what can be achieved from automated hourly, nightly builds. The team along with the product owner needs to decide on how much build automation is required.

Code Complete
Source code changes are done for all the features in the "to do" list." Source code has been commented appropriately.

Unit testing is done
Unit test cases have been executed and are working successfully

Code Refactoring
Source code has been refactored to make it comprehensive, maintainable and, amenable to change.

A common mistake is to not keep refactoring in the definition of done. If not taken seriously, refactoring normally spills out to next sprint or, worse, is completely ignored.

Code check-in
Source code is checked in the code library with appropriate comments added.

If project is using tools which help in maintaining traceability between user stories and the respective source code, proper check-in guidelines are followed.

Code merging and tagging
Finalized source code has been merged with the main branch and tagged appropriately (merging and tagging guidelines are to be used)

Automated Code reviews
Automated code review has been completed using the supported tools/technologies. Violations have been shared with the team and the team has resolved all discrepancies to adhere to the coding standard. (Automated code reviews should be hooked up with CI builds.)

Peer reviews
Peer reviews are done. If pair programming is used, a separate peer review session might not be required.

Code coverage is achieved
Code coverage records for each package are available and whatever the team has decided as the minimum benchmark is achieved.

Project metrics are ready
Burndown chart has been updated regularly and is up to date.

Release Build

- **Build and packaging**
  A Build (successful) is done using continuous integration framework. Change log report has been generated from Code Library and Release notes have been created. Deliverables have been moved to release area.
- **Build deployment in staging environment**
  Build deliverables are deployed in staging environment. If it is easy, this step should be automated.

Functional testing done

- **Automated testing**
  All types of automated test cases have been executed and a test report has been generated. All incidents/defects are reported.
- **Manual testing**
  Quality assurance team has reviewed the reports generated from automation testing and conducted necessary manual test cases to ensure that tests are passing. All incidents/defects are reported.
- **Build issues**
  If any integration or build issues are found, the necessary steps are repeated and respective "Done" points are adhered to.

Regression testing done

Regression testing is done to ensure that defects have not been introduced in the unchanged area of the software.

Performance testing done

A common mistake is to not keep performance testing in the definition of done. This is an important aspect. Most performance issues are design issues and are hard to fix at a later stage.

Acceptance testing done

Each finished user story has been passed through UAT (User Acceptance Testing) and signed off as meeting.

Closure

All finished user stories/tasks are marked complete/resolved. Remaining hours for task set to zero before closing the task.

- **Other necessary/optional activities**
  Anything which is very specific to the project environment
  These might include:
  - Security audit sign off
  - Deployable on multiple platforms such as Windows, Linux, and Mac OS X

1.7 Product Increment

What is Product Increment?

Every sprint produces a product increment, the most important scrum artifact. A product increment is the "goal line" for each sprint and, at the end of the sprint, it must: -

- Be of high enough quality to be given to users
- Meet the scrum team's current definition of done
- Be acceptable to the product owner with properly tested, completed in full shape and ready to use

At the end of each sprint, the team must produce a potentially shippable product increment with the following features

- High Quality
- Tested
- Completed
- Ready to Use
- As per Definition of Done

Should the team always release the Product Increment?

- It depends. If the product increment that is produced is usable and adds the value to the business, product owner may choose to release it right away.
- Though the product increment is working, it may not be the feature complete and product owner may not want to release it
- Some business doesn't want to surprise their customers too often by frequent release
- Whether the product increment is shipped or not, building working software every sprint eliminates the technical uncertainty

If Product Increment gets failed What to do in Scrum?

The following are the reasons of Product Increment gets failed,

- **If the team doesn't complete all the forecasted Product Backlog Items for the Sprint?**
- **If the team does not reach the Sprint Goal?**
- **Different option?**

Now its Product Owner call to reiterate the Sprint since Product Increment fails. Even though Product Owner address and clarifies the requirements at early stage, Till team fails due to technical debts.

1.8 Scrum Events

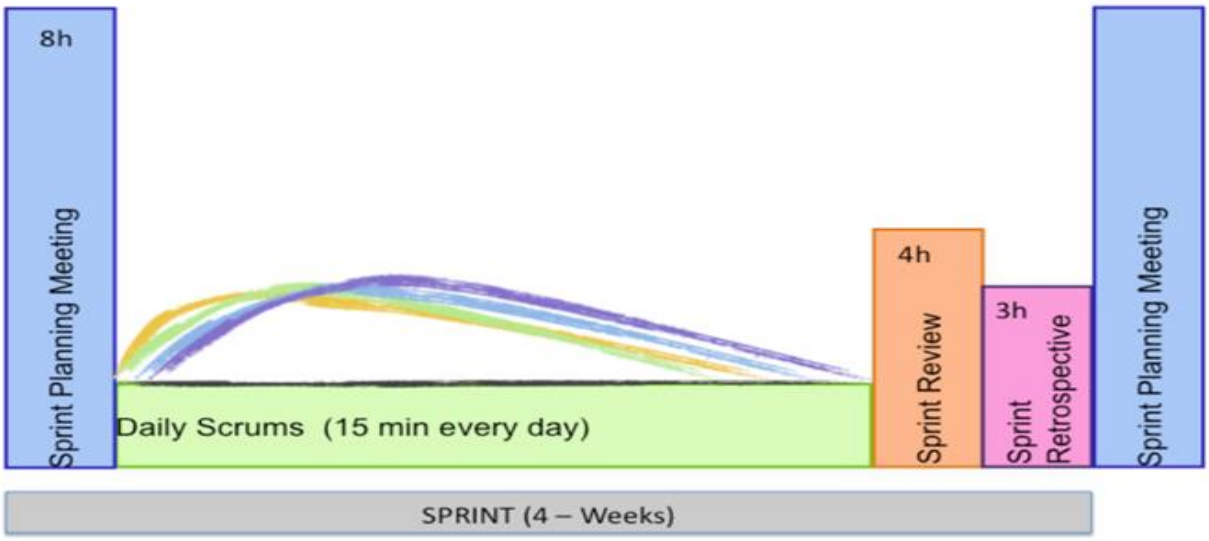
What are the Scrum Events?

Prescribed events are used in Scrum to create regularity and to minimize the need for meetings not defined in Scrum. All events are time-boxed events, such that every event has a maximum duration. Once a Sprint begins, its duration is fixed and cannot be shortened or lengthened. The remaining events may end whenever the purpose of the event is achieved, ensuring an appropriate amount of time is spent without allowing waste in the process.

Other than the Sprint itself, which is a container for all other events, each event in Scrum is a formal opportunity to inspect and adapt something. These events are specifically designed to enable critical transparency and inspection. Failure to include any of these events results in reduced transparency and is a lost opportunity to inspect and adapt.

List out scrum events

- **Sprint Planning**
- **Daily Scrum**
- **Sprint Review**
- **Sprint Retrospective**



What do you mean by the Sprint?

The heart of Scrum is a Sprint, a time-box of one month or less during which a "Done", useable, and potentially releasable product Increment is created. Sprints best have consistent durations throughout a development effort. A new Sprint starts immediately after the conclusion of the previous Sprint. Ideally Sprint is 2 weeks.

Sprints contain and consist of the Sprint Planning, Daily Scrums, the development work, the Sprint Review, and the Sprint Retrospective.

What are the activities of Sprint? | How Sprint is protected?

- **No changes are made that would endanger the Sprint Goal;**
- **Quality goals do not decrease; and,**
- **Scope may be clarified and re-negotiated between the Product Owner and Development Team as more is learned.**

Each Sprint may be considered a project with no more than a one-month horizon. Like projects, Sprints are used to accomplish something. Each Sprint has a definition of what is to be built, a design and flexible plan that will guide building it, the work, and the resultant product.

Sprints are limited to one calendar month. When a Sprint's horizon is too long the definition of what is being built may change, complexity may rise, and risk may increase. Sprints enable predictability by ensuring inspection and adaptation of progress toward a Sprint Goal at least every calendar month. Sprints also limit risk to one calendar month of cost.

When to cancel a Sprint?

A Sprint can be cancelled before the Sprint time-box is over. Only the Product Owner has the authority to cancel the Sprint, although he or she may do so under influence from the stakeholders, the Development Team, or the Scrum Master.

A Sprint would be cancelled if the Sprint Goal becomes obsolete. This might occur if the company changes direction or if market or technology conditions change. In general, a Sprint should be cancelled if it no longer makes sense given the circumstances. But, due to the short duration of Sprints, cancellation rarely makes sense.

When a Sprint is cancelled, any completed and "Done" Product Backlog items are reviewed. If part of the work is potentially releasable, the Product Owner typically accepts it. All incomplete Product Backlog Items are re-estimated and put back on the Product Backlog. The work done on them depreciates quickly and must be frequently re-estimated.

Sprint cancellations consume resources, since everyone has to regroup in another Sprint Planning to start another Sprint. Sprint cancellations are often traumatic to the Scrum Team, and are very uncommon.

What are the activities while performing using Sprint Planning?

The work to be performed in the Sprint is planned at the Sprint Planning. This plan is created by the collaborative work of the entire Scrum Team.

Input: Refined Product Backlog | Latest Product Increment | Team Capacity

Outcome: Sprint Goal | Sprint Backlog

Sprint Planning is time-boxed to a maximum of eight hours for a one-month Sprint. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendants understand its purpose. The Scrum Master teaches the Scrum Team to keep it within the time-box.

Sprint Planning answers the following:

- What can be delivered in the Increment resulting from the upcoming Sprint?
- How will the work needed to deliver the Increment be achieved?

Discussion: What can be done this Sprint?

The Development Team works to forecast the functionality that will be developed during the Sprint. The Product Owner discusses the objective that the Sprint should achieve and the Product Backlog items that, if completed in the Sprint, would achieve the Sprint Goal. The entire Scrum Team collaborates on understanding the work of the Sprint.

The input to this meeting is the Product Backlog, the latest product Increment, projected capacity of the Development Team during the Sprint, and past performance of the Development Team. The number of items selected from the Product Backlog for the Sprint is solely up to the Development Team. Only the Development Team can assess what it can accomplish over the upcoming Sprint.

After the Development Team forecasts the Product Backlog items it will deliver in the Sprint, the Scrum Team crafts a Sprint Goal. The Sprint Goal is an objective that will be met within the Sprint through the implementation of the Product Backlog, and it provides guidance to the Development Team on why it is building the Increment.

Discussion: How will the chosen work get done?

Having set the Sprint Goal and selected the Product Backlog items for the Sprint, the Development Team decides how it will build this functionality into a "Done" product Increment during the Sprint. The Product Backlog items selected for this Sprint plus the plan for delivering them is called the Sprint Backlog.

The Development Team usually starts by designing the system and the work needed to convert the Product Backlog into a working product Increment. Work may be of varying size, or estimated effort. However, enough work is planned during Sprint Planning for the Development Team to forecast what it believes it can do in the upcoming Sprint. Work planned for the first days of the Sprint by the Development Team is decomposed by the end of this meeting, often to units of one day or less. The

Development Team self-organizes to undertake the work in the Sprint Backlog, both during Sprint Planning and as needed throughout the Sprint.

The Product Owner can help to clarify the selected Product Backlog items and make trade-offs. If the Development Team determines it has too much or too little work, it may renegotiate the selected Product Backlog items with the Product Owner. The Development Team may also invite other people to attend in order to provide technical or domain advice.

By the end of the Sprint Planning, the Development Team should be able to explain to the Product Owner and Scrum Master how it intends to work as a self-organizing team to accomplish the Sprint Goal and create the anticipated Increment.

Sprint Goal

A short statement of what the work will be focused on during the sprint.

The Sprint Goal is an objective set for the Sprint that can be met through the implementation of Product Backlog. It provides guidance to the Development Team on why it is building the Increment. It is created during the Sprint Planning meeting.

The Sprint Goal gives the Development Team some flexibility regarding the functionality implemented within the Sprint. The selected Product Backlog items deliver one coherent function, which can be the Sprint Goal. The Sprint Goal can be any other coherence that causes the Development Team to work together rather than on separate initiatives.

As the Development Team works, it keeps the Sprint Goal in mind. In order to satisfy the Sprint Goal, it implements the functionality and technology. If the work turns out to be different than the Development Team expected, they collaborate with the Product Owner to negotiate the scope of Sprint Backlog within the Sprint.

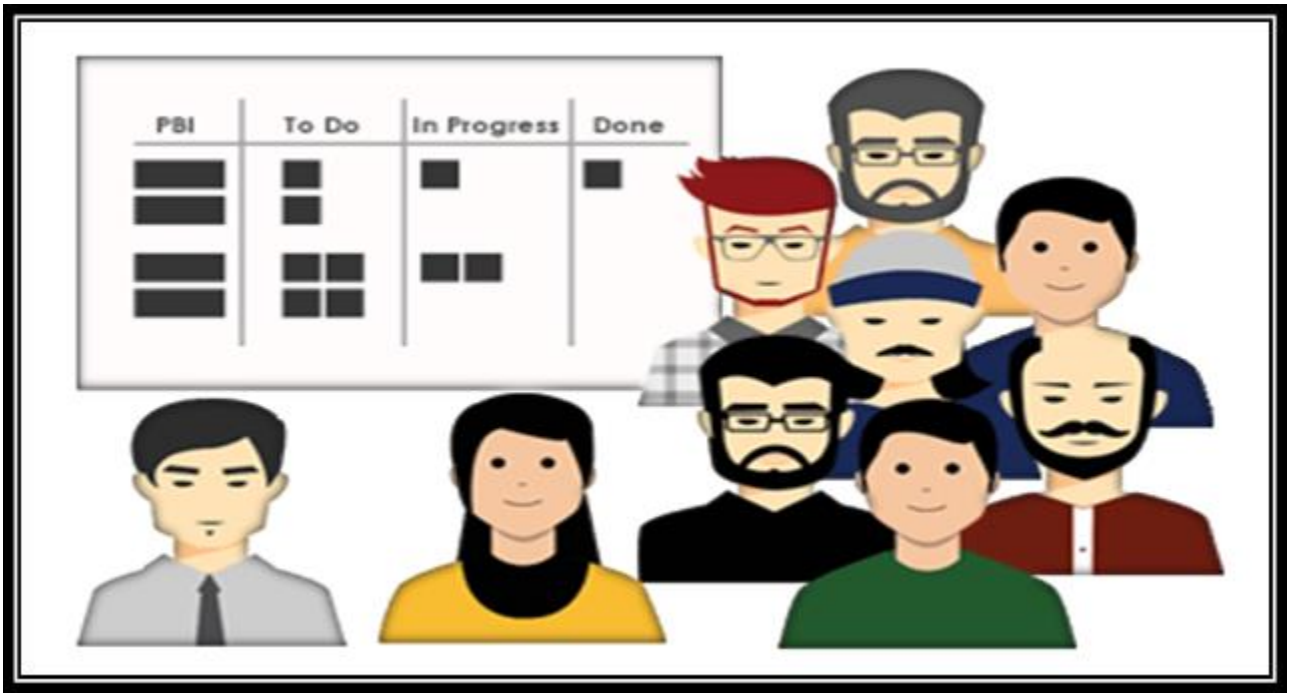
Sprint Planning Summary

- Product Owner and Project team needs to discuss the goals of the upcoming sprint
- Product Owner and team negotiates stories to select in current sprint from prioritized product backlog for the upcoming sprint
    - Selected stories are estimated with agreed acceptance criteria
    - Team identifies and estimates Task, discusses how the work will be accomplished
- Scrum Master, Product Owner and team can attend the meeting

- Product Backlog has to be groomed by PO prior to sprint planning meeting, Product owner reviews with the team items in the updated backlog
- Self-organized Development team defines how the work will be done in the goals of the sprint will be achieved
- Normally split into 2 sets of 4 hours each Timebox: Max 8 hours per sprint
    - First half for choosing the product backlog items with PO
    - Second half for splitting into tasks and assignment (Product Owner is optional for the second half)
- Artifacts – Sprint Goal, Sprint Backlog (Output of Sprint Planning)

Daily Scrum

The Daily Scrum is a 15-minute time-boxed event for the Development Team to synchronize activities and create a plan for the next 24 hours. This is done by inspecting the work since the last Daily Scrum and forecasting the work that could be done before the next one.



Input: Sprint Backlog

Outcome: Updated Sprint Backlog

The Daily Scrum is held at the same time and place each day to reduce complexity. During the meeting, the Development Team members explain:

- **What did I do yesterday that helped the Development Team meet the Sprint Goal?**
- **What will I do today to help the Development Team meet the Sprint Goal?**
- **Do I see any impediment that prevents me or the Development Team from meeting the Sprint Goal?**

The Development Team uses the Daily Scrum to inspect progress toward the Sprint Goal and to inspect how progress is trending toward completing the work in the Sprint Backlog. The Daily Scrum optimizes the probability that the Development Team will meet the Sprint Goal. Every day, the Development Team should understand how it intends to work together as a self-organizing team to accomplish the Sprint Goal and create the anticipated Increment by the end of the Sprint. The Development Team or team members often meet immediately after the Daily Scrum for detailed discussions, or to adapt, or re-plan, the rest of the Sprint's work.

The Scrum Master ensures that the Development Team has the meeting, but the Development Team is responsible for conducting the Daily Scrum. The Scrum Master teaches the Development Team to keep the Daily Scrum within the 15-minute time-box.

The Scrum Master enforces the rule that only Development Team members participate in the Daily Scrum. Daily Scrums improve communications, eliminate other meetings, identify impediments to development for removal, highlight and promote quick decision-making, and improve the Development Team's level of knowledge. This is a key inspect and adapt meeting.

Daily Scrum Summary

- **The daily scrum is also known as a stand-up meeting, typically first activity of the day**
- **This is a 15-minute timeboxed meeting**
- **The daily scrum is held every day at the same time and location**
- **The whole world is invited**
- **The daily scrum is for the development team only, facilitated by Scrum Master, Product Owner optional**
- **This is not problem-solving meeting. Pigs can talk, Chickens observe**
- **Scrum Master to intervene to bring in discipline after due attempts at self-correction**
- **Daily Scrum Meeting Questions for the team members**

- What have I done since the last daily scrum?
- What do I plan to do today?
- Are there any impediments to my progress?

Sprint Review

A Sprint Review is held at the end of the Sprint to inspect the Increment and adapt the Product Backlog if needed.

During the Sprint Review, the Scrum Team and stakeholders collaborate about what was done in the Sprint. Based on that and any changes to the Product Backlog during the Sprint, attendees collaborate on the next things that could be done to optimize value. This is an informal meeting, not a status meeting, and the presentation of the Increment is intended to elicit feedback and foster collaboration.

This is a four-hour time-boxed meeting for one-month Sprints. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendants understand its purpose. The Scrum Master teaches all to keep it within the time-box.

Input: Product Increment

Outcome: Product Backlog (Revised)

The Sprint Review includes the following elements:

- **Attendees include the Scrum Team and key stakeholders invited by the Product Owner;**
- **The Product Owner explains what Product Backlog items have been "Done" and what has not been "Done";**

- **The Development Team discusses what went well during the Sprint, what problems it ran into, and how those problems were solved;**
- **The Development Team demonstrates the work that it has "Done" and answers questions about the Increment;**
- **The Product Owner discusses the Product Backlog as it stands. He or she projects likely completion dates based on progress to date (if needed);**
- **The entire group collaborates on what to do next, so that the Sprint Review provides valuable input to subsequent Sprint Planning;**
- **Review of how the marketplace or potential use of the product might have changed what is the most valuable thing to do next; and,**
- **Review of the timeline, budget, potential capabilities, and marketplace for the next anticipated release of the product.**

The result of the Sprint Review is a revised Product Backlog that defines the probable Product Backlog items for the next Sprint. The Product Backlog may also be adjusted overall to meet new opportunities.

Sprint Review Summary

- **Team presents what it accomplished during the sprint, typically takes the form of a demo of new features or underlying architecture. Done from QA server (Close to Prod)**
- **Hosted at the end of every sprint Timebox: Max 4 hours per sprint**
- **Attendees will be the development team, the product owner, scrum master, and sometimes other project stakeholders**
- **The development team will demo the work created in the increment**
- **The group will decide if "Done" has been achieved**
- **Stakeholders can provide comments which go in to the product backlog**
- **The development team and the product owner will discuss the sprint and the remaining items in the product backlog further procced with**

Sprint Retrospective

The Sprint Retrospective is an opportunity for the Scrum Team to inspect itself and create a plan for improvements to be enacted during the next Sprint.

Input: Feedback | Experience of team members
Outcome: List of improvements

The Sprint Retrospective occurs after the Sprint Review and prior to the next Sprint Planning. This is a three-hour time-boxed meeting for one-month Sprints. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendants understand its purpose. The Scrum Master teaches all to keep it within the time-box. The Scrum Master participates as a peer team member in the meeting from the accountability over the Scrum process.

The purpose of the Sprint Retrospective is to:

- **Inspect how the last Sprint went with regards to people, relationships, process, and tools;**
- **Identify and order the major items that went well and potential improvements; and,**
- **Create a plan for implementing improvements to the way the Scrum Team does its work.**

The Scrum Master encourages the Scrum Team to improve, within the Scrum process framework, its development process and practices to make it more effective and enjoyable for the next Sprint.

During each Sprint Retrospective, the Scrum Team plans ways to increase product quality by adapting the definition of "Done" as appropriate.

By the end of the Sprint Retrospective, the Scrum Team should have identified improvements that it will implement in the next Sprint. Implementing these improvements in the next Sprint is the adaptation to the inspection of the Scrum Team itself. Although improvements may be implemented at any time, the Sprint Retrospective provides a formal opportunity to focus on inspection and adaptation.

Sprint Retrospective Summary

- **The development team meeting posted after the sprint review, but before the next sprint planning meeting**
- **Periodically take a look at what is and is not working**
- **This is a meeting to inspect an adapt Timebox: Max 4 hours per sprint**
- **Lessons learned and opportunities for improvement**
- **Review of the product owner's feedback about the last iteration**
- **An opportunity to improve on their approach based on the retrospective and the last sprint**

----------Discussions----------

What are the ceremonies you perform in scrum?

There are 3 major ceremonies performed in Scrum: -

1. **Planning Meeting – Where the entire scrum teams along with the scrum master and product owner meets and discuss each item from the product backlog that they can work on the sprint. When the story is estimated and is well understood by the team, the story then moves into the Sprint Backlog.**
2. **Review Meeting – Where the scrum team demonstrates their work done to the stake holders**
3. **Retrospective meeting – Where the scrum teams along with the scrum master and product owner meets and retrospect the last sprint they worked on. They majorly discuss about 3 things:**
    - **What went well?**
    - **What could be done better?**
    - **Action Items**

Apart from these three ceremonies, we have one more called "Backlog grooming" meeting. In this meeting, the scrum team along with the scrum master and product owner. The product owner put forward the business requirements as per the priority and the team discussed over it, identifies the complexity, dependencies and efforts. The team may also do the story pointing at this stage.

What do you discuss in Daily stand up meeting? What do daily stand up meetings entail?

Each day, at same time and same place (in front of the task board), the team meets to give updates about their tasks and tickets resolved for the day. This meeting addresses SCRUM's three questions listed below.

– What have you completed since the last meeting?
– What do you plan to complete by the next meeting?
– Any impediments / roadblock

What is the "time Boxing" of a scrum process called?

It's called "Sprint"

What should be an ideal duration of a sprint?

It is recommended to have 2 – 4 weeks of sprint cycle.

Mention what is the difference between Sprint and Iteration in Scrum?

Iteration: It is a terminology used to define single development cycle in general agile methods. It is a common term used in the iterative and Incremental development process.

Sprint: It is used to define one development cycle or iterative step in a specialized agile method referred as Scrum. Sprint is scrum specific, and not all forms of iterations are Sprints.

What do you do in a sprint review and retrospective?

During Sprint review we walkthrough and demonstrate the feature or story implemented by the scrum team to the stake holders.
During retrospective, we try to identify in a collaborative way what went well, what could be done better and action items to have continuous improvement.

During review, suppose the product owner or stakeholder does not agree to the feature you implemented what would you do?

First thing we will not mark the story as done.

We will first confirm the actual requirement from the stakeholder and update the user story and put it into backlog. Based on the priority, we would be pulling the story in next sprint.

In case, the scrum master is not available, would you still conduct the daily stand up meeting?

Yes, we can very well go ahead and do our daily stand up meeting. Either available Scrum Coach | Alternate Scrum Master can handle the meeting

Apart from planning, review and retrospective, do you know any other ceremony in scrum?

We have the Product backlog refinement meeting (backlog grooming meeting) where the team, scrum master and product owner meets to understand the business requirements, splits it into user stories, and estimating it.

What is a retrospective?

A retrospective is a meeting to inspect and adapt the process. This Agile methodology interview question is looking for the many ways to conduct a retrospective—so be ready to explain one or two formats.

How story Board can be defined in agile?

A Story Board is a visual representation of a software project's progress. There are generally four columns 'To do', In Progress', 'Test', and 'Done'. Different coloured post, its notes are placed in each column indicating the progress of individual development items. A story board is typically used in agile development.

Explain what the ideal duration is for Sprint, and how it affects the workflow?

Sprint in Scrum usually lasts for 30 days or two weeks. The two-week sprint is preferred for various reason, first it makes easier for the team to estimate, plan and complete the work in two weeks. Secondly, it gives enough time to the product owner to change the priorities more often and allows the team to adapt quickly to the market pressures.

During Scrum meeting what all things are done?

During scrum meeting

- Team analyze how much time they got to complete task during the Sprint
- From product backlog, team takes the first item and breaks into tasks
- Team estimates how long a task will take
- If there is any time left during the sprint, they will move on to the next item on the product backlog
- Decide the features which have clarity and estimates how many to be scoped for sprint

Mention what is the objective behind holding a Sprint retrospective meeting?

The objective behind Sprint retrospective meeting is to let team members know how things went during the sprint and discuss possible ways for further improvements for future sprints.

What is the Daily Stand-Up?

One of the interview questions on Agile is sure to be about the Daily Stand-Up. The answer? Every day, preferably in the morning, the team meets for no more than 15 minutes to answer three questions:

- What did you do yesterday?
- What do you plan on doing today?
- Are there any blocks or impediments that keep you from doing your work?

This Scrum ceremony is not meant to be a status meeting for stakeholders, but a way to energize the team and get them to set focus for the day.

Describe what happens in the Sprint planning meeting?

In Sprint planning, the Product Owner presents the goal of the sprint and discusses the high priority product backlog items. The Delivery team then chooses the amount of work for the next sprint.

What is the role of the Scrum Master?

Here's how to handle a Scrum Master interview question like this: The Scrum Master serves the team and shields them from any distractions that could prevent them from completing a sprint goal. They also remove blocks, teach the team to become self-organized and serve as a coach who teaches Agile and Scrum values and principles.

Is there a difference between Agile and Scrum?

Yes! Agile is the broader umbrella which Scrum falls under. Agile has four main values and twelve principles. Scrum has its own set of values and principles and provides a lightweight "framework" to help teams become Agile.

What are the problems of sprint retrospective?

Problems that sometimes occur are:
- the retrospective is planned on the last day of the sprint. Sometimes we are still busy finishing our development activities. Worst case the retrospective got cancelled, or when it does take place the team members aren't really participating actively.

- I use a timebox of 2 hours in a 2-week sprint. This should be enough for a good retrospective. However, despite of a focused/lean schedule, the timebox is passed before you realize it. We've discussed most of the issues that happened during the sprint, but we didn't get to the root cause. Therefore, problems keep existing.

I have done some research on how to improve the retrospective. So I think I know how to fix the problems I've described.

Still I am curious what problems you encounter with the retrospective. Just to be sure that my problems are not unique .

What if sprint fails?

A "failed" Sprint is one where the Goal has not been achieved. The most important consideration is how failure is controlled.

If the Goal becomes obsolete, and the Sprint is cancelled, then a "fail early fail fast" model has been applied. Control of the situation, including the minimization of wasted effort, can thereby be evidenced.

On the other hand, if the Sprint Goal is not achieved and this problem only becomes apparent at the end of the Sprint timebox, then a greater level of waste has been incurred, and there is less evidence of control.

Which ceremony the Inspect and Adapt works?

Daily scrum is the ceremony where Inspect and Adapt works.

Which Artifacts follows more transparency to the teams?

Product backlog and Sprint backlog Artifacts follows more transparency to the teams.

1.9 Release Planning

What is Release Planning? What are its inputs and outputs?

The release planning is a tentative plan for the whole release that covers several sprints.

A long-term plan has to be derived for the following questions

- What could be delivered before the end of the year | each quarter?
- How many people do we need to deliver this project?
- When are you going to be done with critical features?
- When are you going to be done with minimum viable features?

The **Input** for release plans are –

- Release strategy, Priority
- Estimated Product backlog that gives the total size of the release
- Velocity of the team which represents the productivity
- Assumptions, constraints and risks

The **output** for release plans are –

- Release scope/date
- Prioritized backlog
- Ordered estimated in size and forecasted in sprint, Release burnt chart

How would you estimate the total size of the backlog?

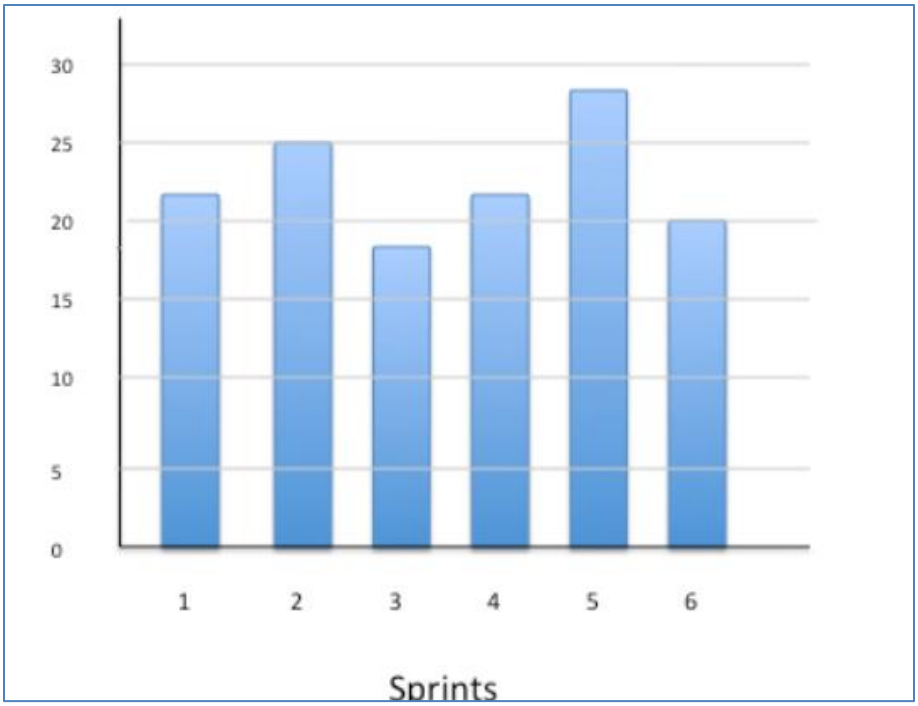
The Total size of the backlog depends on

- Development team estimates items in the backlog. Since all items won't be clear. the team makes their best guess.
- Any estimation technique like planning poker or affinity estimation could be used
- Team can iterate over estimates until they feel that the overall estimate is roughly accurate

What is Velocity?

A velocity can be defined as

- A long-term measure that indicates how much work is "done" per sprint
- Velocity is number of point completed per sprint
- Partially finished stories don't count
- Velocity varies in every sprint



How do we know the team's velocity?

- If the team is in place for some time, look at the history of the team's velocity
- If the team is new, run couple of sprints to establish the initial velocity
- Use the average velocity over several sprints to predict the completion date

How do you plan your release?

Planning a release based on the following calculation

Let's say the release backlog of size 100 points.

Step 1 - First run 4 sprints

The team ran 4 sprints and had velocities of 8, 11, 9, 10

Average velocity for 4 sprints: 38/4 = 9.5

Best Velocity:11 | Worst Velocity: 8

Step 2 – Run remaining sprints

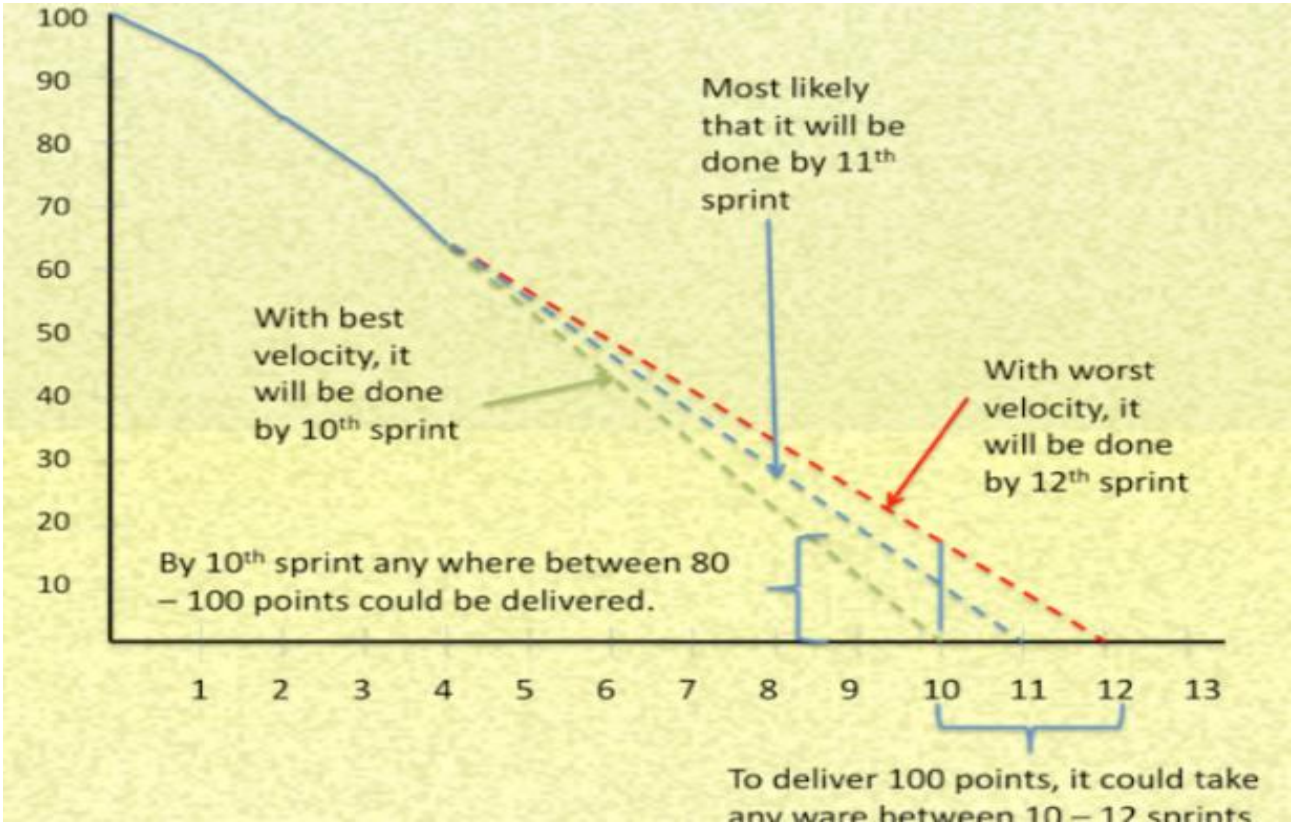
Remaining points after 4 sprints: 62

Required sprints to complete 62 points (Round up value)

@ Average velocity: 62/9.5 = 7 sprints

@Best Velocity: 62/11 = 6 sprints

@ Worst Velocity: 62/8 = 8 sprints


100
90
80
70
60
50
40
30
20
10
Most likely that it will be done by 11th sprint
With best velocity, it will be done by 10th sprint
With worst velocity, it will be done by 12th sprint
By 10th sprint any where between 80 – 100 points could be delivered.
1 2 3 4 5 6 7 8 9 10 11 12 13
To deliver 100 points, it could take any ware between 10 – 12 sprints

----------Discussions----------

How do you measure the work done in a Sprint?

It's measured by Velocity.

What is Velocity?

Velocity is a metric that is calculated by addition of all efforts estimates associated with user stories completed in one iteration. It predicts how much work Agile can complete in a sprint and how much time will it require to complete a project.

How do you track your progress in a sprint?

The progress is tracked by a "burn-down chart". A burn-down chart displays the amount of work a team has burned through—such as hours during the Sprint.

How do you create the burn-down chart?

Burn down chart is a graph which shows the estimated v/s actual effort of the scrum tasks.
It is a tracking mechanism by which for a particular sprint; day to day tasks are tracked to check whether the stories are progressing towards the completion of the committed story points or not. Here we should remember that the efforts are measured in terms of user stories and not hours.

What type of metrics or reports have you used?

Sprint, release burn-down, and burn-up charts are standard reports. Most companies also want to understand how many stories were committed versus completed per sprint and the number of defects identified post-release to production.

What is a Release candidate?

A Release candidate is a build or version of software that can be released to production. Further, testing such as UAT may be performed on this version of the product.

What project management tools are used in agile?

Agile has a new breed of PM tools including Rally Software, Version One and Xplanner, Easybacklog, Icescrum, Agilefant, Agilo. These tools bear no resemblance to the waterfall PM tools like MS-Project or Clarity

How the velocity of sprint is measured?

If capacity is measured as a percentage of 40 hours weeks then completed
= story points * team capacity
If capacity is measured in man hours then completed story points / team capacity.

What does a scrum burn down chart comprise?

A scrum burn-down chart should consist of

- **X-axis that displays working days**
- **Y-axis that displays remaining effort**
- **Ideal effort as guideline**
- **Real progress of effort**

Does maximum velocity mean maximum productivity?

No, in an attempt to maximize velocity, a team may in fact achieve the opposite. If asked to maximize velocity, a team may skimp on unit or acceptance testing, reduce customer collaboration, skip fixing bugs, minimize re-factoring. While potentially offering short-term improvement (if you can call it that), there will be a negative long-term impact. The goal is not to maximize velocity instead the optimal velocity over time, which takes into account many factors including quality of the end product.

How to measure velocity if our iteration lengths change?

You can't measure it easily. Velocity's value comes from its inherent consistency. A fixed iteration length helps drive the reliable rhythm of a project. Without this rhythm, you are constantly revising, re-estimating, and reconciling, and the ability to predict out in the future is minimized due to inconsistent results.

If, on the other hand, almost everyone is going to be out a week for the holidays or a couple days for company-wide meetings then by all means adapt iteration dates or velocity accordingly. Like most agile practices, these are guidelines, not strict rules.

Explain what does the burn down charts show?

Burn down charts is used to track sprint status, they act as an early warning indicator; they can be useful in highlighting the "lack of progress". Also, they will highlight the area where they see redundancy.

Explain what velocity in scrum is and how it is measured?

Velocity in a scrum is a measurement of how much the team gets work done in an iterations or sprint. It is measured by

- **V= Number of total story points / One iteration**

Lesson 2 Agile Coach Discussions

Tell me about your introduction before proceed with?

- Worked as an **Agile coach** in MNC with **16 years of IT experience** out of which **8 Years have been dedicated to Agile**, Scrum & SAFe practices at enterprise level
- My Agile Certifications are - ICAgile Certified Professional in Agile Coaching (**ICP-ACC**) & SAFe Program Consultant (**SPC 4**), Certified Scrum Professional (**CSP**)
- I am **expertise** in Agile transformations, Coaching, training and mentoring & Lead a proposal with a big win to generate more business value, trained more than **1000+ Professionals** on Agile Principles & Practices, XP, Scrum, Kanban, Scaled Agile framework by making the organization as an Agile Organization, build agile culture across **3** continents as an agile coach for **distributed team**
- **Review** how teams are conducting Sprint Planning, Scrum Daily calls, Sprint Review and Sprint Retrospective ceremonies and provide feedback., track **team velocity** for teams after each sprints and plan to Increase it each quarter using Graphs like CFD (Cumulative Frequency Diagram)
- Perform Project Management **duties** like Client Interaction, Estimations, Resource Handling and delivering on time.
- My agile **websites and books** are released by Syntel CEO & President -Rakesh Khanna & Temenos CEO - Susan Gibson. I have published my agile books in amazon such as Handy Agile, Agile A Key of Success, Scrum Alliance Professional, Agile Coaching, SAFe Q&A and FAQ'S
- Having good **experience in technology** such as Java, Big data, SharePoint & .Net projects to support the development team

What is your achievement in Agile Coaching Journey?

- Trained more than **1000+ Professionals** on Agile Principles & Practices, XP, Scrum, Kanban, Scaled Agile framework by making the organization as an Agile Organization.

- Lead a proposal with a big win to generate more business value across **3** continents as an agile coach for **distributed teams.**

Agile Coach Sample Resume

Agile Coach.docx

How Coaching starts for the project from the day 1? (or) You have been appointed as a coach in an organization. What would be your starting point?

- Evaluate the current status of the team, identify whether the project is new or existing project or transformation from waterfall to agile project.
- Develop short term plan for 1-3 months for agile coaching. Engage the coachee in the coaching process.
- Based on the coaching agreement, I will coach the team | enterprise and maintain the confidentiality of each team to grow as a high productive team
- For the existing projects, start coaching them from the current status and guide them through ceremonies, ensures that team is delivering consistent throughput.
- For new projects guide the team in Agile planning to develop the important things for the project such as Project Plan, Release Plan, Iteration Plan, Test Plan, User Stories, Product Backlog & Sprint Backlog.

Start **Sprint 0** to be conducted at the **start for every release**. Mandate to proceed further.

- Validate the core and extended **team members**
- Identify all dependent group sign off needs
- Identify the development and test **environment**
- Identify dependencies on other projects, teams & resources that may influence the release schedule
- Identify the **deliverables** and sign off needed
- Identify **number of iterations** for the release using the team velocity
- Identify the **schedule for release testing** and release iterations, as a guideline, have release iteration (Release Testing) after every three-time boxed development iterations
- Identify any **assumptions** made
- **Risk** involved in the project
- Identify the project with **release schedules**
- Create a **release plan**

Start from **Sprint 1**, guide the team to pull the user stories from the product backlog items to start the sprint in a meaningful way to complete without any dependencies along with the commitment from the development team to complete within the sprint. Start coaching them through ceremonies, ensures that team is delivering the consistent throughput. High risk |value items are to be considered for the first set of sprints.

Review how teams are conducting Sprint Planning, Scrum Daily calls, Sprint Review and Sprint Retrospective ceremonies and provide feedback. Track Team Velocity for Teams after each sprint and guide them to Increase it each quarter using Graphs like CFD (Cumulative Frequency Diagram).

Adopt Agile Life Cycle Management (ALM) tools like Rally, JIRA &Agile AGM to track Epics, Features and User Stories and to capture metrics like Velocity and Burn down Chart, KPI, delivery-commitment index, resource-resource burndown, quality-bugs classification and goals against it

Provide feedback to the team with respective stage to grow further and produce high performance team in the organization. Likewise execute the remaining sprint | iterations as per plan.

What is Coaching? How do you coach the team?

Coaching is partnering with clients in a thought-provoking and creative process that inspires them to maximize their personal and Professional Potential. Coaching helps them to learn rather than teaching them. The art of Agile Coaching is understanding the situation, the values underlying Agile software development, and how the two can combine. Do the experiments to hit on the right approach, work with the teams, come up with great solutions and learn from every team we work with gives the great experience in coaching.

"The key motivations of Coaching presence are to build rapport with the coachee, engage coachee in the coaching process & Keep the coachee in 'towards' state. Apply different techniques they need training, mentoring & coaching based on their roles to help the team. Create an adaptive approach for the current status of the team based on their coaching needs. Coach the Product Owner, Scrum Master & Development Team on key agile practices & day to day activities involved in their project. "

During project execution, Coaching through ceremonies: -

- **Coach the team on how to estimate work using Planning Poker. Prepare required outcome for the planning session.**
- **Sprint planning: Plan team sprints, with the outcome being the sprint backlog, sprint goals, and a team commitment.**
- **Release Planning: Facilitate a conversation about the stories with the scrum team.**
- **Daily Scrum: Coach the teams to self-organize around the work in the current sprint asking the three stand-up questions.**
- **Backlog refinement: Facilitate regular meetings with the scrum team to discuss the stories for next sprint.**

- **Sprint Review Prep:** Prepare for the Sprint demo with team, work through the story DoD, how to demo the stories, etc.
- **Sprint Retrospective:** Conduct sprint retrospectives. Use decided-on team metrics. Make items actionable for continuous self-improvement (Kaizen).
- **Collaboration and Coordination:** Work closely with the PO and Architects as needed, helping write stories and to prioritize the backlog.
- **Coaching and Mentoring:** Help coach the team in the proper application of agile and mentor new team members who have not worked in an agile model in the past.
- Help define and promote the company's Agile vision, methodology, practices and tooling.
- Worked with the Scrum Framework model while working closely with product owners, architects, and other scrum masters to deliver the overall project functionality.
- Coordinate dependencies across the various Scrum teams' work tasks.

How will you coach the Product Owner? How will you support the Product owner to handle the product backlog?

Educate the Product Owner (**PO**) how to maintain and groom the deep product backlog and guide them how product backlog items are prioritized Using MoSCoW technique and how many iterations & respective sprints to be executed for the project. Ensures that product backlog item has more focus on business-value-driven. Ensures that effective Product Owner should be Committed, Responsible, Authorized, Collaborative & Knowledgeable (**CRACK**).

What are the key tasks in coaching the Product Owner? How will you support PO?

The key tasks in coaching the Product Owner has to understand their priorities: -

- Be the vision keeper – in sync with sponsor
- Move from schedule-driven to business-value-driven planning
- Cultivate Business value driven thinking in all interactions
- Maintain a DEEP product backlog
- Match demand with capability
- Learn to trust the team
- Avoid micro-management
- Hold the team for their commitments

How will you coach the Scrum Master?

Educate the Scrum Master (SM) to take-care of the agile process. Guide them to facilitate the scrum ceremonies and help the team to act as a bulldozer for impediments. Be a Servant leader, to lead by serving others and progress tracker towards the goal and ensures the guardian for quality of work to be performed.

What are the key tasks in coaching the Scrum Master? How will you support SM?

The key tasks in Coaching Scrum Master has to understand their priorities: -

- **Facilitator, not decision maker, Keep sustainable pace of development**
- **Grow Self-managing teams -> Planning work, pulling work, tracking work & getting work done**

How will you coach the Development Team?

Educate the Development Team (DT) with key agile practices and frameworks. Motivate the team to grow cross-functional & self-organizing. Guide them to work together as a Collective ownership to make them in Continuous improvement. Walk through ceremonies and provide feedback to grow at higher productive team to generate high productive and quality output.

Team having Too Strong Product owner, Scrum Master is weak enough and Development Team is also not performing up to the expectation. How will you handle the situation as an agile coach?

In this situation, understand the product owner expectations, coach the Scrum master and development team with the agile mind-set, project activities and motivate them continuously for the gap fulfilment and establish smooth flow across the team to meet the product owner expectations. If unforeseen(provokes, Prolongs) situation occurs again, by replacing with the efficient person.

How to ensure team is currently doing progress correctly?

To verify whether how the development team is doing progress based on: -

- **By sharing the metrics to reflect current state (NOW) & productivity of the team**
- **How the team meetings & daily Stand-ups are progressing as per the expectation of client**
- **How the team is adapting flow, process & technical approaches**
- **How the team is taking situational decisions**
- **How the team is delivering frequently as per DOD**

Technical Lead is facing the technical issue, how would you resolve it?

Understand the technical issue faced by the technical lead and help them in: -

- **Team Brain storming**
- **Sit with the team, explore the ideas and resolve it**

How will you coach during sprint planning?

During the sprint planning: -

- **Verify the planning objective**
- **What can be delivered in the increment resulting from the upcoming sprint?**
- **How much work needed to deliver the increment can be achieved?**
- **Plan team sprints, with the outcome being the sprint backlog, sprint goals, and a team commitment.**
- **Verify the stories selected in the current sprint from the prioritized product backlog, selected stories are estimated with agreed acceptance criteria and how the team identifies and estimates task using planning poker method**
- **Share the current velocity, and guide them to grow in a consistent velocity or higher. The accepted stories / sprint only considered for the velocity consideration**

Note: Before Sprint Planning, feasibility study has to be conducted – the team has enough capacity to proceed with, whether the product backlog is groomed properly, met the business requirements with respect to current market conditions, current product developed status & technology to be considered. Without proper planning, execution of the sprint is not effective.

How will you guide the team to split the user Stories?

A Planning poker is used for user stories estimation. For splitting or decomposing a user story ensure every story address all the architecture layers such as Presentation layer, Validation layer, Business layer, Database layer. The best way to slice vertically through the layers.

There are some user stories pending at the end of the sprint. How will you handle?

At the end of the sprint, unexpectedly some user stories are pending it may be due to the requirement not clear with technical feasibility, insufficient time to execute the stories based on the resource skills & availability. Understand the ground reality the incomplete tasks are moved to product backlog, executed in upcoming sprint planning based on the priority. The unforeseen situation occurs some time, but this can be overcome with efficient planning.

Ref: https://www.mountaingoatsoftware.com/blog/handling-work-left-at-the-end-of-a-sprint

Team is doing at the end of the day during the sprint, not working properly during the sprint? How would you handle this situation?

Stories have to be delivered in specified intervals, not at the end of the sprint. Daily Scrum ensures that stories are executed on day to day basis without any impediments. In case of any impediment there, resolve asap. So, motivate the team consistently to get the things done as per schedule and don't encourage them to do in a last day, last minute. The factors to be considered during the sprint such as Velocity, requirements stability and capacity maturity, efficiently to work with the product.

Getting buy-in from Leadership team and Senior Executive? How about Controversies?

The Main Reasons for buy-in leadership team and senior executives in agile arena are: -

- **Managing optimized backlogs of work with constantly changing business priorities**
- **Sincerity in engagement and collaboration between various departments including business customers**
- **Effective planning and estimating for larger initiatives**
- **Faster time to market, no wait cycles to get work completed**
- **Eliminated Errors, no rework and no miscommunication on expectations**

In Controversy, some of the organization facing difficulties in buy-in from leadership team and senior executives in agile arena are: -

- **Managing large backlogs of work with constantly changing business priorities**
- **Lack of engagement and collaboration between various departments including business customers**
- **Ineffective planning and estimating for larger initiatives**
- **Slow time to market, long wait cycles to get work completed**
- **Errors, rework and miscommunication on expectations**

How would you measure the team maturity? How did you track Agile Maturity level of various teams?

To measure the team's maturity: -

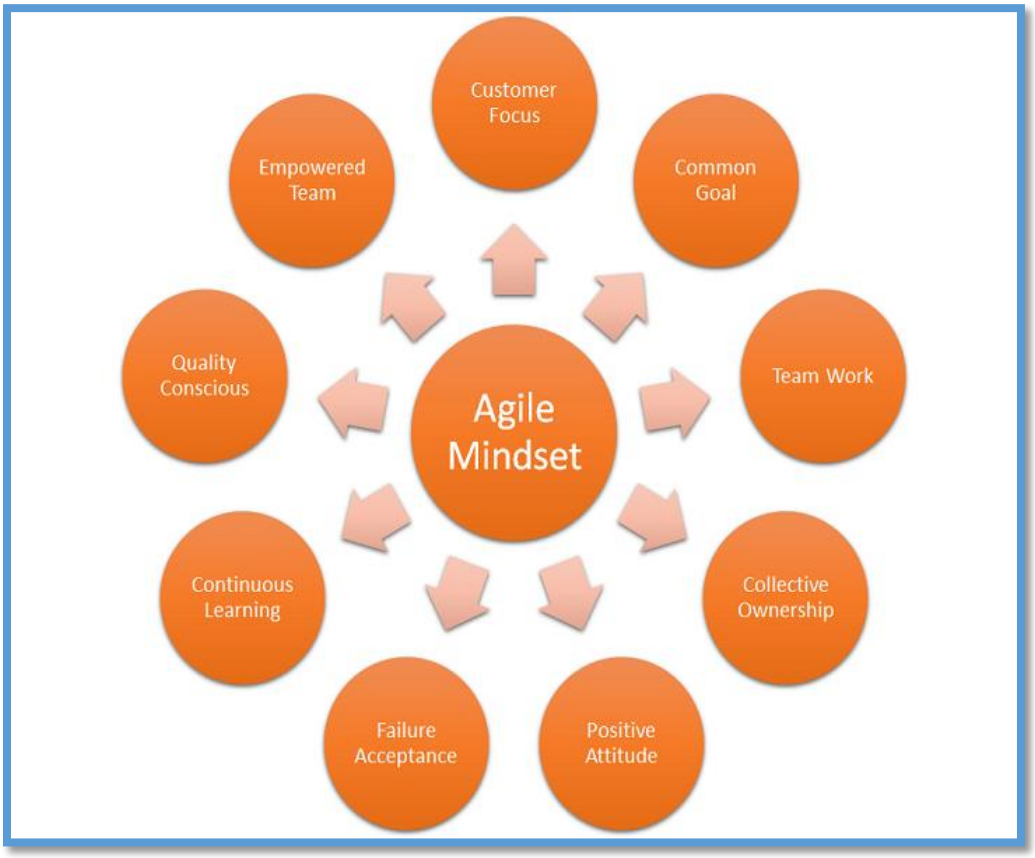
- **How the team is adapting flow, process & technical approaches**
- **How the team is matured enough in executing ceremonies. For example, in sprint planning, the matured teams will conduct Pre-Sprint Planning, user stories estimation has to be completed**
- **How the team is adapting the agile principles, practices and values (in the range of 0-5)**
- **How the team is taking situational decisions**
- **How the team is delivering frequently as per DOD**
- **Whether The team is delivering in consistent team velocity and it has been increased in each sprint or not**

- Whether the team is deliver in consistent throughput or not

What Agile ALM tool set did you recommend and implement?

Adopt Tools like Rally, JIRA and Agile AGM to track Epics, Features and User Stories and to capture metrics like Velocity and Burn Down Chart, KPI, delivery-commitment index, resource-resource burndown, quality-bugs classification and goals against it. Note: Planning Poker is an activity the development team uses to estimate the relative size of the product backlog items.

What aspects of the mindset do you have to change and how do you as an agile coach go about doing it? Also, give the example of what you have done in this regard?



- As an Agile Coach with clear mindset, I have to Create an adaptive approach for the current status of the team based on their coaching needs. Apply different techniques they need training, mentoring & coaching based on their roles to help the team. Coach the Product Owner, Scrum Master & Development Team on key agile practices & day to day activities involved in their project based on the agile mindset.
- Exhibit "RI" stage like Doing Agile & Being Agile. **Before sharing to the team, internally feel how we can adapt successfully and then share to the team.** (Shu - Learn exactly what they taught by master, Ha - Experimental Stage RI - Doing Agile and Being Agile). **"You have to believe in yourself and you**

should be the change agent, trust the change begins from you. So, you need to give has to give the confidence to the team that you will be there to assist and support them"

- To change the team mindset and **adapt the key agile practices** to become a high-performance team.
- To **cultivate the agile mindset** with a team, the team has to focus on a common goal, collective ownership, failure acceptance, positive attitude and continuous learning, quality conscious and team empowerment.
- In my experience with one of the client, where team has not efficient to do their daily routines and many slippages happens after I have given mentoring program by adapting agile principles, practices and project based coaching, they have grown and achieved a high-performance level over a period of time.
- In my career, I have trained 1000+ peoples in Agile practices for embracing the agile mindset.

What parts of being an Agile Coach you struggle the most with (from a personal & professional point of view) and how do you handle them?

As an experienced agile coach, I am feeling very happy to help the team without any struggle.

I handled from a **personal point of view** by

- Holding people accountable
- Maintaining neutrality and confidentiality
- Challenging the status quo
- Personal bias
- Difficult to stay out of politics

I handled from a **professional point of view** by

- To handle the team, move into the agile
- Based on Agile Mind-set and Culture
- Team Maturity

What are the most useful tools in your coach toolbox (give an example of using them successfully in the past)?

The most useful tools in my experience is Rally, JIRA & Agile Manager used to track Epics, Features and User Stories and to capture metrics like Velocity and Burn Down Chart, KPI, delivery-commitment index, resource-resource burndown, quality-bugs classification and goals against it.

From the agile coach goals, which one do you find the most challenging and why? Also, which one you are more comfortable with?

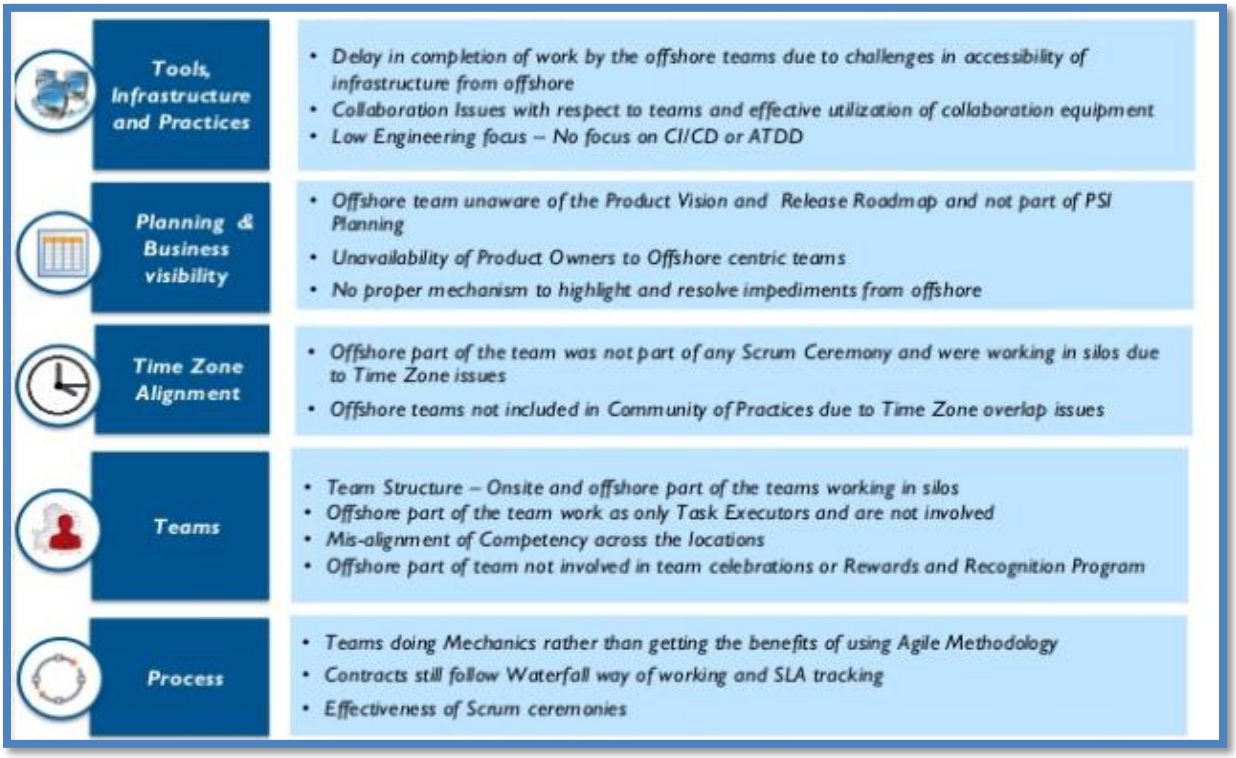
From the agile coach job goals, the most **challenging** goal are:

- Onshore & Offshore collaboration sync is difficult while handling multiple vendors in multiple time zones.

- Enhancing the Lean-Agile mindset & behaviours, Grow the team experience in Agile related practices & tools.
- Increase cohesion within a distributed & multi-cultural organization contribute and rollout our ever-evolving agility operational model with specific skill | experience.
- With my professional experience, I handle efficiently without any challenges.

From the agile coach goal, the most **comfortable** are:

- Handling either Onshore or Offshore with multiple vendors in comfortable in execution. Not in a mix.
- Matured Team members adapting the agile practices and embracing with the lean agile mindset and delivering with high value
- Matured Team accustom with distributed, collaborate & multi-cultural environment to execute ceremonies



(Scenario) An Admin, HR, Finance & Quality has not yet heard about the word Agile and they need your help to reduce the lead time in the daily hiring routine process. How would you go about helping the team?

In my experience, I have created the Agile awareness and provide the Agile key practices and tools to handle the job efficiently by reducing surface delays this makes a good success for other departments associated with IT.

(Scenario) Your manager holds a meeting of 3h every week that you strongly believe adds no value to the team (10 people, all direct-reports) How would you go about addressing the potential issue?

I request the manager to follow the agenda, not to deviate from the focus and execute the ceremonies as per plan

In my experience to address the potential issue, First I will ensure the team to follow first adapt GROW model that helps in better achievements such as G- Goal, R-Reading, O-Options, W-Will (What | When)

Second updating the team growth status by

- **Facilitate Learning goals – Mastery and Competence**
- **Metrics reflect current state (NOW) – Measuring potential or productivity is a lower priority**
- **Focus on positive emotion – Performance and Enjoyment – Decrease negative emotion**

(Scenario) A Scrum Master is running out of time asking agile coach to skip some meeting by today. As an agile how would you handle?

As an agile coach, I advise Scrum Master not to skip any meeting, plan well accordingly and try to adjust without any slippages. In worst-case you can send the prior communication for postponement schedule or send the alternate available scrum master to handle the meeting.

(Scenario) A person travelling from ground floor to third floor in the lift in 30 secs. During this time, you have to convey the benefit of agile to that person? How would you convey?

The benefits of applying | adopting agile in the project are: -

- Reduce turnaround time for features

- Predictability of market releases with respect to content and timing

- Ability to handle complex product enhancements

Comparison Waterfall & Agile?

In Waterfall

Plan driven process, predictive, fixed scope, adjust schedule to preserve scope

Long development cycle, linear, organize work into major phases, delivers value at project completion

In Agile

Agile value driven Process, Adaptive, Fixed Schedule, Adjustable scope to preserve schedule

Short development cycle 2-4 weeks, cyclic, organizes work into small deliverables, delivers values incrementally over time

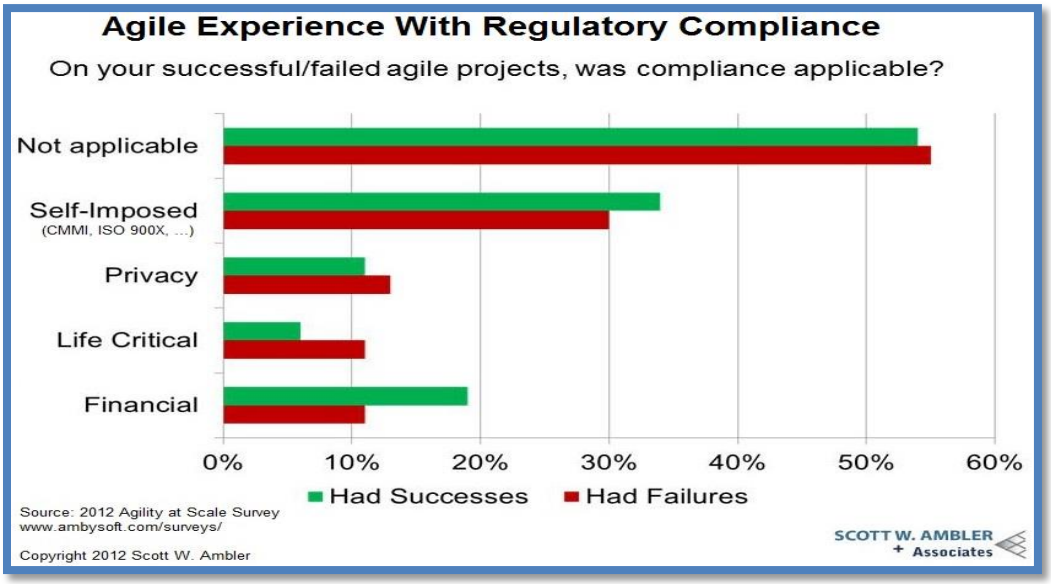
Do you think Agile can be applied to all projects? What benefits you got by applying Agile in your Project?

Not necessary to apply agile for all project. Suppose If my requirements, time, cost etc are all fixed, there is no reason I should follow the agile Life Cycle Methodology. Agile is not a panacea of all the business problems. The problem in the industry today is that organizations think with an Agile approach they would be able to solve all their problems. But that isn't the case. Agile and Digitization are just enablers to solve the business problems. If following a waterfall or a mini waterfall cycle, I can periodically deliver minimum business requirements, I need not go with a full blown agile implementation.

The benefits of applying | adopting agile in the project are: -

- Reduce turnaround time for features
- Predictability of market releases with respect to content and timing
- Ability to handle complex product enhancements

How can you apply Scrum/Kanban in projects, organizations which are highly regulatory in nature?



Yes. It's possible for a team to take an agile approach in a regulatory environment. To addresses regulatory compliance issues via several key strategies: -

- **Adopt a hybrid process**. A hybrid framework that adopts strategies from a range of sources including Scrum, XP, Agile Modelling, Kanban, Unified Process
- **Adopt a full delivery lifecycle**. Most regulations address the full delivery lifecycle, not just construction
- **Focus on solutions, not just software**. Disciplined agile teams produce consumable solutions, not just "shippable software"
- **Take a goal-driven approach**. Recognizing that solution delivery teams find themselves in unique situations
- **Adopt an explicit governance strategy**. DAD has agile governance strategies built right in, including explicit light-weight milestones, metrics, named phases, and many other aspects of governance expected by many regulations.
- **Be enterprise aware**. DAD promotes the concept of enterprise awareness, the recognition that agile teams do not work in a vacuum. This includes strategies for engaging with enterprise architects, how to deal with enhancement requests and defect reports coming in from operations, and how to work with other enterprise professionals. These can be key issues to understand when tailoring agile to be compliant within an existing organizational ecosystem – your entire process needs to comply to the regulations, not just the development portion of it.

What can be the role of the Senior Management in the SAFe organization?

In SAFe Organization, Senior Management will play in SAFe portfolio level and organization level

In Portfolio Level

- **Handle multiple Portfolios at Enterprise level**
- **Lean-Agile budgeting empowers decision makers, so Enhance Lean-Agile Budgeting with Value Stream funding, "CapEx and OpEx"**
- **Enterprise architecture guides for larger technology decisions**
- **Monitor and guide metrics support governance and improvement**
- **Make sure that planned epics are delivered with good value and on time.**

In Organization Level

- **Lead the change**
- **Know the way, Emphasize Lifelong Learning**
- **Develop People**
- **Inspire and align with mission and minimize constraints**
- **Decentralized Decision Making**
- **Unlock the intrinsic motivation of knowledge workers**

Which metrics you think are the most relevant to you?

The metrics relevant to the agile projects that I handled

- **Sprint | Iteration – Burnup | Burn down chart**
- **Release Burn down chart | Risk Burn down chart**
- **Velocity Chart**

Do you think we can always have dynamic requirements?

Not always but sometimes projects we have dynamic requirements. It may raise due to market dynamics, client expectations, technology & competition in the industry. Self-managed development team will handle all the challenges and deliver as per the expectation through self-learning and self-growth.

Contracts- What challenges you see with Agile contracts?

The challenges in Agile coaching contracts are: -

- **Maintain the confidentiality between the coachee and other employees in the organization**
- **Termination of coaching contract in earlier**
- **Maintain the coachee records confidentially and submitting to senior management**
- **Cancellation of Coaching process should be communicated well in advance to coachee**

How Scrum Master can become a good agile coach?

In Agile career, Scrum Master role is the stepping stone to start the agile career in dealing the team and working with ceremonies, day by day coaching experience grows in the agile career to become a good agile coach.

A scrum master can also become a good Agile Coach. First of all, scrum master has to develop the positive attitude and patience. Scrum Master have to believe in himself and he should be the change agent, trust the change begins from him. So, the Scrum Master has to give confidence to the team that he will be there to assist and support them. Some important things habits have to adapt.

- **Lead by Example**
- **Keep up the balance**
- **Set realistic pace**
- **Be cautious about your language**
- **Open to learn**
- **Accept feedback**

How is your role as a coach different from a Scrum Master?

Scrum Master role will be focus on

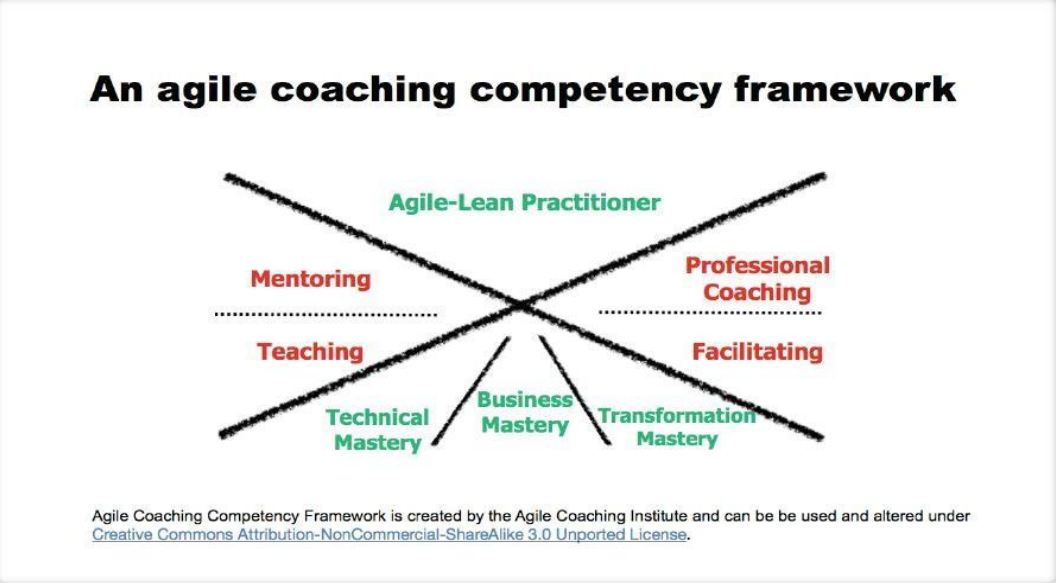
- **Team level | One or more team**
- **Expert in Scrum**
- **Shielding the team**
- **Bulldozer of Impediments**
- **Servant Leadership**
- **Good Facilitator – A neutral process holder who guides and groups through processes that help them come to solutions and make decisions**
- **A Scrum Master ensures that the team is following the Scrum process, doing the ceremonies and behaving the right way.**

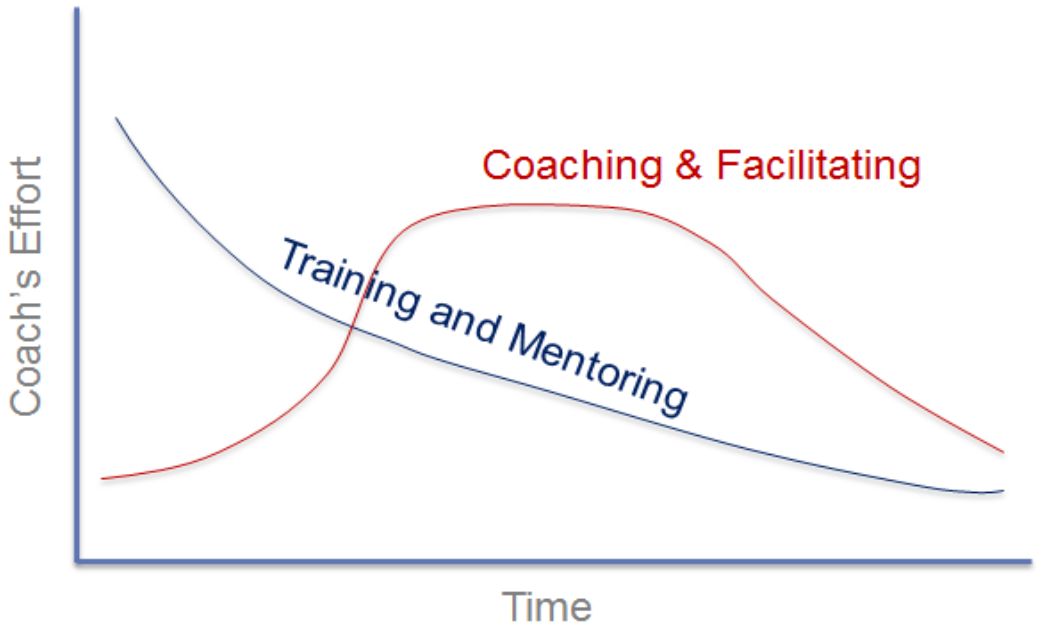
Agile coach role will be more focused at

- **Enterprise level | All the teams**
- **Expert in all Agile Framework with broader and deeper knowledge**
- **Supporting the entire organization with training, mentoring and coaching needs**
- **More like hood to support organizational changes**
- **Professional Coaching unlock the person's potential to maximize their own performance and help them to learn rather than teaching them & Partnering with clients in a creative process that inspires their personal and professional potential**
- **An Agile Coach helps to define what is to be done, how, who does it, when, why, how it fits in with the organization, change management, people management and interactions between agile teams and other parts of the organization (like Dev Ops, Hosting, Build teams, Education, UX/UI, etc.).**

What is the role of an agile coach?

- As an Agile coach, your goal is to develop productive agile teams that think for themselves rather than relying on you to lay down the path for them
- Your need to help them understand the agile from the value point of view rather than practice point of view
- You need to help them change the way of work, communicate, collaborate and understand team based value delivery. During this process, you need to help them unclearing some of their old habits, using your coaching skills, tools and techniques
- You need to understand, each team is different as they have different levels of skills, attitude and knowledge. That means your coaching strategy depends on what the teams need from you





Which ALM tools do you use? Which one you find most relevant and why?

I have used JIRA. Most of my client used the JIRA and most benefitable to the end user and cover lot of features with low cost.

What factors would you consider before you recommend a ALM tool to your organization or customer?

Any Application Lifecycle Management tool that claims to be a complete solution should consist of at least the following modules: -

- **Requirements Management**
- **Software Development**
- **Collaborative Project Management**
- **Quality Assurance & Test Management**
- **Release Management**
- **Document Management**
- **IT Operations (DevOps)**

You want your chosen solution's architecture to be flexible, allowing you to customize artefacts and workflows, reports or views (dashboard), and it should also support all your internal processes as well as any foreseeable development of these processes.

In a nutshell, the following general capabilities and features are considered the most important, when it comes to Application Lifecycle Management tools: -

- **Agile capabilities**
- **Support for various work items**
- **Gapless traceability from requirements to release**
- **Integration points with other tools**
- **Consulting & training services, support**
- **Security and reliability**
- **Available hosting & license types**

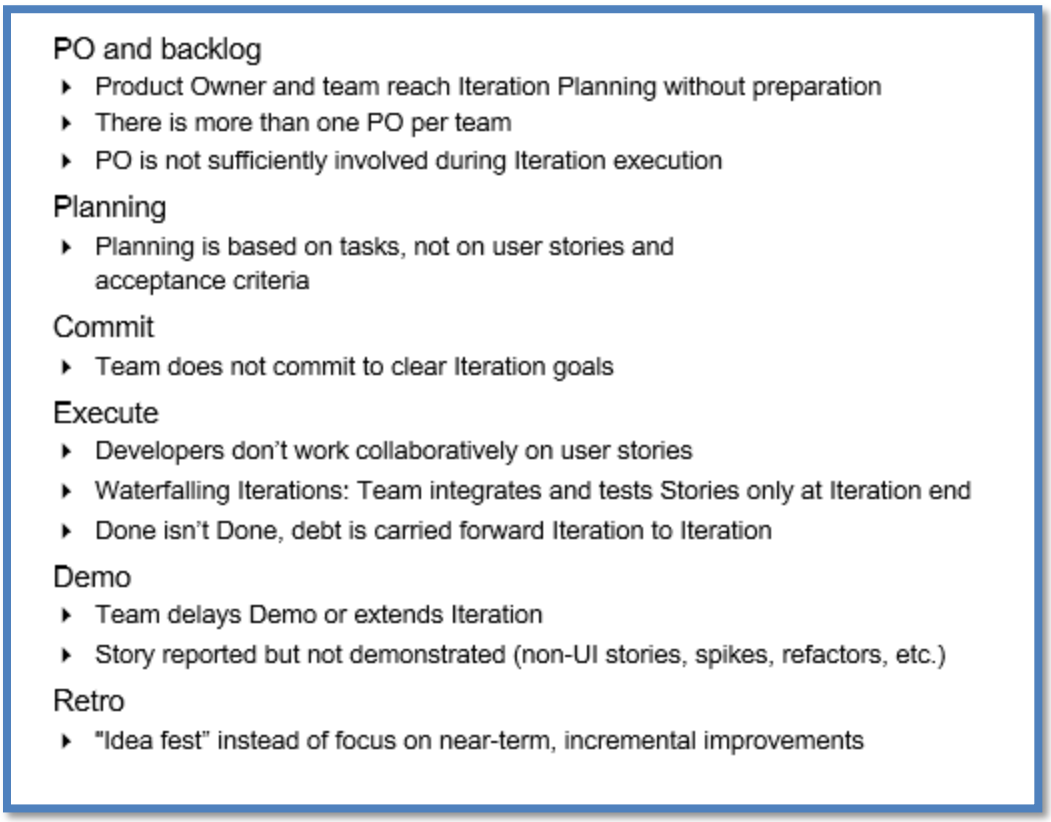
What anti patterns did you notice with your coaches'? What did you do to handle that?

An antipattern is a pattern that you think will improve things, but it doesn't.

The following is a list of antipatterns that I have observed.

- **Backlog**
- **Planning**
- **Daily Stand Ups**
- **No Show Case**
- **Review | Retrospective**
- **Command and Control**
- **Big Bang Improvement**
- **Agile Education**
- **Quality & Definition of Done**
- **Lack of Long-term thinking**
- **Lack of communication**
- **Not making it in a Safe Environment**

Reference: - https://dzone.com/articles/agile-antipatterns



As a coach, you must have trained lot of people as well. What is the difference between a trainer and a coach?

I have trained more than 1000+ Professionals on Agile Principles & Practices, XP, Scrum, Kanban, Scaled Agile framework by making the organization as an Agile Organization.

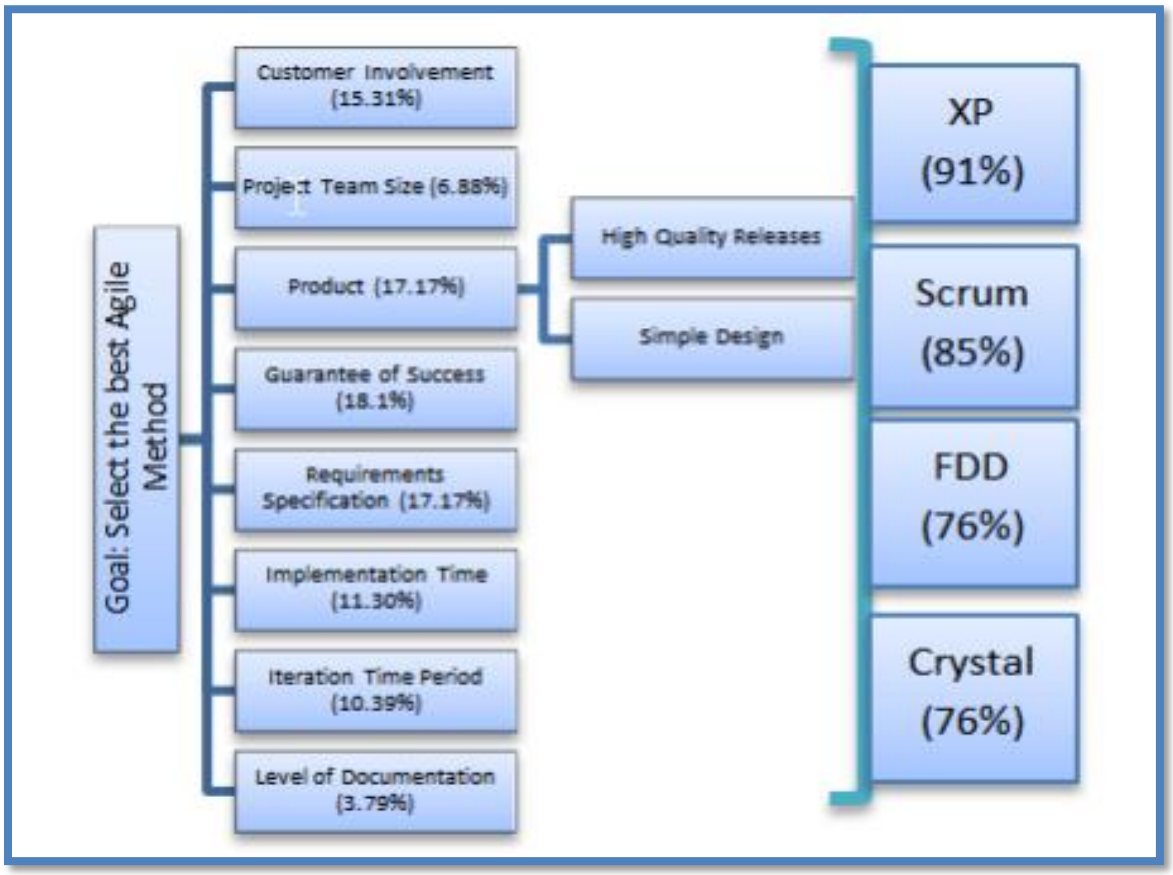
In Training

- **The trainer provides specialized knowledge**
- **Trainer follows a standard agenda and curriculum**
- **The standard agenda is driven by the trainer**
- **Trainer provides similar experience for each trainee**

In Coaching

- **Providing live example of the depth and usefulness of agile values according to the project scenarios**
- **Being agile more than doing agile, makes everyone as agile**
- **Exhibit 'Ri' stage of Agile**
- **Coach provides similar experience for each coachee and make the organization as an agile organization**
-

How do you decide which agile framework would be applicable in your context?



Before Agile framework assessment with the client first conduct the cost benefit analysis.

- What is the business outcome for the applicable agile framework?

- What we have today and what we are going to become tomorrow?

- What is the gap against Engagement Status with Industry Practices?

- Conduct the maturity level of the organization and project teams over people maturity, technology practices & agile tools, Quality Conscious

- Based on the key factors applicable to the project based on that decide the applicable framework

What are the other domains of agility that you have worked with?

I have worked with business, process and technical agility, work concert to create an agile organization. Reference: http://theagiledirector.com/article/2016/11/24/domains-of-agility/

You are talking to the CEO of a company with around fifty thousand employees. CEO has mandated to transform the organization to SAFe Agile. As a coach what do you do?

- **First recognize the current structure so that we can build ART, we need to identify ARTs and the respective sizes based on the type of business they are performing.**

- Secondly, we should not change the big bang, instead focus on bringing in evolutionary changes
- Thirdly pick an ART and train three to five agile teams to start with. Work with this way for two to three months and look for the better results next train the other teams

What are the day today activities of Agile Coach?

Agile Coach Role & Responsibility

- **Facilitator**: Facilitate the team with the knowledge so that team can start the project.
- **Trainer**: Provide training to the team on the agile process; training will continue all the time during the project execution and continuous improvement on velocity, quality, processes etc.
- Make the **winning strategy** according as per the ground conditions
- Help in preparing the overall **planning** of the project that means he will work as a consultant. He will provide various ideas, suggestions, strategies.
- Make sure that team is following **agile processes** in each sprint at user story level as per the Definition of Done (DoD); However, this is the responsibility of the Process Check Master but if project does not have a role of process check master, this activity should be handled by the agile coach.
- Help team to **answer all the questions** on the agile process during the project execution; that means agile coach need to be on the ground so that he can answer the questions immediately.
- Identify **project risks** and raise them proactively
- **Mentor**: Focusing on people and Continuous Improvement all the time; provide team a platform for improvement not only during the retro but all the time. Create a safe environment for healthy conflict and meaningful collaboration.
- Identify **process issues** and improve them
- Help **product owner to write user stories**
- Help **team on the estimating of the user stories** and prepare them for the same
- Provide **capacity calculator** template for the team
- Provide the **common tasking codes** for the team for better tracking on technical front
- Help scrum master to **plan** meetings like: - preplanning, planning, daily scrum, Review & Retrospective

In Scrum, Sprint get delayed not as per plan. How would you handle the situation?

The reason for the sprint get delayed may be due to: -

- Team overcommits – how do you roll user stories (and other product backlog items) into the next sprint?
- Team under commits – should you add new user stories mid-sprint?

- External impediments – how should these be reflected in the burndown chart and velocity?
- Product Owner changes – should you allow them to remove, add or significantly modify the sprint's user stories?

To handle this situation, first make the environment ready to execute the sprint, unless and until the sprint cannot be executed. Vendor organization will focus on billing purpose to generate more sprint without proper planning. In multi-vendor project, without proper plan, lot of money gets exhausted. So, plan carefully, before executing the sprint.

Ref:https://www.axisagile.com.au/blog/planning-and-metrics/sprint-issues-when-sprints-turn-into-crawls/

What are key success of your agile team?

The Key success of my agile team is: -

- Cross Functional Teams
- Empowered Team Members
- Single Voice of Business
- Shared Accountability
- Servant Leadership
- Continuous flow of value
- Value over activity
- Attention to Technical Excellence
- Rapid Risk Reduction
- Early feedback adaptation
- Total Openness and Transparency
- Trust

XP Practices

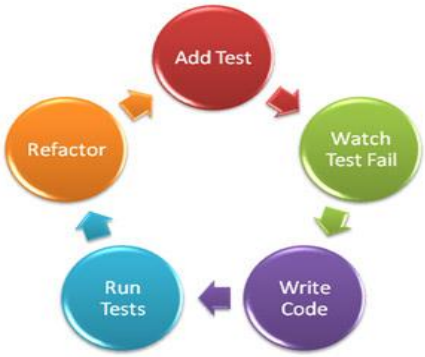
What is TDD & BDD?

Test Driven Development (TDD)

TDD is a software development technique that involves writing automated test cases prior to writing functional pieces of the code. This is popular in agile methodologies as it drives delivering a shippable product at the end of a sprint. This process can be divided into multiple steps:

1. A developer, based on requirement documents, writes an automated test case.
2. The development team runs these automated test scripts against what is currently developed and the tests fail, as they should since none of the features have been implemented yet.
3. development team functional code to ensure the automated test script gives them a green light.
4. The development team can then refactor and organize the code to produce a tested deliverable at the end of the sprint.

Test cases are mostly written in programming languages such as Java, Ruby, etc. and can be written using test automation tools such as Selenium, Watir, Windmill, etc. Since test scripts are written in programming languages, it is hard for a business analyst or test owner to verify the test scripts.



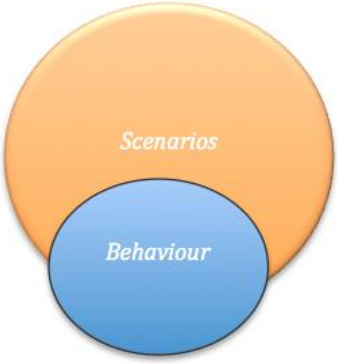
Behavior Driven Development (BDD)

BDD is a software development technique that defines the user behavior prior to writing test automation scripts or the functional pieces of code. Used in an agile sprint, this method ensures that a shippable product is generated at the end of a sprint. This involves:

1. **Behavior of the user is defined by a product owner/business analyst/QA in simple English.**
2. **These are then converted to automated scripts to run against functional code.**
3. **The development team then starts writing the functional code to ensure the automated test script gives them a green light.**
4. **The development team can then refactor and organize the code to produce a tested deliverable at the end of the sprint.**

BDD can be driven by multiple tools such as Cucumber, FitNesse, PowerTools, Docker, etc. The test scripts are written in plain English in Gherkin, Wiki frameworks, etc. Since the behavior is defined in English, it gives a common ground for ALL stakeholders involved in the project. This reduces the risk of developing code that wouldn't stand up to the accepted behavior of the user.

Behaviour Driven Development



TDD vs. BDD

1. BDD is in a more readable format by every stake holder since it is in English, unlike TDD test cases written in programming languages such as Ruby, Java etc.
2. BDD explains the behavior of an application for the end user while TDD focuses on how functionality is implemented. Changes on functionality can be accommodated with less impact in BDD as opposed to TDD.
3. BDD enables all the stakeholders to be on the same page with requirements which makes acceptance easy, as opposed to TDD.

For systems that are driven by actions of the end user such as an ecommerce website or a HR system, BDD acts as a good medium to capture all the user actions. For systems that have third party API calls, cron jobs, data exports/imports, etc., TDD might be a better solution.

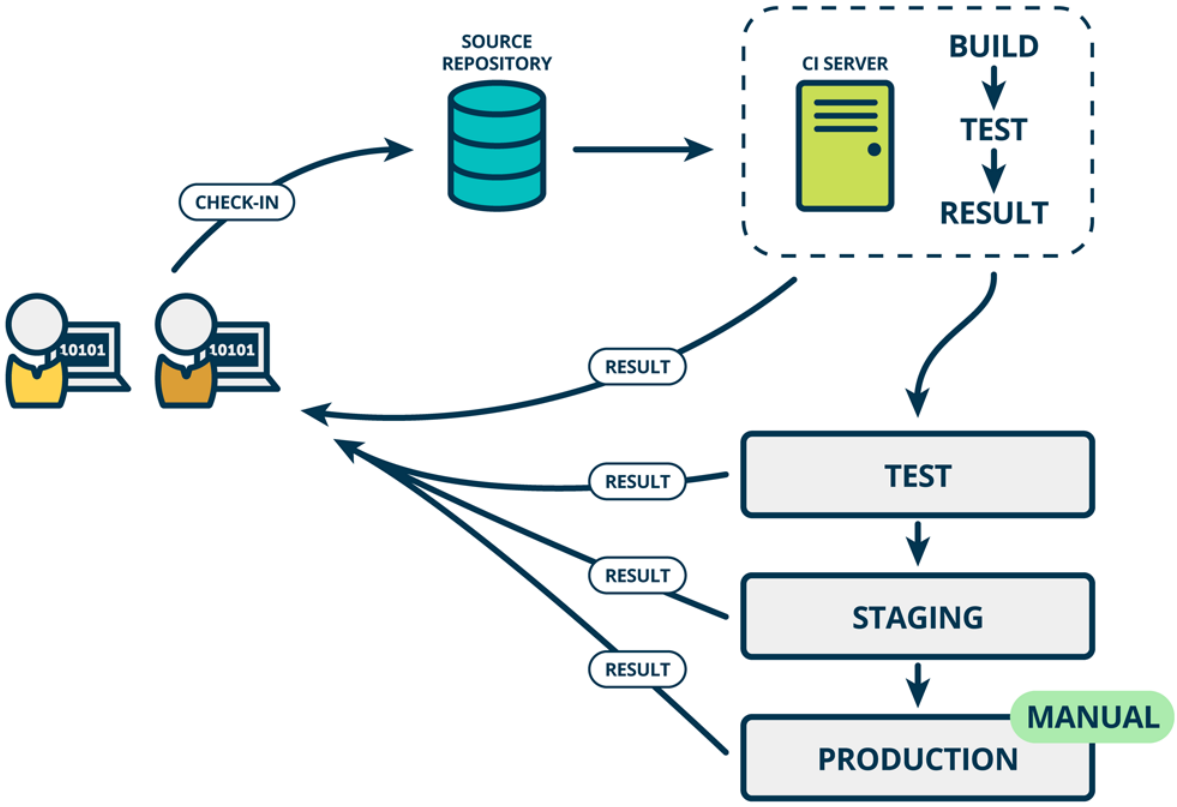
What is Continuous Integration and Continuous Development?

Continuous integration

CI is a process whereby developers and testers validate the newly written code frequently. The developers check in their code to the mainline branch at least once a day, or multiple times a day. Every check-in triggers a build, and all the unit tests are run. If the build fails, notification is sent to all the developers and the new code is rolled back automatically. Here, the tests help as a safety net to capture any bugs that were inadvertently introduced. All tests should run to confirm that the application behaves as the developers expect it to behave. A huge advantage to this process is that bugs are caught as soon as they are introduced, and the compatibility of everyone's code is tested. You always have the latest working version of the code.

Continuous delivery

CD takes CI one step further. After a build and automated unit tests run successfully, you automatically or manually deploy the application to a test, stage, or production environment. Doing this automatically pushes the envelope one step further and is called continuous deployment. CI and CD are the basic recipes for implementing successful DevOps (yes, you have heard and read that term too many times in the recent past) practices in an organization. Often CICD is the toughest portion to implement in DevOps.


SOURCE
REPOSITORY
CI SERVER
BUILD
TEST
RESULT
CHECK-IN
10101
10101
RESULT
RESULT
TEST
RESULT
STAGING
RESULT
MANUAL
PRODUCTION

Name the (39) Tools for Building your CI/CD Stack?

CI Framework

1. Jenkins

Jenkins is a "CI Framework" that wants to be at the center of your CI/CD efforts. Using a CI Framework, you can create hourly or daily builds automatically, run your unit tests and also deploy builds to your QA or production environment. Jenkins has quickly come to dominate CI/CD builds and there is no reason to believe that that will change in the near future.

Software type: Open Source

Also check out:

2. Bamboo
3. Hudson
4. TeamCity

Build automation

5. Maven

Maven is more or less a standard for building Java projects now, taking over older build automation systems like Ant. While there could be a crossover with some of the functionality of CI Frameworks like Jenkins, there are still good reasons to have a standalone build automation system.

Software type: Open Source

6. Gradle

Gradle has come a long way in a short time, and has built a growing legion of fans, who have found this to be a slicker and faster build automation service than Maven. Some of its proponents include tech luminaries like Netflix, Google and LinkedIn.

Software type: Open Source

Also check out:

7. Ant
8. Buildr
9. Gant
10. Ivy

11. Make
12. Rake

Issue and project tracking

13. JIRA

JIRA from Atlassian has become pretty dominant in issue and project tracking for agile development. While smaller operations may get away with something like Trello for an online Kanban/Scrum board, the flexible and grown-up feature set of JIRA has made it popular with agile aficionados. Importantly, its growing body of plug-ins makes it a likely fit with many CI/CD stacks — and increasingly makes JIRA the platform and focal point for many development operations.

Software type: Commercial

14. CA Agile Central (formerly Rally)

Rally was recently bought by CA Technologies, which subsequently re-branded it CA Agile Central. This speaks to the ambitions of the parent company, as they want to make it the focal point of your agile development project management. Although "Rally" was a mainstay of agile dev project management, the acquisition and re-brand may make or break the system.

Software type: Commercial

Also check out:

15. VersionOne
16. Team Foundation Server
17. HPE Agile Manager

Continuous delivery

18. Chef

Chef aims to encapsulate a series of processes, tests and automations to allow you to continuously deliver finished applications to market. It reinforces best practices, DevOps style.

Software type: Freemium

19. Puppet (formerly Puppet Labs)

In their own words: "Our platform is the industry standard for automating the delivery and operation of the software that powers everything around us. More than 30,000 companies – including more than

two thirds of the Fortune 100 – use Puppet's open source and commercial solutions to achieve situational awareness and drive software change with confidence."

Software type: Freemium

20. Electric Cloud ElectricFlow

Electric Cloud's ElectricFlow aims to provide "shared visibility and control over enterprise DevOps tools and CD pipelines"

Software type: Freemium

In this post, we've tried to cover some of the major tools that we're seeing and recommending for a CI/CD stack. Inevitably, this isn't a definitive list and there are a lot more tools and services that we could include. To round this out, here are a few other notable development and testing tools that could or should make the cut:

Note: Freemium is a pricing strategy by which a product or service (typically a digital offering or application such as software, media, games or web services) is provided free of charge, but money (premium) is charged for proprietary features, functionality, or virtual goods.

Test Management

21. QMetry Test Manager
22. HPE Quality Center Enterprise
23. ApTest

Test Automation

23. QMetry Automation Studio
24. Selenium
25. Appium

Repository management

26. Artifactory
27. Docker

Code coverage

28. JaCoCo
29. Atlassian Clover
30. SonarQube
31. Cobertura

Behavior Driven Development

32. jbehave
33. Cucumber

Application performance monitoring

34. AppDynamics

Application security testing

35. HPE Fortify

Cloud management

37. Open Stack

Mobile Device clouds

38. Perfecto
39. Sauce Labs

What is your contribution to the Agile World? How would you reach the agile community?

My Contributions to Agile World

Website Launch & Books Published in Amazon

Agile Website & books are released by Syntel CEO Mr. Rakesh Khanna & Temenos+Agility CEO -Ms. Susan Gibson. My books are published in Amazon.com

Agile Coaching Website - https://agilecoachmaster.wordpress.com/

Scrum Alliance Website - https://scrumallianceprofessional.wordpress.com/

Java Website - https://javasrirama.wordpress.com/

Big data Website - https://hadoop.wordpress.com/

SharePoint Website - https://sharepointpractice.wordpress.com/

Agile Books:

- ACP The Premier Guide
- PMP The Premier Guide
- Agile A Key of Success
- Agile coaching
- Agile in Handy

Java Books:

- Core Java Premier Guide Vol I & II
- Responsive Web Design in Handy
- Java Script in Handy

Big data Books:

- Hadoop The Premier Guide
- Hadoop The Premier Lab Guide
- Bigdata Hadoop the Premier Interview Guide

- Agile Coaching in Handy
- Handy Agile
- Scrum Alliance Professional
- Scaled Agile in Handy
- Agile Coach Interview Guide
- Scrum Master I\Guide
- Web Services in Handy
- AngularJS in Handy
- Spring 5 The Premier Guide
- Python Premier Guide

My Agile Community

Member of PMI – PMP, ACP, Scrum Alliance - CSP, ICAgile, SAFe -SPC4, SA, SP, SSM, SASM

What are the trainings you will provide to create Agile awareness across the organization in 2 days?

To create the agile awareness in 2 days across the organization by 3 levels 1 | 2 | 3 the following topics to be handled: -

Level 1

- Agile Intro
- Active listening
- **Agile Manifesto values and principles**
- Assessing and incorporating community and stakeholder values
- Agile Brainstorming techniques
- Building empowered teams
- Coaching and mentoring within teams
- Agile Communications management
- Feedback techniques for product (e.g., prototyping, simulation,
- demonstrations, evaluations)
- Incremental delivery
- Agile Knowledge sharing
- Agile Leadership tools and techniques
- Prioritization
- Agile Problem-solving strategies, tools, and techniques
- Project and quality standards for Agile projects
- Stakeholder management
- Agile Team motivation
- Time, budget, and cost estimation
- Value-based decomposition and prioritization

Level 2

- Agile frameworks and terminology
- Building high-performance teams
- Agile Business case development
- Collocation (geographic proximity)/distributed teams
- Agile Continuous improvement processes
- Elements of a project charter for an Agile project
- Agile Facilitation methods
- Agile Participatory decision models (e.g., input-based, shared collaboration, command)
- Value-based analysis

Level 3

- Agile contracting methods
- Agile project accounting principles
- Applying new Agile practices
- Compliance (organization)
- Control limits for Agile projects
- Agile Failure modes and alternatives
- Globalization, culture, and team diversity
- Agile Innovation games
- Principles of systems thinking (e.g., complex adaptive, chaos) · Regulatory compliance · Variance and trend analysis
- Variations in Agile methods and approaches
- Agile Vendor management

What are the trainings you will provide to create a Scrum Master across the organization in 2 days?

To create the scrum master in 2 days the following topics to be handled:

Day 1: Scrum Basics, Scrum Framework, Scrum Roles, Scrum Artefacts

Day 2: Scrum Ceremonies, Scrum planning & estimations, Agile best practices

Certification Achievements

What are the trainings you will provide to create an Agile Practitioner across the organization in 2 days?

To create the agile project manager in 3 days the following topics to be handled: -

Certification Achievements

1. Introduction to AGILE methodologies

- What is AGILE
- History & Genesis
- Manifesto & principles
- Introduction to methodologies
- CRYSTAL
- SCRUM
- XP
- FDD
- DSDM

2. AGILE implementation in an organization

- AGILE features
- Team composition
- Team dynamics

3. Agile project Life cycle

- Planning – portfolio level
- Planning – project level (Releases and Iterations)
- Executing
- Monitoring & Control
- Closing
- Professional Ethics & Code of Conduct

4. Agile project communications

- Agile Information radiator
- Agile Team space
- Agile tooling
- Osmotic communications for collocated teams
- Osmotic communications for distributed teams
- Agile Daily stand-ups

5. Planning, Monitoring and Adopting

- Agile Retrospectives
- Agile task and Kanban boards,
- Agile Time boxing
- Agile Iteration and release planning
- Agile WIP limits
- Agile Burn down/up charts (Sprint| Iteration | Risk)
- Agile cumulative flow diagrams (CFD)
- Agile process tailoring

6. Agile estimation

- Agile relative sizing/story points
- Agile wide band Delphi /Agile planning poker / Agile affinity estimating / Team Estimation Game Method
- Agile ideal time
- Agile process tailoring

7. Agile analysis and design

- Agile product roadmap
- Agile user stories and backlog
- Agile story maps
- Agile progressive elaboration
- Agile wireframes
- Agile chartering
- Agile personas
- Agile modelling

8. Product quality

- Agile frequent verification and validation
- Agenda for the session
- Agile test first development
- Agile acceptance test-driven development
- Agile definition of done

- Agile continuous integration

9. Soft skills negotiation

- Agile emotional intelligence
- Agile collaboration
- Agile adaptive leadership
- Agile negotiation
- Agile conflict resolution
- Agile servant leadership

10.Value-based prioritization

- Agile return on investment (ROI)
- Agile net present value (NPV) / Agile internal rate of return (IRR)
- Agile compliance
- Agile customer-valued prioritization
- Agile minimally marketable feature (MMF)
- Agile relative prioritization or ranking

11.Risk management

- Agile Risk-adjusted backlog
- Agile Risk Burn down graphs
- Agile risk-based spike

12.Agile Metrics

- Agile velocity
- Agile cycle time
- Agile earned value management (EVM) for agile projects
- Agile escaped defects

13.Agile Value stream analysis

- Agile value stream mapping
- Agile Flow charts
- Agile lean methodology

What are the trainings you will provide to create an Agile Coach across the organization in 3 days?

To create an agile coach in 3 days the following topics to be handled: -

Coaching Fundamentals

- Teaching vs. Mentoring vs. Coaching
- The Agile Coaching Mind-set
- Setting Boundaries for Coaching
- Coaching Agreement

Coaching skillset

- Professional Coaching Skills
- The Coaching Stance
- Responsibilities and Skills of the Coach

The Coaching Process

- Coaching for Potential
- Coaching for Action
- Effective Coaching Conversation

Mentoring and Coaching Agile roles

- Teaching the Agile Basics
- Understanding Agile roles and the Mind-set Shift
- Mentoring Agile Roles & Transitions

Coaching the Journey toward High Performance

- Understanding Team Development
- Setting up the Team Environment
- Handling Conflict and Dysfunction within the Team
- Handling Organizational Impediments

Review and Assessment of Agile Frameworks

Certification Achievements

What are the trainings you will provide to create a SAFe Coaching across the organization?

To create an SAFe coaching with the following topics to be handled: -

Leading SAFe (2 days)

1. Introducing Scaled Agile Framework
2. Embracing a Lean Agile Mindset
3. Understanding SAFe Principles
4. Implementing an Agile Release Train (ART)
5. Experiencing PI Planning
6. Executing and Releasing Value
7. Building an Agile Portfolio
8. Building Really Big Systems
9. Leading the Lean Agile Enterprise

Implementing SAFe (2 days)

1. Reaching the SAFe Tipping Point
2. Designing the Implementation
3. Launching an ART
4. Facilitating an ART Execution
5. Extending to the Portfolio

SAFe Scrum Master (2 days)

1. Introducing Scrum in SAFe
2. Understanding the Role of Scrum Master
3. Experience PI planning
4. Facilitating Iteration Execution
5. Finishing the PI
6. Coaching Agile Team

SAFe Advanced Scrum Master (2 days)

1. Exploring the Scrum Master role in SAFe Enterprise
2. Applying SAFe Principles – A Scrum Master Perspective
3. Exploring Agile and Scrum Anti-Patterns
4. Facilitating Program Execution
5. Improving Flow with Kanban
6. Building High Performance Teams
7. Improving Program Performance with Inspect and Adapt

SAFe Product Manager | Product Owner (2 days)

1. Introduction
2. Embracing a Lean and Agile Mindset
3. Exploring PM PO Roles
4. Contributing to Portfolio Content
5. Defining and Managing Solution Value
6. Being an Effective Product Manager
7. Being an Effective Product Owner
8. Engaging Stakeholders
9. Building Communities of Practice

SAFe for Teams (2 days)

1. Introducing the Scaled Agile Framework
2. Building an Agile Team
3. Planning the Iteration
4. Executing the Iteration
5. Executing the PI

Certification Achievements

Lesson 3 Agile Transformation

How will you transform from Waterfall to Agile?

In Agile transformation, usually I have **done the assessment and laid out a road map** for the team to execute the transformation. During transformation, I will conduct 5 Phases such as orientation, preparation, and execution, repeat & adapt phase to transform from source to destination framework. Collaborate within the organization (client & vendors) to create a customized agile transformation program as mentioned in the agile transformation summary.

Transformation: Summary

Agile Transformation Sum

Transformation: Highlights

1. Do an **as is analysis** with the current situation
2. Define the **to be state** and get stakeholder approval
3. Do the **gap analysis** of what it takes from as is **to the to be state**
4. Identify **action points and metrics**
5. **Roll out the action points** incrementally (Minimum Viable Action Items)
6. **Report to stakeholders with Metrics** set earlier
7. Take **feedback**, have improvement items defined and start the next increment
8. On a parallel note **launch training's and workshops**
9. **Launch Pilot Projects** using the transition model
10. **Inspect and Adapt**
11. **Do assessment** and define the next steps
12. **Continue the process till its stabilized**
13. **Then start scaling**

In detail

Typical **agile transformation services** include:

- Assessments of the current state of an organization
- Roadmap and vision creation
- Creation and launch of the agile transformation program
- Identification of valuable of metrics
- Assisting in establishing internal agile advocates
- Change management
- Engagement with an agile consultant who will provide on-going support and mentorship

Divide the transformation work with client, coach and the team to execute the transformation successfully.

Client Role

Before Agile transformation client usually conduct the benefit analysis.

- Business outcome for agile
- What we have today and what we are going to become tomorrow?
- Gap against Engagement Status with Industry Practices.
- Conduct the maturity level of the organization and project teams over people maturity, technology practices & agile tools, Quality Conscious

Next client follows the **best Practices in Agile Transformation**

- Executive Sponsorship over all phases
- Limit WIP
- Get Agile Coaches in place
- Rolling out Agile Pattern
- Adapt Key Agile Practices
- Inspection and Adaptation of Agility
- Create Quality Metrics
- Met Acceptance criteria to roll out for the organization

Agile Coach Role

Agile Coach has to run the show effectively: -

In transformation, I have done the assessment and laid out a road map and executed the transformation.

Roadmap Flow:

Define the Strategy -> Governance, Structure, Metrics & Tools

Lead the transformation -> Form Teams, Train Teams, Coach Teams

Prepare to go alone -> Assessment, Targeted Coaching, Sustaining Artefacts

Transformation Stage:

During transformation I have conducted orientation, preparation, and execution, repeat & adapt phase to transform from source to destination framework

Orientation Phase

Activity	Outcome
Step 1 - Agile Maturity Assessment	Client Agile Maturity Profile (Level 1 - Level 5)
Step 2 - Agile Inception Workshop	1. Governance 2. Release, Requirements Management 3. Setting up Expectations
Step 3 - Agile Suitability Readiness	Benefits vs Risk, Benefits vs Lost opportunity
Step 4 - Agile Inception Training	Education - Agile SDLC; Exposure - Processes & Tools
Step 5 - Lifecycle, Governance & Contract Definition	Agile Charter
Step 6 - Program Mobilization	Platform Readiness

Preparation Phase (Product Discovery, Base Sprint)

Activity	Outcome
Step 1 - Product Discovery	"1. Vision / Roadmap - Roadmap, Feature Matrix / Prioritization, UI Standards 2. Solution Design - Prototypes / Mock-ups, Features / User Stories, Interface Requirements, NFR's"
Step 2 - Agile Inception Workshop	Key Risks, Groomed Backlog & Success Criteria for initial few sprints

Step 3 - Define Agile-DevOps Strategy	Operational Governance, DevOps Tools, DevOps backlog
Step 4 - Define QA Strategy	Approach & Implementation Strategy for TDD, Regression Testing
Step 5 - Release Planning	Iteration plan for first two iterations, Release Plan

Execution Phase

Activity	Outcome
Step 1 - Iteration Execution	Delivery of potential shippable incremental product
Step 2 - Agility Assessment	Program Agility Index
Step 3 - Retrospective	Lessons learnt, action items for implementation

Repeat Phase

Activity	Outcome
Lessons learnt, action items for implementation	Lessons learnt, action items for implementation

Adopt Phase

Activity	Outcome
Step 1 - Increase Productivity, Reduce time to Market	Business value realization

What are the challenges encountered by you when you started Agile transformation? (or) Tell us about the most challenging obstacle you found while leading an Agile transformation (team or organization) and how did you handle it? (or) We are looking for a talented and motivated agile coach to help us succeed with our agility transformation Why should we pick you?

Top three challenges, blockers, and barriers:

1. Lack of executive commitment
2. People and culture (Adopting across the continent)
3. Neglecting the need for technical excellence

The **major challenges** faced in the **agile transformation** are: -

- Executive Sponsorship over all phases
- Limit WIP
- Get Agile Coaches in place across the continents
- Rolling out Agile Pattern
- Adapt Key Agile Practices to new team members
- Inspection and Adaptation of Agility
- Create Quality Metrics
- Met Acceptance criteria to roll out for the organization

The challenging obstacle during transformation is the **people resistance to change initially**. To overcome this developed Agile Mind-set and Culture and Team Maturity to make them self-managed and collaboratively work together.

I am expertise and handled 20+ agile transformation successfully for the past 6 years of my agile career.

How would you convince your organization to transform to agile?

Reference: -

https://www.mountaingoatsoftware.com/articles/introducing-an-agile-process-to-an-organization

https://www.scrumalliance.org/community/articles/2014/march/managing-organizational-change-in-agile-transforma

In which scenario should we still follow waterfall and not agile?

Reference: - http://www.thedigitalprojectmanager.com/agile-vs-waterfall/

Lesson 4 Scrum Exercises

Agile Intro – Exercise (Say True or False)

1. Agile Manifesto suggests build everything that customer wants – **True**
2. Principle behind Agile Manifesto suggest emerging architecture – **True**
3. As per Agile Manifesto Architecture is not important but the functionality is – **False**
4. Agile Manifesto suggests defining and implementing architecture in first few iterations to enable teams to deliver working software in subsequent iterations – **False**
5. Scrum teams place value i planning but value responding to change even more – **True**
6. Scrum framework is good for projects, which are simple and predefined – **False**
7. According to Agile Manifesto, control and management is important – **False**
8. When Agile Manifesto suggests "Responding to change", it means releasing products soon and often to test assumptions – **True**
9. Regular and frequent feedback is essential to address customer collaboration suggested by Agile Manifesto –**True**
10. Agile Manifesto suggests not to document anything but write code – **False**
11. Time boxing is one of the ways to maintain sustainable pace suggested by Agile principles – **True**
12. Any Process which exhibits transparency and helps the team to inspect and adapt based on evidence is called Empirical Process – **True**

Scrum Roles – Exercise – 1 (Say True or False)

Product Owner

1. Product Owner manages the release – **True**
2. Product Owner clarifies the requirement – **True**
3. Product Owner manages the tasks – **False** (Development Team takes care)
4. Product Owner tracks the progress in the sprint – **False**
5. Product Owner visits customers – **True**
6. Product Owner is responsible for maximizing the value of work done by the Scrum Team – **True**
7. Product Owner participates in daily scrum –**False**
8. Product Owner understands the competitors – **True**
9. Product Owner is available to team during the sprint – **True**

10. Product Owner decides how much work to do in the sprint – **True**
11. Product Owner defines the architecture – **False**
12. Product Owner understands how our product is used by the customers – **True**
13. Product Owner evangelizes the product within and outside of organization – **True**
14. Product Owner orders the product backlog – **True**
15. Product Owner maintains the product backlog – **True**
16. Product Owner envision is the product – **True**
17. Product Owner is responsible for the profitability of the product –**True**

Scrum Master

1. Scrum Master commits work to product owner – **False**
2. Scrum Master manages the progress in the sprint – **True**
3. Scrum Master's product is a well-known working team – **True**
4. Scrum Master is NOT responsible for the schedule – **False**
5. Scrum Master needs to coach Product Owner – **False**
6. Scrum Master is a change agent for Scrum – **True**
7. Scrum Master needs to be the master of Scrum – **False**
8. Scrum Master works in full time – **False**
9. Scrum Master is the manager of the team – **False**
10. Scrum Master is responsible for solving the team's problem – **True**
11. Scrum Master manages the dependency with other teams – **True**
12. Scrum Master helps team adopt engineering practices – **True**
13. Scrum Master manages the project – **False**
14. Scrum Master facilitates the Scrum meetings- **True**
15. Scrum Master is the interface towards outside the team – **False**

Development Team

1. Team owns and improves its process – **True**
2. Team decides the priority of the work – **False**
3. Team manages the dependency with other teams – **False**
4. Team sets its own direction – **True**
5. Team commits sprint goal –**True**

6. Team maintains product backlog – **False**
7. Team maintains sprint backlog – **True**
8. Team manages the tasks – **True**
9. Team updates the sprint Burn down chart – **True**
10. Team updates the release Burn down chart – **False**
11. Team decides the architecture and design – **True**
12. Team follows Scrum Master's command – **False**
13. Team follows Product Owner's command – **False**
14. Team is not allowed to have conversation with customers – **False**
15. Team does testing – **True**
16. Team is responsible for getting done – **True**

Scrum Roles Mapping

A servant-leader for the Scrum Team!!!

Accountable for building the high value products!!

Acts as a change agent that increases the productivity of the Scrum Team!!!

Coaches the Development Team for self-organization and cross-functionality!!!

Removes impediments to the Development Team's progress!!!

Details out the product backlog items appropriately!!

Empowered to make decisions on how to build the product increment!

Empowered to make decisions regarding the product backlog management to achieve the vision!!

Ensures the Product Owner knows how to order the Product Backlog items to maximize value!!!

Facilitates Scrum events as requested or needed!!!

Finds and teaches techniques for effective Product Backlog management to Product Owner!!!

Helps employees and stakeholders of the organization understand and enact Scrum and empirical product development!!!

Helps engineering team in estimating the backlog items by clarifying backlog items!!

Helps optimize the external interactions with the scrum team to maximize the value created!!!

Helps product owner in backlog management by explaining technical constraints.

Helps product owner prioritize the work. Teaches PO and stakeholders value based prioritization!!!

Helps Product Owner understanding product planning in an empirical environment!!!

Helps the Scrum Team understand the need for clear and concise Product Backlog items!!!

Leads and coaches the organization in its Scrum adoption!!!

Optimizes the value of the work the development team performs!!

Participates in all Scrum events!

Responsible for building the product fast by eliminating the waste!!!

Responsible for creating and establishing the product vision!!

Responsible for creating and managing the release plans!!

Responsible for creating and managing the sprint backlog!

Responsible for creating the product increment!

Responsible for deciding whether to release the product increment or not!!

Responsible for ensuring Scrum is understood and enacted!!!

Responsible for identifying and eliminating "Technical Debt"!

Responsible for implementing good engineering practices!

Responsible for keeping the Product Backlog transparent, clear and visible to everyone!!

Responsible for learning all the functions required to deliver a product increment!

Responsible for managing the Product Backlog!!

Responsible for ordering the backlog items to maximize the value delivery!!

Responsible for the quality of the product increment!

Responsible for tracking the progress of the sprint!

Responsible for tracking the release progress!!

Responsible for understanding and answering any question related to product domain!!

Reviews the product increment and gives feedback to the development team!!

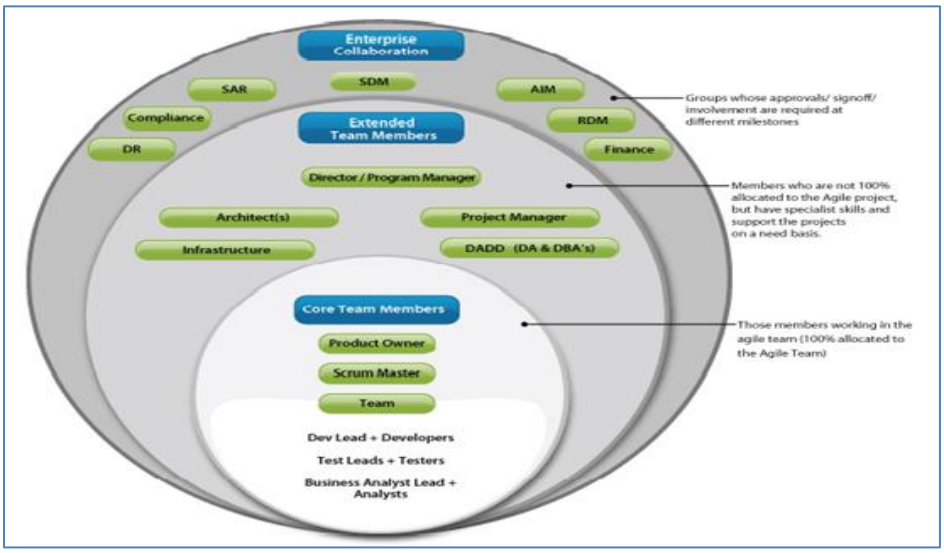
Answers

Ends with! – Development Team | Ends with!! – Product Owner | Ends with!!! – SCRUM Master

Scrum Roles – Exercise - 2

Role Mapping

Map the roles listed below to Scrum Roles based on the similarity in responsibilities and skills required



- **Product Manager – Product Owner**
- **Product Marketing Manager – Product Owner**
- **Test Engineer – Development Team**
- **Programmer – Development Team**
- **Technical Architect – Development Team**
- **Build and Release Engineer – Development Team**
- **Pre-sales – Product Owner**
- **Technical Writer – Development Team**
- **Project Manager – Other roles**
- **Business Analyst – Development Team**
- **User Experience Expert – Development Team**
- **User Interface Specialist – Development Team**

Responsibility Mapping

Map the roles listed below to Scrum Roles based on your understanding of Scrum so far

- **Facilitate Delivery – Scrum Master**
- **Motivate Team – Scrum Master**
- **Manage product release schedule – Product Owner**
- **Manage Product Budget – Product Owner**
- **Manager Product Scope – Product Owner**
- **Manage release decisions – Product Owner**
- **Identify and address systematic problems and address them – Scrum Master**
- **Build learning team – Scrum Master**
- **Estimate Work – Development Team**
- **Provide technical solutions – Development Team**
- **Remove Impediments – Scrum Master**
- **Manage Product Risks – Product Owner**
- **Manage Product dependencies – Product Owner**
- **Conduct User Research – Development Team**
- **Product Promotion – Product Owner**
- **Product Positioning – Product Owner**
- **Maximize team potential – Scrum Master**

Skill Mapping

Map the roles listed below to Scrum Roles based on the responsibilities they need to perform

- **System Thinking – Scrum Master**
- **Decision Making – Product Owner**
- **Product Marketing – Product Owner**
- **Customer Empathy -Product Owner**
- **Business Acumen -Product Owner**

- **Technical Skills – Development Team**
- **Domain knowledge – Product Owner**
- **Process Expertise – Scrum Master**
- **Leadership – Scrum Master**
- **User Experience Skills – Product Owner**
- **Product Discovery Skills – Product Owner**
- **Stakeholder Management – Product Owner**
- **Facilitation – Scrum Master**
- **People Skills – Scrum Master**
- **Coaching – Scrum Master**
- **Process Expertise / Champion – Scrum Master**
- **Collaboration – Scrum Master**
- **Emotional Intelligence – Scrum Master**
- **Team Building – Scrum Master**

Scrum Artifacts – Exercise

Say True or False

General

1. Product Owner is responsible for establishing the product vision – **True**
2. Higher ordered Product Backlog Items are usually clearer and more detailed than lower ordered ones – **True**
3. Anyone can create Product Backlog Items, but the Product Owner has overall responsibility to manage it – **True**
4. Life of the Product backlog is same as life of the product itself – **False**
5. The Development team is responsible for estimating of backlog items – **True**
6. Product Backlog Refinement is the act of adding detail estimates, and order to items i n the Product Backlog – **True**
7. Team and PO decide the frequency and duration of the backlog refinement meeting. However, it is time boxed at 10% of total available time – **True**
8. Product Backlog Refinement is an Ongoing activity throughout a scrum project – True
9. Having a Product Backlog Refinement meeting helps sprint planning to go smoother – **True**
10. A single team works from multiple product backlogs – **False**
11. Product Owner keeps the product backlog somewhere secretly so that no one can touch it – **False**
12. The product backlog is sorted from small items (At Top) to large (At Bottom) – **False**
13. The product backlog should be updated only in backlog refinement meeting – **False**
14. In backlog refinement, the team goes through the entire backlog and refines it – **True**
15. Development team may work on critical engineering items without placing them in product backlog – **False**
16. Once the product backlog is created it is usually does not change – **False**
17. Scrum Master tells product owner what should be the priority in backlog refinement meeting – **False**

18. Once can item be placed in the product backlog, it is never re-ordered – False

Product Backlog

1. It is an ordered list of everything that might be needed in the product – **True**
2. It's a single source of requirements for any changes to made the product – **True**
3. The Product Owner is responsible for the product backlog, including its content, availability, and ordering – **True**
4. A Product Backlog is never complete – **True**
5. As long as a product exists, its product backlog also exists – **True**
6. The defect should not be added to product backlog – **False**
7. Changes in business requirements, market conditions, or technology may cause changes in the product backlog – **True**
8. Multiple scrum teams can work together on the same product – **True**
9. Product Backlog items can be updated at any time by the product owner or at the product owner's discretion – **True**
10. All the items in product backlog are comprehensively detailed – **False**
11. The Product Backlog Item should always be written in the form of a User Story – **False**

Sprint Backlog

1. The Sprint Backlog is a forecast by the Development Team about what functionality will be in the next increment – **True**
2. The Development Team modifies the Sprint Backlog throughout the Sprint, and the Sprint Backlog emerges during the Sprint – **True**
3. Sprint Backlog is just a list of tasks to be completed during the sprint – **True**
4. Development team should pull stories in the order of priorities set by PO – **True**
5. Only the Development team can change its Sprint Backlog during a sprint – **True**
6. By tracking the remaining work throughout the Sprint, the Development Team can manage its progress- **True**
7. Sprint Backlog is used by the team to manage their hours – **True**

8. **The main purpose of sprint Backlog is for the team to manage themselves – True**
9. **Sprint Backlog is for Product Owner to understand what the team has committed for the sprint – True**
10. **Sprint Backlog does not contain plan for delivering the product Increment and realizing the Sprint Goal – True**

Product Increment

1. The increment is the sum of all the Product Backlog items completed during a Sprint and the value of the increments of all previous Sprints – **True**
2. At the end of a Sprint, the new increment must be "Done", which means it must be in useable conditions and meet the Scrum Team's Definition of Done" – **True**
3. Development team is responsible for producing product increment – **True**
4. Product Increment is "Done". Only when stakeholders accept the stories in Sprint Review – **True**
5. Product Increment is always ready to be deployed at customer site – **True**
6. Engineering practices like TDD, continuous integration, pair programming improve the quality of Product increment– **True**
7. Velocity is the measure of how much valuable work that a team can do in a Sprint. – **True**
8. One of the ways to determine it is to sum up the story points associated with Product Backlog Items that are part of the Product Increment at the end of the Sprint – **True**
9. Product Owner guides team how to produce the Product Increment– **True**

Product Backlog Refinement

1. It's best practice to hold Backlog Refinement at least once per Sprint – **True**
2. The duration of the Backlog Refinement meeting is 2 hours per Sprint – **False**
3. It's the act of adding details, estimates and order PBI's – **True**
4. It's an ongoing collaboration between the Product Owner and the development team – **True**
5. It's a part time activity during the Sprint. How and when is decided by the Scrum Team – **True**
6. PBI is always estimated in hours in Backlog refinement meeting – **False**
7. Backlog Refinement is requirement to keep the Product Backlog DEEP for at least 2 Sprints- **False**
8. The focus of the Backlog Refinement is future release – **False**

9. **Backlog Refinement is also known as Story Time, Backlog grooming ad user Story Workshop – True**
10. **Development Team is responsible for all estimates – True**
11. **The Product Owner may influence the Development Team by helping it understand and select trade-offs, but the people who will perform the work make the final estimate – True**
12. **The PBI's once estimated are never reviewed and revised during Backlog Refinement – False**

Scrum Meetings – Exercise 1

Say True or False

What did you do yesterday? What will you do today? And which team members do you need to talk to? are the three questions of a daily scrum (FALSE)

Scrum Master should use the Daily Scrum to solve any impediments the team raises (FALSE)

Product Owner is not allowed to talk during the Daily Scrum (TRUE)

Main purpose of the Daily Scrum is to inspect the previous day's work and plan for the next day (TRUE)

Daily Scrum is a good way for the Scrum Master to secretly micro-manage the team (FALSE)

If the team is talking about a critical issue in daily scrum, the meeting could be extended until the discussion is completed (FALSE)

Daily Scrum is time boxed to 15 minutes (TRUE)

The purpose of the Sprint Review is to get feedback on what the team has accomplished during the sprint (TRUE)

Though Product owner calls few key stakeholders to Sprint Review, anyone who is interested can go (TRUE)

In the Sprint Review, team gives a power point presentation on what they worked on in the sprint (FALSE)

Sprint Review is the meeting where User Acceptance Testing is performed (FALSE)

In Sprint Review, Scrum Team discusses the current state of the product backlog and future direction for the product (TRUE)

At the end of the Sprint Review, release plan is updated if needed (TRUE)

Product Owner should wait till sprint review to give any feedback to the development team (FALSE)

Any feedback in Sprint Review that turns into new backlog item should be maintained in a separate backlog (FALSE)

Sprint Retrospectives should be held only at the end of the last sprint in a project or release (FALSE)

Sprint Retrospective is the place for scrum master to do the post mortem of the sprint (FALSE)

The Scrum Master participates as a peer team member in the sprint retrospective meeting from the accountability over the Scrum process (TRUE)

Scrum Master should create a safe environment for the team so that they will open up in Sprint Retrospective (TRUE)

Executives can also attend the Sprint Retrospective to know team's problems (FALSE)

Team should only inspect and adapt in retrospective and shouldn't think about it rest of the time (FALSE)

At the end of the Sprint Retrospective, major improvements are identified and an action plan to is created (TRUE)

The main theme of the Sprint Retrospective is to answer "How can we get better as a team?" (TRUE)

Sprint Planning is time boxed to 8 hours for 4 weeks sprint, but most teams try to get done in about half as long (TRUE)

Tasks in the Sprint Backlog are estimated collaboratively by the team even though one person will do the task (TRUE)

During the Sprint Planning meeting, the Scrum Master assigns tasks to individual team members (FALSE)

During the Sprint Planning, the Product Owner decides what items should be implemented in the sprint (FALSE)

Main purpose of the Sprint Planning is to come up with the forecast for the sprint (TRUE)

One of the main topics of the Sprint Planning is Product Owner explaining the details of the items (TRUE)

Team talks about possible solutions, high level design, automation strategy etc., during the Sprint Planning (TRUE)

Scrum Master comes up with the Sprint Goal at the end of the Sprint Planning (FALSE)

Scrum Meetings – Exercise 2

Say true / false

The Sprint

1. It's a time-box of one month or less – **True**
2. Sprint length varies from Sprint to Sprint based on the PBI's selected for the Sprint– **True**
3. Sprint is done when all the committed PBI's meet definition of Done – **True**
4. Sprint Length is decided based on the size of the team – **True**
5. Sprint Length is decided based on how soon the Development team can deliver – **True**
6. Working product and how often the requirement changes – **True**
7. During first sprint, the Team works on architecture and plan for the rest of the Sprint – **True**
8. The team works on accomplishing the Sprint goal irrespective of whether its first or last Sprint – **True**
9. No changes are allowed to Sprint Goal, Length and Team during the Sprint – **True**
10. The Scope of the Sprint can't be reorganized between the team and PO – **True**
11. The Sprint can be cancelled before the Sprint Time Box is over – **True**
12. None of the Work done in a cancelled Sprint is accepted – **True**

Sprint Planning

1. The team can proceed with Sprint Planning without Product Owner – **True**
2. The PO presents his/ her wish list and team asks questions to understand what needs to be built – **True**
3. While estimating PBI size in Sprint Planning is technically allowed, it's best to estimate prior to planning – **True**
4. The Product owner assigns development tasks to the team in Sprint Planning – **True**
5. The Product Owner gets involved in figuring out how code is written – **True**

6. The Team may split the PBI's in to tasks and estimate them in hours – **True**
7. The Team can commit to Sprint Goal based on either velocity or capacity – **True**
8. The Sprint Planning for a 2-week Sprint should be time-boxed to 4 hours – **True**
9. Adding a Backlog Refinement session half way through your sprint can help with making Sprint Planning shorter and more effective – **True**
10. During Sprint Planning, PO decides how many stories will be delivered by the end of the Sprint – **True**
11. The Scrum Team is responsible for crafting the Sprint Goal – **True**

Daily Scrum

1. It's fine for the Daily Scrum to last up to 45 minutes – **True**
2. People from outside the team may attend, provided it's okay with the team and they don't interfere – **True**
3. The main purpose of Daily Scrum is to inspect where the team stands with respect to Sprint Goal and adapt its plan to get there – **True**
4. The Daily Scrum is a status update meeting for the Product Owner – **True**
5. The Participation of Product Owner in Daily Scrum is decided by the Team – **True**
6. Discussions of how to resolve issues should be done during a sidebar after the Daily Scrum – **True**
7. While PO is optional for DSM, PO can attend to see the progress being made and determine if team needs help – **True**
8. It's helpful for the Daily Scrum to be in the same location, at the same time every day – **True**
9. Managers should attend the Daily Scrum to help ensure everyone is doing enough work – **True**
10. The Daily Scrum should not start until all participant arrive – **False**
11. If a team member is consistently late for the Daily Scrum, the Scrum Master should talk to the team member to figure out next course of action – **True**
12. The Sprint Burndown Chart should be updated in the Daily Scrum – **True**

13. The Team members should use only Daily Scrum to communicate impediments – **True**

Sprint Review

1. It's expected that a team will demo a feature even if they did not finish it in the Sprint – **True**
2. The main purpose of the Sprint Review is for the Product Owner to accept the work done by the team – **True**
3. The main purpose of Sprint Review is for the Stakeholders to review the work done by the team and provide feedback – **True**
4. A PBI accepted by PO during the Sprint remains accepted even if the stakeholders disagree – **True**
5. Comments from stakeholders on accepted PBI's might result in a new PBI that will be added to Product Backlog – **True**
6. Team should spend at least 4 hours preparing for the review, and the review should include power point slides – **True**
7. If the team cannot demo a feature because it was not completed, it's okay for a manager to question the team's approach in this meeting – **True**
8. The Sprint Review is mainly for show and has little impact on the product – **True**
9. The Sprint Review is one of the ways in which the SCRUM address customer collaboration – **True**
10. The Sprint Review helps team in responding to change better as the assumptions will be tested here – **True**
11. A manager should pick who does the demos during the Sprint Review – **True**
12. Feedback from stakeholders during the Sprint Review should be collected and put on the backlog for consideration by the Product Owner instead of spawning long conversations during the review – **True**
13. The Product Owner discuss how the Product Backlog stands at the moment and projected dates of completion during the Sprint Review – **True**

Scrum Meetings Comparison

1. Sprint Planning		2. Daily Scrum	
Who	Product Owner, Dev Team	***Who***	Dev Team
When	First day of Sprint	**When**	Entire Sprint duration
Time box	Max 8 hours	**Time box**	Max 15 Mins
What	* Sprint Back log * What to do? – Product Owner * How to do? – Team	**What**	* What have I done since last scrum? * What do I plan to do today? * Are there any impediments to progress?
Input	Prioritized Product Backlog	**Input**	Sprint Backlog
Outcome	Sprint Goal, Sprint Backlog	**Outcome**	Updated Sprint Backlog
3. Sprint Review		**4. Sprint Retrospective**	
Who	Stakeholders, Product Owner, Scrum Master	***Who***	Scrum Master, Product Owner, Team
When	Last day of Sprint	**When**	End of Sprint
Time box	Max 4 hours for 4 weeks	**Time box**	Max 3 hours
What	Review Product Increment & give feedback, feedback goes to product backlog	**What**	* What went well * What did not go well * Any Improvements * Identify the improvements list
Input	Product Increment	**Input**	Feedback / Experience of team members
Outcome	Product Backlog (Revised)	**Outcome**	List of Improvements

Lesson 5 Scrum Videos

1: Introduction to Scrum

1.Introduction – https://www.youtube-nocookie.com/embed/KnAYUxnQ9G0

2.Scrum Master- https://www.youtube.com/watch?v=eNe0UEsBalA

3.Product Owner- https://www.youtube.com/watch?v=502ILHjX9EE&t=1s

4.Development Team - https://www.youtube.com/watch?v=3kK-QMdl-Ew

5.Motivational: https://www.youtube.com/watch?v=u6XAPnuFjJc&t=266s

2: Knowing Scrum

1.Introduction: Knowing Scrum – https://www.youtube-nocookie.com/embed/_7TgCB53x6U

2.Core Risks of Software Project – https://www.youtube-nocookie.com/embed/_EF9BduJAPY

3.Definition of Scrum – https://www.youtube-nocookie.com/embed/MmnEjsN1gjI

4.Empirical Process – https://www.youtube-nocookie.com/embed/16I25gISCZQ

5.Scrum Values – https://www.youtube-nocookie.com/embed/47mO9l_kAOY

6.Scrum in Action – https://www.youtube-nocookie.com/embed/4ErD5UHzAHI

7.Mitigating Software Project Risks with Scrum – https://www.youtube-nocookie.com/embed/1QdtS6gR4tM

8.Summary: Knowing Scrum – https://www.youtube-nocookie.com/embed/mT-53w5kXEQ

9.Knowing Scrum Quiz - http://quampus.com/common/popups/viewswf.aspx?sfref=hx68dCGWGWj9cFphXfwdLg==

3: What is Agile?

1. What is Agile? Intro – https://www.youtube-nocookie.com/embed/ThQkdEDn8m0

2.Agile Manifesto – https://www.youtube-nocookie.com/embed/hwEpYnPkvNE

3.Agile Principles – https://www.youtube-nocookie.com/embed/UeeTQ-kVwB0

4.Agile and Scrum – https://www.youtube-nocookie.com/embed/3fqr5m4iVTI

5.What is Agile? Quiz – http://quampus.com/common/popups/viewswf.aspx?sfref=TaXBigibK-_!JdB-_!ikRa3RIQ==

4: Scrum Roles

1.Scrum Roles – https://www.youtube-nocookie.com/embed/vXiz4kkTmk4

2.Product Owner – https://www.youtube-nocookie.com/embed/EVbf5ZcrUIU

3.Scrum Master – https://www.youtube-nocookie.com/embed/I6Upen0gKY8

4.Development Team – https://www.youtube-nocookie.com/embed/FpeJ7HiRis8

5.Scrum Roles Quiz – http://quampus.com/common/popups/viewswf.aspx?sfref=V3rL-!_m4gNVrLRxMds9rQiw==

5:Sprint

1.Sprint Selection Introduction – https://www.youtube-nocookie.com/embed/mfqw7eGftF4

2.Sprint – https://www.youtube-nocookie.com/embed/ec3DONr7vm8

3.Sprint Length -https://www.youtube-nocookie.com/embed/kVS68xhxJtA

4.Consistent Sprint Duration – https://www.youtube-nocookie.com/embed/GcQNLTcaGPE

5.No Change During the Sprint – https://www.youtube-nocookie.com/embed/rsrMxK-dy30

6.Sprint Anti Pattern – https://www.youtube-nocookie.com/embed/vn4luD5zyqM

7.Sprint Selection Summary – https://www.youtube-nocookie.com/embed/BvjlRRVtnx4

8.Sprint Quiz – http://quampus.com/common/popups/viewswf.aspx?sfref=-!_u1YFU9quM3feMxBqL63bQ==

6: Product Backlog

1.Product Backlog Introduction – https://www.youtube-nocookie.com/embed/ZjrfktzQ2BI

2.Product Backlog – https://www.youtube-nocookie.com/embed/LY7r1SMi58A

3.Product Backlog Items – https://www.youtube-nocookie.com/embed/-gzMphR5hnE

4.Estimating PBI – https://www.youtube-nocookie.com/embed/YtxsQdq2JPg

5.Prioritizing Backlog – https://www.youtube-nocookie.com/embed/qWoHTd0YMhI

6.Product Backlog Refinement – https://www.youtube-nocookie.com/embed/9wSW1A8RVrA

7.Product Backlog: Summary – https://www.youtube-nocookie.com/embed/VIlXBu2NlEU

8.Product Backlog Quiz – http://quampus.com/common/popups/viewswf.aspx?sfref=ItgQQS0YQCNsN0C-!_zGoMCg==

7: Planning Sprint

1.Planning Sprint Introduction – https://www.youtube-nocookie.com/embed/ieyLBCVN3AI

2.Definition of Done(DOD) – https://www.youtube-nocookie.com/embed/xSY5FQ9YNyM

3.Sprint Planning Meeting – https://www.youtube-nocookie.com/embed/8ATyApKEN08

4.Sprint Goal – https://www.youtube-nocookie.com/embed/83KcVpHDHvQ

5.Sprint Backlog – https://www.youtube-nocookie.com/embed/iMYwlhcQ_4c

6.Release Planning – https://www.youtube-nocookie.com/embed/Y9QahfmVVzs

7.Summary – https://www.youtube-nocookie.com/embed/ZCUvfJLmcSQ

8.Planning Sprint Quiz – http://quampus.com/common/popups/viewswf.aspx?sfref=l0yO7H-!_M4tBPMwL6Fy6G0A==

8: Executing and Monitoring Sprint

1.Executing and Monitoring Sprint – https://www.youtube-nocookie.com/embed/GVZBFRp4qfo

2.Scrum Artefacts – https://www.youtube-nocookie.com/embed/d26O6eyunhA

3.Not to Waterfall Sprint – https://www.youtube-nocookie.com/embed/93byBriNDIw

4.Working with Task Board – https://www.youtube-nocookie.com/embed/hqA_BWrJxi4

5.Datily Scrum – https://www.youtube-nocookie.com/embed/d9kKMXJTEIY

6.Technical Practices – https://www.youtube-nocookie.com/embed/zrdMRM-HAEI

7.Sprint Burn-Down Chart – https://www.youtube-nocookie.com/embed/9w7P-F8-PH0

8.Executing and Monitoring: Summary – https://www.youtube-nocookie.com/embed/dG-wxFdGhUc

9.Executing and Monitoring Sprint Quiz

– http://quampus.com/common/popups/viewswf.aspx?sfref=69PNIPi2H-!_orfenGecpXtw==

9: Concluding Sprint

1.Concluding Sprint: Introduction – https://www.youtube-nocookie.com/embed/95pleABLp-U

2.Sprint Review – https://www.youtube-nocookie.com/embed/2g3TSqsQZOQ

3.Sprint Retrospective – https://www.youtube-nocookie.com/embed/zHiR3gHC8HY

4.Metrics – https://www.youtube-nocookie.com/embed/hznAQn2uwqM

5.Scrum Events – https://www.youtube-nocookie.com/embed/0X10DqBZn0Q

6.Concluding Sprint: Summary – https://www.youtube-nocookie.com/embed/8KFFgS2IMek

7.Concluding Sprint Quiz – http://quampus.com/common/popups/viewswf.aspx?sfref=BopfF-_!NQc-!_eKGGuV-!_in-_!Eg==

10: Advance Topics

1.Scaling Scrum – https://www.youtube-nocookie.com/embed/i0dOwyEu-64

2.Project Manager and Scrum – https://www.youtube-nocookie.com/embed/dSI2QvIb_SE

3.Advance Topics Quiz

– http://quampus.com/common/popups/viewswf.aspx?sfref=vezP9S6CgwShV17VadSrNQ==

4.What Next? – https://www.youtube-nocookie.com/embed/BRq3_bIm83Y

11: Additional Learning

1.Scrum Guide – http://quampus.com/content/get/rVC50cVg2i05qJ0OYFk83A==.pdf

2.Knowing Scrum Flash Cards – http://quampus.com/content/get/n7smItQ3qrgefCgdTIbi2Q==.pdf

3.Scrum Roles Flash Cards – http://quampus.com/content/get/k6bgILWLshyddFneIUH-!_GQ==.pdf

Lesson 6 Certified Scrum Master – Practice Test

1. What does the Product Owner do during a Sprint? Discuss

- A. Protects the Team and the process
- B. **Clarifies requirements and answers questions**
- C. Guides the Team in its work
- D. Intervenes when required to make sure the pace of work is sustainable

2. Which role is responsible for turning the Product Backlog into incremental pieces of functionality? Discuss

- A. Product Owner
- B. Scrum Master
- C. **Team**
- D. Everyone within the Project

3. If a Team member is consistently late for the Daily Scrum, what is usually the first thing a Team should do?

- A. Report the Team member to his or her manager
- B. Have the Team member do the testing
- C. **Meet with the Team member to determine a solution**
- D. Ask the Scrum Master to move the Team member off the Team

4. During a Daily Scrum meeting, Olivia mentions she has found some open source code she thinks will solve one of the problems she has been working on. She wants to implement it immediately. What is the best next step? Discuss

- A. The Scrum Master tells Olivia to prepare an example and presentation for the Team so they can consider using the code
- B. **After the Daily Scrum meeting is held, a separate meeting is conducted to discuss the open source solution**
- C. All members of the Team are told to evaluate Olivia's solution and report back to the team at the next Daily Scrum meeting.
- D. The Product Owner notes the impediment and solves the problem after the meeting

5. Who is responsible for facilitating the Sprint Retrospective Meeting? Discuss

- A. Team
- B. Product Owner
- C. **Scrum Master**
- D. No one

6. Which role is MOST LIKELY to communicate an impediment during a Daily Scrum? Discuss

- A. Product Owner
- B. **Team**
- C. Stakeholders
- D. Scrum Master

7. Who is primarily responsible for maintaining the Product Backlog? Discuss

- A. Scrum Development Team
- B. Scrum Master
- C. **Product Owner**
- D. Stakeholders

8. What is the Scrum Master's role in the Sprint Retrospective?

- A. To determine the re-composition of the Team
- B. **To facilitate the Team's search for improvements**
- C. To lead the Team in the evaluation of each individual Team member
- D. To provide answers to the challenges that the Team identifies

9. Which of the following is true concerning impediments?

- A. The Team should not use daily Scrum meetings to report impediments
- B. **It is the Scrum Master's top priority to remove impediments**
- C. It is the Product Owner's job to remove impediments.
- D. A slow running server is not considered an impediment

10. Why should the Product Owner attend the Daily Scrum? Discuss

- A. To comment on the Team's progress
- B. To make sure the Team is still on target for its Sprint goals
- C. To tell the Team which tasks to work on next and update the Product Backlog
- D. **To see the progress being made and determine whether the Team needs help**

11. In a 30-day Sprint, how long is the Sprint Review Meeting?

- A. **Four hours maximum**
- B. Four to eight hours
- C. At least eight hours
- D. As long as required

12.Which of the following is a responsibility of the Product Owner? Discuss

- A. **Determine the appropriate release dates**
- B. Determine the appropriate technical solution for the project.
- C. Determine the Team composition necessary for success
- D. Determine the length of the Sprints

13. What are the desirable qualities of a Product Vision? Discuss

- A. Features a detailed overview that enlightens and inspires
- B. Provides a complete breakdown structure of the ROI formula
- C. **Describes why the project is pursued and the product desired end state**
- D. Outlines traceability back to overall corporate governance in IT investment

14. Which of the following statements is TRUE about Scrum teams and planning? Discuss

- **A. Scrum Teams place value in following a plan, but they value responding to change even more**
- B. Planning is not important in Scrum.
- C. Scrum is intended to be an efficient way to carry out plans that have already been made
- D. Traditional planning is replaced by the Sprint Burn down chart

15. The Scrum Master… Discuss

- A. Controls the priority order of items in the team's backlog
- **B. Is the Team's Scrum expert.**
- C. Creates, refines and communicates customer requirements to the Team
- D. Is the keeper of the product vision

16. According to Scrum guidelines, who is responsible for hiring or assigning a new person into a Team? Discuss

- A. Scrum Master
- B. Product Owner
- C. This is outside of the scope of Scrum
- **D. The self-managing Team**

17. If a Team determines that it has over-committed itself for a Sprint, who should be present when reviewing and adjusting the Sprint goal and work? Discuss

- A. Product Owner and Stakeholders
- **B. Product Owner, Scrum Master, and Team**
- C. Scrum Master, Project Manager, Team
- D. Team

18. What does Scrum's definition of done help a Team produce? Discuss

- A. Functionality that has been deployed to the users and delivers real business value
- B. Functionality that has been designed and analysed
- C. Product functionality ready to be tested
- **D. An increment of potentially shippable product**

19. How do the principles behind the Agile Manifesto suggest approaching architecture? Discuss

- **A. Architecture emerges**
- B. Architecture is not important, but functionality is important
- C. Architecture is defined and planned up front
- D. Architecture is defined and implemented in the first iterations

20. How are the Product Owner's responsibilities BEST described?

- A. Directing the Team's daily activities
- B. Keeping stakeholders from distracting the Team

- C. Managing the project and ensuring that the work meets the commitments to the stakeholders
- D. **Optimizing the business value of the work**

21. After a Team has committed to a Sprint goal, what authority does it have?

- A. The Team has authority to swap Sprint backlog items with Product Backlog items if it cannot finish them.
- B. The Team works under the authority of the Product Architect, who has set the definition of done
- C. The Team works according to the priorities set by the Scrum Master, as the Scrum Master is committed to the Scrum framework
- D. **The Team does whatever is necessary to achieve the goal**

22. Which of the following is a MAIN purpose of a Sprint Backlog? Discuss

- A. For the Scrum Master to manage the progress during the Sprint
- B. **For the Team to manage themselves during the Sprint**
- C. For the Team to manage the number of hours spent on tasks in the Sprint
- D. For the Product Owner to understand what the Team has committed to for a Sprint

23. What is the main purpose of a Sprint Review?

- A. For the Team to review their work and to determine what is needed to complete the next set of backlog items
- B. **For Stakeholders to review what the Team has built and to give input on what to do next**
- C. For Stakeholders to "hold the Team's feet to the fire" – to make sure something is produced during the Sprint
- D. For the Product Manager to be able to show progress to the Stakeholders

24. What does the Team do during the first Sprint?

- A. Delivers design documents
- B. Develops a plan for the rest of the Sprints
- C. **Accomplishes the Sprint goal**
- D. Predetermines the complete architecture and infrastructure

25. When using Scrum, who is primarily responsible for making scope versus schedule trade-off decisions?

- A. The Project Manager
- B. The Team
- C. **The Product Owner**
- D. The Scrum Master

26. Can the Product Owner and the Scrum Master be the same person?

- A. **No. The person would have too much power and it would create confusion**
- B. Yes, if the person has the authority and empowerment to do both things
- C. No. It would take too much of one person's time

- D. Yes, as long as the person can balance both responsibilities with care

27. How does the Agile Manifesto address planning?

- A. Planning is not required in an agile project, as the project is focused on current status
- **B. Responding to change is more important than following a plan**
- C. Sign-off on the detail of Product Backlog items is mandatory before any item can be planned into iteration.
- D. Upfront planning and design is an integral stage before development can begin.

28. When should the Burn Down Chart be updated?

- A. After every Sprint
- **B. After every day**
- C. After every release
- D. After every week

29. Once a Team starts a Sprint, who determines how the Team does it work?

- A. Project Manager
- B. Team lead
- C. Scrum Master
- **D. Team**

30. Which statement is accurate about the role of the Product Owner during the Daily Scrum?

- **A. The Product Owner's participation is defined by the Team**
- B. The Product Owner outlines the additional changes that must be absorbed by the Team in the Sprint
- C. The Product Owner ensures the Burn down rate is maintained at the estimated rate
- D. The Product Owner provides instruction to the Team on how to implement a workable solution

31. The CEO asks the Team to add a story to the current Sprint. What should the Team do?

- A. Respect the CEO's authority and add the story to the current Sprint without any adjustments
- B. Add the story to the next Sprint.
- **C. Inform the Product Owner so he/she can work with the CEO**
- D. Add the story to the current Sprint and drop a story of equal size

32. What is MOST likely to result if the Product Owner is not available during a Sprint? Discuss

- A. The Team extends the length of the Sprint until the Product Owner returns
- B. The Scrum Master assumes the responsibilities of the Product Owner.
- C. The Sprint is abnormally terminated
- **D. The product increment may not meet expectations**

33. Agile Manifesto says to value responding to change over following a plan. Which of the following statements illustrates this?

- A. Changes are accepted only if other features are removed from the backlog such that a fixed end-date is maintained
- B. Changes are accepted up until the point that the first Sprint begins. Then, changes are deferred to a future release
- **C. Changes are accepted at any time during the development effort depending on the business value of the change, the Product Owner's acceptance, and the ability of the team to respond in a timeframe acceptable to the Product Owner**
- D. Changes are accepted up until about halfway through the project, then all changes are deferred to a future release

34. What is the approach that Scrum encourages when a Team determines it will be difficult to deliver any value by the end of a Sprint? Discuss

- **A. Together with the Product Owner, focus on what can be done and identify a way to deliver something valuable at the end of each Sprint**
- B. Extend the Sprint by a few days to accommodate the extra work
- C. Suggest the Product Owner abnormally terminate the Sprint
- D. Immediately escalates to Senior Management.

35. What are the three components of an empirical process? Discuss

- **A. Transparency, inspection, and adaptation**
- B. Planning, committing, and measuring
- C. Feedback, courage, and simplicity
- D. Planning, taking action, and checking for quality

Scrum Master: Job Description

- **Coordinates project meetings, communicates status, readiness, business issues, technical issues, resource requirements and their resolution across functional organizations and management.**
- **Facilitates and organizes daily stand-up meetings, iteration sizing and planning, retrospectives, and demonstrations**
- **Collaborates with Product Owners, and Engineering leads to plan releases by fostering continued backlog grooming**
- **Tracks teams velocity and enables the team to plan accordingly**
- **Ensures the delivery teams are practicing agile methodology and software engineering best practices**
- **Identifies and removes any impediments or obstacles that prevents the delivery team from attaining sprint goals**
- **Reports schedule progress, team status and issues to external stake-holders**
- **Mentors and educates scrum team members in the practical benefits of agile processes**
- **Cultivates a team that recognizes the value of agile development and uses that recognition to improve**
- **Fosters a culture of continuous improvement, creativity, and innovation**
- **Aids the Scrum team with determining suitable commitments for stories, defects, and tasks**
- **Coordinates and schedules the execution of deliverables across multiple teams**
- **Protects time allocated to the Scrum team to explore innovative solutions to business and technical problems**
- **Business Acumen - Uses understanding of how the business works to make decisions. Draws accurate conclusions from technical, financial and other quantitative information.**
- **Communication - Conveys messages effectively using all methods**
- **Driving change -Anticipates the need for change in the business and acts as champion to drive it throughout the organization.**
- **Intellectual Agility – demonstrate the ability to quickly plan, do, measure and adjust without being encumbered by non-valued detail.**


BEING AGILE
STAY AGILE
THINK AGILE
GO BEYOND
INSIGHTS