Scrum interview - guidance questions and answers which helps in cracking interview. But real time experience always matters. Cracking interview is just a first baby step

**SCRUM INTERVIEW QUESTIONS**

1Q) Assume that 10 User Stories are agreed to delivery for your current sprint. You are in the middle of the sprint. PO asked to deliver a critical story along with current sprint delivery. What will you do?

Ans) We will accept the story but we will ask the PO to reconsider the scope as a new one is getting added. PO should suggest which story can be parked aside for time-being to accept the new one. Also, check with the Scrum Team on the complexity and dependencies of the newly yet to add requirement. If Scrum Team agrees it then it will be pulled in to the sprint.

Fewer time, the developers will finish the coding without descoping any other requirements. And again, if testers have enough capacity to test then only it will be accepted, if not PO should be informed.

2Q) What will you do if Scrum Team gives higher story points for a user story which you think is not a very difficult task and the dev is a senior guy and a difficult person. How will you make sure he gives a valid estimate?

Ans) As a scrum master I will check with the team whoever is attending the PBR on the reasons why it is of higher story points. And if they have some technical reasons and if it repeats for next sprint then definitely, I will understand is doing higher estimations and delivering less valued shippable product compared to the previous sprints.

I will arrange a catch-up discussion with the team and ask them if they have any challenges or do they really need any cultural trainings or skill polishing trainings or the team should still get matured on the domain and I will jar don the points and will try to motivate the team. If not immediately I can see the team will progress gradually.

This happened during the My Finance Community grooming’s. The team was over estimating but moving the stories to the walkthrough’s within couple of days. I have noticed this and asked the team to reconsider the estimations what they are giving and what they are burning and explained them the consequences of what happens when the stories are over estimated every time. And later the team understood the principle and were cautious while giving the estimations.

3Q) What is Velocity? What is the average velocity for your last sprint?

Ans) Velocity is the unit completed in that sprint. The unit is measured in terms of manhours or story points. The average Velocity of my last sprint was 90 points with developers 11.

4Q) What is a user story and how will you manage to guide a new joinee about User Story?

Ans) A User story is a short, simple description of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system. A user story is a technology and/or business requirement described in terms of the user, providing context as to what they want and why.

As a SM a workshop to be conducted to educate the team on importance of stories DoD and how to give relative estimates.

5Q) What is product backlog refinement? Can we estimate user stories in PBR meeting? What is your role in PBR meeting?

Ans) Product Backlog Refinement is an on going process and is needed within each iteration to refine items to be ready for future iterations. Key activities of PBR are:

• Splitting big items, creating and adding more details to PBI’s

• Detailing items until ready

• Estimating

• Prioritizing

It usually happens “mid-sprint””. Attendees include the PO, SMEs or Business Requestors and either all members of the team or a couple of representatives from multiple teams.

Split big items in to smaller items complying to the team DoR.

Estimate all non-estimated items in the backlog (including the newly created ones). One could use different techniques such as Planning Poker, Affinity Estimates etc.

Yes, estimations happen in PBR meeting.

As a SM I will facilitate the PBR meeting and I will check if the user stories are well mapped and have Acceptance Criteria, also will check with the team if they are providing right estimates.

6Q) What is your role as a SM in Retrospective meeting?

Ans) The crucial ceremony of Scrum is Retro meeting. In this meeting as a Scrum Master, I have to ensure all the team is participating actively by giving their inputs on how the Sprint has been through. Also, I have to take voting’s on improvements or suggestions given by the team and assign it to a volunteer for getting it resolved. Also, I have to ensure that the team enjoys the retro by conducting fun retros and also concentrating on what went good, bad and suggestions part.

Whatever improvements suggested in the previous Retro should be discussed again to let the team know the status or progress of those.

7Q) Tell me some 3-4 impediments you faced and how you resolved it?

Ans) 1) When the developer is having a query on the story given by the PO and PO is on leave and no one around to clear the developer’s query. I checked directly with PO head and after 1 day I got the response.

2) Recently, the QE has faced an issue for testing a logic. The code has been moved in to the Integration environment but tester couldn’t cover all the scenarios due the needed quantity of the test data. We need an environment in which we have more than 10k+ records and when I have sat with the tester and when we have written a query in the DB for all the integrated environments then we came to know that the Pre-Prod environment is having the required quantity of data.

Immediately, as a SM I sent an email to the deployment team asking if we can merge our code base to the environment for testing. And we got a response saying that pre-prod environment is used for hyper care release and can be available after couple of days. After couple of days, I have followed up and got confirmation that we can merge the code, I informed the developer to do the merging and once done the dev informed the tester to test it.

3) Fewer time due to multiple teams working on the same product the environment goes down. Then I will inform the other team saying the environment is down due to the changes they have merged and in meantime our tester will raise a defect. I will keep a track on it and will inform the other team to let us know once it is resolved as it is an obstacle for the testers to proceed in testing. Once the team resolves then I will update the team to continue their testing.

4) Fewer times, when the defect is raised by QE and assigned to developer to resolve, the developer says that the defect already exists in prod and it is not due to their code changes. I will inform BA to do analysis on that and to confirm. When BA informed me about the status, I scheduled a quick catch-up with Dev, Tester, BA & PO to decide on final outcome whether to create a PBI under BAU or under project or to resolve it as a defect.

8Q) What are Burn up and Burn Down Charts?

Ans) A Burndown chart shows the amount of work remaining (the remaining effort) where as a Burn up chart shows how much work has been completed. A burn up chart however shows more details showing both total works achieved and the work done in previous increments.

9Q) Let’s assume you have 2 weeks of sprint and till the 8th day your team has finished only 30% of work. What will be your role as a SM to get it completed 100%. Or how you will manage to communicate with stake holder if not finished 100% by 10th day?

Ans) In the Daily Stand-up I inform the team to move the status of the stories correctly. If delays happen that were mostly due to UAT pending. As a scrum master I will remind Pos frequently how important it is to move the status of the stories to Done if UAT is completed. And also I used to remind Pos to start UAT and close the stories.

Later, the Pos started to assign the POCs from support team to perform UAT and to support them in closing the stories. This way I always ensure that the stories will be flowing on time. And in case the developers or testers cannot finish their part of work then I always remind them to inform it in advance, so that we can inform the business stakeholders about the deliverables. Ans) In DS meeting, based on the estimated story points given by dev and testers, I will ask if there are any blockers or issues that’s causing delay for them in moving the status. Mostly, I will check personally so that the team shouldn’t think I am micro managing them. This way they will open up and I will assist them in getting it resolved or they themselves will resolve by communicating with each other. And in next day’s stand-up I will definitely cross check if that has been resolved or need assistance from anyone. If yes, then cool. If no, then set-up a quick catch-up with that dev or tester and techlead to understand the problem. Mostly, team will resolve on their own by having a quick discussion among themselves until unless it is a serious issue in which Pos to be included.

10Q) One of your Dev team member is not performing up the mark since last 2-3 sprints, how you will overcome this problem?

Ans) During the 1st sprint itself I will check with that team member on why he/she is unable to deliver the story. I will have a one-one discussion to check if there are any technical issues or personal issues. I will try to support him/her saying that if there are any technical issues then team is here to help and support. If it is personal issue, we have our organisation provided helplines to assist in helping. And if still the issue persists then I will discuss with PO and try to assign low priority easy stories and build confidence in him/her and will inform management that he/she is trying to mature him/herself and will start delivering the valuable outcome soon.

11Q) What all points you keep in mind while creating User Story?

Ans) User stories will be created by BA or PO. In very rare scenarios a SM will create and should enter the required fields information and details.

12Q) What are your roles, responsibilities and interaction with PO?

Ans) A Scrum Master should work collaboratively with PO and assist in the following:

• Backlog prioritisation

• Checking if backlog is having fine grained items at the top and coarse-grained items at the bottom

• Guide the PO in case the story is missing the Acceptance Criteria

• To verify if the backlog contains the newly suggested requirements from the business stakeholders

13Q) How you estimate User Stories?

Ans) User Stories are estimated in PBR or grooming sessions. We use Fibonacci series or affinity estimates to estimate a user story effort. Estimates are provided by the scrum team.

14Q) What is Sprint Backlog? What is DoD and DoR?

Ans) Sprint Backlog contains the stories which meets the DoR and is owned by the Scrum Team. It contains those items from Product Backlog which the team will work in the next Sprint. It also contains agreed sprint goal.

Definition of Done: Build the thing right (quality)

• Code completed and reviewed

• Unit tests written and passing

• Static analysis OK

• Integration tested

• Code coverage > 70%

• Documented as needed

• PO accepted

In the Sprint Planning meeting, when Developers are identifying tasks (on How-part breaking stories), encourage to keep close on eye on the DoD built by the team. Also encourage your team to embrace for wider DoD in your Retros.

DoR – Definition of Ready – Requirements defined clearly enough that all members of the team understand what to be done.

Includes clear statement of resulting business value that allows PO to prioritise.

Includes any required enabling specs, wireframes etc.

Fully meet INVEST criteria for user stories

Free from external dependencies

Sample DoR:

Requirements clearly understood by the team

Scope is sufficiently clear

“Sufficiently”= able to break-up in technical tasks to do

Size = small enough ex. 1/4th of a Sprint

Meet the INVEST characteristics

No technical orientation

Dependencies (if any) cleared out

Assumptions list that are fulfilled

15Q) What is Sprint Review meeting and what is your role in it?

Ans) Sprint Review meeting is Sprint Demo given by the Scrum Team to the business stakeholders, POs, Architect and Manager, showcasing on what shippable product is ready for deployment. 16Q) Tell us about a day at work is for you?

Ans) My day starts with emails checking which could be the scrum team tagging me in comments of a User Story or defect in jira or could be from other squad or can be from manager. Based on the content I will take action accordingly. Then will run the stand-up meeting and if team needs any guidance on impediments then I will support them, if not I will check with PO’s and BA if to arrange for a PBR meeting and with testers regarding defect triage meeting. I will keep an eye on product backlog and its healthiness. This way my day starts and ends.

Everyday will not be the same as a Srum Master. It will be differing based on the issues and action items to be taken.

17Q) Can a SAFE SM conveniently play the role of a Release Train Engineer?

Ans) As you get hold on ART, and if you are part of multiple Product Increments, then yes one can easily play the role of RTE. You can treat this as Sr SM role and got to facilitate SoS along with PI planning ceremonies. SAFE SM with relevant experience will be ready to take up RTE roles.

18Q) Difference between Agile Coach & a Scrum Master, please explain?

Ans) It is true that Scrum Masters and Agile Coaches do similar things, however at different levels. It is also true, as you state, that Coaches to get a wider mandate, not only to coach the executives but SM and teams as well.

The biggest differences are what you are expected to do. A Scrum Master works with “A” team. An Agile Coach works with ALL teams, AND executives AND other teams/groups. A Scrum Master ensures that the team is following the Scrum process, doing the ceremonies and behaving the right way. 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).

The main difference is the level that the two are operating, single team or enterprise.

19Q) What can you do as a Scrum Master?

Ans)

• Support CD improvement initiatives

• Accept that it doesn’t come free

• Propose to the team to have very visible testing reports (all levels)

• Propose quality indicators that should increase sprint-by-sprint.

• Respect the scrum framework, it goes hand-in-hand with CD

• Raise awareness (Company, Pos…)

20Q) Impediments – The scenario I am thinking is an unforeseen and undesired changes in team composition. In a ST, a tester leaves the project and this becomes a blockage. What should be done in this circumstance. How will SM remove this impediment. Can SM help the team by doing the tester role for that sprint or should SM discuss with the ST.

Ans) In this scenario, SM should raise a concern to the manager asking to share a tester from other squad for helping on that story.

21Q) This new team has over committed stories in the sprint planning, they didn’t take the holiday (Jan 26th) into consideration, one person’s father impacted with Covid unfortunately and he went on emergency leave. PO already gave commitments to stakeholders and already invited them for review on Wednesday. How do you handle this situation?

Ans) SM should come up with the team availability including Planned Leaves, company holiday during the sprint planning. Team can select the story from prioritized list by keeping the availability in mind. Unplanned leaves are natural. During daily scrum this point has to discuss and see how to take it forward. In case few user stories are not getting completed because of unplanned leaves, it has to be communicated to stake holders in advance. By doing this we can have scrum values actually implemented.

We usually plan 75% Bandwidth for committed cards for the sprint and rest of the bandwidth for Tech Debts (purely developers initiative for continuous learning). As per contingency plan, team will start filling the gap, keeping tech debts aside. Tech Debt – Word itself depicts we (team) is liable to optimize product value. How do we optimize and what needs to optimize – will be delivered from brainstorming sessions (with smaller audience like Arch + Dev Teams). Lets take ex. Lets ship the product now and deal with consequences. And, all the consequences might turn as Tech Debts (Design, Creepy code, Architecture). Any shortcuts, quick solutions etc. Product maintenance will become difficult if we continue like this. If we talk about tech debt, we need to consider the trade-off between scope and schedule. Whether to get more later or less sooner.

For this blocker, my approach would be to reduce the capacity of the team members, as it was a holiday, but it may happen if team members take leaves. When capacity reduces, everyone will be in red zone. Now here we can have two scenarios:

• Ask the team members if the tasks of that person can be shared by others and if they can complete. Based on positive feedback, we can move ahead with the review as scheduled.

• If they are not confident of completing the tasks and not to take ownership of the person’s task, then this needs to be shared with PM as considering PO is aware. If PM can bring in another resource, we can go ahead as planned. If not, then PO/P needs to review the dates with stake holders and explain them the situation and consider next release. Being an SM we need to take care of Teams as well as business needs of the Org. When all the options burn out, that’s the only solution I feel.

22Q) Do you have experience of software development, delivering functional products? If yes, please provide a brief overview?

Ans) I worked as a mainframe developer in waterfall methodology. During those times we used to receive functional specifications from client and I used to prepare Analysis document, high level estimation, low level estimation, High level & Low level design documents, test cases with test outcomes document.

23Q) In a product-centred environment, how your approach would differ for a product vs a project?

Ans) In Product-centred environment, the organisations are driven by the desire to focus their attention on building and bringing the products to the market rather than the customers that purchase their products.

24Q) Please can you describe the most challenging problem you have faced within technical delivery. Why was it difficult, what did you do, and what did you learn from it?

Ans) I would like to tell you about a time where I was working for a XX bank as a PM/SM in Hybrid role and our team was responsible for migrating various communication templates and also decommissioning legacy systems and replacing with new solution.

In middle of the sprint while our DB engineer were migrating data from an old 32bit ODBC to a 64-bit system. They have noticed that there were a lot of inconsistencies and data was also getting truncated. This was not foreseen at the time of estimating the US and caused a big blocker, I was made aware of this issue, so I have carefully conducted a deep dive and root cause analysis to establish what went wrong and to determine corrective actions I arranged a brainstorming session with Devs, and one of the Dev proposed a solution of acquiring a paid Addon which is effective and can handle these 32 bits to 64-bit migration effectively.

I have communicated these corrective actions, alternative delivery approaches and risks to get the project back on track, since I have always had contingency plans/funds available I proposed the above solution, got the stakeholder approval for the revised delivery plan and risk profile.

I was able to get the funds sanctioned, the addon did its job and we were able to successfully migrate the legacy 32-bit DB to 64-bit solution.

25Q) For requirements traceability, we have purchased test rail plugin for Jira software. I’m planning to setup a demo project and see how the flow works. Any suggestions which Jira board (Kanban or Scrum) should be ideal to start with.

Ans) If it is BAU support related User stories then Kanban, if project funded use Scrum board.

26Q) What do we exactly mean by POD in Agile or is it something with DevOps?

Ans) PODs in Agile means small agile teams may be a customized team that can be allocated for a portion of backlog items. A set of items assigned to the PODs and they need to design, develop test build and release. For ex. Set of ControlM jobs development for generating Tableau reports. Here the team need to work on development of control jobs for the Tableau reports where data, tables and even reports may be already available.

27Q) PO written user stories and called SM to help in prioritizing the stories. How will you help PO as a SM in prioritizing?

Ans) As per scrum values & guide we need to be clear on the roles and responsibilities. And also, SM can educate ideally on priority techniques. Yes, agreed prioritization is Pos responsibility, but fewer times when every requirement is on priority then Pos will provide weightage by considering HLE for which SM will facilitate a quick catchup meeting with Tech Lead and QE Lead to understand technical dependencies, criticality, efforts. And also, SM helps PO with efficient backlog management using many techniques.

28Q) How do you manage dependencies and bring transparency to dependencies?

Ans) When multiple teams are there, there are high chances of dependencies or within a single backlog there could be items which blocks or being blocked by others.

29Q) What is the basic difference/advantage of scrum and Kanban?

Ans) If you are working in sprint mode then Scrum otherwise Kanban. If you have planned out your work in definite timelines use Scrum else Kanban.

Scrum is iterative delivery approach and continuous flow is Kanban.

SCRUM KANBAN

Time boxed by means of sprints Not enforced any timelines

Minimum three roles (PO,SM,Dev Team) No predefined roles

Scoping until sprint start Can be modified

Velocity is calculated based sprints delivery Velocity can be calculated by total number of story point delivered for defined time

Scheduled estimates based on number of sprints No pre-defined estimates

Work delivered in batch at the end of sprint Work delivered, each time the story is delivered30Q) What is the meaning of Control of scope and risks within Agile/SAFE framework?

Ans) In agile-speak, scope definition is demonstrated as user stories — also known as high-level requirements — in the product backlog. These user stories are prioritized based on factors like business value, complexity, and cost; and worked upon incrementally in sprints.

Control of scope is the process of monitoring the status of the project and product scope and managing changes to the scope baseline. The key benefit of this process is that it allows the scope baseline to be maintained throughout the project.

Control of Risk is the process of identifying risks, analysing risks, prioritising risks, creating action plans (responses) to deal with high priority risks, continuous monitoring and follow-up to ensure the action plan is mitigating the risk.

Risk is defined as an uncertain event or set of events that can affect the objectives of a project and may contribute to its success or failure.

31Q) What is SonarQube (code quality tool)?

Ans) This is commonly used for code quality/security.

• Static analysis of the code

• Rating in different areas

1. Reliability

2. Security

3. Maintainability

• Reports on

1. Unit Test Coverage

2. Code Duplication

3. Complexity

4. Documentation

31Q) What is Product Goal?

Ans) The Product Goal describes a future state of the product which can serve as a target for the Scrum Team to plan against. The Product Goal is Product Backlog. The rest of the Product Backlog emerges to define “what “will fulfill the Product Goal.

32Q) What is Staging?

Ans) Staging environment is like a mirror image or a close replica of product environment, everything in staging environment is almost a close copy of actual Prod environment.

33Q) What is the difference between Capacity & Velocity?

Ans) In general, both are counted for a sprint irrespective of its duration. Capacity is the team availability (in terms of hours or days or any time measurement unit) for a particular sprint.

Velocity is the story points that team can deliver in a given sprint. Completed number of story points (completed velocity) and velocity are interchangeably used. So there are committed velocity and completed velocity. Usually completed velocity <= Committed velocity. In some cases, if team does extremely well then completed velocity can be greater than committed velocity.

34Q) What is the biggest challenge faced by you in your Scrum team? How did you resolve it?

Ans) When we were working for one of the most demanded medium sized project BID in Podium the challenge was we have to finish the MVP in 3 months and then to add additional features to the delivered MVP. The MVP itself was of 300+ points with the new Lightning Web Components of SF technology which team never worked on it. Also we decided to use the SmartComm services to generate the compliance documents to which as well the team was new.

So here there were 2 risks – Skill Risk, Tight Deadlines

So, along with the Engineering Manager I have sat down to understand the roadmap and to guide the team according to that.

I have facilitated the grooming sessions and shaping sessions everyday twice and thrice as the piece of work got UI changes, Document Changes and SmartComm changes as well. And there was impact on integration with Salestools as well, where I have gathered the people from other team as well to discuss about the impacts, they will have due to the changes we are doing. That coordination was little tough as they got their own priorities apart from the work what we had and also getting the PO’s time and collaboration initially was very challenging as the PO’s were busy in getting the prototype mocked-up with the help of Ux designer and also, they were busy in conducting workshops with the business stakeholders in getting the feedback on the prototype built continuously.

The locking of sprint scope became so challenging as we have to deliver the MVP in 3 months and we had set of 200+ stories to be addressed and all of high priority.

I along with Tech lead helped PO’s in maintain the backlog for letting us know the next prioritisation. And me and my manager has motivated the team that if we can crack this MVP this will be an achievement in our career life. And also, the request was to stretch ourselves (but it was not a force) and it was in Covid lockdown, we were not supposed to come to office, the way we worked and the collaboration and communication among the team became so challenging. The challenge was on the availability.

In very fast pace we adjusted ourselves to the situation and created a group in teams where in they can share their issues, impediments, queries on spot and get it resolved on spot with follow-ups with PO’s.

Also, the SmartComm work coordination was too challenging but as a team we made it and our CRM has been awarded as best business platform by the Aus business awards which is a great achievement to the whole team.

35Q) How do you measure your success as a Scrum Master?

Ans) I measure based on how the team is performing, healthy mindset of the team, the business value what as a team we are delivering, customer satisfaction, management satisfaction, how we are achieving the sprint goal

36Q) What are the outcomes from your past retrospective meetings? How did you work with these outcomes?

Ans) I feel personally as a SM that Retro meeting is the crucial ceremony of Scrum. This is the place where team can share their good, bad experiences and suggestions or improvements for better serving the customers. This is the platform where team can be identified and can appreciate each other for the good work they performed. Also, they can share the hindrances or issues they have underwent and how they resolved among themselves.

Fewer times, there will be some suggestions or improvements team provide for which I will request someone from the team to volunteer it until it gets resolved.

To say few of the examples are:

 Initially when the BID project started it was hectic to Pos and BA to capture all the requirements in one go. And though the stories were not meeting the DoR still we had to move those in to the sprint due to the deadly timelines and due to the large team with only one BA availability. And, when team started raising the same concern in more than 2 retros then as a SM & BA I took utmost care in drafting stories and analysing it and getting the Acceptance Criteria. This was action on me which I have taken and cross verified with the team till we have delivered the MVP.

 There was a concern raised on the availability of the team in this Covid situation and communication gaps. The scenario was if SF developer makes any changes on the formula fields which were newly introduced then that to be communicated to the SmartCoom developers as well, as they will follow the same formulas to produce the details in the reports. But due to the knowledge gap and the impacts the communication was lacking between the developers. Later, as a SM/BA whenever PO gives a feedback on the formula changes I used to communicate the same in the email and also in the SmartComm group so that everyone will be on the same page.

 Once, the team has raised a concern on domain knowledge sharing. As team was getting busy with their own work and they were not getting time to check on the logics and changes what their co-peers were working on and some how among developers and testers the bridge on domain knowledge was not persistent. Hence, I started arranging the CoP sessions in which a team member should volunteer themselves to share what they have learnt.

37Q) What are the sources of learning do you use? What are your recent learnings?

Ans) I use Plural Sight, Linked In Learning, Youtube for learnings. Recently have learnt SAFe Agilist, Abinitio, Tableau, SF Admin course, APIs. Actually, I always keep on learning to keep my brain active.

38Q) How do you deal if requirements change frequently?

Ans) This issue became very common in Agile-Scrum framework – the changing requirements. As a SM though I follow strictly with PO’s about the DoR, fewer times what happens is when the story is pulled in the Sprint and when Sprint is started, the PO comes up the other day saying there is a change in the requirement as we received the feedback from the customers just before though we have sent them very early.

I keep on checking with PO’s that the requirements what they are sharing with us are approved ones. And I will explain them why I am checking on this every time in Sprint Planning, as we are committing to a sprint goal and the changes to the requirements should not cause a scope creep and shouldn’t impact the sprint goal.

39Q) How do you know that your work is effective and efficient as a Scrum Master?

Ans) By checking on Velocity Report, Burndown chart and based on the business value the team is delivering, based on customer satisfaction levels, based on feedback from management on team’s success, based on how agile we are.

40Q) How do you influence and motivate your team? Give an example?

Ans) As a Scrum Master, I have to ensure the team that I am always there to listen to my team. I should be checking in the Daily Stand-up if team is having any impediments or blockers or concerns that to be raised to the management or in the SoS meeting.

By giving the team the opportunity & time to learn, to be self-organised, by providing feedbacks, by appreciating their hard work, by considering the feedback given by the team on SM, by helping the team to resolve impediments and blockers, by actioning on the suggestions provided in the Retros.

41Q) PO converts requirements in to tickets and asks them to estimate agree or disagree?

Ans) This should not happen as tickets are for BAU or incidents raised by end users. And if there will be functionality change as per these incidents then and there as well these will be considered as BAU.

Fewer times, PO’s will identify the issues in production before identified by end users and in such scenarios, we will log those production gaps as BAU tickets.

Also, few times when the project is not having funding and management decides to add some of the new requirements to be part of BAU bucket, so that time we all agree to do so.

42Q) Received a story on last day of the sprint to test and defects are found in it. Will you mark it as done?

Ans) Actually, we follow a process of code lock down 1 week prior to the deployment, and the dev team should handover their stories to testers before this code lockdown, so that if any defects are raised those will get resolved well in advance before lockdown.

But, fewer times this is not achievable. What happens is the dev will consume more time in developing a story due to many reasons (to say, could be due to criticality, due to technical challenges or due to uncertainties), in such cases the tester will receive the story just before the code lock down or after code lock down (in this case the team should decide on if the story can be part of changeset and also should be discussed with PO’s about the risks in case more defects are raised). If the team raises the defects on that late handed over story by dev to test then as a SM I will arrange a catch up with PO to check the impact of those defects on the functionality we will be delivering. If the PO approves we can go ahead with the raised defects and the defects to be solved in the new Sprint then the story will be marked as done and defects will be spilled over to the next sprint. It is a good practice to close all the defects related to the story and move the story to done, but practically fewer times we may not achieve this.

43Q) Why User Stories not estimated in man hours?

Ans) Because hours are difficult to come to conclusion. Jr dev can do in 5 days, but Sr Dev can do in 1 day. Complexity will be an easy factor to come to an agreement with. That why we use fib series or relative estimation.

44Q) What is build breaker?

Ans) An inadvertent mistake by a software developer that sometimes stops the build process, or causes unacceptable warnings , and/ or failures in the automated test environments, is known as a 'Build Breaker'. The onus on the developer then, is to get the build to normal as soon as possible.

45Q) How will you form a Scrum Team?

Ans) I never got a chance to form a new Scrum Team, but I can check on few things like how far my team is agile. I will conduct workshops to guide them on how to be agile and how to follow agile principles. Few workshops that can be conducted are on Relative estimates, meeting DoR & DoD.

Assisting Pos on how important it is to maintain healthy product backlog.

46Q) Are personas can be used to understand human behaviour?

Ans) Personas are fictional characters. You create personas based on your research to help you understand your users' needs, experiences, behaviours and goals. Creating personas will help you identify with and understand the user you're designing for.

A user persona is a semi-fictional character created to represent different customer types that use a company's products or services. User personas are profiles of imagined individuals that reflect a business' core customer base.

47Q) Is continuous integration important?

Ans) Continuous integration, or just CI, is a software development practice of integrating code continuously (at least once a day — per developer), and in an automated way. Also, it is about verifying if the new code you just wrote broke or not the code that was already working, since the automated tests and other tasks (like syntax verification) are executed when integrating the code.

Using this approach allows software development teams to have a very fast feedback loop about the changes they are doing in a specific application, and this is a cheaper way of solving issues when they are found, because the just changed code is still fresh on the developers’ mind.

This is also one of the practices from the eXtreme programming discipline

One of the important points of using CI is about having less conflicts when integrating code. Once the code is frequently merged (from an specific branch, for example, to the trunk branch), it has less chances of breaking what already exists. And even if it breaks what was already working, it is easier to solve.

Another very important thing when talking about CI is that it needs to be supported by a suite of automated tests (not only unit tests, but also by integration tests, and even better, if possible, by end-to-end tests).

48Q) What are the characteristics of a good requirement?

Ans)

 Unambiguous - There should be only one way to interpret the requirement. Sometimes ambiguity is introduced by undefined acronyms:

REQ1 The system shall be implemented using ASP.

Does ASP mean Active Server Pages or Application Service Provider? To fix this, we can mention a full name and provide an acronym in parentheses:

REQ1 The system shall be implemented using Active Server Pages (ASP).

Here’s another example:

REQ1 The system shall not accept passwords longer than 15 characters.

It is not clear what the system is supposed to do:

• The system shall not let the user enter more than 15 characters.

• The system shall truncate the entered string to 15 characters.

• The system shall display an error message if the user enters more than 15 characters.

The corrected requirement reflects the clarification:

REQ1 The system shall not accept passwords longer than 15 characters. If the user enters more than 15 characters while choosing the password, an error message shall ask the user to correct it.

Some ambiguity may be introduced through the placement of a certain word:

REQ1 On the “Stored Flight” screen, the user can only view one record.

Does this mean that the user can “only view,” not delete or update, or does it mean that the user can view only one record, not two or three?

One way to fix the problem is to rewrite the requirement from the system’s point of view:

REQ1 On the “Stored Flight” screen, the system shall display only one flight.

 Testable (verifiable) - Testers should be able to verify whether the requirement is implemented correctly. The test should either pass or fail. To be testable, requirements should be clear, precise, and unambiguous. Some words can make a requirement untestable [LUD05]:

• Some adjectives: robust, safe, accurate, effective, efficient, expandable, flexible, maintainable, reliable, user-friendly, adequate

• Some adverbs and adverbial phrases: quickly, safely, in a timely manner

• Nonspecific words or acronyms: etc., and/or, TBD

Such a requirement might look something like this:

• REQ1 The search facility should allow the user to find a reservation based on Last Name, Date, etc.

In this requirement, all search criteria should be explicitly listed. The designer and developer cannot guess what the user means by “etc.”

Other problems can be introduced by ambiguous words or phrasing:

• Modifying phrases: as appropriate, as required, if necessary, shall be considered

• Vague words: manage, handle

• Passive voice: the subject of the sentence receives the action of the verb rather than performing it

• REQ1 The airport code shall be entered by the user.

• REQ2 The airport code shall be entered.

The first example shows a classic example of passive voice. In active voice it would read “The user shall enter the airport code.” As the second example shows, another result of the use of passive voice is that the agent performing the action is sometimes omitted. Who should enter this code—the system or the user?

• Indefinite pronouns: few, many, most, much, several, any, anybody, anything, some, somebody, someone, etc.

• REQ1 The system shall resist concurrent usage by many users.

What number should be considered “many”—10, 100, 1,000?

 Clear (concise, terse, simple, precise) - Requirements should not contain unnecessary verbiage or information. They should be stated clearly and simply:

• REQ1 Sometimes the user will enter Airport Code, which the system will understand, but sometimes the closest city may replace it, so the user does not need to know what the airport code is, and it will still be understood by the system.

This sentence may be replaced by a simpler one:

• REQ1 The system shall identify the airport based on either an Airport Code or a City Name.

 Correct - If a requirement contains facts, these facts should be true:

• REQ1 Car rental prices shall show all applicable taxes (including 6% state tax).

The tax depends on the state, so the provided 6% figure is incorrect.

 Understandable - Requirements should be grammatically correct and written in a consistent style. Standard conventions should be used. The word “shall” should be used instead of “will,” “must,” or “may.”

Feasible (realistic, possible) - The requirement should be doable within existing constraints such as time, money, and available resources:

• REQ1 The system shall have a natural language interface that will understand commands given in English language.

This requirement may be not feasible within a short span of development time.

Independent - To understand the requirement, there should not be a need to know any other requirement:

• REQ1 The list of available flights shall include flight numbers, departure time, and arrival time for every leg of a flight.

• REQ2 It should be sorted by price.

The word “It” in the second sentence refers to the previous requirement. However, if the order of the requirements changes, this requirement will not be understandable.

 Atomic - The requirement should contain a single traceable element:

REQ1 The system shall provide the opportunity to book the flight, purchase a ticket, reserve a hotel room, reserve a car, and provide information about attractions.

This requirement combines five atomic requirements, which makes traceability very difficult. Sentences including the words “and” or “but” should be reviewed to see if they can be broken into atomic requirements.

Besides these criteria for individual requirements, three criteria apply to the set of requirements. The set should be

Consistent - There should not be any conflicts between the requirements. Conflicts may be direct or indirect. Direct conflicts occur when, in the same situation, different behavior is expected:

• REQ1 Dates shall be displayed in the mm/dd/yyyy format.

• REQ2 Dates shall be displayed in the dd/mm/yyyy format.

Sometimes it is possible to resolve the conflict by analyzing the conditions under which the requirement takes place. For example, if REQ1 was submitted by an American user and REQ2 by a French user, the preceding requirements may be rewritten as follows:

• REQ1 For users in the U.S., dates shall be displayed in the mm/dd/yyyy format.

• REQ2 For users in France, dates shall be displayed in the dd/mm/yyyy format.

This can eventually lead to the following requirement:

• REQ3 Dates shall be displayed based on the format defined in the user’s web browser.

Another example of a direct conflict can be seen in these two requirements:

• REQ1 Payment by PayPal shall be available.

• REQ2 Only credit card payments shall be accepted.

In this case the conflict cannot be resolved by adding conditions, so one of the requirements should be changed or removed.

Indirect conflict occurs when requirements do not describe the same functionality, but it is not possible to fulfill both requirements at the same time:

• REQ1 System should have a natural language interface.

• REQ2 System shall be developed in three months.

Some requirements do not conflict, but they use inconsistent terminology:

• REQ1 For outbound and inbound flights, the user shall be able to compare flight prices from other, nearby airports.

• REQ2 The outbound and return flights shall be sorted by the smallest number of stops.

To describe the same concept, in the first requirement the term “inbound flights” is used, and in the second requirement the term “return flights” is used. The usage should be consistent.

Nonredundant - Each requirement should be expressed only once and should not overlap with another requirement:

• REQ1 A calendar shall be available to help with entering the flight date.

• REQ2 The system shall display a pop-up calendar when entering any date.

The first requirement (related to only the flight date) is a subset of the second one (related to any date entered by the user).

Complete - A requirement should be specified for all conditions that can occur:

• REQ1 A destination country does not need to be displayed for flights within the U.S.

• REQ2 For overseas flights, the system shall display a destination country.

What about flights to Canada and Mexico? They are neither “within the U.S.” nor“overseas.”

All applicable requirements should be specified. This is the toughest condition to be checked. There is really no way to be sure that all the requirements are captured and that one week before the production date one of the stakeholders won’t say, “I forgot to mention that I need one more feature in the application.”

A good requirement should have more criteria. However, they usually can be expressed as a combination of the criteria we have just discussed:

• Modifiable: If it is atomic and nonredundant, it is usually modifiable.

• Traceable: If it is atomic and has a unique ID, it is usually traceable.

49Q) How to manage conflicts between two stakeholders during the requirement gathering?

Ans) If two different stakeholders got 2 different big piece of requirements which to be addressed in the upcoming sprint and if the team’s capacity can support only either of 1 requirement, then as a SM set up a meeting to discuss the project/business goals. Avoid taking an approach of telling them in the meeting what the goals are, but instead ask the conflicting stakeholders/groups to reiterate the higher-level goals themselves. It will quickly become apparent that not everyone is on the same page.

50Q) How do you manage risks as a Scrum Master?

Ans) A simple approach for Risk Management in Scrum

1. Identifying the risk.

2. Analysing each risk to determine its exposure (severity of impact)

3. Prioritizing the identified risks based on their exposure.

4. Creating action plans (responses) to deal with the high-priority risks.

5. Continuous monitoring and follow-up to ensure that your action plans are mitigating the risks.

51Q) Daily Scrum Dos and DOnts?

Ans) DO

• Same Place, same Time

• Meet where the work happens

• Move through work items from highest to lowest priority

• Keep it short and focused

• Keep the energy level up

• Track any blockers, emergency items or items that are stuck

DON’T

• It is not an event to solve the problem

• It is not a event to catch-up on non-project related activities

• It is not a status meeting

• It is not event to share “HOW”the work is done

• It is not event for a story telling

• Don’t wait for this event to share any impediments

52Q) What is Empathy Map work?

Ans) Empathy mapping is understanding more about user feelings/emotions. How end users are feeling about our product, what do they hear, what are they saying in the market. This can be used in initial phases of project like designing Ux requirements.

53Q) What is vanilla scrum?

Ans) Basic scrum is vanilla scrum and can add visual boards Kanban and add new engineering practices. As team learns the framework gets evolved, we can add new wrappers to it.

54Q) Which two things are appropriate for a scrum master to do, if the scrum team does not have the tools and environment to completely finish each selected Product Backlog item?

a. Declare the ST not ready for Scrum

b. Have the ST establish a DoD that is actually possible to achieve given current circumstances

c. Refocus the current Sprint on establishing the ST’s environment instead of delivering an increment

d. Encourage the PO to accept partially done increments until the situation improves

e. Coach the Scrum Team to improve its skills, tools and environment over time and adjust the DoD accordingly

Ans) B & E

55Q) What reports do you pull at the end of the sprint?

Ans) Velocity Chart, Burndown chart, Cumulative workflow, Burnup chart

56Q) What is capacity?

Ans) Capacity is the team availability (in terms of hours or days or any time measurement unit) for a particular sprint

57Q) What are Sprint Metrics?

Ans) At the outset of the sprint, the team forecasts how much work they can complete during a sprint. A sprint burndown report then tracks the completion of work throughout the sprint. The x-axis represents time, and the y-axis refers to the amount of work left to complete, measured in either story points or hours.

58Q) Different techniques for writing User Stories?

Ans) User Stories should follow INVEST rule followed with Acceptance Criteria. It should be written from the point of view of the person who will be using the product. It should be clear and detailed enough to understand easily with screenshots for representing before and after changes.

59Q) What is Retro?

Ans) The retrospective session is basically an “improvement” meeting held to find ways and means to identify potential pitfalls, past mistakes, and seek out new ways to avoid those mistakes, which are attended by all – the product owner, scrum master, development team members, and optionally with the stakeholders.

60Q) Technology used in your project?

Ans) Salesforce, Salestools, Devops, Azure

61Q) What is the role of a SM during a Daily Scrum?

a. Lead the event and give orders to the developers

b. If he is actively working on the pending Sprint work items, he participates as a developer.

c. Allow the PO to participate when he/she deems it necessary

d. Mention the problems being experienced during the Sprint in order to improve them

Ans) C

62Q) What is difference between an impediment and a blocker?

Ans) An impediment can be by-passed, but a Blocker cannot be by-passed

A blocker is actually stopping work for continuing for a work item. An impediment could be slowing down or impeding work on a work item.

Impediments can slow down the work. If that is not resolved that may lead to a blocker. Blocker is something which you cannot proceed further until you resolve this.

63Q) How much time normally it takes in PBR and is it also a time boxed?

Ans) According to scrum guide it says 10% of Sprint done. However, in reality we can consider as 1 hour or 2 hours based on the content and complexity. The main intention of PBR is to refine the stories until the stories meet the DoD by the development team. If it is a long PBR then we can divide it in to 2 or 3 meetings

64Q) What is Sprint zero?

Ans) It is more of readiness. It is purely meant for Sprint Planning and Pre-activities like environment setup, resource upskilling etc.

Sometimes we use Sprint zero for even getting SOW and stakeholder sign offs for start of project.

We get budget approvals for projects in Sprint zero phase and that’s call Statement of Work including what’s in scope at high level. Each organisation can tailor this as per their process.

Ans) Sprint zero is there to cover activities such as product backlog creation, infrastructure set-up, architectural planning, resourcing the team and test plan composition. Along with prototyping, design planning and test validation.

An ex. Of Sprint 0 considerations include:

How do you know what technologies you are going to use?

How are you going to structure a product and the team around it?

Ans) Sprint 0 COVERAGE:

• Team Structure – Team onboarding, roles & responsibilities of both client and partners

• Backlog management – Story creation, Acceptance Criteria, Story sizing & story splitting

• Trainings – Functional, Technical & Agile

• Process – Collaboration tool configuration i.e. Jira/TFS/Rally, Communication tools – Webex/Video Conference setups among team members.

Agile Delivery approach – Sprint cadence, Metrics, Governance, Quality Practices

Test Strategy & Development Strategy

• Environment – Development Env readiness, testing env readiness

• Access – Application access to the team

65Q) How as a SM you help the team to mature their DoD?

Ans) As a SM, I guide the team on following the code reviews, code quality, code standards, code comments, testing of units, integration testing, documentation, UAT, defects approval by PO, if all these are met the DoD is met. I remind the team on regular intervals to follow these guidelines.

66Q) Is any one following the concept of adding story points to sub-tasks?

Ans) Latest practice is adding hours to subtasks, story points to stories, T-shirt sizing to features & epics. Will the hours reflect in burn down? And if sub-tasks are moved to done then story points in original story will get deducted automatically? For original estimates yes time-tracking will be adjusted automatically. In Sprint Planning itself after estimating the story points then the team will add the sub-tasks to the stories and estimates the sub-tasks in to hours as they have clarity on what is required and these number of hours will be considered against the capacity and which will stand as input for the Sprint Planning. It is auto built on how the hours are auto-adjusted from story points when sub-task is moved to done. How will we able to finally move a US to Done if 90% of its sub-task is only done and the remaining 10% is dependent on a subtask in another user story or sprint?

67Q) What is scope creep?

Ans) If team works on agreed scope expansion then it is not considered as direct scope creep.

scope creep is “adding features and functionality (project scope) without addressing the effects on time, costs, and resources, or without customer approval”. Change on projects is inevitable, so the possibility for scope creep is also inevitable.

By working on unapproved features of a product, a project team devotes time to the unauthorized changes. The work to incorporate these changes must usually be done within the original time and budget estimates, leaving less time for approved parts of the scope. That could mean approved features don't get completed, and the end-product is not what was chartered. Or, it can mean that time and cost overruns to finish the authorized parts of the scope will occur.

There are many ways scope creep can occur on projects. Executives at the sponsor level frequently don't want to be involved in every decision. So, project teams make them. Some change requests are or appear to be small, so again, project teams act on them instead of following a formal change management process. An inflexible or cumbersome change control process may also contribute to unauthorized scope additions.

For various reasons, the project team may want to exceed expectations and deliver “more value” by adding unrequested functionality. IT managers often fail to negotiate more time and budget when requests for additional functionality are made, and the scope creeps.

Few reasons for scope creep that include:

• Lack of clarity and depth to the original specification document.

• Allowing direct [unmanaged] contact between client and team participants.

• Customers trying to get extra work “on the cheap.”

• Beginning design and development of something before a thorough requirements analysis and cost-benefit analysis has been done.

• Scope creep “where you do it to yourself” because of lack of foresight and planning.

• Poorly defined initial requirements.

• “Management promises the sun and the moon, and breaks the backs of the developers to give them just that in impossibly tight time frames.”

• It is impossible to control scope creep, so always work on the highest-priority features.

68Q) What is the difference between Scrum and Traditional project management?

Ans) Scrum is an iterative, incremental approach which runs on three pillars or empirical process - Transparency – Giving visibility to the process outcome to the business

Inspection – Timely checks on the progress towards a sprint goal (to minimize deviations via Sprint Planning, Coding standards, Quality of code, checking Acceptance Criteria, DoD, PO inspecting the functionality, Business users inspecting in Sprint Demo/Review and entire Scrum Team in Retro)

Adaptation – Adjusting a process to minimize issues

Scrum is built on transparency, adaptation, inspection and short learning cycles

Traditional Project Management is a sequential process and no working software available till the end of the project. Difficulty in responding to change. Less Business & IT alignment

Feasibility-Plan-Design-Build-Test-Production-Support

69Q) Are there instances where a waterfall approach should be favoured over Scrum?

Ans) If the requirements are clear and crisp and there will not be any changes to those Business Requirements and if client is happy to wait for the product to get delivered to them then in such case we can opt Waterfall model.

70Q) Are you familiar with other Agile frameworks? If so, which ones?

Ans) Yes with Scrum, Kanban, Extreme Programming

Kanban encourages flow and and seeks to keep work items from being stuck, blocked, or delayed.

The idea is that the team works on fewer items at a time focusing on reducing the time spent on each stage of development. This way there is not a lot of time between when tasks or features start and finish.

There are key guidelines for implementing Kanban:

• Limit "works in progress." The key to Kanban is to limit the number of items in development (tasks or features), not so that you do less but rather that you start and complete more items.

• Visualize workflow. Put all work items on a wall and use columns to denote their status. This allows the team and the whole office to see the progress.

• Manage flow. By analyzing the point at which items get stuck, blocked, or slowed down you can identify (and then remove) bottlenecks.

• Improve collaboratively. Continuous improvement and teamwork are vital concepts in Kanban.

Extreme Programming or XP is an Agile framework that focuses a lot on the quality of practice and the habits of the software practitioner (i.e., the developers on the team). Its main guidelines are as follows:

• Developers will adhere to coding standards, all writing code the same way.

• Use test-driven development. This is a process where developers write the code for a test that a feature should pass (or validate) before proceeding. It is a key part of XP.

• Developers write code in pairs. Usually, one developer writes the test code and the other writes feature code.

• Work is done in short iterations (usually two weeks) and planning happens before each iteration.

• Just enough design and architecture are involved to build the features for the current iteration.

• Code is frequently checked against the master code base so errors can be instantly detected (this is called continuous integration of code with the code base).

71Q) What is meant by Self Organising team?

Ans) A Self Organising team is the team with defined roles and works in collaboration like when there is no manager to bring the team together and push orders, it is upto the members to communicate effectively and work with each other. Respect & trust each other with ownership quality.

72Q) Which Scrum ceremony is most critical? Why?

Ans) My personal feel is Retro is the crucial and critical scrum ceremony. This is the time where the team has to pause on present and talk about the past and think about the future. Like, the team should participate actively on encouraging appreciating eachother on what has been went good. And should be open and courage to share what hurdles they have faced and how they can resolve or avoid it from being repeated. And what are the suggestions on process improvements.

73Q) What is a story point?

Ans) A story point is a relative estimate given by scrum team. Story point can be estimated based on the Criticality, Volume, Uncertainty & Knowledge. We have different methods to provide estimations. If it is at Feature or Epic level then T-Shirt sizing or Bucket sizing is the best way to provide estimates. If it is at Story level then we have Poker Planning & Fibonacci series.

74Q) Describe the relationship between the Scrum Master and the Product Owner?

Ans) Scrum Master & Product Owner are the two vital roles in Scrum. The goal for the both is to deliver a viable product using the Scrum best practices.

A PO is responsible for maximizing the value of the product resulting form the work of the Development Team.

A SM concentrates on project success by assisting PO & scrum team in achieving the target by following the process and principles of agile.

A SM and PO should collaborate with each other and SM assists PO

• in backlog prioritisation if asked by PO

• has the backlog been prioritised based on the latest ideas by PO?

• are all the requirements backlogged?

• Is the size of backlog being still maintainable, i.e., the fine-grained items are at the top of the backlog the coarse-grained items at the bottom?

• The items at the top are following INVEST rule with Acceptance Criteria mentioned

• Help PO to adjust the release plan based on the Sprint Review and Sprint Retro meeting feedback

75Q) Do you believe that a Scrum team should be a part of the product discovery process? Why or why not?

Ans) In Product Discovery Process the real user problems will be discussed and the product teams or tech architects will share their ideas to resolve them and also will raise concerns if they feel the ask is technically not viable.

Involving the entire scrum team is not suggestable when the initial workshops on product discovery is going on. Tech Lead, Tech Architect, Business Stakeholders, Product Owner and Business Analyst will be enough.

76Q) What is a user story, and why is it important?

Ans) A User Story is the need or requirement of the business stakeholder. It is important to ensure the User Story follows INVEST rule with Acceptance Criteria so that the Scrum Team can follow the description provided in the User Story and develop the code accordingly and tested based on the Acceptance Criteria.

77Q) What is the average length of your typical sprint?

Ans) Our cadence is 2 weeks and deployments are fortnightly

78Q) Is velocity a good representation of productivity potential? Why or why not?

Ans) Velocity is units of work completed. Here units can vary based on individual organisation’s own way of choosing the man hours or story points to estimate a User Story.

Velocity is mainly used to understand the time frame to complete the product backlog by the scrum team by taking the average velocity of the team from 3 to 4 sprints. But considering the velocity to measure the productivity is not a right idea. As velocity varies based on the relative capacity of the team and which tends to change over time.

79Q) How do you determine if Agile is successful when working for a company?

Ans) Based on the Product Quality, Customer Satisfaction, Business Value, Process Improvement, Product Visibility, Productivity

80Q) As a Scrum Master, which metrics provide you with the most value regarding tracking a project’s progress?

Ans) As a Scrum Master I use Sprint Burndown chart, Velocity chart, Cumulative Flow Diagram

81Q) How can a SM track the progress of a sprint?

Ans) The Scrum Master ensures that the entire team knows how to monitor Sprint progress following these steps.

1. Review overall progress

2. Review Sprint Burndown

3. Review work items needing attention

4. Review and update impediments and risks

5. Review blocked work items

6. Review team member task assignments

7. Review stories in progress

82Q) What value does a SM provide?

Ans) At the business level, the Scrum master creates a development environment that is creative, safe, productive and supportive and enables multi-directional collaboration.

At the product owner level, the Scrum master facilitates planning and helps product owners understand and adhere to Scrum techniques and practices.

83Q) When you get a team that is new to the Scrum process, how do you motivate them to follow the framework?

Ans) The responsibility of the Scrum Master is to create an environment for the Scrum Team to flourish. Creating an environment for a Development Team to self-organize will help with motivation.

Keep guiding the team in following the agile principles

Conduct workshops in making the team understand what are relative estimations and how it can be done

Helping PO in organising and prioritising the product backlog

Helping the team in understanding the DoR & DoD

84Q) How do you motivate a team in Scrum?

Ans) By giving the team the opportunity & time to learn, to be self-organised, by providing feedbacks, by appreciating their hardwork, by considering the feedback given by the team on SM, by helping the team to resolve impediments and blockers, by actioning on the suggestions provided in the Retros.

85Q) How can I improve my scrum process?

Ans) By encouraging the communication among the team members

By conducting effective Retros and actioning accordingly

Motivating team to be self-organised and take decisions

86Q) What a Scrum Master should not do?

Ans) Should not behave like a project manager

Should not do micro management by checking up too frequently

Should be replaceable easily

Should not be the solo decision maker

87Q) How can a scrum team improve productivity?

Ans) By following the below on time:

• Removing impediments

• Daily meetings

• Prioritised Product Backlogs with User Stories following INVEST rule & Acceptance Criteria

• Make work visible

• Avoid multitasking

• By knowing the teams strengths and weakness

• Awards & Rewards

• Good work environment

88Q) What is a KPI in Scrum?

Ans) KPI is Key Performance Indicator to measure the performance of a company in achieving its business objectives.

89Q) What do you say to motivate your team?

Ans)

• Feel free to come to my office anytime.” ...

• “You can ask me any question” ...

• “I'll look into that and give you an update” ...

• “There's good news and also bad news” ...

• “Here's your area of weakness that you need to work on” ...

• “Here's an assessment of how well you're living up to the company's expectations”

90Q) Who is responsible for ROI in Scrum?

Ans) ROI is Return on Investment. It is Pos responsibility to maximize ROI by identifying the product features and prioritizing those for the Sprint by deciding on which items the team should work first and refining the backlog on regular basis.

91Q) What is ROI in Scrum?

Ans) Return on Investment (ROI) for a scrum project calculates the total revenue generated from a product vs. the cost of the sprints required to develop it.

92Q) What is agile triangle?

Ans) Value delivered to the customer)

Quality (how qualitative product we are delivering to the customer)

Constraints (Scope, Schedule & Cost)

93Q) Why does scrum use Fibonacci series?

Ans) The reason for using the Fibonacci sequence is to reflect the uncertainty in estimating larger items. A high estimate usually means that the story is not well understood in detail or should be broken down into multiple smaller stories.

94Q) How many hours is a story point?

Ans) Each Story Point represents a normal distribution of time. For example,1 Story Point could represent a range of 4–12 hours, 2 Story Points 10–20 hours, and so on. This time distribution is unknown during estimation.

95Q) Why story points are better than hours?

Ans) The way we do story point estimation is better than hourly estimates as it is more accurate and has less variation. ... Story points are therefore faster, better, and cheaper than hours

96Q) Why are story points bad?

Ans) Story points estimates can encourage a number of bad behaviours. They can encourage teams to “game the system” by continually increasing their estimates. This seems to increase velocity, but is fake and makes a mockery of the process.

97Q) What are 5 scrum ceremonies?

Ans)

• Backlog grooming (product backlog refinement)

• Sprint planning.

• Daily scrum.

• Sprint review.

• Sprint retrospective.

98Q) Are story points linear?

Ans) Story points are linear (otherwise it would be impossible to use them as a measure of velocity). However, the scale is non-linear, to stop people arguing over whether something is a "5 or a 6" - by using a psuedo-fibonacci sequence

99Q) What is a spike sprint?

Ans) At the end of a sprint, the spike will be determined that is done or not-done just like any other ordinary user story. A Spike is a great way to mitigate risks early and allows the team ascertain feedback and develop an understanding on an upcoming PBI's complexity.

100Q) What is spike in Jira?

Ans) Spikes are enablers which represents research, design, investigation, exploration, prototyping, infrastructure.

101Q) What is a code Spike?

Ans) It is used to determine how much work will be required to solve or work around a software issue.

102Q) What is a design spike in agile?

Ans) A design spike is a method of intentionally designing a solution to a problem. ... Such a document is usually drafted first by one person and then reviewed by the rest of the team in a Design Review meeting. In this way, one person does the “legwork”, but the whole team gets to have input on the decision.

103Q) Do you estimate spikes in agile?

Ans) Spikes should not be estimated with story points, however some Agile management tools allow you to estimate and track hours on stories or tasks. This would be a better place to track the effort and time spent by the team on spikes. it is termed as a spike because of lack of clarity.

104Q) What is a sprint zero in agile?

Ans) Sprint 0 is there to action the project initiation part of an agile project. ... Although not officially recognized in the Scrum and agile worlds, Sprint 0 is there to cover activities such as product backlog creation, infrastructure set-up, architectural planning, resourcing the team and test plan composition.

105Q) How do you handle impediments in the Scrum?

Ans) The Scrum Master's main responsibility is to identify, track and help remove impediments. Often, team members remove their own impediments. Sometimes, Impediments are beyond the ability of the Team to remove. In that case, the Scrum Master may have to get support from outside of the Team.

106Q) If there is conflict on your team, how do you address it?

Ans) Conflicts are inevitable, even in the most engaged of workplaces. Regardless of the source of the conflict, if they are left unresolved, conflicts can quickly impact employee morale and productivity.

It’s important to practice the following skills when resolving team conflict in the workplace:

• Create a healthy culture. Treat everyone in your team fairly and equally, provide them with praise and recognition, and be open and honest at all times.

• Learn to spot the early signs of conflict. Read team members’ body language (e.g. crossed arms), facial expressions and tone of voice.

• Deal with conflict promptly. Take action early to help your people resolve the situation before it escalates.

• Develop rules for handling conflict. Ensure team members listen to one another, respect each other’s points of view, and refrain from interrupting each other.

• Never take sides. Your role is to help the team members address the issues causing the conflict and to reach a resolution that works well for them.

To consider as an example, once there started internal conflicts between testers and developers. Developers complaining that testing team is raising so many defects for minor issues instead of collaborating all minor issues as one defect and assigning. Testers says that they are following the process of raising defects wherever they are finding it is not meeting the acceptance criteria. The concern of developers here is about their development metrics, it impacts those defects count on their feedback while considering for ratings. I scheduled a catch-up meeting with the team to discuss on this and to close the conflicts. Finally Tech lead, and QE Lead agreed on the thing that whenever testers find a defect they will raise it on spot instead of waiting for more defects as it will unnecessarily cause delays in dev team to resolve if they wont find any further.

107Q) How would you react if a team member wanted to place design and implementation into one sprint, and testing into a different one?

Ans) Practically, this never happens and if still team member suggests this, then as a SM I will guide and make the team understand that no user story will be implemented without going through the proper workflow. Every requirement should follow the DoR & DoD practice to achieve the customer satisfaction and business value.

108Q) If the product owner began assigning user stories to team members without consulting you first, what would you do?

Ans) As a SM, I will inform the team in stand-up meeting as that is the one platform where we all meet everyday and will guide them saying like there is a process to follow in accepting a new story. I should be across if PO gives a new story to the team as this may impact the sprint scope and may impact sprint goal as well.

I will talk to PO’s and make them understand that if team will have capacity then that additional workload can be considered and if team is already loaded then any of the story from the sprint to be descoped and the new one should be considered based on the priority and this should be agreed by all (PO, ST & SM).

109Q) What steps do you take to calculate the velocity and capacity of your Scrum team?

Ans) The average velocity of 3 to 4 sprints will give me the current velocity of the team. Capacity is the team availability (in terms of hours or days or any time measurement unit) for a particular sprint. I will check with the team before Sprint Planning if they have a plan to take leaves or if there are any public holidays that to be considered for Sprint Plan.

110Q) If your team’s velocity is unstable, what may be causing the issue?

Ans) Team’s velocity may not be persistent due to few factors like based on capacity for that sprint, based on complexity of the requirements, based on the team maturity.

112Q) What steps would you take if a Scrum team member consistently missed the daily meetings?

Ans) If this happens, I will talk to that team member in person and will try to understand the concern he/she is having. Also, I will give him/her some time to bounce back if they are sick or having some personal issues and will tell him/her to post their updates or any impediments or blockages to me if any. And also will remind him/her that stand-up meeting is so important to understand on what others are doing, to understand what blockages team is having, to learn the progress of peers. As Scrum team members are self-organized soon that person will join the stand-up back after couple of days as usual.

113Q) If the product owner wants a user story included in the upcoming sprint, but it won’t be ready until the day after the sprint begins, what would you do?

Ans) Fewer times this happens with PO’s when Pos don’t have enough information or confirmation on the requirement but that requirement is having high priority. In those cases, what we followed was we wont estimate that story and in Sprint Planning we will keep 1 Dev efforts aside to address that requirement. Also, we will consider extra stories for Sprint Plan to be at safer side in case the PO won’t receive the confirmation on the requirement then that Dev can start looking into the smaller stories which we considered extra.

If PO comes on time after the Sprint starts then I will arrange for a grooming session with that Dev/QE Lead/BA/PO so that the requirement and acceptance criteria are well captured and story is estimated and brought into the sprint.

If PO comes after 3 to 4 days then in the grooming session, I will check with PO on the release date, if it is nearby then I will check with Dev and Tester if it is possible to achieve. Based on their confirmation and Pos approval the release date will be finalised on that requirement.

114Q) Does a Scrum Master need technical expertise in the product niche to succeed? Why or why not?

Ans) A Scrum Master role is not a technical role. But it is no harm if a SM has development background and got sound knowledge on coding.

115Q) What trait is most critical to your success as a Scrum Master?

Ans) A successful SM should have:

Flexibility to adapt changes - A great Scrum Master represents sensibility to adapt to new changes. For agile functional process, Scrum Masters practice the persistence of flexibility. And this persistence becomes essential when a Scrum Master organizes meetings for the team members and ensures everyone’s presence.

Encourages Collaborative Effort - To promote collaborative culture throughout the team is the hallmark quality of a good Scrum Master. Furthermore, the Scrum Master encourages team members for open discussions. But how does a Scrum Master create an atmosphere that garners collaboration? He/she does that by mincing the right words and actions at the right time.

Responsible Behavior - In a traditional organization, the required authority is bestowed upon a selected individual to carry out the completion of tasks. For Scrum Master, it’s a different picture entirely. Though a Scrum Master undertakes the responsibility of the team adoption and success, the individual doesn’t assume the required power that may help achieve the set goals.

Full-Time Commitment - It requires someone whose shared willingness and commitment extends far beyond the workload of a full-day responsibility. Furthermore, a good Scrum Master would transfer the same level of commitment to the team members to work on a particular project. Similarly, selective hindrances raised by the team members don’t have to be resolved in a single day by a Scrum Master. A mature Scrum Master understands that some issues require extended discussion time and managerial approval.

Ability to Influence - ‘Command-and-obey’ is not even the last resort for a good Scrum Master. Though the initiation of a project may require the team members to be influenced to work collaboratively. However, Scrum Master’s ability to influence team members and outside organizational members requires broad-minded perceptiveness and objectivity. A Scrum Master, for instance, can influence the team members via pair programming, multi-level testing, or even through test-driven development.

Extensive Knowledge - The technical knowledge of a Scrum Master helps the team members to strive towards the pursuit of planned goals. The wealth of knowledge a Scrum Master possesses can dramatically increase the probability of team members to address and fix project issues.

Although the nature of the knowledge may be subjective, a good Scrum Master would have the mastery to handle the key technical issues. Additionally, the comprehensive understanding of specialized terms and procedures should be set up by Scrum Masters.

Conflict Management - It is vital to remember that each Scrum Master has a different identity, and thus, every Scrum Master utilizes a different style to manage work. Furthermore, an experienced Scrum Master also solves issues that team members can’t seem to agree on. At the end of the day, a Scrum Master wants to be useful as a facilitator or moderator to achieve the end goal.

Scrum Masters, for instance, use compromise as a conflict resolution tool. However, it requires team members to have rational aptitudes to navigate through contradictions and make necessary arrangements for everyone.

Courage to Lead - Although the role of the Scrum Master is strictly responsible as servant leadership, the courage to uphold and tap into that leadership quality is often misguided and misunderstood. Besides, a good Scrum Master would never tell a team what to do.

Be that as it may, it’s easier said than done. And to transition to such a mindset requires individuals to let go of all authoritative personality trait that may hinder the birth of a Scrum Master.

Awareness to be Expendable - Here’s a thing; a good Scrum Master is fully aware of the possibility of being dispensable. And as Scrum Masters consistently support the growth of the team members, their regular need concurrently decreases. However, the overall value distribution, periodic assessment, and urgency to analyze data make Scrum Masters more than wanted.

Derives Empathy - For an observant Scrum Master, the longevity of the work allows him or her to develop empathetic feelings towards the team members. Over a long period of time, Scrum Master learns to listen, understand and appreciate the entire team. And this understanding positively translates into the work efficiency of the group.

Confidence to Self-Organize - The most used yet effective trait of a Scrum Master is the ability to convey assignments to various team members throughout the day. But the same ability expands when Scrum Masters have to train the entire group to self-organize. This model significantly discourages the reluctant reliance on executives.

The same model can also be used to decide respective choices about work timing, modify work readiness, and collaborate with colleagues that fail to meet up group objectives.

Understands Workflow Rhythm - A mature Scrum Master understands the significance of continuous workflow rhythm. But to maintain the rhythm is relatively much more demanding than creating it. The rhythm aligns workforce to inherently become more aware of the data, time, and intent behind each planned event.

Source of Inspiration - A brilliant Scrum Master will find the source of inspiration for each project. This inspiration becomes the key to comfortably stretch work hours. A Scrum Master gets the right dose of inspiration and love for the project before it can be transferred to clients or team members.

The essence of a well-thought-out strategy about the item eventually becomes the need of the hour for every Scrum Master. At last, the work inevitably differentiates and reflects the hard work of the Scrum Masters. Though not to mention, pouring a soul into each project requires certain strong commitment.

Humble Attitude - A good Scrum Master does not perform his or her duties for the sake of the ego. This unbiased and unprejudiced mindset figuratively and helps the group to work collectively.

The end goal for a Scrum Master ultimately starts and ends with helping the team achieve a specific goal within a particular time frame. The humble attitude of a Scrum Master is not restricted to job responsibility but extends beyond recognizing the real value of each team member.

116Q) Tell me about a time your Scrum team was on the path toward failure. What did you do to bring things back around?

Ans) We has a requirement on Onboarding the brokers to the 3rd party plugin Activepipe. So, initially when the grooming was done, we have agreed that we will cater all the 4 scenarios on onboarding & Offboarding a broker group. In the initial the plan for deployment was in Dec before moratorium.

Later, the management has decided to cater for this interface after Jan. But, to our surprise they decided it to be in prod by Feb 26th in the 2nd week of Jan. The total estimated efforts were 40+ and we had only one .net developer in our team. Also it was having dependency on API changes which was part of another team. The Tech Arch who has to look in to the API changes went on vacation and will not be back till Feb 22nd. And the challenging thing was he didn’t pass the required changes to be done to any of his team members.

I have scheduled a meeting with another Tech Arch who was involved in few of the discussions, and the API team’s manager, the API dev, .net dev, tester, PO & Project Manager. We have revisited the requirements and explained the new API dev on what exactly was needed. Also, I have raised a concern that we cannot cater all the scenarios for this Feb 26th deployment as we have only one dev and if another dev can be given from other squad. Then the management agreed to provide an extended hand of another developer. But still we cannot achieve this as the capacity is less and demand is more.

Then I have asked if we can consider only happy case scenarios instead trying to pull all the use cases for this deployment given shortage of resources. Then we all agreed on this after so many levels of discussions and helped the .net developers to collaborate and work with the API Dev.

117Q) What would you do if a product owner wanted to alter a sprint that was already in progress?

Ans) Changing the Sprint scope is inevitable. It happens sometimes that PO’s will change the Sprint scope but that will not be a drastic change but adding of one or couple of stories. Sometimes it ends up with descoping the other items and adding the new ones, fewer times team will achieve all the newly added stories within the capacity.

But as per Scrum guide this should not happen if we are in the mid of the sprint and sprint scope should be locked. I will explain the PO’s why this is not encouraged, so that going forward PO’s will have a better plan well in advance for Sprint Planning.

118Q) What do you find more challenging, leading a team that’s new to Scrum or dealing with a first-time product owner? Why?

Ans) I never got a chance to lead a new Scrum team and also to deal with a first-time PO.

here are five great questions for the end of your Scrum Master interview:

1. How long has the company been using Agile practices? What about Scrum specifically?

2. Are there any duties that deviate from the standard Scrum Master responsibilities, such as those leaning more into the project management field?

3. What metrics are used to define success for this role?

4. Can you tell me about the average day for a person here in this position? What about the most challenging day?

5. Will this role oversee just one Scrum project at a time, or several?