Which Agile methodology would you recommend and why?

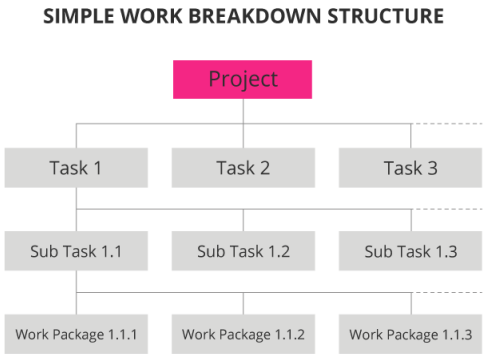
Agile Methodology: I would recommend using SCRUM. SCRUM would help segregate the tasks in small sprints of 2 weeks which keep the window with the total of 4 sprints for the entire project plan. As each sprint planning would be done for a period of three months at a time, the modification of the product backlog to accommodate changes that might crop up due to external product dependencies will be better taken care of. With a sprint of two weeks, the scope to reshuffle the items/epics to be tracked and handled for a particular sprint will also remain flexible.

How would you structure it? What roles would Product, Development, testing resources play within it? How would you use it to coordinate projects between your three teams?

The scope of my project is one of the most critical parts of my plan. It will help me determine what needs to be done to obtain or deliver the desired final result, and this document will include the project goals, activities, dependencies, and deadlines.

A work breakdown structure(WBS) is the second thing, I would work on deriving after consultation with each team for each team in the project workflow. Once this kind of a structure is known for each teams' deliverables, it will be easier to identify the dependency between tasks amongst the teams. Using Microsoft Project plan to segregate the tasks, I can achieve the Gantt chart for the project, which would help me track the progress of the teams and the tasks completed for the project.

Once items are known as in terms of tasks to be completed, the dependencies will carve out. With the dependencies identified, a simple Gantt chart(like one shown below) will detail out the process and timelines of coordination.



Each of the three teams, say Team A, B and C would each have a Scrum Master, a Product Owner and Developers in the Scrum Team. The group of testing resources involved in each Sprint will follow a lean pattern and would definitely be lesser in comparison to the number of developers in the team. Following a matrix structure for the team organization will permit me to have a structure in which there is more than one line of reporting product owners. This will bring in the expertise of the team for the complete product development while breaking the monotony and give more flexibility to the expertise held. Teams work with members of the other teams employing the best practices in the different functions involved

Product Owner: He or She will hold the vision for the product under their individual supervision and would be responsible to create the product backlog items that would be handled as a part of the sprint planning cycle.

Development team: The individual development team will be given full control to decide which epics or items would become a part of the sprints during the sprint planning. As the teams A, B and C might have interdependencies, the sprint planning meeting will allow each team to observe the other team's dashboard of tasks and shuffle the items interdependent amongst themselves accordingly.

Testing team: At the initial stage of planning, this team will set the sprint tasks keeping in mind that they have to perform the task of setting up the platform for UAT and prepare use cases/test cases that need to be executed once the tasks are completed and handed over to them by the development team. For subsequent sprints, the idea of which items will be delivered will be known to the team and planning of the items will be done keeping in mind the timelines projected during sprint planning.

Microsoft Project will be used as a tool to depict the expectations for each stage of development and testing, which in turn will contain the successor and predecessors for each task planned during the sprint. This will help coordinating the tasks between the teams A, B and C. I will prepare this chart after discussion with the Product Owners of the three teams, who are knowledgeable, to help me guide the dependency flow among individual pieces across the three teams.

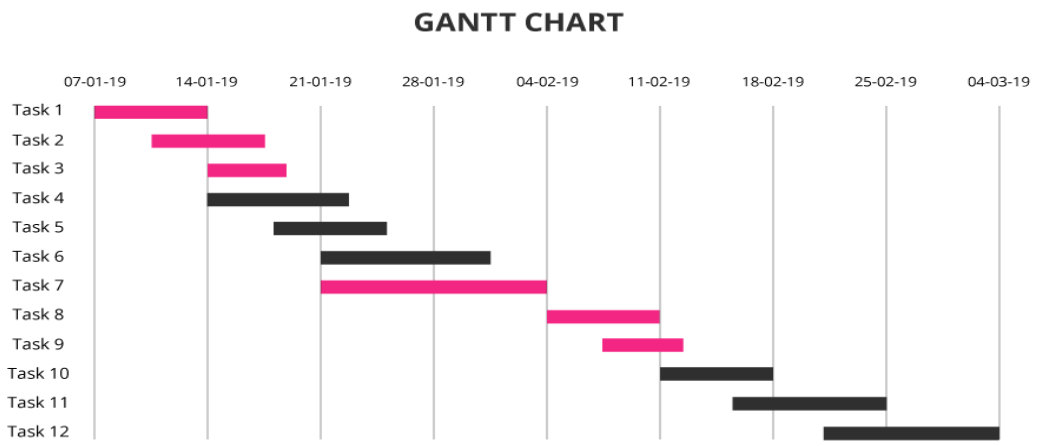
How would projects involving teams on other products and methodologies be coordinated?

To tackle the dependency that might be present with external parties will be handled by means of an excel tracking sheet at my end. I will be the one responsible to make sure the deliverables and dependencies are communicated across the board. However, the dependency of their tasks amongst themselves, at their end need to be handled by them explicitly. I will communicate and keep a track of what would be delivered by my team and what do I expect, if any, as a part of their inputs to my team. This Microsoft Excel sheet tracking will help me know the commitments projected by my team and the teams of other related projects. This sheet will be shared periodically with the other teams to make sure we are all on the same page, during intra team meetings.

How would you handle dependency identification and tracking, both between your teams and with other teams?

Using the Microsoft project plan document, I would personally create a to-do list for each work package by establishing everything that needs to be done to get from the starting point to the finish. Doing this will help me underline dependencies between teams and determine which tasks are dependent upon one another (in other words, activity B cannot begin until activity A is complete) and which tasks can run concurrently.

Considering the length of each task and whether certain task due dates can have a float or slack, I would have a simple Critical Path method chart made out that would help me list out the amount of time an activity can be delayed without affecting the project completion date. A Gantt chart, like on example shown below, derived from the project plan will also help me track the items as per the project plan.



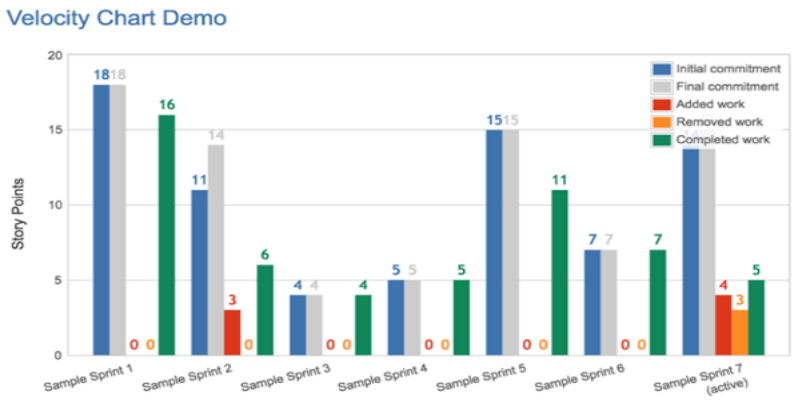
Which KPI would you track and report on? What would they indicate? What steps could you take to correct negative trends? How would they be used to track scope-driven or time-limited projects?

Please note: I am using dummy image data as provided on internet

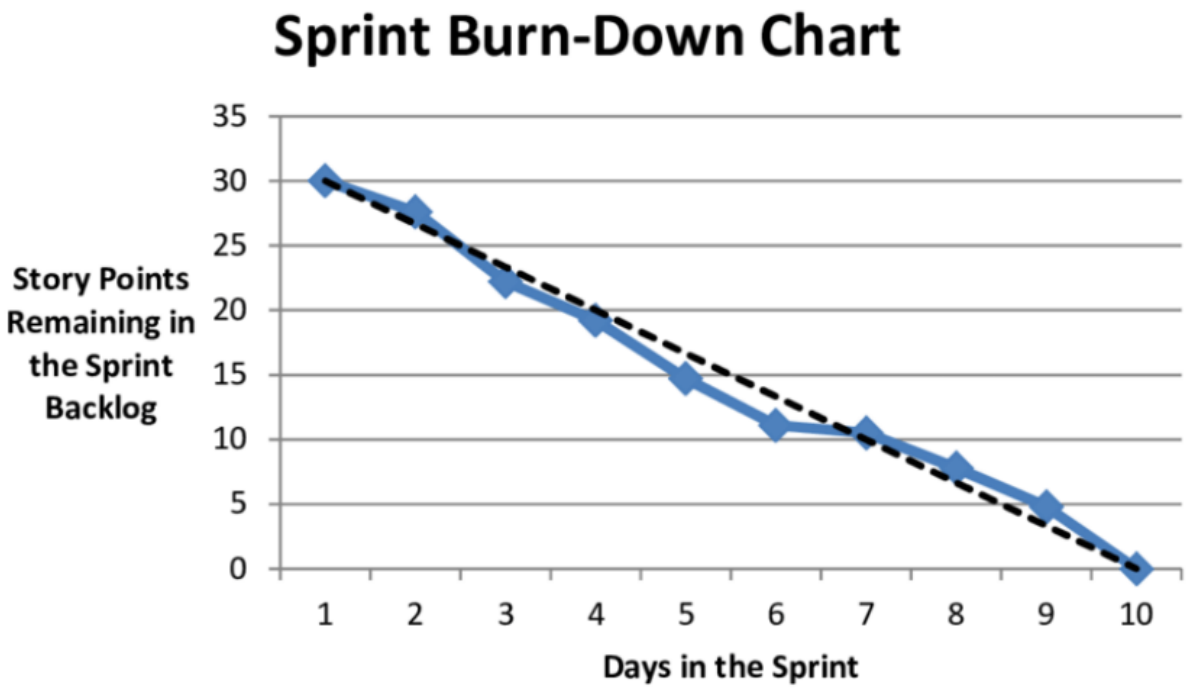
I would use the tools provided by Jira to produce KPI reports.

1. Velocity Chart: The velocity chart is a KPI that would help me take an overview of how much work has been delivered for each sprint. It will help me to have a clear view on future perspectives and on the workload of my team/s.

   The Scrum master of each team A, B and C would be expected to produce this metric which would help me get an overview of the units of work for each task assigned and how long is the time taken in real. During the sprints every member of the team would communicate on the tasks that have been finished or completed. And at the end of the interval that we will thus be able to analyze the completed number of units of work.

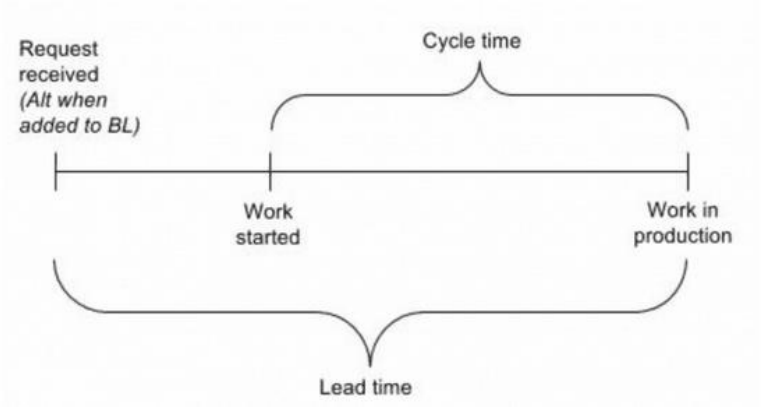


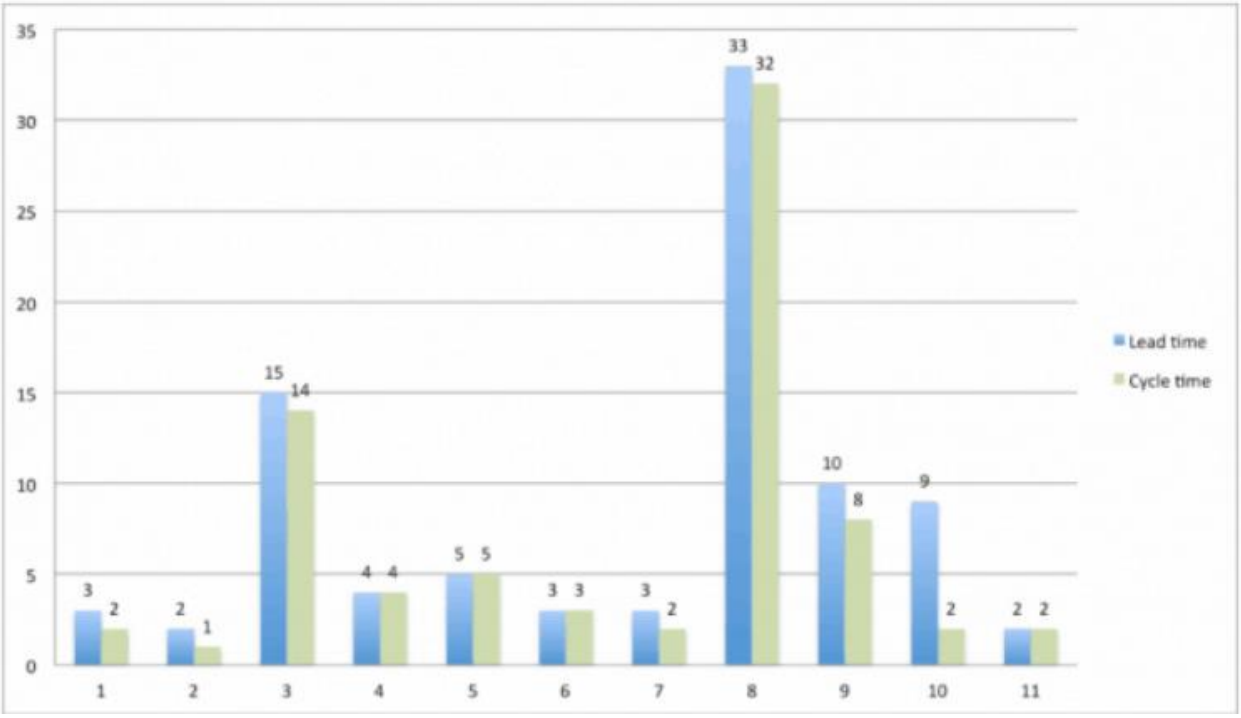
2. Burndown Chart: This KPI would be produced by the Scrum Master, analyzed by the Product Owner and passed on to me to help me determine the amount of work that has been completed in a sprint, and the total work remaining. Burndown charts will help me predict the team's likelihood of completing their work in the time available. A positive burndown number would indicate a good result and remove the chances of blockages that might arise between the interdependent teams. If negative burndown numbers are shown, I will have to sit down with the Scrum Team to understand the impediments that might be hampering their working performance.



3. Lead Time Chart: The “lead time” elapsed between the formulation of a user story and that story being used “in production”, that is, by actual users under normal conditions would help me know the efficiency of my team. It will help me determine rate at which I can expect the implementation of user stories to keep up with their arrival. It’s calculated as the time available for work divided by the number of completed stories needed. The higher or positive number for the lead time shown by the chart would mean that the formulation of the user story to its completion for delivery has been performed well.

4. Cycle Time: Within a sprint, the lower this number, the better is the result or team performance. Scrum Master can produce this report as a KPI to help me understand the capability of the team by projecting the time taken by the team to complete the work and push it to production from the time it actually got started. As the two KPI’s of Lead time and Cycle time show me the ability of the team deliverables with a sprint, it has to be kept in mind that I should see a lower number in terms of cycle time as that would mean, the team has been efficient to produce the deliverable in a lower number of hours/days.


Request
received
(Alt when
added to BL)
Cycle time
Work
started
Work in
production
Lead time


35
30
25
20
15
10
5
0
1
2
3
4
5
6
7
8
9
10
11
3
2
2
1
15
14
4
4
5
5
3
3
3
2
33
32
10
8
9
2
2
2
Lead time
Cycle time