Scrum Activities Overview / Suggestions

Team: Cooked

Created by Gavin Frisby (SEII Spring 25')

Sprint Planning

- **Pre-Condition(s)**
  - A set of PBIs ready to be broken down into tasks and estimated
  - Sprint goal describing what the team's goal is during the sprint
  - A clear sprint board that does not carry over any tasks over from previous sprints.

- **Process**
  - Sprint Planning is kicked off with the PO presenting the sprint goal, as well as PBIs to reach said goal. Following this, the PO will briefly explain / go over each PBI and its acceptance criteria until the dev team and PO reach a mutual understanding of each item. The sprint velocity will then be outlined if it is not already understood. The dev team will then approach each PBI and break them down into effort-estimated tasks that populate the sprint backlog.

  - For the case of task verification and ensuring a task is sprintable, the S.M.A.R.T (Specific, Measurable, Achievable, Relevant, and Timely) principles should also be tied to different aspects of a task.

  - In the case that there were any incomplete tasks from the previous sprint, it is up to the discretion of the dev team to include these tasks in the next sprint. However, the relevance and importance of these tasks should be re-evaluated as they pertain to the current sprint goal.

  - Visualizing PBI breakdowns can help in task creation and create more clear discussion.

- Time box for this activity is variable, typically can be around 35 – 45 minutes In the case of Kinser's class.

- The method in which effort estimates are reached is ultimately up to the discretion of the dev team. Below is one example of an effort estimate approach used in previous semesters. In this example, velocity and tasks are all valued in hours. However, for the planning poker activity, the "story point" amount is number people raise with their hand to estimate. For anything longer than four hours, the task likely needs to be broken down further or researched.

    - 0 "Story Points" = 30 Minutes
    - 1 "Story Points" = 1 Hours
    - 2 "Story Points" = 2 Hours
    - 3 "Story Points" = 3 Hours
    - 4 "Story Points" = 4 Hours

- **Post-Condition(s)**
    - A populated sprint backlog that aligns with sprint velocity.

Daily Standup

- **Process**
    - Daily Standup is a sprint activity the team will participate in at the start of each day. During this activity, each member of the team will present the following within a 90-second time limit.

        - What I have worked on since the last daily standup
        - What I plan to work on today / moving forward
        - Any potential blockers

- After each member has participated, the meeting will move over to crosstalk. During crosstalk, any member can bring up any larger issues or discussions that need to be had as a team.

- Daily Standup is timeboxed at 15 minutes and should not exceed this time.

Sprint Grooming

- **Pre-Condition(s)**
  - The team has a foundational understanding of the goals of the project, as well as its current state.

- **Process**
  - Sprint Grooming is a process during which the team will reapproach the requirements and epics of the project and form new PBIs for future sprints. The PO will discuss the technical vision and goals of the sprint while the dev team approaches technical aspects and feasibility.

  - Time box for this can vary based on team need and the current state of the product backlog. The goal for the product backlog at any point is to have enough to fill at least two sprint, but no more than 4.

- **Post-Condition(s)**
  - Enough PBIs to fill at least 2 sprints, but no more than 4.

Sprint Review

- **Pre-Condition(s)**
  - A potentially shippable deliverable/deployable product to be presented to the PO.

- **Process**
    - The goal of the sprint review is to examine what has been completed during the sprint, as well as present the final deliverable to the PO. The PO will then review the product and decide whether it can be shipped or not.

    - The way in which Tasks / PBIs are verified or examined is ultimately up to the discretion of the Product Owner. My team had success with a presentation approach, where the dev team member responsible for each task can explain, and potentially demonstrate, the work related to their task. This gives the dev team members an opportunity to show off their work, the PO to ask questions, as well as ensure everyone is on the same page regarding what a task entails.

- **Post-Condition(s)**
    - An updated product backlog that is ready for the next sprint.
    - A deployable product depending on the result of the PO's review.

Sprint Retrospective

- **Pre-Conditions(s)**
    - The previous sprint has been completed, as well as its sprint review.
    - Burndown Chart and Sprint Overview (Value of the burndown chart is dependent on where the team is at during the semester).

- **Process**
    - The overall goal of a sprint retrospective is to allow the team to reflect on the positives and negatives of the previous sprint. The dev team will answer the following questions:

        - What Went Well?
        - What Do We Need to Improve Upon? (Or what didn't go as well)

- The post-it note approach is one way this activity can be completed. Following some discussion regarding what notes were brought up, one aspect that went well and one aspect to be improved upon will be voted as a priority during the next sprint. Finally, the dev team will brainstorm actions to take in order to achieve / maintain the focused aspects.

- Alternatively, Retrospective can be performed digitally on many different platforms, Mentimeter being the one my team (Cooked) utilized. Menti offers more many customization options that can make the activity more interactive / engaging for the team. Furthermore, it can remove some of the existing group think that is present in the voting process when using the post-it note approach (dev team members can see each other's votes). My team saw a fair amount of success with this over the post-it note approach once implemented.

- While not every aspect will have a direct action associated with them, each one should be examined and acknowledged by the team.

- Alternatively / Additionally, this time can also be used to re-approach the team's definition of done to ensure it still aligns with the current goals / status of the project.

- **Post-Condition(s)**
  - One aspect to be improved upon during the next sprint, as well as what efforts will be made to achieve this.

12 Agile Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software

2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4. Business people and developers must work together daily throughout the project.

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6. The most efficient and effective method of conveying information to and within a Dev Team is face-to-face conversation.

7. Working software is the primary measure of progress.

8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9. Continuous attention to technical excellence and good design enhances agility.

10. Simplicity--the art of maximizing the amount of work not done- -is essential.

11. The best architectures, requirements, and designs emerge from self-organizing teams.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Paraphrased 12 Agile Principles

1. The first priority of any team is to consistently provide valuable deliverables to the customers to ensure the end product fulfills their needs.

2. Keep an open mind to changes and new perspectives throughout the entirety of the project, as this will aid in creating a more viable product. In short, be agile.

3. Deliver working and valuable software consistently and often.

4. Communication between businesspeople and the development team should be conducted as often as needed. This goes for both internal stakeholders and the customer.

5. Form a team of hard-working self-starters that are motivated to succeed and provide them with the means and trust to succeed. If you're not forming the team, then be that self-starter.

6. Face-to-face conversations are the best way of communicating with one and other when possible.

7. The efficiency and success of a team is measured by their ability to build a deliver working valuable software.

8. Consistency in pacing, process, and communication is essential for a team long term. Sustainability is key.

9. Make sure you are continuously holding yourself and your team to a high standard technically. Good design makes #2 easier in the later stages of a project.

10. Work smarter, not harder. Always look at more efficient ways to solve a problem, or better yet, solve multiple problems at once.

11. When possible, teams that have the opportunity to organize themselves are more likely to succeed. However, this is not always possible.

12. Reflection and consistent improvement are essential in growing as both an individual and a team.