TASK 13P

Sprint Retrospective

SWE30010 - Managing IT Projects

Class: Fri 08:00 DT7.2 **- Tutor:** Pham Thi Kim Dung

GROUP 1: The Beavers

Trac Duc Anh Luong - 103488117

Minh Nghia Nguyen - 103806269

Gia Minh Nguyen - 103487156

Cong Anh Nguyen - 103792960

Tran Dat Dinh - 103487143

Anh Duc Nguyen - 103488489

Minutes of Meeting

Location: Online
Time: Start at 9:45 am - End at 10:00 am (26/03/2024)
Link to video recording:
https://drive.google.com/file/d/1a9QkXkXPi_59uUApkSoxtZCifrkf0vzV/view?usp=sharing

Attendees

- Trac Duc Anh Luong (Product Owner)
- Minh Nghia Nguyen (Scrum Master)
- Gia Minh Nguyen (Scrum Team Member)
- Anh Duc Nguyen (Scrum Team Member)
- Tran Dat Dinh (Scrum Team Member)
- Cong Anh Nguyen (Scrum Team Member)

Notes

1. **Your team's velocity – ideal (from your ideal burn-down chart) vs actual (from your final burn-down chart)**
   1.1. Did your team overestimate your ability? Or Did you underestimate the effort required to complete the tasks?

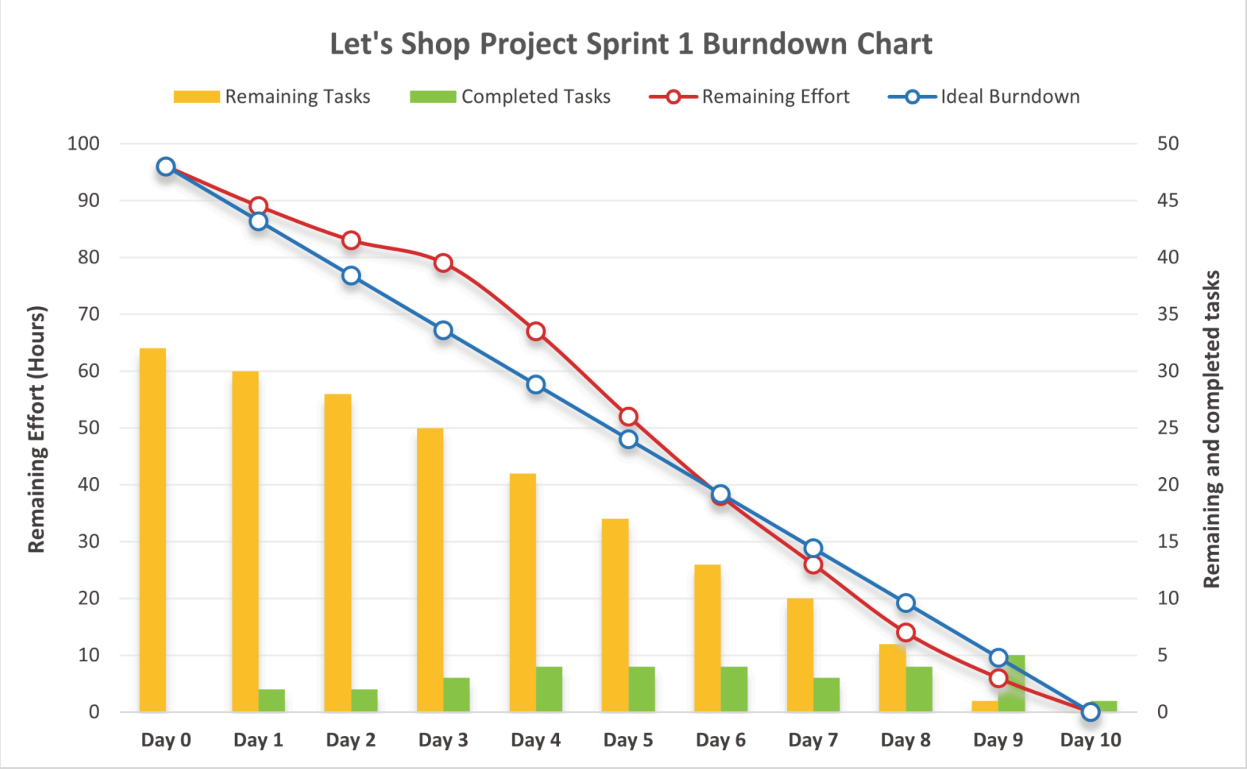

Figure 1: Sprint 1 burndown chart

Upon reviewing our Sprint 1 Burndown Chart, we can observe that the remaining effort line initially indicates that the team started with an optimistic outlook, closely following the ideal Burndown trajectory until day 2. However, from Day 3 until Day 5, the actual burndown deviates above the ideal line, suggesting an overestimation of the team's capacity to complete tasks. For instance, by Day 4, we expected to have about 60 hours of effort remaining, yet we reported around 70 hours. This 10-hour difference may not seem substantial, but it reflects a notable shift in our anticipated progress. In contrast, the second half of the sprint shows a marked improvement in efficiency; despite starting Day 6 with a higher-than-expected remaining effort, we see a steeper decline in the remaining effort line compared to the ideal burndown. By Day 9, we almost realigned with the ideal trajectory, finishing the sprint with the remaining effort converging on the ideal line.

The significant decrease in remaining tasks and effort in the second five days, as opposed to the first five, is encouraging. Although there were initial setbacks or misjudgments, our team adapted quickly by streamlining processes and increasing focus and productivity to catch up with the planned schedule. This adaptation is evident in the sharp decrease in remaining tasks, from over 20 tasks at the start to zero tasks left by the end of the sprint.

Overall, this burndown chart indicates that we can learn and adjust throughout the sprint to ensure that project goals are met, albeit with some initial challenges in accurately gauging task complexity and effort.

1.2. What can you do in order to get a better understanding of the "complexity" of the tasks required? Or What can you do in order to get better time estimates next time?

There are several steps we can take to get a better understanding of the "complexity" of the tasks required:

1. **Breakdown Tasks into Subtasks:** When faced with a large task, breaking it down into smaller, more manageable subtasks will help the team members understand the various components and complexities involved in completing the task.
2. **Estimate Time for Each Subtask:** Once the big task is broken down into subtasks, we can estimate the time required to complete each subtask based on complexity, dependencies, and team members' skill levels.
3. **Track Time Spent:** As team members work on the tasks, tracking the actual time spent on each subtask is important to monitor progress and identify any deviations from the planned estimates.
4. **Compare Planned vs. Actual Time:** Regularly comparing the planned time estimates with the actual time taken to complete each subtask will help the team understand where there may have been inaccuracies in estimation and provide valuable insights for future planning.
5. **Review and Improve Estimates:** Based on the comparisons between planned and actual time, we will conduct thorough reviews to identify areas where estimates were inaccurate, analyse the reasons behind any discrepancies and refine estimation techniques for future tasks.

2. Your team's process

2.1. What is working? Why?

During our project development, one of the key practices that are working and have significantly contributed to our productivity and efficiency is the daily morning meeting. This daily routine is a keystone for our project management approach, allowing us to review progress, plan for the day ahead, and identify potential challenges or choke points early on. Each morning, team members share updates on the tasks they completed the previous day. This review helps track progress and ensures everyone is suitable for their assigned work. Our daily meeting also serves as a platform to set clear objectives for the day. By outlining what needs to be accomplished, each member always knows what to do, which helps reduce uncertainty. While we review our completed tasks, we also discuss upcoming tasks, dependencies, and potential roadblocks. This proactive planning allows us to address issues early and adjust priorities if needed, keeping the project on track.
The most valuable aspect of the daily morning meeting is the ability to identify choke points within our project workflow. These chokepoints are areas where the development progress might be slowed, tasks might be stuck, or dependencies might be unclear. We enhance our efficiency and promote effective problem-solving by actively seeking out and addressing these issues.

2.2. What is not working? Why not? Any suggestions to improve the situation if this occurs in the future?

The first sprint that our group conducted has come to an end. Our product has been developed quite smoothly and achieved many good results. However, there are still limitations and difficulties that we need to overcome in future sprints. Testing completed features is necessary to check whether those features have acquired minimum requirements. All members carry out cross-checking on each other's work. This ensures that every task is observed objectively and each member can improve his work. For the cross-checking method to be conducted, a member has to wait until another fellow finishes his work. Moreover, a lack of communication between members of our group may result in some members doing the same tasks. This slows the process and reduces the team’s performance.

To improve the above situation, all members have agreed that we will no longer create a task independently. Instead, a member has to request the leader to create a task, and then he will be assigned to that one. This will ensure that no members do the same tasks as others and that they do not have to wait for cross-checking. If a member has finished his work, he can freely continue with the next task, and the checking phase could be carried out later. This method will enhance the performance, and we will have more time to plan for the next sprint if needed.

3. Comments/suggestions from the team for improvements

We must continue to utilise and improve our adaptive management strategy. This method assures the best use of available resources and enables us to respond quickly to changing conditions. Keeping a careful balance in the duties assigned to team members is a crucial component of this method, as it guarantees that each person's burden is fair and doable. Promoting open communication and a culture of helpful criticism is critical during task reviews. We can improve our processes and achieve more efficiency and effectiveness by improving our communication skills and aggressively soliciting feedback from every team member. In addition, we should expand our understanding of OutSystems and web development, given the significance of lifelong learning. This will strengthen each of our skills and make our team more capable as a whole when it comes to meeting new difficulties.

Self-improvement proposal

No.	Proposal	Member
1	Improve backlog item selection and planning for more straightforward task estimation through WBS.	Trac Duc Anh Luong
2	Our team needs to log the duration of each subtask more carefully and consistently compare expected versus actual time spent to identify trends or opportunities for refining the process.	Nguyen Anh Duc
3	Improve communication skills so meetings can have higher quality.	Cong Anh Nguyen
4	I want to deepen my understanding and proficiency in low-code development, and I am first trying to learn more about OutSystems.	Tran Dat Dinh
5	I need to prepare better to test features by applying tools and technologies.	Nguyen Gia Minh
6	Improve my time management skills to enhance my productivity and effectiveness within future team projects.	Nguyen Minh Nghia

Figure 2: Self-improvement proposal