

Diary Scrum Simulation
Nguyen Xuan Binh - 887799

I. Discuss honestly what you learned during the simulation (and mention if you feel you didn't learn well something about Scrum).

I was new to this Agile development technique, so I need to step back to see what really happens in this field. The term 'scrum' itself is an abbreviation from scrummage. Scrumming is often used to describe a tightly packed disorderly crowd. But in Rugby, it defines a joining together in a tight organized formation. Scrum turned from being used as a metaphor to actually becoming synonym to "adding flexibility to product development". Within Scrum, a sprint is a short, time-boxed period when a scrum team works to complete a set amount of work. Unlike the word iteration though, the word Sprint has an "urgency" or "straining effort" attached to it. In a sporting event sprint, we will run as fast as possible towards the finish line. Note: this diary is also a learning diary for me to learn SWE concepts.

Nowadays, Scrum is a popular agile framework used primarily for software development. It is designed to help teams work collaboratively to deliver high-quality products incrementally. Scrum emphasizes *adaptability, feedback, and collaboration*. In Scrum, there are officially recognized roles and artifacts/events, which are at the central heart of the Scrum ideology.

The primary roles in Scrum that I have observed in the simulation are:

1. Scrum Master:

Role: The Scrum Master is responsible for ensuring that the Scrum process is followed correctly and efficiently. They act as a coach to the team, helping them to understand and implement Scrum practices and principles.

Responsibilities:

- Facilitating Scrum ceremonies (Daily Standup, Sprint Planning, Coding, Sprint Review).
- Removing obstacles that the team might face during the sprint.
- Protecting the team from external interruptions.
- Ensuring that the team is functioning well and collaborating effectively.
- Working with the Product Owner to ensure the product backlog is well-maintained.

2. Agile Coach:

Role: An Agile Coach is similar to a Scrum Master but usually operates at a higher level, often working with multiple teams across the organization to adopt and improve Agile practices.

Responsibilities: Helping teams understand Agile principles, providing training, facilitating workshops, and assisting in organizational change efforts.

3. Product Owner:

Role: The Product Owner is responsible for defining and prioritizing the product backlog. They represent the stakeholders and customers, ensuring that the team is working on the most valuable features.

Responsibilities:

- Creating and maintaining the product backlog.
- Prioritizing the backlog items based on business value and stakeholder feedback.
- Accepting or rejecting the work results at the end of each sprint.
- Collaborating with the Scrum Master and the development team to ensure clear understanding of backlog items.

4. Customers/Stakeholders (not official role):

Role: The Product Owner represents the interests of the stakeholders, which include customers. So customers or stakeholders have the direct influence on the Product Owner, who will be responsible for understanding customer needs, preferences, and feedback, and then translating those into actionable items in the Product Backlog.

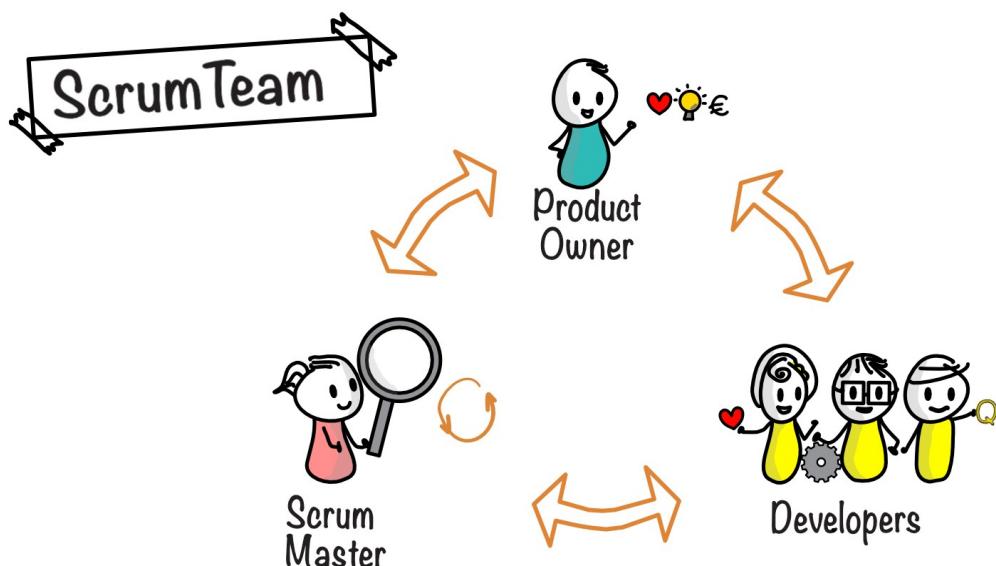
Feedback Loop: One of the core principles of Scrum is to build products that provide value to the customers and stakeholders. Customers provide essential feedback, often during Sprint Reviews or after product releases, which informs the Product Owner and the team about what needs to be improved or developed next.

5. Development Team:

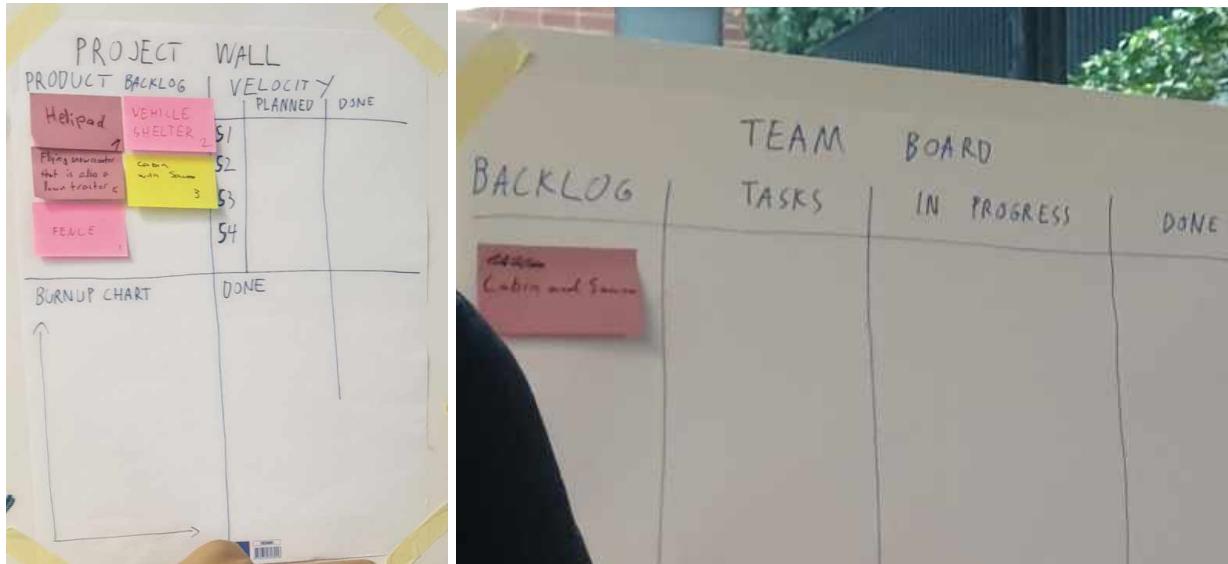
Role: The Development Team consists of professionals who carry out the actual work of delivering a potentially shippable product increment at the end of each sprint. This includes designers, developers, testers, and other roles necessary to produce the product.

Responsibilities:

- Estimating the effort required for each backlog item.
- Selecting the amount of work they believe they can complete in a sprint during the Sprint Planning. If the work is not completed, it will be moved forward towards the next Sprint
- Collaborating to design, develop, test, and deliver the product features.
- Participating in all Scrum events.



All of the members in Scrum interact with each other via **Scrum artifacts/events**. Scrum artifacts are essential components of the Scrum framework that capture information about the product and the work being done. They provide transparency and opportunities for inspection and adaptation. Here's a categorized list of the primary Scrum artifacts:



"Project Wall" and "Team Board" are visual tools used in project management and team collaboration. They help in tracking work, visualizing progress, and facilitating communication among team members.

Project Wall is typically used to provide an overview of an entire project. It gives a general view of the project's status, milestones, key deliverables, and major tasks.

Content:

- High-level tasks or features.
- Milestones or key dates.
- Important project-related information, such as risks, dependencies, and stakeholders.
- Overall project progress indicators.

Audience:

- Suitable for both team members and external stakeholders.
- Provides a quick snapshot for senior management, clients, or other departments to understand the project's status.

Usage:

- Used for project planning, tracking major milestones, and monitoring project health.
- Might not be updated as frequently as a Team Board.

Team Board (Scrum Board or Kanban Board) is more granular and focuses on the day-to-day tasks and activities of a team for each sprint.

Content:

- Backlog items, detailed tasks, and user stories.
- Columns representing different stages of task progress ("To Do," "In Progress," "Done").
- Information about task ownership, deadlines, and blockers.

Audience:

- Primarily for the team members to manage and track their daily work.
- Facilitates daily stand-ups or other team meetings.

Usage:

- Used for daily task tracking, workload management, and identifying bottlenecks.
- Frequently updated as tasks move from one column to another.

Some of the universal artifacts are:

1. Product Backlog:

- Description: The Product Backlog is an ordered list of everything that is known to be needed in the product. It is the single source of product requirements.
- Responsibility: The Product Owner is responsible for the Product Backlog, including its content, availability, and ordering.
- Purpose: To provide a clear understanding of the items that can be worked on to achieve the product's goals and vision.

2. Sprint Backlog:

- Description: The Sprint Backlog is a set of Product Backlog items selected for the Sprint, plus a plan for delivering the product Increment and realizing the Sprint Goal.
- Responsibility: The Development Team creates the Sprint Backlog during the Sprint Planning and modifies it throughout the Sprint as needed.
- Purpose: To provide a clear understanding of the work to be done during the Sprint and to offer a plan on how to achieve the Sprint Goal.

3. Increment

- Description: the sum of all the Product Backlog items completed during a Sprint and the value of the Increments of all previous Sprints. It should be in a "Done" state, meaning it's potentially releasable.
- Responsibility: The Development Team is responsible for delivering a potentially releasable Increment of "Done" product at the end of each Sprint.
- Purpose: To provide a tangible output after each Sprint that adds to the overall product value and is potentially shippable.

- 4. Velocity:** a metric used in Scrum to measure the amount of work a team can tackle during a single Sprint. It's calculated at the end of the Sprint by summing up the points for all fully completed User Stories or Product Backlog items.

- 5. Burn-down Chart:** A graph that shows the amount of work remaining in a Sprint

- 6. Burn-up Chart:** A graph that shows the amount of work completed over time.

Something that I don't quite understand clearly in the simulation:

1. What is the role of a Scrum Master, and how is it different from a traditional project manager in SWE?
2. How long should a Sprint be? It is only 20 minutes in this case so it's quite fast.
3. What happens in a Daily Scrum (or Daily Stand-up) in real-life projects?
4. What is the purpose of the Sprint Review and Sprint Retrospective? Does it take place on the weekends?

Other than that, I think the simulation provides me with a good warm-up on Scrum techniques.

II. Do you think that the simulation will help your project succeed? How?

Participating in the Scrum simulation was a hands-on experience that allowed me to grasp the practical aspects of Scrum, from the iterative nature of Sprints to the collaborative dynamics among the Scrum roles. By actively engaging in the various Scrum events, from Sprint Planning to Retrospectives, I gained a general understanding of how continuous feedback, adaptability, and team collaboration can lead to more efficient problem-solving and product development.

I am aware that this is just a simulation, but it really helps me get to know my team members and how the sprints could actually happen during the next few weeks of the Software Project. The simulation emphasized the importance of transparency, open communication, and different Scrum roles involvement, all of which are critical for aligning project goals with user needs and business objectives. Additionally, the simulation also allows the developers and the Scrum master to question the Product Owner about his Product's specifications, which is directly relevant to how we will be interacting with the actual company that commissions our project.

The simulation features the Product Owner who wants to build different facilities in the Arctic for tourism. The products include the Cabin, Sauna, the helicopter, the tractor and vehicle shelter. This is the “**Code Base**” (**Legos**), which developers used to build the products above, which is kinda fun and interactive in my opinion. Except Coding time, developers must not touch Legos.



Therefore, I believe that implementing the principles and practices learned from this simulation can significantly enhance our project's chances of success by ensuring that we remain focused on delivering value, adapting to changes, and fostering a culture of continuous improvement.

III. Was there something surprising / controversial compared to what you have learned about Scrum previously?

Actually no, because this is the first time I'm exposed to Scrum working techniques so I don't have anything surprising or controversial in my own knowledge.

IV. Give feedback on the simulation: What was good/what could be improved?

Good things: our team has managed to build all required products within the 3 sprints. After each sprint, we discuss task specialization, what is still not completed according to the Product Owner, and strategy to boost up coding progress. There is also a useful idea in Scrum that we can cheat as much as possible to get the work done, but don't cheat too much and we will be convicted of crimes (metaphorically). The good things in my opinion are the collaboration between the developers, the fun tasks in building the products from Legos, the sprint concept learning, detailed instructions by Scrum master and interaction with the Product Owner.



Bad things: While this Scrum simulation offers valuable hands-on experience, it probably oversimplifies complex real-world scenarios, leading developers to underestimate potential challenges. The Scrum master revealed that in real-life projects, most often the sprints are not followed strictly and everyone finishes the tasks listed in their own schedules.