

# Sprint to Success

Backlogs & Tasks the Agile Way

# Who?

- It depends who you are in the agile development cycle...
- 1. **Product Owner** - The Product Owner is a key role in agile development and has a significant role in managing the tasks and the backlog.
- 2. **The Team** - The development team collaborates closely with the Product Owner and shares certain responsibilities related to task management and backlog.
- 3. **Scrum Master** - While not directly responsible for task management and the backlog, the Scrum Master supports the team's agile practices.
- For us we essentially combine the product owner and scrum master. Focused on the product and the team.

# 1. Product Owner

- **Backlog Ownership:** The Product Owner is primarily responsible for maintaining the product backlog. This includes adding, prioritizing, refining, and removing items based on business value and customer needs.
- **Requirements:** The Product Owner works closely with stakeholders and users to understand their requirements, and then translates those requirements into user stories or backlog items.
- **Prioritization:** The Product Owner is responsible for determining the order in which backlog items are addressed. This involves considering factors like business value, risk, and dependencies.
- **Acceptance Criteria:** The Product Owner defines clear and concise acceptance criteria for each backlog item, helping the development team understand the expectations for delivering value.
- **Stakeholder Communication:** The Product Owner ensures effective communication between the team and stakeholders. They gather feedback, answer questions, and provide clarifications on backlog items.

## 2. The Team

- **Task Ownership:** While the Product Owner owns the overall backlog, the development team takes ownership of the tasks within each sprint. They break down user stories into actionable tasks during sprint planning.
- **Estimation:** The development team estimates the effort required for each task or user story. This helps with planning and ensuring that the team takes on a manageable amount of work during the sprint.
- **Task Management:** The development team manages the movement of tasks on the sprint board or Kanban board. They update the status of tasks as they progress from "To Do" to "In Progress" to "Done."

## 2. The Team

- **Daily Standups:** In the daily standup meetings, each team member shares their progress, challenges, and plans for the day. This helps the team stay aligned and identify any impediments.
- **Collaboration:** The development team collaborates with the Product Owner to clarify requirements, provide feedback on feasibility, and ensure that user stories are well-defined.
- **Continuous Improvement:** The development team participates in sprint retrospectives to reflect on the process, identify areas for improvement, and suggest changes to the way tasks are managed.

### 3. Scrum Master

- **Process Facilitation:** The Scrum Master helps facilitate sprint planning, daily standups, sprint reviews, and retrospectives to ensure that the team follows the agile process effectively.
- **Removing Impediments:** The Scrum Master works to remove any obstacles that the team encounters during the sprint, enabling smooth task execution.
- **Continuous Improvement:** The Scrum Master encourages the team to adopt continuous improvement practices and adapt their processes based on feedback and retrospective findings.

# Acceptance Criteria

- **Be Specific:** Avoid vague language. Use clear and precise wording to describe the expected behaviour or outcome of the task.
- **Testable:** Each criterion should be testable. It should be possible to determine whether the task has been successfully completed based on the criteria.
- **Measurable:** Include measurable aspects to determine when the task is complete. This could be in the form of quantitative metrics or observable behaviour.
- **User-Centric:** Express the criteria from the user's perspective. Consider how the user will interact with the feature or functionality.
- **Functional and Non-Functional Requirements:** Address both functional aspects (what the system should do) and non-functional aspects (performance, security, usability, etc.).
- **Clear Boundaries:** Define any constraints or boundaries within which the task needs to function. This can help prevent scope creep.

# Acceptance Criteria

- **Dependencies:** If the task relies on other tasks or conditions, mention those dependencies in the acceptance criteria.
- **Conditions of Satisfaction:** Outline the conditions under which the user or stakeholder would consider the task to be successfully completed and providing value.
- **Avoid Implementation Details:** Focus on the "what" rather than the "how." Leave room for the development team to decide the best technical approach.
- **Review and Refine:** Collaboratively review and refine acceptance criteria with stakeholders and the development team to ensure they accurately capture the required functionality.
- **Use Examples:** Provide examples to illustrate the desired behaviour. Examples help clarify expectations and reduce ambiguity.
- **Keep It Concise:** Avoid unnecessary details or overly complex wording. Keep the criteria concise and easy to understand.



# Example – User Story

As a **customer**, I want to **track my order status** so that I can **stay informed about the progress of my delivery**.

# Acceptance Criteria

1. The "Order History" link is **prominently** displayed in the user's account menu.
2. Clicking the "Order History" link redirects the user to a page displaying a list of their recent orders.
3. Each order in the list includes the order number, date of purchase, and total amount.
4. When the user clicks on an order from the "Order History" list, they are taken to the order details page.
5. The order details page displays the order status (e.g., "Processing," "Shipped," "Delivered"), shipping address, and a summary of the items purchased.
6. If the order is shipped, a tracking number is provided with a link to the carrier's tracking page.
7. If there is an issue retrieving the order status, an error message is displayed, and the user is prompted to try again later or contact support.
8. If the tracking number is invalid, the system displays a message advising the user to check back later or contact support

# Exercise: User stories

In your groups take the following user stories and create example acceptance criteria for the following examples. This is a theoretical exercise take creative liberties and highlight any scenario that you want:

- 1. As a user, I want to reset my password so that I can regain access to my account.**
- 2. As an administrator, I want to be able to add new users to the system.**
- 3. As a shopper, I want to see related products when viewing an item's details page.**