



# Agile Project Management

**ISEN** | école  
d'ingénieurs  
ALL IS DIGITAL!

---

ISEN Master 1 - 2025

**JUNIA** Grande  
école  
d'ingénieurs

# Questions from last Module?



# **Agile planning, sprint management and deepening Kanban**

---

## Module 2



# Goals of Module 2

1

Learn how to plan a sprint.

2

Understand the concept of velocity and how to estimate User Stories.

3

Manage a Sprint Backlog and track progress with a Kanban board

4

Explore advanced Kanban practices.

# Agenda - Theoretical Part (2h)

## **Sprint Planning (Scrum):**

- Definition and objectives of a sprint.
- Team capacity planning.
- User Stories estimation techniques

## **The Sprint Backlog and velocity:**

- Creation and management of the Sprint Backlog.
- Tracking team velocity.
- Sprint adjustment based on velocity.

## **Advanced Kanban practices:**

- Queue and priority queue management.
- Key Metrics: Cycle Time, Lead Time, Throughput.
- Continuous improvement with Kanban

## **Tracking tools:**

- Using the Kanban board to track progress.
- Common digital tools

# **Sprint Planning (SCRUM)**



Part 1

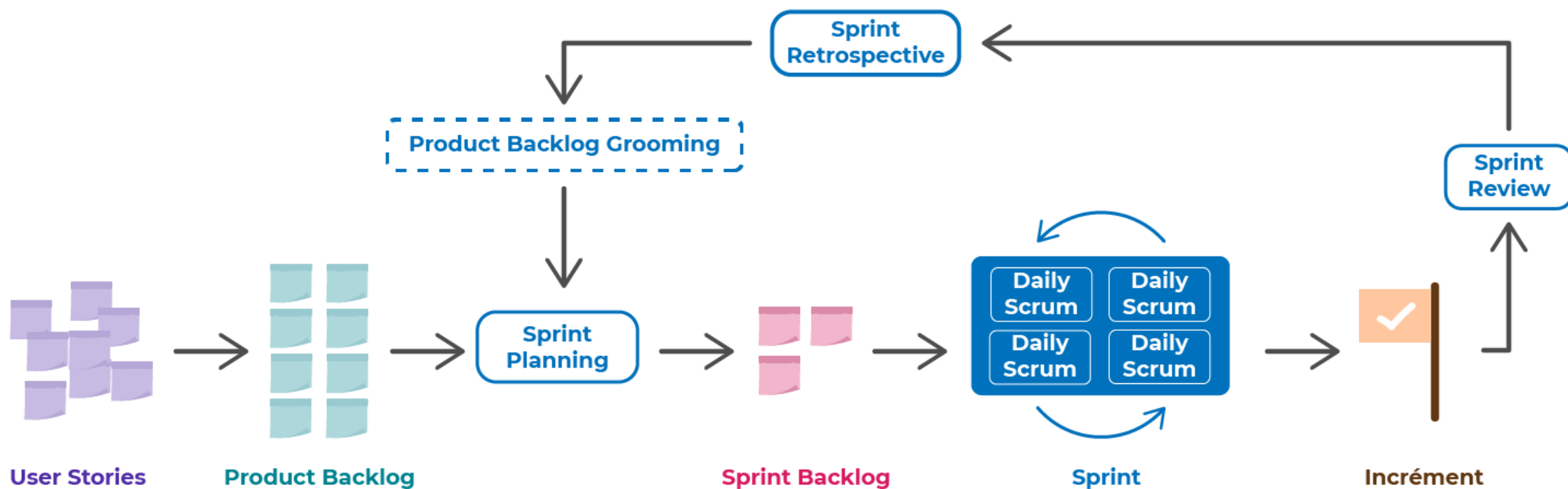
# Sprint Planning (SCRUM)

Definition and objectives of a sprint.

Team capacity planning.

User Stories estimation techniques

# Scrum



Itération de 1 à 4 semaines





## Let's start a sprint!

- Box time from 1 to 4 weeks (maximum).
  - This is the period during which all product features will be developed and incremented.
- The duration of the sprints is **the same** throughout the duration of the project.
- The number of sprints depends on the team's ability to cover needs.
- A new sprint begins as soon as the previous one is completed.
- The decision to cancel a sprint or end it prematurely can only be made by the Product Owner.

# What is a Sprint?

Every SPRINT

starts with \_\_\_\_\_

and ends with \_\_\_\_\_

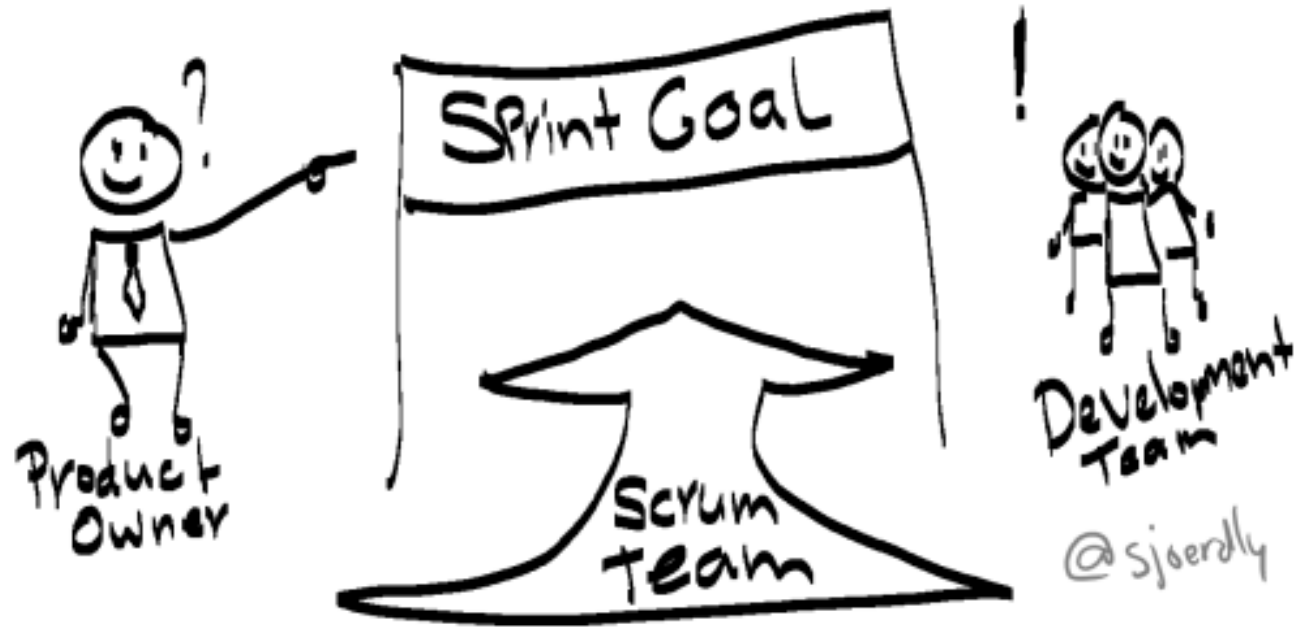
and \_\_\_\_\_

# What is a Sprint?

Every SPRINT  
starts with a Sprint planning meeting  
and ends with a Sprint review  
and a Sprint Retrospective.

# Sprint Goal

- The Sprint Goal is an overall goal that the team strives to achieve during the sprint.
- It provides clear direction and helps with decision-making during the sprint
- For example, if a task does not contribute to this objective, it can be postponed.



# Assess Team Capacity

Team capacity is the amount of work it can accomplish within a sprint.



## Team availability:

- Take into account planned absences (vacation, days off, etc.).
- Take into account time not available for development (meetings, training, etc.).

## Historical capacity:

- If the team has worked together before, look at their velocity over previous sprints to get an estimate of capacity.

## Member expertise:

- Identify members who have expertise in critical areas or specific tasks, and adjust the workload based on their skills.

# Align workload with capacity

## Balancing tasks

- Ensure that the tasks in the Sprint Backlog are distributed fairly among team members, taking into account each person's skills and availability.

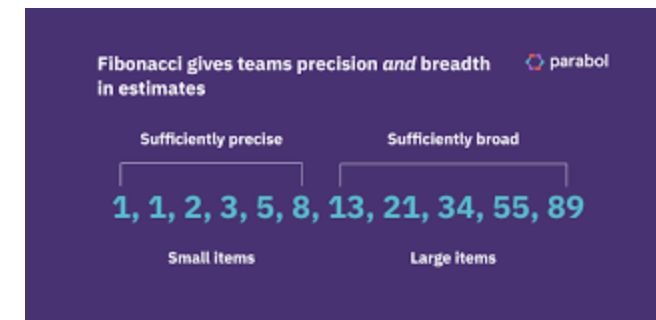
## Limiting commitments

- The team should avoid overestimating what it can accomplish, ensuring that it only takes on tasks that can reasonably be completed in the allotted time.

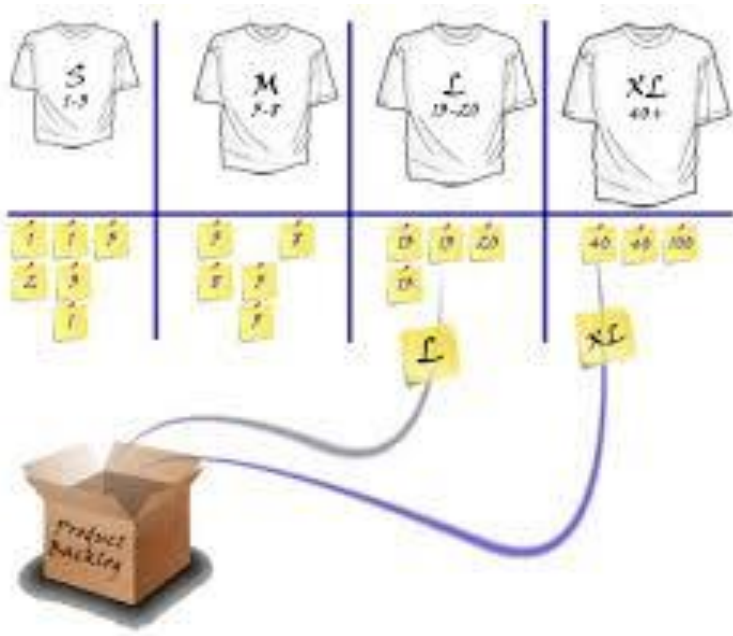
# Planning Poker



- Planning Poker is a popular and collaborative estimation method where each team member gives their estimate of the effort needed to complete a User Story in terms of effort points (often Story Points).
- Here's how it happens:
  - Each team member receives cards with numbers (often based on the Fibonacci sequence: 1, 2, 3, 5, 8, 13, 21, etc.).
  - A User Story is read and explained to the team.
  - Everyone chooses a card in secret and reveals it at the same time as the others.
  - If the estimates differ significantly, a discussion begins to better understand the complexity of the task.
  - The process is repeated until the team reaches consensus on an estimate.
- Benefits :
  - Encourages discussion and knowledge sharing among members.
  - Considers different perspectives on task complexity.



# T-Shirt Sizing



- This estimation method categorizes User Stories by clothing size categories (XS, S, M, L, XL, etc.), representing the relative size or complexity of the task.
- The steps are as follows:
  - User Stories are presented to the team.
  - The team briefly discusses the estimated size of each User Story and assigns a size (XS for a very simple task, XL for a very complex task).
- This method is quick and intuitive, especially in the early phases of the project, when information is limited.
- Benefits :
  - Simplicity and speed in estimates.
  - Allows you to quickly compare User Stories with each other to get an overview.



# Estimation Based on hours



- Although agility often prioritizes points of effort, it is sometimes necessary to estimate certain tasks in hours.
- Each task in the Sprint Backlog can be broken down into subtasks estimated in hours.
- This ensures that the capacity of the team matches the amount of work to be accomplished in the sprint.
- Use : Often used for more technical tasks or when a need for specific detail is necessary for time management.

# **Sprint Backlog and Velocity**



# Sprint Backlog and Velocity

Creation and management of the Sprint Backlog.

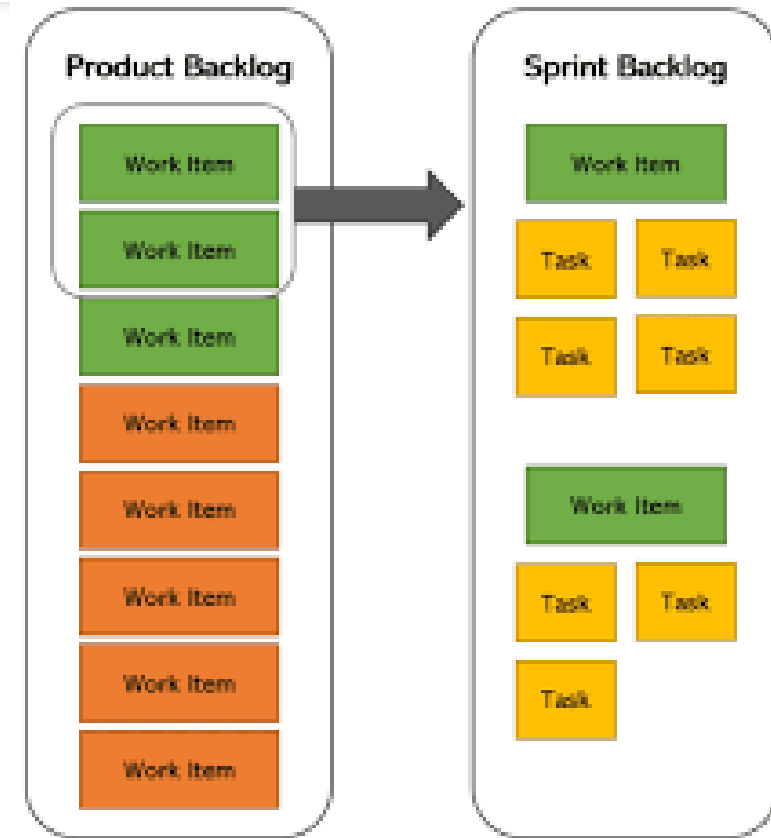
Tracking team velocity.

Sprint adjustment based on velocity.

# What is the Sprint Backlog?

The **Sprint Backlog** is all the elements of the **Product Backlog** selected by the team to be carried out during the sprint.

It also includes the tasks necessary to convert these items into a functional and potentially shippable increment.



# Creation of the Sprint Backlog

## Selection of User Stories

During Sprint Planning, the team selects priority User Stories from the Product Backlog, these being often prioritized by the Product Owner.

The selected User Stories must directly contribute to the sprint objective (the Sprint Goal).

## Breaking down User Stories into tasks

Each chosen User Story is then broken down into smaller, achievable subtasks (usually estimated in hours or days).

This helps clarify the work to be done and facilitates management and monitoring during the sprint.

## Feasibility assessment

The team checks whether the workload associated with the selected User Stories is realistic in relation to its capacity (based on the estimates made).

Complex or ambiguous tasks can be discussed and re-estimated if necessary.

# Management of the Sprint Backlog

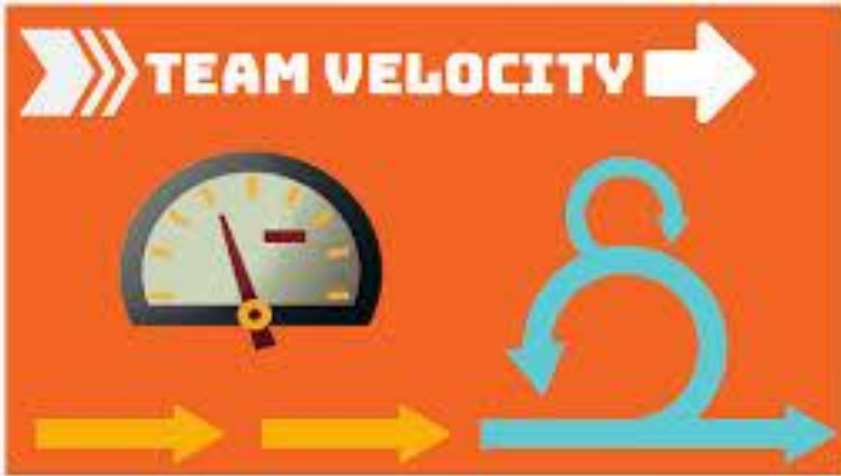
Continuous  
updating

The Sprint Backlog is a living document. If the team discovers additional information or encounters unforeseen events during the sprint, adjustments can be made (adding or removing tasks).

Transparency

All team members should be able to access and view the progress of the Sprint Backlog

# What is Velocity?



- **Velocity** is a metric that indicates the **amount of work** (usually expressed in Story Points) that a team was able to complete during a given sprint.
- It is based on the total number of points for completely completed User Stories at the end of the sprint.

# How to measure Velocity?

- At the end of each sprint, the team adds up the Story Points of User Stories that have been completely completed.
- Incomplete tasks do not count, even if they are almost complete
- This number constitutes the team's velocity for this sprint.
- Example: If the team has completed three User Stories with 3, 5, and 8 Story Points respectively, its velocity for this sprint is 16 points.

$$\text{Velocity} = \text{Nb of Story points completed} / \text{Nb of Sprints}$$



# Why do we use Velocity?

Velocity  
monitoring over  
several sprints

- Velocity must be monitored over several sprints to establish a stable trend.
- A team can thus use the average velocity over the last 3 to 5 sprints to better estimate its capacity during future sprints.

Improve  
predictability

- Velocity allows the team to predict how many Story Points they can likely complete in upcoming sprints.
- It also helps the Product Owner to forecast delivery times for certain features or a product version.

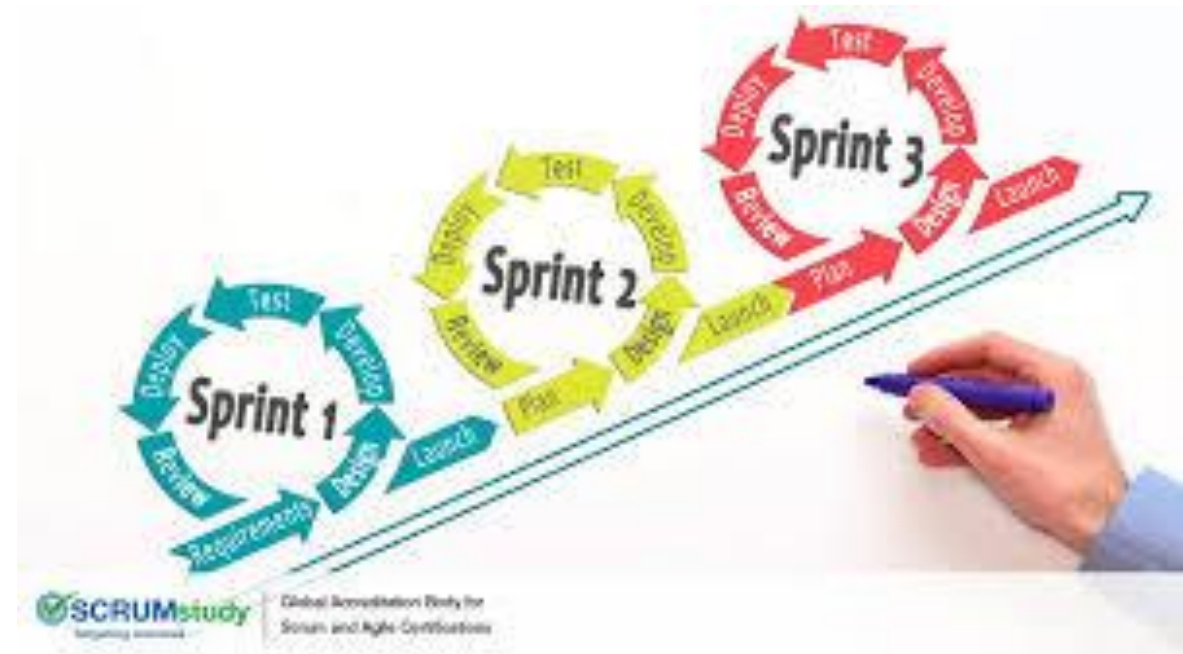
# Adjustment during the Sprint



- Review the workload:
  - If the team notices that it is moving more slowly than expected, it can adjust its priorities during the sprint to focus on the elements most important to the Sprint Goal.
- Manage contingencies:
  - If additional unplanned tasks (like critical bugs) arise, the team can re-evaluate its Sprint Backlog to add these tasks and adjust its capacity accordingly.

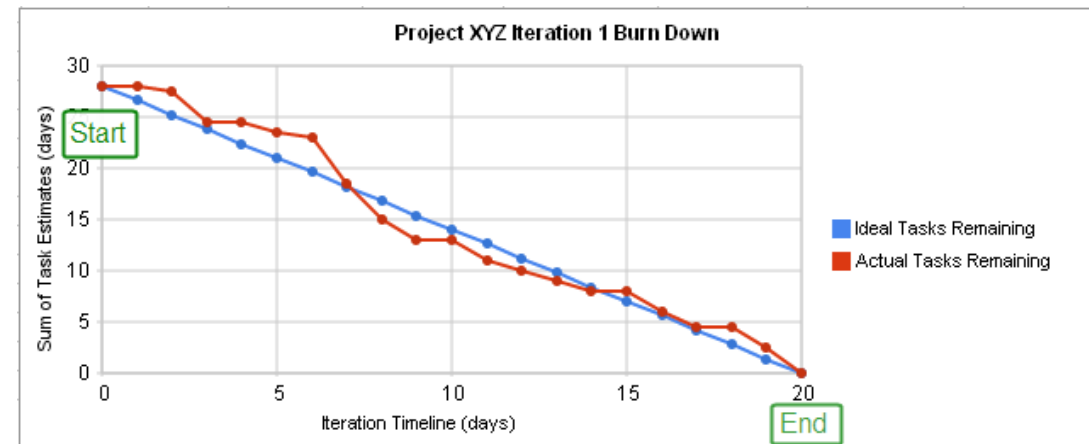
# Adjustment between sprints

- Analyze past velocity:
  - After each sprint, a retrospective analysis allows the team to better understand whether it underestimated or overestimated its capacity.
  - The team can adjust its estimates for future sprints based on its observed velocity.
  - Example: If the team's average velocity is 20 points, the team should take no more than 20 work points from the Sprint Backlog for the next sprint.
- Adapt the Sprint Planning:
  - Based on the performance of previous sprints, the team can readjust its forecasts for Sprint Planning.
  - For example, if the team's velocity decreases because the User Stories are too complex, the team can try to break the User Stories into smaller pieces or allocate fewer points for the next sprint.



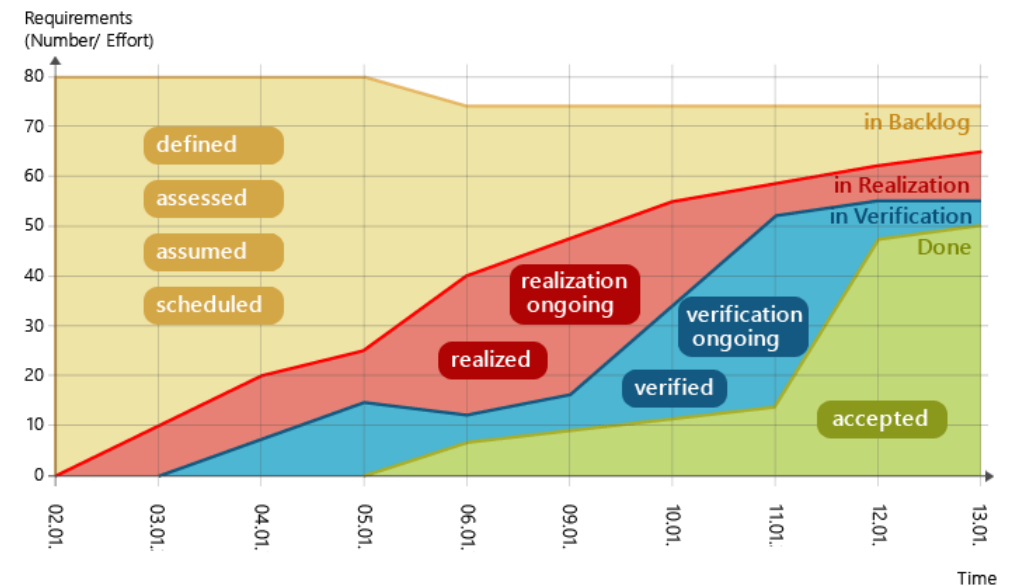
# Tools to adjust planning: Burndown Chart

- This chart shows the progress of remaining work as the sprint progresses. If it becomes clear that the team is not following the planned trajectory, an adjustment may be necessary.
- The vertical axis represents the amount of work left to do (in Story Points).
- The horizontal axis corresponds to the duration of the sprint (in days).



# Tools to adjust planning: Cumulative Flow Diagram (CFD)

- For teams using Kanban, this diagram shows the status of tasks in the different columns of the board (To Do, In Progress, Done).
- If the workflow is stuck, the team must adjust its processes or priorities.



# Advanced Kanban practices



Part 3

# Advanced Kanban practices

Queue and  
priority queue  
management.

Key Metrics:  
Cycle Time,  
Lead Time,  
Throughput.

Continuous  
improvement  
with Kanban  
(Kaizen).

# Queue & Priority Queue

## Queue

- In a Kanban system, queues form when tasks or work items are blocked waiting to be processed by the team.
- They typically occur at stages where necessary resources or skills are limited or when there are external dependencies.
- Example: A task in “Waiting for validation” which cannot progress until it has been approved by a superior.

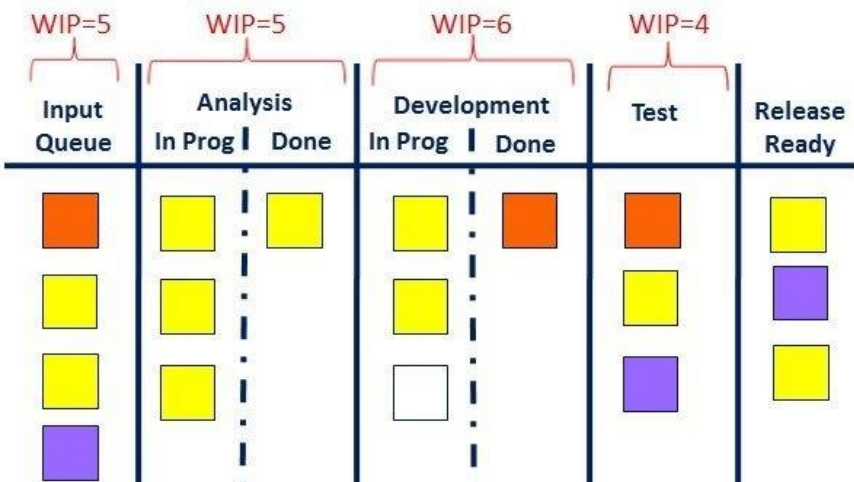
## Priority Queue

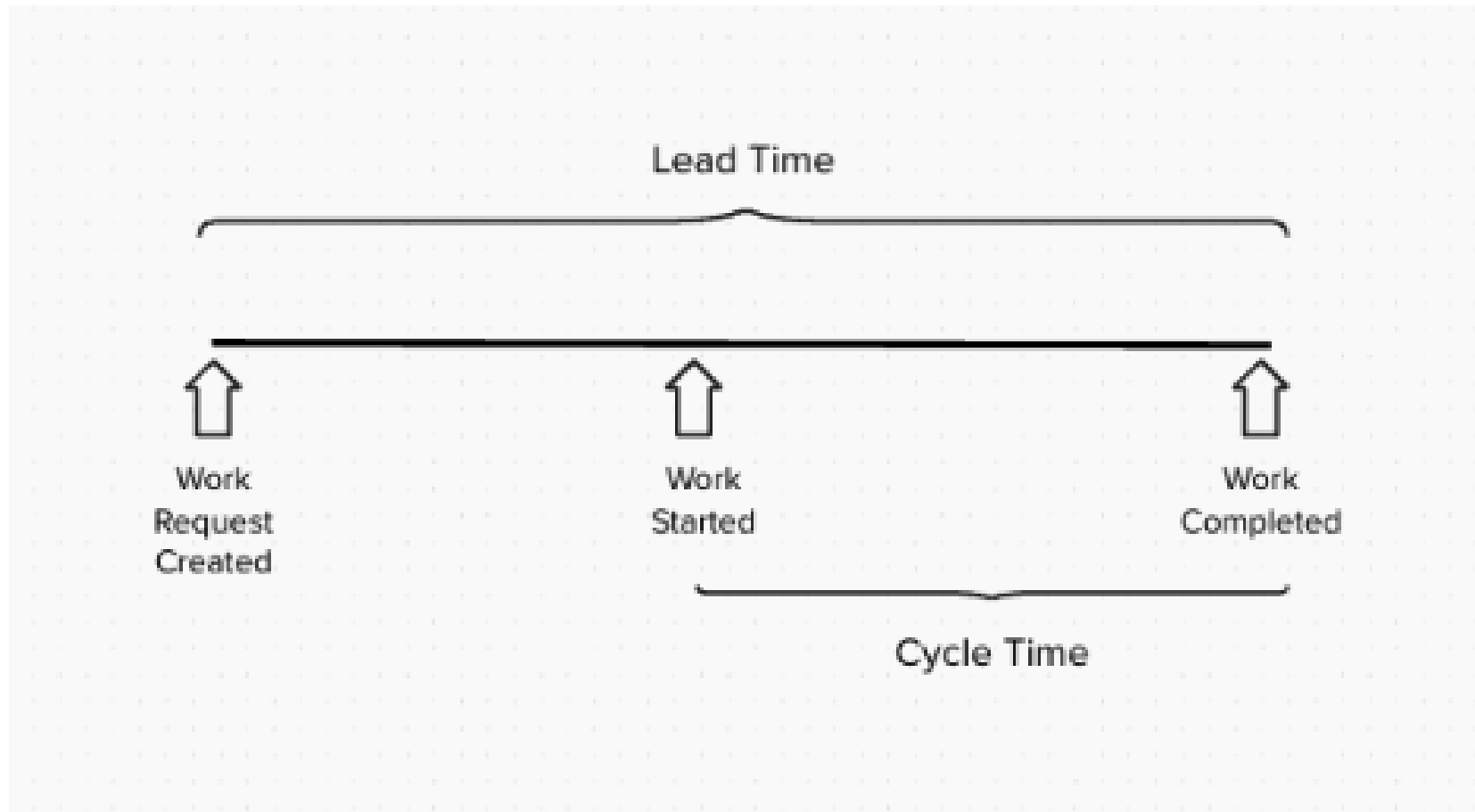
- Priority queues allow you to structure the work to be carried out according to its importance or urgency.
- The highest priority items should be addressed first to maximize the value delivered.
- Example: Critical bugs or urgent customer requests can be moved to the top of the priority queue to be resolved first.



# Best Practices

- Limit queues:
  - Implement **WIP** (Work in Progress) limits to prevent too many tasks from piling up in queues, causing bottlenecks.
  - This helps maintain a smooth workflow and avoids bottlenecks.
- Define clear criteria for priorities:
  - Use Classes of Service to prioritize tasks according to their urgency and impact:
    - Urgent: Must be treated immediately.
    - Fixed date: At a strict deadline.
    - Expeditious: Should be processed quickly, but not with extreme urgency.
    - Routine: Little risk in case of delay.
- Proactive queue management:
  - Regularly observe bottlenecks in the workflow (for example, too many tasks stuck at one step) and adjust priorities or resources to unblock them.





# Cycle Time

## Definition

- This is the time a task takes from the time it begins to be actively worked on ("In Progress") to the time it is completed.

## Importance

- Reducing Cycle Time means that the team is more efficient and tasks are processed more quickly.

## Optimization

- Identify the stages of the process where the Cycle Time is the longest and seek to improve or automate these stages

# Lead Time

## Definition

- The time elapsed between the creation of a task (when it enters the Kanban board) and its completion.

## Difference from Cycle Time

- Lead Time includes the waiting time before work on the task begins, while Cycle Time only covers the duration of active processing.

## Optimization

- Reduce waiting times in queues or priorities.

# Throughput ( le débit)

## Definition

- The total number of tasks completed in a given period (for example, per week or per sprint).

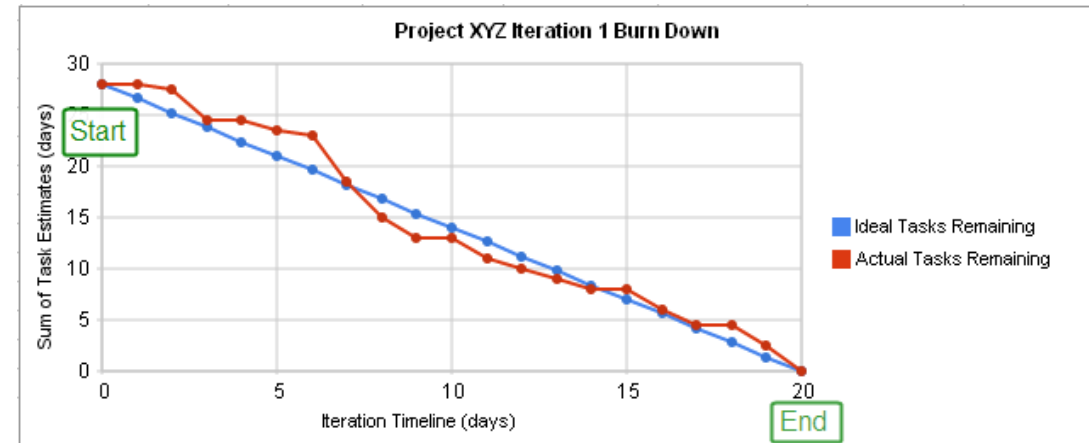
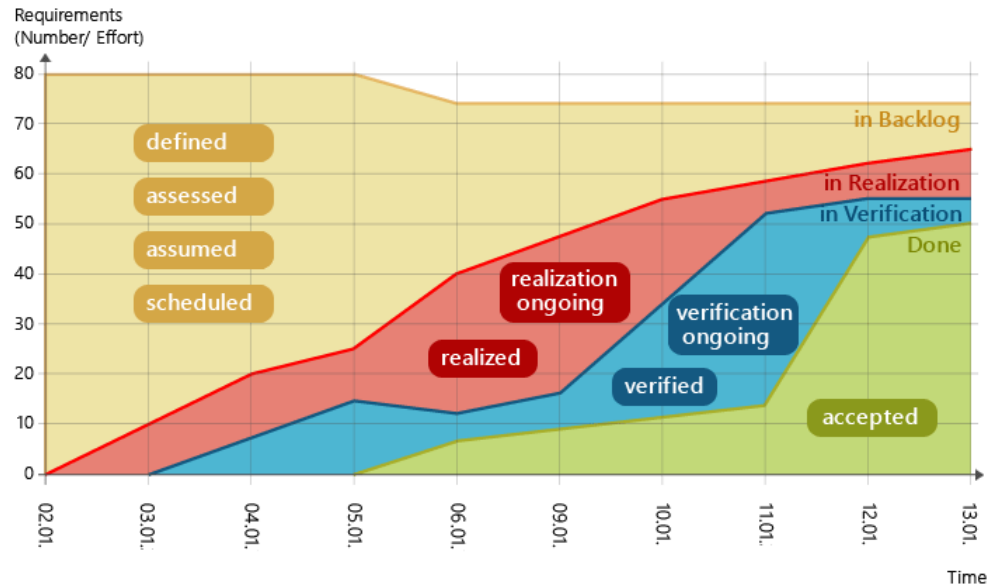
## Importance

- Throughput allows you to measure team productivity and adjust capacity or priorities accordingly.

## Optimization

- Increase Throughput by identifying and reducing bottlenecks or balancing workloads.

# Cumulative Flow Diagram (CFD) and Burndown Chart



# Kaizen

- Kaizen involves making incremental and continuous improvements to processes.
- In Kanban, this means constantly looking for ways to optimize workflow, reduce bottlenecks, and improve the quality of deliverables.

改善

# Some Methods for continuous improvements

- **Regular Kanban Retrospective Meetings:**
  - Although Kanban is a continuous workflow, it is essential to hold periodic retrospectives to discuss challenges, progress, and opportunities for improvement.
- **Experimentation and incremental adjustments:**
  - Make small adjustments (like changing WIP limits, reprioritizing, or changing team roles) and observing their impact on productivity.
- **Focus on waste reduction (Muda means waste in Japanese):**
  - Identify activities that do not provide **direct value** to the customer and which could be eliminated or simplified.
- **Example:**
  - After analyzing the metrics, the team notices that the time spent in the "Under Review" queue is too long. They decide to strengthen the review processes, add resources for this step, or establish clearer validation rules to speed up this phase.



# Tracking Tools



Part 4

# Tracking Tools

Using the Kanban board to track progress.

Common digital tools (Jira, Trello).

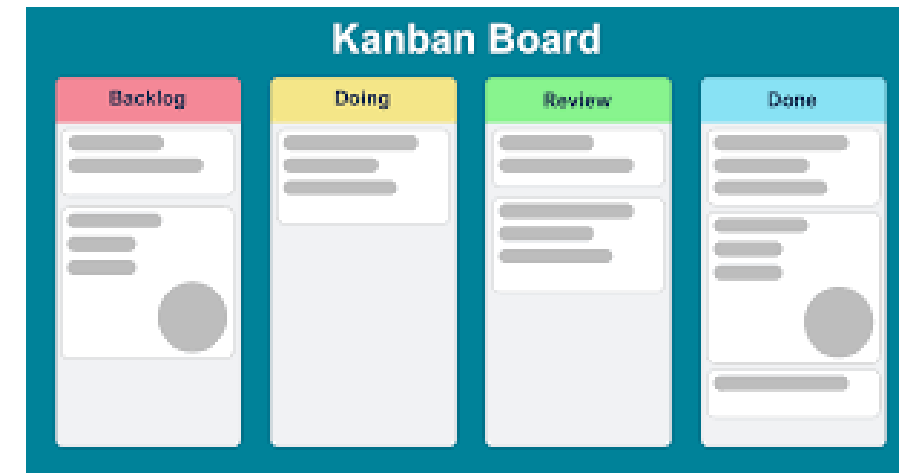
# Physical or Digital Board

- A physical Kanban board is often divided into columns representing different stages of the process (for example: "To do", "In progress", "Done").
- Each task is represented by a card that the team moves from one column to another as progress progresses.
- Advantage: Simple and visual, it allows the entire team to monitor the status of tasks in real time and quickly identify blockages.



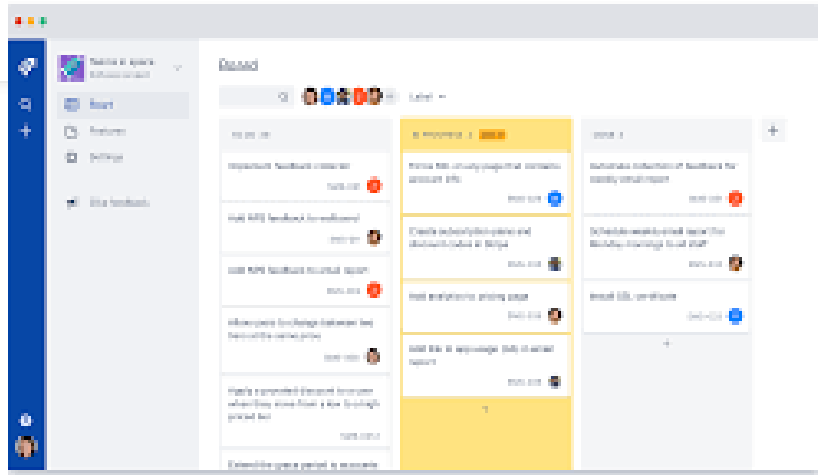
# Optimization of the Board

- Visible WIP Limits:
  - Indicate directly on the table the work in progress limits for each column.
- Categorization:
  - Use labels, colors or classes of service to better organize and visualize priorities

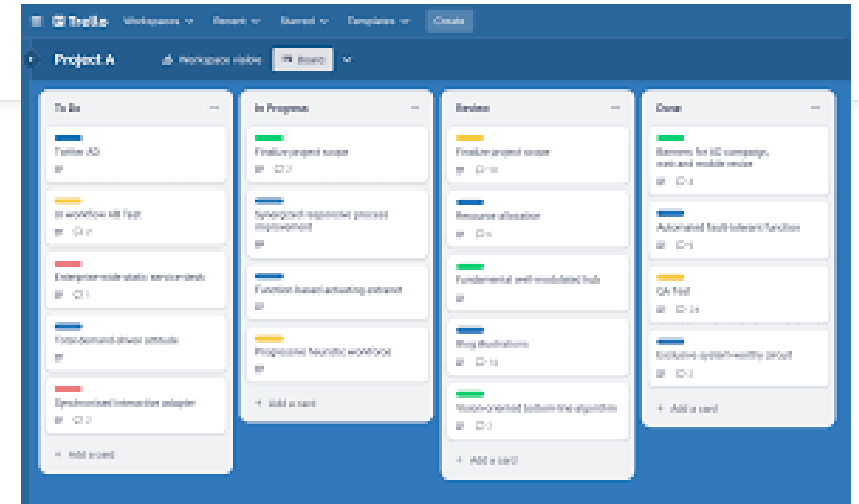


## 4 - Tracking Tools

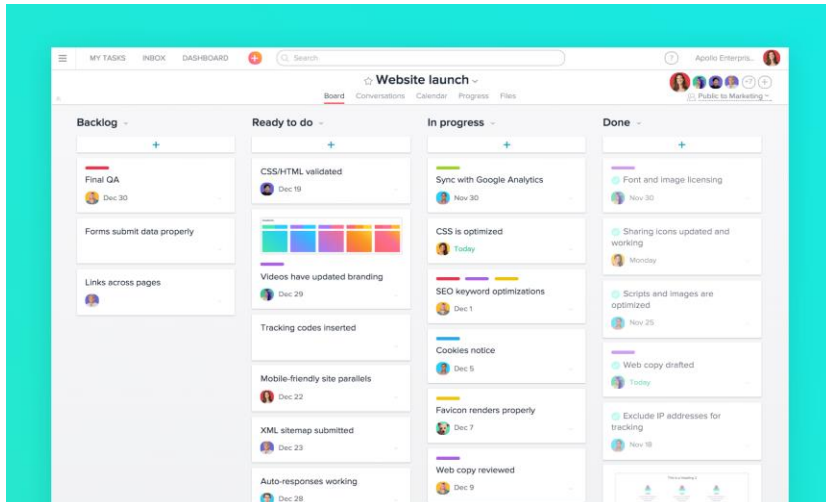
# Common digital tools



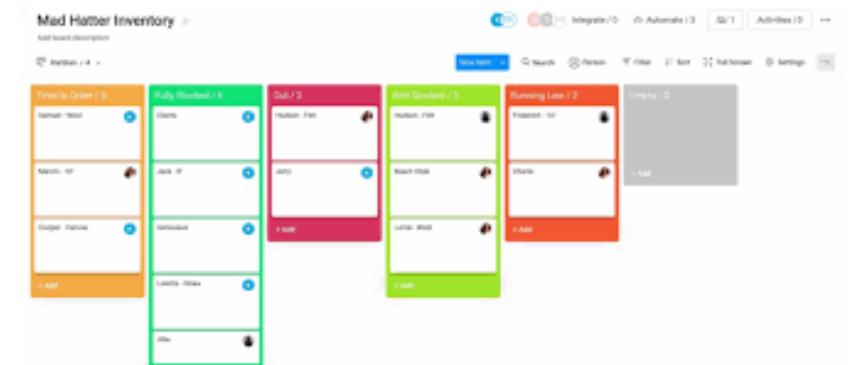
JIRA



Trello



Asana



Monday.com

# Agenda - Practical case(2h)

- **Practical exercise:**

- Simulation of Sprint Planning and Kanban optimization:
  - The pairs review their Product Backlog created during the first session.
  - Estimation of User Stories and prioritization for the next Sprint.
  - Kanban board optimization by integrating WIP limits and using metrics.

# Development of a mobile task management application

- Context: You are part of the development team of a technology start-up called "TaskMaster".
- Your team's mission is to develop a new mobile task management application intended for professionals and students.
- The goal is to create an intuitive application that allows you to efficiently manage daily tasks, plan projects, and collaborate with other users.
- The product should be simple to use, yet powerful enough to accommodate the needs of advanced users.

- **Main features of the application:**

- Task management:
  - Creating, modifying and deleting tasks.
  - Assignment of priorities (low, medium, high).
  - Setting due dates.
  - Ability to add subtasks.
- Project planning:
  - Organization of tasks into projects.
  - Visualization of project progress (e.g. simple Gantt chart).
  - Distribution of tasks between team members.
- Collaboration:
  - Sharing projects with other users.
  - Assigning specific tasks to other users.
  - Discussions/comments on tasks.
- Notifications:
  - Push notifications for imminent or overdue tasks.
  - Customizable reminders.
- Cross-platform synchronization:
  - Real-time synchronization between devices (smartphones, tablets, computers).

- **Additional requirements:**

- The application must be available on iOS and Android.
- The user interface should be simple, modern and intuitive.
- A web version of the application must be provided to allow access via browser.
- The final product must meet data security standards, particularly for personal information and communications between users.



# What is expected from you?

- Sprint planning and setting up a Kanban board
  - Define your "fictive" team and your capacity
  - Organize a Sprint Planning to plan the tasks for the first two-week sprint.
  - Estimate User Stories using techniques like Planning Poker.
  - Create the Sprint Backlog and define the tasks to be carried out.
  - Set up a Kanban board to visualize the progress of tasks.
  - Identify and set WIP (Work In Progress) limits for certain Kanban board columns.
- Explanations on
  - The perceived differences between traditional methods, Scrum, and Kanban.
  - The difficulties encountered during the exercise.
- **PDF document sent by Google Form before today at 01:00 PM in French or in English**