



EURO PIM
GO STUDY EUROPE

Introduction to Agile

Areej Aldaghamin
Jorge Maldonado Carranza



SPONSORED BY THE



Table of Content



1	Introduction to Agile
2	Scrum
3	Kanban
4	Scrumban
5	Case study



The genesis of **Agile** is found in a group of software development methodologies ^[1]

Introduction to Agile: [Introduction to Agile](#)



Agile **tools** and **techniques** can be applied to projects in general ^[1]



Agile methodology is based on **iterative development** with which requirements and solutions evolve through collaborative environment ^[1]



Agile theory is based on the **Agile manifesto** ^[1]



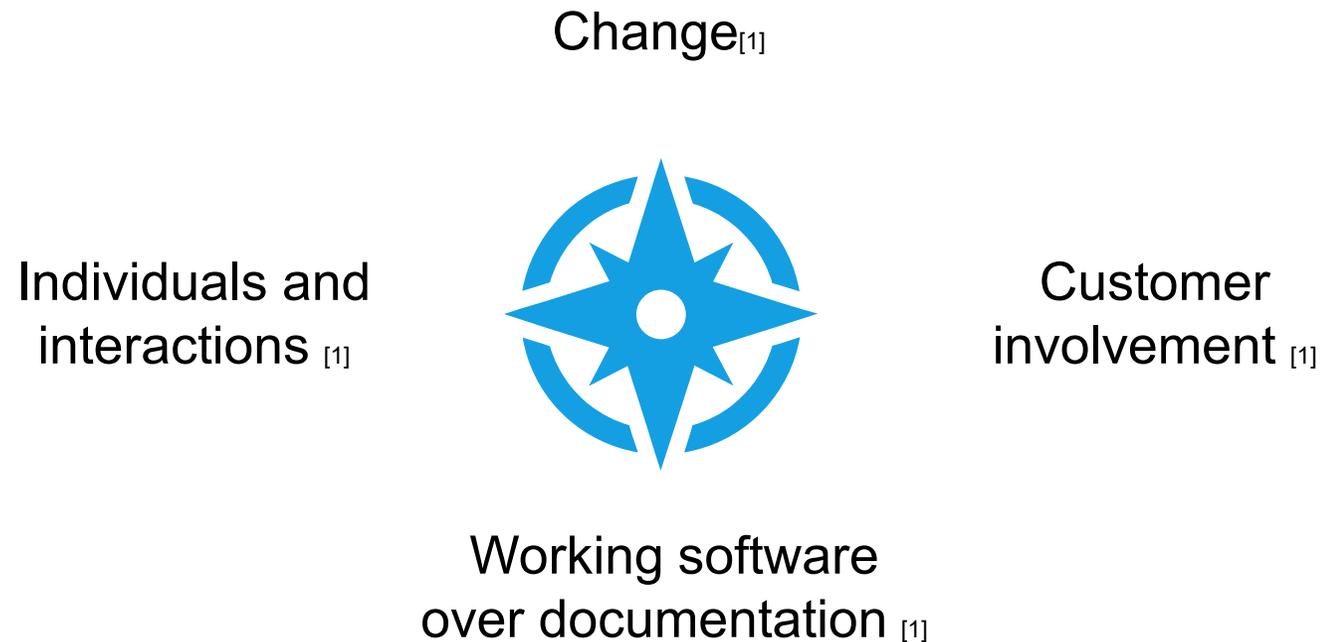
Agile supports **self-organizing** and **cross-functional** teams ^[1]



The **agile manifesto** was written in February 2001 by 17 independent-minded partitioners with mostly programming background [1]

Introduction to Agile: [Introduction to Agile](#)

Practitioners differed on what constitutes agile theory but agreed on **four main values**:



The principles provide a better understanding of the framework and intention of agile methods [x]

Introduction to Agile: [Introduction to Agile](#)

1. Our highest priority is to satisfy the customer [1]
2. We welcome changing requirements even late in development [1]
3. Deliver working software frequently from a couple of weeks to a couple of months with a preference for the shorter time scale [1]
4. Business people and developers must work together daily throughout the project [1]
5. Build projects around motivated individuals [1]
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation [1]

History on how these 12 principles were developed originally remains a bit fuzzy ^[1]

Introduction to Agile: [Introduction to Agile](#)

7. Working software is the primary measure of progress ^[1]
8. Agile processes promote sustainable development ^[1]
9. Continuous attention to technical excellence and good design enhances agility ^[1]
10. Simplicity—the art of maximizing the amount of work not done – is essential ^[1]
11. The best architectures, requirements, and designs emerge from self-organizing teams ^[1]
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly ^[1]

There are several Agile methodologies and certifications [1]

Introduction to Agile: [Introduction to Agile](#)

Some of the most common Agile methodologies include: [1]

Scrum

Feature-Driven
Development (FDD)

Extreme
Programming (XP)

Dynamic System
Development Method (DSDM)

Lean

Crustal

Kanban

Just-in-Time

Recognized entities that provide certifications in agile methodologies include but are not limited to: [1]

Project
Management
Institute

Disciplined
Agile

Platinum
Edge

Agile
Alliance

Scaled Agile
Framework

Scrum
Alliance

The most recognized word when Agile is applied to project management is **Scrum** [1]

Introduction to Agile: **Scrum**



Scrum is a **framework** for developing, delivering, and sustaining complex products [2]



Scrum is **lightweight, simple** to understand but **difficult** to master [2]



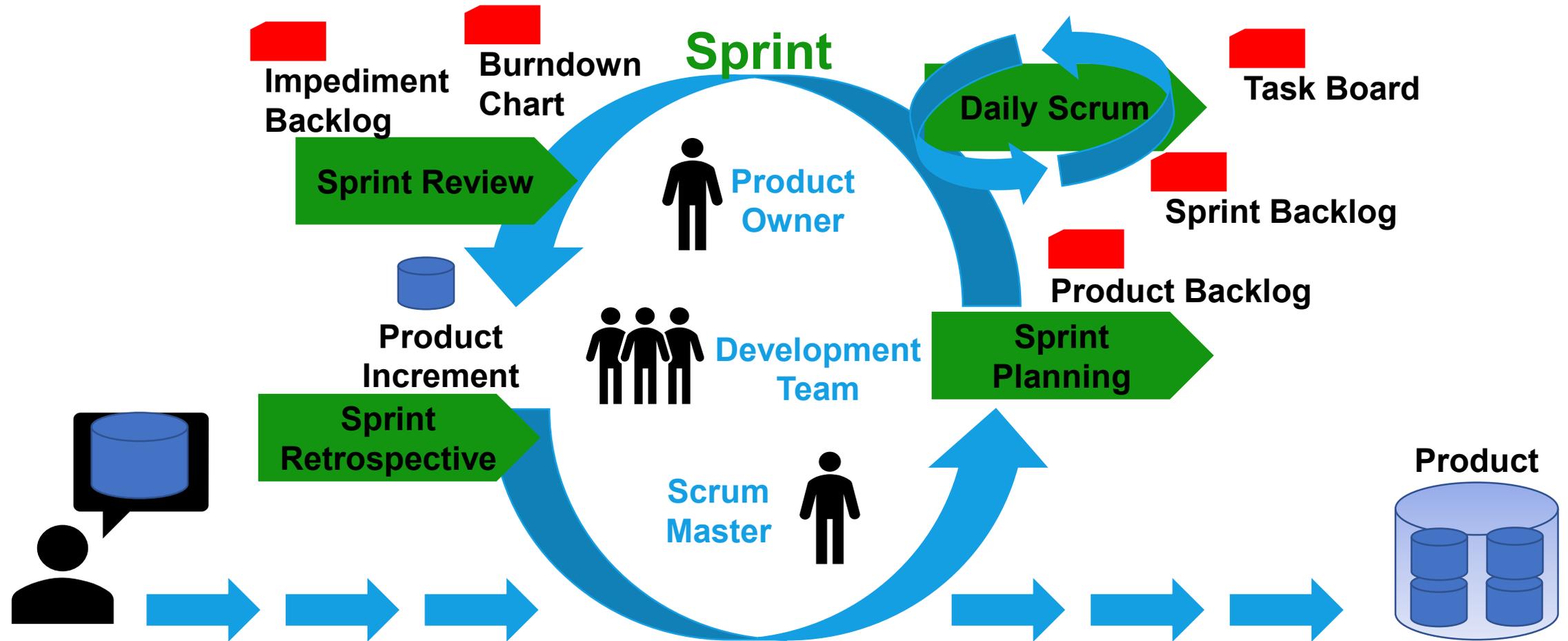
It has been used to manage work on **complex products** since the early **1990s** [2]



It makes clear the relative efficacy of the product management and work techniques to allow **continuous improvement** of the **product, team, and working environment** [2]

The Scrum framework involves **roles**, **artifacts**, and **events** [2]

Introduction to Agile: **Scrum**



Scrum teams deliver products iteratively and incrementally, maximizing opportunities for feedback [2]

Introduction to Agile: [Scrum](#)

Role

Description

Product Owner	Responsible for maximizing the value of the product resulting from work of the Development Team. It is the sole person responsible for managing the Product Backlog. The Product Owner is a person, not a committee [2]
Scrum Master	Is responsible for promoting and supporting Scrum. They help everyone understand Scrum theory, practices, rules, and values. The Scrum Master is a servant-leader for the Scrum Team. The Scrum Master helps those outside of the Scrum Team understand which of their interactions with the Scrum Team are helpful and which aren't" [2]
Development Team	Consists of professionals who do the work of delivering a potentially releasable Increment of "Done" product at the end of each Sprint. Only members of a Development Team create the Increment. They are self-organized, cross-functional, and with no distinction in their titles. The size of the team goes from 3 to 9 persons [2]

Artifacts are specifically designed to maximize transparency of key information [2]

Introduction to Agile: Scrum

Artifact	Description
Product Backlog	The Product Backlog is an ordered list of everything that is known to be needed in the product, and the Product Owner is responsible for it [2]
Sprint Backlog	Is the set of Product Backlog items selected for the Sprint, plus a plan for delivering the product increment and realizing the Sprint Goal [2]
Task Board	The overview of the individual tasks for developing functions can be displayed on a Task Board. This can be a flip, an excel sheet, or post-it note on the Scrum board with the heading “task”

Scrum's Artifacts represent work or value to provide transparency and opportunities for inspection and adaptation [2]

Introduction to Agile: Scrum

Artifact	Description
Burndown Chart	Graphical control and steering instrument for the individual Sprints. It visualizes the work done and the work still to be done on a given date. The display usually takes the form of story points
Impediment Backlog	Is the documentation of the obstacles during a Sprint in the form of a list. The obstacles refer to anything that prevents a team from completing the tasks. It can be a list on a flip, Excel sheet, a ticket in a task management tool, or a post-it note on a scrum board titled "Impediments"
Product Increment	An increment is the result of a sprint (partial delivery object). It comprises the functions and characteristics to fulfill the user stories of the Sprint. At the end of a Sprint, the increment must be in a working state and meet the team's definition of "Done" [2]

All events are time-boxed, such that every event has a maximum duration [2]

Introduction to Agile: Scrum

Event	Description	Time Boxing
Sprint	A time box during which a “Done”, usable and potentially releasable product increment is created [2]	Four weeks
Spring Planning	The work to be performed in a Sprint is planned here and it is done by the collaborative work of the entire Scrum Team [2]	At the beginning of Sprints 8h
Daily Scrum	A Daily Scrum is held every day of the Sprint, there the Development Team plans work for the next 24 hours [2]	Daily 15 min
Sprint Review	Held at the end of the Sprint to inspect the Increment and adapt the Product Backlog if needed, feedback from customer [2]	End of Sprints 4h
Sprint Retrospective	Reflection of the current sprint, there improvements are identified, planned and implemented [2]	End of Sprints 3h

Kanban is a popular framework used to implement agile and DevOps software development [3]

Introduction to Agile: [Kanban](#)



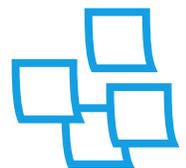
Kanban requires **real-time** communication of capacity and full transparency of work [3]



Work items are represented **visually** on a **Kanban board**, allowing team members to see the state of every piece of work at any time [3]



Kanban methodology of work dates back more than 50 years, in the late **1940s Toyota** optimized their process based on the **supermarket model** used to stock their shelves [3]



Toyota applied this system to its factory floor to align their massive inventory levels with the actual consumption of materials and to communicate capacity levels on real time workers would pass a card or “**Kanban**” between teams [3]

Agile development teams today can leverage these same JIT principles by matching the **WIP** to the team's capacity [3]

Introduction to Agile: [Kanban](#)

Taking care of the work in progress (WIP) limits gives team more **flexible planning options**, **faster output**, **clearer focus**, and **transparency** throughout the development cycle [3]

Makes it easier to **identify inefficiency** in a team's workflow [4]

Sets the **maximum amount of work** that can exist in each status of a workflow [4]



WIP limits

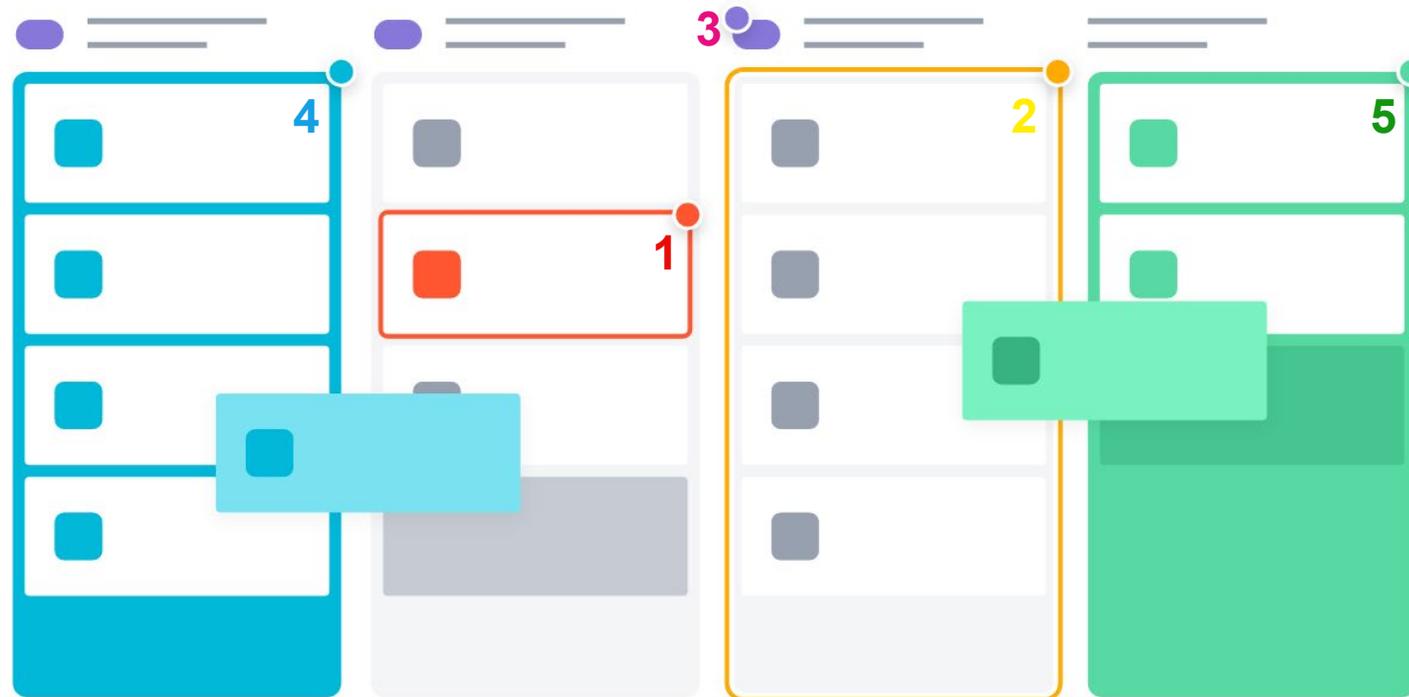
Points out areas of **chronic idleness** or **overload** [4]

Allows to identify **bottlenecks** [4]

Encourage a culture of **“done”** [4]

The work of all Kanban teams revolves around a Kanban board, a tool used to visualize work and optimize the flow of the work among the team [5]

Introduction to Agile: Kanban



Kanban boards can be broken down into five components [5]

1. Visual signs – Visual cards, stickies, tickets, etc. [5]

2. Columns – Each column represent a specific activity from the “workflow” [5]

3. Work In Progress (WIP) Limits – Maximum number of cards that can be in one column at any given time [5]

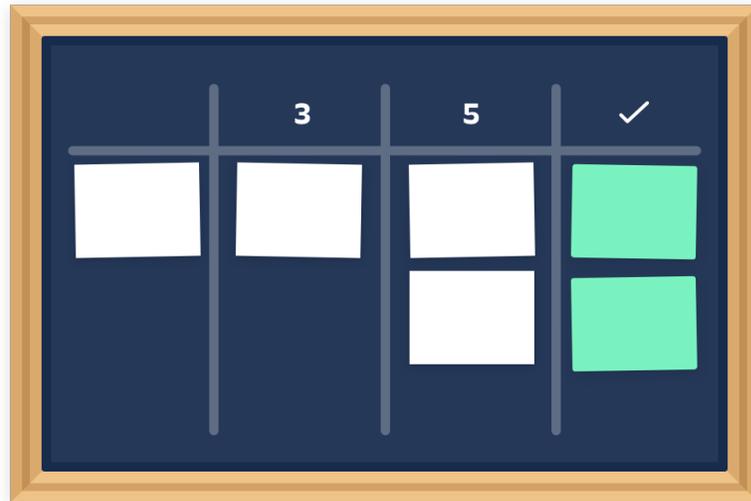
4. Commitment point – When an idea is picked up by the team and work starts on the project [5]

5. Delivery point – The end of the Kanban team’s workflow [5]

The type of **environment** adapting Kanban often dictates if the board is **physical** or **digital** [5]

Introduction to Agile: **Kanban**

Physical Kanban Board



The **simplest** Kanban board is the **physical** one, divided into vertical columns with sticky notes that move throughout the workflow [5]

Digital Kanban Board

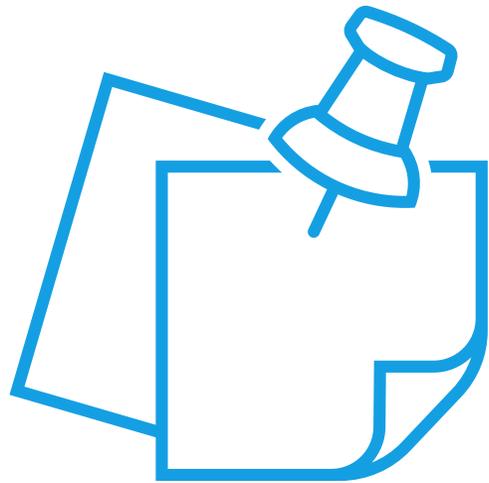


Through digital transformation, Kanban boards become **digital**, allowing teams that do not share a physical office space to use a board **remotely** and **asynchronously** [5]

In Japanese, **Kanban** literally translates to "**visual signal**", for Kanban teams, every work item is represented as a separate card on the board [3]

Introduction to Agile: **Kanban**

The **main purpose** of representing work as a card in a board is to allow team members to **track progress** of work through its workflow in a visual manner [3]



A **Kanban card** features **critical information** about a work item giving the entire team;



Visibility of who is responsible for that item [3]



A brief description of the work being done [3]



How long that piece of work is estimated to take [3]



And so on... [3]

Kanban offers several additional advantages to task planning and throughput for teams of all sizes [3]

Introduction to Agile: [Kanban](#)

Planning flexibility

The Kanban team is only focused on the work that's actively in progress. The product owner is free to reprioritize work in the backlog without disrupting the team [3]

Shortened time cycles

Cycle time is a key metric for Kanban teams, being the amount of time it takes for a unit of work to travel through the workflow. By optimizing it, a team can confidently forecast the delivery of future work [3]

Fewer bottlenecks

WIP limits highlight the bottlenecks and backups in the team's process due to lack of focus, people, or skill sets [3]

Kanban is one of the most popular software development methodologies adopted by agile teams today [3]

Introduction to Agile: [Kanban](#)

Visual metrics

Charts provide a visual mechanism for teams to ensure they're continuing to improve, when a team can see data it's easier to spot bottlenecks in the process and remove them. Two common reports used are control charts and cumulative flow diagrams [3]

Continuous delivery

Continuous delivery is the practice of releasing work to customers frequently, focusing on a just-in-time and one-at-a-time delivery of value. Kanban teams focus on optimizing the flow of work out to customers [3]

The difference between Kanban and Scrum is quite subtle [5]

Introduction to Agile: Kanban

By most interpretations, Scrum Teams use a Kanban board with Scrum Processes, Artifacts and Roles along with it [5]

Some key differences between Scrum and Kanban include:

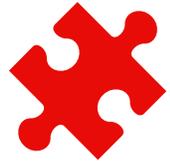
- Scrum sprints have start and stop dates whereas Kanban is an ongoing process [5]
- Team roles are clearly defined in Scrum while Kanban has no formal roles [5]
- A Kanban board is used throughout the lifecycle of a project whereas a Scrum board is cleared and recycled after each sprint [5]
- A Scrum board has a set number of tasks and strict deadline to complete them [5]
- Kanban boards are more flexible with regards to tasks and timing since they can be reprioritized, reassigned, or updated as needed [5]



Some teams **blend the ideals** of Kanban and Scrum into “**Scrumban**” [3]

Introduction to Agile: **Scrumban**

For **Scrumban** they...[3]



Take fixed-length **sprints** and **roles** from **Scrum** [3]



Focus on **work in progress limits** and **cycle time** from **Kanban** [3]



Case Study

Introduction to Agile: **Case Study**

Prepare prints of the teams (mixed universities)

- Get to know your team (check the your team name on your table and head to your team area)
- Start to prepare your team methodology
- Create and prepare your team roles
- Create and prepare your project concept and work strategy
- Create your backlog and prepare your board.
- Create the concept of the agility of modules.
- Prepare the communication methods using the tools provided (teams, Jira...)
- Prepare and schedule your artifacts (Daily Scrum etc) starting from the **Sprint Planning**
- **On Thursday please prepare your **Sprint Review** meeting with all of us (15 min)**

EURO PIM

GO STUDY EUROPE

areej.aldaghamin@fh-dortmund.de

jorge.maldonadocarranza@fh-dortmund.de



SPONSORED BY THE



Co-funded by the
Erasmus+ Programme
of the European Union



Case Study

“Module Development within OpenCoPs”

References

- [1] Terra Vanzant Stern, P. (2020). Lean and Agile Project Management : How to Make Any Project Better, Faster, and More Cost Effective, Second Edition: Vol. 2nd ed. Productivity Press.
- [2] Schwaber, K. and Sutherland, J., (2017). The Scrum Guide - The definitive Guide to Scrum: The Rules of the Game. [online] Scrumguides.org. Available at: <<https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>>.
- [3] <https://www.atlassian.com/agile/kanban>
- [4] <https://www.atlassian.com/agile/kanban/wip-limits>
- [5] <https://www.atlassian.com/agile/kanban/boards>