PRODECYS

ENLIGHTEN YOUR TRAINING

INTRODUCTION TO AGILE

TABLE OF CONTENTS

PRODECYS

ENLIGHTEN YOUR TRAINING

WHAT IS AGILE?

WHAT IS AGILE?

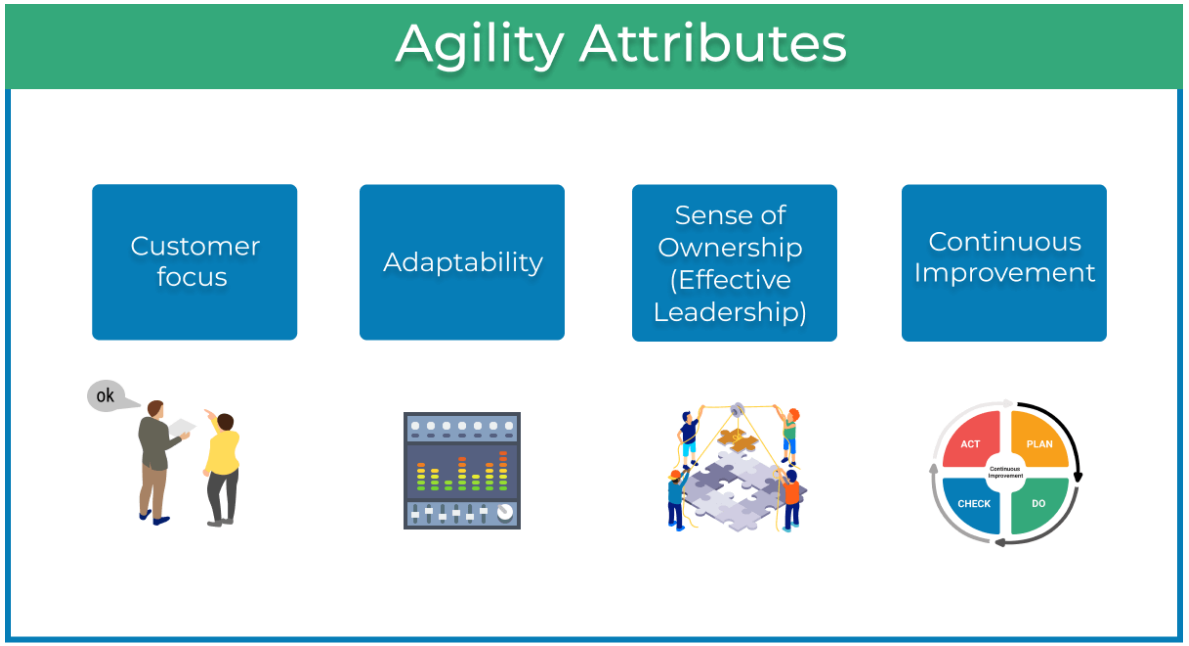
Agile Project Management is a flexible and iterative approach to delivering projects and products.
It emphasizes on :

- collaboration,
- customer satisfaction,
- adaptation to changing requirements.

Work is divided into small increments, or sprints, and progress is reviewed and adapted regularly.

The Agile approach values individuals and interactions, working software, customer collaboration, and responding to change, over following a strict plan.

WHAT IS AGILE?

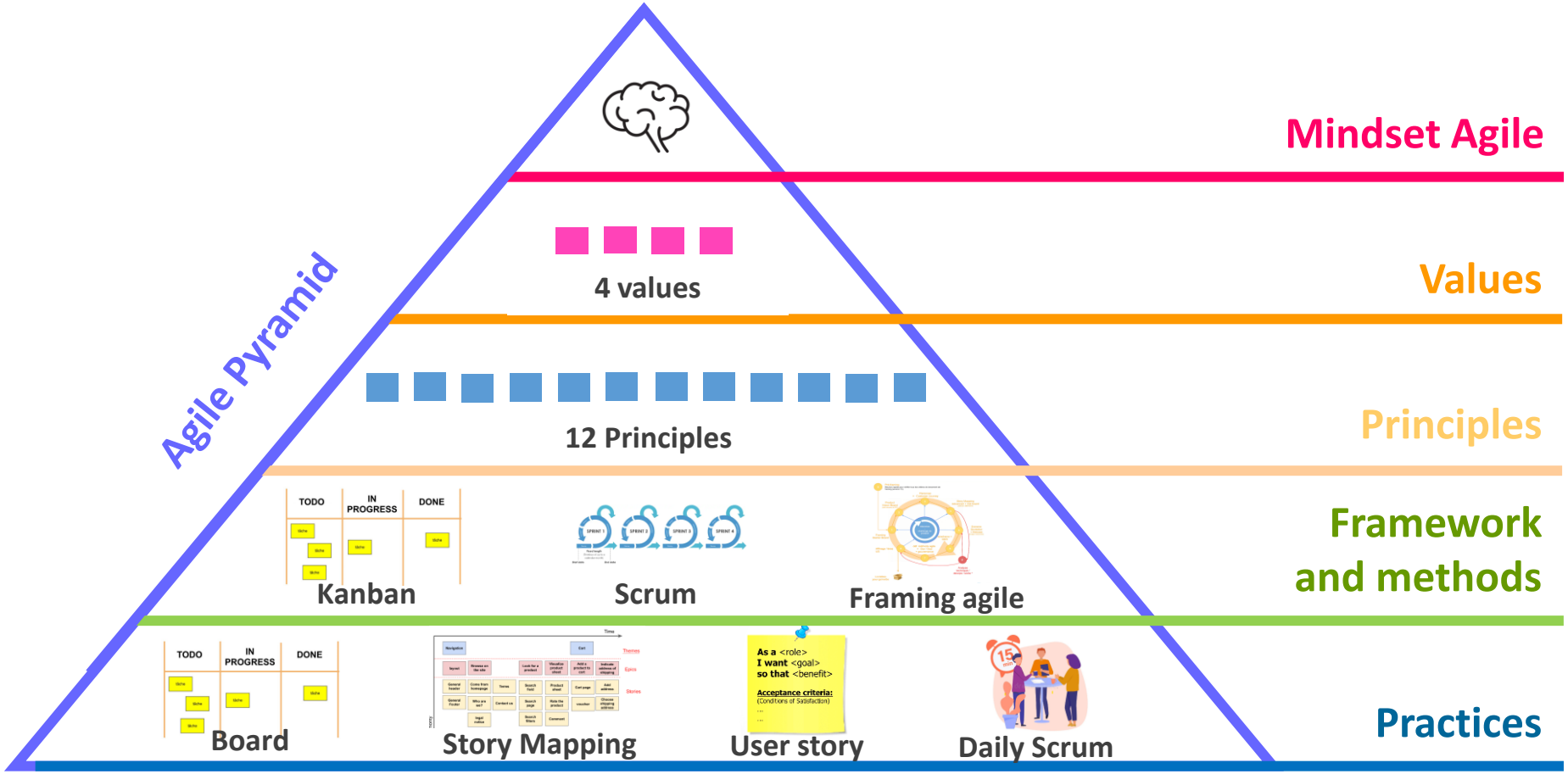


- Agile project management is commonly used in software development but can be applied to any type of project.

- Key principles of Agile project management include :
  - continuous improvement
  - Transparency / Effective Leadership
  - Adaptability
  - delivering value to the customer

WHAT IS AGILE?

The Agile Pyramid



COMPARISON WITH TRADITIONAL METHODOLOGIES

Agile

- Prioritizes **customer satisfaction** and **collaboration** over following a strict plan
- Emphasizes **flexibility** and **adaptation to change**
- Divides work into small **increments** and allows for **regular review** and adaptation
- Values working software and **continuous delivery** over extensive documentation
- Uses lightweight, **iterative** planning and delivery processes

Traditional

- Prioritizes following a **strict plan** and **controlling scope** over customer satisfaction and collaboration
- Emphasizes **predictability** and **stability** over flexibility and adaptation to change
- Delivers work in **large chunks** and allows for review and adaptation only at designated points
- Values **extensive documentation** and **planning** over working software and continuous delivery
- Uses heavy, **sequential** planning and delivery processes

FLASHBACK

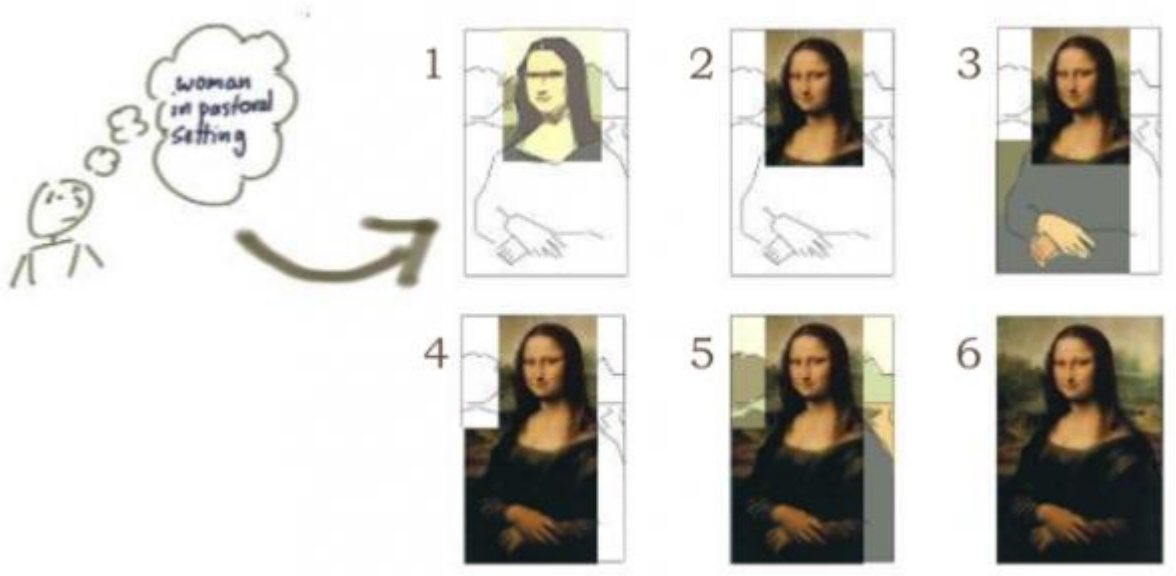
Iterative, Incremental or Hybrid

Iterative

Incremental

FLASHBACK

Iterative, Incremental or Hybrid



KEY PRINCIPLES OF AGILE PROJECT MANAGEMENT

- There are 12 principles outlined in the Agile Manifesto:

1. Customer satisfaction through continuous delivery of valuable software.
2. Welcome changing requirements, even if they occur late in the project.
3. Deliver working software frequently, with a preference for shorter timescales.
4. Close collaboration between customers and the self-organizing and cross-functional development team.
5. Project management and development work should be done face-to-face as much as possible.
6. Build projects around motivated individuals and give them the environment and support they need.
7. Working software is the primary measure of progress.
8. Sustainable development, able to maintain a constant pace indefinitely.
9. Technical excellence and good design enhance agility.
10. Simplicity, the art of maximizing the amount of work not done, is essential.
11. Self-organizing teams lead to the best architectures, requirements, and designs.
12. Regularly reflect on how to become more effective, and adjust accordingly.

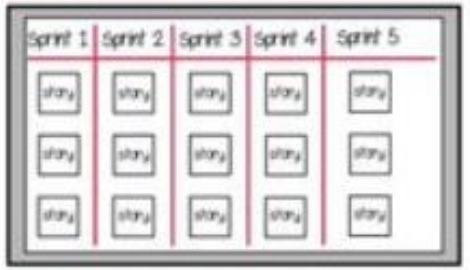
Knowledge TRAIN

The 12 agile principles*

1 Satisfy the **customer**

2 Welcome **change**

3 Deliver **frequently**



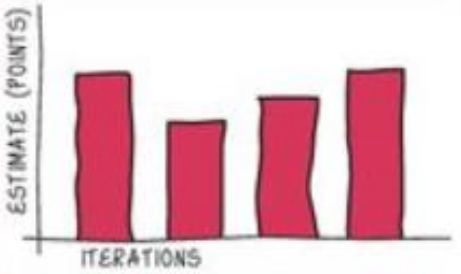
4 Work **together**

5 Trust and **support**

6 Face-to-face **conversation**

7 Working **software**

8 Sustainable **development**



9 Continuous **attention**

10 Maintain **simplicity**

11 Self-organizing **teams**

12 Reflect and **adjust**

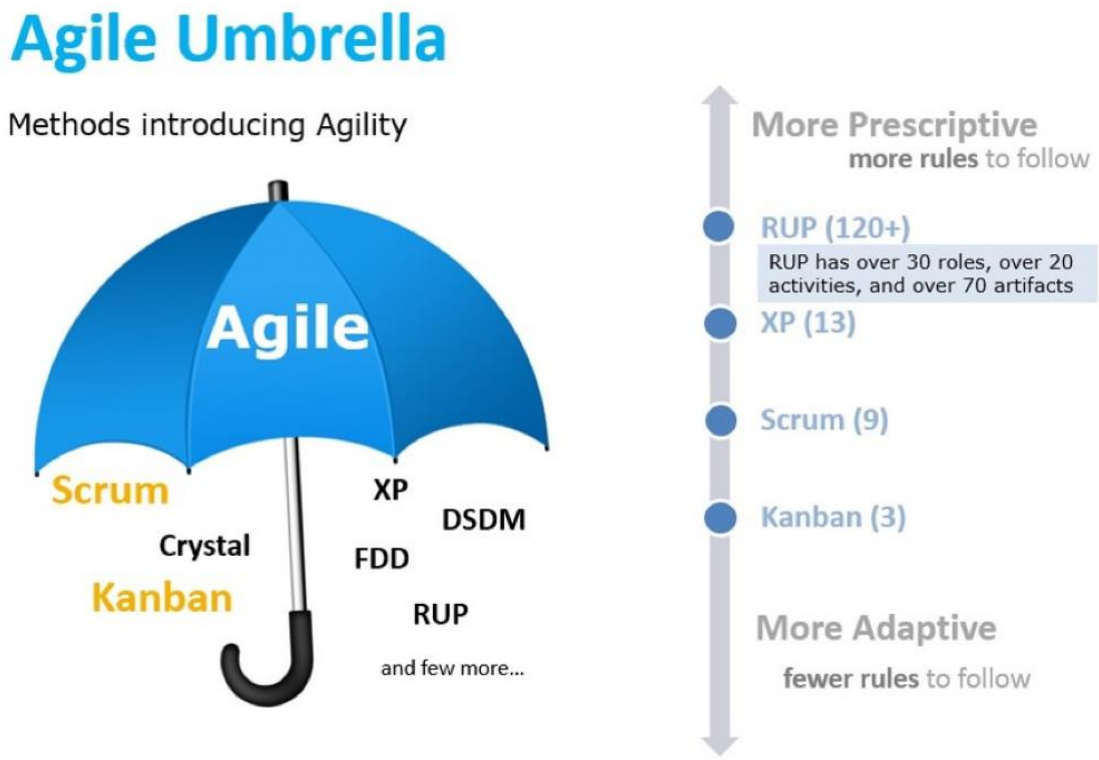
PRODECYS

ENLIGHTEN YOUR TRAINING

AGILE FRAMEWORKS

FRAMEWORK DEFINITION

- A framework is a set of **guidelines**, **principles**, and **rules** that provide a structure and help organizations achieve **specific goals**.
- In the context of project management, a framework is a set of processes, practices, and tools that organizations can use to manage projects and achieve their objectives.
- An Agile framework provides a structured approach to managing projects in an Agile manner, including guidelines for **planning**, **prioritizing**, and **delivering** work.
- The framework provides a common understanding of the Agile process and helps organizations implement Agile principles and practices effectively.
- The framework can be adapted to fit the specific needs of an organization or project, but the core principles and processes remain the same.



POPULAR AGILE FRAMEWORKS

- **Scrum**: Scrum is one of the most popular Agile frameworks and is commonly used in software development. It is an iterative and incremental framework that emphasizes collaboration, flexibility, and continuous improvement. Key elements of Scrum include sprints, daily stand-up meetings, and retrospectives.

- **Kanban**: Kanban is a visual management system that helps teams prioritize and manage work as it flows through the development process. Unlike Scrum, Kanban does not prescribe time-boxed iterations, but instead allows work to be pulled into the process as capacity allows.

- **Lean**: Lean is an Agile framework that is based on the principles of the Toyota Production System. It emphasizes the elimination of waste and continuous improvement and is often used in manufacturing and service industries.

- **XP (Extreme Programming)**: XP is a software development framework that emphasizes the importance of coding and testing. It uses a set of practices, including pair programming and continuous integration, to ensure high-quality software is delivered quickly.

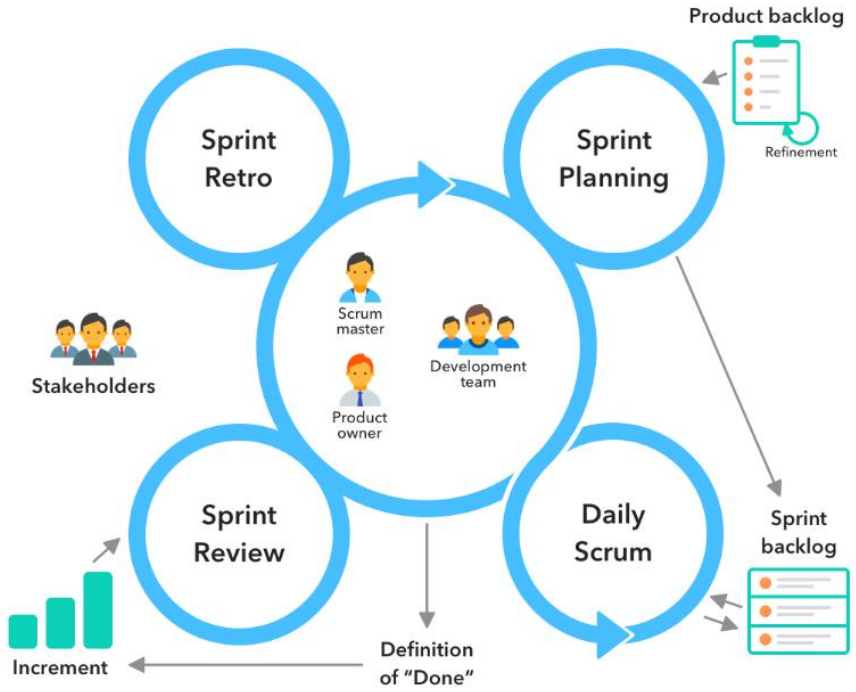
- **Crystal**: Crystal is a family of Agile methodologies that are tailored to different levels of risk, team size, and project complexity. Crystal methodologies prioritize people over processes and emphasize the importance of clear communication and adaptability.

SCRUM

Process and main definitions

Scrum: It is an iterative and incremental framework that emphasizes collaboration, flexibility, and continuous improvement. Key elements of Scrum include sprints, daily stand-up meetings, and retrospectives.

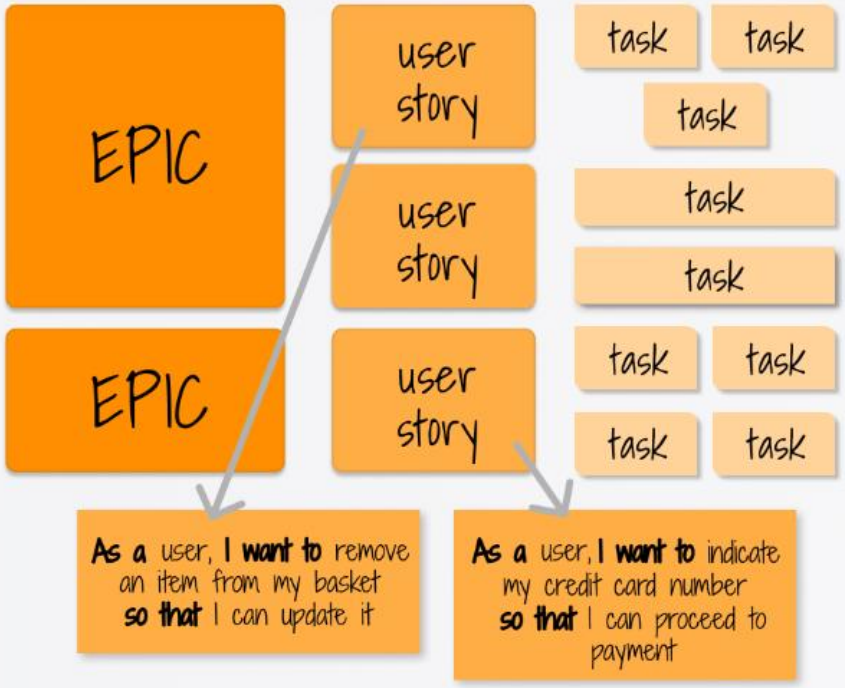
- Product Backlog: a prioritized list of features and requirements for the product being developed
- Sprint: a time-boxed iteration of work, typically lasting one to four weeks, during which a usable increment of the product is delivered
- Sprint Backlog: a list of items from the product backlog that the team plans to complete during the current sprint
- Daily Stand-Up Meeting: a short, daily check-in meeting to discuss progress and plan for the day
- Sprint Review: a meeting held at the end of each sprint to review what was accomplished and plan for the next sprint
- Sprint Retrospective: a meeting held at the end of each sprint to reflect on what went well and what could be improved in future sprints



SCRUM

Process and main definitions

- User story: A user story is an informal, general explanation of a feature written from the perspective of the end user or customer. The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer.
- Epic: is a body of work that can be broken down into specific user stories based on the needs/requests of customers or end-users
- Tasks vs User story: A user story is something that is generally worked on by more than one person (and defined by the customer generally), and a task is generally worked on by just one person (could be verry technical)



Understanding the Agile Scrum Method: the difference between Epic, User Story and Task

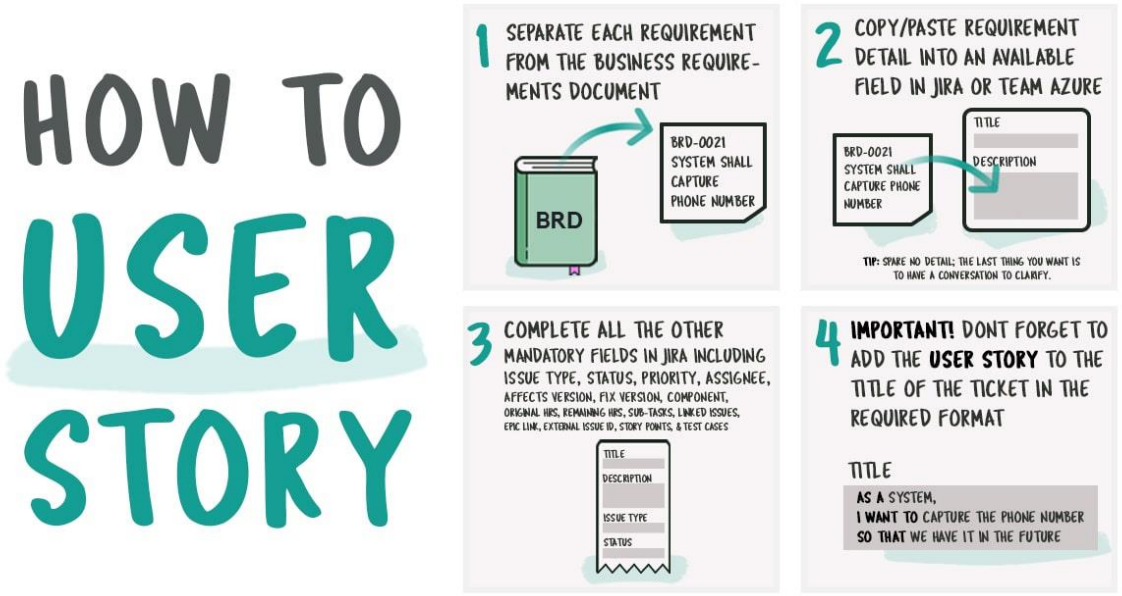
SCRUM

Focus on User Story

- User stories are a way of capturing the requirements for a software project from the perspective of the end user.
- User stories are also used to break down the requirements into smaller, more manageable tasks that can be estimated and tracked as part of an Agile sprint or iteration
- They are short, simple descriptions of a feature or requirement, expressed in the form of a statement from the user's point of view.
- A user story typically follows the format: "As a [type of user], I want [some goal or need], so that [some reason or benefit]."

For example, "As a customer, I want to be able to search for products on the website, so that I can find what I need quickly and easily."

- User stories provide a way to capture the high-level requirements of a project, and they are often used in combination with other Agile techniques, such as story mapping, to help teams understand how the requirements fit together..



SCRUM

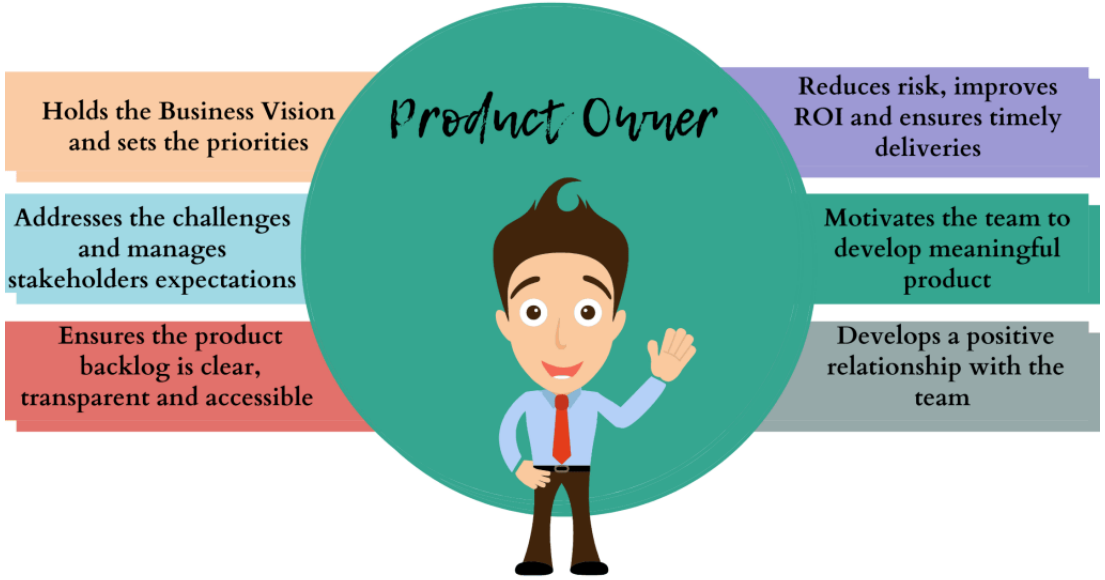
Key actors

- Scrum Master: a facilitator who helps the team follow the Scrum process and removes any obstacles to progress
- Product Owner: is responsible for defining and prioritizing the product backlog
- Development Team: a self-organizing, cross-functional team responsible for delivering the product

SCRUM

Product owner

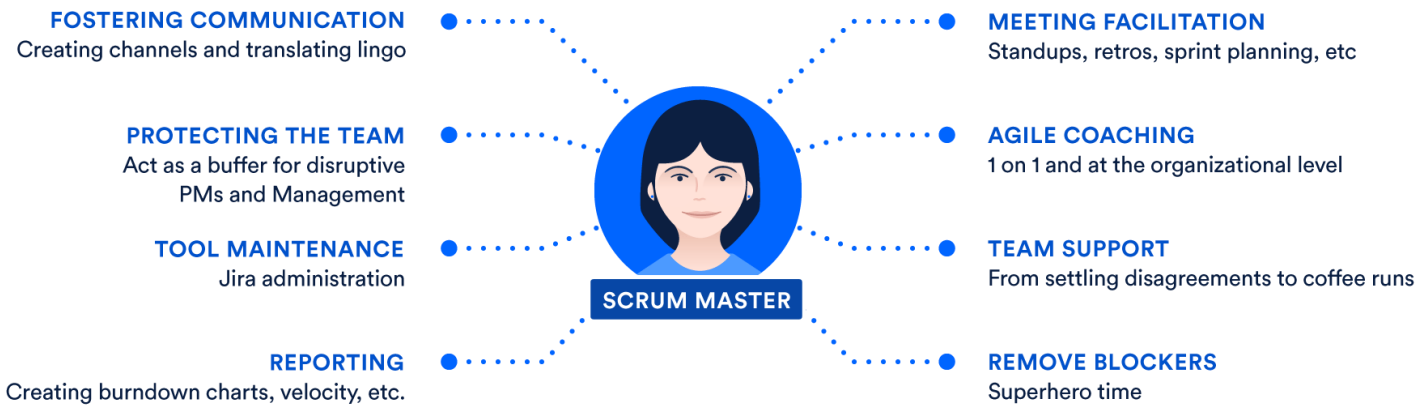
- Responsible for defining and prioritizing the product backlog.
- Represents the stakeholders and works with the team to ensure that the product backlog is clear, concise, and meets the needs of the stakeholders.
- Takes decisions about what work the team will do in each sprint and accepts or rejects completed work based on the acceptance criteria.



SCRUM

Scrum Master

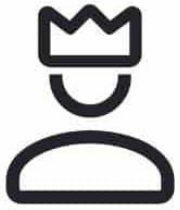
- Responsible for facilitating the Scrum process and helping the team to follow the Scrum framework.
- Helps to remove any obstacles that the team may face, facilitates meetings such as daily stand-ups, sprint planning, and retrospectives, and helps the team to adhere to the Agile principles and values.
- Acts as a coach for the team, helping them to continuously improve their processes and work methods.



SCRUM

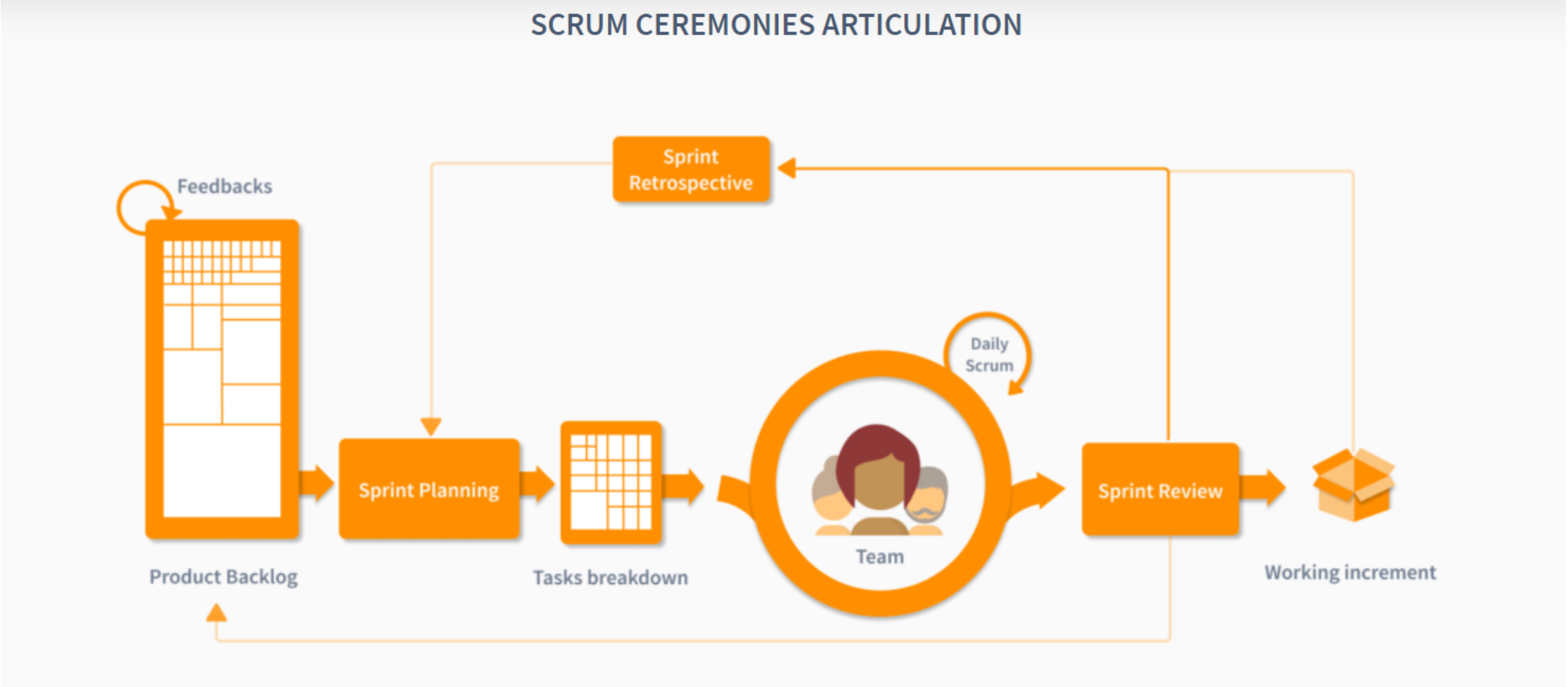
Scrum Master vs Product Owner

- The Product Owner and Scrum Master play complementary roles in ensuring the success of an Agile project.

 Criteria	 Scrum Master	 Product Owner
Nature of work	Acts as a team coach and is responsible for maintaining the quality of the product	Collaborates with all stakeholders and brings the vision of a product into the product backlog.
Responsibilities	Ensures the Scrum framework is followed and helps the development team create a quality product.	Responsible for completing the project on time. Acts as an intermediary between the development team and the customers
Accountability	Accountable for the quality of the entire project and for giving updates to management about the completion of the product.	Responsible for the project backlog the timely completion of the product and providing updates to the clients and stakeholders.
Reporting	Reports to top management about the efficiency of the team and the quality of the product.	Reports to top management and the clients.
Qualities	Thorough knowledge of Scrum theory and practices. Being able to lead a team but without a sense of authority.	Communication and leadership skills,creativity, critical thinking, and a sharp mind are key assets for any Product Owner.

SCRUM

Scrum ceremonies articulation



SCRUM

How Agile teams execute and deliver work

- Sprint planning: At the start of each sprint, the team:
  - Meets to plan the work that they will complete during the sprint
  - Reviews the user stories that have been prioritized for the sprint and discusses how they will be completed.
  - Decides on the sprint goal and identifies any dependencies or constraints that need to be taken into account.

- Daily stand-ups: During the sprint, the team holds daily stand-up meetings (daily scrums) to:
  - Discuss progress
  - Identify any obstacles that need to be addressed

  These meetings are short and focused, and they provide a way for the team to stay aligned and to ensure that everyone is working towards the same goal.

- Task execution: During the sprint, the team members work on the tasks that they have been assigned, using Agile practices and tools, such as pair programming and continuous integration, to ensure that the work is completed efficiently and effectively.

SCRUM

How Agile teams execute and deliver work

- Sprint review: At the end of each sprint, the team holds a sprint review to:
  - Demonstrate the work that has been completed
  - Receive feedback from stakeholders.

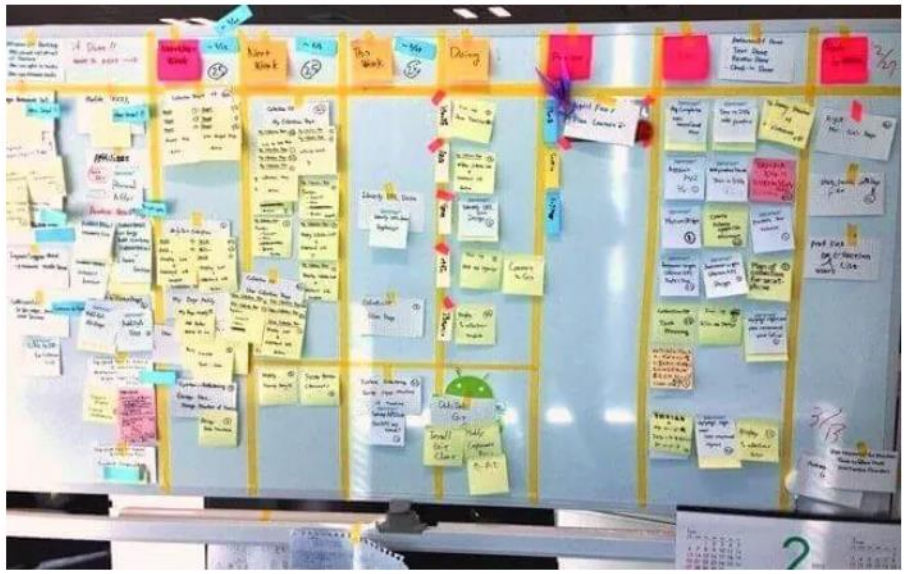
  The team discusses what has been delivered and what needs to be improved, and they make any necessary adjustments to the backlog or the sprint plan for the next sprint.

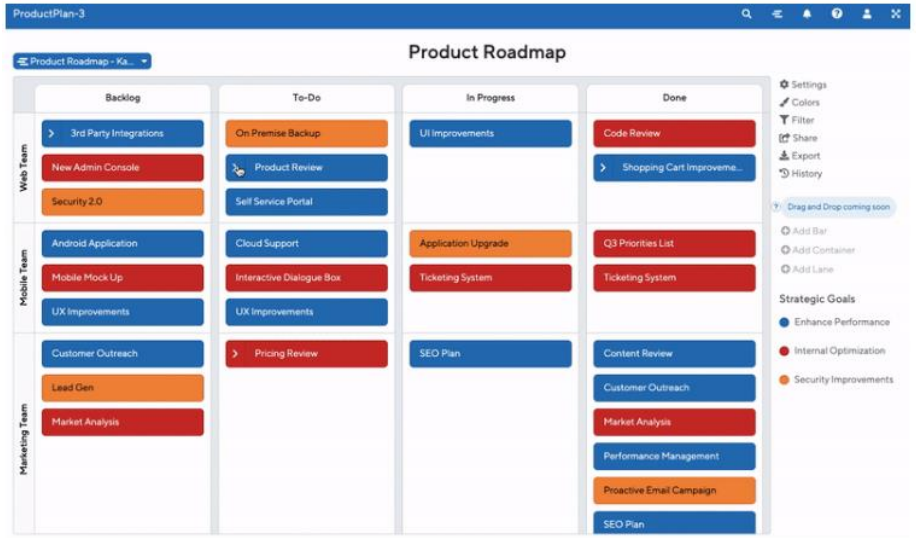
- Sprint retrospectives: After the sprint review, the team holds a sprint retrospective to reflect on the sprint and to identify any areas for improvement. The team discusses what went well, what didn't go well, and what they can do differently in the next sprint to make the process even more effective.

KANBAN

Kanban boards examples

Original Kanban system, source: Official TOYOTA internet site



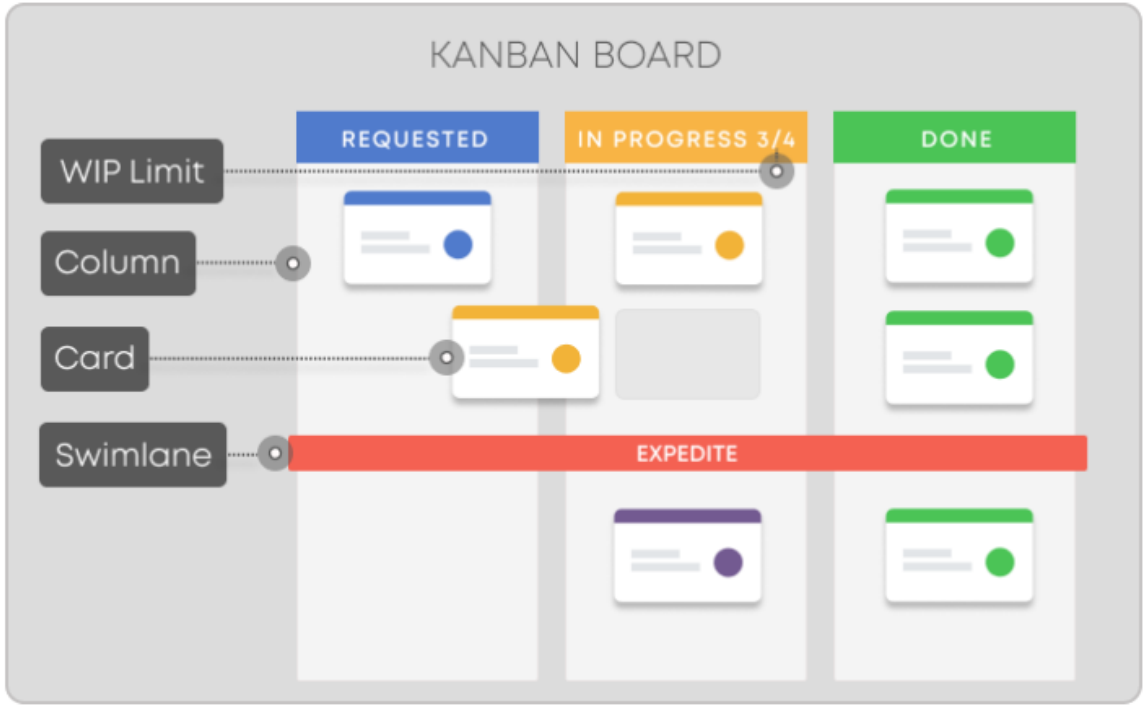


KANBAN

Main definitions

Kanban: Kanban is a visual management system that helps teams prioritize and manage work as it flows through the development process. Unlike Scrum, Kanban does not prescribe time-boxed iterations, but instead allows work to be pulled into the process as capacity allows.

- Work Items: tasks or features that need to be completed
- Workflow: a visual representation of the steps involved in delivering a work item from start to finish
- Work-In-Progress Limits: limits placed on the number of work items that can be in progress at any given time, to prevent overburdening the team
- Continuous Flow: a focus on ensuring work flows continuously through the process, with minimal waiting time
- Visual Management: the use of visual aids, such as boards and sticky notes, to manage the flow of work
- Swimlane: are used to separate different types of work



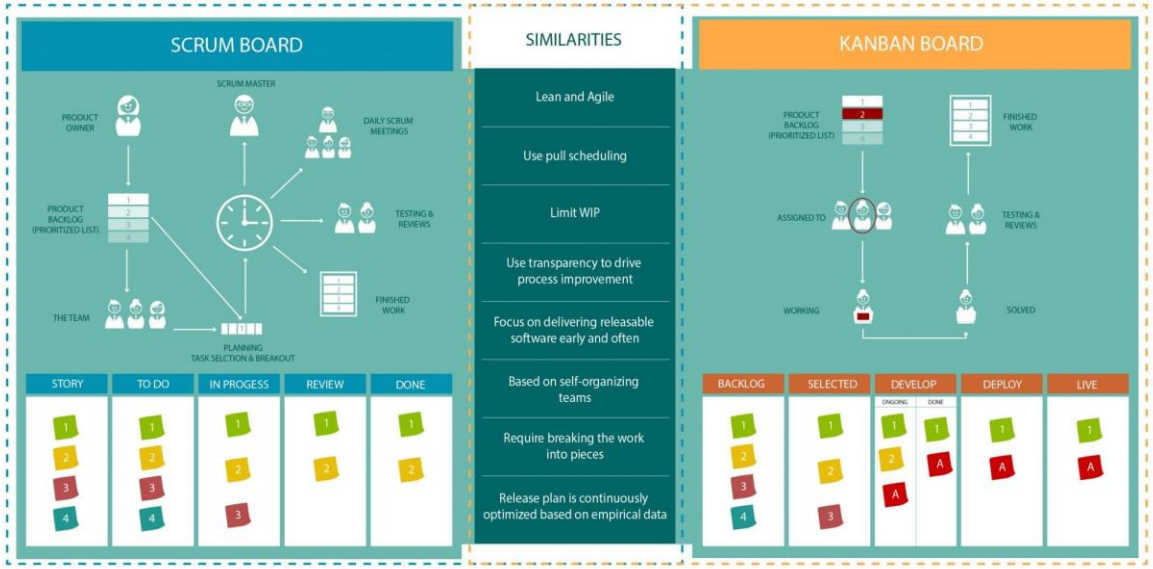
LEAN

Main definitions

Lean: Lean is an Agile framework that is based on the principles of the Toyota Production System. It emphasizes the elimination of waste and continuous improvement and is often used in manufacturing and service industries.

- Value Stream Mapping: a visual representation of the steps involved in delivering value to the customer, used to identify and eliminate waste
- Continuous Improvement: a focus on continuously improving processes and eliminating waste, with the goal of delivering more value to the customer
- Kaizen: a Japanese term for continuous improvement, often used in Lean
- Pull System: a system for allowing work to be pulled into the process as demand allows, rather than being pushed into the process
- Elimination of Waste: a focus on identifying and eliminating any activities that do not add value to the customer

COMPARISON



DIFFERENCE BETWEEN	
SCRUM	**KANBAN**
Timeboxed iterations prescribed.	Timeboxed iterations optional. Can have separate cadences for planning, release, and process improvement. Can be event driven instead of timeboxed.
Team commits to a specific amount of work for this iteration.	Commitment optional.
Uses Velocity as default metric for planning and process improvement.	Uses Lead time as default metric for planning and process improvement.
Cross-functional teams prescribed.	Cross-functional teams optional. Specialist teams allowed.
Items must be broken down so they can be completed within 1 sprint.	No particular item size is prescribed.
Burndown chart prescribed.	No particular type of diagram is prescribed.
WIP limited indirectly (per sprint).	WIP limited directly (per workflow state).
Estimation prescribed.	Estimation optional.
Can not add items to ongoing iteration.	Can add new items whenever capacity is available.
A sprint backlog is owned by one specific team.	A Kanban board may be shared by multiple teams or individuals.
Prescribes 3 roles (PO/SM/Team).	Doesn't prescribe any roles.
A Scrum board is reset between each sprint.	A Kanban board is persistent.
Prescribes a prioritized product backlog.	Prioritization is optional.

PRODECYS

ENLIGHTEN YOUR TRAINING

AGILE PLANNING AND ESTIMATION

ACCEPTANCE CRITERIA

- Acceptance criteria are the specific conditions that a feature or requirement must meet in order to be considered complete. They provide a clear definition of what is expected from the project and serve as a benchmark for measuring the success of the project.

- Acceptance criteria are typically defined for each user story and are used to ensure that everyone involved in the project has a clear understanding of what is expected. They provide a shared understanding of what needs to be delivered and help to ensure that everyone is working towards the same goal.

- Acceptance criteria should be concise, specific, and measurable. For example, acceptance criteria for a user story that involves the creation of a new website might include:
    - *The website must load in less than 2 seconds on a broadband connection.*
    - *The website must be visually appealing and user-friendly.*
    - *The website must be accessible to users with disabilities.*

- Acceptance criteria should also be realistic and achievable. They should be based on the available resources, such as time, budget, and technical constraints, and they should be agreed upon by all stakeholders.

HOW TO USE ESTIMATION TO MANAGE SCOPE AND SCHEDULE

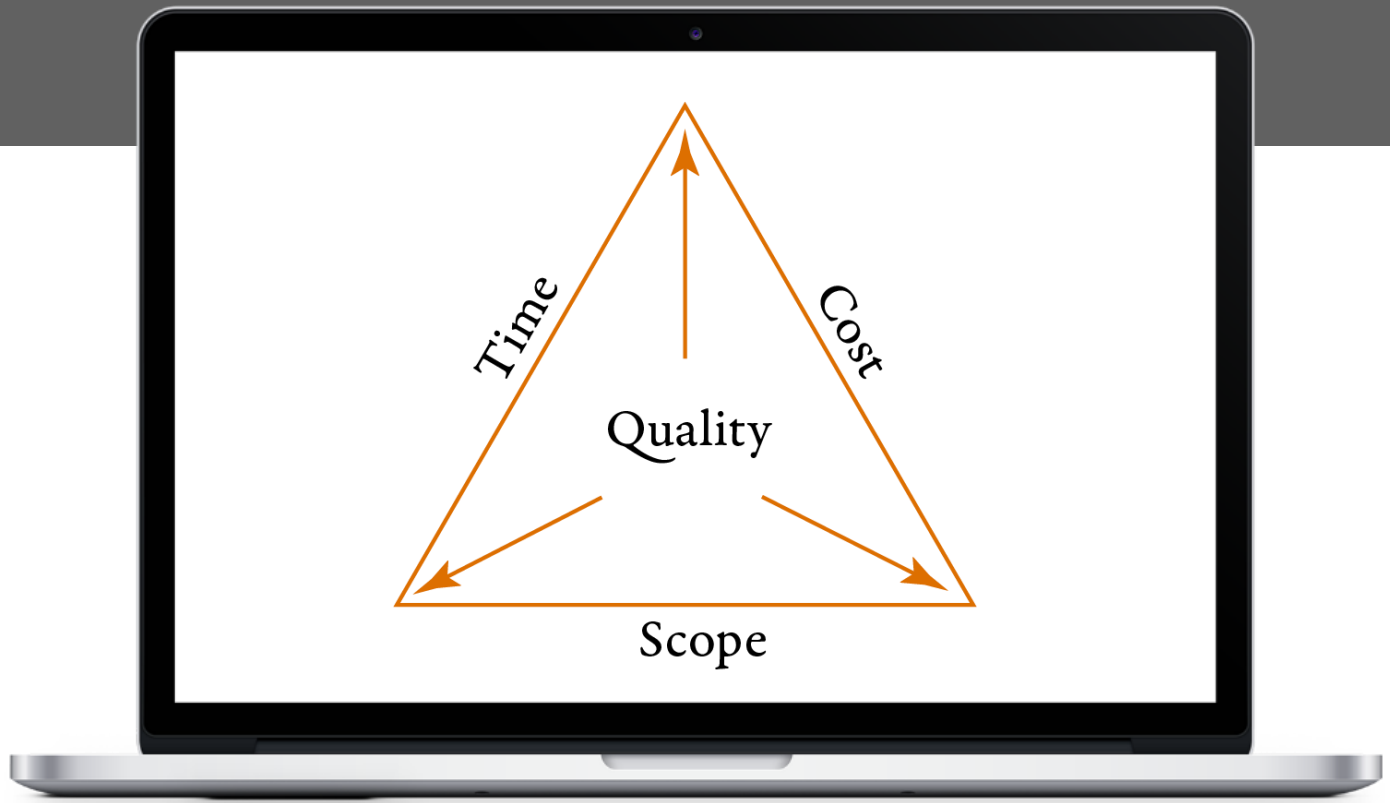
- Teams should establish a consistent estimation process that works consistently from one sprint to the next. Using a specific estimation technique (ex: planning Poker or Affinity Estimation), or a specific tool (ex: story points).
- Use accurate and relevant data to estimate the size or effort required to complete a project or task. Using historical data from previous similar projects is also a good practice
- Encourage active participation from all team members. This can help to reduce the influence of any one team member and can help to ensure that everyone has a say in the estimation process.
- Regularly review and adjust: This can help to ensure that the process remains relevant and effective and can help to account for any changes in the project or the team.
- Use estimation to manage the scope: By prioritizing the tasks that need to be completed and by determining what can be delivered in an upcoming sprint or iteration. This can help to ensure that the project stays on track and that the team is working towards the same goal.
- Use estimation to manage the schedule: By determining how much time they need to complete each task and by determining how many sprints or iterations they will need to complete the project. This can help to ensure that the project stays on track and that the team is working towards the same goal.

HOW TO USE ESTIMATION TO MANAGE SCOPE AND SCHEDULE

Flashback : The project (iron) triangle

- 3 parameters will impact the quality of the project
  - Scope
  - Time
  - Cost

The basics of project management is to create a balance between the defined scope, the given timeframe and the agreed cost

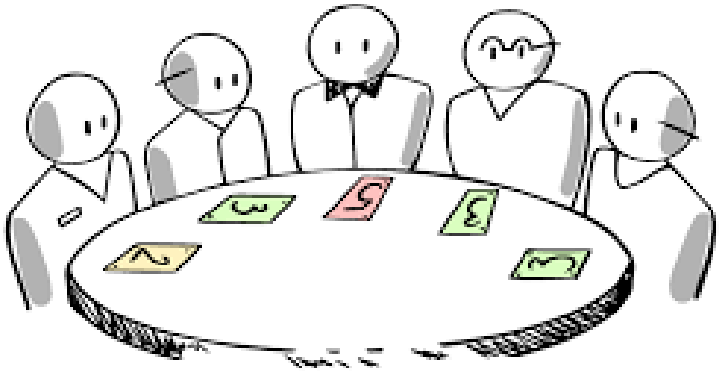


PLANNING POKER

- Planning Poker: method used to estimate the effort required to complete a user story. It is a consensus-based technique used to estimate the size or complexity of a user story and to help teams make informed decisions about what can be delivered in an upcoming sprint or iteration.

- The process of Planning Poker involves a group of team members, each with a deck of cards, each representing a different estimate of effort. The cards represent the number of story points or ideal days required to complete the user story.

- During a Planning Poker session, the team discusses the user story and each team member privately estimates the effort required to complete it. The team then reveals their estimates at the same time, and the team members discuss any differences in their estimates. The goal is to reach consensus on the effort required to complete the user story.
- Planning Poker helps to encourage active participation from all team members and to ensure that everyone has a say in the estimation process. It also helps to reduce the influence of any one team member and to eliminate biases in the estimation process.



AFFINITY ESTIMATION

- Affinity estimation works by breaking down a project or task into smaller, more manageable chunks and then grouping these chunks into categories based on size or effort. For example, a project or task might be broken down into smaller tasks, and each of these tasks might be grouped into categories such as "small", "medium", or "large".

- The team then uses their knowledge and experience to estimate the size of each chunk based on the categories they have created. For example, the team might estimate that a particular task is "medium" in size. This estimate is then used to calculate the total effort required to complete the project or task.

- Affinity estimation is a useful technique for teams that are new to Agile or that have limited experience with estimation. It is simple and straightforward, and it can help to reduce the influence of any one team member and to encourage active participation from all team members.

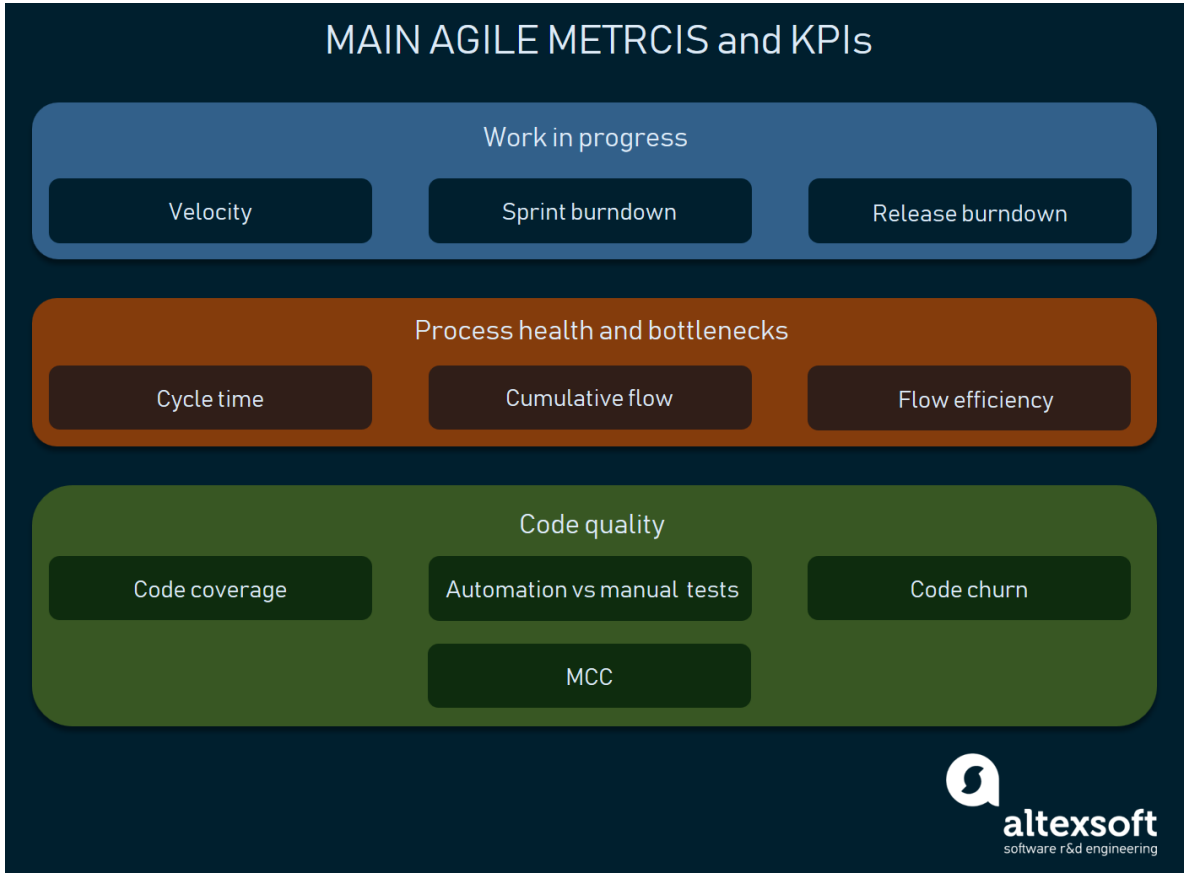
PRODECYS

ENLIGHTEN YOUR TRAINING

AGILE METRICS

AGILE METRICS

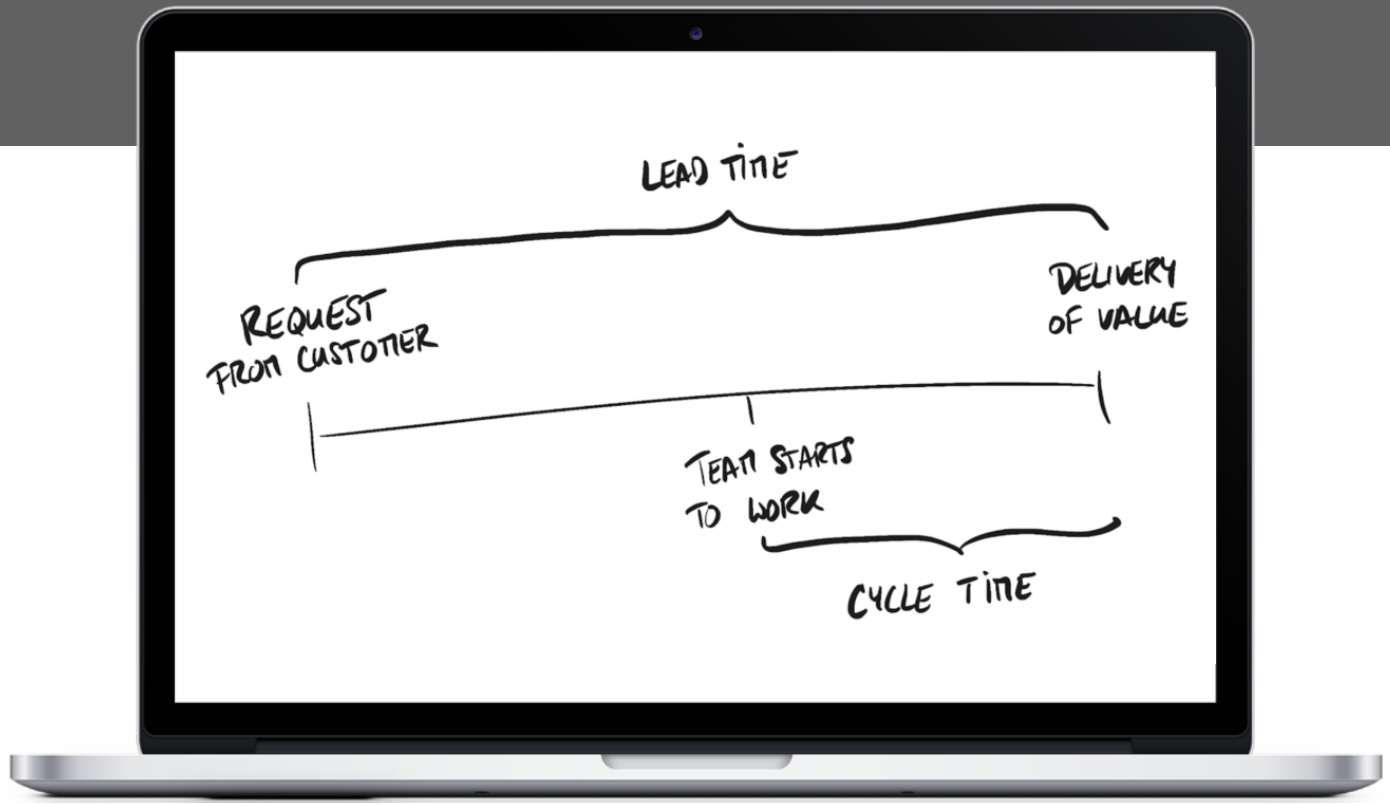
How to use metrics to measure progress and improve performance



- Velocity and cycle time are two key metrics used in Agile project management to measure the progress and performance of a team.

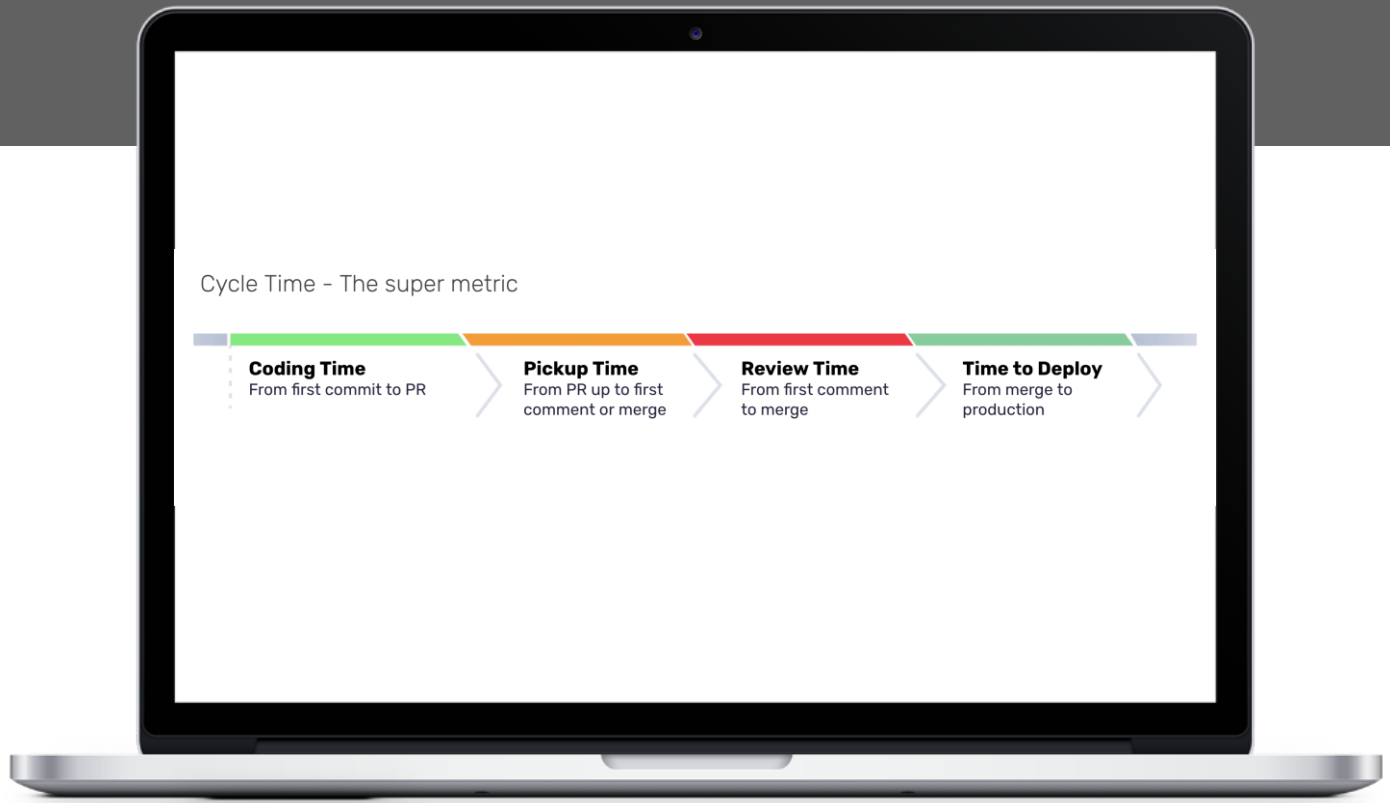
VELOCITY

- Velocity is a measure of the amount of work that a team can complete in a sprint or iteration.
- It is calculated as the sum of the estimated size of the user stories that have been completed during a sprint.
- The estimated size can be expressed in a variety of units, such as story points, ideal days, or ideal hours, depending on the team's preference.
- Velocity is used as a way of predicting the amount of work that the team can complete in future sprints and it is used to help the team plan their sprints more effectively.



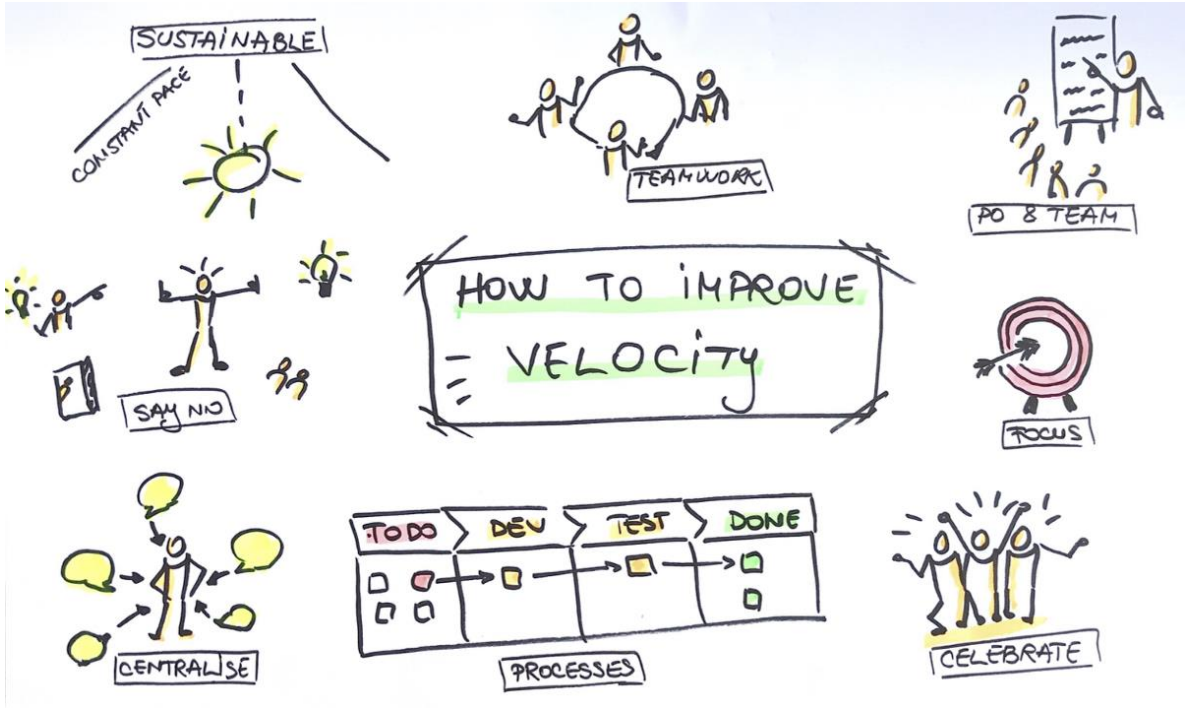
CYCLE TIME

- Cycle time: is the time it takes for a piece of work to be completed from start to finish, including all the stages, such as development, testing, and deployment.
- Cycle time can be used to measure the efficiency and effectiveness of the team's processes and it can help the team to identify any bottlenecks or inefficiencies in their work.



HOW TO MEASURE AND IMPROVE

- Collect data: on velocity and cycle time on a regular basis, such as after each sprint, to get an accurate picture of their performance.
- Analyze data: to understand the trends and patterns in their performance, such as which user stories take the longest to complete and which have the highest velocity.
- Identify areas for improvement: Teams should use the data to identify areas for improvement, such as bottlenecks in the work processes or inefficiencies in the team's work methods.
- Implement improvements: Teams should implement the improvements that they have identified, such as streamlining their work processes or introducing new tools and techniques to improve their work methods.
- Monitor progress: Teams should monitor their progress after the improvements have been made to ensure that the changes have had the desired effect.

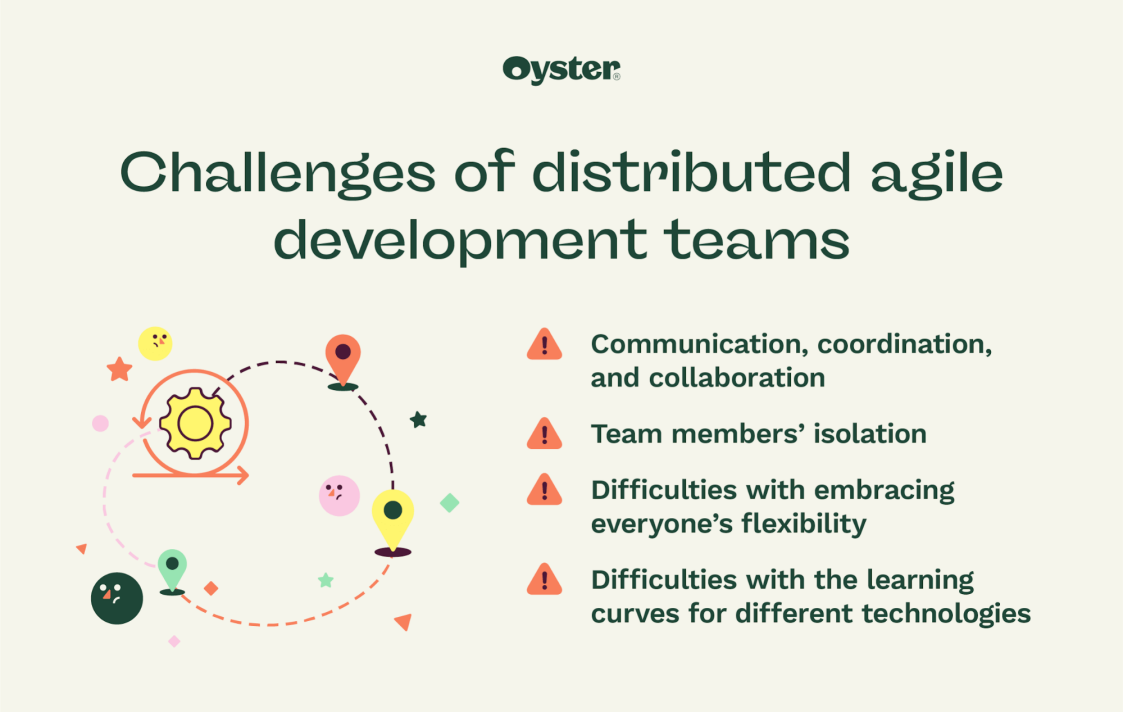


PRODECYS

ENLIGHTEN YOUR TRAINING

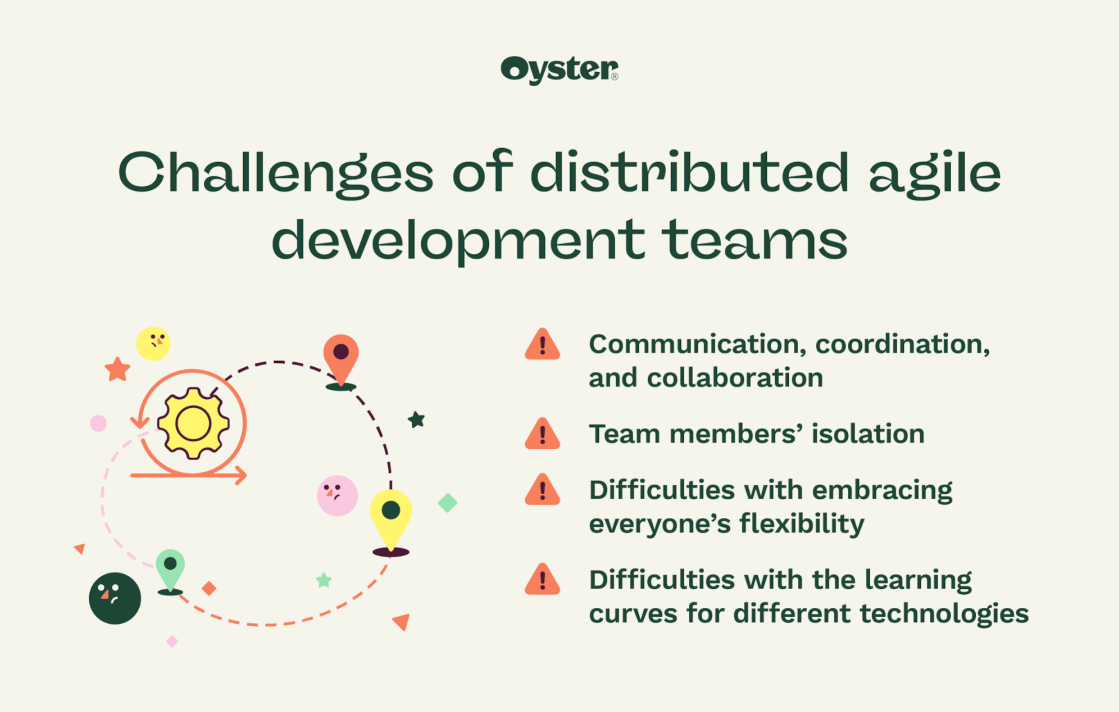
AGILE CHALLENGES AND CULTURE

CHALLENGES WITH AGILE



- Resistance to change: Agile methodology can be a significant change for organizations that are used to traditional project management approaches, and there can be resistance to this change from team members, stakeholders, and even senior management.
- Lack of clear requirements: In Agile, requirements are often not fully defined at the start of the project and can change during the project. This can lead to confusion and misunderstandings and can make it difficult to manage the scope and schedule of the project.
- Difficulties in prioritizing work: The Agile approach of continuously delivering working software can make it difficult to prioritize work and ensure that the most important tasks are being worked on first.
- Miscommunication between stakeholders: Agile requires regular and effective communication between stakeholders, team members, and customers. Miscommunication can lead to misunderstandings, confusion, and delays in the project.

CHALLENGES WITH AGILE



- Resistance to self-organizing teams: In Agile, teams are expected to be self-organizing, which can be a challenge for organizations that are used to hierarchical structures.
- Difficulty in measuring progress: In Agile, progress is often measured in terms of working software and user stories, which can be difficult for stakeholders to understand and track.
- Challenges with scaling: Scaling Agile from small projects to larger, more complex projects can be challenging, as the processes and tools that work for small projects may not be appropriate for larger projects.
- Resistance to collaboration: In Agile, collaboration is key to success, and resistance to collaboration can lead to delays and misunderstandings.

PRODECYS

ENLIGHTEN YOUR TRAINING

CONCLUSION

KEY TAKEAWAYS

- Agile is a flexible, iterative, and customer-focused approach to project management that values delivering working software over following a rigid plan.
- Agile methodology is based on 12 principles that emphasize collaboration, customer focus, and continuous improvement.
- Popular Agile frameworks include Scrum, Kanban, XP, and Lean. Each framework has its own set of core elements and can be used in different situations.
- Agile planning practices include user stories, acceptance criteria, estimation, and prioritization.
- Agile teams deliver work through sprints and continuously measure progress through velocity and cycle time.
- The role of the product owner is to define the product backlog and ensure that the team is working on the most important tasks. The role of the scrum master is to facilitate the Agile process and ensure that the team is following Agile practices.
- Creating an Agile culture involves fostering collaboration, communication, and continuous improvement.
- Common challenges with the Agile methodology include resistance to change, difficulties in prioritizing work, miscommunication, and scaling issues.


thank you

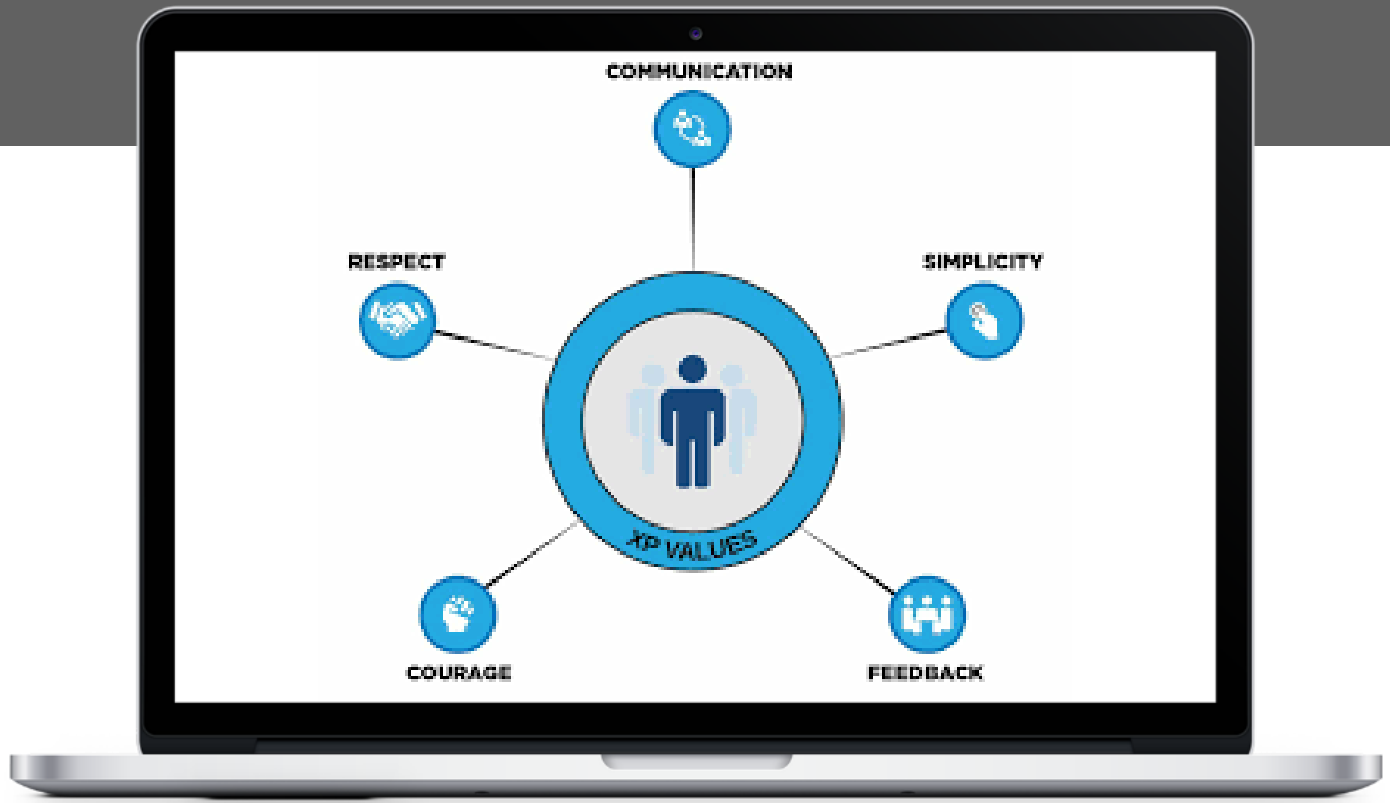
PRODECYS

ENLIGHTEN YOUR TRAINING

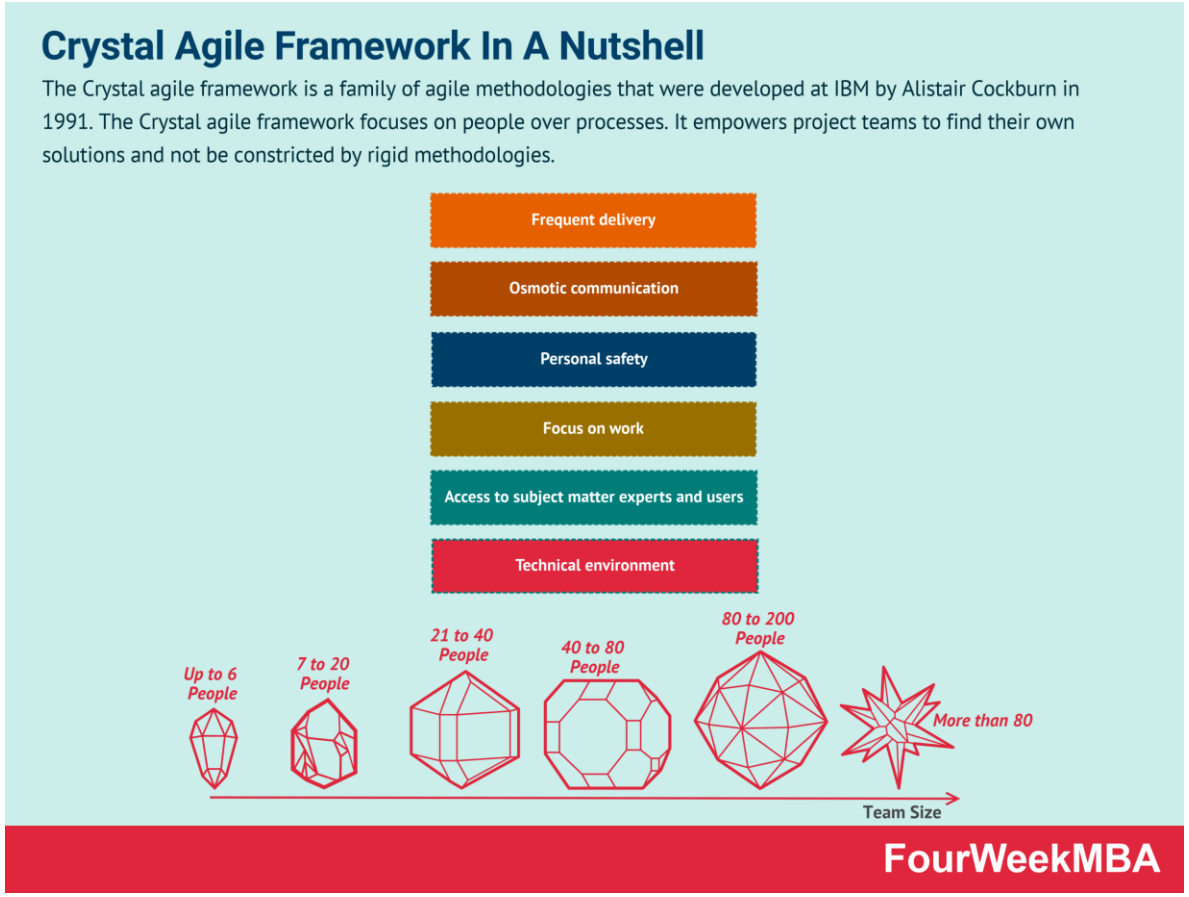
BACK-UP SLIDES

XP (EXTREME PROGRAMMING)

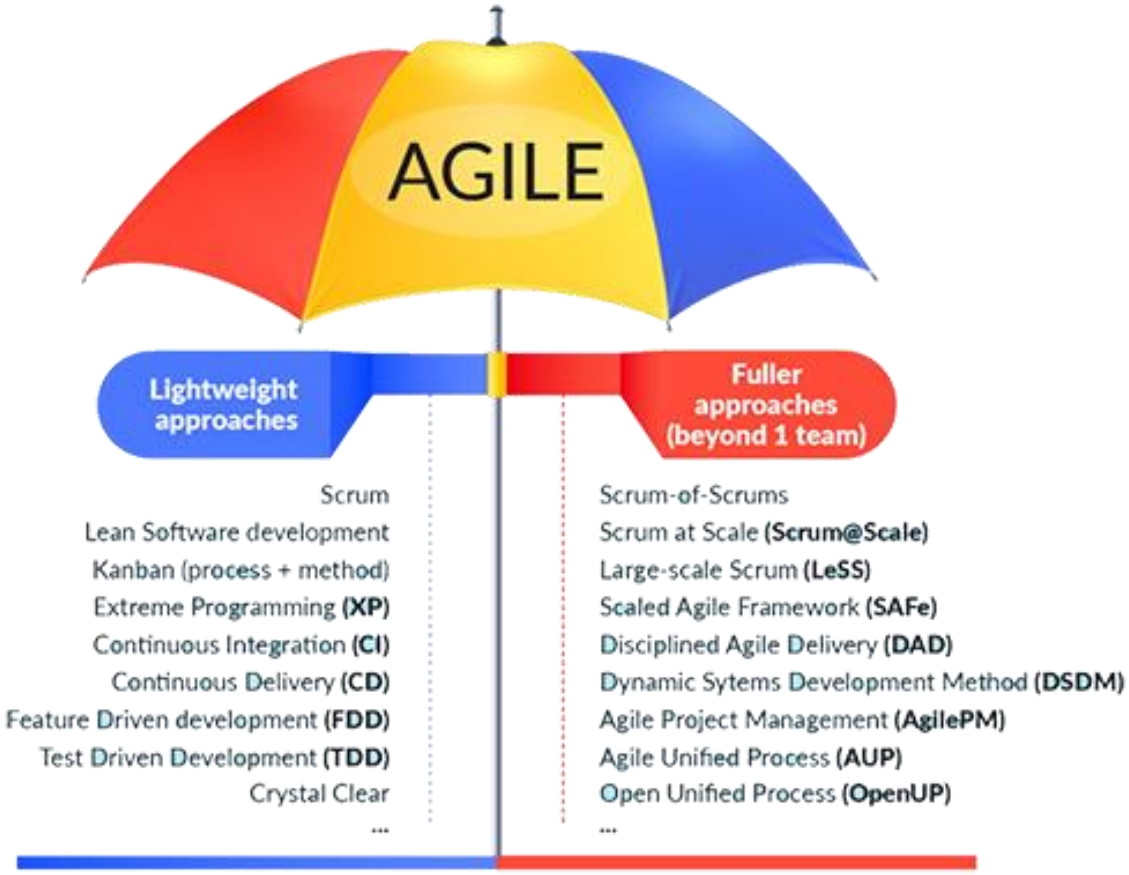
- **Continuous Integration:** a practice of integrating code changes into a shared repository several times a day, to ensure integration issues are detected and resolved quickly
- **Pair Programming:** a practice where two developers work together at the same computer, with one typing and the other reviewing and suggesting improvements
- **Test-Driven Development:** a practice where tests are written before code is written, to ensure code meets requirements and is of high quality
- **Continuous Refactoring:** a practice of continually improving the design of the code, to ensure it remains maintainable and scalable
- **Small Releases:** a focus on delivering small, usable increments of the product to the customer, rather than trying to deliver the entire product all at once



CRYSTAL



- Clear Communication: a focus on ensuring clear and open communication between team members, stakeholders, and customers
- Personal Safety: a focus on creating a safe and supportive environment for team members to express their opinions and ideas
- Adaptability: a focus on being flexible and adaptable, and being willing to change course if necessary
- Continuous Improvement: a focus on continuously improving processes and delivering more value to the customer
- Team-Oriented Processes: a focus on creating processes that support and empower the team, rather than dictating how the team should work.


AGILE
Lightweight approaches
Fuller approaches (beyond 1 team)
Scrum
Lean Software development
Kanban (process + method)
Extreme Programming (XP)
Continuous Integration (CI)
Continuous Delivery (CD)
Feature Driven development (FDD)
Test Driven Development (TDD)
Crystal Clear
...
Scrum-of-Scrums
Scrum at Scale (Scrum@Scale)
Large-scale Scrum (LeSS)
Scaled Agile Framework (SAFe)
Disciplined Agile Delivery (DAD)
Dynamic Sytems Development Method (DSDM)
Agile Project Management (AgilePM)
Agile Unified Process (AUP)
Open Unified Process (OpenUP)
...