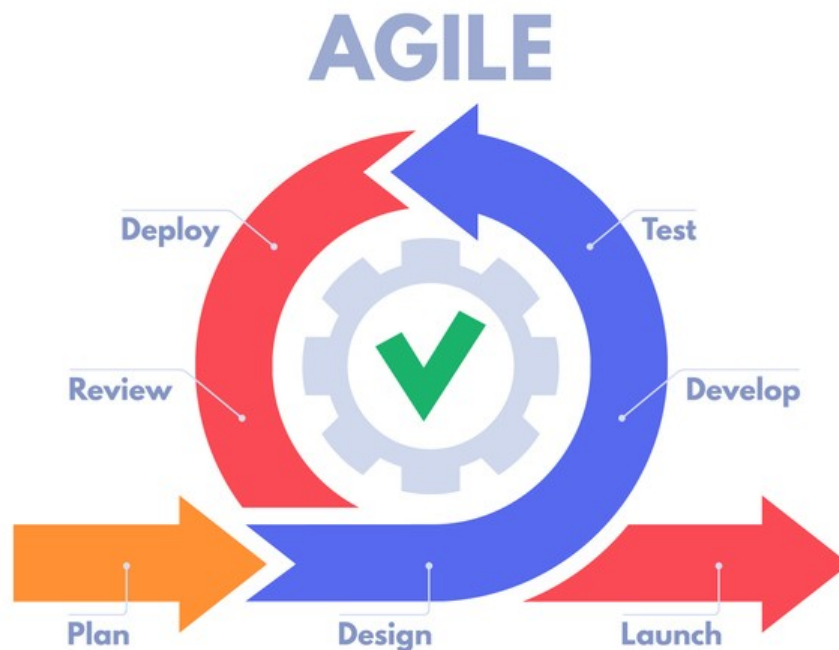


Topic 2. Agile Methodologies

Introduction

It is getting more and more frequent to hear the word "agile" regarding software development. In fact, it is also a term used in project management. This gives us the idea that everything related to project management implements this methodology.



Every organization seeks to improve processes, and to this end they are working on approaches and methodologies so that their work teams can adapt faster and achieve better differentiation. Among the existing approaches there are a couple that stand out, which we will study in this topic.

Explanation

2.1. Agile management vs. Traditional management

First, **traditional management** is a methodology in which projects are executed in a sequential cycle that includes **initiation, planning, execution, and measurement stages**. Projects, in their entirety, are planned, with special emphasis on linear processes, documentation, advance planning and prioritization. Under this approach, it is assumed that time and costs are

variable, and requirements are fixed, and for each step of the approach there are tools and techniques such as **PMBOK** or **PRINCE2**.

Some benefits to consider in respect to the traditional methodology are the following:

- Controllable processes
- Greater accountability
- Clearly defined objectives
- Clear documentation

Agile project management is an approach used mainly in software development, while in a traditional system, everything is focused on the initial planning (and importance is given to factors such as costs, scope, and time), in **agile management**, priority is given to the work team, collaboration with the client, tasks and flexibility. This process is interactive and focuses on customer feedback and continuous releases at every interaction of a development project.

Agile development emphasizes evolutionary changes and collaborative effort to achieve results rather than following a strictly predefined process, less time is spent on initial planning and prioritization, because Agile is more flexible in terms of specification changes and developments.

Just as there is PMBOK and PRINCE2 for traditional management, in Agile, there is Scrum and Kanban, well-known for encouraging decision-making and avoiding time consumption on variables that are bound to change. Both methods emphasize customer satisfaction and use available teams to accelerate software development at every stage.

There is an Agile Manifesto, and it has four important values.

1. Individuals and interactions over processes and tools.
2. Software running on extensive documentation.
3. Collaboration with the customer over contract negotiation.
4. Responsiveness to change over following a plan (agilemanifesto.org, 2021).

2.2. Choosing the right approach

The choice of approach depends purely on the project. Traditional methodologies rely on robust planning in the analysis and design phase of the project, team members are expected to follow the plan step-by-step with minimal guidance or direction, and progress is determined after the project is completed; there are no frequent checks and balances.

On the other hand, in agile methods there are shorter and faster iterations, team members are encouraged to perform checks at regulated intervals, this way it is easier to determine actual progress, and it helps people to maintain accountability in their work. The following table is intended to show some fundamental differences between the two approaches.

Characteristic	Agile approach	Traditional approach
Planning	It is planned from Sprint to Sprint.	Everything is planned in detail.
Requirements	Dynamic	Well, defined before starting.
Scale of projects	Small and medium	Large
Organizational structure	Interactive	Linear
Customer involvement	Customers are involved from the moment the work is started.	Customers are involved at the beginning of the project, but not once execution has begun.
Escalation management	When problems occur, the whole team works together to solve them.	The problem is escalated to project managers.
Model preferences	The agile model favors adaptation.	The traditional model favors anticipation.
Product or process	Less focus on formal, directive processes.	More focused on process than product.
Effort estimation	Scrum Master facilitates the tasks, and the team does the estimation.	Project manager estimates and obtains approval from the project owner.
Reviews and approvals	Reviews are done after each iteration.	Constant reviews and approvals by project leaders.

Why is the agile approach preferred over the traditional approach in software development?

Many project managers prefer to use the agile methodology for several reasons, here are some of them.

- **It is more flexible:** when it comes to making changes in the product or in a process, the agile methodology is much more flexible than the traditional one (also known as waterfall). It is worth mentioning that, due

to the linear nature of the traditional approach, making any change is costly and complicated, in agile any last-minute change can be introduced in the process without affecting the final result and interrupting the project schedule.

- **It is transparent:** customers and decision makers are actively involved from the start, planning, review, and in the testing part of a product everyone participates.
- **Ownership and accountability:** a notable difference in the two management approaches is the level of ownership and accountability each provides to team members. In traditional management, the project manager is the captain of the ship, he owns it and is in charge, although customers also participate, in fact, their participation is limited to the planning phase, after which their participation is nil. In agile each member shares ownership of the project, and each plays an active role to complete the sprint in the estimated time.
- **Feedback:** in the traditional approach it is very well defined and planned from the beginning of the project, since the project must be completed within the estimated time and budget, therefore, any major changes or comments that may affect the deadline are omitted. In agile management constant feedback is allowed, this helps to achieve a better result.
- **Project complexity:** this is where a fundamental question comes in: who isn't afraid to manage a big project?

The consultancy The Standish Group periodically publishes the Chaos Report, where it illustrates the success or failure of projects according to the methodology used, if we compare both methodologies, we will realize that also the size of the project plays an important role.

CHAOS RESOLUTION BY AGILE VERSUS WATERFALL

SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

The resolution of all software projects from FY2011-2015 within the new CHAOS database, segmented by the agile process and waterfall method. The total number of software projects is over 10,000.

Chaos resolution by agile versus waterfall

The Standish Group International, Inc. (2015). *CHAOS REPORT 2015*. Retrieved from https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf

In the agile methodology we divide our project into smaller, more manageable parts, and we do it little by little, so it will be easier to succeed.

2.3. Scrum

The non-profit website ProyectosAgiles.org defines Scrum as a process in which a set of practices is applied on a regular basis to work collaboratively, in teams, and get the best possible outcome from a project (proyectosagiles.org, 2021).

Schwaber and Sutherland (2020) define it as the lightweight framework that helps individuals, teams, and organizations generate value through adaptive solutions to complex problems.

Specifically, Scrum requires a Scrum Master to foster an environment where:

1. A product owner orders work on a complex problem into a product backlog.
2. The Scrum team converts a selection of work into a Value Increment during a Spring.
3. The Scrum team and its stakeholders inspect the results and adapt for the next Sprint.
4. And the cycle repeats (Schwaber and Sutherland, 2020).

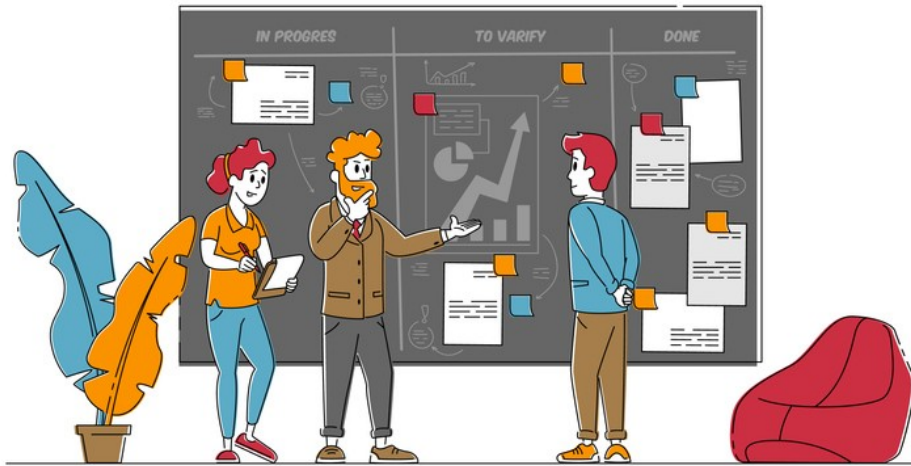
In this topic, we will address all the elements related to the Scrum methodology, which consists of the following:

- Scrum team
 - Developers
 - Product owner
 - Scrum Master
- Scrum events
 - The Sprint
 - Sprint planning
 - Daily scrum
 - Sprint review
 - Sprint retrospective
- Scrum artifacts
 - Product backlog
 - Commitment: Product objective
 - Sprint backlog
 - Commitment: Sprint goal
 - Increment
 - Commitment: Definition of completion

Participants

Developers: individuals on the **Scrum team**, who are committed to creating any aspect of a usable increment in each sprint are developers (Schwaber and Sutherland, 2020). The specific skills needed by developers are often broad and vary by scope of work, However, developers are always responsible for the following:

- Creating a plan for the sprint, the sprint backlog.
- Instill quality by adhering to a **definition of completion**.
- Adapt their plan each day towards the sprint goal.
- Hold each other accountable as professionals.



Product owner: is responsible for maximizing the value of the product resulting from the Scrum team's work. How this is done can vary widely among organizations, Scrum teams, and individuals (Schwaber and Sutherland, 2020). The product owner is also responsible for the effective management of the product backlog, which includes the following:

- Developing and explicitly communicating the product goal.
- Creating and clearly communicating the elements of the product backlog.
- Ordering the elements of the product backlog.
- Ensuring that the product backlog is transparent, visible, and understood.

Although the product owner may delegate tasks to others, he/she is still responsible for getting the job done, and for the product owner to be successful, the entire organization must respect his decisions. These decisions are visible in the content and order of the product backlog and through the inspectable Increment in the Sprint review.

It should be clarified that the product owner is a person, not a committee. The product owner can represent the needs of many stakeholders in the product backlog. Those who want to change the product backlog can do so by trying to convince the product owner.

Scrum Master: is responsible for establishing Scrum as defined in the Scrum guide (Schwaber and Sutherland, 2020). It does this by helping everyone understand the theory and practice of Scrum, both within the Scrum team and the organization.

The Scrum Master is responsible for achieving Scrum team effectiveness. He does this by supporting the Scrum team in improving their practices, within the

Scrum framework. It can be said that Scrum Masters are true leaders who serve the Scrum team and the organization as a whole.

The Scrum Master serves the Scrum team in several ways, including the following:

- Guide team members in being self-managing and cross-functional.
- Helping the Scrum Team focus on creating high-value Increments that meet the **definition of completion**.
- See to the removal of impediments to the Scrum team process.
- Ensure that all Scrum events are performed and are positive, productive, and stay within recommended time limits.

The Scrum Master serves the product owner in several ways, including the following:

- Assist in finding techniques for effective **product goal** definition and product backlog management.
- Helping the Scrum team understand the need for clear and concise product backlog elements.
- Help to establish empirical product planning for a complex environment.
- Facilitate stakeholder collaboration as requested or needed.

The Scrum Master serves the organization in several ways, including the following:

- Lead, coach, and guide the organization in its adoption of Scrum.
- Plan and mentor Scrum implementations within the organization.
- Help employees and stakeholders understand and apply an empirical approach to complex work.
- Eliminate barriers between stakeholders and Scrum teams.

Scrum events

The Sprint is a container for all other events, each event in Scrum is a formal opportunity to inspect and adapt Scrum artifacts. These events are specifically designed to enable the required transparency. Failure to operate any event as prescribed results in missed opportunities to inspect and adapt. Events are used in Scrum to create regularity and minimize the need for undefined Scrum meetings (Schwaber and Sutherland, 2020).

Optimally, all events should be held at the same time and place to reduce complexity.

The Sprint: Sprints are the heart of Scrum, where ideas become value. They are fixed-duration events of one month or less to create consistency. A new Sprint starts immediately after the conclusion of the previous Sprint.

All work necessary to achieve the **product goal**, including Sprint planning, daily Scrums, Sprint review, and Sprint retrospective, occurs within Sprints (Schwaber and Sutherland, 2020).

During the Sprint:

- No changes are made that jeopardize of **the Sprint objective**.
- Quality does not decrease.
- The product backlog is refined as needed.
- Scope can be clarified and renegotiated with the product owner as more is learned.

Sprints enable predictability by ensuring that progress toward a product goal is inspected and adapted at least every calendar month. When the horizon of a Sprint is too long, the Sprint goal may become invalid, complexity may grow, and risk may increase. Shorter Sprints can be used to generate more learning cycles and limit the risk of cost and effort to a shorter period. Each Sprint can be considered a short project (Schwaber and Sutherland, 2020).



There are several practices for forecasting progress, such as burn-downs, burn-ups, or cumulative flows. While they have proven useful, they do not replace the importance of empiricism. In complex environments, what will happen is unknown. Only what has already happened can be used for forward-looking decision-making (Schwaber and Sutherland, 2020).

A Sprint could be cancelled if the Sprint objective becomes obsolete. Only the product owner has the authority to cancel the Sprint.

Sprint planning: Sprint planning initiates the Sprint by establishing the work to be performed for the Sprint. The Scrum Team creates this resulting plan through collaborative work (Schwaber and Sutherland, 2020).

The product owner ensures that the attendees are prepared to discuss the most important elements of the product backlog and how they relate to the

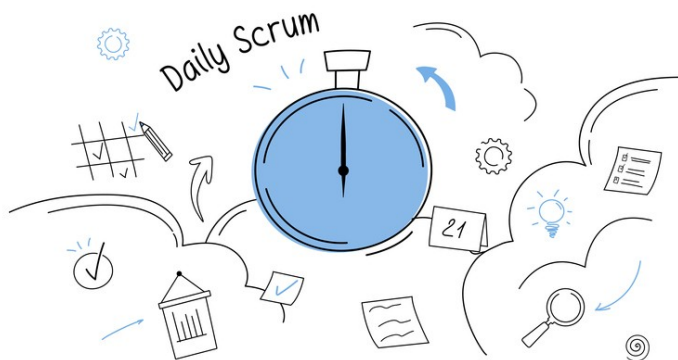
product goal. The Scrum team may also invite others to attend the Sprint Planning to provide advice.

The Sprint planning addresses the following topics:

- **Topic one:** why is this Sprint valuable? The product owner proposes how the product can increase its value and usefulness in the current Sprint. Next, the entire Scrum team collaborates to define a Sprint goal that communicates why the Sprint is valuable to the stakeholders. The Sprint goal must be completed before the end of Sprint planning.
- **Topic two:** what can be done in this Sprint? Through a conversation with the product owner, the Developers select items from the product backlog to include in the current Sprint. The Scrum team can enhance these items during this process, which increases understanding and confidence. Selecting when to complete within a Sprint can be a challenge. However, the more the developers know about their past performance, their current capability, and their **definition of completed**, the more confident they will be in their forecasts for the Sprint.
- **Topic three:** how will the chosen work be done? For each item in the selected product backlog, the developers plan the work required to create an increment that meets the **definition of completion**. Often, this is done by breaking down the product backlog items into smaller work items of one day or less. How this is done is at the sole discretion of the developers. No one else tells them how to turn product backlog items into increments of value.

"The Sprint goal, the product backlog items selected for the Sprint, plus the plan for delivering them are together called the Sprint backlog"
(Schwaber and Sutherland, 2020).

Sprint Planning has a time limit of maximum eight hours for a one-month Sprint. For shorter Sprints, the event is usually of shorter duration.



Daily Scrum: The purpose of the daily Scrum is to inspect progress towards the Sprint goal and adapt the Sprint Backlog as needed, adjusting incoming planned work. The daily Scrum is a 15-minute event for the Scrum team developers. To reduce complexity, it is held at the same time and place every business day of the Sprint. If the product owner or Scrum Master is actively working on elements of the Sprint backlog they participate as developers (Schwaber and Sutherland, 2020).

Developers can select the structure and techniques they want, as long as their daily Scrum focuses on progress toward the Sprint goal and produces a workable plan for the next day's work. This creates focus and improves self-management.

Daily Scrum's improve communication, identify impediments, promote rapid decision-making and, consequently, eliminate the need for other meetings.

The daily Scrum is not the only time when developers can adjust their plan. They often meet during the day for more detailed discussions on how to adapt or re-plan the rest of the Sprint work.

Sprint review: the purpose of the Sprint review is to inspect the outcome of the Sprint and determine future adaptations. The Scrum team presents the results of their work to key stakeholders and progress toward the product goal is discussed (Schwaber and Sutherland, 2020).

During the event, the Scrum team and stakeholders review what was accomplished and what has changed in their environment. Based on this information, attendees collaborate on what to do next. The product backlog can also be adjusted to meet new opportunities. **The Sprint review is a working session, and the Scrum team should avoid limiting it to a presentation.**

The Sprint review is the second to last event of the Sprint and has a time limit of maximum four hours for a one-month Sprint. For shorter Sprints, the event is usually shorter.

Sprint retrospective: the purpose of the Sprint retrospective is to plan ways to increase quality and effectiveness. The Scrum Team inspects how the last Sprint went with respect to people, interactions, processes, tools, and their **definition of completion**. The elements inspected usually change depending on the scope of work. Assumptions that led them astray are identified and their origins explored. The Scrum team analyzes what went well during the Sprint, what problems were encountered, and how those problems were (or were not) resolved.

The Scrum team identifies the most useful changes to improve its effectiveness. The most impactful improvements are addressed as soon as possible. They can even be added to the Sprint Backlog for the next Sprint.

The Sprint retrospective concludes the Sprint. It is time-limited to a maximum of three hours for a one-month Sprint. For shorter Sprints, the event is usually shorter.

Scrum artifacts

Scrum artifacts represent work or value. They are designed to maximize transparency of key information, so everyone inspecting them has the same basis for adaptation (Schwaber and Sutherland, 2020).

Each artifact contains a commitment to ensure that it provides information that improves transparency and focus against which progress can be measured.

- For the **product backlog**, it is the **product goal**.
- For the **Sprint backlog**, it is the **Sprint goal**.
- For the **increment**, it is the **definition of completion**.

These commitments exist to reinforce Scrum empiricism and Scrum values for the Scrum team and its stakeholders.

Product backlog: The product backlog is an emergent, ordered list of what is needed to improve the product. It is the sole source of the work performed by the Scrum team (Schwaber and Sutherland, 2020).

The elements of the product backlog, which the Scrum team can release for **completion** within a Sprint, are considered ready to be selected in a Sprint planning event. They usually acquire this degree of transparency after refinement activities. Product backlog refinement is the act of breaking down and further defining product backlog items into smaller, more precise items. This is an ongoing activity to add details, such as a description, order, and size. Attributes usually vary depending on the scope of work.

The developers who will perform the work are responsible for sizing. The product owner can influence the developers by helping them understand and select their best alternatives.

Commitment: product goal. The product goal describes a future state of the product that can serve as a target for the Scrum team to plan for. The product goal is in the product backlog. The rest of the product backlog emerges to define "what" will accomplish the product goal.

A product is a vehicle to deliver value. It has a clear boundary, known stakeholders, well-defined users, or customers. A product can be a service, a physical product or something more abstract (Schwaber and Sutherland, 2020).

The product goal is the Scrum team's long-term goal. They must meet (or abandon) one goal before taking on the next.

Sprint backlog: The Sprint backlog consists of the Sprint goal (why), the set of product backlog items selected for the Sprint (what), as well as an action plan to deliver the increment (how) (Schwaber and Sutherland, 2020).

The Sprint backlog is a plan made by and for the developers. It is a highly visible, real-time picture of the work that the developers plan to do during the Sprint to achieve the Sprint goal. Consequently, the Sprint backlog is updated throughout the Sprint as more is learned. It should have enough details for them to inspect their progress in the daily Scrum.

Commitment: Sprint goal. The Sprint goal is the sole purpose of the Sprint. While the Sprint goal is a commitment from the developers, it provides flexibility in terms of the exact work needed to achieve it. The Sprint goal also creates consistency and focus, which encourages the Scrum team to work together rather than on separate initiatives.

The Sprint goal is created during the Sprint planning event and added to the Sprint backlog. As the developers work during the Sprint, they keep the Sprint goal in mind. If the work turns out to be different from what they expected, they collaborate with the product owner to negotiate the scope of the Sprint backlog within the Sprint without affecting the Sprint goal.

Increment: an increment is a concrete stepping block towards the product's goal. Each Increment is added to all previous increments and thoroughly verified, which ensures that all increments work together. To provide value, the Increment must be usable (Schwaber and Sutherland, 2020).

Multiple increments can be created within a Sprint. The sum of the increments is presented in the Sprint review thus supporting empiricism. However, an Increment can be delivered to stakeholders prior to the Sprint. The Sprint review should never be considered a gateway to release value.

Work cannot be considered part of an increment unless it meets **the definition of completed.**

Commitment: definition of completion. The definition of completion is a formal description of the state of the Increment when it meets the required quality measures for the product. The moment a product backlog item meets the definition of completion, an Increment is born (Schwaber and Sutherland, 2020).

The definition of completion creates transparency by providing everyone with a shared understanding that the work was completed as part of the Increment. If an item in the product backlog does not meet the definition of completed, it

cannot be published or presented in the Sprint review. Instead, it returns to the product backlog for future consideration.

If the definition of completion for an Increment is part of the organizational standards, all Scrum teams must follow it as a minimum. If it is not an organizational standard, the Scrum team must create an appropriate completion definition for the product.

Developers must adhere to the definition of completion. If there are several Scrum teams working together on a product, they must mutually define and adhere to the same completion definition.

It is highly recommended to keep in mind the following statement:

"One team focused on one product". The goal was to eliminate the concept of a separate team within a team that has led to "proxy" or "us and them" behavior between the product owner and the development team. Now there is only one Scrum Team focused on the same goal, with three different sets of managers: product owner, Scrum Master, and developers.

This means that there is only one team of developers, the old practices of separating and competing teams by cells (backend team, frontend team, UX designers, testers, among others) no longer exist. Now everyone is responsible for the product, everyone is a developer.

Conclusion

Everything related to project management is an exciting world that, as a developer, you need to know about. There are points of view where experienced talent seeks to be on the development side, but there are also developers who by knowing project management may be able to be the captains of that ship. Understanding how project planning works, how it is managed and how it is executed makes you an important part of a team. Remember that developments today are done entirely by well-established teams.

Checkpoint

Make sure that you:

- Differentiate the types of project management methodologies.
- Identify the approach your project requires.
- Comprehend the fundamentals of Scrum.
- Identify the master of the terms used in Scrum.

References

-
- agilemanifesto.org. (2001). *Manifesto for Agile Software Development*. Retrieved from <https://agilemanifesto.org/iso/en/manifesto.html>
 - proyectosagiles.org. (2021). *Qué es SCRUM*. Retrieved from <https://proyectosagiles.org/que-es-scrum/>
 - Schwaber, K., and Sutherland, J. (2020). *The Scrum Guide*. Retrieved from <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf#zoom=100>
 - The Standish Group International, Inc. (2015). *CHAOS REPORT 2015*. Retrieved from https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf

Additional Resources

The following links do not belong to Tecmilenio University, when accessing to them, you must accept their terms and conditions.

Videos

- To learn about **Scrum vs Waterfall**, watch the following video: Code With RaiGenics. (2018, December 1). *Scrum Vs Waterfall: 3 fundamental benefit of Scrum* [Video file]. Retrieved from <https://www.youtube.com/watch?v=VYZDyLX7J4>
- To learn about **Scrum**, watch the following video: freeCodeCamp.org. (2018, February 12). *Scrum in 16 minutes* [Video file]. Retrieved from <https://www.youtube.com/watch?v=vuBFzAdaHDY>
- To learn about **Agile scrum**, watch the following video: Simplilearn. (2020, August 26). *Agile Scrum Full Course In 4 Hours | Agile Scrum Master Training | Agile Training Video | Simplilearn* [Video file]. Retrieved from <https://www.youtube.com/watch?v=VFQtSqChIsk>

Readings

- To learn more about **Agile methodology**, we recommend reading: agilemanifesto.org. (2021). *Manifesto for Agile Software Development*. Retrieved from <https://agilemanifesto.org/iso/en/manifesto.html>

- To learn more about **Scrum vs Water**, we recommend reading: Chaudhary, L. (2022). *Scrum vs Waterfall*. Retrieved from <https://www.educba.com/scrum-vs-waterfall/>

Activity 2

Description

The student will learn more about the differences in the predominant methodologies in the market for project planning.

Objective

Computer with Internet access to research concepts.

Instructions

After what you have seen in class, design a concept map in which you describe the differences between agile and traditional methodologies.

1. Include the templates and formats that each approach offers or requires.
2. List the applications and web applications that are on the market that use these methodologies.
3. The map can be done by hand or using any diagramming software.

Deliverable(s)

Screenshots, photograph, or file with the concept map in a document in concept map format.

Evaluation criteria

1. Legible, understandable, and descriptive concept map.
2. List of templates or formats.
3. List of applications in the market with as much information as possible (free or paid, web page, among others).
4. References from where the information was obtained.

Homework 2

Instructions

Research everything related to **Scrum board** and include the following in a document.

1. The concept.
2. A graphical representation of the Scrum board.
3. The applications that can be used for the Scrum board.
4. Describe in your own words whether it is possible to apply the concept to your academic life and how you could implement Scrum board.

Deliverable(s)

Submit your results in the corresponding space of the platform.