Agile

“Agility is the ability to create and respond to change, aiming to create value in a turbulent business environment.

It is the ability to balance flexibility and stability.”

Jim Highsmith

Agile

In February 2001, seventeen prominent figures in the software development field gathered to rethink and challenge the prevailing mindset regarding management practices.

Thus, the Agile Manifesto emerged:

https://agilemanifesto.org/

Built on four major pillars, essential to understand any agile method:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

Agile - Scrum

- Scrum is the most popular of the agile frameworks used in Project Management;
- It ensures the rapid delivery of value throughout the project;
- It ensures transparency in communication and creates an environment for everyone to collaborate and improve;
- It is based on cross-functional, self-organized, and autonomous teams that divide project work into small, focused cycles called Sprints.

Intro to Scrum in under 10 minutes

https://www.youtube.com/watch?v=XU0llRltyFM

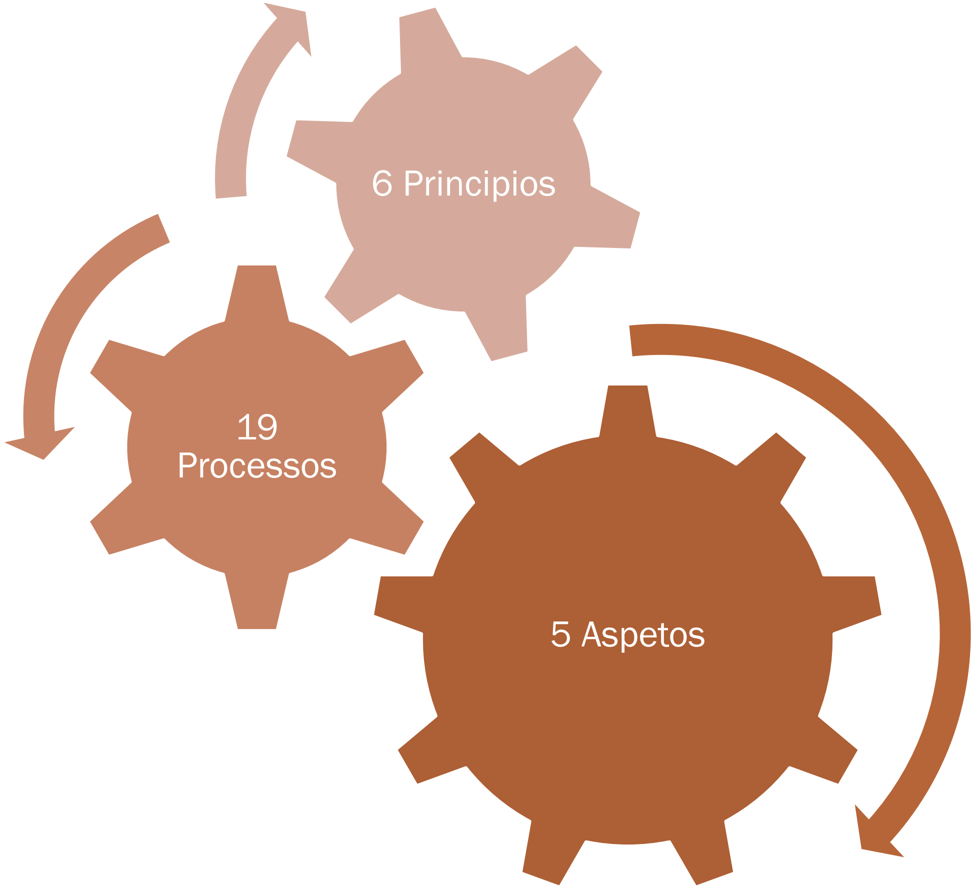
Agile

Framework of Scrum Book of Knowledge

6 Principles
The guiding principles for applying SCRUM. They are mandatory in all projects.

5 Aspects
The aspects that must be identified and managed in a SCRUM project.

19 Processes
The steps and moments of the project, including inputs, tools, and outputs.

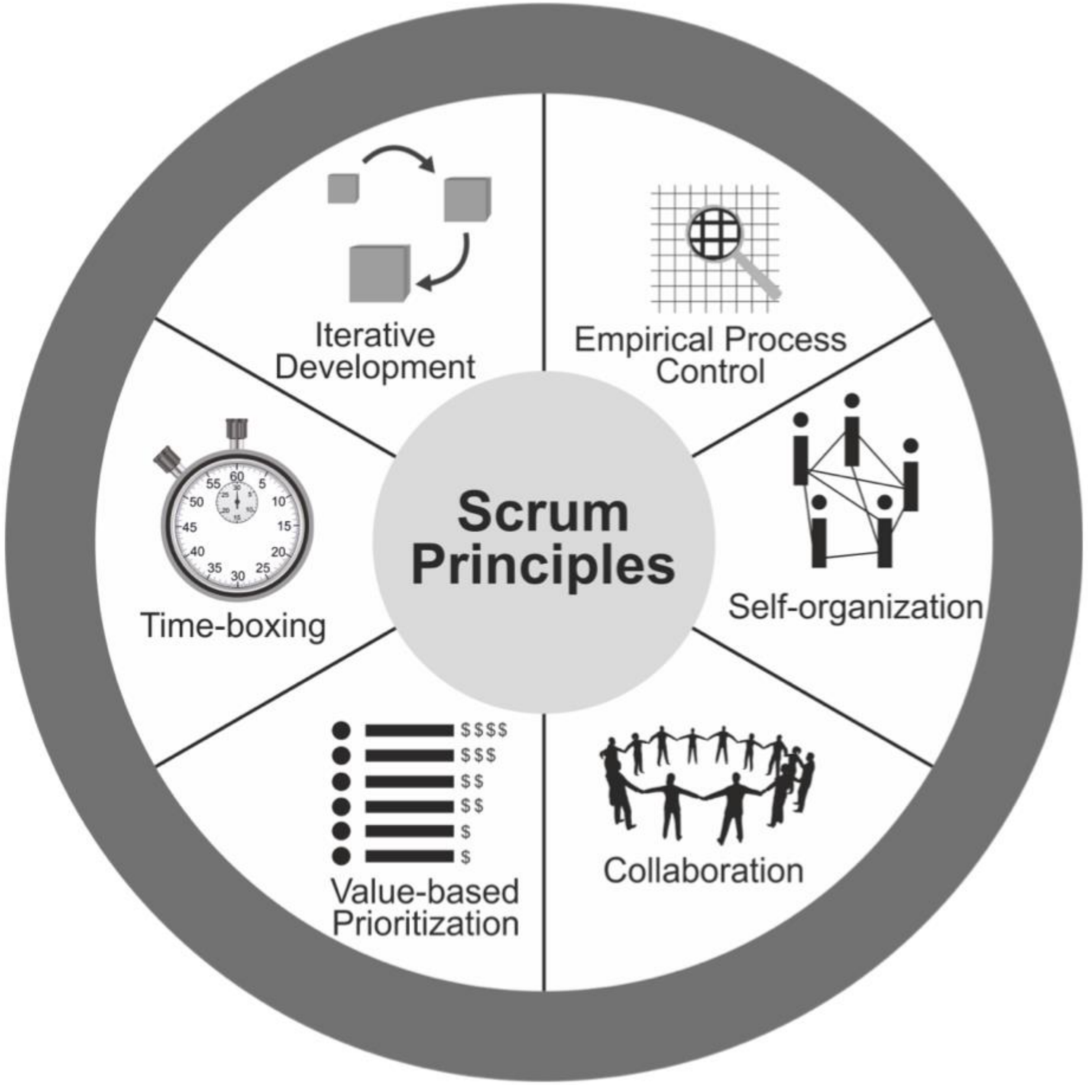


Agile

Framework of Scrum Book of Knowledge

6 Principles

1. Empirical process control;
2. Self-organization;
3. Collaboration;
4. Value-based prioritization;
5. Time-boxing;
6. Iterative development.

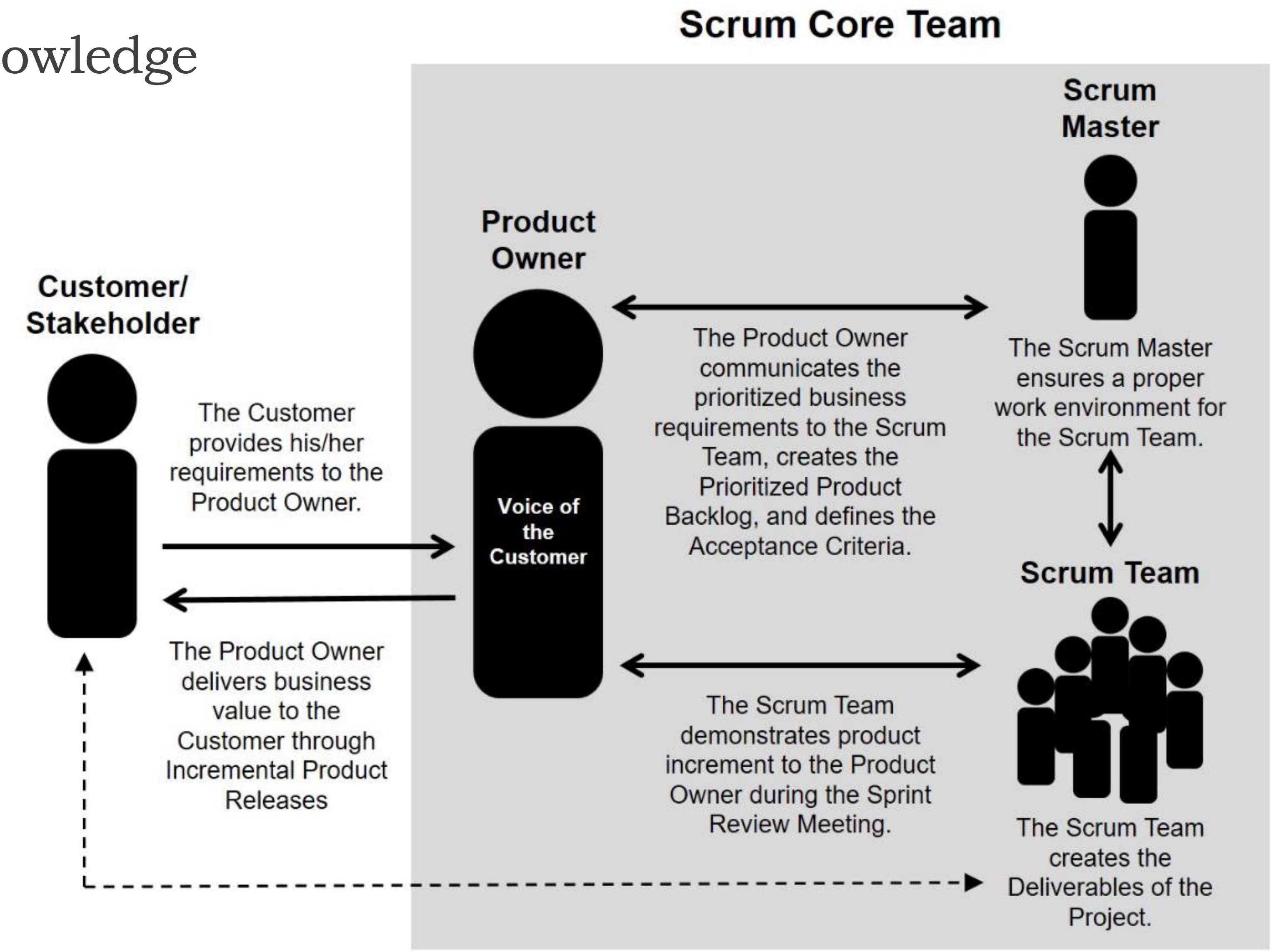


Agile

Framework of Scrum Book of Knowledge

5 Aspects

1. Organization (figura ao lado);
2. Business justification;
3. Quality;
4. Change;
5. Risk;



Scrum Core Team

Customer/ Stakeholder

Product Owner

Voice of the Customer

Scrum Master

Scrum Team

The Customer provides his/her requirements to the Product Owner.

The Product Owner delivers business value to the Customer through Incremental Product Releases

The Product Owner communicates the prioritized business requirements to the Scrum Team, creates the Prioritized Product Backlog, and defines the Acceptance Criteria.

The Scrum Master ensures a proper work environment for the Scrum Team.

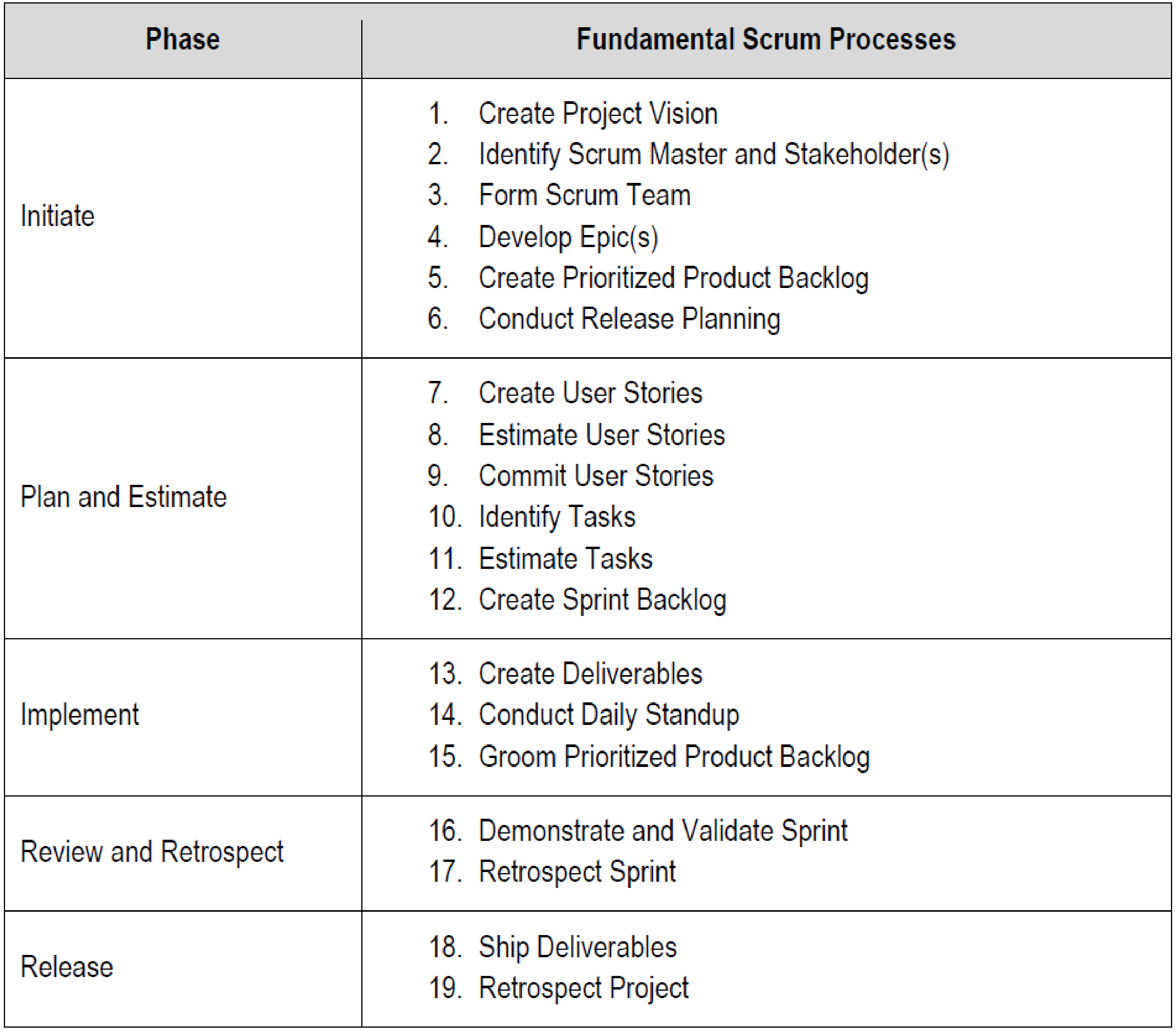
The Scrum Team demonstrates product increment to the Product Owner during the Sprint Review Meeting.

The Scrum Team creates the Deliverables of the Project.

Agile

Framework of Scrum Book of Knowledge

The 5 stages and the **19 processes** of Scrum



Phase	Fundamental Scrum Processes
Initiate	1. Create Project Vision 2. Identify Scrum Master and Stakeholder(s) 3. Form Scrum Team 4. Develop Epic(s) 5. Create Prioritized Product Backlog 6. Conduct Release Planning
Plan and Estimate	7. Create User Stories 8. Estimate User Stories 9. Commit User Stories 10. Identify Tasks 11. Estimate Tasks 12. Create Sprint Backlog
Implement	13. Create Deliverables 14. Conduct Daily Standup 15. Groom Prioritized Product Backlog
Review and Retrospect	16. Demonstrate and Validate Sprint 17. Retrospect Sprint
Release	18. Ship Deliverables 19. Retrospect Project

Agile

Roles in SCRUM

The roles in Scrum are divided into two major categories:

- **Core Roles:** These are the primary, necessary, and mandatory functions to produce the project's product or service. They are responsible for the success of each Sprint and the project as a whole.

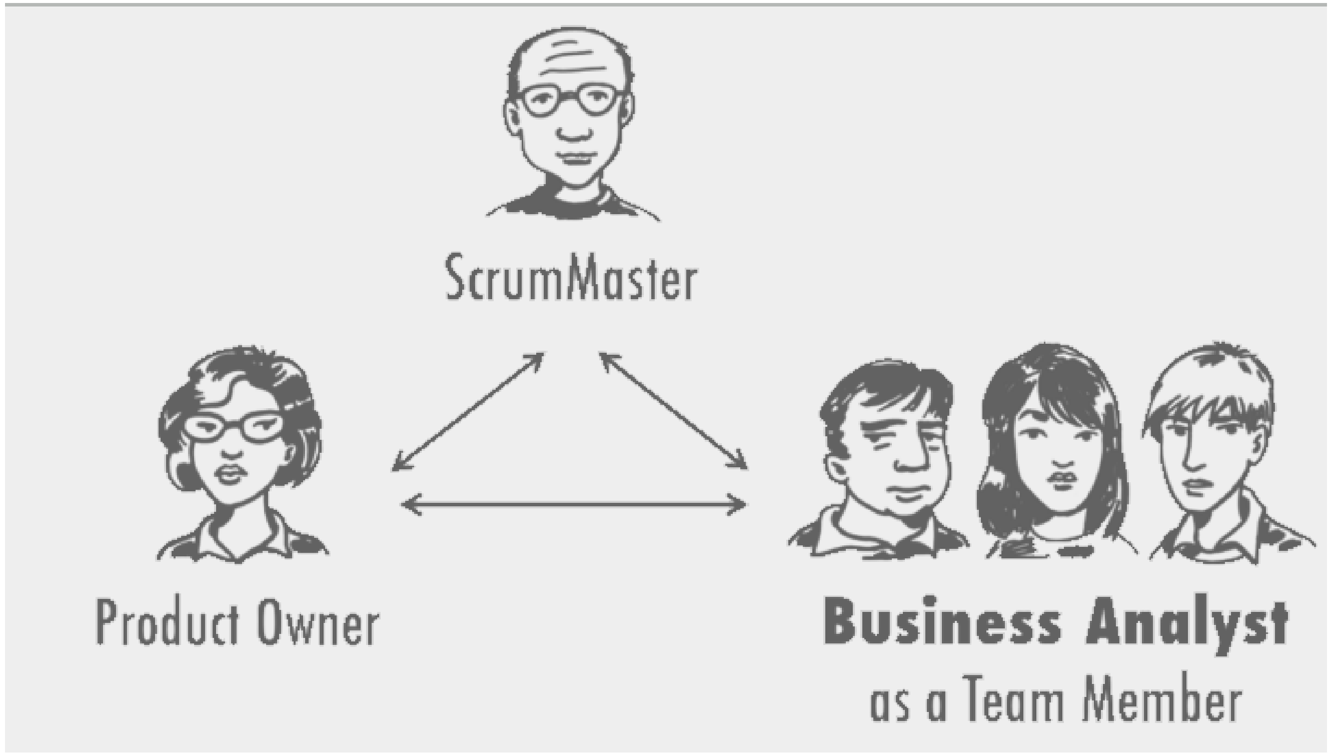
- **Non-Core Roles:** Not necessarily required for the Scrum project; these roles may include team members interested in the project who do not have a formal role in the project team. They may interact with the team but are not responsible for the project's success.

Agile

Roles at SCRUM

Core roles

- Product Owner (PO)
- Scrum Master (SM)
- Scrum Team (ST)



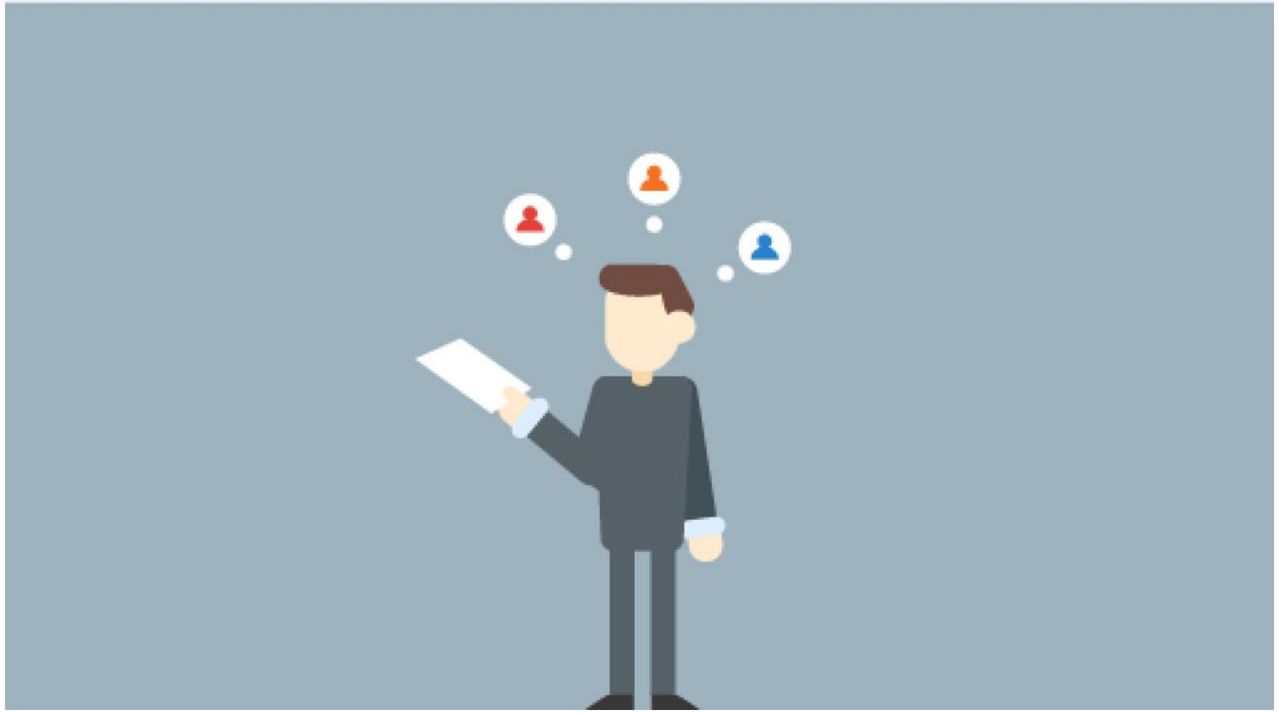
Important Note: The role of a business analyst is integrated into the Scrum Team; it is not necessarily the Product Owner (PO).

Agile

Roles at SCRUM

Product Owner (Core Role)

- It is the representative of business stakeholders. It is the representative of stakeholders in the project team.
- For this reason, it is sometimes called the Voice of the Customer. It is responsible for communicating clearly with the Scrum Team about the functionalities and the expected value of them.
- Its main focus is on Return on Investment (ROI).

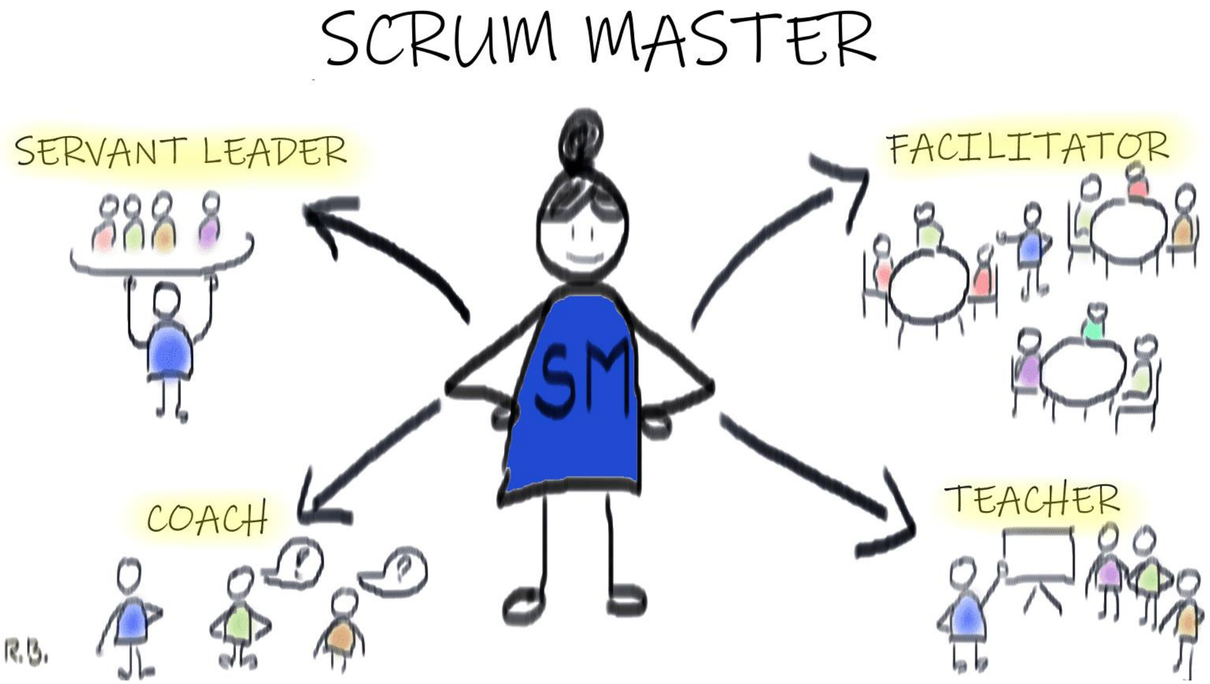


Agile

Roles at SCRUM

Scrum Master (Core Role)

- Its main responsibility is to ensure the correct execution of the Scrum process by all members of the Core Team, including the Product Owner.
- It is a facilitator: it must ensure that the team has the means, tools, and data necessary to perform the work. It removes difficulties and impediments.
- It is based on the concept of Servant Leadership, where results are achieved by satisfying the needs of the people they lead.



Agile

Roles at SCRUM

Scrum Team (Core Role)

- The ultimate responsible parties for:
  - Understanding the business requirements specified by the PO
  - Estimating the User Stories
  - Creating and delivering the product
- The appropriate size should be between 6 to 10 members (big enough to ensure the necessary skills with redundancy and small enough to collaborate efficiently). In the case of software development projects, it is composed of various technical profiles, including developers, testers, designers, DBAs, sysadmins, etc.

Agile

Roles at SCRUM

Non-Core roles

- Stakeholders:
  - Internal or external clients who acquire the product/service or other project deliverables.
- Users:
  - Those who will directly use the product, the end-users.
- Sponsor:
  - The ultimate project sponsor who ensures resources and support for the project.
- Vendors:
  - Suppliers of products and/or services necessary for implementation.
- Scrum Guidance Body:
  - Not all organizations have this role. It is a governance body that supports the organization in Scrum best practices. To facilitate understanding, it can be seen as a kind of PMO (Project Management Office) for Scrum. It does not directly interfere with the project.

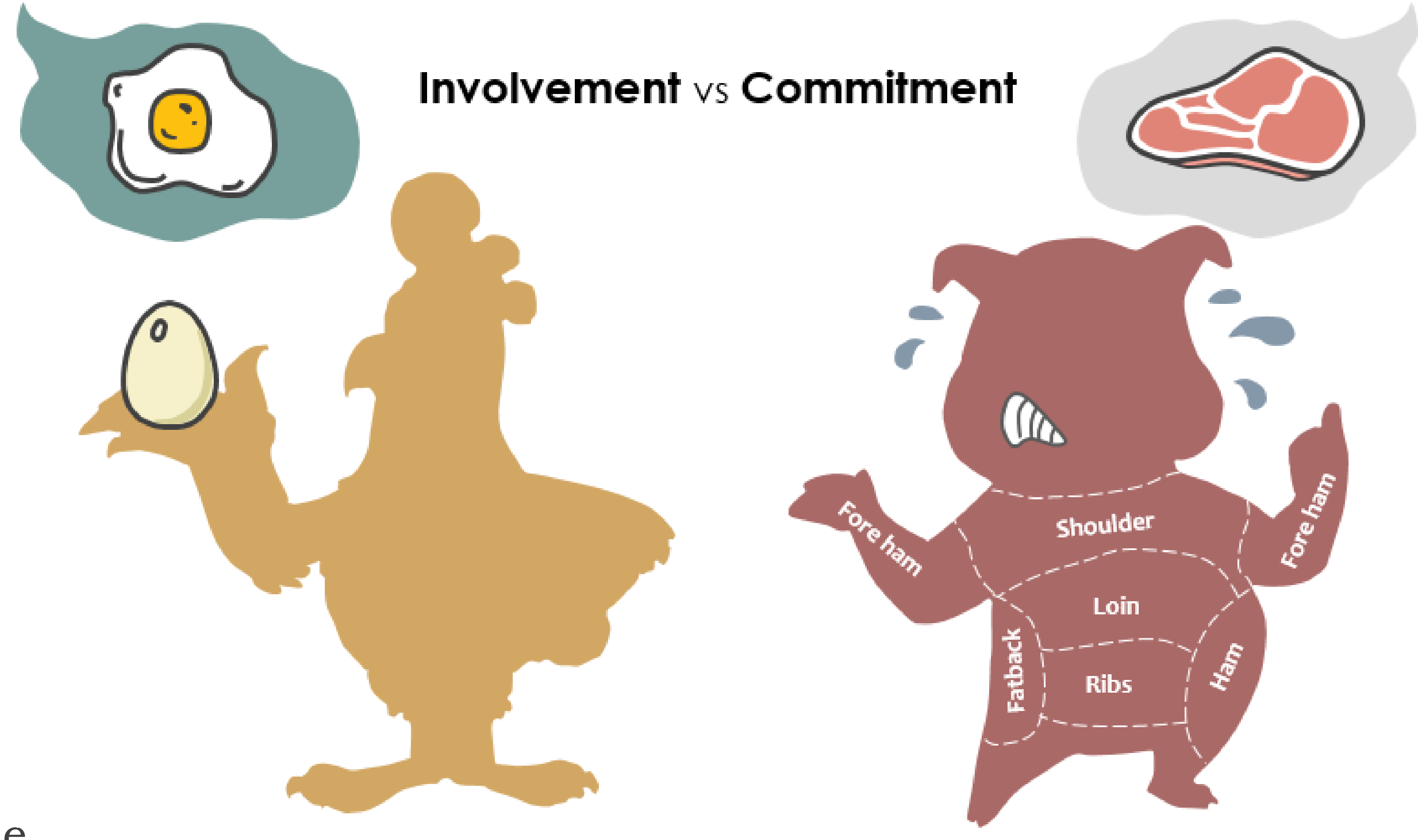
Agile

Roles at SCRUM

The fable of the chicken and the pig illustrates the difference in commitment and involvement between core and non-core roles.

That's why it is said that core roles are the pigs, and non-core roles are the chickens.

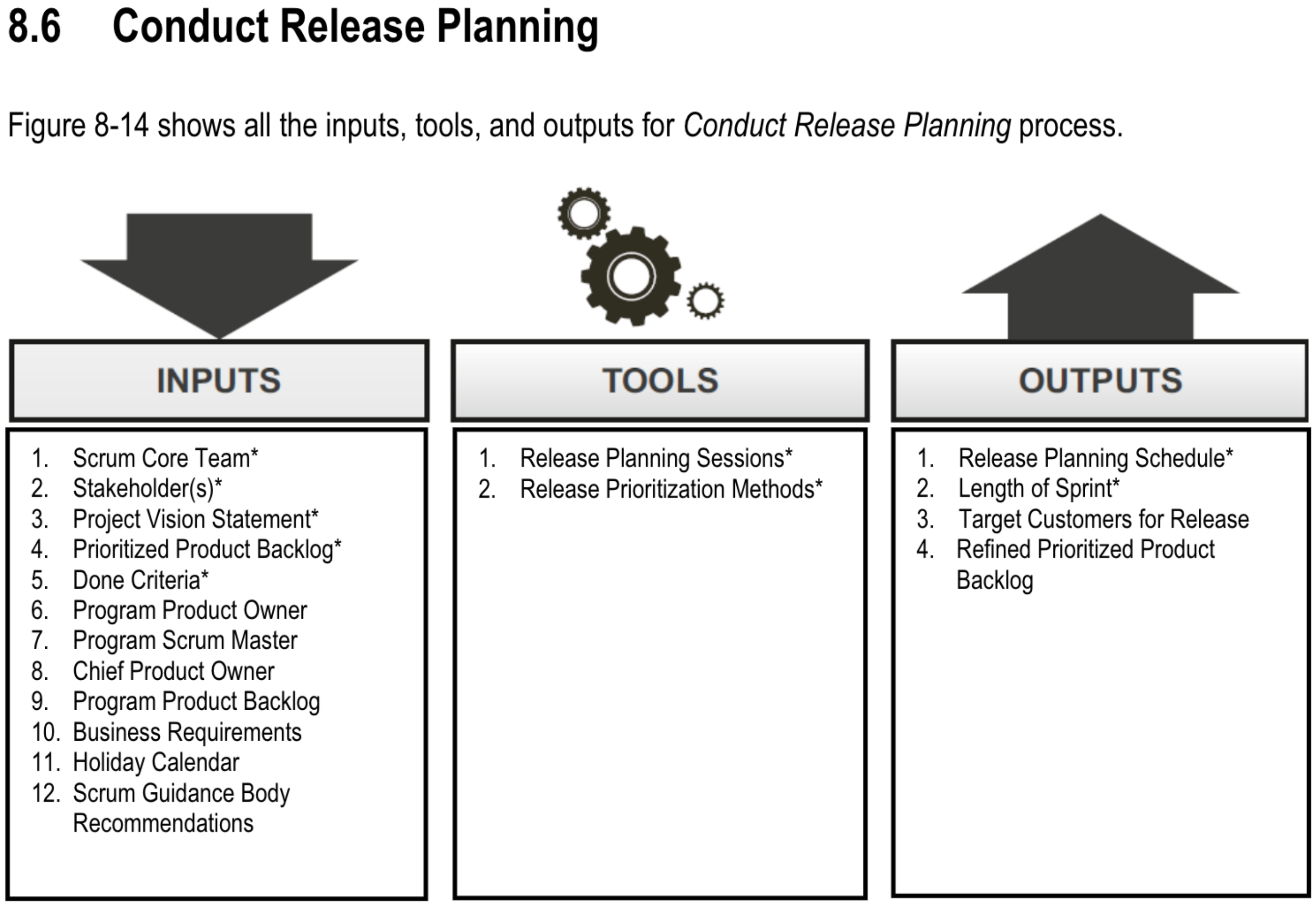
The pig provides the bacon, demonstrating full commitment, while the chicken supplies the egg, showing only partial involvement.!



Agile

To better understand the processes of Scrum, we will refer to the Scrum Body of Knowledge (available on Moodle), which is a framework that helps us understand each process.

- **Inputs -** roles, documents, information necessary for each process
- **Tools -** practices, methodologies, ceremonies used in each phase
- **Outputs -** the deliverables of each process



8.6 Conduct Release Planning

Figure 8-14 shows all the inputs, tools, and outputs for *Conduct Release Planning* process.

INPUTS

1. Scrum Core Team*
2. Stakeholder(s)*
3. Project Vision Statement*
4. Prioritized Product Backlog*
5. Done Criteria*
6. Program Product Owner
7. Program Scrum Master
8. Chief Product Owner
9. Program Product Backlog
10. Business Requirements
11. Holiday Calendar
12. Scrum Guidance Body Recommendations

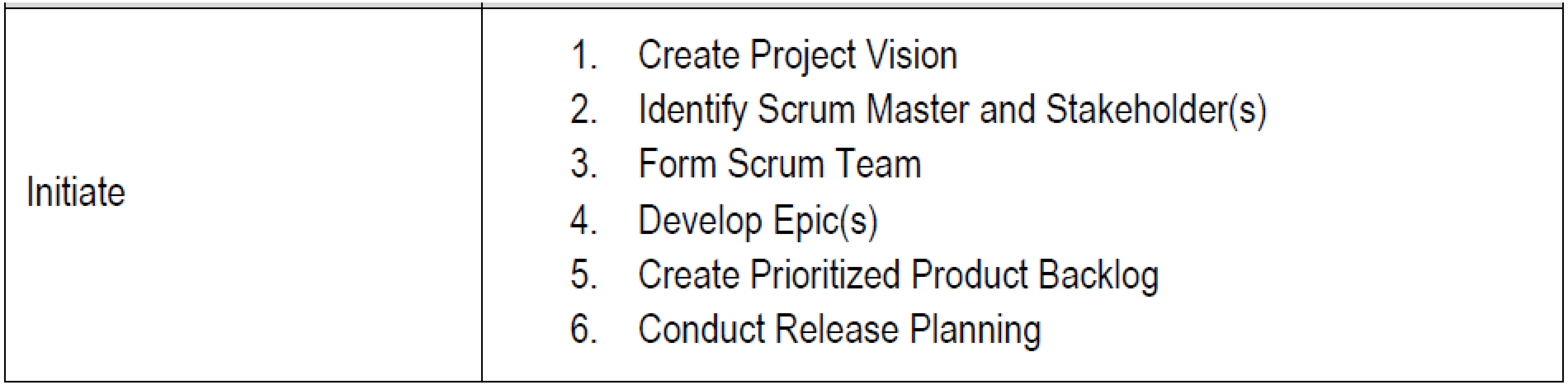
TOOLS

1. Release Planning Sessions*
2. Release Prioritization Methods*

OUTPUTS

1. Release Planning Schedule*
2. Length of Sprint*
3. Target Customers for Release
4. Refined Prioritized Product Backlog

Agile



Initiate	1. Create Project Vision 2. Identify Scrum Master and Stakeholder(s) 3. Form Scrum Team 4. Develop Epic(s) 5. Create Prioritized Product Backlog 6. Conduct Release Planning

Agile

INITIATE Phase: As the name suggests, it refers to the initiation of the project, and its stages are executed only once in the initial phase. It consists of 6 processes.

Create Project Vision: This is the starting point of the project, taking the business case as input. At this moment, the sponsor and/or senior management select the Product Owner. From this point forward, the PO is responsible for producing the Project Vision Statement (equivalent to the Project Vision Scope) by gathering requirements from stakeholders.

The Project Vision Statement should have the appropriate level of detail—neither too broad nor excessively detailed. It should focus on the problem to be solved and not on the solution. The solution will be worked on by the Scrum Core Team throughout the project!

Agile

Identify Scrum Master & Stakeholders: It is the responsibility of the Product Owner (PO) to select the Scrum Master.

The criteria for selecting the Scrum Master (SM) should be clear and typically revolve around certain technical and behavioral competencies:

- Problem-solving skills
- Availability
- Commitment to the project
- Servant Leadership characteristics

The PO and the SM should collaboratively identify stakeholders and ensure an effective communication plan. Don't overlook the criticality of managing stakeholders who may have significant influence on the project!

Agile

Form the Scrum Team: The Product Owner and the Scrum Master should collaborate in creating the Scrum Team.

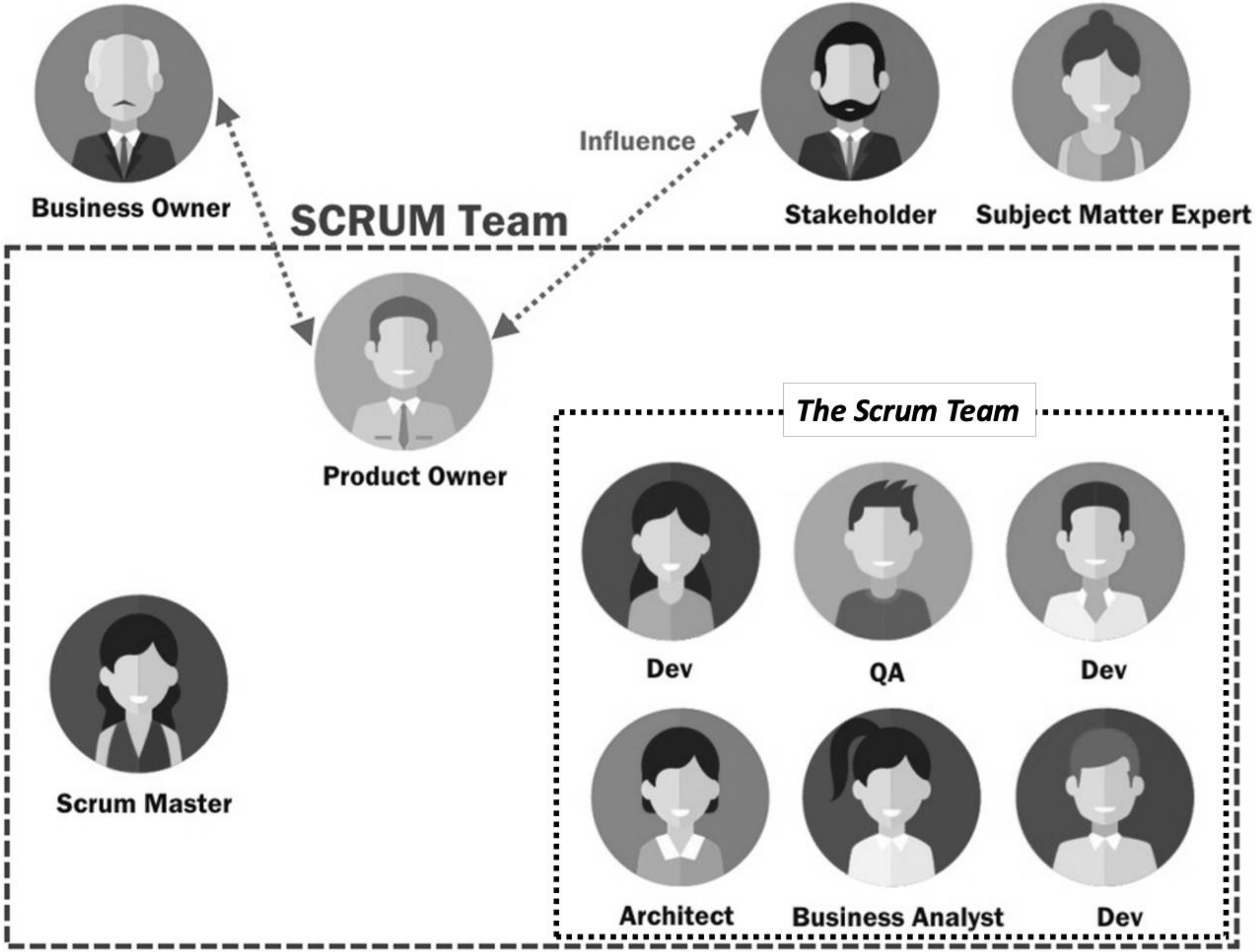
Considerations:

- The team size should be between 6 to 10 members and should cover all needs in terms of technical skills.
- For the team to be self-organized, there needs to be a balance in terms of seniority. The team should not be overly junior!

Agile

Once the Scrum Team selection is complete, the Scrum Core Team is considered complete!

Recalling that the Scrum Core Team is composed of the PO (Product Owner), SM (Scrum Master), and ST (Scrum Team).



Agile

Develop Epic(s): Epics are extended User Stories with a low level of detail, written at the beginning of the project, reflecting features or product requirements. Typically, they represent reasonably large chunks of work that may take more than one sprint to implement.

Each epic will lead to one or more refined and detailed User Stories that will be planned to be integrated and completed during a sprint.

The PO is primarily responsible for developing epics, although the Scrum Team can also collaborate!

Epics can be identified and created in User Group Meetings, User Story Workshops, Focus Groups Meetings, and questionnaires.

Agile

Develop Epic(s): Another deliverable from this process is personas, which represent the future users of the product.

Personas play a significant role in the development of User Stories (to be discussed later).

Examples of personas:
- End users of the application
- Users for consultation/supervision only
- Users for data entry
- Users for data collection for report generation, etc.
- Users who will administer the system

Personas

Real Project Example

Fernando Matos

Operador de Abastecimento de Colas

Colaboradores como o Fernando necessitam de manter as linhas de produção da Tabaqueira devidamente abastecidas com Colas.

Trabalha no chão de fábrica, e utiliza um dispositivo móvel para executar todas as suas tarefas.

Carla Oliveira

Supervisor

A Carla é responsável por manter toda a informação de referência sobre os Operadores, Turnos, Circuitos e Pontos de Entrega.

Tem ainda a responsabilidade de interagir com o Gestor de Armazém sobre stocks de Colas, e reagir a alarmística proveniente do chão de fábrica.

Agile

Create prioritized product backlog: The **product backlog** is a list of features (epics) to be implemented. This product backlog list should be ordered by priority, with the most prioritized items at the top.

Relevant concepts:

- The Product Owner is responsible for the final decision on the prioritization of epics.
- Prioritization is done according to three vectors: Value, Risk, and Uncertainty, and Technical Dependencies between tasks.
- There are four prioritization methods that can be used (although each organization is free to define its prioritization criteria):
  - Kano Model
  - MoSCoW
  - Paired Comparison
  - 100 Points Method

Agile

Kano Model

A framework designed for prioritizing features in a product based on the likelihood of satisfying and/or delighting customers, considering the cost/benefit relationship. In Scrum, Epics are classified into four categories:

1. Exciters/Delighters:

- Features that are new or of high value to the customer.

2. Satisfiers:

- Features that deliver value to the customer.

3. Dissatisfiers:

- Features that, when present, do not affect customer satisfaction levels. However, their absence will lead to dissatisfaction/disappointment. They are considered necessities and must be included. For example, a web app not being responsive.

4. Indifferent:

- Features that neither positively nor negatively influence. They should be given low priority or even eliminated from the scope.

Watch an explanatory video

https://www.productplan.com/glossary/kano-model/?wvideo=9kboybqg9g

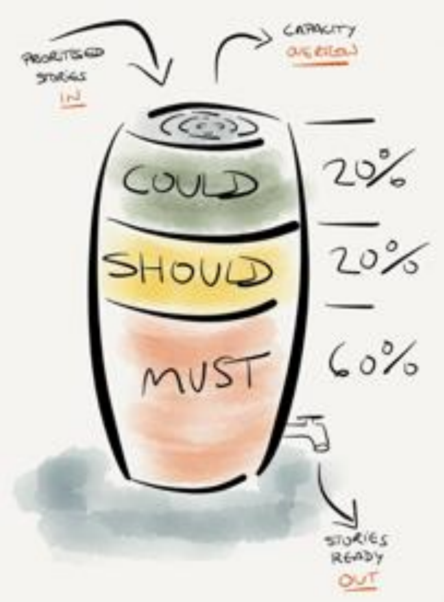
Agile

MoSCoW method

The method is known as the MoSCoW Method, and the name derives from the letters:

- **Must have:** Represents features that are non-negotiable and must be incorporated into the product.
- **Should have:** Refers to features that, while not vital, add value to the product.
- **Could have:** These are the so-called "Nice to have." If left out, the impact on quality is reduced.
- **Won't have:** Involves features that are clearly not priorities at the moment.

To prevent abuses in classifying everything as "Must have," the following reasoning is accepted.



Agile

Paired Comparision

This is a method that involves comparing all epics with each other, deciding which one is the most important, to eventually obtain a prioritized list. For example, using a symmetric matrix:

Step 1: In each matrix entry, indicate the most important epic.

Step 2: In addition to indicating the most important epic, specify the degree of superiority using a scale of 1 to 3, where 1 is slightly more important and 3 is much more important.

Step 3: Once the matrix is completed, evaluate the weight of each epic in the total:

- A = 3 (37.5%, 3 points out of a total of 8).
- B = 1 (12.5%, 1 point out of a total of 8).
- C = 4 (50%, 2 points out of a total of 8).
- D = 0

	A: Epic A	B: Epic B	C: Epic C	D: Epic D
A: Epic A		A, 2	C, 1	A, 1
B: Epic B			C, 1	B, 1
C: Epic C				C, 2
D: Epic D				

Conclusão: a lista ordenada é C, A, B e D

Agile

100 Points Method

Many times also known as the 100-Dollar Method, it involves distributing 100 points among all team members who are prioritizing epics. These team members then assign points to each epic based on its relevance:

Epic	Marketing	Sales	Customer care	Total
Epic A	30	25	35	90
Epic B	20	15	25	60
Epic C	25	25	20	70
Epic D	25	35	20	80
Total	100	100	100	

Conclusion: The prioritized list will be A, D, C, and B.

Agile

Create prioritized product backlog.

The primary output is a product backlog ordered by the priority of epics.

Concepts to remember:

- There is no method that is inherently better than others. The organization should choose the one that it believes best fits its reality.
- Choosing a method is critical, and it must be accepted by the entire organization and become the sole truth to consider.
- Any adopted method should contain rules that prevent the emergence of situations where "everything is important, it's not possible to define priorities." Saying that is equivalent to not making a decision!

Agile

Conduct release planning: The stage in which the release of the product is defined. It is also when the duration of the sprints is determined, which typically remains consistent throughout the project. The duration of a sprint can vary between 1 and 6 weeks, with 4 weeks being recommended!

Important: Not to confuse sprints with product releases! Each organization has its release strategy, which can be:

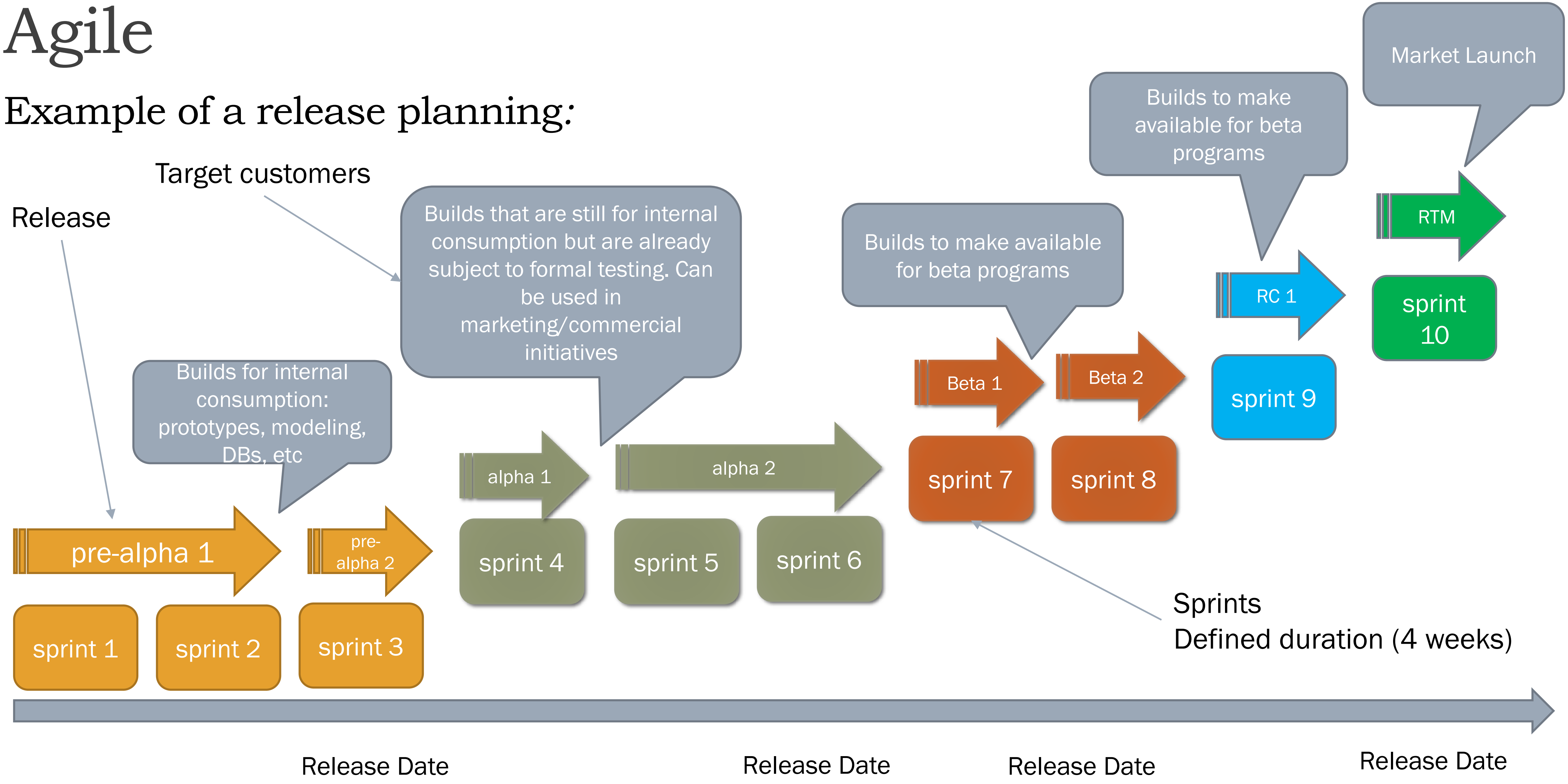
- **Feature-driven:** The product is released when it reaches a specific set of features.
- **Time-driven:** The approach is to launch in the market at regular intervals, and the scope is aligned accordingly.

It's also essential to define the target customers for each release. For example, alphas may be for internal consumption only, betas for beta customer programs, etc.

This implies that a product release may require one or more sprints!

Agile

Example of a release planning:



Target customers

Release

Builds for internal consumption: prototypes, modeling, DBs, etc

Builds that are still for internal consumption but are already subject to formal testing. Can be used in marketing/commercial initiatives

Builds to make available for beta programs

Builds to make available for beta programs

Market Launch

pre-alpha 1

pre-alpha 2

alpha 1

alpha 2

Beta 1

Beta 2

RC 1

RTM

sprint 1

sprint 2

sprint 3

sprint 4

sprint 5

sprint 6

sprint 7

sprint 8

sprint 9

sprint 10

Sprints

Defined duration (4 weeks)

Release Date

Release Date

Release Date

Release Date

Agile

Relating release planning to Software Release Lifeycle. A software product follows a life cycle from its design, commissioning and maintenance to the end-of-life. The meaning of releases indicate the stage in which the product is

Release	Meaning
Pre-alpha	Very early stages of the product, with little or no testing process. Typically, these are merely technical versions of architecture/structure design
Alpha	They already have features and/or behavior that can be subject to formal testing. It is in the Alpha phases that the features are built. Typically the alpha versions end with the complete feature, that is, the later phases are stabilization or inclusion of detected improvements that bring high value to the product
Beta	The beta phases are for stabilization and testing the market in terms of usability. These are builds typically made available to customers in beta programs. (Microsoft adopts the term Community Technology Preview, CTP versions)
Release Candidate	Final stages of beta programs, where only changes to the code are introduced to fix serious anomalies. As the name implies, it is a candidate for the final release version.
Release to Manufacturing (RTM)	It is the final version of the market release. Depending on the industry, it can adopt terms such as GOLD, STABLE, GENERAL AVAILABILITY.
Service Release (SRx)	Maintenance Builds that incorporate bug fixing of anomalies already detected in the market. They may also include evolutionary improvements (e.g. legal requirements that have come out in the meantime). Also known as Service Packs.