“Agile” is **not prescriptive** which means that it’s not by itself defying exactly HOW to obtain agility. Through the last 40 years there were many approaches (with better and worse results) to creating a concrete guidebook or **framework** that will tell exactly how to obtain agility in organization. Some of those frameworks were formalized even before Agile Manifesto (2001) but in their most updated versions they are in line with manifest’s message.

Scrum and it’s foundations (Transparency, Inspection & Adaptation)

Scrum as a complete framework for software development was first described and formalized by two American practitioners of software engineering Jeff Sutherland & Ken Schwaber in 1995. (Ken Schwaber Was one of the engineers that originally created “Agile Manifesto”).

Most recent update of Scrum Guide (a document describing Scrum) took place at the end of 2020 and changed the concept of Scrum as a framework that may be used **not only** in software development. You can find all information about Scrum (Scrum Guide, training, certifications, knowledge base etc.) on www.scrum.org.

Scrum:

- founded on Empiricism to asserts that knowledge comes from experience and making decisions based on what is observed.
- founded on Lean thinking to reduce waste and focuses on the essentials.
- employs an iterative, incremental approach to optimize predictability and to control risk;
- engages groups of people who collectively have **all the skills and expertise** to do the work and share or acquire such skills as needed.
- combines formal ***Scrum Events*** for inspection and adaptation within a containing event, the **Sprint.**
- These events work because they implement the empirical Scrum pillars of **transparency, inspection, and adaptation**

Transparency:

- process and work must be visible to those performing the work as well as those receiving the work;

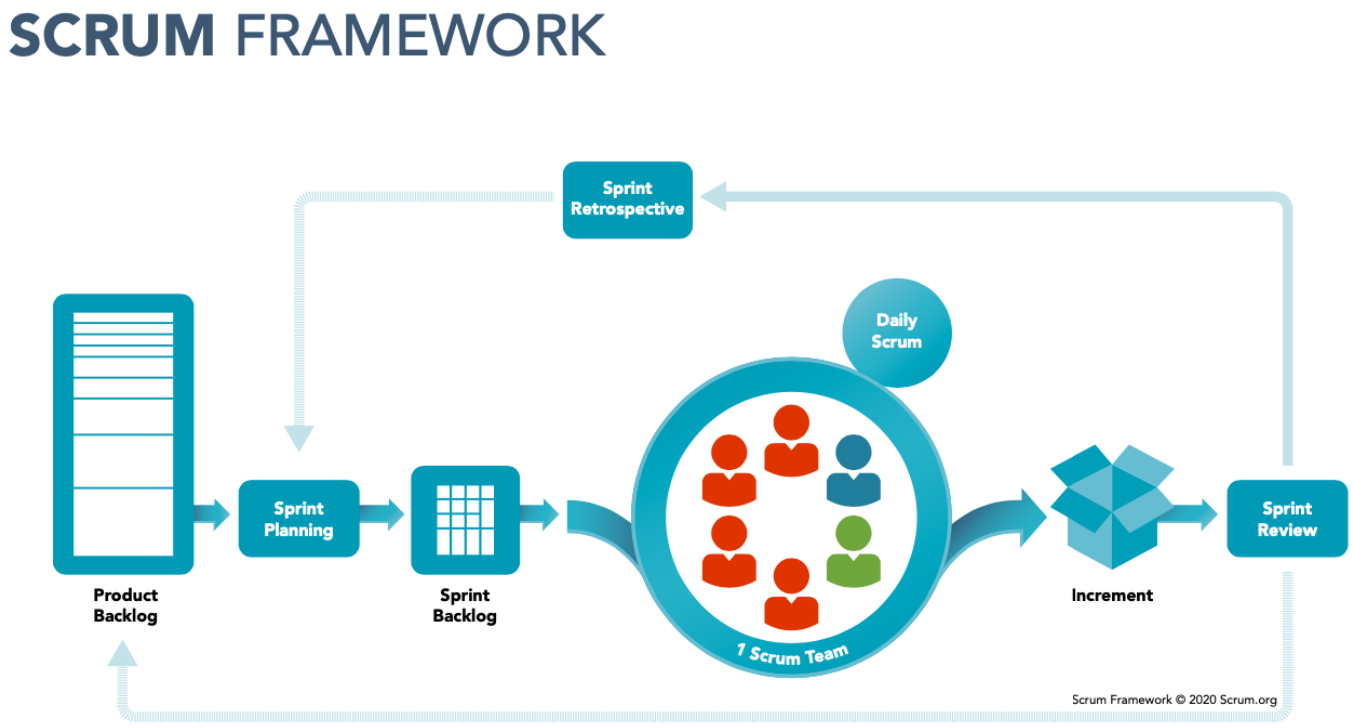
Inspection:

- ***Scrum artifacts*** and the progress toward agreed goals must be inspected frequently and diligently to detect problems early.
- inspection without transparency is misleading and wasteful.
- inspection without adaptation is pointless (Scrum has to provoke change);

Adaptation:

- Anty adjustment must be made as soon as possible to minimize further deviation.
- Adaptation becomes more difficult when the people involved are not empowered or self-managing.
- A ***Scrum Team*** is expected to adapt the moment it learns anything new through inspection;

Scrum Framework, Events, Artifacts, Scrum Team & roles Responsibilities



Scrum Team = **Product Owner** + **Scrum Master** + **Developers**

In a nutshell, Scrum requires a **Scrum Master** to foster an environment where:

1. A **Product Owner** orders the work for a complex problem into a **Product Backlog**
2. The self organizing **Scrum Team** turns a selection of the work into an **Increment** of value during a **Sprint**.
3. The Scrum Team and its **stakeholders** inspect the results and adjust for the next Sprint.
4. Repeat

As you see it's not a very complicated process but for scrum to operate properly there have to be proper definitions (and **understanding of those definitions**) of a) Scrum Events, b) Scrum Artifacts and c) Scrum Team Roles & Responsibilities , d) Stakeholders Responsibilities

a) Scrum Events:

- create regularity and minimize the need for meetings not defined in Scrum;
- are time-boxed to not allow waste of time;
- **It's totally OK** to organize other, non-scrum events/meetings as long as they serve their purpose and are not a waste of time;

Sprint – (Scrum Team):
Scrum Event that is time-boxed to one month or less, that serves as a container for the other Scrum events and activities. Sprints are done consecutively, without intermediate gaps.

Sprint Planning – (Scrum Team):
Scrum Event that is time-boxed for a maximum 4-8 hours (or less). During it Scrum Team (product owner+developers+scrum master) inspect the work from the Product Backlog that's most valuable to be done next and design that work into Sprint backlog.

Daily Scrum – (Developers):
15-minute time-boxed event held each day. During it developers are planning work for the next 24 hours and are inspecting what was done in the last 24 hours to better predict if Sprint Goal is achievable. The Daily Scrum is held at the same time

and place each day to reduce complexity

Sprint Review – (Scrum Team + Stakeholders):
Maximum 4 hour event to conclude the development work of a Sprint. During it Stakeholders inspect the Increment of product resulting from the Sprint and assess the impact of the work performed on overall progress and update the Product backlog in order to maximize the value of the next period/sprint.

Sprint Retrospective – (Scrum Team):
Maximum 3 hour event to end a Sprint. During it Scrum Team inspect the past Sprint and plan for improvements to be enacted during the next Sprint.

b) Scrum Artifacts:

- represent work or value to provide transparency and opportunities for inspection and adaptation;
- are specifically designed to maximize transparency of key information so that everybody has the same understanding of artifact's state;
- every artifact is owned by somebody who is responsible for it:

Product Backlog – (owned by Product Owner):

- Consists an ordered list of the work to be done in order to create, maintain and sustain a product. Managed by the Product Owner who is responsible for clearly stating Product Goal.

Sprint Backlog – (owned by Developers):

- Scrum Artifact that provides an overview of the development work to realize a Sprint's goal, typically a forecast of functionality and the work needed to deliver that functionality. Managed by the Developers.

Increment – (owned by Scrum Team):

- Defines the complete and valuable work produced by the Development Team during one Sprint. Work cannot be considered part of an Increment unless it meets the Definition of Done. The sum of all Increments form a product

c) Scrum Team Roles & Responsibilities

- define who is responsible for what and why;
- within a Scrum Team, there are no sub-teams or hierarchies. It is a cohesive unit of professionals focused on one objective at a time, the Product Goal.

Developers:

- Developers are the people in the Scrum Team that are committed to creating any aspect of a usable Increment each Sprint;
- specific skills needed by the Developers are often broad and will vary with the domain of work
- Create a plan for the Sprint, the Sprint Backlog;
- apply **Definition of Done**;
- Adapting their plan each day toward the Sprint Goal
- hold each other accountable as professionals.

Product Owner

- Entire organization (including Stakeholders) must respect
- Product Owner decisions!
- The Product Owner is one person, not a committee.
- The Product Owner may represent the needs of many stakeholders in the Product Backlog.
- Those wanting to change the Product Backlog can do so by trying to convince the Product Owner.

Product Owner responsibilities

- accountable for maximizing the value of a product by managing and expressing business and functional expectations for a product to the Developers;
- Developing and explicitly communicating the Product Goal;
- Creating and clearly communicating Product Backlog items;
- Ordering Product Backlog items;
- Ensuring that the Product Backlog is transparent, visible and understood.
- Product Owner may do the above work or may delegate the responsibility to

others. Regardless, the Product Owner remains accountable.

Scrum Master:

- accountable for guiding, coaching, teaching and assisting a Scrum Team, Key Stakeholders and its environments in a proper understanding and use of Scrum.
- creating Definition Of Done and maintaining it.

d) Stakeholder Role & Responsibilities

- a person external to the Scrum Team with a specific interest in and knowledge of a product that is required for it's incremental development (it will be mostly Product Sponsor/Inventor but may be also customers in B2C model);
- Stakeholders have to be represented by product Owner (or they have to delegate his responsibilities to someone);
- They must be actively engaged with the Scrum Team at Sprint Review;
- Stakeholders must respect **Product Owner's** decisions!

Definition of Done (DoD)

- formal description of quality measures that are required for product backlog item to become and Increment
- If a Product Backlog item does not meet the DoD, it cannot be released or even presented at the Sprint Review. Instead, it returns to the Product Backlog for future consideration.

Scrum Values

Is it harder for Scrum Master to implement a certain framework of "how we work" or to change the whole company culture ? Company culture it's something that is defined mostly at the very beginning of its existence thus the hardest part of a Scrum Master job during implementation of scrum is to "foster an environment". This means to lay strong foundations in company culture for Scrum to even occur. Those foundations are: Courage, Focus, Commitment, Respect & Openness.



Summary: Pros, cons of Scrum

Scrum provides a clear **iteration** structure for:

- developing and sustaining complex products (for example software development);
- smooth delivery of value to end customer;
- creating a feedback loop from end customer to production team
- adapting to changes in product requirements

Pros of Scrum:

- Scrum specifies precisely how to set an **iteration structure** for product development. If company does not have any structure for product development Scrum it’s most probably the easiest way of introduction to Agility (if Agile is a valuable concept for development of such a product);
- Scrum gives a chance to adjust if **change of product requirement** Existence of time-boxed Sprint restricts those most recent changes to be agreed only during every sprint planning (we should avoid to change requirements during ongoing sprint!). This helps a lot if we are working with a Client who is **constantly shifting** his approach to every piece of planned development thus we are suffering from the cost of constant context switching & non-stop research. Convincing that Client to use Scrum and maintaining it it’s a good way to make your product better.

Cons of Scrum:

- Scrum specifies precisely how to set an **iteration structure** thus it’s tempting to just apply it and not think about consequences. It may seem easy to apply Scrum everywhere but in reality the most challenging part is to change behaviour of people that are using this framework. Focusing constantly **only on some aspects** of Scrum may result in **Cargo Cult.**
- Scrum gives a chance to adjust if **change of product requirement** But what if gathering feedback from client/project sponsor is made only during sprint review at the end of the 4-week sprint? It is highly probable that requirements have changed but developers **had no idea about it** and delivered increment that have some value but could have way more value! Adaptation to changes of requirements do not magically happen just by applying scrum or any other framework. If values like transparency and openness are not cultivated in organization adaptation to any change will be almost impossible.