

# Course of Digital Management Consulting

## **Scrum. Introductory Concepts**

These slides have been designed  
together with Carmine Paragano and Cristina Paternoster

**Prof. Corrado Cerruti**

### Scrum. Introductory Concepts

- Scrum framework
- Events: scrum sprint
- Artifacts: product backlog & more
- Roles: scrum team
- ... beyond Scrum

The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game  
Scrum.org - The Home of Scrum  
<https://scrumguides.org/>

Scrum.org - The Home of Scrum: Introductory videos  
<https://www.scrum.org/resources/introductory-video-series-scrum>

SAFe® 6.0 Scaled Agile Framework® for Lean Enterprises  
<https://scaledagileframework.com/safe/>

Foundations of Agile – Google  
<https://www.youtube.com/watch?v=km7n3DI5IWk>

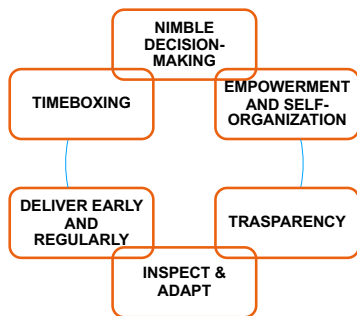
## Scrum

SCRUM is an **AGILE FRAMEWORK** for developing and sustaining **COMPLEX ADAPTIVE PRODUCTS** at the **HIGHEST VALUE**.

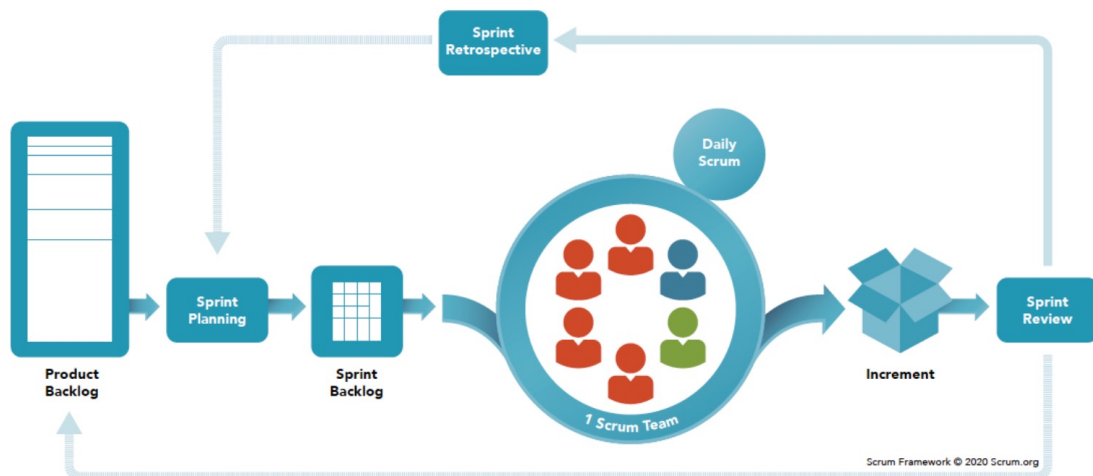
Scrum is more about project delivery rather than project management.

Scrum Framework should be used entirely in its **EVENTS**, **ARTIFACTS** and **ROLES**

In rugby a scrum is where the team comes together to move the ball forward  
In Agile Scrum is where the team comes together to move the product forward.



## EVENTS: Scrum sprint



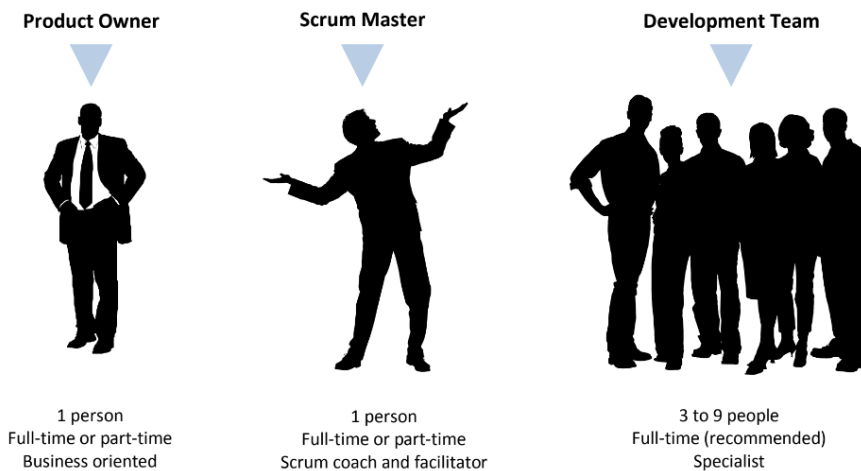
Sprints are fixed length periods of work that last one month or less.

They create consistency and ensure short iterations for feedback in order to inspect and adapt both how work is done and what is being worked on.

If cycles are longer, then the spirit of frequent feedback cycles can be lost. Longer Sprint may also get too complex and may increase risk.

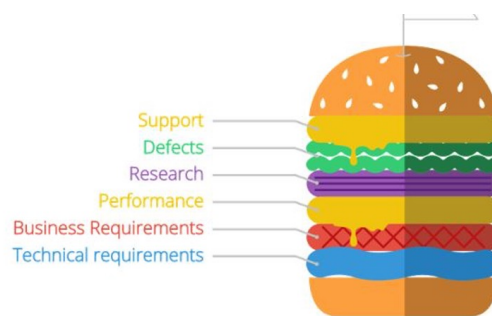
A new Sprint starts immediately after the conclusion of the previous Sprint.

## ROLES: Scrum team



The term “Scrum Team” refers to all the project team members: everyone internal to the project.

## ARTIFACTS: Backlogs and increments



Scrum’s artifacts represent work or value. They are designed to maximize transparency of key information. Thus, everyone inspecting them has the same basis for adaptation.

Each artifact contains a commitment to ensure it provides information that enhances 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 Done.

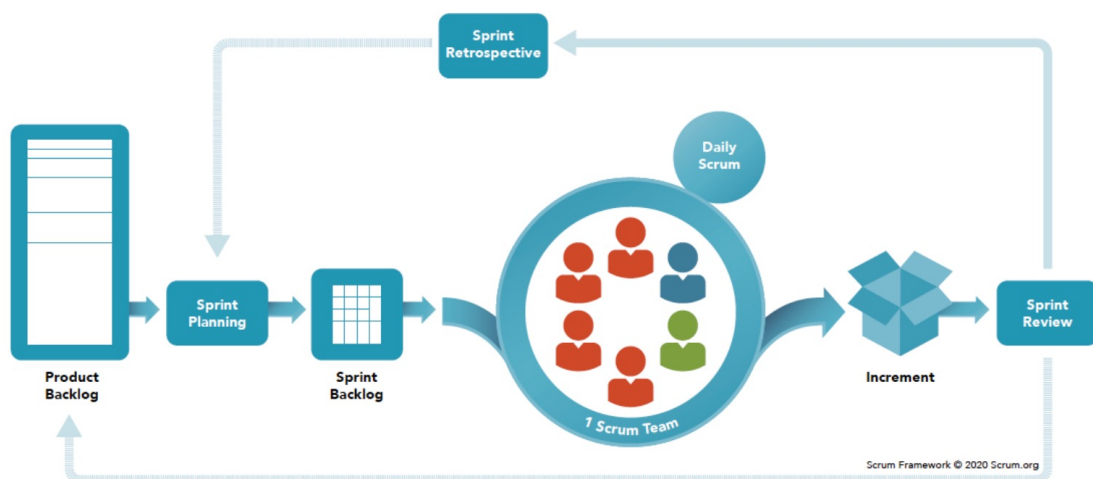
## Agile vs. Scrum

**AGILE** is a MINDSET  
a "way of thinking"  
a "way of seeing" things



SCRUM is one of the many methodologies that AGILE offers:  
adopting SCRUM does not therefore mean transforming a company  
into Agile

## EVENTS: Scrum sprint



Sprints are fixed length periods of work that last one month or less.  
They create consistency and ensure short iterations for feedback in order to inspect and adapt both how work is done and what is being worked on.  
If cycles are longer, then the spirit of frequent feedback cycles can be lost. Longer Sprint may also get too complex and may increase risk.  
A new Sprint starts immediately after the conclusion of the previous Sprint.

## Backlog preparation

The backlog is the ordered list of all the work, presented in the form of stories, for a team. There is no need to create all of the stories for the entire project before the work starts – only enough to understand the first release in broad brushstrokes and then sufficient items for the next iteration.

Product owner might produce a product roadmap to show the anticipated sequence of deliverables over time. The product owner replans the roadmap based on what the team produces,



## Sprint planning

Sprint Planning initiates the Sprint by laying out the work to be performed for the Sprint and addressing the following topics:

- **Why is this Sprint valuable?**

The Product Owner proposes how the product could increase its value and utility in the current Sprint. The whole Scrum Team then collaborates to define a Sprint goal that communicates its value to stakeholders.

- **What can be Done this Sprint?**

Through discussion with the Product Owner, the Developers select items from the Product Backlog to include in the current Sprint. Selecting how much can be completed within a Sprint may be challenging. However, the more the Developers know about their past performance, their upcoming capacity, and their Definition of Done, the more confident they will be in their Sprint forecasts.

- **How will the chosen work get done?**

For each selected Product Backlog item, the Developers plan the work necessary to create an Increment that meets the Definition of Done. This is often done by decomposing Product Backlog items into smaller work items of one day or less.

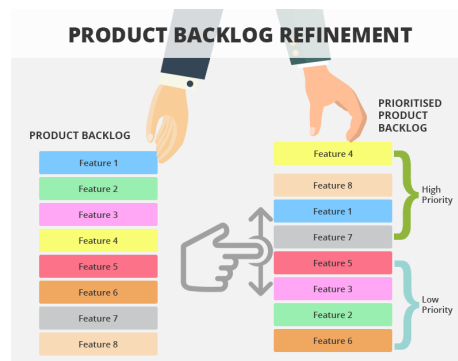
How this is done is at the sole discretion of the Developers. No one else tells them how to turn Product Backlog items into Increments of value.

## Backlog refinement

In iteration-based agile, the product owner often works with the team to prepare some stories for the upcoming iteration during one or more sessions in the middle of the iteration. The purpose of these meetings is to refine enough stories so the team understands what the stories are and how large the stories are in relation to each other.

Refinement meetings allow the product owner to present stories ideas to the team and for the team to learn about the potential challenges or problems in the stories. If the product owner is unsure of the dependencies, the product owner can request the team to spike the feature in order to understand the risks.

Teams often have a goal of spending not more than 1 hour per week refining stories for the next batch of work. Teams want to maximize the time spent doing work as opposed to planning work.



## Daily standups

Teams use standups to microcommit to each other, uncover problems, and ensure the work flows smoothly through the team.

Timebox the standup to no longer than 15 minutes. The team “walks” the Kanban or task board in some way, and anyone from the team can facilitate the standup.

One of the antipatterns typically seen in standups: they become status meetings. Teams who have traditionally worked in a predictive environment may tend to fall into this antipattern since they are used to providing a status.

The team begins to solve problems as they become apparent. Standups are for realizing there are problems – not for solving them.

## Demonstration/review

The team periodically demonstrates the working product. The product owner sees the demonstration and accepts or declines stories.

In iteration-based agile, the team demonstrates all completed work items at the end of the iteration. In flow-based agile, the team demonstrates completed work when it is time to do so, usually when enough features have accumulated into a set that is coherent. Teams, including the product owner, need feedback to decide how early to ask for product feedback.

A fundamental part of what makes a project agile is the frequent delivery of a working product. A team that does not demonstrate or release cannot learn fast enough and is likely not adopting agile techniques. The team may require additional coaching to enable frequent delivery.

## Retrospective

Retrospectives help the team learn from its previous work on the product and its process. One of the principles behind the Agile Manifesto is: “At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.”

Many teams use iterations – especially 2-weeks iterations – because the iteration prompts a demonstration and a retrospective at the end. However, the team does not need iterations in order to retrospect.

First and foremost, a retrospective is not about blame; the retrospective is a time for the team to learn from previous work and make small improvements. The retrospective is about looking at the qualitative (people’s feelings) and quantitative (measurements) data, then using that data to find root causes, designing countermeasures, and developing action plans, the project team may end up with many action items to remove impediments.

Consider limiting the number of action items to the team’s capacity to address improvement in the upcoming iteration or work period. Trying to improve too many things at once and not finishing any of them is much worse than planning to complete fewer items and successfully completing all of them.



## ROLES: Scrum Team

Product Owner



1 person  
Full-time or part-time  
Business oriented

Scrum Master



1 person  
Full-time or part-time  
Scrum coach and facilitator

Development Team



3 to 9 people  
Full-time (recommended)  
Specialist

The term “Scrum Team” refers to all the project team members: everyone internal to the project.

Scrum aims to ensure that the team will deliver timely and relevant outputs. so there should be no confusion about who does what..

## Agile/Scrum teams characteristics



Characteristic	Goal
Dedicated people	<ul style="list-style-type: none"> <li>Increased focus and productivity</li> <li>Small team, fewer than ten people</li> </ul>
Cross-functional team members	<ul style="list-style-type: none"> <li>Develop and deliver often</li> <li>Deliver finished value as an independent team</li> <li>Integrate all the work activities to deliver finished work</li> <li>Provide feedback from inside the team and from others, such as the product owner</li> </ul>
Co-location or ability to manage any location challenges	<ul style="list-style-type: none"> <li>Better communication</li> <li>Improved team dynamics</li> <li>Knowledge sharing</li> <li>Reduced cost of learning</li> <li>Able to commit to working with each other</li> </ul>
Mixed team of generalists and specialists	<ul style="list-style-type: none"> <li>Specialists provide dedicated expertise and generalists provide flexibility of who does what</li> <li>Team brings their specialist capabilities and often become generalizing specialists, with a focus specialty plus breadth of experience across multiple skills</li> </ul>
Stable work environment	<ul style="list-style-type: none"> <li>Depend on each other to deliver</li> <li>Agreed-upon approach to the work</li> <li>Simplified team cost calculations (run rate)</li> <li>Preservation and expansion of intellectual capital</li> </ul>



## Product Owner

Product Owner



1 person  
Full-time or part-time  
Business oriented

The Product Owner is accountable for maximizing the value of the product resulting from the work of the Scrum Team.

The Product Owner is accountable for effective Product Backlog management, which includes:

- Developing and explicitly communicating the Product Goal;
- Creating and clearly communicating Product Backlog items;
- Ordering Product Backlog items; and,
- Ensuring that the Product Backlog is transparent, visible and understood.

For Product Owners to succeed, the entire organization must respect their decisions. These decisions are visible in the content and ordering of the Product Backlog, and through the inspectable Increment at the Sprint Review.

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.

*The Product Owner is the bridge  
between the Business Stakeholders and the Development Team*

Source: The Scrum Guide - <https://scrumguides.org/>

## Development Team

Development Team



3 to 9 people  
Full-time (recommended)  
Specialist

Development team is self-organizing and cross-functional

- Self-organizing: choose how best to accomplish their work, rather than being directed by others
- Cross-functional: have all competencies needed to accomplish the work without depending on others not part of the team

Development team delivers products iteratively and incrementally, maximizing opportunities for feedback.

All team members should be available to the project full time.

Scrum recognizes no titles for Development Team members other than Developer. Scrum recognizes no sub-teams in the Development Team, regardless of particular domains that need to be addressed like testing or business analysis.

Individual Development Team members may have specialized skills and areas of focus, but accountability belongs to the Development Team as a whole.

## Scrum Master



The Scrum Master is accountable for the Scrum Team's effectiveness.

The Scrum master aims to enable the Scrum Team to improve its practices, within the Scrum framework.

The Scrum Master serves the Scrum Team in several ways, including:

- Coaching the team members in self-management and cross-functionality;
- Helping the Scrum Team focus on high-value Increments that meet the Definition of Done;
- Causing the removal of impediments to the Scrum Team's progress; and,
- Ensuring that all Scrum events take place and are positive, productive and timeboxed.

The Scrum Master serves the Product Owner in several ways, including:

- Helping find techniques for effective Product Goal definition and Product Backlog mgmt;
- Helping the Scrum Team understand the need for clear & concise Product Backlog items;
- Helping establish empirical product planning for a complex environment;
- Facilitating stakeholder collaboration as requested or needed.

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

- Leading, training, and coaching the organization in its Scrum adoption;
- Planning and advising Scrum implementations within the organization;
- Removing barriers between stakeholders and Scrum Teams.

Source: The Scrum Guide - <https://scrumguides.org/>

## Servant Leadership empowers the team

Agile approaches emphasize servant leadership as a way to empower teams. Servant leadership is the practice of leading through service to the team, by focusing on understanding and addressing the needs and development of team members in order to enable the highest possible team performance.

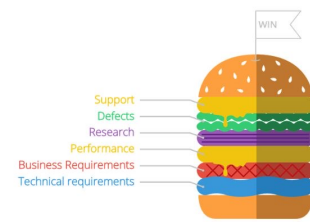
Servant leaders approach project work looking at:

- Purpose. Work with the team to define the “why” or purpose so they can engage around the goal for the project, optimizing at the project level, not the person level
- People. Once the purpose is established, encourage the team to create an environment where everyone can succeed. Ask each team member to contribute.
- Process. Do not plan on following the “perfect” agile process, but instead look for the results. When a cross-functional team delivers finished value often and reflects on the product and process, the teams are agile.

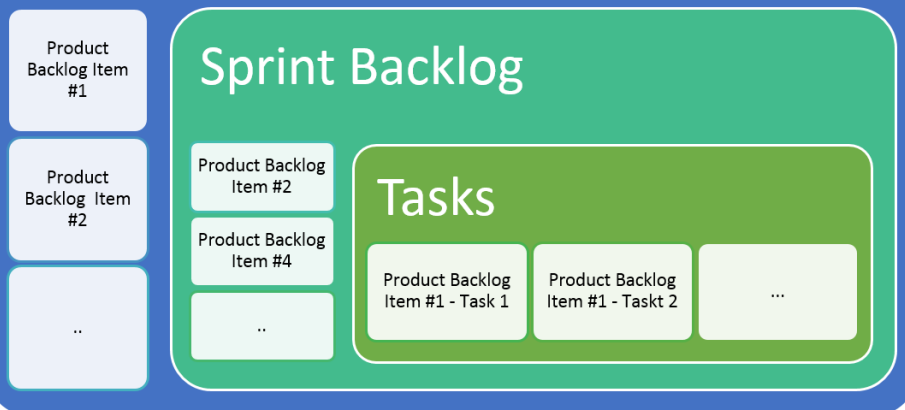
The following characteristics of servant leadership facilitate the agile team's success:

- *Promoting self-awareness*
- *Listening*
- *Serving those on the team*
- *Helping people grow*
- *Coaching vs. controlling*
- *Promoting safety, respect, and trust*
- *Promoting the energy and intelligence of others*

## Product backlog structure



## Product Backlog



## Product backlog

The Product Backlog is an emergent, ordered list of what is needed to improve the product. It is the single source of work undertaken by the Scrum Team.

Product Backlog items that can be Done by the Scrum Team within one Sprint are deemed ready for selection in a Sprint Planning event. They usually acquire this degree of transparency after refining 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 often vary with the domain of work.

The Developers who will be doing the work are responsible for the sizing. The Product Owner may influence the Developers by helping them understand and select trade-offs.

Product Backlog is the single source of all desired work on project.

The Product Owner is accountable for the content, availability (make it public), and prioritization.

Product Backlog is reprioritized at start of each sprint.

Product Backlog is never complete: is a dynamic artifact.

## Product Backlog content

Product Backlog includes:

- User Stories, Epics, Features, etc.
- Solution and Transition Requirements
- “Spikes” (help the team to gain technical understanding)
- Known bugs
- Other work items

## User story

A user story is the smallest unit of work in an agile framework. It's an end goal, not a feature, expressed from the software user's perspective. A user story is an informal, general explanation of a software feature written from the perspective of the end user or customer.

A user story represents a small, concise statement of functionality needed to deliver value to a specific stakeholder,

Example: as a Security Officer, I need to only allow authorized users to access the xyz functionality so I can ensure we enforce *abc* security directive.

## User story

<input type="radio"/>	Story ID:	Story Title:
<b>User Story:</b>		<b>Importance:</b>
As a: <role>		<input type="text"/>
I want: <some goal>		<b>Estimate:</b>
So that: <some reason>		<input type="text"/>
<b>Acceptance Criteria</b>		<b>Type:</b>
And I know I am done when:		<input type="checkbox"/> Search
		<input type="checkbox"/> Workflow
		<input type="checkbox"/> Manage Data
		<input type="checkbox"/> Payment
		<input type="checkbox"/> Report/ View

- As a <type of user> - **WHO**
- I want to <feature> - **WHAT**
- So that <benefit / purpose> - **WHY**

## Spike

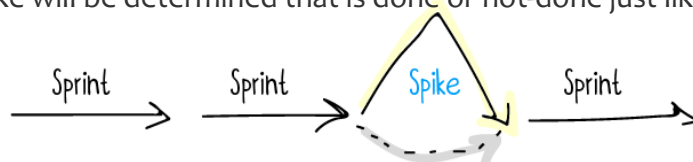
Spikes are a special type of user story that is used to gain the knowledge necessary to reduce the risk of a technical approach, better understand a requirement, or increase the reliability of a story estimate.

The Development Team may not have knowledge of a new technology, and spikes may be used for basic research to ensure the feasibility of the new technology/domain/approach ... no new working code is produced!

The story may contain significant technical risk, and the team may have to do some experiments or prototypes to gain confidence in a technological approach that may allow them to commit the user story to some future timebox.

The Development Team might be unsure if they can complete the story due to some potential blockers and probably can't even estimate the story. Thus, you may consider a spike as an investment for a Product Owner to figure out what needs to be built and how the team is going to build it.

A spike has a maximum time-box as the Sprint it is contained in it. At the end of a sprint, the spike will be determined that is done or not-done just like any user story.



## Increment and definition of Done

An Increment is a concrete stepping stone toward the Product Goal. Each Increment is additive to all prior Increments and thoroughly verified, ensuring that all Increments work together. In order to provide value, the Increment must be usable.

Multiple Increments may be created within a Sprint. The sum of the Increments is presented at the Sprint Review thus supporting empiricism. However, an Increment may be delivered to stakeholders prior to the end of the Sprint. The Sprint Review should never be considered a gate to releasing value.

Work cannot be considered part of an Increment unless it meets the Definition of Done, which is a formal description of the state of the Increment when it meets the quality measures required for the product. The moment a Product Backlog item meets the Definition of Done, an Increment is born.

The Definition of Done creates transparency by providing everyone a shared understanding of what work was completed as part of the Increment. If a Product Backlog item does not meet the Definition of Done, it cannot be released or even presented at the Sprint Review. Instead, it returns to the Product Backlog for future consideration.

## Agile approaches

