



Does it mean
better if it's agile?



Anna Kuśmierz
Senior Consultant



Piotr Pawlak
IT Specialist

Agile introduction

Scrum Team

Scrum events & artifacts

User Stories

People & Teams

Agile project management methodologies are suitable for complex and unpredictable projects.

When should we use Agile:

1. At **complex** projects and problems
2. For **empirical** and **unpredictable** projects

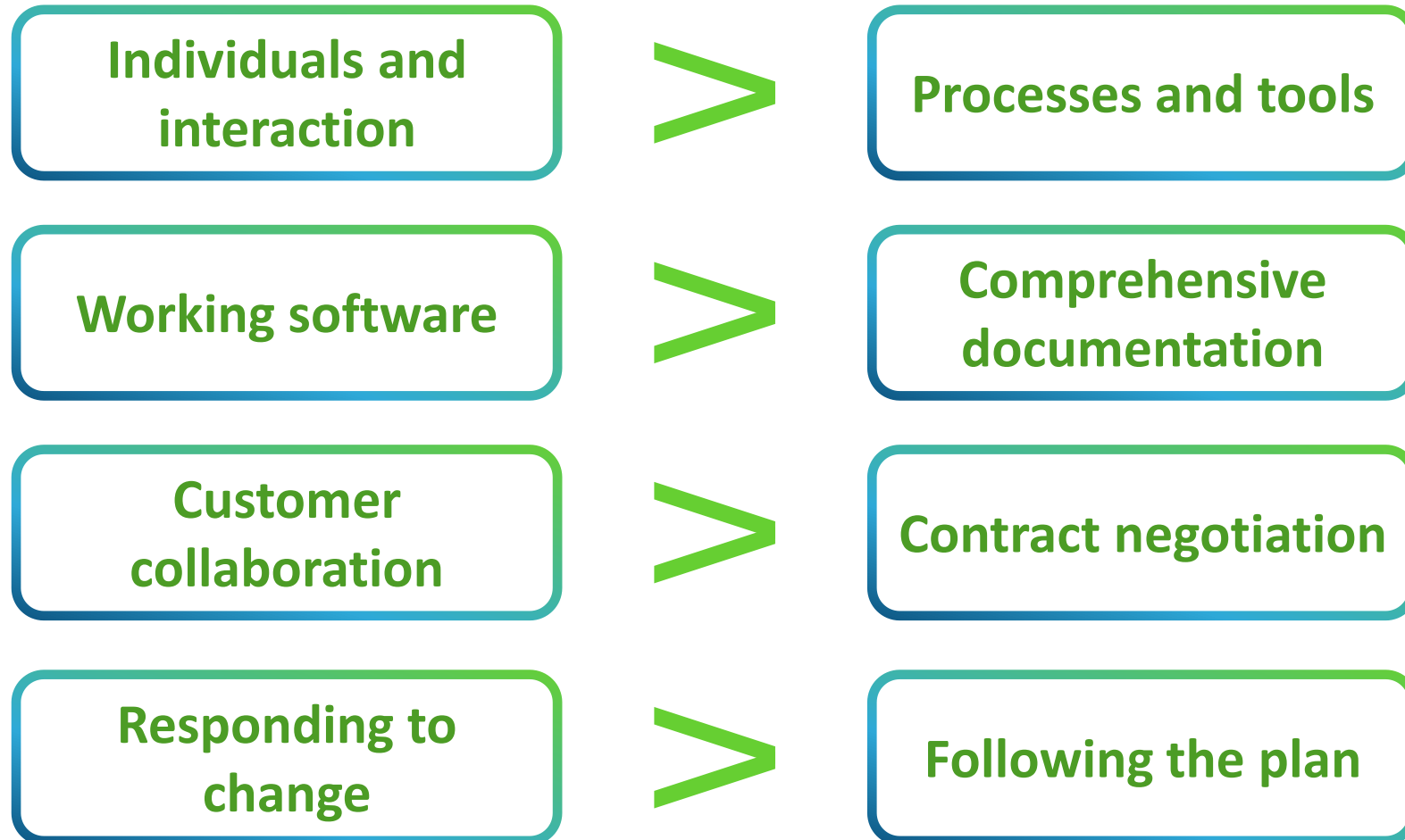


When are other methodologies more suitable:

1. In **easy** and **straightforward** projects and problems
2. For **predictable** projects



Publication of Agile Manifesto in 2001 is the beginning of agile methods' usage for solutions' delivery.



**Individuals and
interaction**



Processes and tools

cooperate to find the best value-adding process

work with users to understand their needs

communicate face to face



Working software



Comprehensive
documentation



enhance business value through **technical excellence**

restrict documentation - create only the one that adds value to the team

gauge the progress using **operational software** as the measure

**Customer
collaboration**



Contract negotiation

A decorative graphic on the left side of the slide consists of a thick curved line that transitions from green at the top to blue at the bottom. Three white circles with blue outlines are spaced along this curve.

involve customer actively in the project

establish a **joint team** with the client

request **users' feedback** regularly based on the working software

**Responding to
change**



Following a plan

adapt to changes

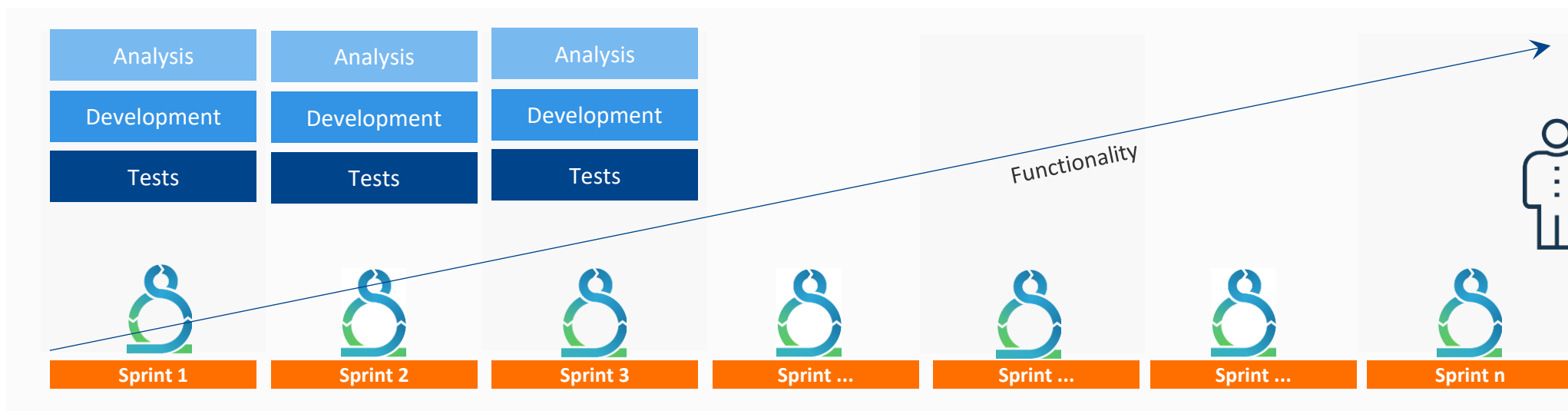
manage priorities and constantly **adjust backlog**

improve the way you work to increase effectiveness

Waterfall



Agile



- Easier change management
- Project's transparency
- Increased productivity
- Faster business value delivery

	Agile	Waterfall
Driver	Value	Plan
Flexibility	Very high	Low
Cooperation with the client	Very frequent	Moderate
Team sense of responsibility	High	Low
Problem identification	Very early	Late
Complexity level	Moderate	High
Business feedback	Early	Late
Risk management	Early	Late

You see what is happening

You inspect progress towards Sprint Goal

You adjust the way you work

TRANSPARENCY

INSPECTION

ADAPTATION

Agile introduction

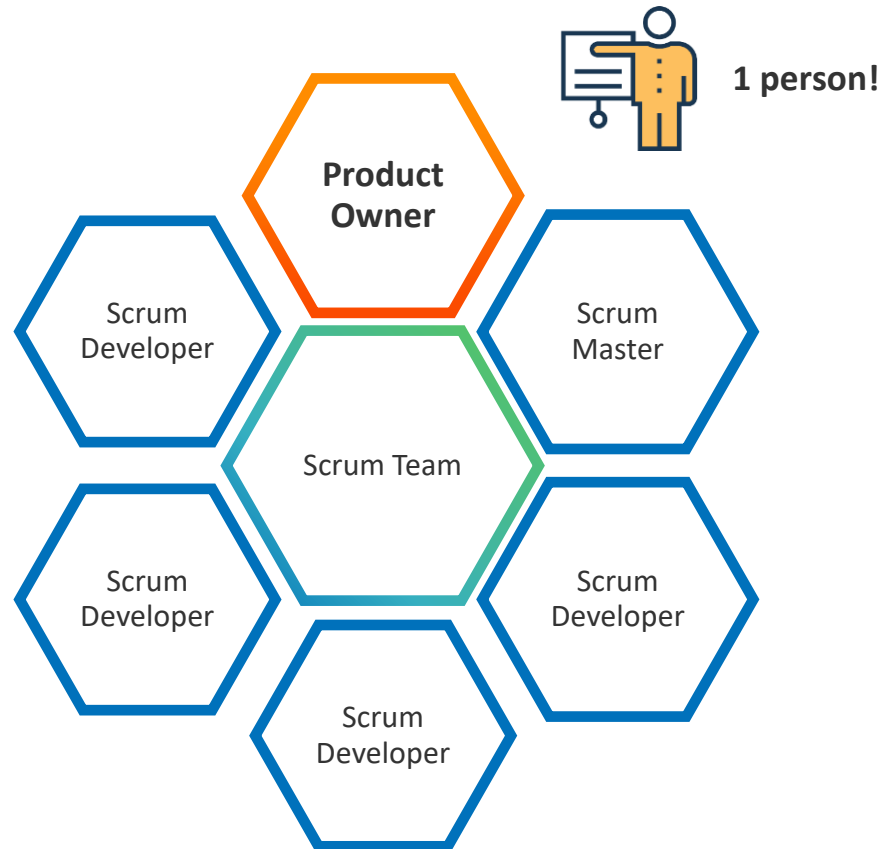
Scrum Team

Scrum events & artifacts

User Stories

People & Teams

Product Owner is responsible for maximising the value of the product.



Product Owner's responsibilities

- Managing the Product Backlog
- Ordering the items in the Product Backlog
- Cooperation with stakeholders
- Monitoring and acceptance of new functionalities delivered by the team
- Ensuring that the Product Backlog is visible, transparent and clear to all
- Ensuring that Dev Team understands items in the Product Backlog

Determined



**Decisional /
empowered**



**Experienced /
expert in the
field**



„Omniscient *”



*Has:

- "Big picture" in his head
- Can see dependencies and is synchronised with others

A Development Team consists of 3-9 professionals who do the work of delivering a potentially releasable Increment.



Key attributes of a successful team

- Self-organising
- Cross-functional
- Consists only of Scrum Developers
- Accountability always belongs to Dev Team as a whole

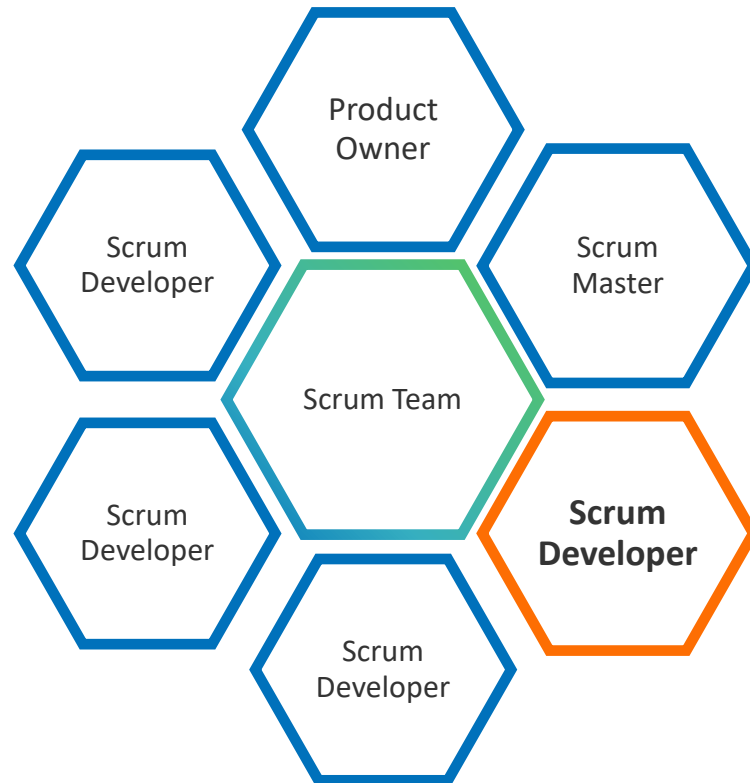
Why such team?

Experience shows that:

- Too small teams create bottleneck risk
- Too big teams are unmanageable
- Smaller team creates better relations
- In a big team „lazybones“ can hide

**Self-organising means that no one (not even Scrum Master) tells the Dev team how to turn Product Backlog into Increment.
Cross-functional means that all skills necessary to create a product Increment are within Dev Team members.**

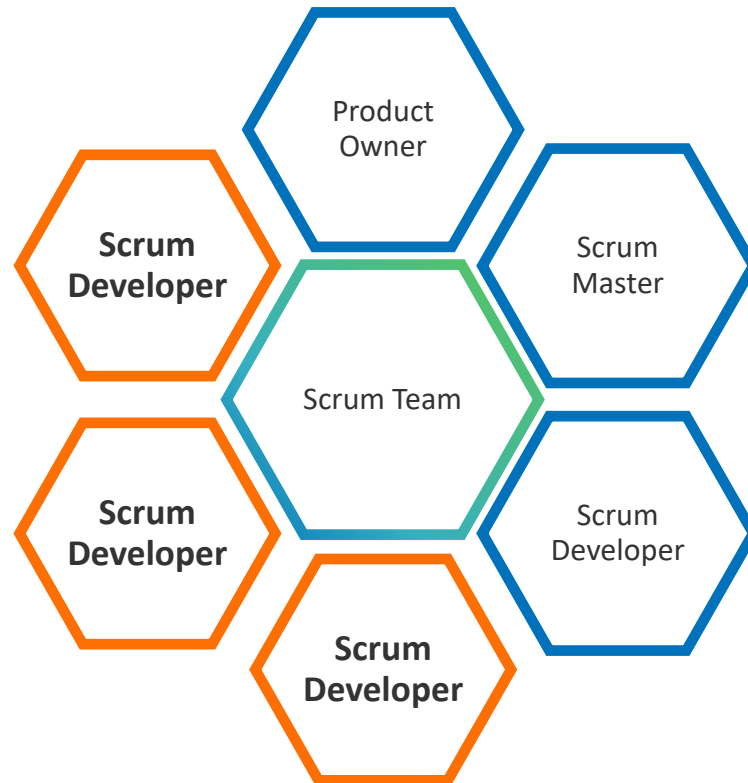
Business Analyst and Tester role can be fulfilled by any person from Development Team.



Analyst's responsibilities

- Gathering business requirements
- Cooperating with developers during the design of solution
- Verifying solutions in terms of business needs
- Testing
- Preparing documentations

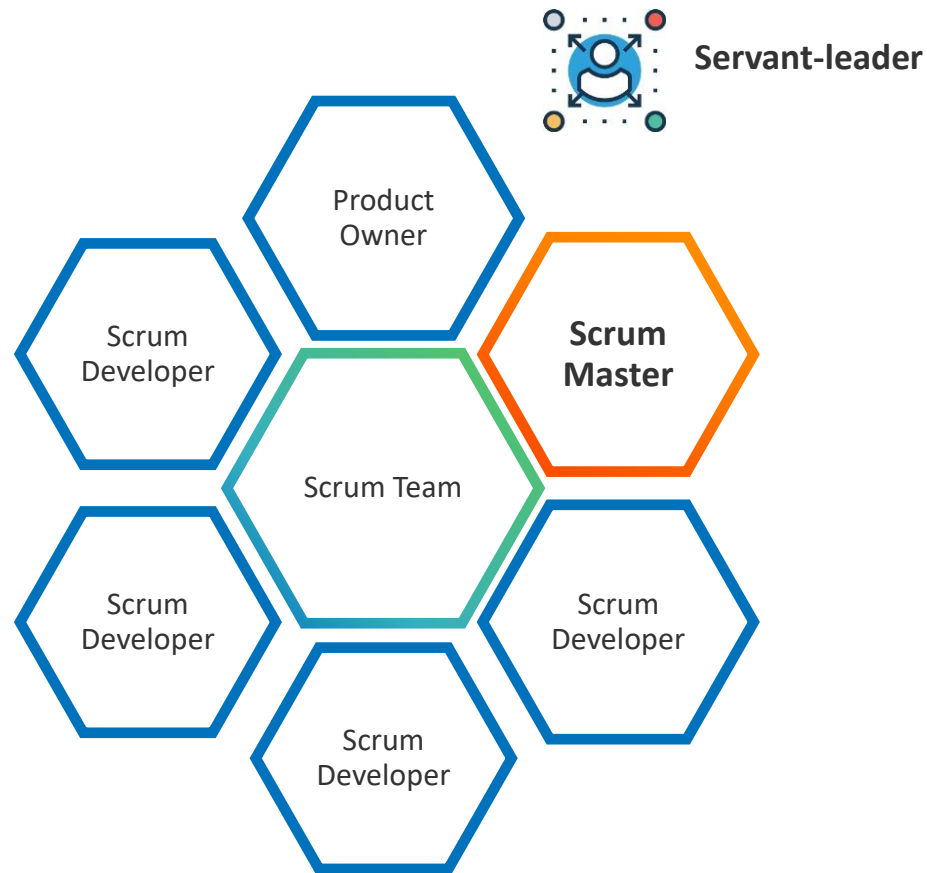
Developer role can be fulfilled by any person from Development Team.



Developer's responsibilities

- Designing a technical solution in accordance with business requirements
- Sustaining high quality of code
- Reviewing code committed by other team members

Scrum Master is responsible for promoting and supporting Scrum by helping everyone understand Scrum theory, practices, rules and values.



Scrum Master's responsibilities

- Managing the Scrum Process
- Helping the team with self-organisation
- Resolving team's obstacles
- Aiding communication between the team and Product Owner or organisation
- Facilitating Scrum Meetings
- Assisting Product Owner with Backlog maintenance

Experts support the scrum teams.

Experts support the team with their expertise

- Support user stories analysis
- Answer team members questions
- Are not assigned to a single team but support all the teams accordingly
- Are available for the team



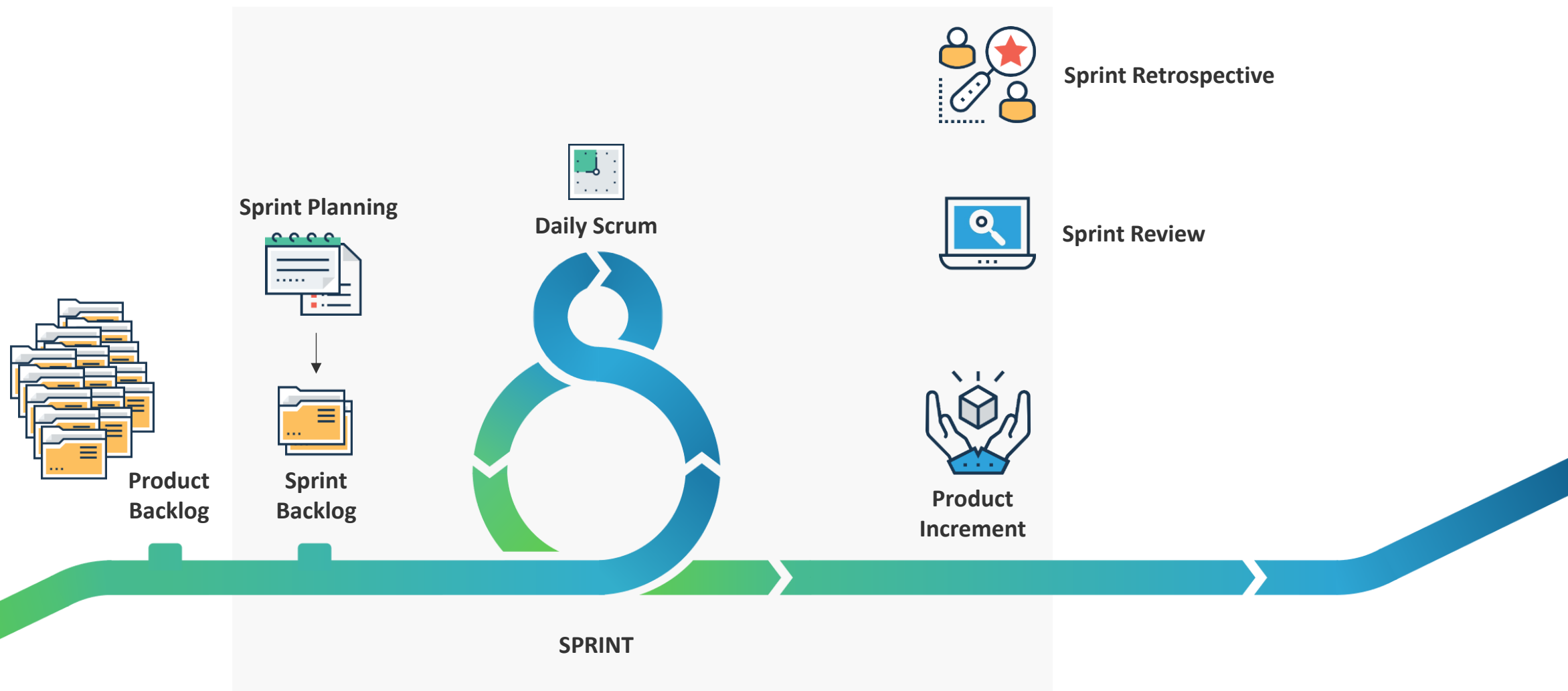
Agile introduction

Scrum Team

Scrum events & artifacts

User Stories

People & Teams



Product Backlog – the single source of all known requirements needed in the final product.


- **Product Backlog is owned by Product Owner**
- Presents complete list of features, functions, requirements, enhancements and fixes that define product
- Product items have the following attributes: description, order, estimate and value
- It is constantly updated and prioritised by the Product Owner
- Let's the team focus on priorities
- Higher backlog items deliver more value and are better described than lower items
- Single source of work for Dev Team



The Product Backlog is dynamic and constantly changes to identify what the product needs.

Sprint Backlog - set of Product Backlog items selected for the Sprint, plus a plan for delivering the product Increment and achieving the Sprint Goal.

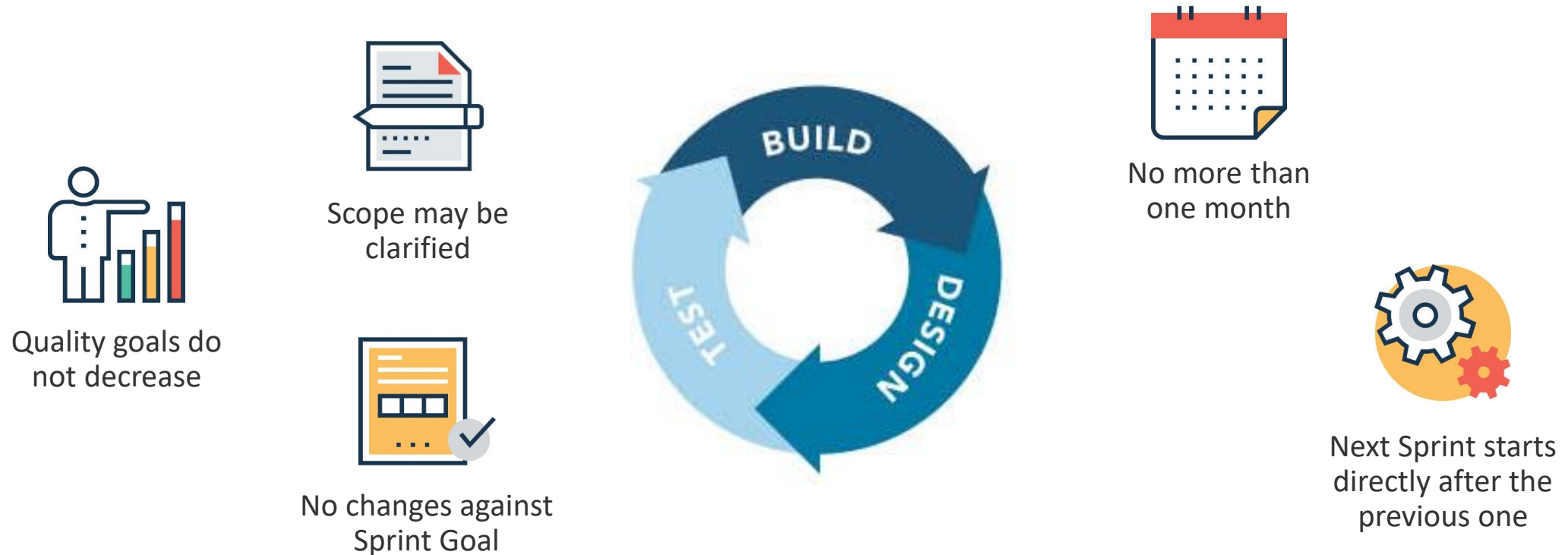
- **Scrum Team defines Sprint Goal**
- Sprint Backlog makes all the work that the Dev Team identifies as necessary to meet the Sprint Goal visible
- Forecast by the Development Team about what functionality will be in the next Increment and the work needed to deliver that functionality
- The Dev Team modifies Sprint Backlog throughout the Sprint whenever it learns more about work needed to achieve Sprint Goal
- The Dev Team selects the User Stories considering the resources availability



Sprint
Backlog

Development Team owns Sprint Backlog and only Dev Team can adjust and modify Sprint Backlog.

Sprint is the heart of Scrum – a time-box during which a "Done", a usable and potentially releasable product Increment, is created.



The result of Sprint is Increment – sum of all Product Backlog items completed during Sprint plus increments from all previous Sprints. Increment must be "Done" – usable condition, ready to inspect and potentially releasable.

The aim of that meeting is to plan how to reach the Sprint's Goal.

INPUT:

- Product Backlog
- Last Product Increment
- Dev Team capacity
- Past performance of Dev Team



Choice of tasks
from the Product
Backlog for the
next Sprint



Giving User
Stories' details



Division of User
Stories into tasks



Estimation of
work effort of
particular tasks



Product Owner,
Scrum Master, Dev
Team



8h for a month-
long sprint

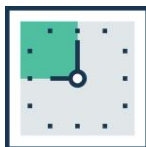
OUTPUT:

- Sprint Goal
- Sprint Backlog

Daily Scrum – the main goal of the meeting is to inspect the progress towards meeting the Sprint Goal and to adapt if needed to the changing situation.



1. **What I did yesterday**, that helped the team to achieve goals?
2. **What I will do today**, that will help the team to achieve goals?
3. **If I see any problems**, that might block me or the team to achieve sprint goals?



15 min. long



Focus on the three questions



Development Team and Scrum Master



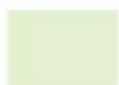
Same place



Same time



Facilitated by Scrum Master



User Story



Defect

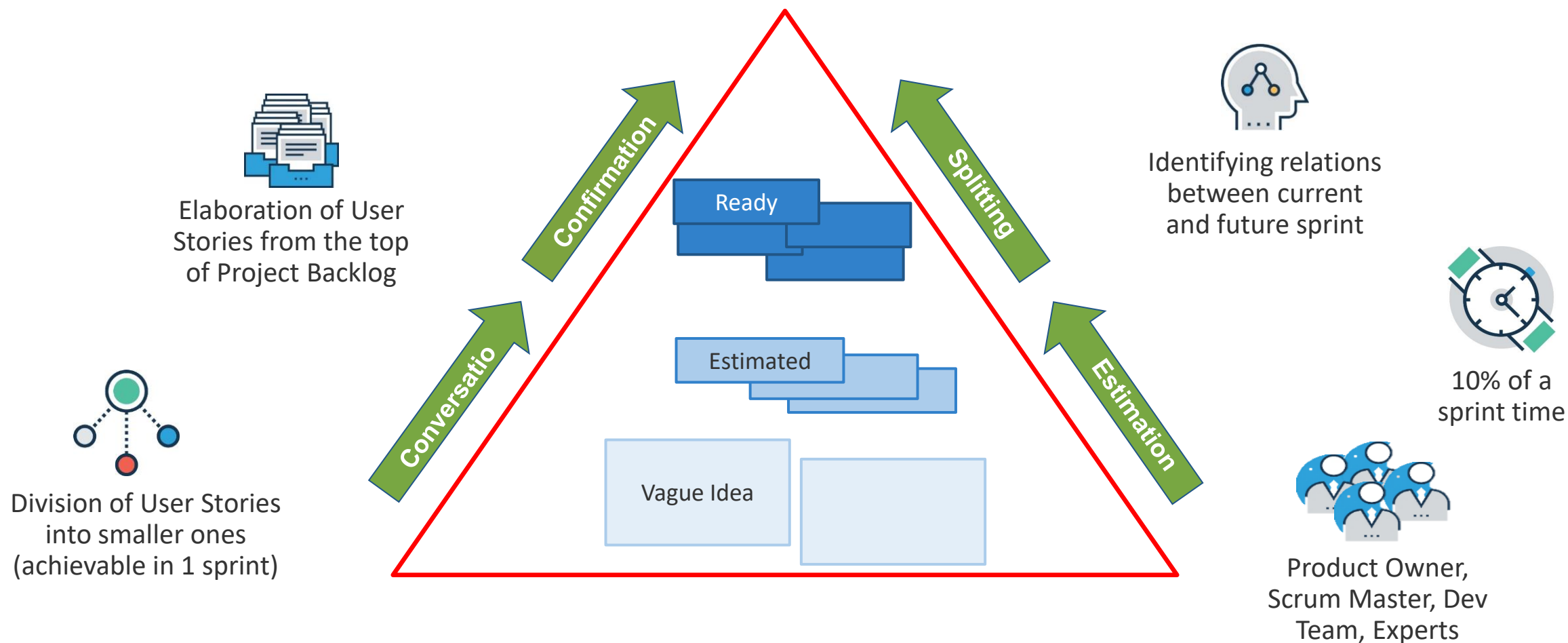


Task



Feature

The aim of the Backlog Refinement is to add details, estimates and order to Product Backlog items.



Definition of Ready – defines which criteria has to be met to decide if a task is ready to be taken on Sprint Planning meeting.




Same criteria for all of the stories

- ① Only clearly defined stories, where all tasks are pointed out
- ② Enables the team to directly start working on the story
- ③ Ensures a common understanding between the team and the Product Owner
- ④ Responsibility for the PO and SMEs



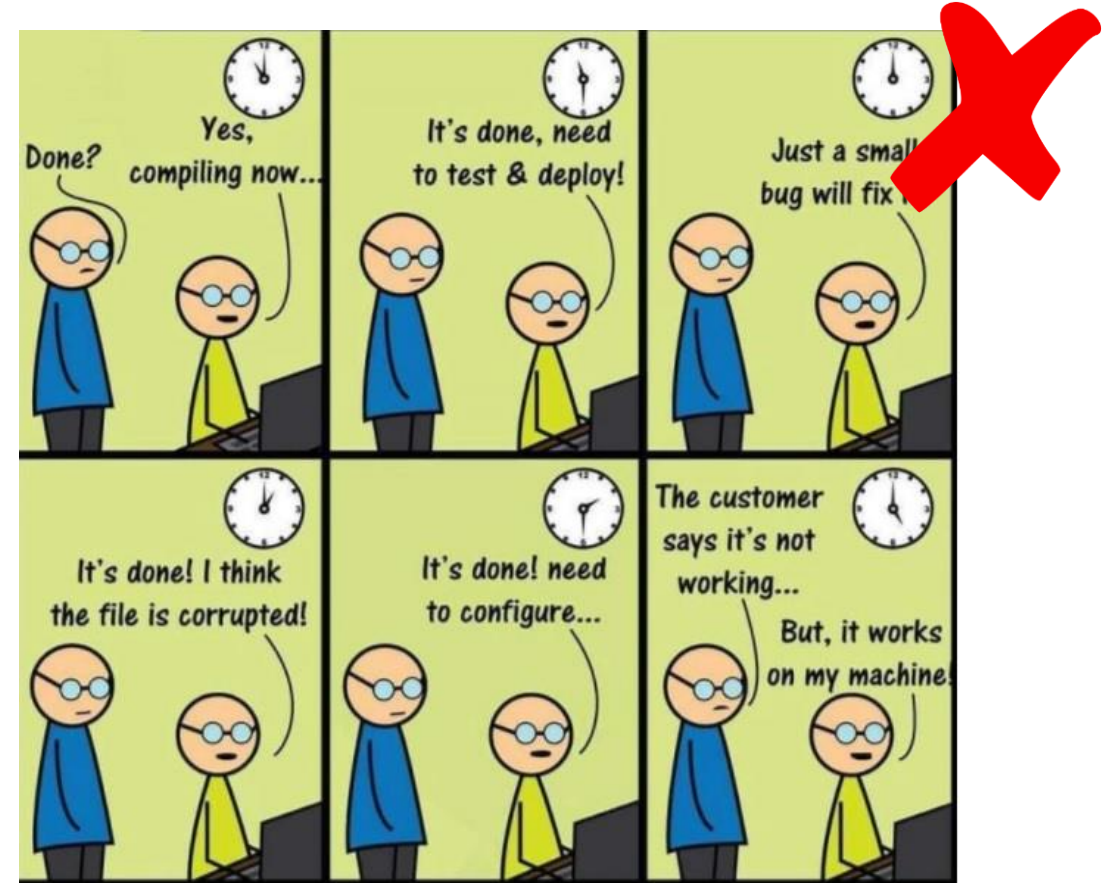
Definition of Done – defines what criteria has to be met to decide if a task is completed.



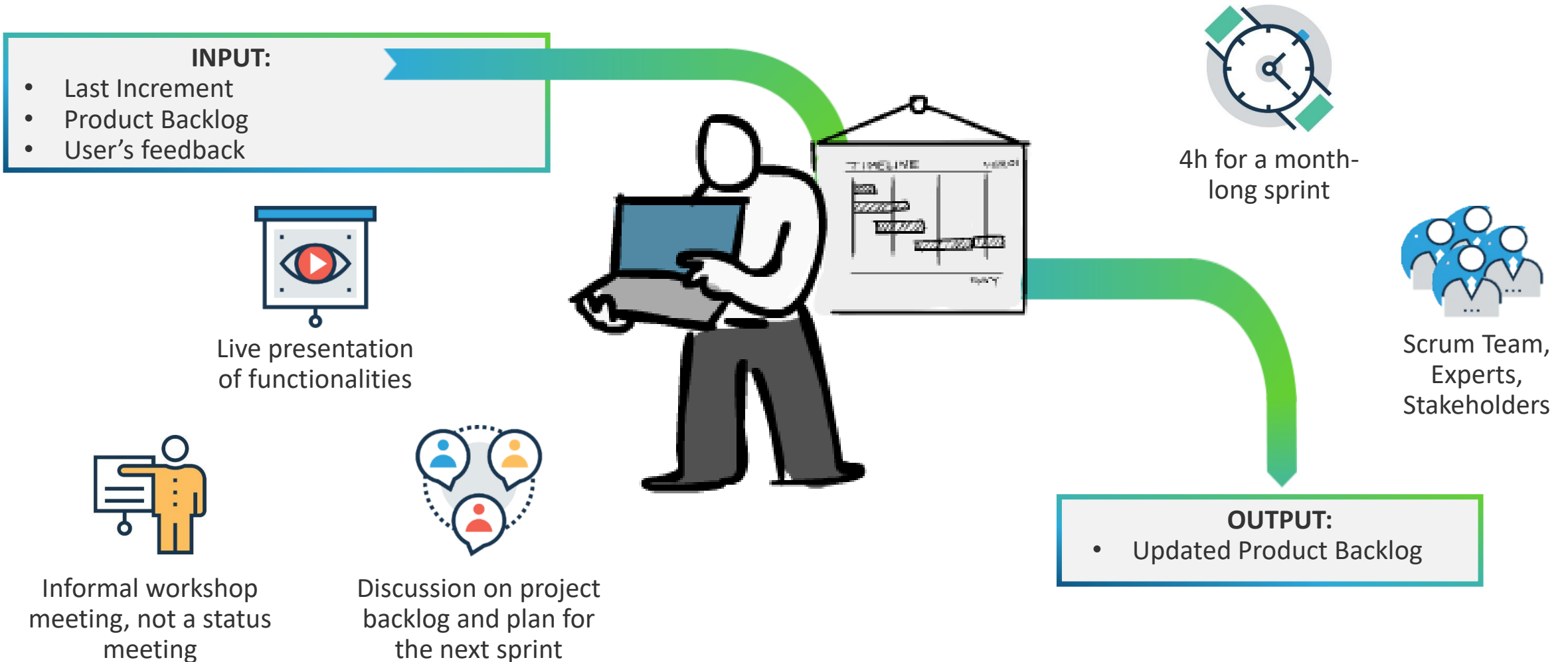
Definition of Done (Example)

- ✓ Unit tests passed
- ✓ Code reviewed
- ✓ Acceptance criteria met
- ✓ Functional Tests passed
- ✓ Non-Functional requirements met
- ✓ PO accepted the User Story
- ✓ User manual updated

Every User Story = Same Criteria



During the Sprint Review the business value of the Increment is inspected and Product Backlog is adapted if needed.





During Retrospective Scrum Team inspects the work methods, processes, tools, relationships, people and anything else impacting the Scrum Team work and tries to adapt it for a better effectiveness.



Discussion how the Sprint went



Improvement implementation during next Sprint



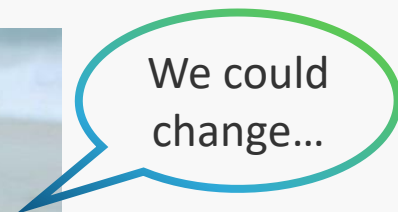
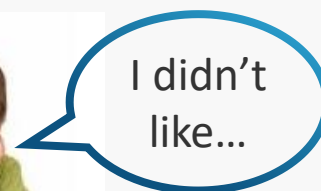
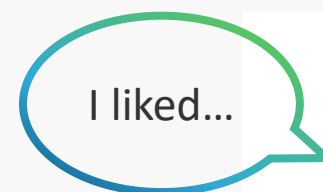
Choosing and focusing on particular improvements



3h for a month-long sprint



Product Owner, Scrum Master, Dev Team



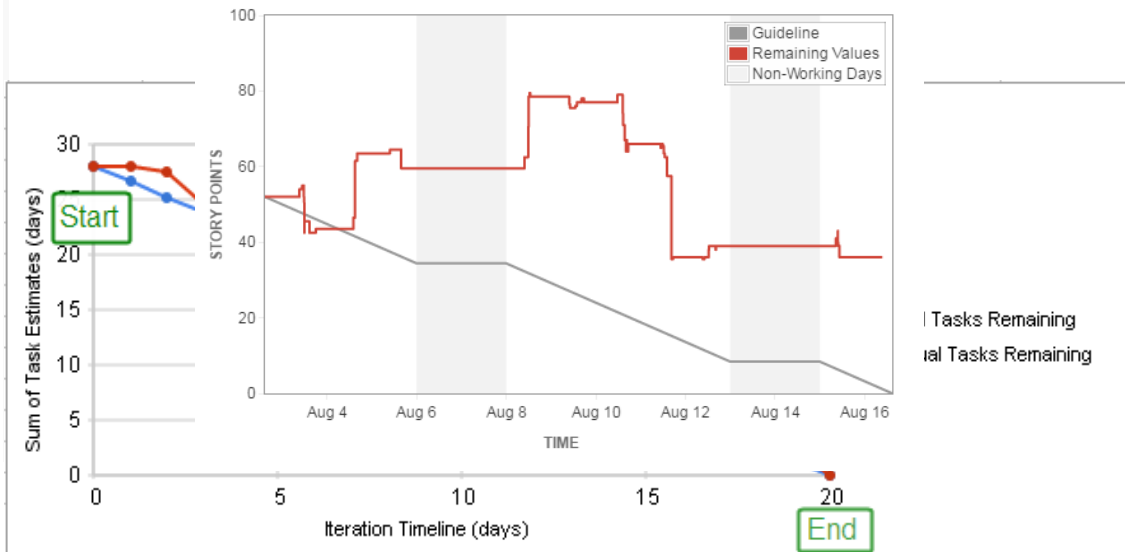
Are you too busy to improve?



The progress of agile project fulfillment can be measured i.e. by the Burndown and Velocity Chart.

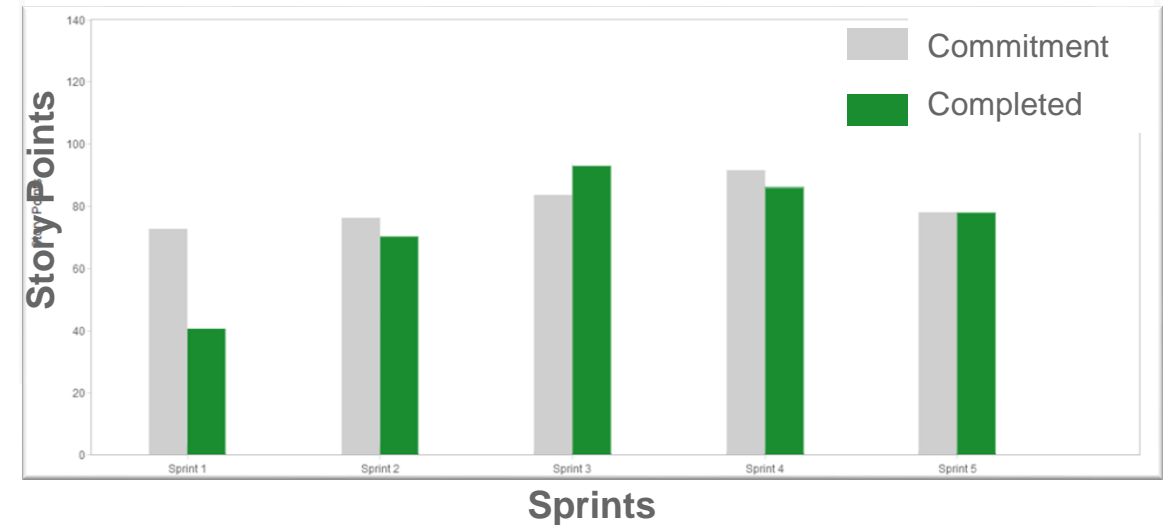
Burndown Chart

Used to monitor progress during a single sprint but also progress in the whole project – **Optimal pace vs. reality**



Velocity Chart

Shows pace of the team work and a change of this pace in the subsequent sprints - **Scheduled vs. Completed Story Points** in each sprint



Agile introduction

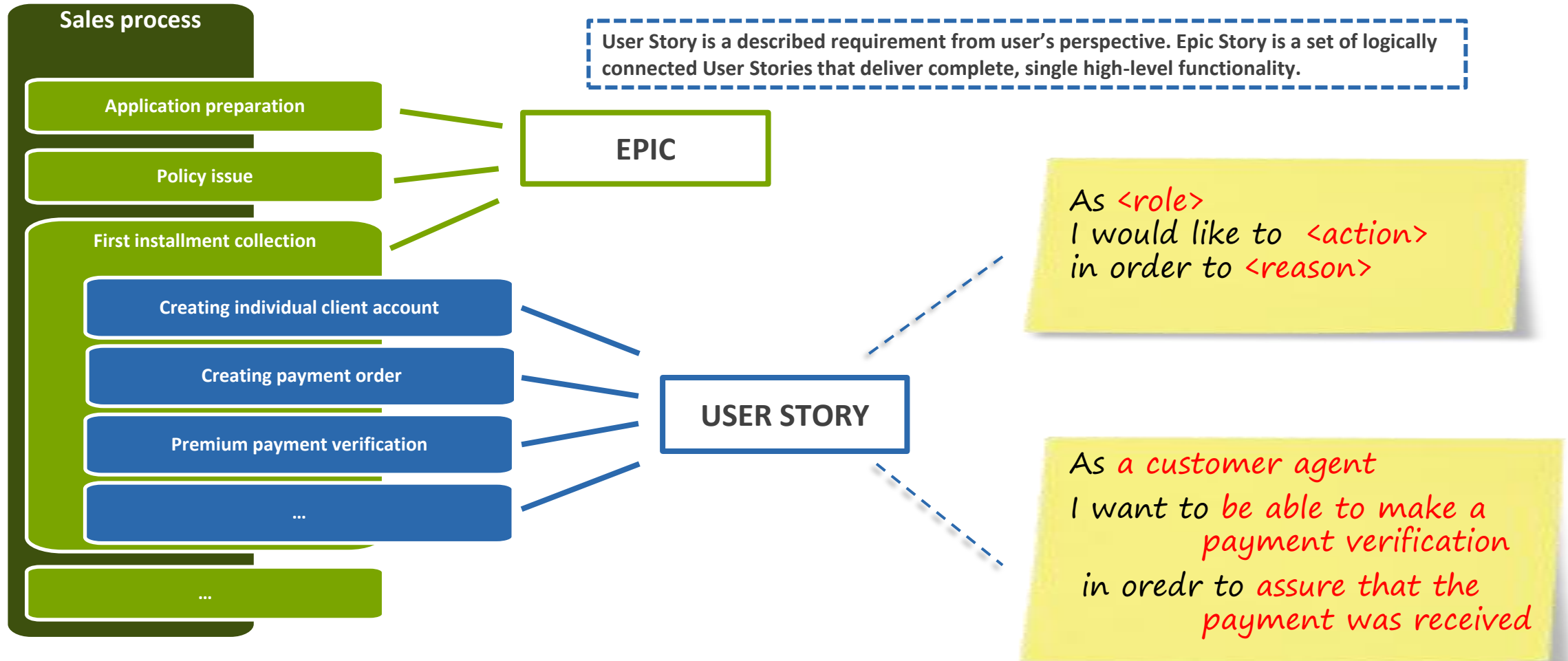
Scrum Team

Scrum events & artifacts

User Stories

People & Teams

Backlog can be defined on different levels of detail, depending on types of issues.



What is User Story?

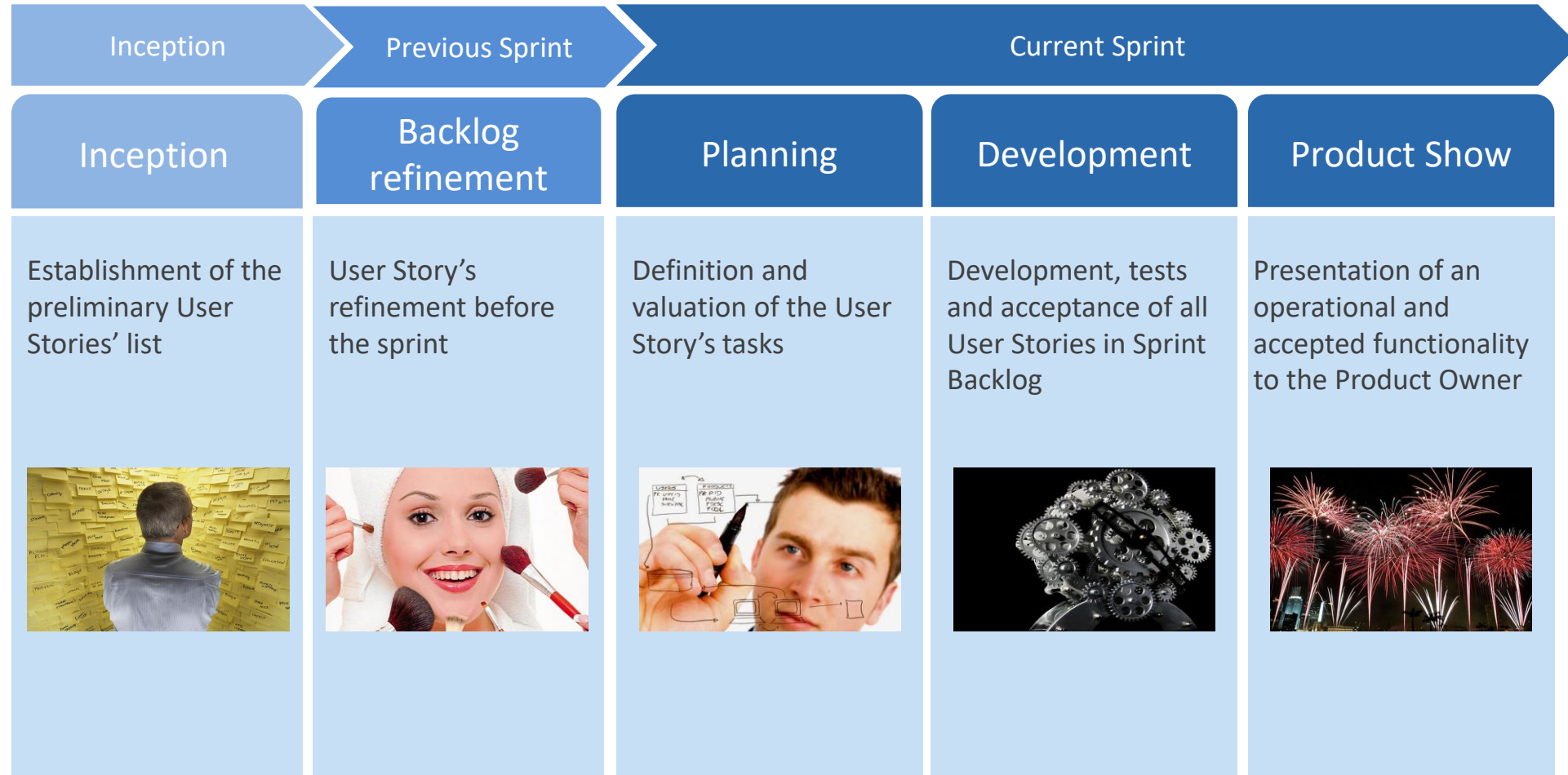
As *<a role>*
I would like to *<an action>*
so that *<an objective>*.

As *a customer agent*
I want *to be able to make a*
payment verification
so that *I know that payment was*
received

User Stories are high-level user-sourced requirements

I ndependent	Independent from the others
N egotiable	Modifiable until the sprint starts
V aluable	Deliver value
E stimable	Possible to estimate
S mall	Completable during one sprint
T estable	Possible to test

User Story's lifecycle starts with the inception and ends on the Product Show.



User Stories are estimated using Story Points, a popular relative measurement.



XL



L



M



S

Each Story Point is combined of:

- **Complexity** – how hard is the problem?
- **Work effort** – how much work do we need?
- **Uncertainty** – how much is still unknown?

Acceptance Criteria - specify conditions under which a User Story is fulfilled.

Acceptance Criteria specify **conditions** under which a **User Story** is **fulfilled**. Concisely written criteria help development teams avoid ambiguity about a client's demands and prevent miscommunication. They:

- ✓ define boundaries
- ✓ help to reach consensus
- ✓ serve as a basis for tests
- ✓ allow an accurate planning and estimation



Definition of “Done” is the **global checklist** that can be applied to all Product Backlog Items or User Stories.
Acceptance Criteria are the things that are **specific to the individual User Story**.

Acceptance Criteria should follow the SMART rule.

S**Specific****M****Measurable****A****Actionable****R****Realistic****T****Time-Based**

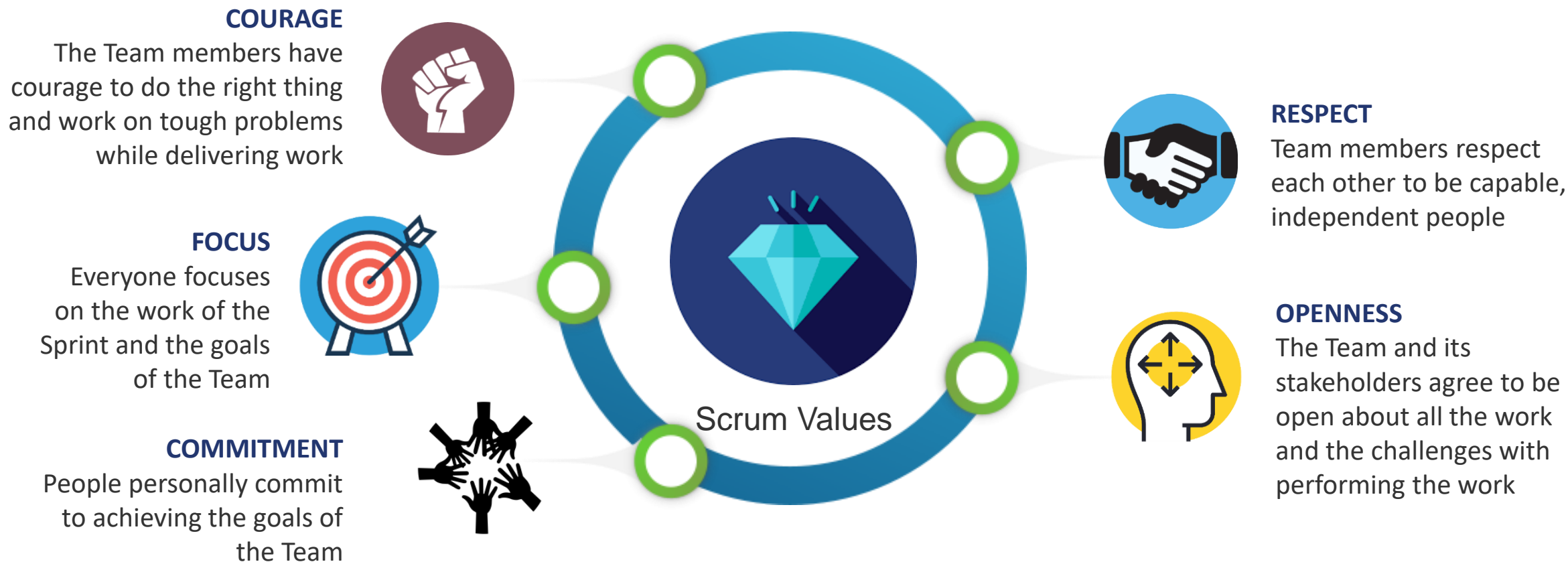
Agile introduction

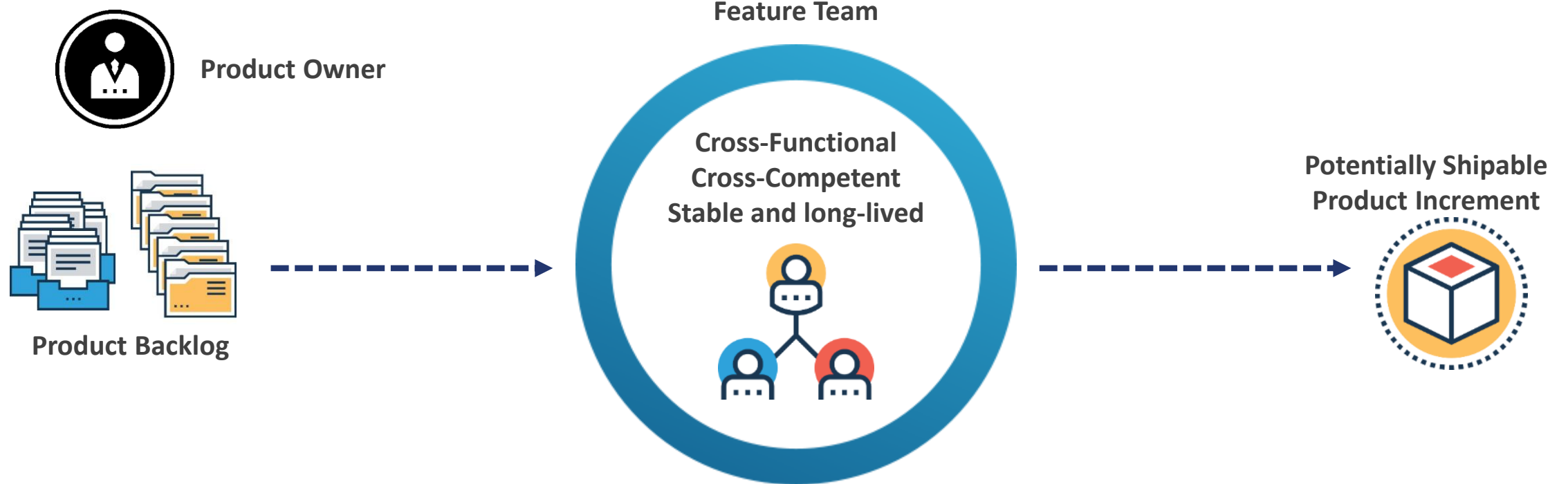
Scrum Team

Scrum events & artifacts

User Stories

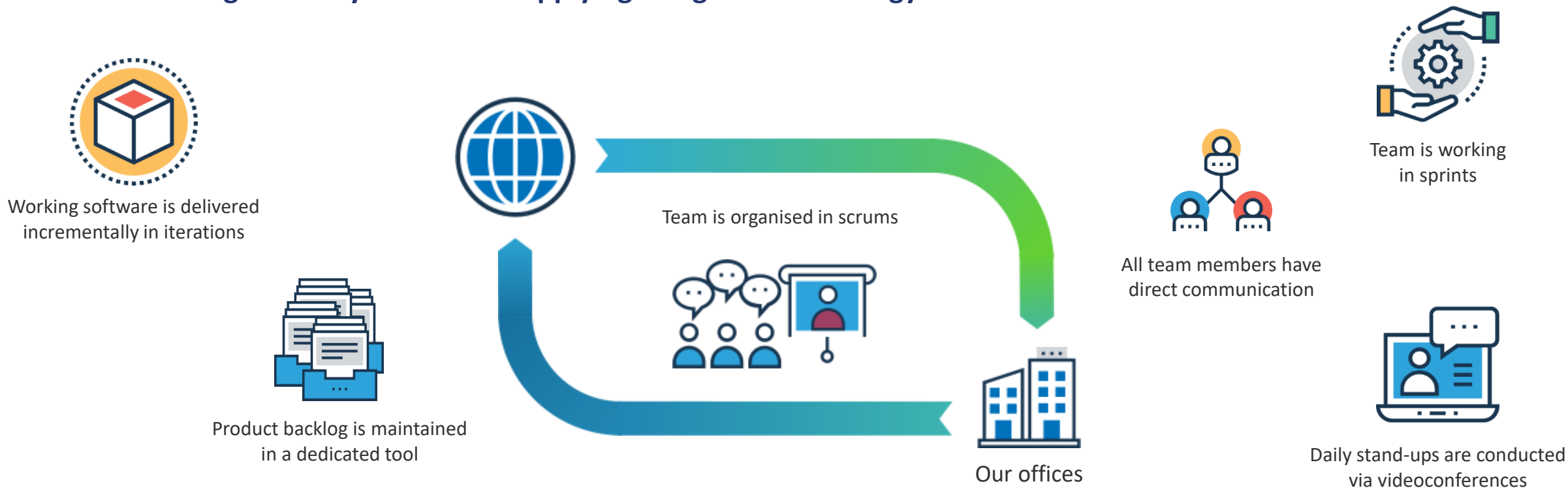
People & Teams





- Each Team has **all skills to turn Product Backlog into Increments** of working software
- Work is **integrated continuously** within each Sprint
- Vertical slicing: work is **divided by end-user functionality**
- Each Team delivers **end-to-end functionalities**
- **Transparency ensured**; No unknown, undone

While working remotely the team is applying all agile methodology attributes.



Team can tune the working model to adjust it to specific project needs

Reaching cost efficiency for the project and delivering better value for business