

How to work in Agile

WATERFALL

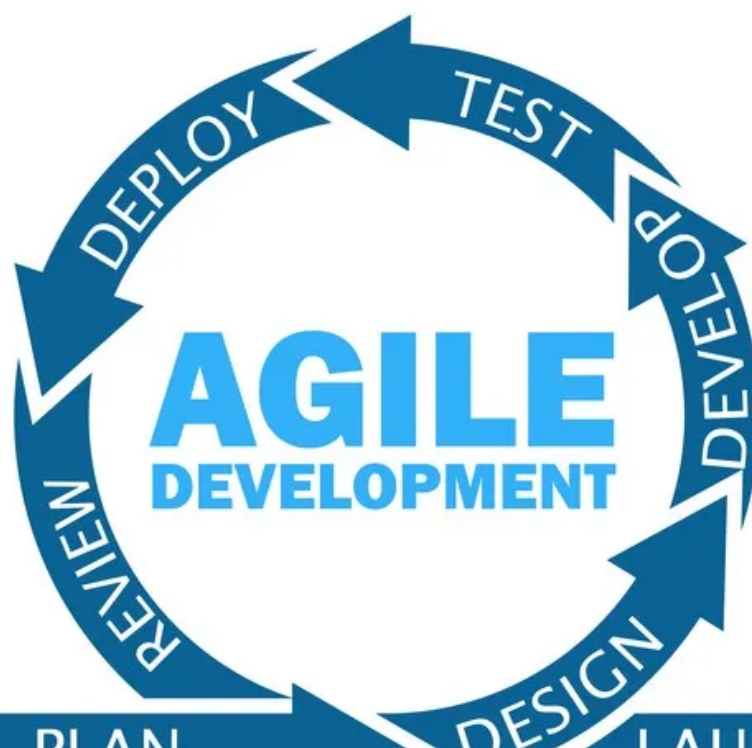
AGILE

Many brilliant minds strived to provide an approach for IT that permitted the adding of changes to the digital project without losing transparency, priorities and order in the complicated working process of the development agency.

The result of their work is Agile – a method designed to make it possible to work under conditions of instability and ever-changing requirements.

Agile should be understood rather like a philosophy to adopt than a set of rules imposed from above. Its most significant trait is to organise the work of a team efficiently so that the client can see valuable results. You can discover more about the principles of Agile in its [Manifesto](#).

Agile was thus born to support the process of product evolution while making software development a more client-friendly process.



Customer side

Agile values real-life communication over precise documentation. Here we can represent the original citation from the Manifesto:

"Working software over comprehensive documentation".

This means that you, as a client, will set the short-term goals along with the developers, discuss priorities and evaluate results on the go. You provide your vision on the critical issues so the team can solve any given problem ASAP.

Developers side

This encourages teams to be self-organised, to brainstorm solutions instead of just doing the coding based on 100+ pages of prescribed documentation. In Agile, teamwork is as important as ever. Team members are encouraged to ask for and provide help to each other and to be competent in related areas.

Advantages and disadvantages of Agile

Agile is very beneficial in some situations, but at times can do more harm than good. Let's have a look at some examples.

When to use it:

- **Eliminates risks with a new product on a new market.** When the project requirements are swiftly changing, Agile is the solution. It's designed to find out ON THE GO what works best within current market terms.
- **Spares labour and material resources with a simplified solution.** First, develop a basic solution – for example, a simple material fluctuation report. Later you can get back to its deployment: add more features or an attractive design. Then, immediately after release, you already have a tool to hand that calculates all the data for you.

IMPOTANT VALUES TO THE AGILE SOFTWARE DEVELOPMENT MODEL



When to avoid it:

- **Requirements are crystal clear.** There is simply no use working in Agile if all the software functionality is defined and clear – just feed all the demands into the software requirement specification and set the deadline.
- **High-load project with multiple complexities.** With the development of a very large and resource-intensive project, such as Google, using Agile is not recommended. It's necessary to aggregate and analyse the rough scope of work and plan the program architecture in advance, otherwise, the hardware demands or similar limitations can halt development halfway through. Yet even developers remain ambivalent on this point.

Frameworks

Sets of practices inside Agile that give a more precise idea of what to do to increase the efficiency of teamwork and get as close as possible to continuous delivery are called frameworks.

Companies that have adopted Agile implement different frameworks based on what they focus on.

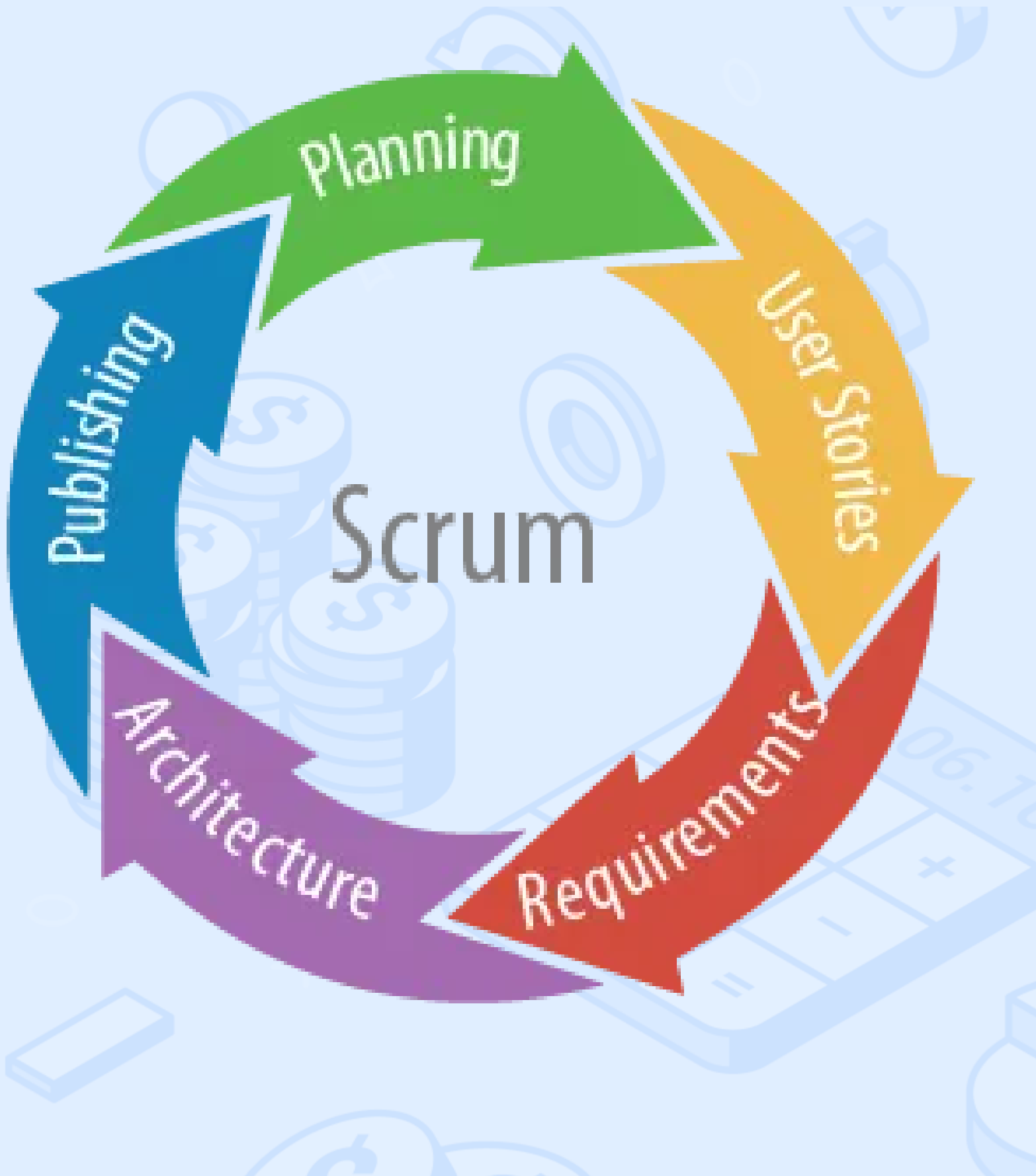
Scrum

The most popular Agile framework is called Scrum.

At the base of this framework, there are three roles:

- a Product Owner,
- a Team
- and a Scrum master.

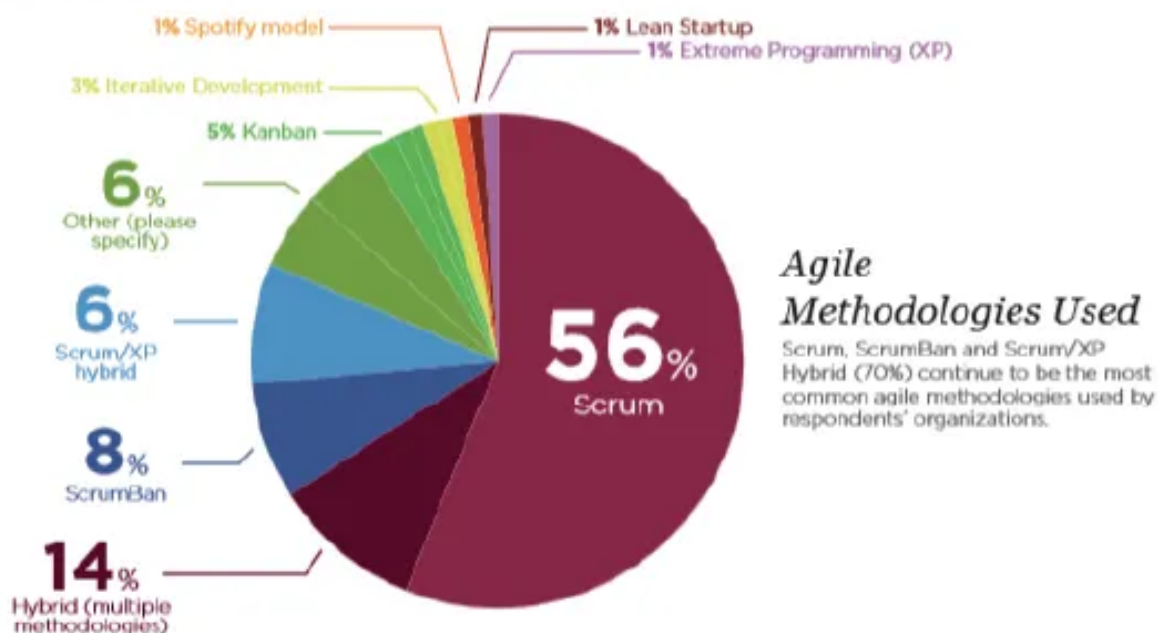
They work in short (2-4 weeks) iterations.



The culmination of each step is a demo, a project presentation.

- The first great advantage of Scrum is that the client is always aware of the in-progress results. The framework is intentionally made very transparent.
- The second is that, at the end of an iteration, you get a project with complete, planned features. It's legit to test it or to show it to sponsors.

AGILE METHODS AND PRACTICES



Kanban

At the heart of the Kanban framework lie 6 principles.

These are more like a working philosophy to adopt than a set of requirements.

- Kanban focuses on delivering work just in time. It was first used for manufacturing and continues to be used in this way today – for example, in Toyota factories. It is perfect for projects with high social responsibility.

- Unlike Scrum, Kanban has no roles – the PM simply adds tasks and declares their business priorities.
- The main goal for an IT team working under Kanban is to solve the task, optimising the means of implementing it. So if the members really follow this framework rather than just claiming to, deadlines will certainly be met.

Lean

This is a popular method not just in development but also in business – for example, in building startups. Lean saves time on unnecessary operations. This is also called “waste reduction”.

Lean’s main principle says: “If you aren’t sure you need it, don’t do it”.

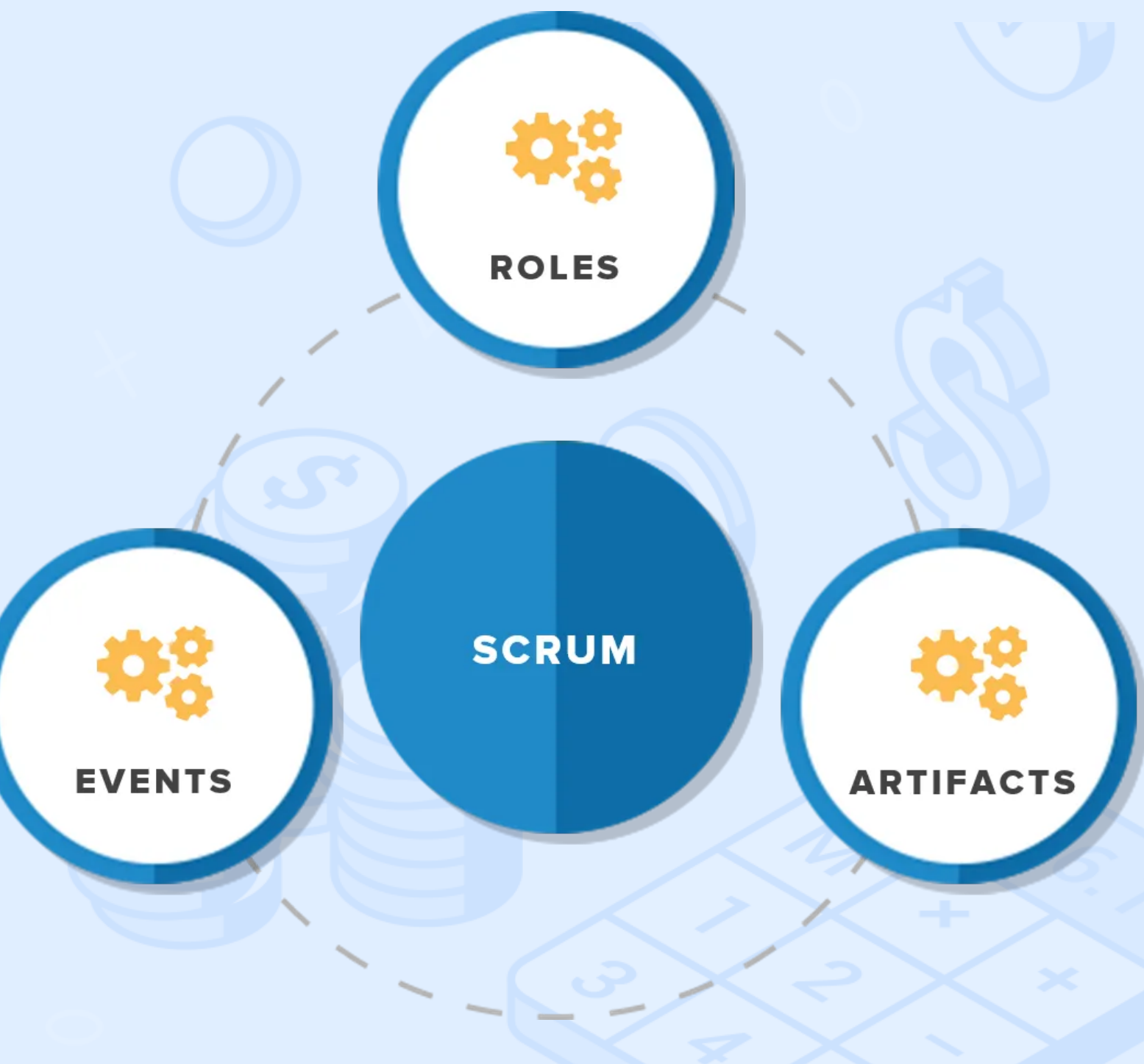
Discover more about this framework [here](#).

Introduction to Scrum

Let’s talk in more detail about Scrum. In our company, like many other enterprises in the world, we also work with it. Let’s have a closer look at how it all works.

Agile Scrum team structure

A Scrum team consists of three parties – a Product Owner, a Scrum team and a certified Scrum Master.



In Agile, a **Product Owner** isn't a client as you might imagine one. Rather, they're an expert with strong development knowledge, working as a project manager and a brand manager in a traditional business structure. They aggregate the customer's requirements and orchestrate product development, implementing the communication between the client and the developers.

A **Scrum team** is planned based on project demands and usually includes designers, developers, business analysts, UI experts, testers, etc.

Every team is autonomous and the members interchangeable. Via different Agile team building activities, the synergic effect of team members working in tandem makes the development process be more fruitful.

United crew members share their competence, brainstorm tasks and help each other where necessary.

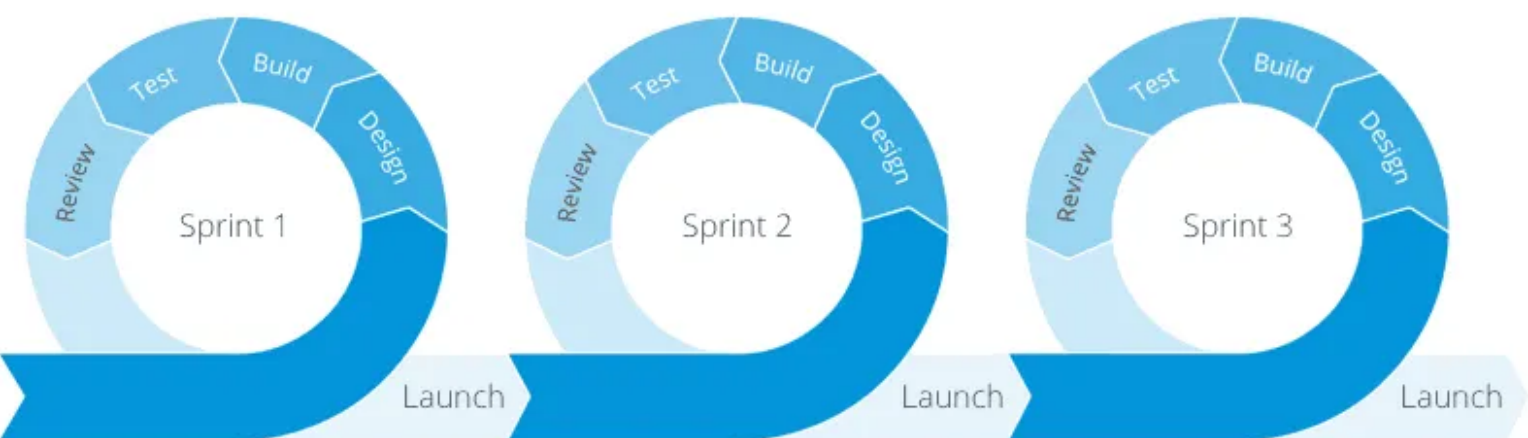
A Scrum master is likely the most unique among the Agile software development team roles. This person is responsible for integrating the principles of Scrum into the company. They care for the team, their comfort and professional growth. They facilitate meetings and point out the mismatches in the following Scrum. Ideally, a Scrum master is guided by the philosophy of "[servant leadership](#)".

SERVANT LEADERSHIP CHECKLIST

☐ Self-Aware☐ Foresight☐ Collaborative☐ Humble☐ Listen☐ Trusting☐ Integrity☐ Doesn't Abuse
Authority☐ Coach☐ Result-Oriented☐ Intellectual
Authority☐ Resolve
Conflict

Sprints as a measure of time

Contrary to Waterfall, Agile frameworks are divided into short time periods. Usually, they are around 2-4 weeks long. During each stage of development, the team completes a planned set of tasks from the backlog.



What is User Story in Agile methodology?

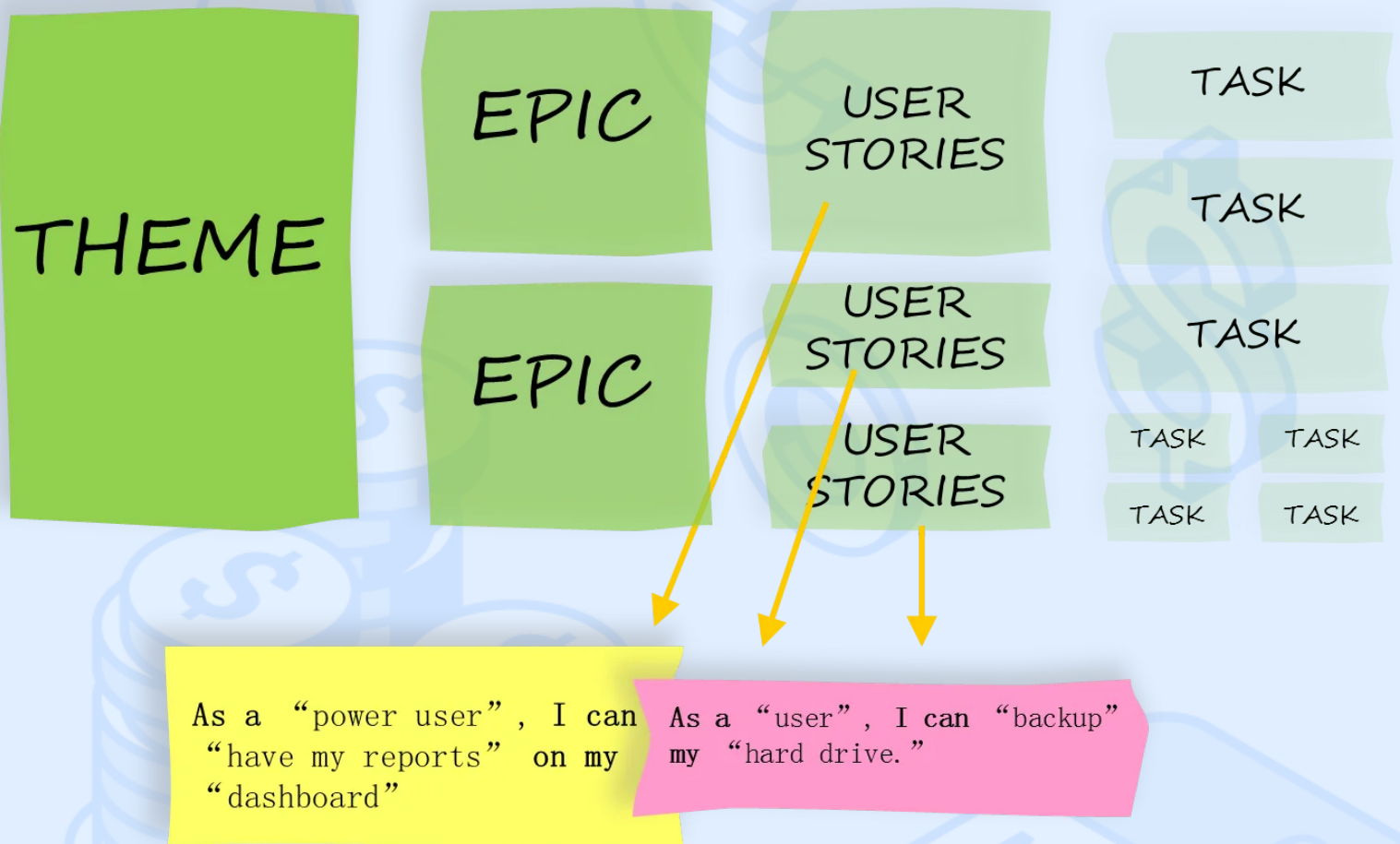
Right from the first meeting, while working with Scrum, we create not software features but User Stories. These are descriptions of user business goals.

For a traffic monitoring tool, for example, the goal can be formulated like this: “As an internet marketer, I want to see the statistics for how many users visited the website in my account”.

For a SaaS development, it can sound like this: “As a Project manager I want to have information about completed and open issues and their assessments in order to be sure the product will be ready by the due date.”

Guide: [how to create a user story.](#)

USER STORIES



www.agile-scrum.be

User Story Benefits

- Reflect users' needs;
- Easy and quick to understand;
- Can be set as milestones;
- Simplify project sprint planning.

Epic Stories

One of the significant rules is that every user story should be finished inside the iteration. Sometimes the goals are so big that this is impossible to cover within one sprint. Such items are called Epic User Stories or epics.

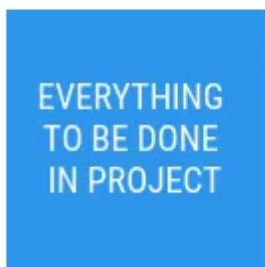
To be finished within an iteration, they should be subdivided into smaller User Stories. To assess the complexity of a User Story, use Story Points.

How to use a Backlog

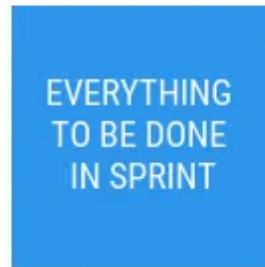
First of all, a Backlog is a list of backlog items that is being constantly enriched with additional goals. Some of these are planned at the first meeting with the clients, while others appear unexpectedly throughout the course of development.

Product Backlog, Sprint Backlog Process

PRODUCT BACKLOG



SPRINT BACKLOG



STEPS TO COMPLETE



PROJECTMANAGER

To deal with these items, Scrum teams often use boards. These can be physical task boards with stickers or Scrum online boards.

To make such a Scrum task board you must first collect a set of features, subdivide them into smaller operations and write everything down on a sticker.





















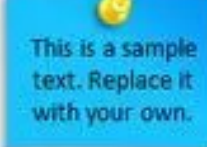
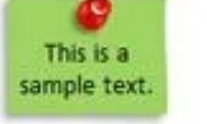

The Scrum task board is divided into three parts (sometimes more) such as:

- to do;
- in progress;
- done.

The stickers are moved from the left to the right until all of them end up in the last graph.

Scrum Task Board Template

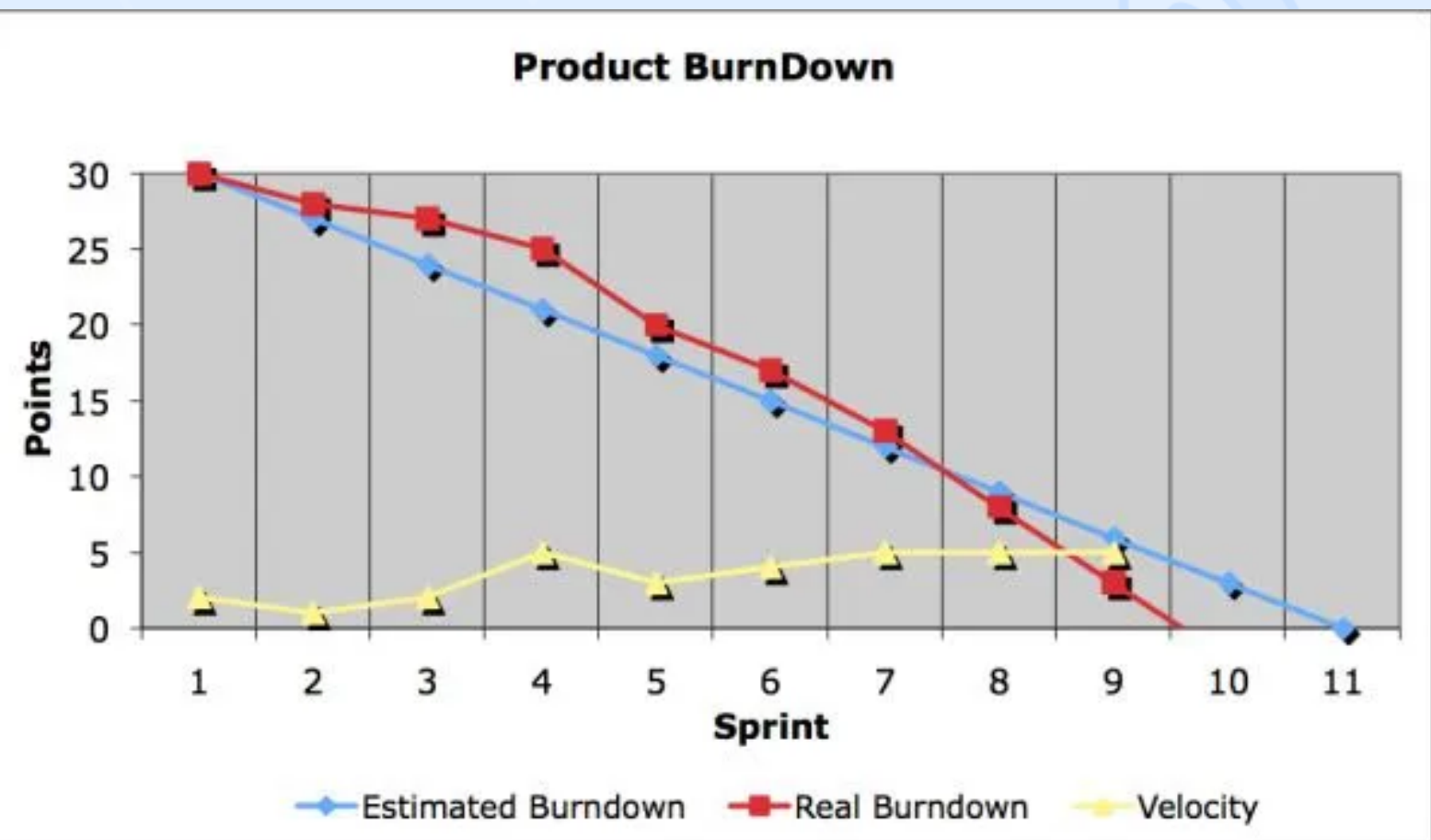
Company name

Stories	To Do		In Progress	Testing	Done
 <p>This is a sample text. Replace it with your own text.</p>	 <p>This is a sample text. Replace it with your own text.</p>	 <p>This is a sample text. Replace it with your own text.</p>	 <p>This is a sample text.</p>	 <p>This is a sample text.</p>	 <p>This is a sample text. Replace it with your own text.</p>
	 <p>This is a sample text. Replace it with your own text.</p>	 <p>This is a sample text. Replace it with your own text.</p>	 <p>This is a sample text.</p>	 <p>This is a sample text.</p>	
			 <p>This is a sample text.</p>	 <p>This is a sample text.</p>	 <p>This is a sample text. Replace it with your own text.</p>
 <p>This is a sample text. Replace it with your own text.</p>	 <p>This is a sample text.</p>	 <p>This is a sample text.</p>	 <p>This is a sample text.</p>	 <p>This is a sample text.</p>	
	 <p>This is a sample text.</p>	 <p>This is a sample text.</p>	 <p>This is a sample text. Replace it with your own.</p>	 <p>This is a sample text.</p>	 <p>This is a sample text. Replace it with your own text.</p>

Burndown chart

A burndown chart is used during longer iterations to see if the team will be able to deliver all of the proclaimed features.

Each chart has two lines, showing the estimated vs. actual effort of the Scrum tasks. Find out how to draw them [here](#).



Events in Scrum

Scrum has 4 official events that must be conducted with regulated frequency. The main difference between this and other frameworks is that as soon as one of its rules is broken, it is no longer Scrum.

1. Sprint Planning Meeting

The starting point of the sprint. The product owner sets sprint goals and, with the team, selects user stories from the backlog to achieve such goals.

At the planning meeting, the team also brainstorms how it can be realised step by step. The members assess the tasks' complexity and how many hours it will take. In the process, the relevant information about each User Story is posted on the Scrum board.

2. Daily Scrum or Daily Stand-Ups

According to Scrum guidelines, Daily Standups should be conducted for the team so that every member is aware of what point their teammates have reached.

Everyone is asked these 3 questions:

- What did you do yesterday?
- What do you plan to do today?
- Are there any impediments?

The daily stand up meetings are facilitated by the Scrum master. They must never exceed 15 minutes.

If there are any major outstanding issues, the team members can agree to discuss it afterwards.

Event	Inspection	Adaptation
Sprint Planning	<ul style="list-style-type: none"> Product Backlog (Commitments Retrospective) (Definition of Done) 	<ul style="list-style-type: none"> Sprint Goal Forecast Sprint Backlog
Daily Scrum	<ul style="list-style-type: none"> Progress toward Sprint Goal 	<ul style="list-style-type: none"> Sprint Backlog Daily Plan
Sprint Review	<ul style="list-style-type: none"> Product Increment Product Backlog (Release) Market-business conditions 	<ul style="list-style-type: none"> Product Backlog
Sprint Retrospective	<ul style="list-style-type: none"> Team & collaboration Technology & engineering Definition of Done 	<ul style="list-style-type: none"> Actionable improvements

3. Sprint Review

At the end of every sprint, the team has a review meeting. The members demonstrate what has been done to the Product Owner. The client may participate in the meeting as well. The Product Owner goes through the acceptance criteria and, if everything is okay, all the tasks are marked as done.

The Product Owner collects feedback from the client and analyses the backlog to actualise the product status according to new inputs. A revised backlog is a necessary result of a sprint review.

4. Retrospective Meeting

This is the final meeting of the iteration where the Scrum team meets, discusses and documents the following:

- What went well during the Sprint?
- What could have been done better?
- Lessons learned.
- Points of action.

The team should strive to implement the best practices and learn from their mistakes in the next iteration.



How we do it at Magora

At Magora, we've also chosen Scrum as the main Agile framework. We believe that it serves as the best way to ensure we satisfy our clients' and their users' needs.

As for the nuances of Agile development, we'll prolong these until our next publication. In the meantime, we're always glad to answer your questions.