software
development
academy

Agile and Scrum

Paweł Golec


software
development
academy
Welcome
Agile and Scrum

Few words about me

- Working in IT from 2013
- Professional Scrum Master III, Professional Scrum Product Owner and Professional Agile Leader
- Developer, Scrum Master, Product Owner
- Currently freelancer, one of the authors of blog No Fluff Jobs and Trainer at Software Development Academy

Agenda

1. How we can develop software, about Waterfall (cascade model) and Agile.
2. Why Agile?
3. What is Agile Manifesto?
4. What Scrum is?
5. Why most comapnies work inScrum?

Benefits

- Increase your chance during the recruitment process.
- A better chance of finding a job
- Learning the most popular software development methodology in the world
- Prepare for the first days of your work
- The opportunity to change your career path

software
development
academy

How software can be created?

Cascade Model - Waterfall

Agile, Scrum

In this video you will learn:

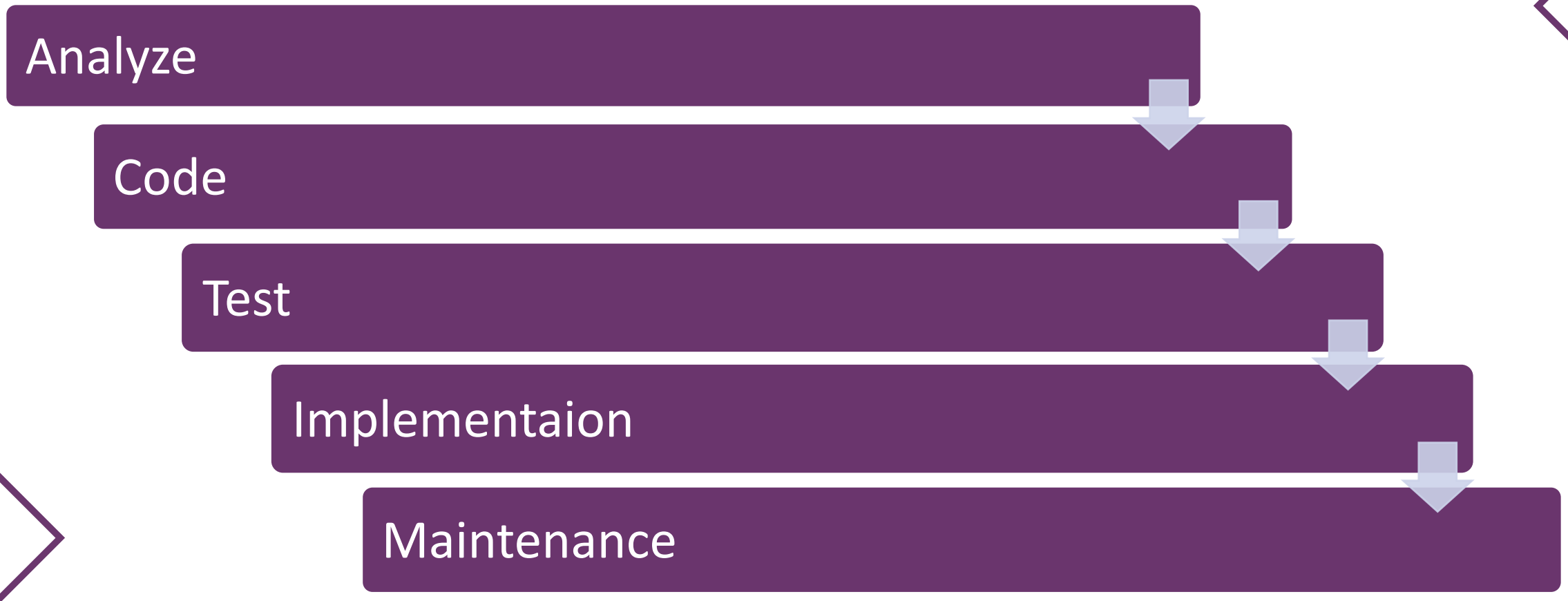
- What characterizes the cascading approach- Waterfall?
- What phrases does it contain?
- Why did companies stop using Waterfall?

What is Waterfall?

Waterfall (also called the cascade model) was written around 1970.

It assumes **cascading way of work** - step by step, phase by phase, stage by stage

Possible examples of model phases



Why did we stop using Waterfall?

What is Waterfall?

None of the production phases can go on simultaneously (the phases do not interpenetrate).

Waterfall comes from management methods specific to the tape ("factory") method of software development.

Waterfall


1UP
010900
HIGH SCORE
010900
L=02
BONUS
4700


Macintosh
Printer Disk 5
Installation 5
Druckersoftware 5
Stampante Macintosh 5
Pripter Disk 5
Software Impresora 5
Skrivarprogram 5
Skriverprogramvare 5
Printersoftware, 5
Kirjoitinohjelmisto 5

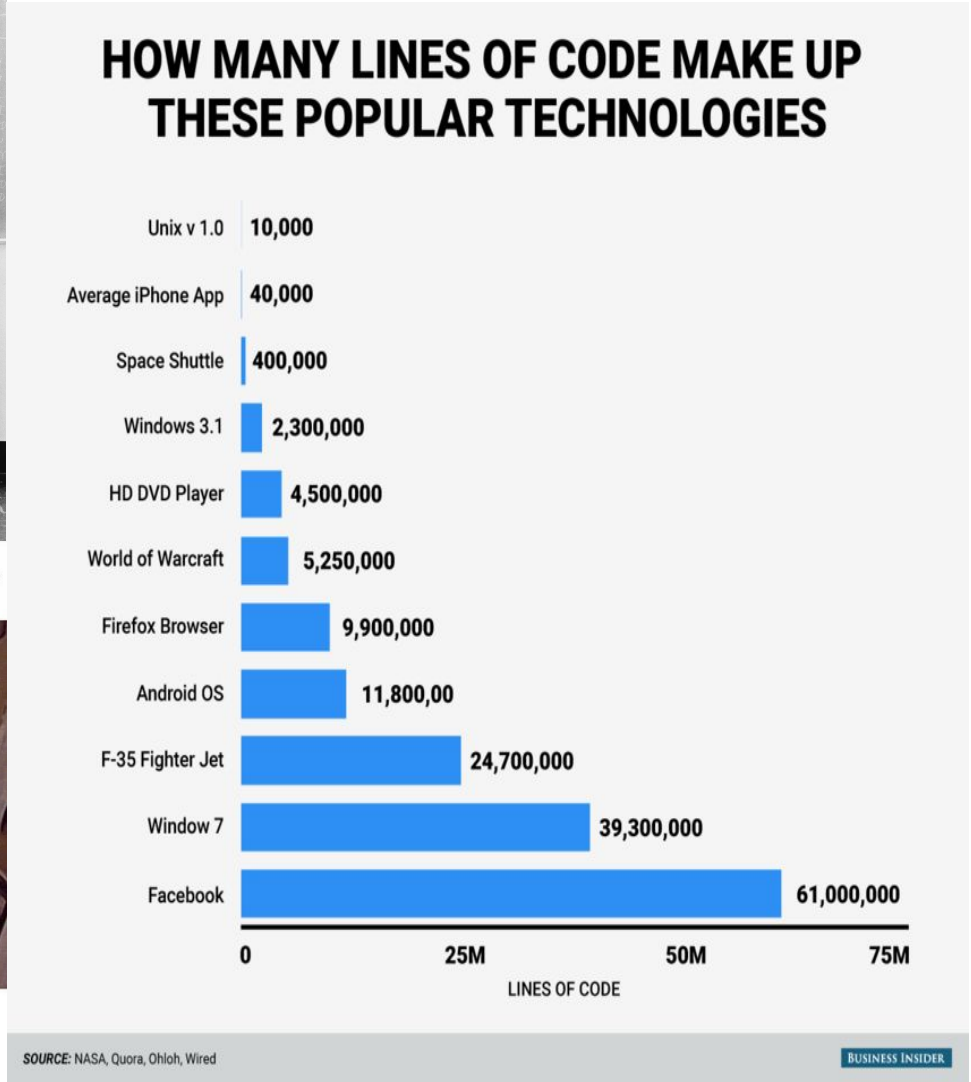

FAMILY COMPUTER
Nintendo

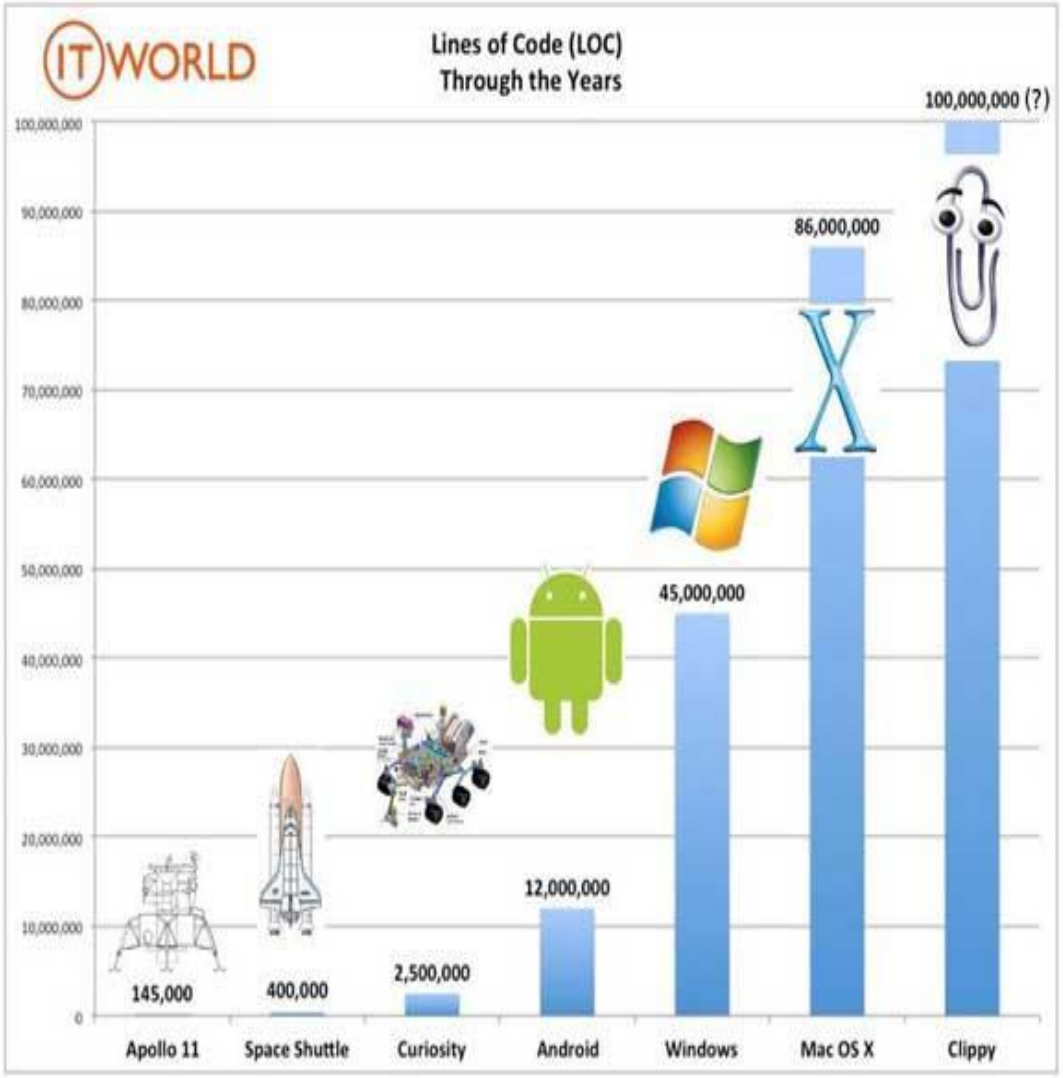

CBM

Waterfall

Ladies and gentlemen: Jeff Bezos (CEO of Amazon) in his first office 1999

Inspirational photo of the day. Everyone starts somewhere. ~N





Why did we stop using Waterfall?

The development of the IT industry has required a change in the way software is developed - examples such as the increasing complexity of the software being created, the number of people involved in creating the product, and the need to maintain it have caused Waterfall to cease to be as effective as it used to be. However, Waterfall's biggest drawback was ...

Why did we stop using Waterfall?

Low flexibility when it comes to changes and long delivery time of the finished product.

After watching this video:

- You learned main principles of the Waterfall cascade approach in software development.
- You have learned what the biggest disadvantages of this methodology are.

software
development
academy

A short trip into the past

Agile Manifesto

Agile, Scrum

In this video you will learn:

- What was the history od the creation of Agile methodologies?
- What is Agile Manifesto and how is it characterized?

Short trip into the past.

Agile methodologies were created in the 90s., but in February 2001 there was **a very important meeting in the history of software development**.

17 people met in Snowbird, USA, who created a document called the **Agile Manifesto.**

Agile Manifesto

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

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

That is, while there is value in the items on
the right, we value the items on the left more.

Individuals and interaction **over** processes and tools

- Every organization consists of people.
- Each organization is different because the people who work in it are different.
- Processes and tools are to serve us -never the other way around.
- Processes and tools are important, but people and interactions should be more important to us.

Working software over comprehensive documentation

- Software is something that users will use.
- This is the way through which our client gains funds or other benefits.
- We create documentation at the right moment and in a degree corresponding to reality.
- Documentation is important, but this running software should be more important to us.

Customer collaboration **over** contract negotiation

Formal agreements (such as contracts) are important, but we should always be looking for cooperation (partnership).

Responding to change over following a plan

- Each plan, even the best, does not guarantee success.
- Changes in the produced software are something natural.
- One of the biggest advantages of Agile is the ability to react quickly and effectively to change.
- Following (blindly) our plan can cause huge damage.

After watching this video:

- You learned the history of the creation of Agile Manifesto.
- You found out what the main assumptions of Agile methodologies are.

In this video you will learn:

- What is Agile?
- What phases it is devided into?

How software can be created?

Agile

Agile, Scrum

What is Agile?

Agile is an alternative to Waterfall when it comes to developing software.

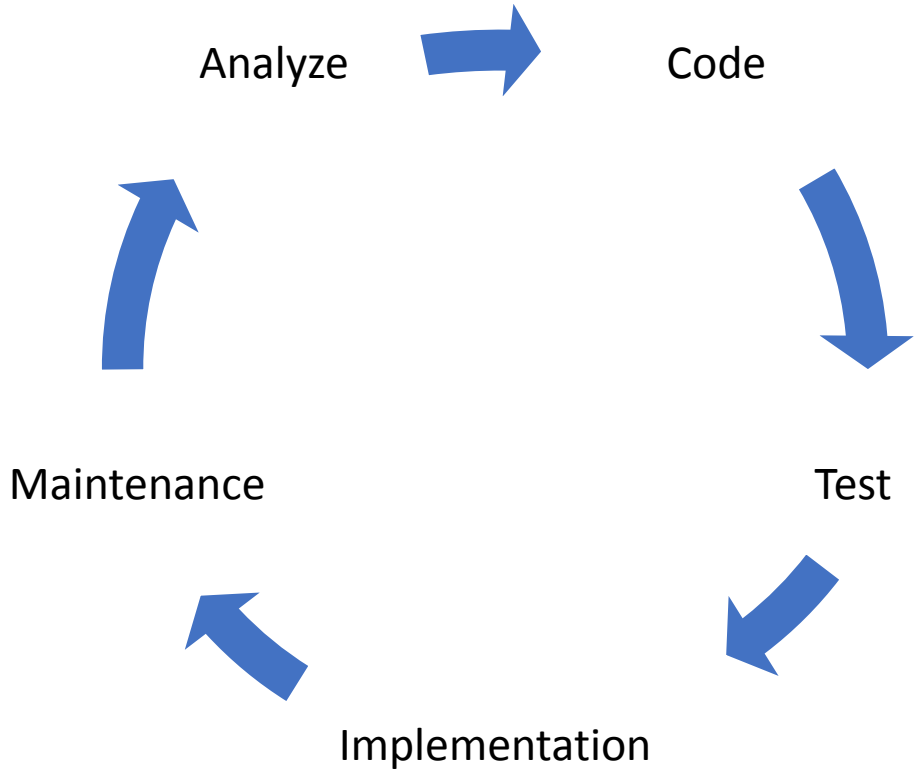
The model assumes software development in an Agile way, in short iterations and frequent delivery of value.

Agile production phases

Agile consists of exactly the same production phases.

The difference is the way of working in these phases and the method by which the software is produced.

Agile development phases



Agile development phases

The difference to the Waterfall model is that all these phases are running simultaneously in a fixed time period called **Iteration**.

In each iteration, every phase is present in smaller scale.

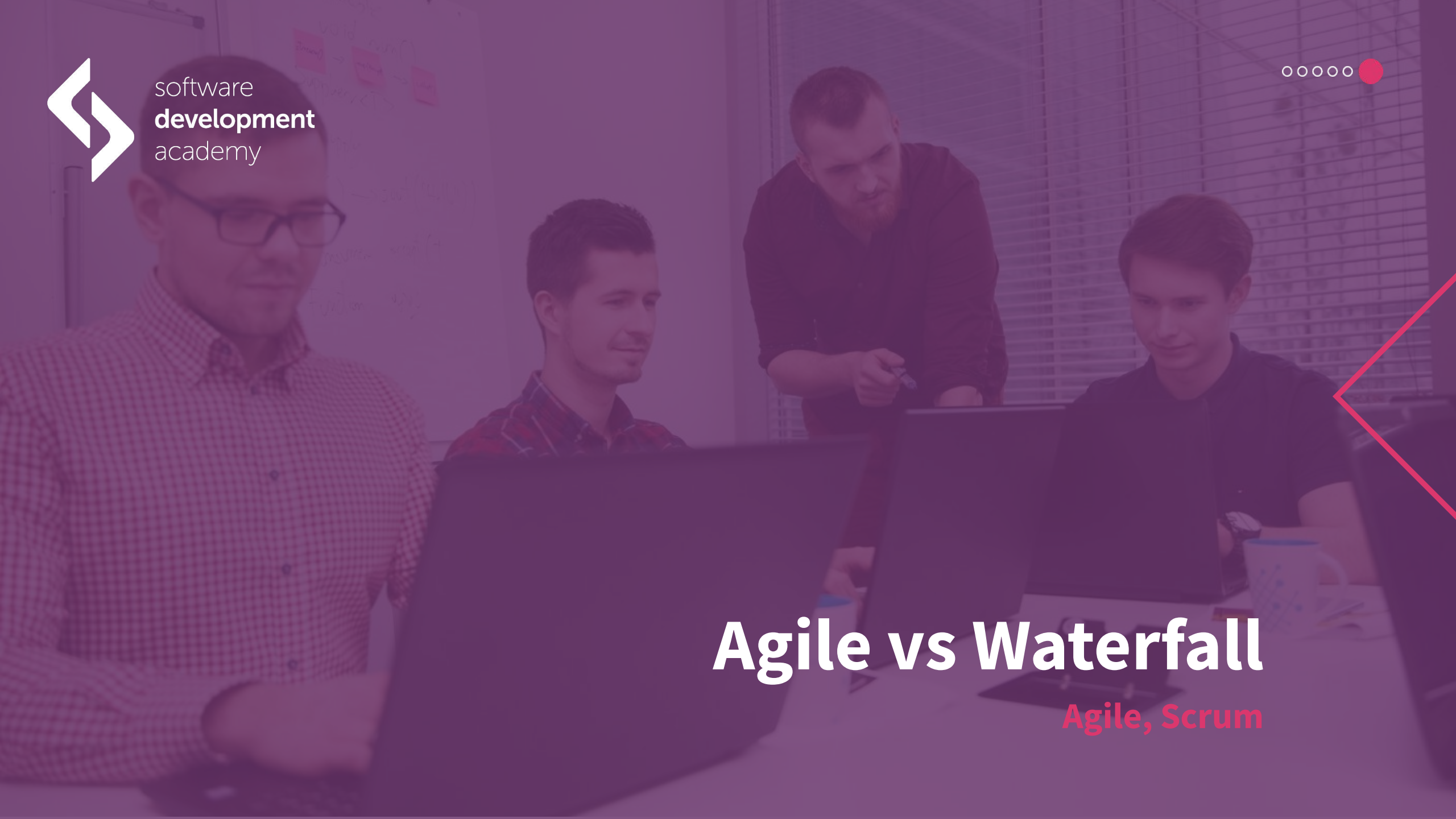
Agile development phases

The advantages of this approach are, for example:

- Bigger resistance to changes (in case of a mistake, we are able to correct the phase which lasted relatively short).
- **Faster delivery of working software** (we learn to deliver the product in **small, ready-to-use fragments**).
- Ability to work on **several elements at the same time** (less risk of wasting valuable resources).

After watching this video:

- You know the definition ofAgile.
- You know what interation is.
- You can name the main phases of software development:
  - Analyze
  - Code
  - Test
  - Implementation
  - Maintenance



software
development
academy

Agile vs Waterfall

Agile, Scrum

In this video you will learn:

- What differences can be identified between the Waterfall cascade approach and the Agile approach?

Agile vs Waterfall

Agile

- Agile can be used in small and large products
- Relatively new
- High readiness for changes
- Product release often, in small fragments

Waterfall

- We use Waterfall rather for large products (e.g. contracts between corporations)
- It has existed for several dozen years
- Reluctant to change
- Product release after a long time

After watching this video:

- You can point out the main differences between Agile and Waterfall.

What is Scrum?

Agile, Scrum

In this video you will learn:

- What is Scrum?
- What Scrum Flow looks like?

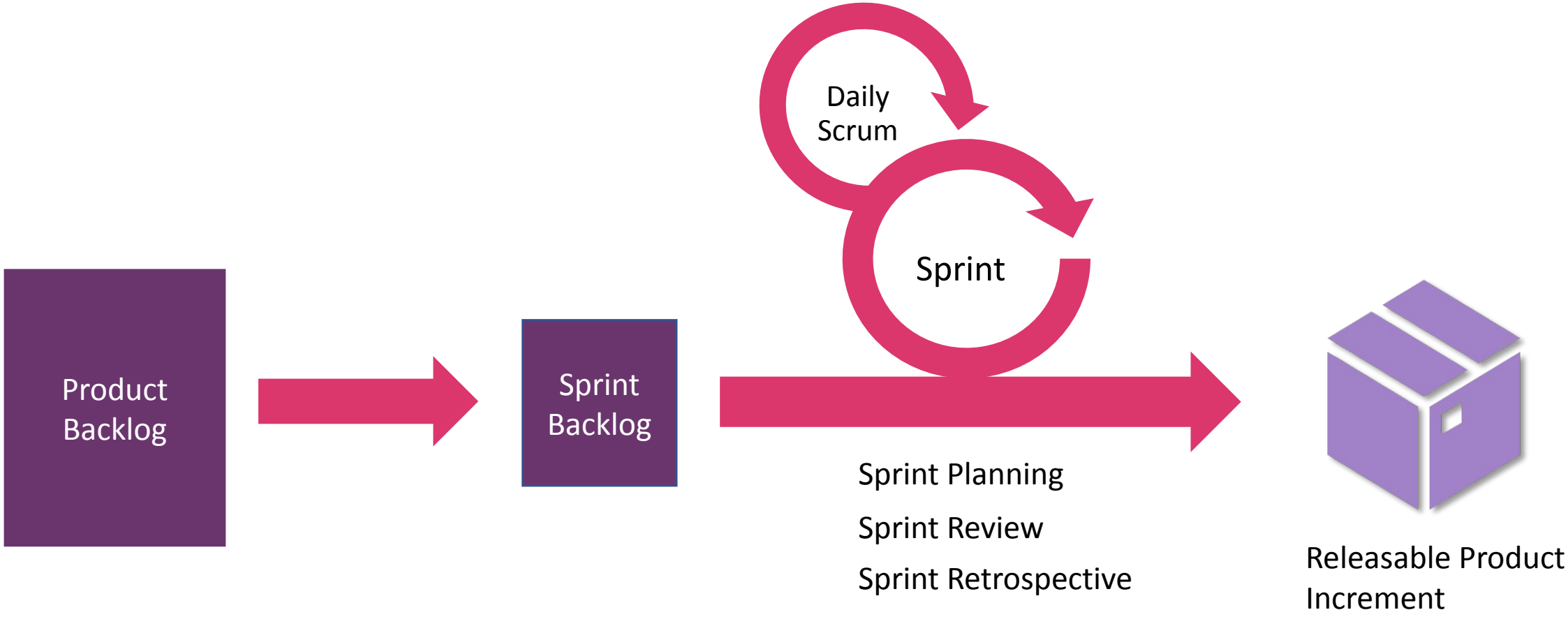
Scrum Guide

- **Title:** The Scrum Guide
  The Definitive Guide to Scrum: The Rules of the Game
- **Author(s):** Ken Schwaber and Jeff Sutherland
- **Link to source material:** http://scrumguides.org/
- **License:** http://creativecommons.org/licenses/by-sa/4.0/legalcode
- **No Changes:** No changes were made to the original document, but parts of it are used and cited/quoted in this course.
- **No Restrictions:** I do not apply any legal terms or technological measures that legally restrict others from doing anything the license permits.
- **The licensor of this license in no way endorses me or my use.**

What Scrum is?

- Scrum is the most popular Agile implementation.
- It is the most popular Agile framework and is a kind of extension of the Agile idea.
- Scrum Guide is just a 13-18 (depending on the language version) pages document that describes this framework.
- Its current version is from November 2020.

Scrum flow


Daily
Scrum
Sprint
Product
Backlog
Sprint
Backlog
Sprint Planning
Sprint Review
Sprint Retrospective
Releasable Product
Increment

After watching this video:

- You know that Scrum is the most popular Agile implementation.
- You know all Scrum flow phases.

software
development
academy

House of Scrum

Agile, Scrum

In this video you will learn:

- What is House of Scrum?
- What it consists of?

House of Scrum

Scrum is based on three pillars, and those pillars are:

- **Transparency –** visibility and understanding
- **Inspection -** periodic checking of our work progress
- **Adaptation -** adaptation to possible changes

After watching this video:

- You learned the main assumpions of House of Scrum:
  - Transparency
  - Inspection
  - Adaptation

Scrum Values

Agile, Scrum

In this video you will learn:

- What are the Scrum values?

Scrum Values

Scrum defines five values which are:

- **Commitment**
- **Courage**
- **Focus**
- **Openness**
- **Respect**

After watching this video:

- You learned the five Scrum values.

software
development
academy

Accountabilities in Scrum

Agile, Scrum

In this video you will learn:

- What accountabilities does Scrum anticipate?
- You will learn characteristics of accountabilities :
  - Scrum Master
  - Product Owner
  - Developers

Accountabilities in Scrum

Scrum distinguishes three accountabilities:

- Scrum Master
- Product Owner
- Developers

This three accountabilities combined are called the Scrum Team.

Accountabilities in Scrum

Scrum Team:

- A small group of specialists (usually 10 or less) with one Product Owner, one Scrum Master and Developers.
- Adequate structure, enabling effective work on product development and delivering Increment at Sprint.
- **Cross-functional** and **self-managing.**
- No division into sub - groups/sub - teams.

Developers

Examples of tasks and characteristics of Developers:

- Developing usable Increment of working product each Sprint.
- Highly independent when it comes to choosing how the Increment will be developed.
- Cross-functionality (Developers have all the skills needed) and self-management, high autonomy and needed skills.
- **Developers are responsible for estimating the work that needs to be done.**

Developers

Examples of tasks and characteristics of Developers:

- No division into team roles (everyone in the team is called a developer).
- They are responsible for creating the **Sprint Backlog**, continuously adjusting it to achieve the **Sprint Goal** and ensuring quality through the implementation of the **Definition of Done**.

Product Owner

Examples of tasks and characteristics of a Product Owner:

- **Maximizing the value of the product.**
- Creating and maintaining (eg through prioritization, clearly defining) over the Product Backlog.
- Contacting Stakeholders and understanding their needs.
- Providing answers to questions about developed Product.
- **Creating and communicating a clear, understandable Product Goal.**

Product Owner

Examples of tasks and characteristics of a Product Owner:

- He should have a strong, respected position in the organization.
- **One person** (not a committee!), one Product should have only one Product Owner.
- **The sole owner of the Product Backlog (can delegate part of his responsibilities but still remains accountable).**

Scrum Master

Some examples of tasks and characteristics of Scrum Master:

- Creates an appropriate work environment enabling effective work and implementation of Scrum.
- A *true leader* who serves the organization and the Scrum Team.
- **Responsible for the effective work of the Scrum Team.**
- Facilitates (simplifies) carrying out Scrum events.
- Works closely with the Product Owner (for example helps him manage the Product Backlog effectively, supports cooperation with stakeholders).

Scrum Master

Some examples of tasks and characteristics of Scrum Master:

- Helps with the identification and removal of impediments which are limiting the progress of the Scrum Team (but he doesn't do it himself!)
- Spreading good Scrum practices in the organization (e.g. through training) enabling efficient Scrum implementation.
- Increases the work efficiency of Scrum Teams (e.g. by creating self-managing teams and cross-functionality).

After watching this video:

- You have learned the main accountabilities in Scrum and familiarized themselves with their sample tasks and role chracteristics.

software
development
academy

Artifacts in Scrum

Agile, Scrum

In this video you will learn:

- How to characterize Artifacts in Scrum?
- What is:
  - Product Backlog
  - Sprint Backlog
  - Increment

Artifacts in Scrum

Scrum distinguishes **three** artifacts:

- **Product Backlog** (commitment – Product Goal)
- **Sprint Backlog** (commitment – Sprint Goal)
- **Increment** (commitment– Definition of Done)

Product Backlog

Examples of Product Backlog characteristics:

- **The evolving list of items** we want to produce for the product we are working on, exists as long as the product itself.
- **It is managed (and is responsible for it) by the Product Owner,** although some of its responsibilities may be transferred to Developers.
- It is the **sole source of knowledge** about the developed product.
- **One product = one Product Backlog.**

Product Backlog

Examples of Product Backlog characteristics:

- It is used for effective management of the product manufacturing process.
- **Product Backlog will consist of Product Backlog Items (PBI's)** can be **for example Epics and User Stories** – these are the most common items (elements) inside Product Backlog.
- Necessary artifact to plan iteration.

Product Backlog

User story example:

*As ... **(role)** I want to be able to ... **(benefit, effect).***

As a **website visitor,** I want to **be able to create an account** on this website.

Product Goal

Product Goal enables us to focus on what our product aims to achieve in the long term. Typically, to achieve a Product Goal, you need to spend several Sprints on it.

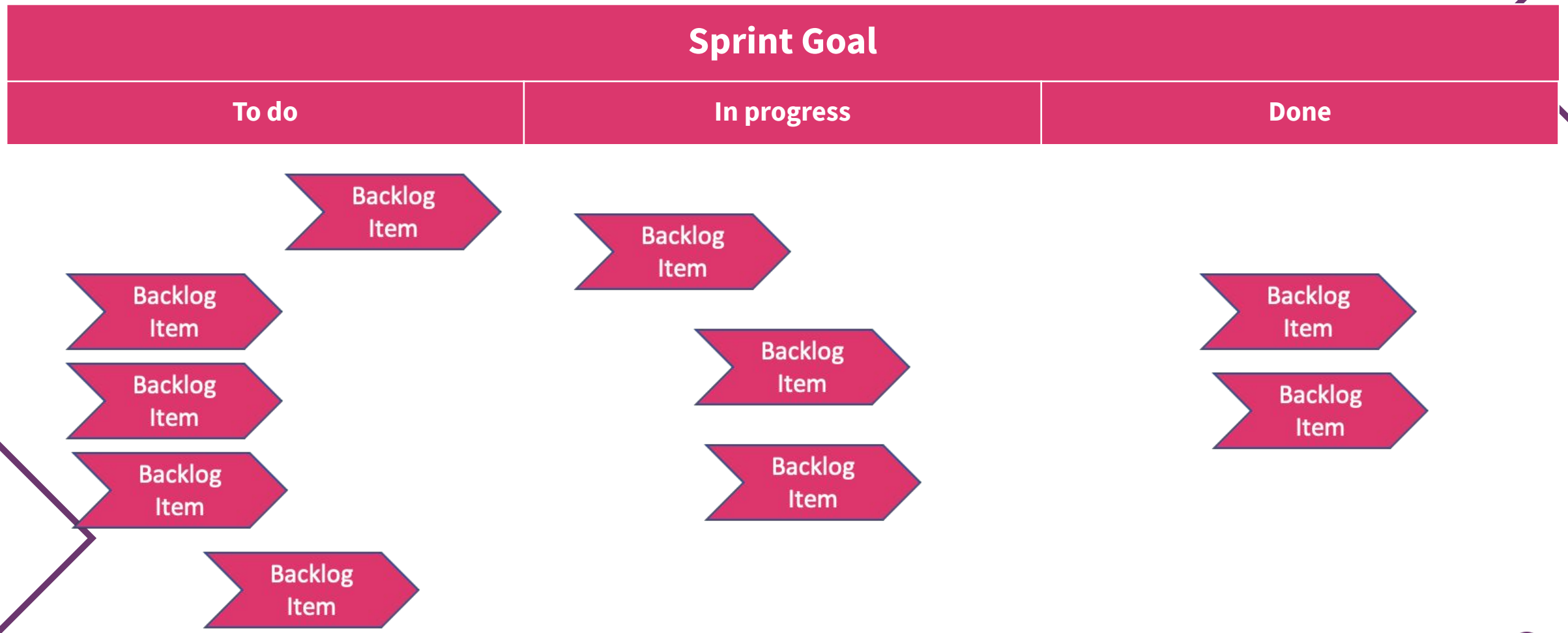
There can be only one Product Goal at the same time - a new Product Goal may be set when the previous one has been fulfilled or a decision has been made to cancel this Product Goal.

Sprint Backlog

Examples of Sprint Backlog characteristics:

- Is the result of Sprint Planning.
- **It contains the work we want to do during the Sprint, a plan how the work will be done, and the Sprint Goal.**
- It contains items that we want to create in the upcoming iteration (Sprint).
- **For example,** it may contain elements such as User Story or technical tasks (development task).

Sprint Backlog (Example)



Sprint Backlog

Examples of Sprint Backlog characteristics:

- The elements contained in it may change during the Sprint (it's more about change regarding the way how to perform the work, for example: when it turns out that we want to use a different tool than assumed).
- Updated on a regular basis with the progress of work and time (usually, during Daily Scrum).
- Sprint Backlog makes easier to observe the progress of the work performed during Sprint.

Sprint Goal

Examples of Sprint Goal characteristics:

- It helps to define what we want to achieve in a given Sprint.
- We gain focus and understanding of what is most important in the ongoing Sprint.
- **Sprint Goal is the commitment of the Developers.**
- Developers decide how to achieve the Sprint Goal (but not the Sprint Goal itself - this is formulated by the Scrum Team!)
- **Sprint Goal can’t be changed,** but the scope of the Sprint Backlog may be refined during the Sprint.

Increment

An Increment is work that has been completed during the Sprint and that complies with the Definition of Done. The increment(s) made in the current iteration are added to the increments from previous iterations.

All Increments in the iteration are presented in the Sprint Review, however, which is very important:

a) More than one Increment may be produced during the iteration.

b) The delivery of an Increment may take place at any time during the Sprint.

Definition of Done

Definition of Done are the requirements that the work performed by us must meet in order to be considered completed. Only then can we conclude that the work has been completed and, after it’s being released, it becomes an Increment.

Definition of Done

Examples of Definition of Done characteristics:

- It often takes the form of a checklist of requirements (e.g. quality, but not only) that must be met as part of the work performed.
- Boosts transparency.
- It allows everyone in the Scrum Team to understand when the work can be considered **DONE.**
- Thanks to Definition of Done, we eliminate the phenomenon of "almost done" work.
- There may be an organization-wide, common Definition of Done (but if it does not exist, the Scrum Team should formulate it themselves).

Definition of Done

The definition of Done is evolving. We should review DoD at least once in a Sprint (at Sprint Retrospective).

After watching this video:

- You learned three Artifacts of Scrum:
  - Product Backlog
  - Sprint Backlog
  - Increment

software
development
academy

Events in Scrum

Events

Agile, Scrum

In this video you will learn:

- What are Scrum events?
- What types of events does Scrum stand out?
- How does Sprint Planning work?

Events in Scrum

Scrum distinguishes the following events (also often called meetings):

- **Sprint**
- **Sprint Planning**
- **Daily Scrum**
- **Sprint Review**
- **Sprint Retrospective**

Backlog Refinement (grooming the Backlog, e.g. by adding details, splitting PBIs into smaller parts) is not an event but an activity that should take place during the Sprint.

Events in Scrum

Those events helps us to:

- Inspect and adapt Scrum artifacts – resigning from any of the events may result in losing the chance for inspection and adaptation.
- Events help us maintain an appropriate pace of work, regularity and reduce complexity (fixed time of meetings, cycle predictability, reduction of the need to organize other meetings).
- Events are also often called meetings.
- Each event in Scrum has a **timebox** - the maximum time in which we should achieve the goal of the event. Events should not exceed the allotted time frame (but may be shorter).


software
development
academy
Get set, ready… Sprint!

Sprint

All work in Scrum takes place in an iteration (repetitive activity) called **Sprint**.

Sprint is a unit of time where all of the mentioned earlier phases of software development (e.g. analysis, coding, testing ...) takes place. During the Sprint, all events in Scrum take place. In the Scrum Guide, sprints are referred to as *Scrum's pulse.*

Sprint characteristics

- **Sprint lasts no more than one month**, it cannot be shortened or extended.
- **Sprints should be of a fixed length** (we shouldn't shuffle their length), **there is no transition period between sprints.**
- Sprint may be cancelled (but this is an extremely rare and very unfavorable activity) when the Sprint Goal is no longer valid - decision is made by Product Owner.

Sprint characteristics

- During Sprint we are developing a working increment of potentially releasable product.
- There is no ”Sprint 0” or ”Release Sprint”.
- Every Sprint should have a clearly defined **Sprint Goal.**
- During the Sprint, we do not make changes that could endanger the achievement of the Sprint Goal.

Sprint planning

Sprint Planning

Sprint Planning is a meeting, as the name suggests, on which we **plan our work for the upcoming Sprint**. It takes place at the beginning of each Sprint.

The entire Scrum Team participates in this meeting.

Product Owner is is responsible for clarifying which elements of the Product Backlog have the highest value (**WHAT**), which of them we want to implement during the next Sprint and whether they are correctly understood, what value we want to deliver in this Sprint, and how they will contribute to the achievement of the Product Goal.

Sprint Planning

Developers are responsible for determining **HOW** the work will be done.

Scrum Master helps (facilitates) with efficiently conducting the meeting.

Sprint Planning

Sprint Planning characteristics:

- **For monthly Sprint**, **timebox is 8 hours** (it should be usually shorter for shorter Sprints).
- People from outside the Scrum Team may be invited to this meeting (as advisors).
- We plan only upcoming Sprint.
- The Scrum Guide distinguishes three topics/points of this meeting (quote from the Scrum Guide):
  ”Why is this Sprint valuable?”
  ”What can be Done this Sprint?”
  ”How will the chosen work get done?”

Sprint Planning

- ***Why is this Sprint valuable?***

The Product Owner describes how value and utility can be added to the product.
Scrum Team collaborates to define **Sprint Goal.**

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

Product Owner together with Developers select the Product Backlog Items (PBIs), Scrum Team adds details the work to be done.
The amount of work that can be done depends on many factors, e.g. team performance, availability, size, etc.

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

Developers (and only them) decide how the work will be done (so that it meets the Definition of Done), usually they decide to divide the PBIs into smaller elements.

Sprint Planning

Sprint Goal + selected Product Backlog Items + plan for delivering them

=

Sprint Backlog

Daily Scrum

Daily Scrum

Daily Scrum is everyday, **15 minutes** event (usually*) of Developers. Timebox is fixed – it doesn't matter how long the Sprint is).

*If the Scrum Master and / or Product Owner have work to do related to the items in Sprint Backlog, they participate in this meeting as Developers.

The main purpose of this meeting is to establish a plan for the next 24 hours, check progress towards the Sprint Goal and possibly react to obstacles that hinders effective work.

Daily Scrum

Daily Scrum characteristics:

- **This is not a status meeting – it's synchronization meeting**, during which we **are analysing the progress of work in this Sprint.**
- Developers makes the decision regarding how this meeting will be organized (they can pick whatever technique they want as long as it's effective).
- To reduce complexity, Daily Scrum should be held in the same place and at the same time everyday during Sprint.
- This is not the only opportunity for Developers to synchronize themselves – it's just to be sure that they have at least one such meeting.
- And no – you don't have to stand during this meeting ☺️

software
development
academy

Sprint review

Sprint Review

Sprint Review is a meeting during which we summarize (and present) what we have managed to accomplish during the Sprint. The entire Scrum Team and interested persons (e.g. stakeholders, Client) are present at this meeting.

Timebox for this meeting is **no longer than 4 hours** (for a monthly Sprint – usually shorter for shorter Sprints).

No standardized agenda - it all depends on internal arrangements.

Sprint Review

Sprint Review characteristics:

- Although we present what we have managed to do, **it should NOT be equated only with the Product Demo.**
- **The goal of this event is to obtain feedback and present plans for upcoming Sprint.**
- Usually, after the completion of the Sprint Review, there are further discussions (e.g. between the Customer and the Product Owner or the Developers).

Sprint Retrospective

Sprint Retrospective

Sprint Retrospective is another meeting during which we summarize our work, but **focusing on how we worked in the previous Sprint and what to do to work more effectively.**

Timebox for this meeting is **no longer than 3 hours (**for a monthly Sprint – usually shorter for shorter Sprints).

Sprint Retrospective

Sprint Retrospective characteristics:

- No standardized agenda – it's Scrum Team internal decision how the event will be organised.
- The goal is to find elements that are interfering with our work or ways to improve our work.
- Each retrospective should end with conclusions and improvements that we want to implement.
- **Usually, only Scrum Team participates in this meeting.**
- Contrary to popular opinion, anyone can conduct this meeting (not only Scrum Master) ☺️

Events Summary

Events summary

Event	Timebox (for one calendar month Sprint)	People involved	Goal
Sprint	Up to one calendar month	Stakeholders and Scrum Team	Effective work, supporting inspection and adaptation, container for other events
Sprint Planning	Up to 8 hours (proportionally shorter for shorter Sprints)	Scrum Team and (**possibly**) invited persons (e.g. specialists/advisors)	Planning the next Sprint, (VALUE, WHAT and HOW), setting the Sprint Goal
Daily Scrum	Every day 15 minutes (regardless of the length of the Sprint)	Developers **(Product Owner and Scrum Master participate if they do work in the Sprint Backlog),** other people are optional (they can listen but not participate)	Plan for the next 24 hours
Sprint Review	Not longer than 4 hours (proportionally shorter for shorter Sprints)	**Scrum Team and stakeholders** (e.g. customers, business owner)	Presentation of the software Increment, obtaining feedback and presentation of the expected scope of work for the next Sprint
Sprint Retrospective	Not longer than 3 hours (proportionally shorter for shorter Sprints)	**ONLY Scrum Team**	How we can improve our work/get more effective?

Events summary

Each and every one of those events serves to implement the pillars on which Scrum is based:

transparency, inspection and adaptation

After watching this video:

- You know the characteristics of all Scrum events:
  - Sprint
  - Sprint Planning
  - Daily Scrum
  - Sprint Review
  - Sprint Retrospective

Backlog Refinement and Estimation

Agile, Scrum

In this video you will learn:

- What is Backlog Refinement?
- What is Estimation?

Backlog Refinement

Worth mentioning: Backlog Refinement very often is mistaken with a meeting. In fact, **it is an activity** that aims to **manage the Product Backlog.**

Manage Product Backlog reffers to activities such as **granulating the elements of the Product Backlog** (e.g. by dividing them into smaller parts), removing unnecessary Product Backlog Items, adding a description or acceptance criteria to Product Backlog Items, etc.

Estimation

Usually:

- **The estimation is done in abstract units (these are not estimates in days or hours).**
- There are various techniques and methods of estimation.
- **The estimation of the work to be done and the size of the Product Backlog elements is made by Developers.**

In this video you learned:

- You know what is Backlog Refinement and what is used for.
- You know what Estimation is.

software
development
academy

Why most of the companies work in Scrum?

Agile, Scrum

Why most of the companies work in Scrum?

Surely, you are able to answer this question yourself already ☺️

However, we will go to a **brief summary** of what you have so far learned about Scrum (and Agile itself). It will also help you understand why most companies decide to implement Agile methodologies.

Why most of the companies work in Scrum?

- Faster delivery time to market.
- Greater readiness for changes and flexibility of the Product.
- Lower product manufacturing cost (most of the time).
- Quick feedback (and decision on further product development or stopping further work).
- Less investment failures and savings for the company.
- Real impact on product implementation and higher morale.

Why is it worth coming to training with Scrum ?

Agile, Scrum