

Agile Softwareentwicklung

or why the classical construct of
specification + fixed price offer has had its day



Classical project procedure in practice

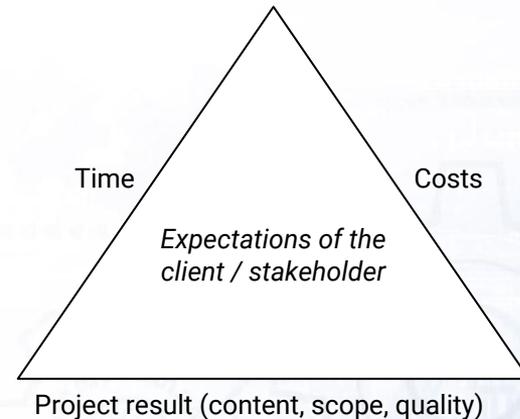
Price, project result and deadline change



Practice has shown for decades:

In software projects, the overall view is that

- despite specifications the project result was not met.
- despite time schedule the project duration was not met.
- despite fixed price the total costs were not met.



The Magic Triangle

Price, project result and date should be fixed

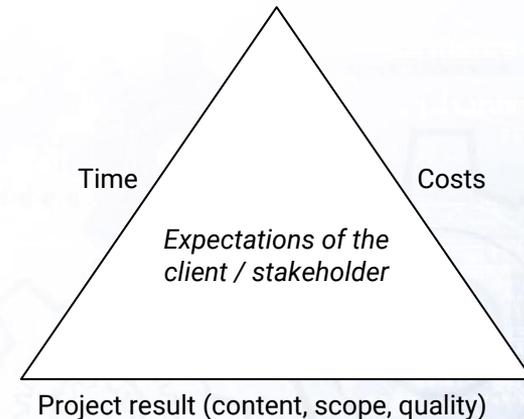


Initial situation:

As a customer, you would like to apply the project management of products (I buy x parts at the price y on the delivery date z) to software development projects.

Classic tools are (e.g. in the waterfall model):

- Scope statement
- schedule
- fixed-price



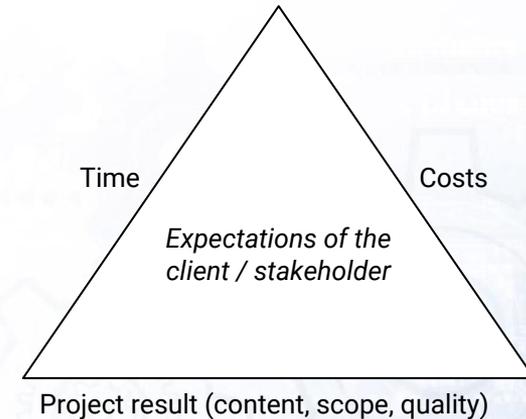
**All participants
are dissatisfied!**



Changed project results

Why is that?

- Digitization and software development is not just an off-the-shelf product
- The requirements in the specifications are quickly overtaken by reality
- The actual users of the software are too poorly integrated



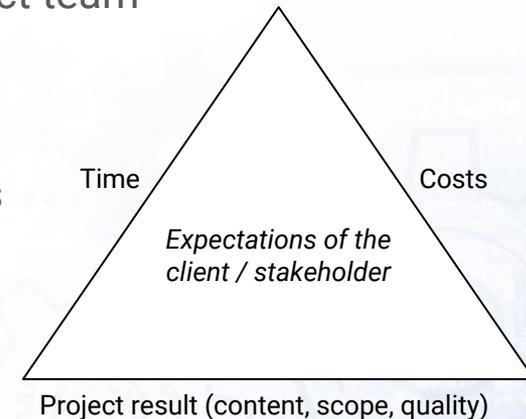
Typical project course

Why are the requirements much too old?

Initial situation: process or deficit is to be optimized by means of digitization / software

- Budget approval process and compilation of project team
- Writing the specifications
- Call for tenders and awarding of projects
- Software development according to specifications

There are often years in between!



Typical project course

Why is there a need for improvement?



Problem: Only at the end does the project get in touch with the end users and thus with "reality".

- Real/helpful feedback by real users can only be generated in a real productive environment, previously only fictitious, theoretical wishes
- Only by operating the prototype can conclusions be drawn about the realistic requirements and needs of the users.



**At the actual end of the project, the software must be usable
(= adapted to reality).**

Typical project course

Why are the projects more expensive than agreed?



The consequence: The project must be made usable via change requests

Change requests require...

- Time ⇒ delay the project
- Money ⇒ increase in budget
- Changes to the specifications ⇒ Other project scope

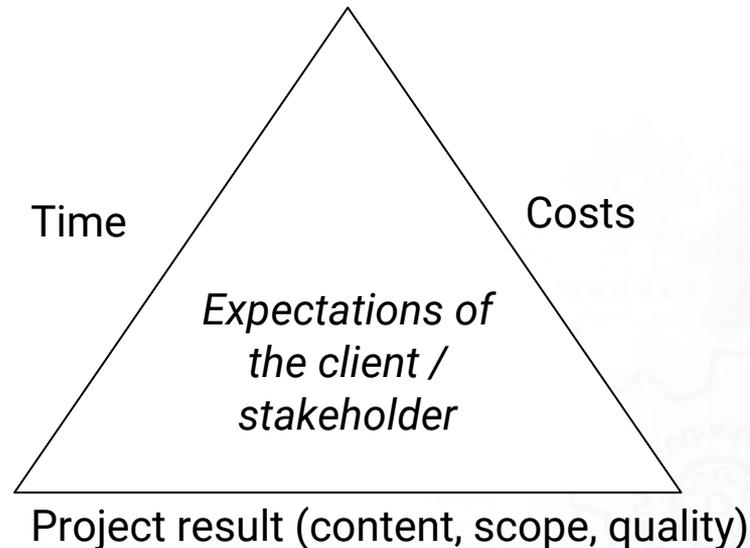


In spite of a fixed price and an exact project schedule, project extension through change requests will be almost inevitable and will make the project significantly more expensive and delays project progress.

Das magische Dreieck

Fazit der klassischen Herangehensweise

Keine der Aspekte des magischen Dreiecks werden bei Festpreis-Projekten eingehalten



Solution:

**The agile Scrum
method**



The Agile Scrum Method

What's that, anyway?



Instead of a detailed requirement specification, there is a fixed product vision with a defined time and cost axis. It is accepted that "content and scope" are not exactly fixed in advance.

The Agile Manifesto puts it in a nutshell:

- **Individuals and interactions** count for more than processes and tools
- **Functioning software** counts for more than comprehensive documentation
- **Cooperation with the customer** counts more than contract negotiation
- **Responding to change** is more important than following a plan

Your advantages: The Agile Scrum method

Always ready to react



Projects are divided into short, two-week cycles:

- Every 2 weeks: new prioritization of requirements and development orders
- Every two weeks: Delivery of executable productive versions to end users

Real advantages:

- Every 2 weeks: real end-user feedback and lessons learned process
- Every 2 weeks: you permanently develop "the most important thing first"
- Every 2 weeks: Software is always up to date (concerning technology, requirements, company structure etc.)



**Project could be terminated at any time,
because it always contains the most essential parts.**

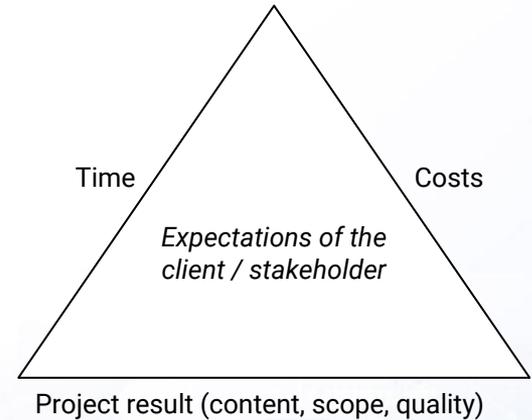
Your result

Win-win situation for your next software project



Thus according to the triangle

- The price is actually fix
"I'm running x sprints for sum y = total budget z."
- The time aspect is indeed fix
"The x sprints for 2 weeks last x times 2 weeks,
after which the project is completed"
- Content and scope are **not** fix,
but guaranteed closer to **the actual customer needs** than with requirement specification + fixed price, since every 2 weeks user feedback is collected and meaningful impact on project can be made easily



**Satisfied users and mature software,
the best possible result for your budget**

Classical project planning vs. Agile method

In Summary



Traditional: fixed Specifications + fixed price

Requirements are 2 years old

The requirement specification more than a year old at the start of the project and even older at the end of the project. Processes, stakeholders, and market requirements change and thus software is already developed to be out-of-date.

Months of development on ancient requirements

- Not until the end of the project one learns what the users really need
- One permanently develops what was considered important 2 years ago by a plan

Too late & too expensive & dissatisfied users

- only at the end the project, the product is made suitable
- ("Change Requests" = more time + more costs)



Neither scope, nor price, nor schedule are actually fixed

Agil: Scrum Framework

Requirements are 2 weeks old

Every 2 weeks

- requirements and development direction re-prioritized
- executable productive version delivered to end users

Short development cycles and current requirements

- Every 2 weeks there is real end user feedback and a "lessons learned" process
- Ability to always develop the most important features first

On time & on budget & satisfied users

- The project can be terminated at any time, if the most important things are already developed and only unimportant features remain to be implemented



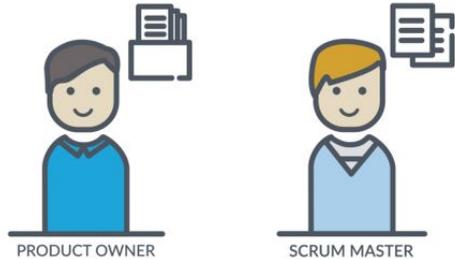
Price and Schedule are fixed and we provide best possible scope



Protect your Interests:

Develop excellent Products with our team at K&K Software AG using state-of-the-art agile methods, now.

Don't I know this from Silicon Valley?



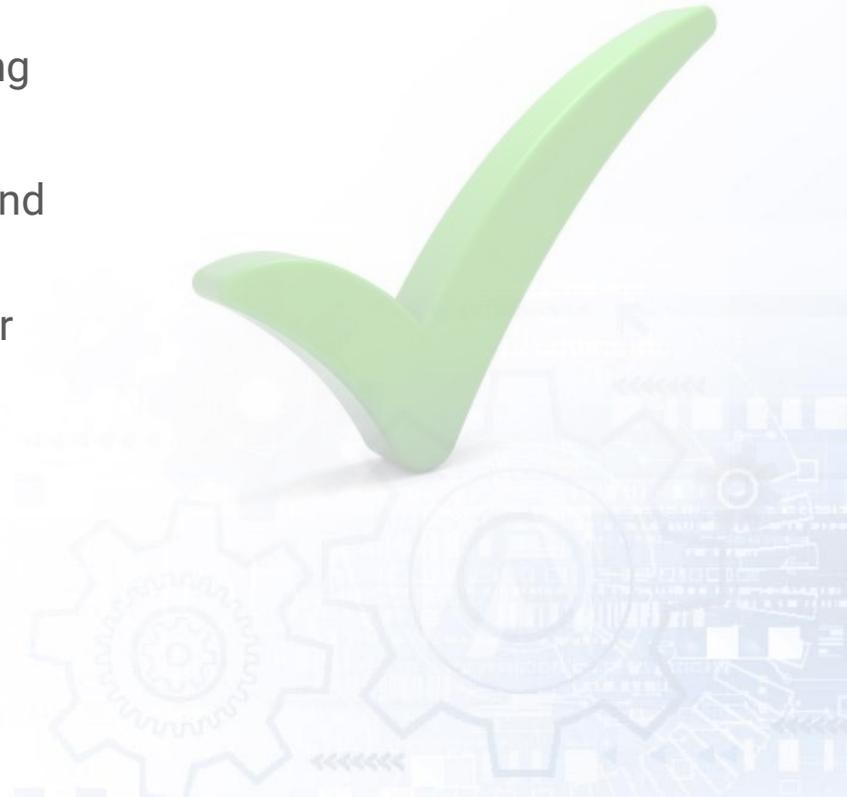
SCRUM TEAM



Scrum is a simple framework to organize work in teams. Today, Scrum is primarily associated with the development of software in young Silicon Valley startups. The disciplined implementation of Scrum can make any team productive.

Benefits of Scrum

- Streamlined communication channels
- High flexibility/agility through adaptive planning
- High effectiveness through self-organization
- High transparency through regular meetings and backlogs
- Timely realisation of new product properties or increments
- Continuous improvement process
- Short-term problem identification
- Low administration and documentation costs



Key Features of the Scrum approach

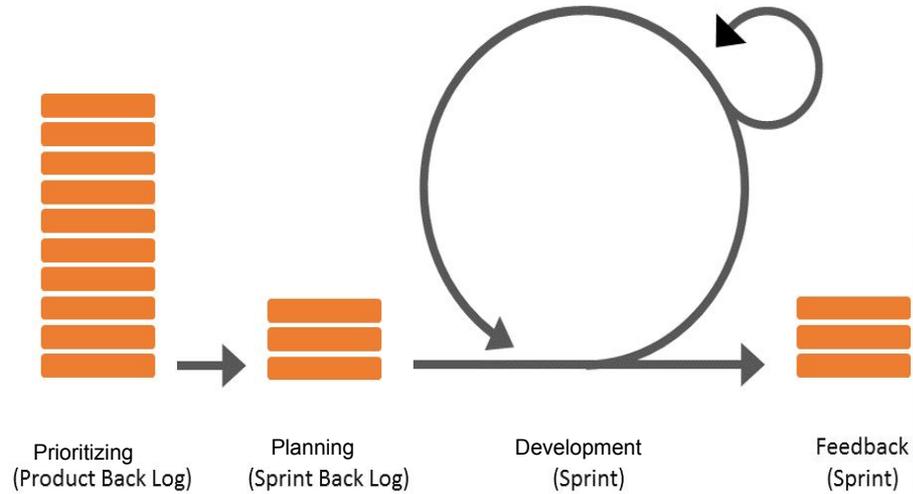


It is essential for Scrum that projects are no longer planned from A to Z, because due to the constantly changing external factors today it is hardly possible to develop a final, all-encompassing plan already at the start of a project.

This has consequences for traditional project management: In the traditional way, project goals are often only insufficiently achieved, in some cases hardly achieved at all.

In contrast, Scrum follows an agile approach. Instead of a final plan, it initially formulates a product vision based on user needs and market requirements. These will be defined step by step in the course of the project. The development does not take place in a straight line, but in cyclical steps. At the start of the project, the user needs are summed up in so-called "user stories" by means of one or two concise sentences.

Der Scrum Prozess



Roles within Scrum

Scrum manages with only relatively few roles and rules and is therefore exemplary in the field of agile project management.

Since there are also only a manageable number of artifacts and activities, the two most important "promises" for agile project management can be kept: to keep the complexity of projects as low as possible and to guarantee a high degree of flexibility at the same time.

The three central roles in Scrum are **Product Owner**, **Scrum Master**, and **Development Team**.

Stakeholders



Stakeholders are persons or groups of persons with a particular interest in the outcome of a (business) process. As internal and external stakeholders, they are either directly or at least indirectly affected by the respective entrepreneurial activities. Classically, stakeholders include owners, employees, shareholders, customers, partners, dealers and suppliers.

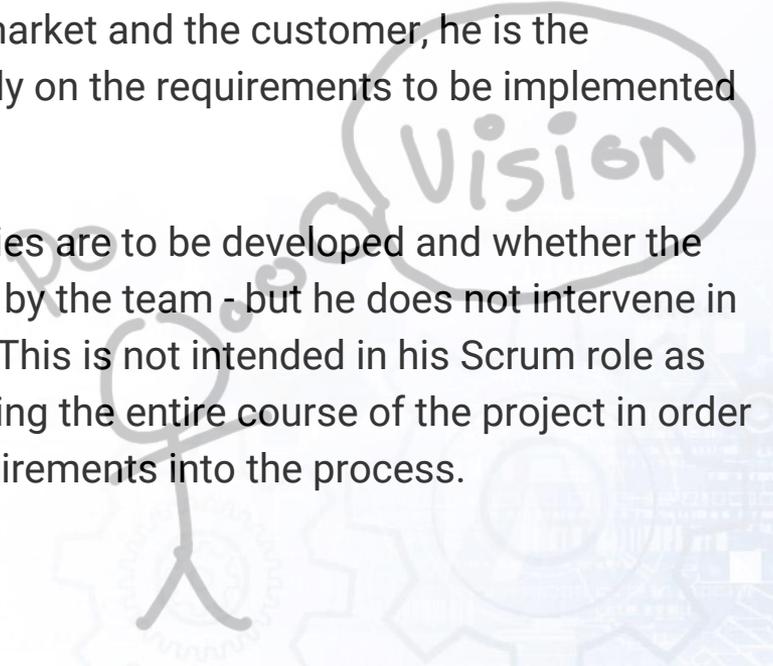
In Scrum, one of the primary tasks of the product owner is to "implement" this market or user view into product development. In a way, he acts as a representative and advocate of the stakeholders and their requirements for the product to be developed.



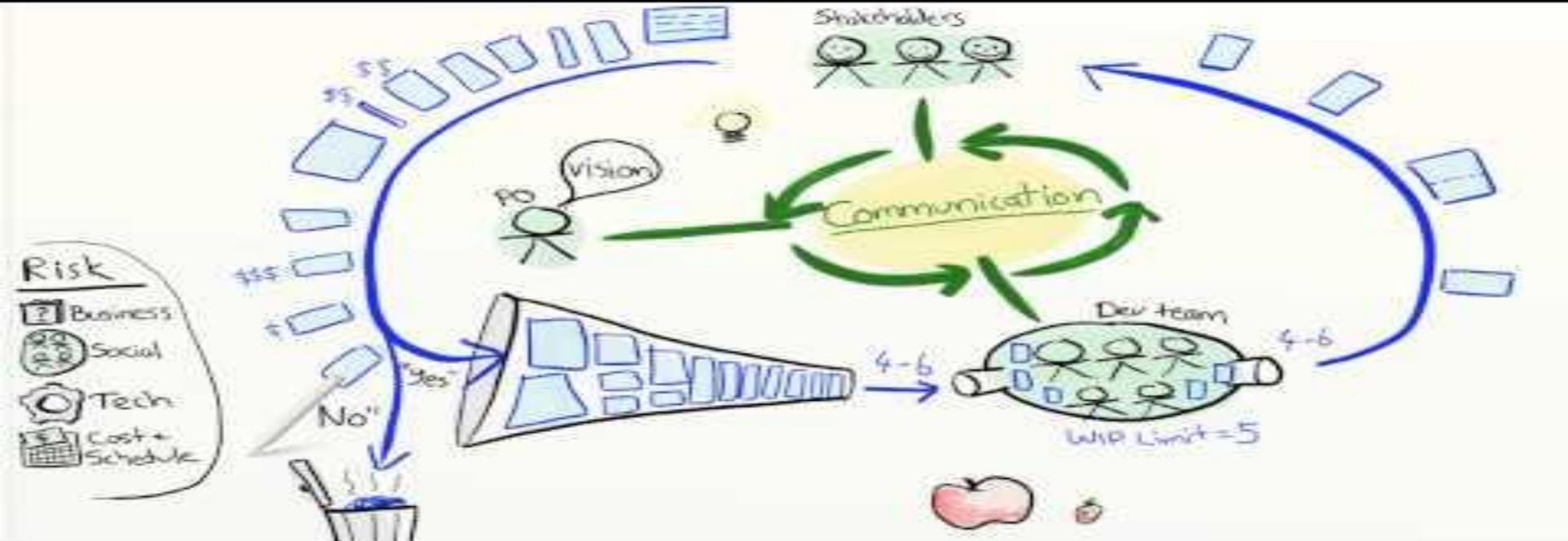
The Product Owner

The task of the product owner is to know the interests of the users and stakeholders (user stories) exactly and to represent them consistently. To this end, the product owner views the project strictly from their perspective. As a link to the market and the customer, he is the "godfather of success" insofar as he focuses exclusively on the requirements to be implemented in development.

Thus he decides which product features or functionalities are to be developed and whether the respective requirements have been completely fulfilled by the team - but he does not intervene in the development process during the individual sprints. This is not intended in his Scrum role as Product Owner. Nevertheless, he must be available during the entire course of the project in order to be able to flexibly incorporate current customer requirements into the process.

A faint, light blue background illustration. It shows a stylized human figure with a large thought bubble above its head. Inside the thought bubble, the word 'Vision' is written in a handwritten, grey font. The figure is surrounded by various icons and symbols, including gears, a lightbulb, and a circuit board, suggesting a process of thinking and innovation.

Excellent explanation from product owner's point of view (PO roll is typically taken over by the customer): <https://www.youtube.com/watch?v=502ILHjX9EE>



The Scrum Master

The Scrum Master acts as moderator and service provider in the Scrum process and organizes the communication of the development team with the "outside world". It is his responsibility to monitor compliance with the values and rules of a project and to create suitable framework conditions for a successful course of the project. This includes, for example, providing the necessary resources, removing any obstacles, and mediating between the product owner and the development team.

The role of the Scrum Master is therefore primarily aimed at enabling the team to work undisturbed within the requirements framework set by Product Backlog and Sprint Backlog. As a neutral service provider, he is not "in front" of anybody and does not make any content decisions.



The development team

The team in Scrum usually consists of five to ten project members and is interdisciplinary. Specific hierarchies between the individual areas of competence are not planned for the time being. During the sprints, the team organizes itself and implements the required new product increments on its own responsibility.

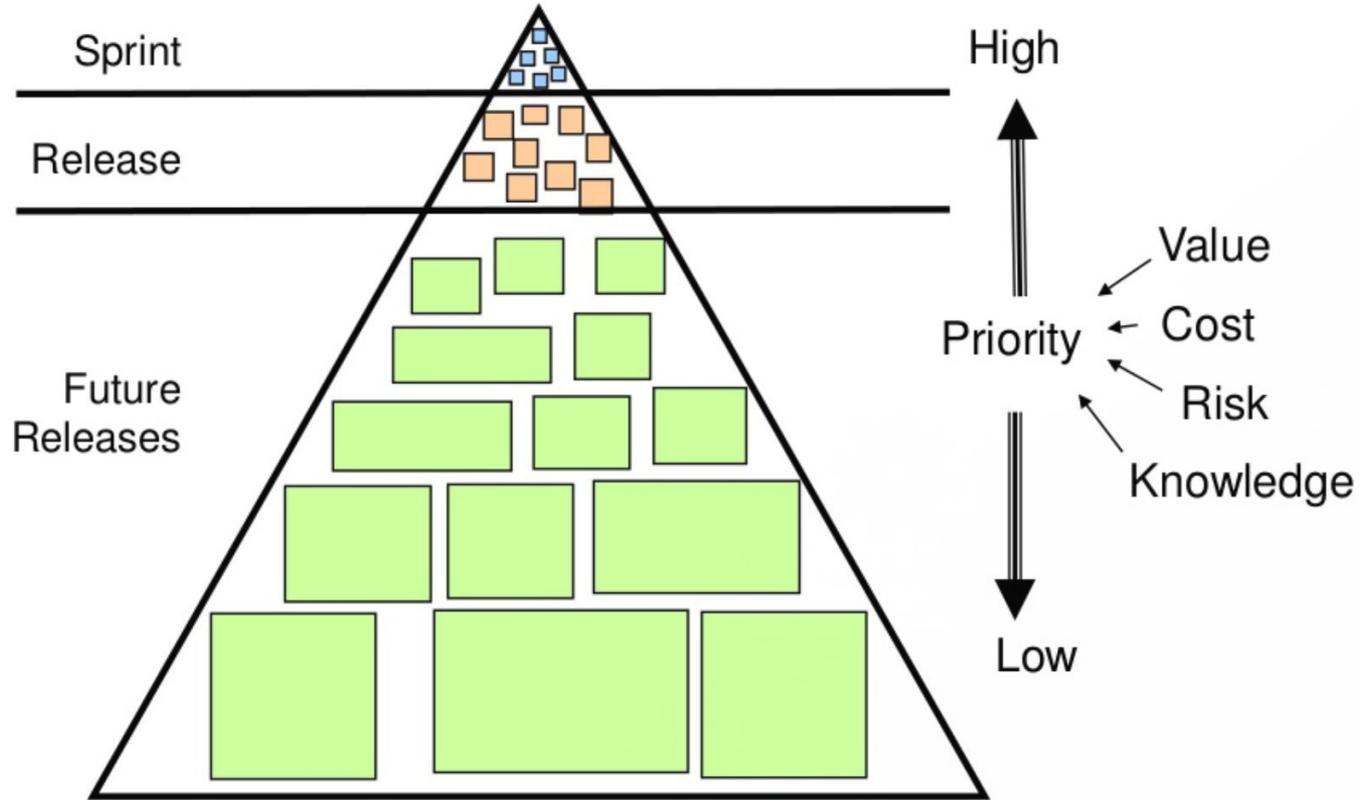
The aim is to create a creative atmosphere which is largely free of disturbing external influences and which promotes the skills of each team member and thus contributes to achieving optimum results in a joint effort.



The Product Backlog in Scrum

The Product Backlog is the link between the Product Owner and the team in Scrum. The product owner creates a prioritized list of requirements/requirements for the product to be developed according to criteria such as economic benefit, effort, risk and necessity. The basis for this is formed by the user stories (short requirement formulated in everyday language) and epics (requirement description). This "to-do list", generated from the user's point of view, is the result of release planning, reflects the product vision and is continuously refined as the project progresses.

The product backlog in Scrum is therefore not a final plan set for all times at project start as in classical project management, but can be extended by new requirements at any time. By being dynamic and open to change, it fulfills one of the essential characteristics of agile project management.



Scrum Meetings



- Sprint Planning Meeting
- Daily Scrum Meeting
- Sprint Review Meeting
- Sprint Retrospective Meeting



Sprint Planning Meeting



In Sprint Planning, the implementation of the next product increment is planned - the meeting takes place every few weeks between the sprints. It is divided into two parts:

In a first step, the requirements analysis, the product owner presents a pre-selection of prioritized requirements from the product backlog and answers questions from the development team.

In the second step, the team then decides which requirements can actually be implemented in the next step. This is then broken down into individual tasks and transferred to the Sprint Backlog.

Daily Scrum Meeting

The Daily Scrum is a morning meeting in Scrum that takes place at a fixed time and should not last longer than a quarter of an hour. The participants are Scrum Master, Product Owner (ideally) and Developer. The overall purpose of this inspection and adaptation meeting is to support and facilitate the exchange of information and self-organization of the team. The team members answer three questions:

- What progress have I made since the last meeting?
- What were the challenges and obstacles?
- What to-do's have I set out to do today?

If it turns out that a developer does not make any progress with the tasks taken on due to a problem that cannot be solved by the team itself, the Scrum Master will take care of the problem after the meeting.

Sprint Review Meeting

A 3D illustration of several stylized human figures in white and grey. One figure in the foreground is holding a large banner that says 'GOAL' in white capital letters on a dark background. Other figures are running towards the banner, and a ribbon is draped across the scene. The background is a light blue gradient with faint, larger-scale versions of the same illustration.

Each Sprint in Scrum ends with a Sprint Review, during which the development team delivers the new product increment to the Product Owner. The purpose of this meeting is to determine whether the respective sprint target has been achieved or whether the requirement has been met.

If this is the case, the product is accepted by the product owner. The so-called Definition of Done serves as an objective basis for assessment, on the basis of which the product owner decides whether the requirement has been implemented completely and without errors.

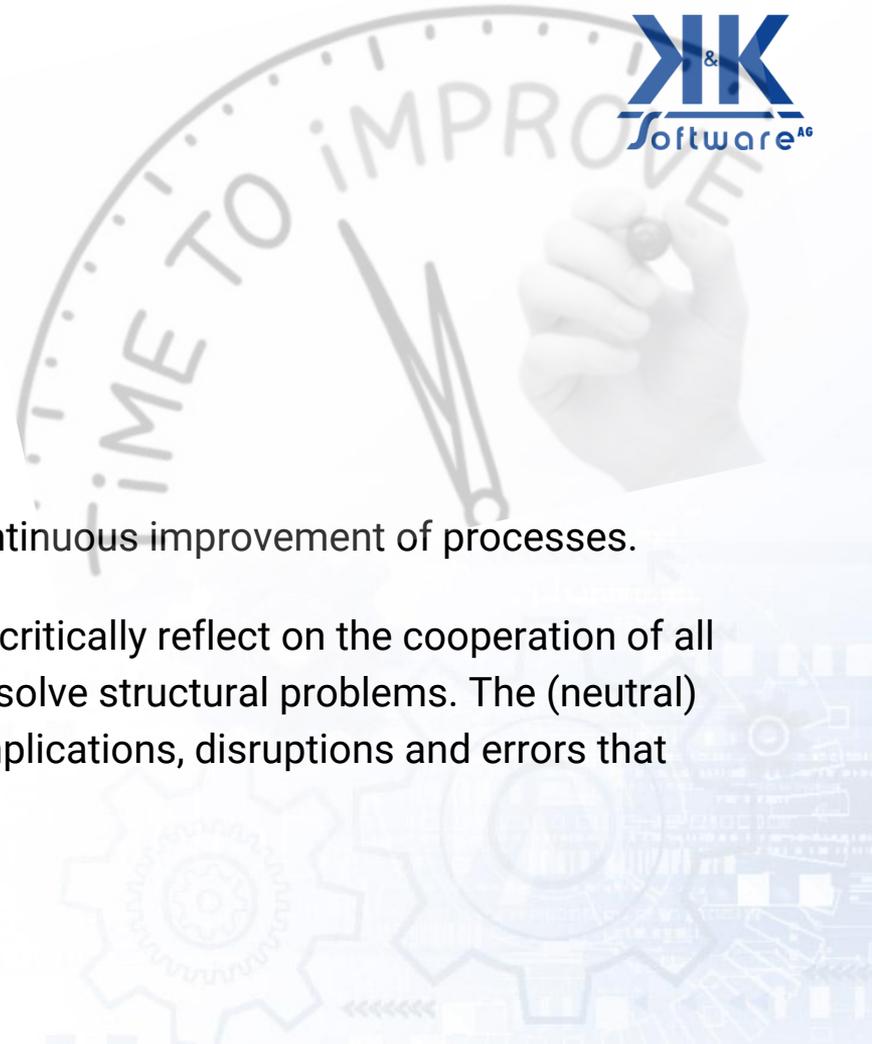
If the sprint target is demonstrably missed, the product owner refuses acceptance and the requirement is deemed not completed.

Sprint Retrospective Meeting



The goal of the Sprint Retrospective Meeting is the continuous improvement of processes.

The meeting takes place after the sprint and serves to critically reflect on the cooperation of all persons involved in the project in order to identify and solve structural problems. The (neutral) moderation by the Scrum Master ensures that the complications, disruptions and errors that occurred during the sprint can be addressed openly.



Any questions? Please contact us!



We are committed to your request and will be happy to answer detailed questions promptly. We look forward to your inquiry.

I am your contact person:

Laura Köpl

Head of Marketing and Sales

Phone: 09382 / 3102-241

Mail: koepl@kk-software.de





Acting future-proof

Your K&K Software AG as a reliable partner.