# Agile Project Management

Agile project management is an iterative approach to managing software development projects that focuses on continuous releases and incorporating customer feedback with every iteration. The advantage of agile project management is that unexpected changes can be fixed faster because the planning is more adaptive.

Advantages:

* Less planning / control effort necessary
* Reduced risks
* Increased flexibility
* Customer satisfaction
* Better control

Disadvantages:

* Limited documentation
* Difficult measurement
* No finite end

## Scrum

The Scrum framework recognizes three leadership responsibilities: Product Owner, Developer and Scrum Master. The totality of these responsibilities is called a Scrum team. A Scrum team enters into contact with the participants, the so-called stakeholders. Progress and intermediate results are transparent to all stakeholders. Stakeholders are allowed to listen in on most events.

Advantages:

* teams complete project deliverables quickly and efficiently
* ensures effective use of time and money
* large projects are divided into easily manageable sprints
* works well for fast-moving development projects

Disadvantages:

* scope creep, due to the lack of a definite end-date
* chances of project failure are high if individuals aren't very committed or cooperative

### Product Owner

The product owner is responsible for the characteristics and commercial success of the product. He designs the product with the aim of maximizing its benefits. The benefit could be based on the company's sales, for example. He creates, prioritizes, and explains the product features to be developed, and he judges which features have been completed at the end of a sprint. He is a person, not a committee. He alone is responsible for deciding on the product, its features, and the order of implementation. Thus, it balances features, delivery dates, and costs.

Every implementation step has to be written in the Backlog. The Product Owner needs to update and manage the Product Backlog.

### Developer

The developers are responsible for delivering the product functionalities in the order requested by the product owner. They are also responsible for compliance with the agreed quality standards. The Scrum team organizes itself. It does not allow anyone, not even the Scrum Master, to dictate how it must implement backlog entries.

A Scrum team should be able to achieve the goal of a respective sprint without major external dependencies. Therefore, an interdisciplinary composition of the Scrum team is important, e.g. with architect, developer, tester, documentation expert and database expert. Good and bad results are never attributed to individual team members, but always to the Scrum team as a unit. The ideal team member is both a specialist and a generalist so that he or she can help teammates achieve the common goal.

### Scrum Master

The Scrum Master is responsible for ensuring that Scrum succeeds as a framework. To this end, he works together with the development team, but is not part of it himself. He introduces the Scrum rules, checks their compliance and takes care of the elimination of disruptions and obstacles. These include lack of communication and cooperation as well as personal conflicts in the development team, which should be effectively countered with healthy and clear communication, disruptions in the cooperation between the product owner and the development team as well as disruptions from outside, for example requests from the business department to work on additional tasks during a sprint. The Scrum Master moderates the Sprint Retrospective and often also the Sprint Planning and Backlog Refinement.

### Product Backlog

The Product Backlog is an ordered list of the requirements for the product. The Product Backlog is dynamic and constantly evolving. All work done by the development team must originate in the Product Backlog. The product owner is responsible for maintaining the product backlog. He is responsible for the order or prioritization of the entries.

### Scrum Process

A Sprint is a work stage in which an increment of a product functionality is implemented. It begins with Sprint Planning and ends with Sprint Review and Retrospective. Sprints follow each other immediately. During a Sprint, no changes are allowed that affect the Sprint goal.

A sprint covers a time window of one to four weeks. Ideally, all sprints should have the same length to give the project a beat. A sprint is never extended - it is over when the time is up.

A Sprint can be terminated prematurely by the Product Owner if the Sprint goal is no longer to be achieved, for example, because the targets set by stakeholders or general market conditions change. In this case, the current Sprint is ended with a Sprint Retrospective and the new Sprint is started normally with Sprint Planning. The Scrum Guide describes Sprint terminations as resource-intensive and unusual.

In Scrum, we speak of events instead of meetings to make it clear that we are talking about work. All Scrum events have fixed time windows (time boxes) that should not be exceeded.

Sprint Planning answers three questions:

* Why is the upcoming sprint important?
* What do we need to do to achieve the sprint goal?
* How will the resulting work be done in the upcoming Sprint?

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

### Daily Scrum

At the beginning of each working day, the development team meets for a Daily Scrum of max. 15 minutes, during which the Scrum Master and Product Owner are often present, but not actively involved, if they are not working on backlog elements themselves. The purpose of the Daily Scrum is to exchange information. No problems are solved in the Daily Scrum - it is rather about getting an overview of the current status of the work. For this purpose, it has proven useful for each team member to use the task board to say what they have achieved since the last Daily Scrum, what they would like to achieve by the next Daily Scrum, and what is in the way.

During Daily Scrum, it may become obvious that the completion of a task takes longer than planned. In this case, it makes sense to divide the task into smaller tasks that can then be taken over by other members of the development team.

If questions arise during Daily Scrum that cannot be answered within the strict 15-minute time limit, they are either noted down and handed over to the Scrum Master, or their answer is postponed to a later meeting, often directly afterwards.

### Sprint Backlog

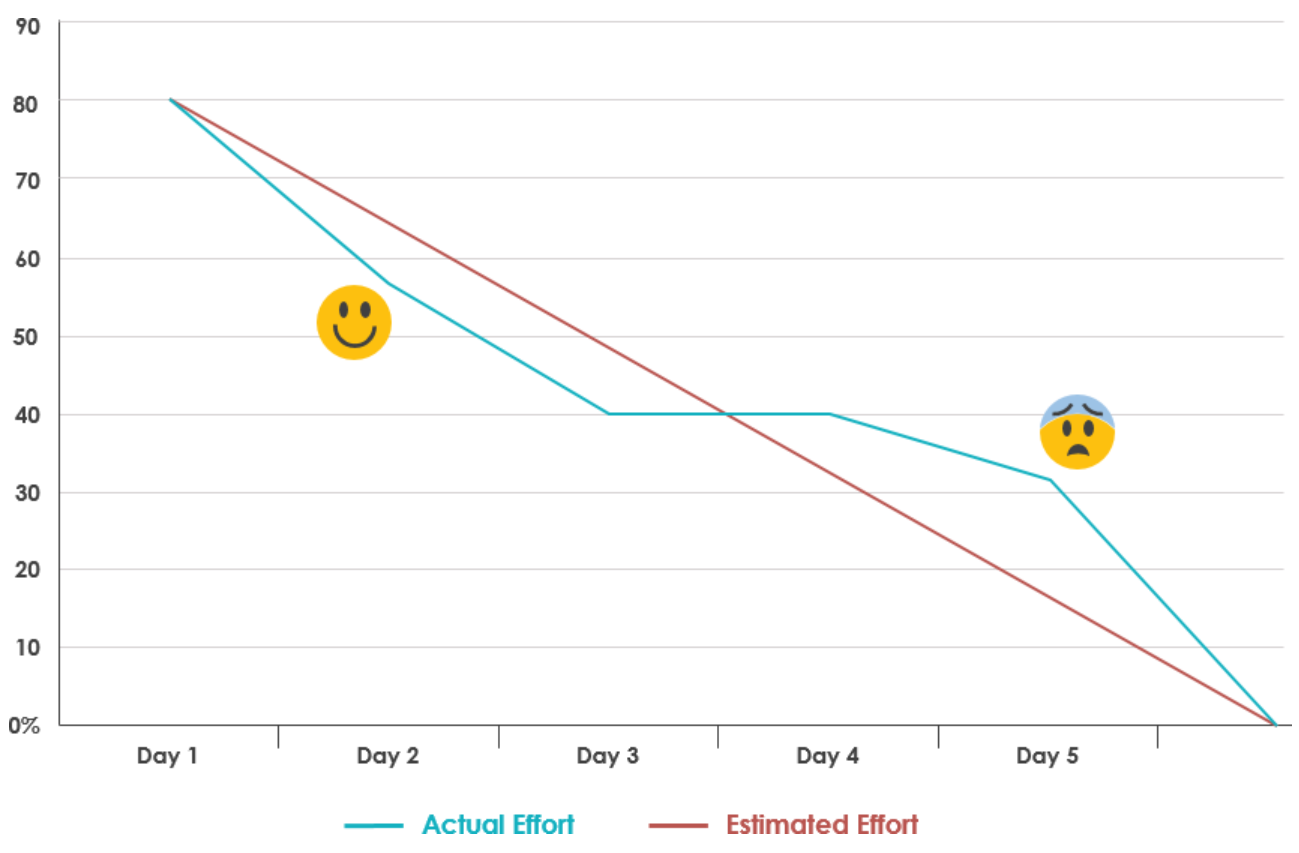
The Sprint Backlog is the current plan of tasks to be completed for a Sprint. It includes the product backlog items that have been selected for the sprint and the tasks required for them (e.g. development, testing, documentation). The Sprint Backlog is continuously updated by the team members after the completion of a (partial) task. This serves as an overview of the current processing status. A task board is often used to make it visible to everyone.

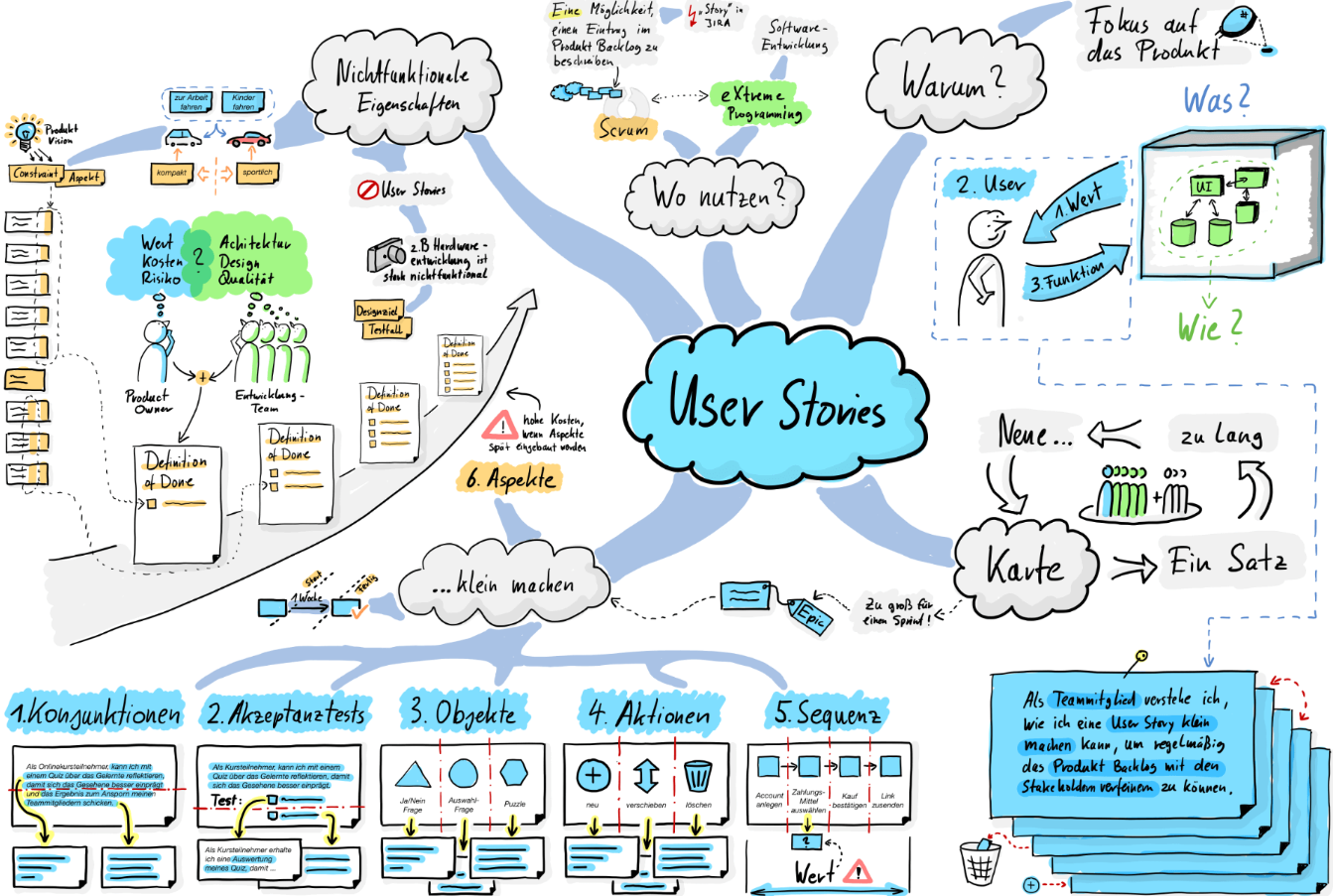
Product Increment

The increment is the sum of all Product Backlog entries completed during the current and all previous Sprints. At the end of a Sprint, the new Increment must be in a usable state and meet the Definition of Done.

### Burn down chart

A burn down chart is a graphical representation of work left to do versus time. Burn down charts can be applied to any project containing measurable progress over time.





## Kanban

Kanban reduces costs and time; it lets you predict your schedule and staffing data you need to deliver that value on time and budget. Kanban offers a high adjustment potential in case of short-term changes in demand, which is the key difference between agile project management and traditional project management. With the running out of a required pre-product, the order for post-production is triggered promptly. Here, information is always forwarded up to date and thus adapted to the current demand situation from the consumer to the producer or to the supplier. This drastically reduces unnecessary inventories and significantly shortens throughput times.

The aim of Kanban systems is to reduce inventories and thus reduce capital commitment and increase flexibility with regard to changing demand quantities. This goal is to be achieved without loss of delivery readiness, without deterioration of scrap rates, without increase of rework, additional transports etc. All this is to be achieved with strongly reduced planning efforts. All this is to be achieved with greatly reduced planning effort.

The frequently observed improvements in delivery readiness, scrap rates, rework, etc. are secondary effects.

Advantages:

* More motivation
* Less planning
* flexibility

Disadvantages:

* Standardization is necessary
* individual and special products are nonsensical with Kanban

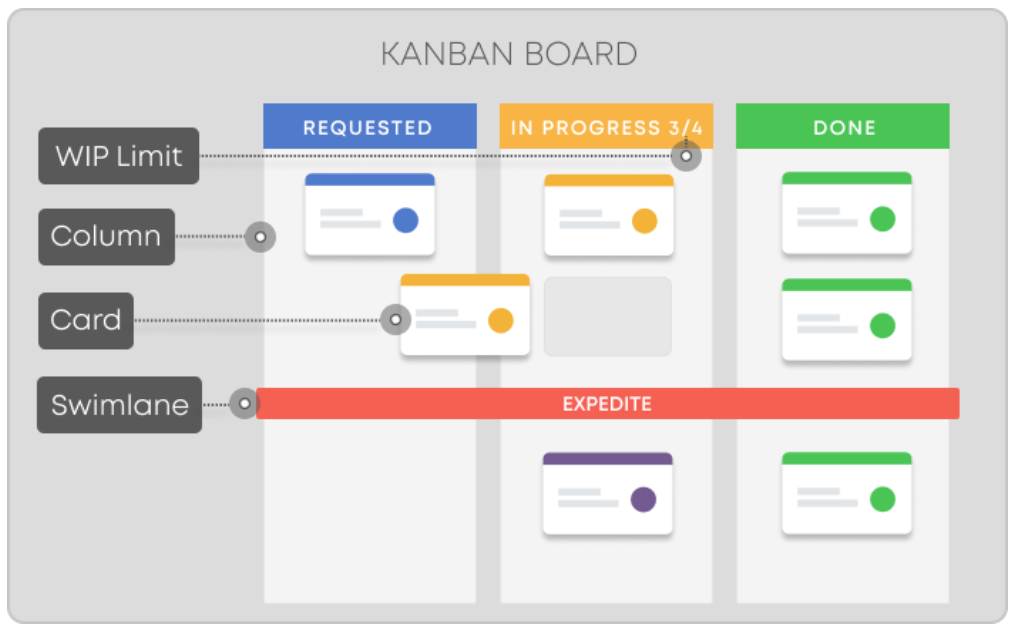
### Kanban Board

Kanban boards use cards, columns, swimlanes, and WIP limits to enable teams to effectively visualize and manage their workflows

* **Kanban Columns** - Each column on the board represents a different phase in your workflow. The cards move through the workflow until they are completely finished.
* **Work-in-Progress Limits** - You limit the maximum number of tasks in the different phases of the workflow. Limiting WIP allows you to complete work items faster by helping your team focus solely on current tasks.
* **Kanban Swimlanes** - Horizontal lanes that you can use to separate different activities, teams, service classes, etc.
* **Commitment Point** - A commitment marks a point in the work process where a work task is ready to be pulled into the system.
* **Delivery Point** - The point in the workflow at which work tasks are considered complete.

Examples:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

### Kanban Ticket

A Kanban ticket is on entity in the kanban board. It has these possibilities of different states:

* **New requests**: Here the customers have submitted their requests where it is then integrated into the support team’s Kanban board and workflow.
* **Request feedback**: When a customer submits a request to the support team’s Kanban board, the support team responds to the customer’s request.
* **In progress**: When a customer ticket reaches the “In Progress” column, this means that it has been assigned and passed on to a team member for resolution.
* **Additional support**: If the ticket requires additional resources or support, it may be moved into this column. Who provides additional support to these requests will be determined by the team’s established workflow and practices.
* **Waiting on the customer**: When a customer is yet to provide feedback, the support team places a card here while they wait for a response. When they resume working on the request, they place the card back in the “In Progress” column.
* **Done**: Moving a ticket to the “Done” column indicates that a task has been completed.