Agile Project Management

Prof. Dr.-Ing. Michael Arndt
THM

Content

- Introduction
- Review Project Management
- Managing Agile Projects
- What to think about...

Lecturer: Michael Arndt
Duration: 1,5 h
Recorded: Yes

Who is this lecture for?

This lecture is for students who have done an introductory lecture on project management (have learned the basics).

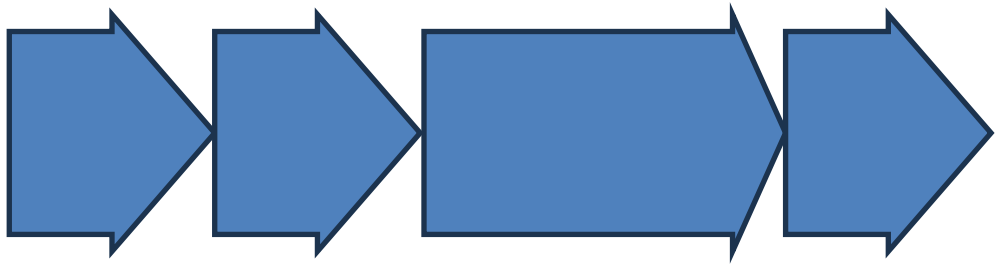
They know want to expand their knowledge and use this knowledge in a real world project under difficult circumstances (partially online, partially in presence, short time, different cultures, new topic, ...)

I assume, you know...

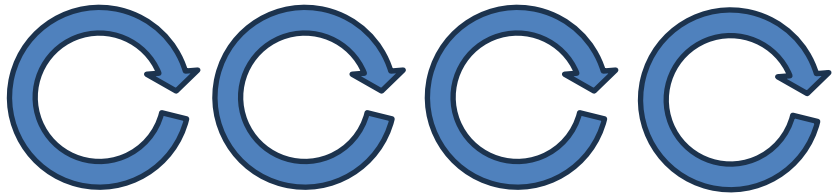
... what a project is and how it can be characterised.
... different types of projects.
... the relation between budget, quality and time within a project.
... the development process of a technical product and how it relates to a development R&D project.
... the phases of a linear project and project management tasks during the phases.
... the tasks in project initiation, -planning, -execution and -closure.
... requirements engineering and the resulting technical specification of a product.
... risk management in projects.
... which roles are involved in typical projects and what these roles do.
... how teams work and what are typical risks of teamwork (and how you can avoid them).

Process Models for Projects

There exist two fundamental process models for projects:

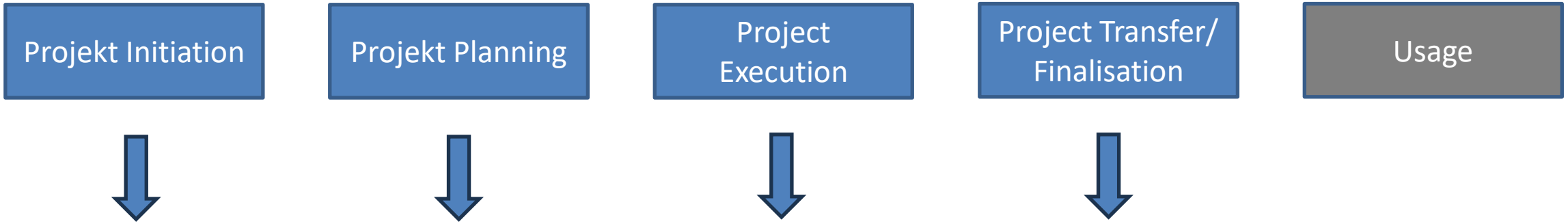


Linear Process



Cyclic Process (Agile)

Linear Process - Project Phases



Project Idea
Targets

Work Packages
Schedule
Resources
Budget

Results

Project Documentation
Lessons Learned
Results Transfer

Main Artefacts of a Project

- Project Contract (Motivation, Targets, expected Results, Milestone Dates, initial Budget etc.)

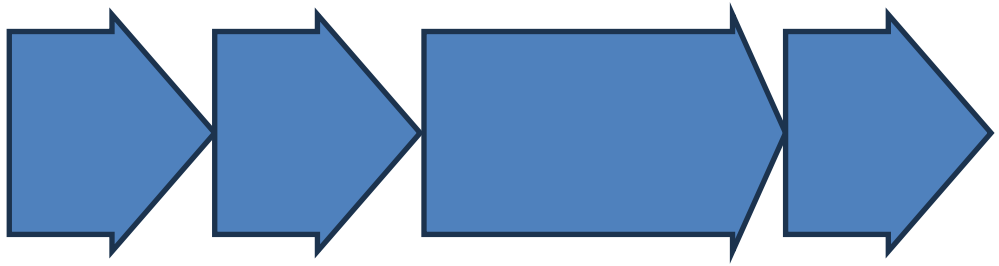
- Project Plan
  - Schedule (WBS, Detailed Schedule)
  - Resource Plan
  - Budget Plan
  - Results Specification
  - Team

- Documentation of Project Execution (Protocols, OPL, Schedules, etc.)

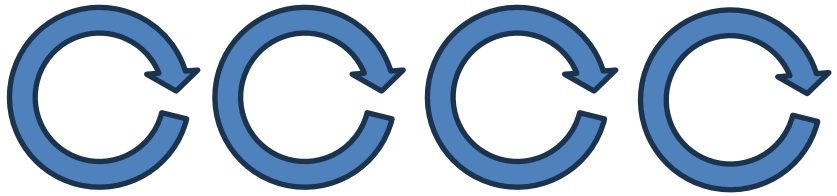
- Project Results (Documentation, Physical Objects, Test Results, etc.)

Process Models for Projects

There exist two fundamental process models for projects:

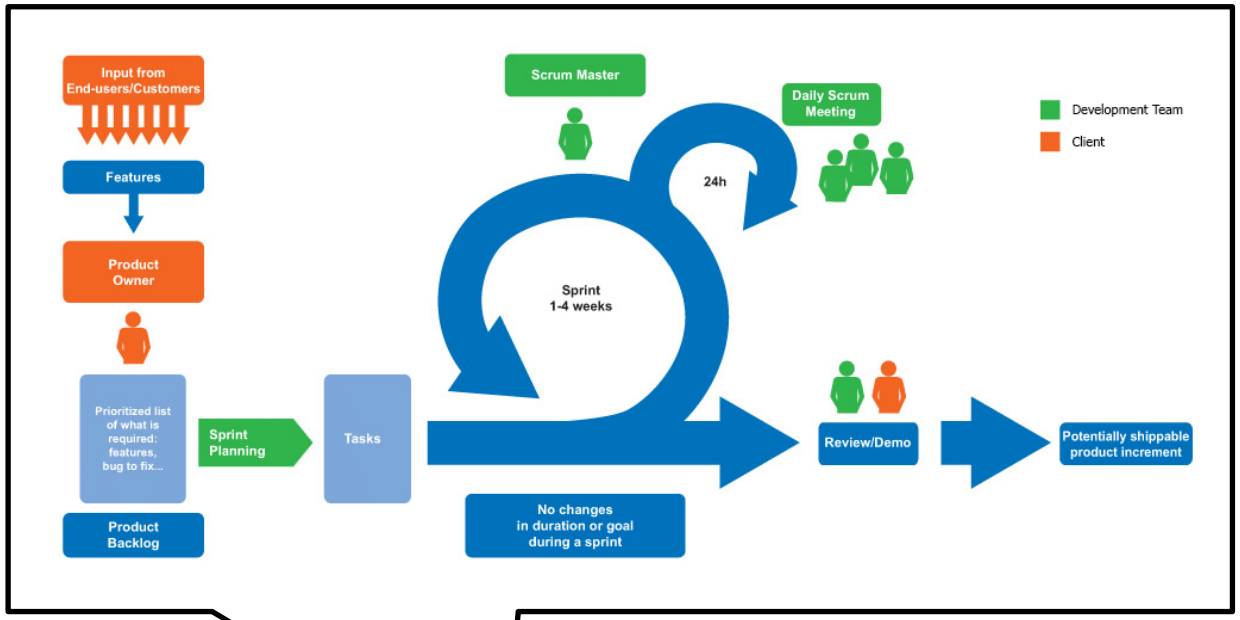


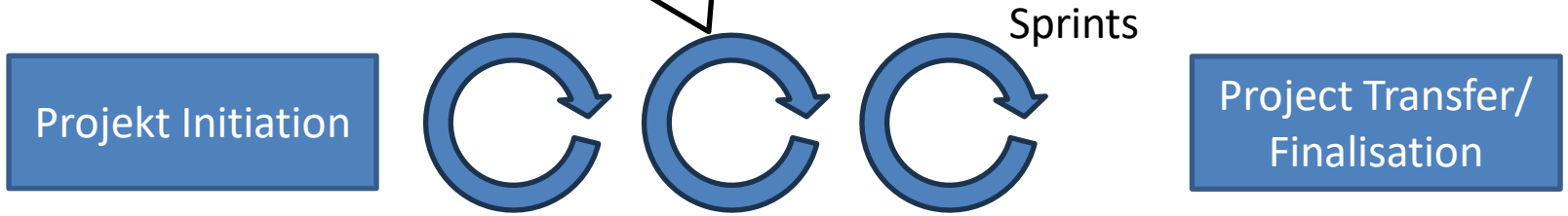
Linear Process



Cyclic Process
(Agile)

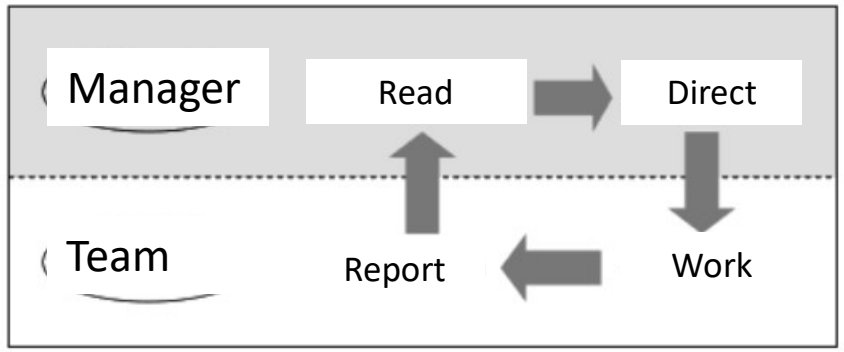
Cyclic Process





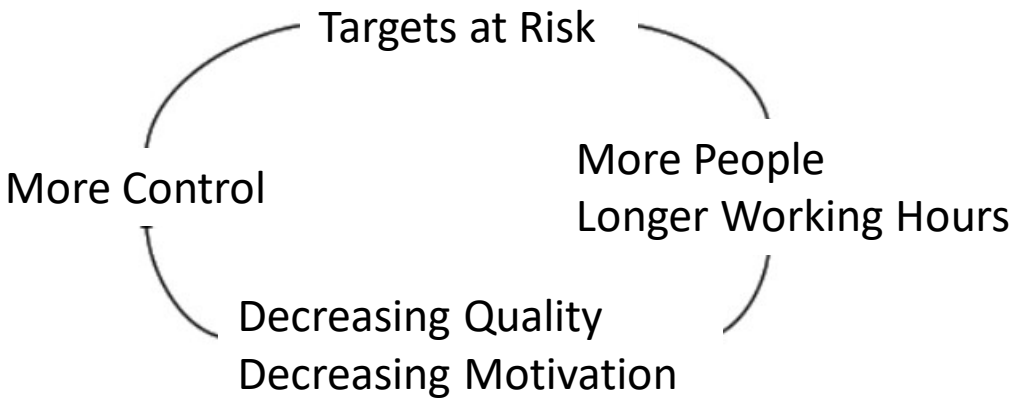
Agile Working in Projects

Risks of traditional project work



Slow!

Risk of inefficiency



Risk of Demotivation

2002

Manifesto for Agile Software Development

We're opening up better ways to build software, by doing it ourselves and helping others to do it.
Through this activity, we have come to appreciate these values:

- Individuals and interactions are more than processes and tools
- Working software is more than comprehensive documentation
- Cooperation with the customer is more than contract negotiation
- Responding to change is more than following a plan

That is, although we find the values on the right important,
we estimate the values on the left side to be higher.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

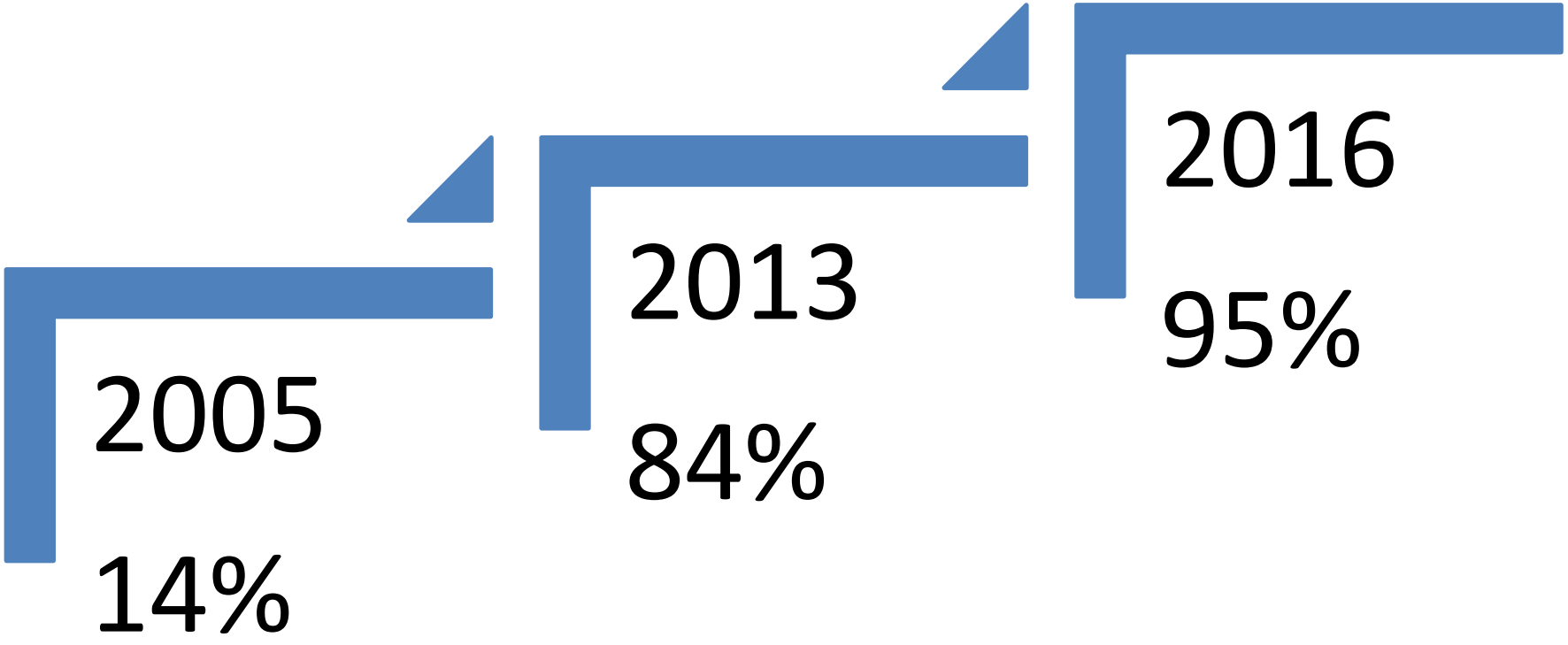
James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

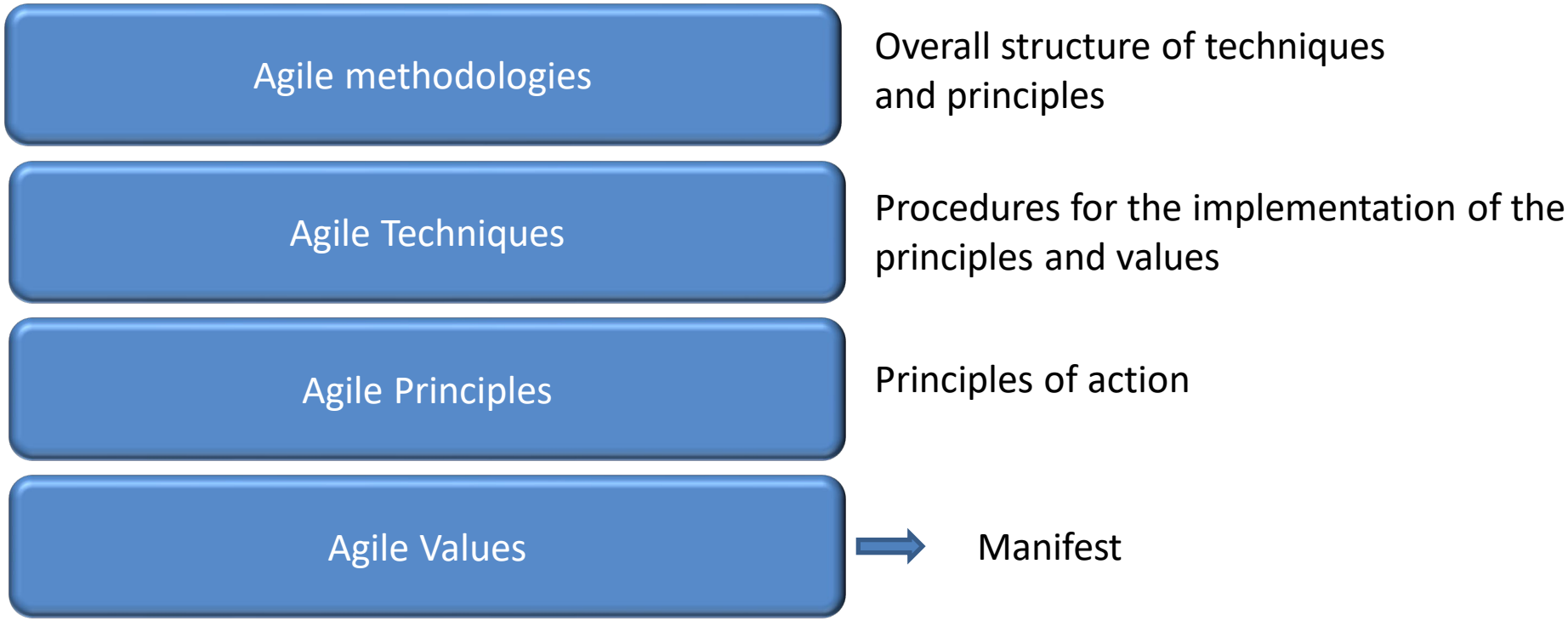
History

- The first approaches were found as early as the early 1990s.
- Agile software development first gained popularity in 1999 when Kent Beck and others published the first book on extreme programming. The interest in extreme programming paved the way for other agile processes and methodologies as well.
- The term agile for this type of software development was chosen at a meeting in Utah in February 2001. At this meeting, the Agile Manifesto was also formulated.
- At the end of 2005, Forrester Research published a study stating that 14% of companies in North America and Europe develop their software using agile processes.

Use of Scrum in SW development (share of companies in the USA)



Systematics of agile project management



Agile Principles 1

- Satisfaction of the customer through early and continuous **delivery** of the functioning product, **iterative work**
- Work in regular, preferably short periods of time (a few weeks), **timeboxing, incremental work**
- **Self-organization of the teams** in planning and implementation. Almost daily collaboration between subject matter experts and developers during the project (e.g. Collective Code Ownership, Cross-functional Teams)
- **Self-reflection** of the teams on their own behaviour to adapt with a view to increasing efficiency, **retrospectives**

Agile Principles 2

Transparency and collaboration: Sharing information to optimize team performance. Face-to-face conversations as much as possible

Only what is necessary is done – **simplicity**

Changes (e.g. in requirements) are considered normal, not disruptive.

Reviews and feedback are regularly obtained in order to adapt the product to the customer's wishes.

Agile Techniques

- Use of task boards
- Use Cases, User Stories and Epics for the Requirements Description
- **Daily Standup Meetings**
- WIP (Work in Progress) Limits
- **Burn Down Charts**
- **Definition of Done**
- Planning Poker
- Osmotic communication
- Use of personas
- Behavior Driven Development (BDD)
- Test Driven Development

Agile methodologies

What all **agile methods** have in common is that they use numerous techniques to keep the effort curve as flat as possible.

The most well-known are:

Feature Driven Development (FDD)

Extreme Programming (XP)

Scrum

Unified Process

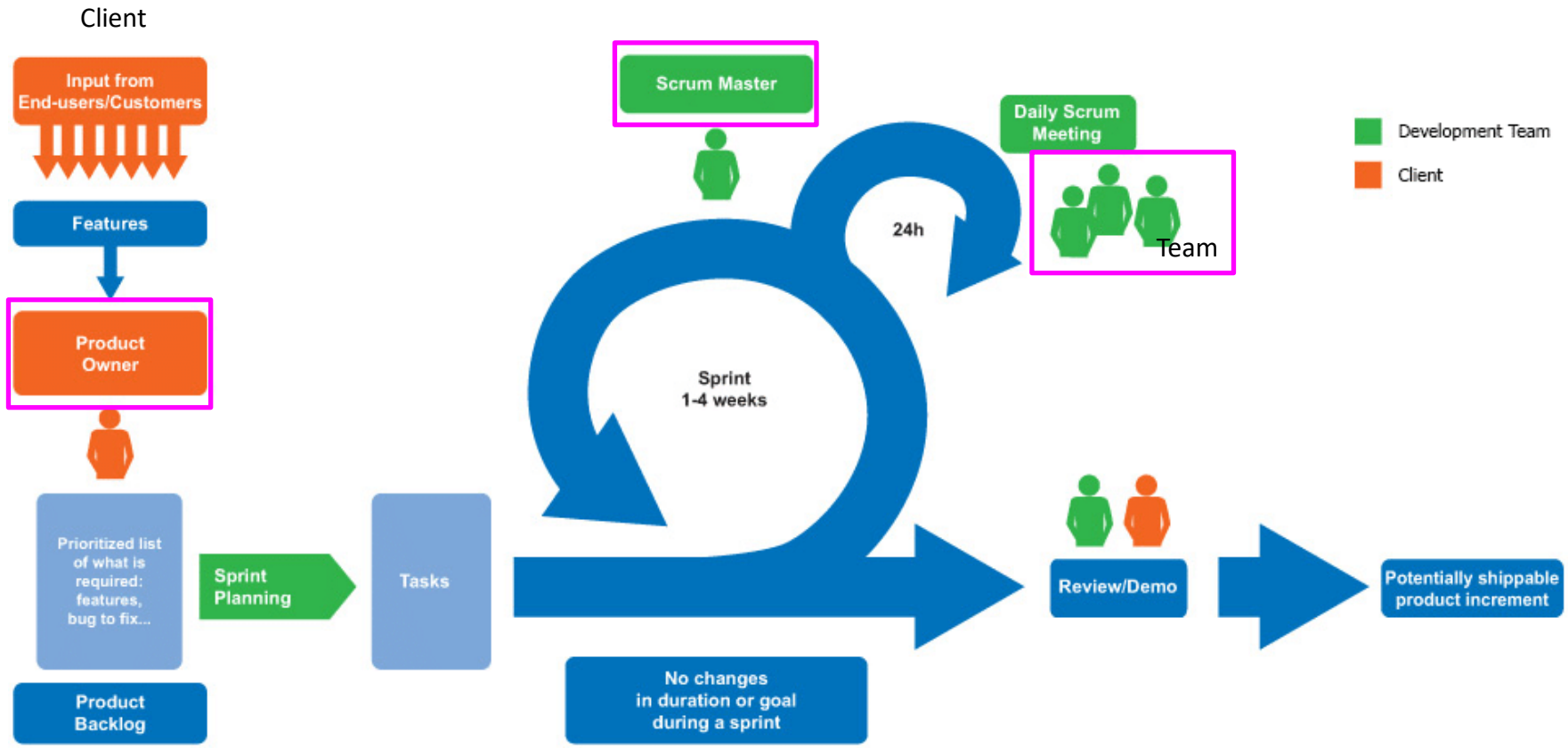
Design Thinking

Rapid Application Development (RAD)

Adaptive Software Development (ASD)

In contrast to standard project management methods, agile methods can be scaled and be used in smaller projects!

Agile project management in Scrum



Scrum – Three main roles

Product Owner

The PO defines the product vision and goals. He defines and prioritizes the product characteristics and is responsible for the economic success of the product.

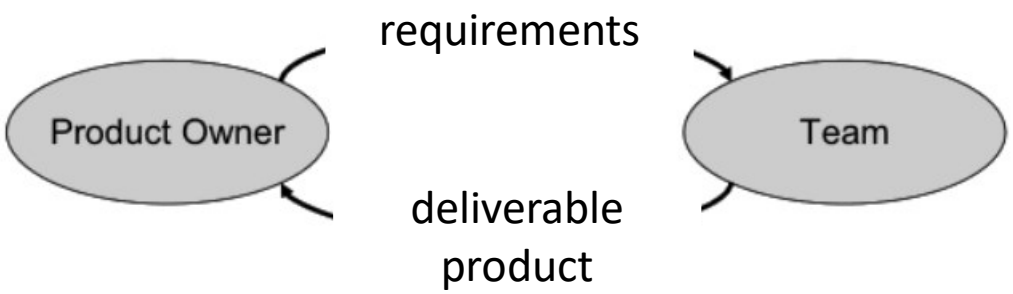
Scrum Master

The Scrum Master ensures that the team can work optimally. He takes care of conflict resolution, removes obstacles and keeps out outside interference. He is responsible for the smooth running of Scrum.

Team

The team works in a self-organized and cross-functional manner. It develops the product. The team is responsible for the sprint result and product delivery.

Product Owner (PO)



Tasks

The PO is part of the team.

The Product Owner is responsible for...

- the characteristics and economic success of the product.
- the definition of characteristics of the product
- the creation and maintenance of the product backlog
- decisions on implementation sequence
- informing stakeholders
- receiving feedback from stakeholders

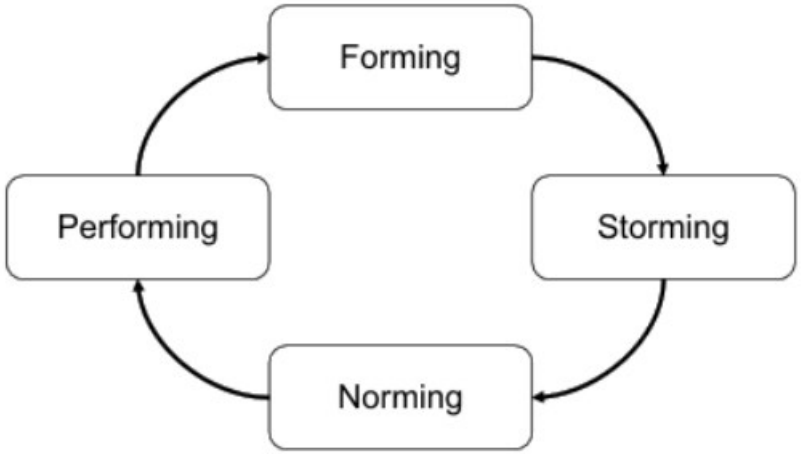
Interfaces

- Team
- Client
- Stakeholder

Team

Tasks

- Delivery of product functionalities in the order requested by the PO.
- Adherence to the agreed quality standards.
- Organization of tasks within the team
- At Amazon, for example, also maintenance/operation of the product after release (fully accountable team)



Interfaces / Procedure

- Product Owner
- Scrum Master
- Interdisciplinary staffing to be able to work independently on the implementation (architect, developer, expert, tester)
- 3-9 members

Scrum Master

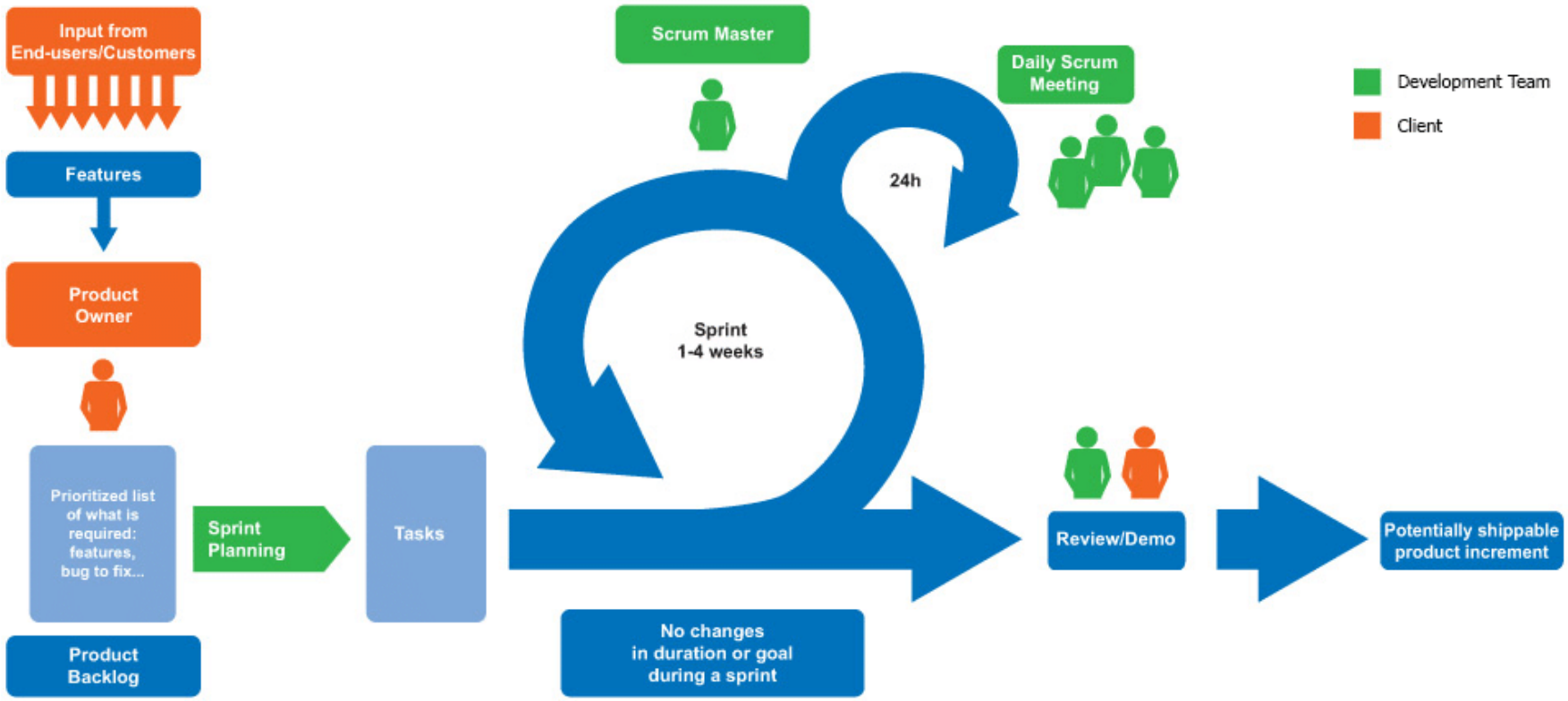
Tasks

- Responsibility for the success of Scrum
- Introduction and review of the Scrum process
- Elimination of faults and obstacles
- Conflict resolution (internal, external)
- "servant leader"
- Not a project manager!
- Feedback to the team (performance)
- Monitoring, Tracking, Reporting

Interfaces

- Team
- Stakeholder
- Linie
- Product Owner

Agile project management in Scrum



Tasks of project management in Scrum

Task	Responsibility
Scope / Goals	PO for the Product/Results Team for the Sprints
Schedule	PO for the Release Planning Team for the Sprint Backlog
Budget	PO
Communication	PO for the Reporting to Clinet and Management Team for Sprint Meetings, Retrospective
Risk Management	PO with Team Input
Quality	PO for the Product Features Team for the Development Environment Scrum Master for the Process
Supplier Management	Team and PO
Staff	Management for the right persons, Team for training and quality

Stakeholder 1

Customer / Client

The product will be made available to the customer(s) upon completion. Depending on the situation, customers can be internal departments as well as external persons or groups. It is the task of the product owner to inspire his customers by delivering the desired product. Therefore, product owners and customers should be in close contact for the duration of the project.

Users / Users

Users are the people who use the product. A user can, but does not have to, be a customer at the same time. The role of the user is of particular importance to the Scrum team, because only the user can judge the product from the user's perspective. Users and customers should be involved in the Sprint Review and Product Backlog Refinement to test the product and provide feedback.

Stakeholder 2

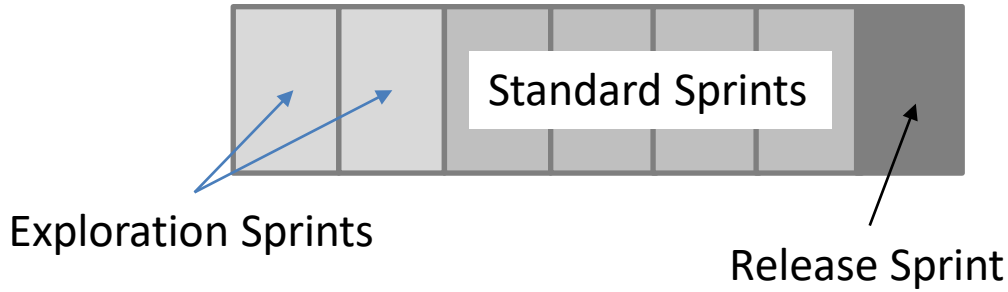
Management / Line

The management is responsible for ensuring that the framework conditions are right. This includes the provision of rooms and work equipment, but also general support for the course taken. Management is responsible for protecting the Scrum team from external work demands, finding adequate staffing, and supporting the Scrum Master in removing obstacles.

Sprint Types in Scrum

Exploration sprints for clarification, preparation sprints

- Generation of Knowledge, Installation of infrastructure etc.



Development Sprint (Standard Sprint)

- Generation of Product Features, Generation of Customer Value

Release Sprint for Completion

- Completion of a Product Release

Scrum Rules Meetings

Sprint planning (before each sprint)

- Define the content of the sprint.

Daily Scrum (daily)

- Daily approx. 15 minute discussion on the topics: "what happened yesterday", "what is planned for today", "what kind of problems / blockages are there".

Sprint Review (after each sprint)

- The team presents the result of a sprint to the Product Qwner and, if necessary, to the end customer.

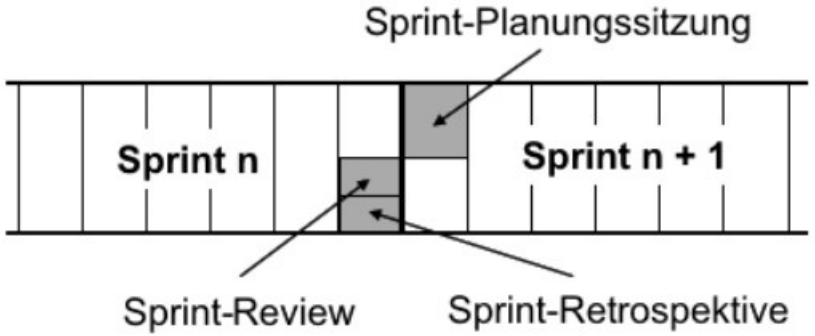
Sprint Retrospective (after each sprint)

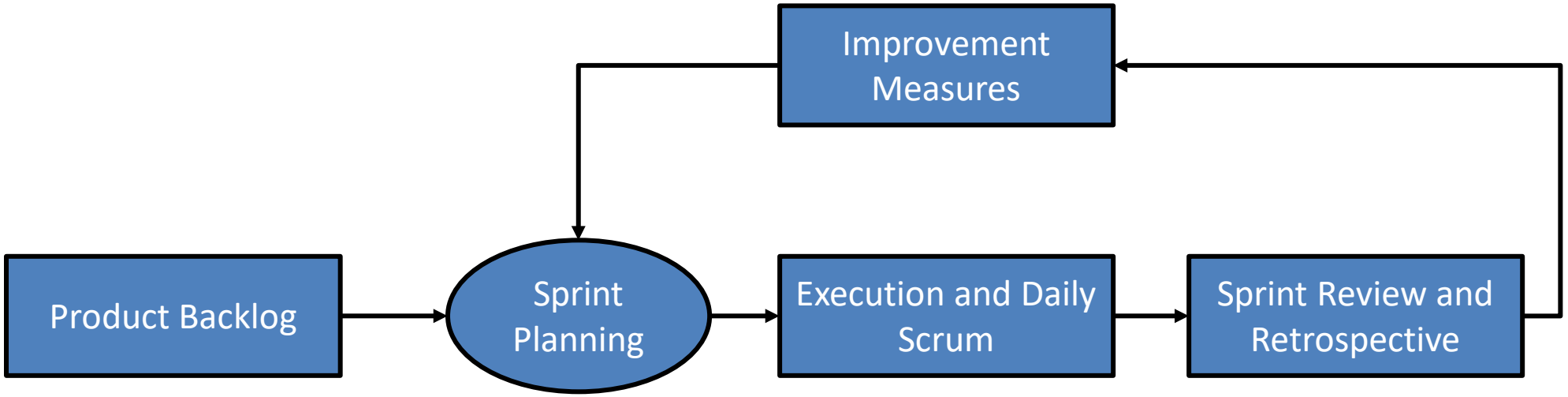
- Review of the team's development performance ("What was good?", "What was bad?", "Improvement measures").

Backlog Refinement (permanent)

- The team works together on the product backlog with the goal of refining and improving it.

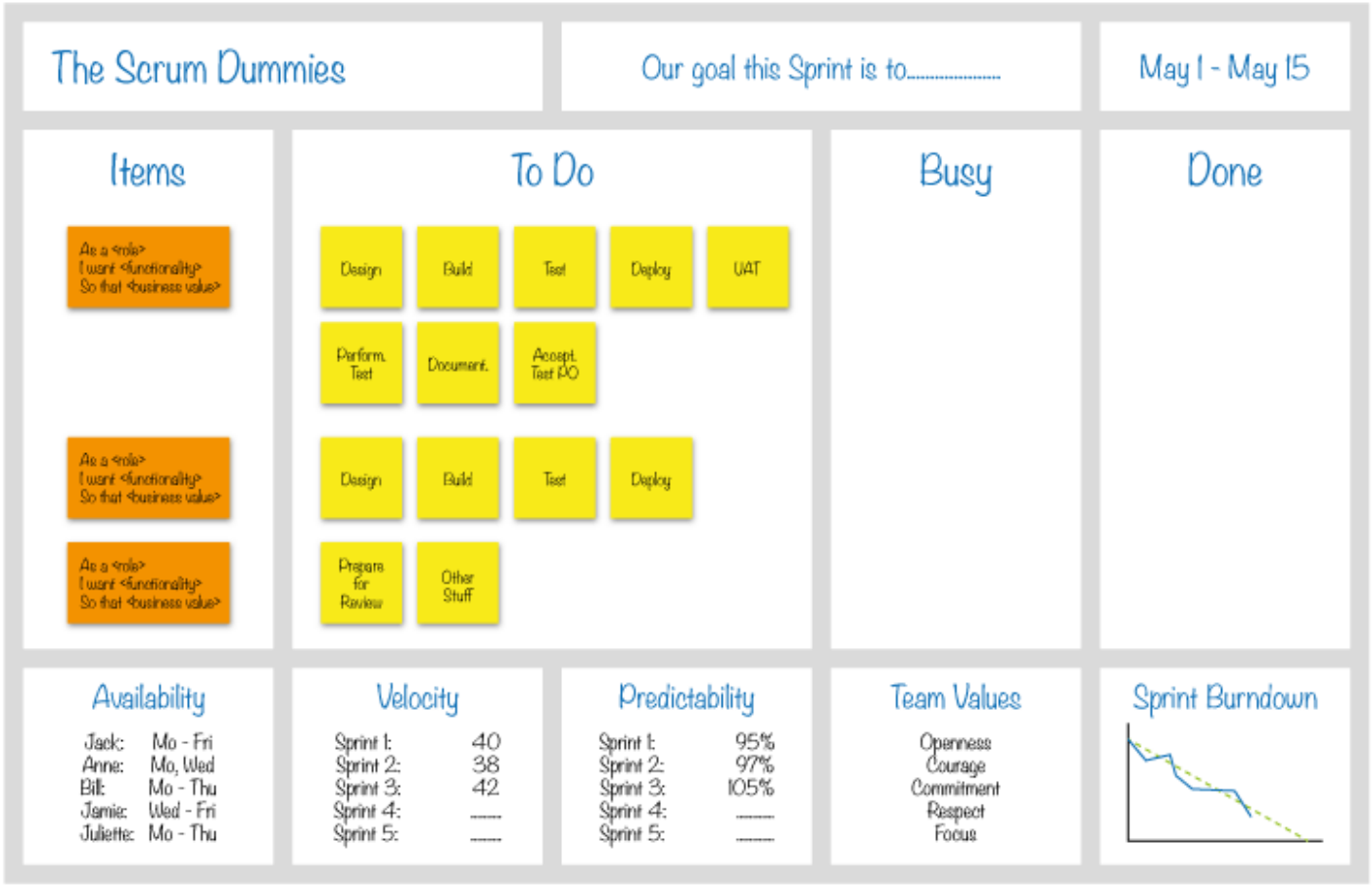
Scrum Meetings





Das Scrum Board (Whiteboard)

Also called **Kanban Board**
=> see Asana



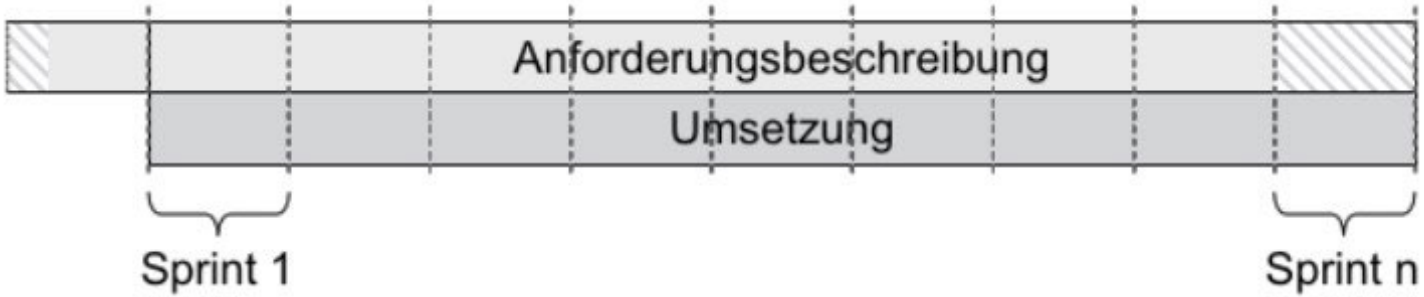
Template Scrum Board EPQM

Team Name		Goal of the sprint		Sprint start Sprint end
Sprint Backlog	Work in progress	Completed	Accepted	Sum Storypoints S1: S2: S3: S4: …
Team Values	Barriers	Tasks Scrum Master		

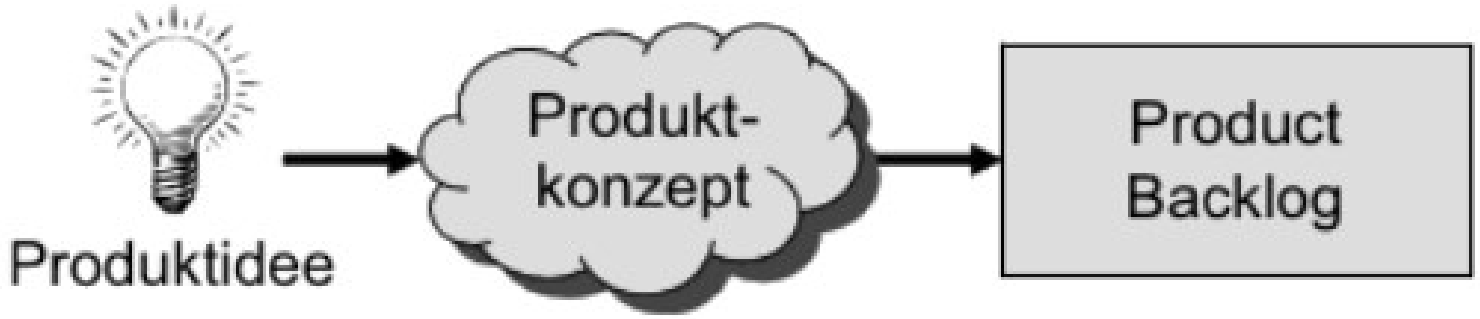
Requirements Management in Scrum

Traditional Requirements Management

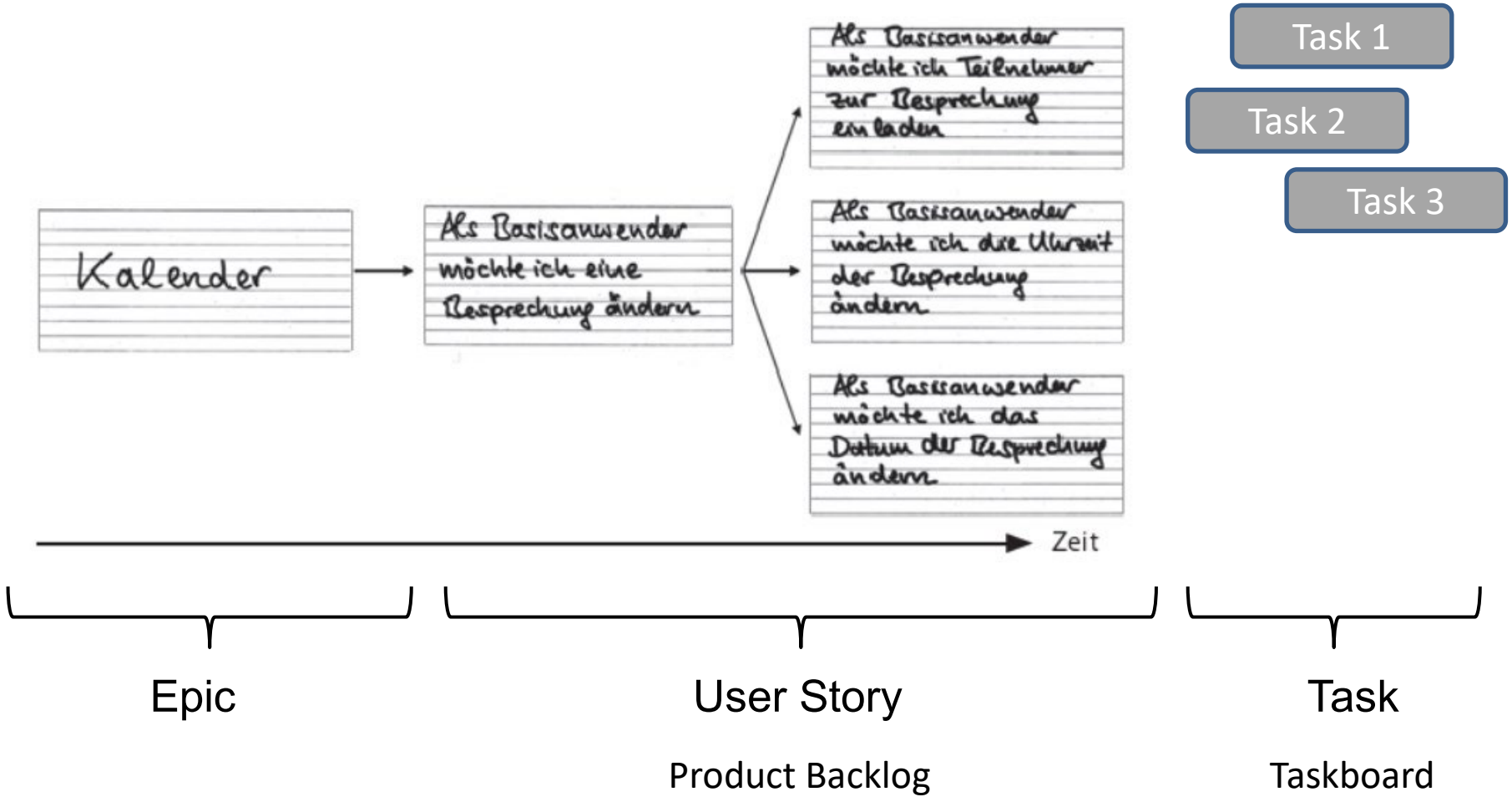
Requirements Management in Scrum



Product Backlog



Product Backlog – Epics, User Stories and Tasks



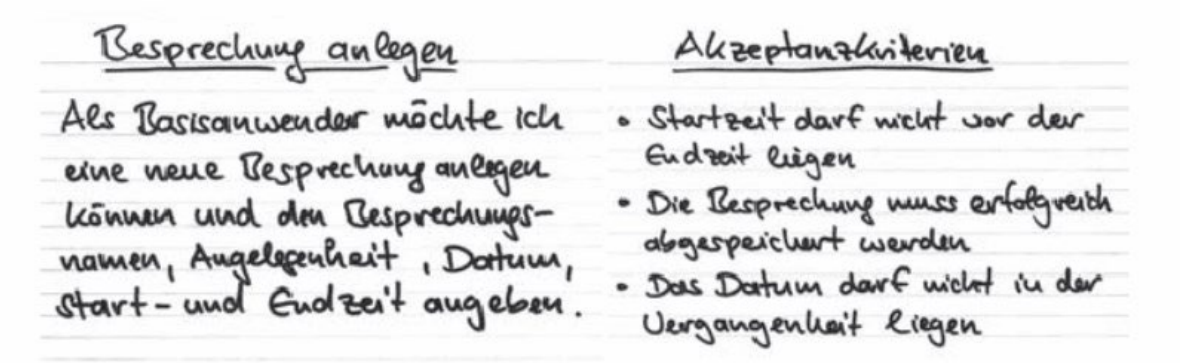
Product Backlog - Aufbau

Akzeptanzkriterien Storypoints (PO)

Priorität	Thema	Beschreibung	Akzeptanzkriterien	Aufwand
1	Kalender	Als Basisanwender möchte ich eine Besprechung anlegen.	Teste das Eingeben ungültiger Werte, z.B. Endzeit liegt vor Startzeit.	1
2	Kalender	Als Basisanwender möchte ich eine Besprechung stornieren.	Teste, dass dieselbe Besprechung nicht zweimal gelöscht werden kann.	3
3	Kalender	Als Basisanwender möchte ich eine Besprechung ändern.	Teste, ob die Änderungen persistiert wurden.	2
...				

- Priorisierung
- Epic => Thema
- Beschreibung => User Story
- Akzeptanzkriterin für Umsetzung
- Initiale Aufwandsschätzung

Product Backlog - User Story / Story Cards



Besprechung anlegen

Als Basisanwender möchte ich eine neue Besprechung anlegen können und den Besprechungsnamen, Angelegenheit, Datum, Start- und Endzeit angeben.

Akzeptanzkriterien

- Startzeit darf nicht vor der Endzeit liegen
- Die Besprechung muss erfolgreich abgespeichert werden
- Das Datum darf nicht in der Vergangenheit liegen

Titel	Prio	Points
Description: As <Rolle> if I want to <Nutzen><goal/desire>		
Additions: Risk, Explanation		

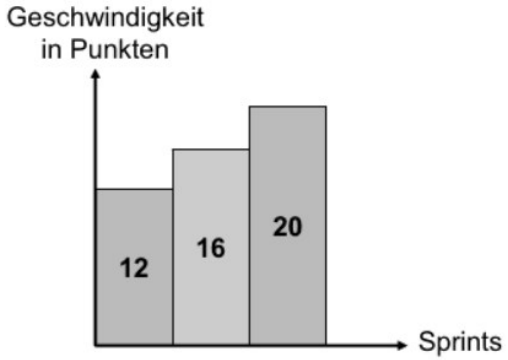
Status	ID
Acceptance criteria	
Test	

Effort Estimation - Storypoints

Planning-template

Punktewert	Semantik
0	Kein Aufwand
1	Sehr kleiner Aufwand
2	Kleiner Aufwand: doppelt so groß wie ein sehr kleiner Aufwand
3	Mittlerer Aufwand: so groß wie ein sehr kleiner und ein kleiner Aufwand zusammen
5	Großer Aufwand: so groß wie ein kleiner und mittlerer Aufwand zusammen
8	Sehr großer Aufwand: so groß wie ein mittlerer und großer Aufwand zusammen
13	Riesiger Aufwand: so groß wie ein großer und sehr großer Aufwand zusammen

Determination of the speed of development



Zugesagte Anforderungen	Anforderung abgenommen?	Geplante Punkte	Erzielte Punkte
Benutzergeschichte A	Ja	2	2
Benutzergeschichte B	Ja	2	2
Benutzergeschichte C	Ja	1	1
Benutzergeschichte D	Ja	2	2
Benutzergeschichte E	Ja	3	3
Benutzergeschichte F	Nein	2	0

Method: Planning Poker

- Unit of measurement is fixed and known to everyone (story points, hours, days, etc.)

- In the case of a relative estimate, the reference story is known.
- The user story is explained and questions are answered
-
- Everyone estimates the effort and turns over their guess card when everyone is done.
-
- Those with the highest and lowest Estimation must explain your reasons.
-
- Repetition until unity is reached.

Fibonacci Reihe

Definition of Ready / Definition of Done

Definition of Ready

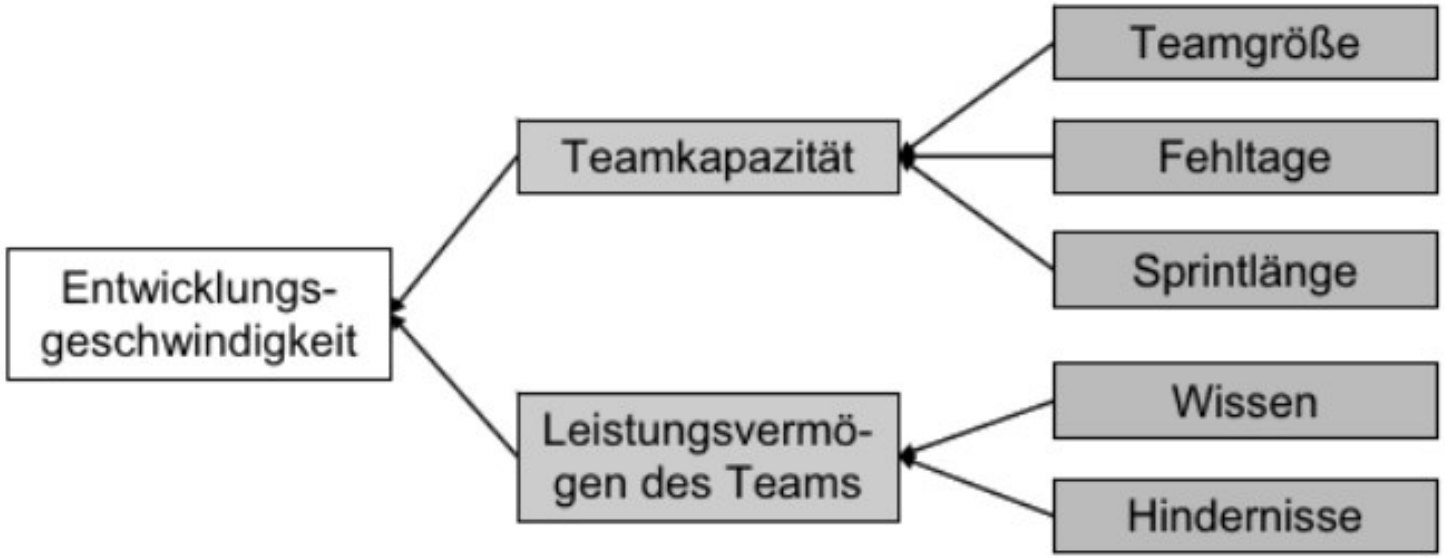
The Definition of Ready describes the state: "Ready for implementation". Before the implementation of the user stories / tasks within a sprint can begin, the Definition of Ready should be fulfilled for each user story / task. This favors and enables the smoothest possible implementation.

Definition of Done

The Definition of Done defines when the user stories / tasks of a sprint are fully implemented. It helps the Scrum team to ensure the quality of its work.

Speed of development

Dependence of the speed of development on internal and external influences

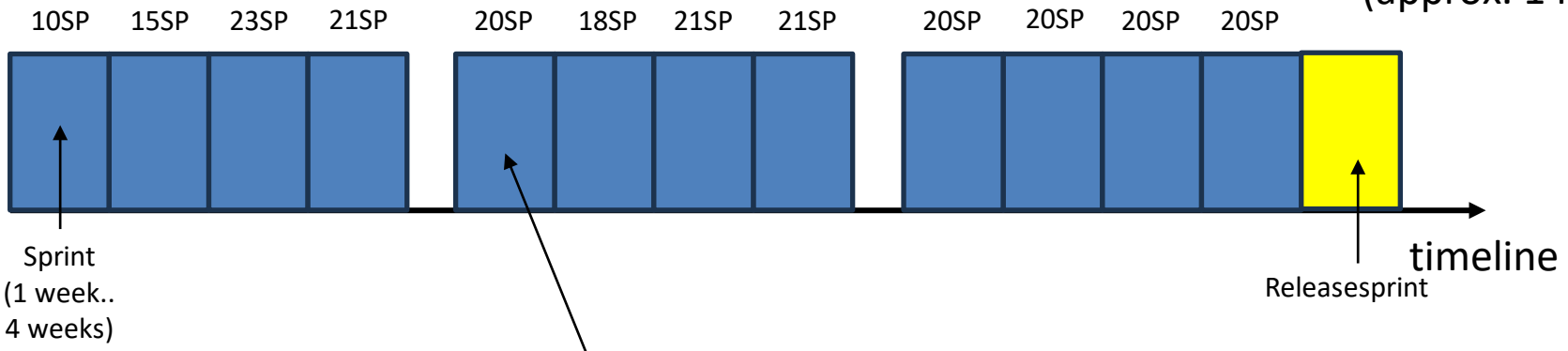


The development speed (Performance of the team) can only be determined after several sprints.

How to calibrate team Performance

In agile projects you also would like to know when there will be results finished.
This normally is measured in the number of Sprints.

A Backlog worth 229 SP was realised after 12 Sprints (approx. 14 weeks)

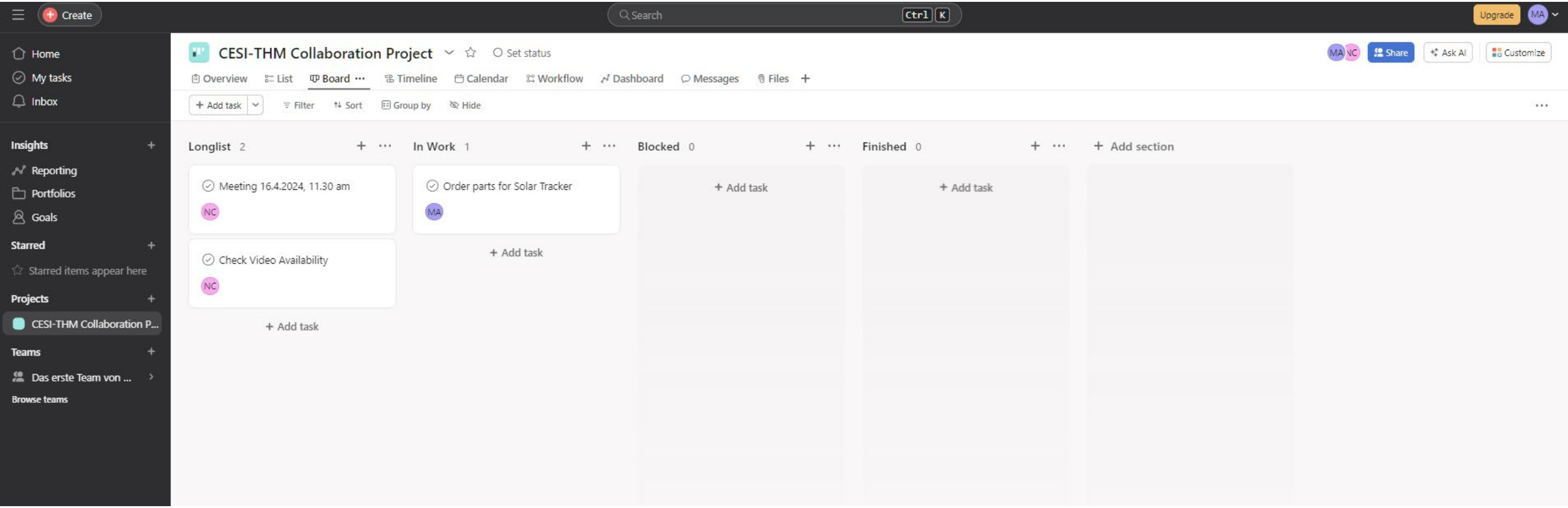


The question now is:
How many Storypoints can the team finish in one Sprint?

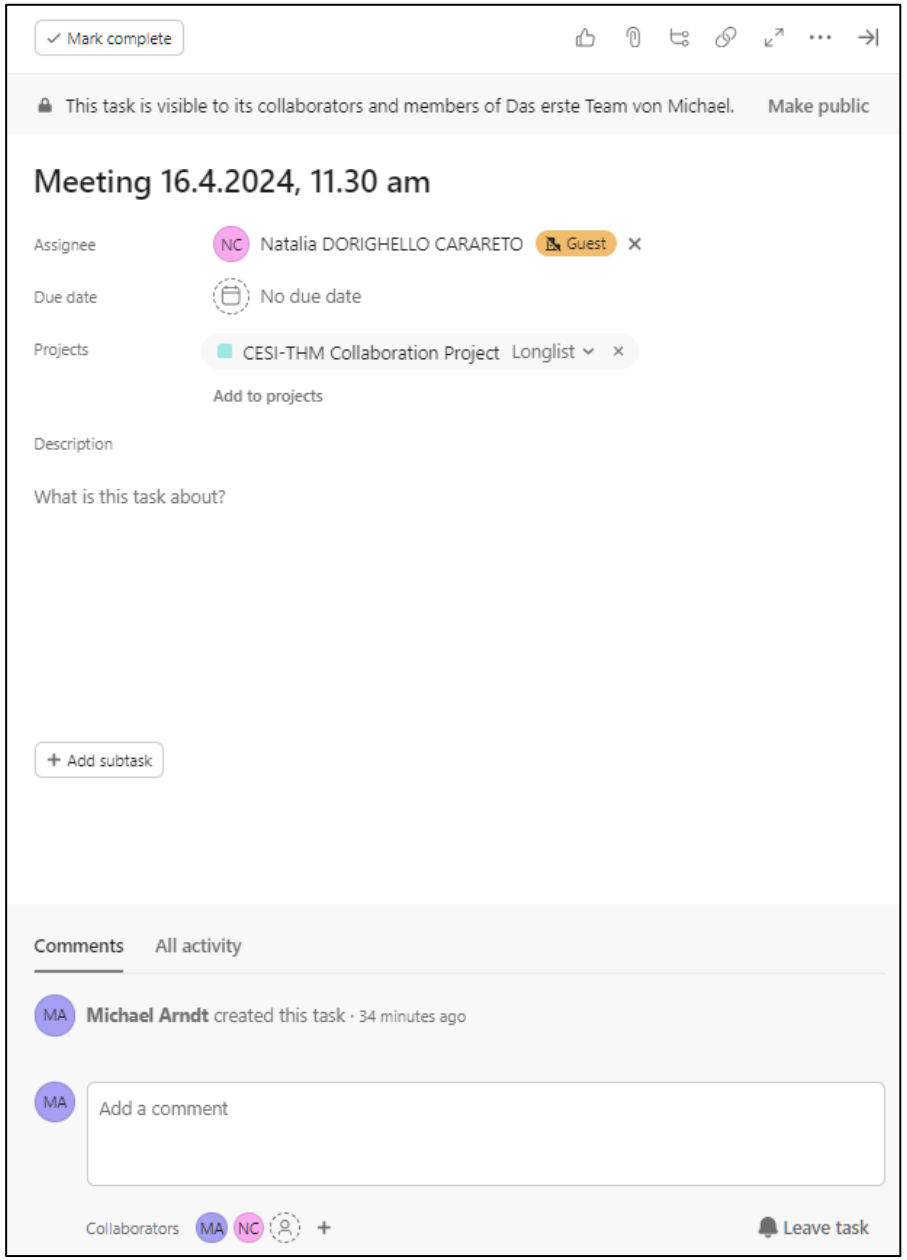
After some Sprints, there will be an average number of storypoints, that the team can finish.
Example: 20 SP

Asana
Agile Board

Asana Board


Create
Search
Ctrl K
Upgrade
MA
Home
My tasks
Inbox
Insights
Reporting
Portfolios
Goals
Starred
Starred items appear here
Projects
CESI-THM Collaboration P...
Teams
Das erste Team von ...
Browse teams
CESI-THM Collaboration Project
Set status
Share
Ask AI
Customize
Overview
List
Board
Timeline
Calendar
Workflow
Dashboard
Messages
Files
Add task
Filter
Sort
Group by
Hide
Longlist 2
In Work 1
Blocked 0
Finished 0
Add section
Meeting 16.4.2024, 11.30 am
NC
Order parts for Solar Tracker
MA
Check Video Availability
NC
Add task

Asana Task


Mark complete
This task is visible to its collaborators and members of Das erste Team von Michael.
Make public
Meeting 16.4.2024, 11.30 am
Assignee
NC
Natalia DORIGHELLO CARARETO
Guest
Due date
No due date
Projects
CESI-THM Collaboration Project
Longlist
Add to projects
Description
What is this task about?
Add subtask
Comments
All activity
MA
Michael Arndt created this task · 34 minutes ago
MA
Add a comment
Collaborators
MA
NC
Leave task

Some Tasks

How would you organize...

...Communication within your Team?

...Exchange of Files in your Team?

...the first team meeting? What should be results of this meeting?

...the rules of your team?

...an Asana project for your team?

...the working process of your team?

...risks?

...a social event to get to know the other team members?