Agile Development in Cloud Computing Environments

Information Technology (M.Eng.)

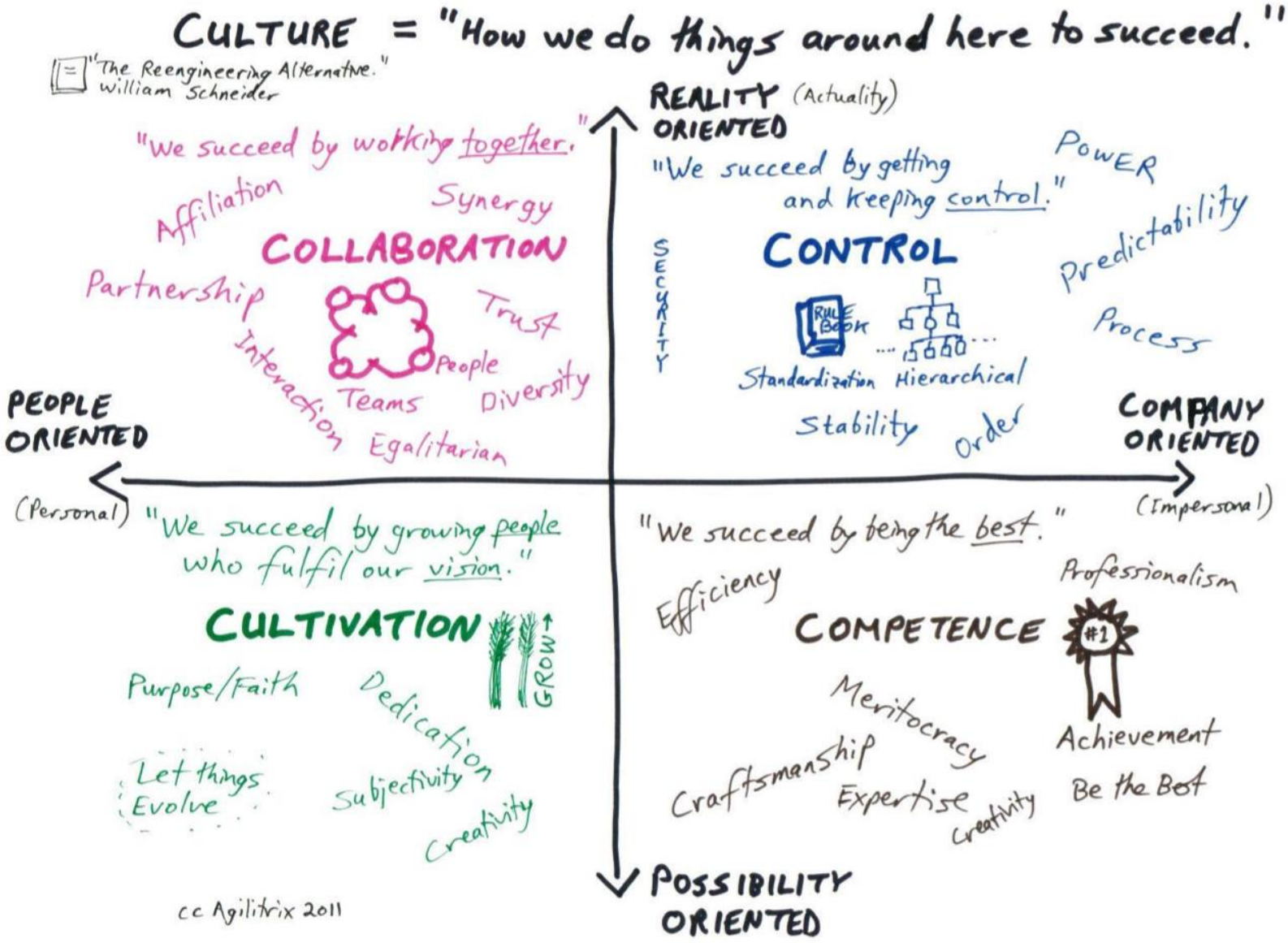
Module 11: Optional Technical Subject

SoSe 2022

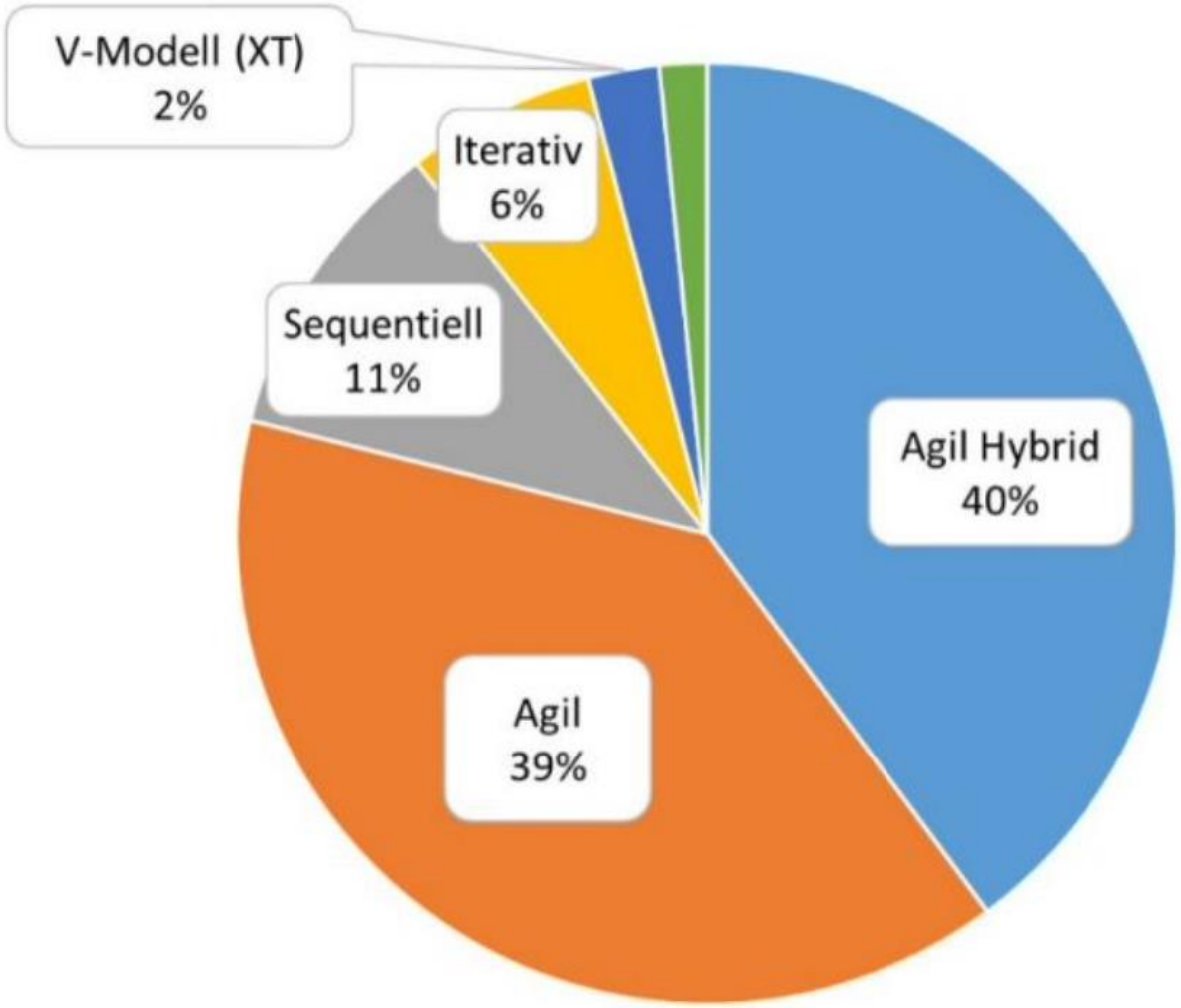
Dr. Patrick Wacht (email: patrick.wacht@fb2.fra-uas.de)

Frankfurt University of Applied Sciences, Frankfurt a. M., Germany

Four key cultures in agile development

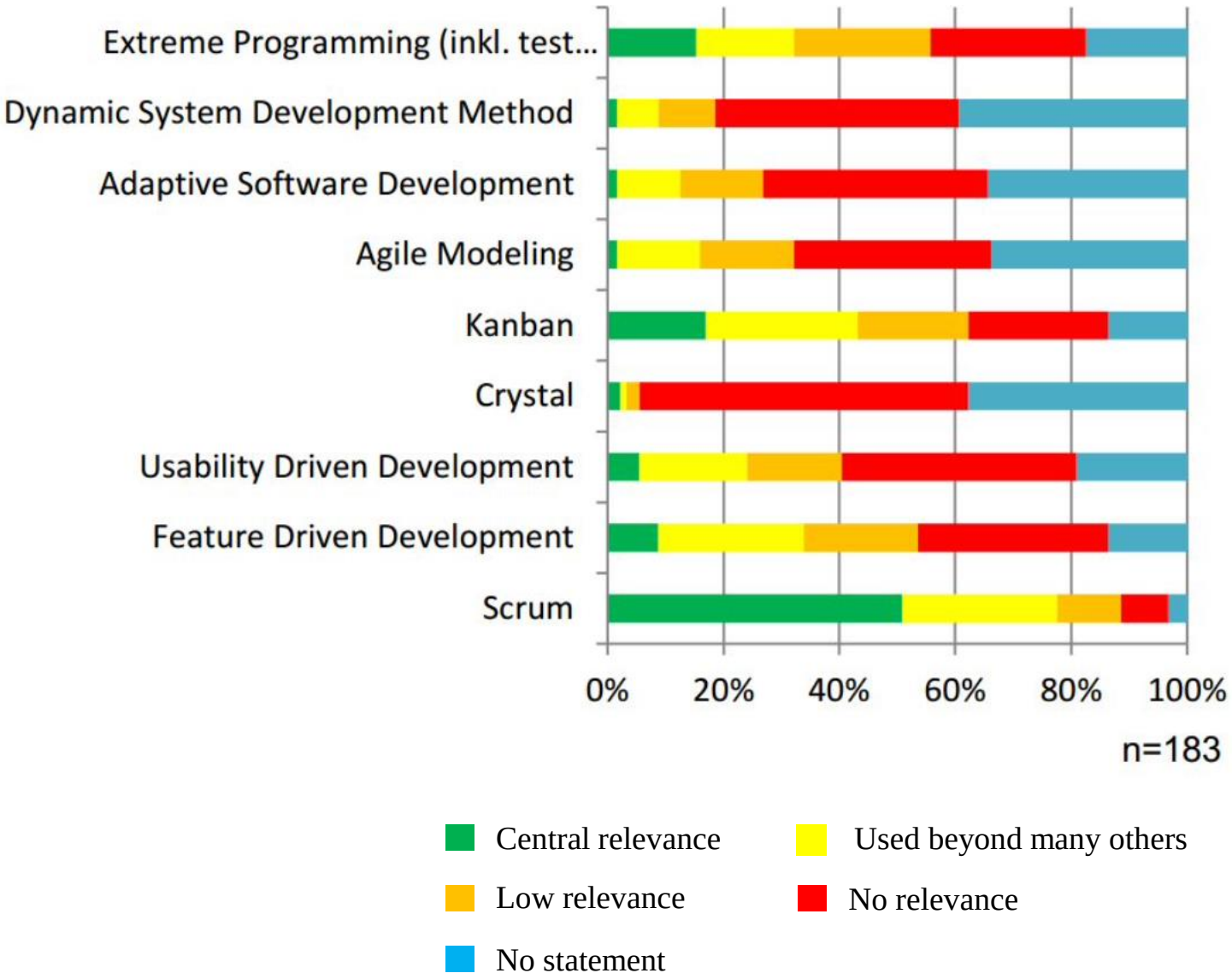


Usage of different development methodologies (2019)

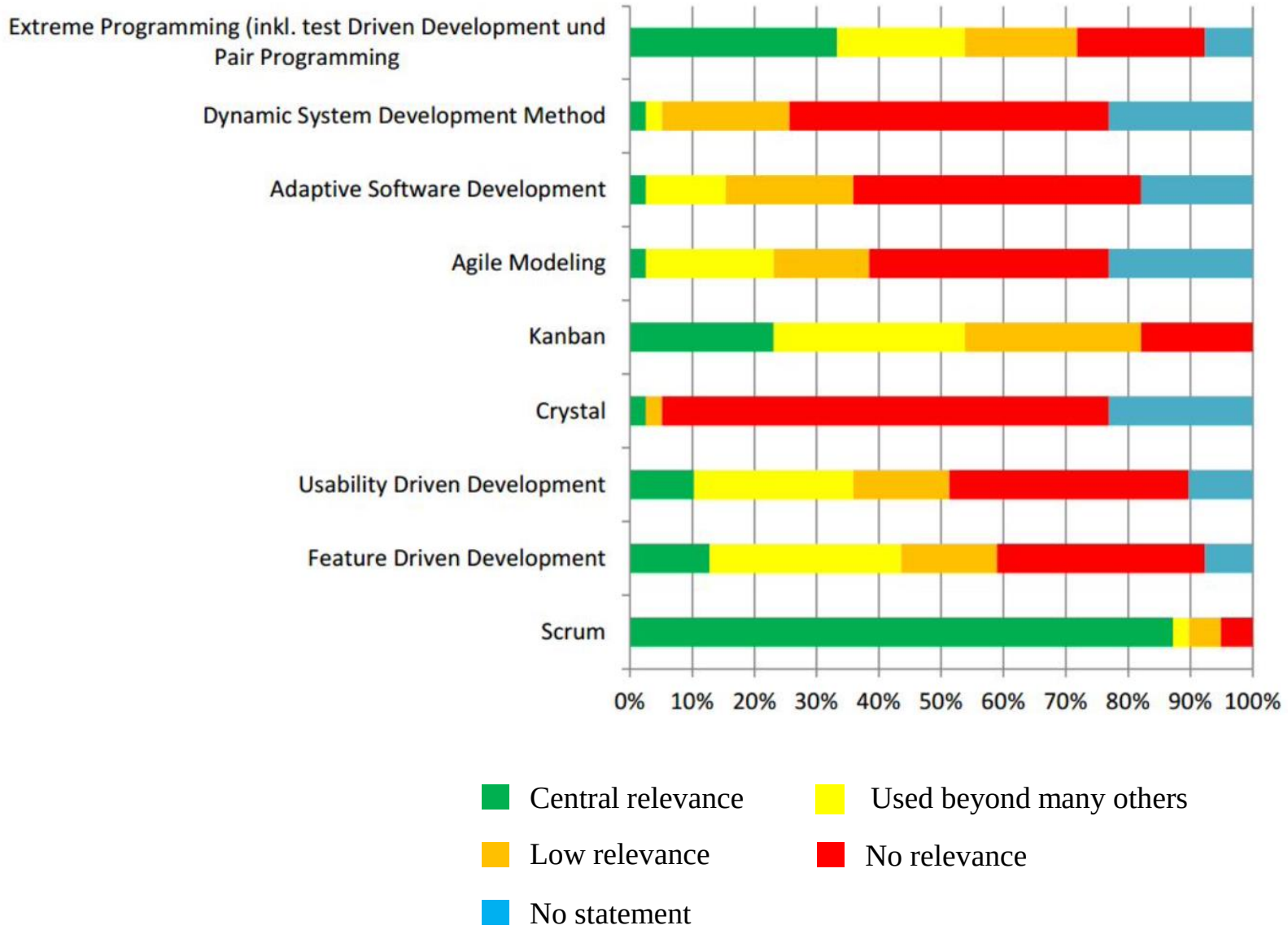


[UX, 2019]

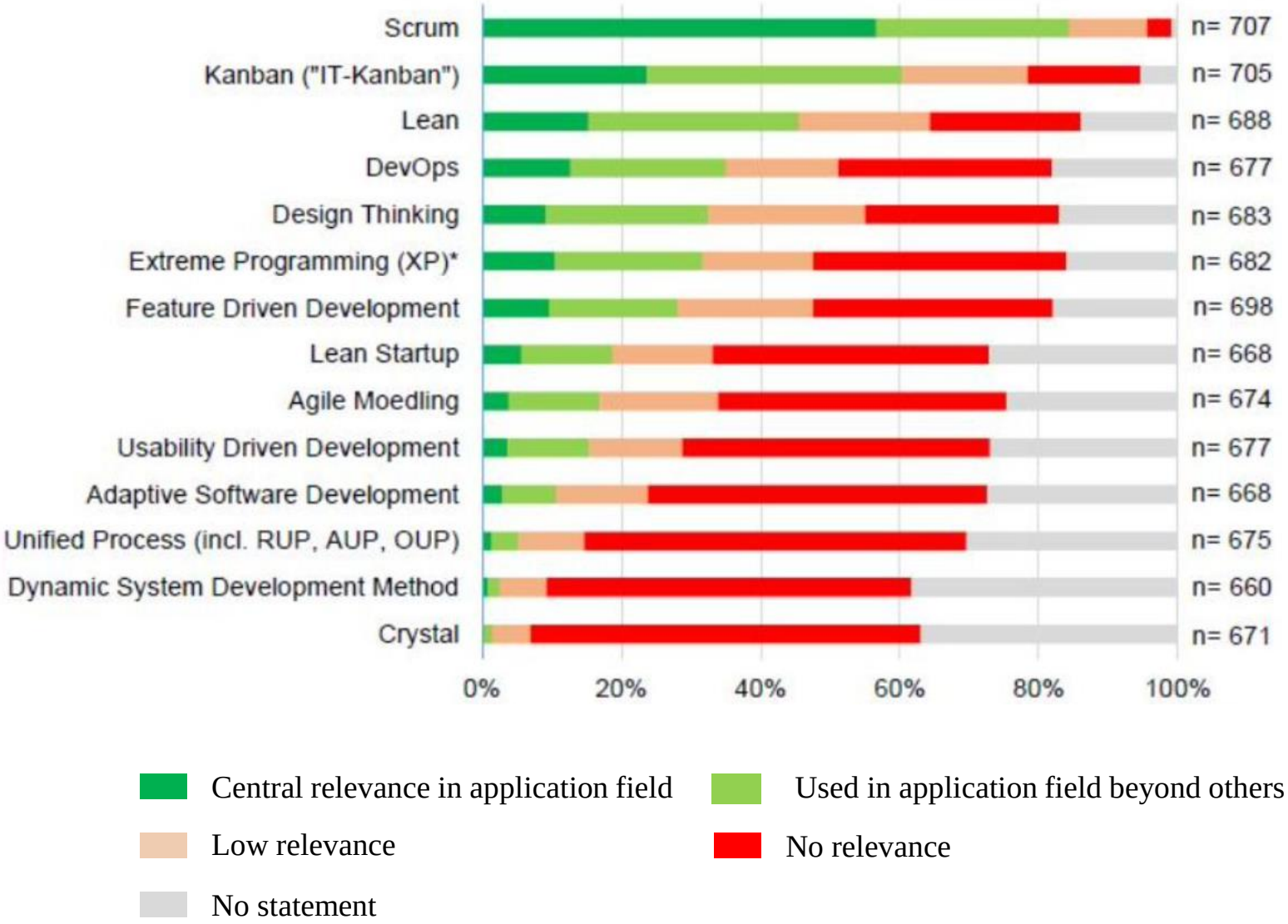
Which methodology is widely used?



From which methodology functions/aspects are adopted?



Significance of methodologies



Definition: Scrum

A Framework within which people can tackle complex adaptive tasks and by being able to deliver products with the highest possible value productively and creatively.

Scrum…

… is a framework for the development and maintenance of complex products, within which different processes and techniques can be used

… is
- lightweight,
- easy to understand,
- difficult to master.

… consists of
- Roles,
- Events,
- Artefacts and
- Rules, that connect the individual elements.

The rules are discussed in the context of roles, events and artifacts.

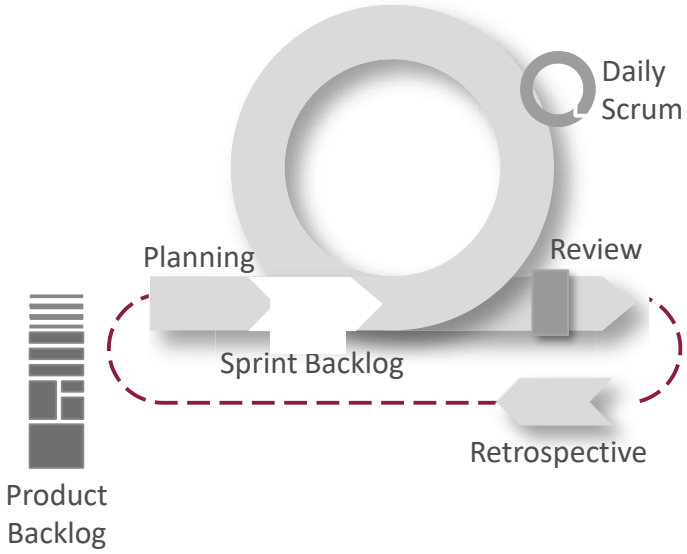
Roles: The Scrum Team

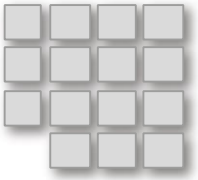
Events (Time Box)
- Sprint (max. 4 Weeks)
- Sprint Planning (max. 1* Day = max. 8* hours),
- Daily Scrum (max. 15 minutes)
- Sprint Review (max. 4* hours)
- Sprint Retrospective (max. 3* hours)

Artefacts
- Product Backlog
- Sprint Backlog
- Increment
- Definition of Done

* The times are reduced for shorter sprints.

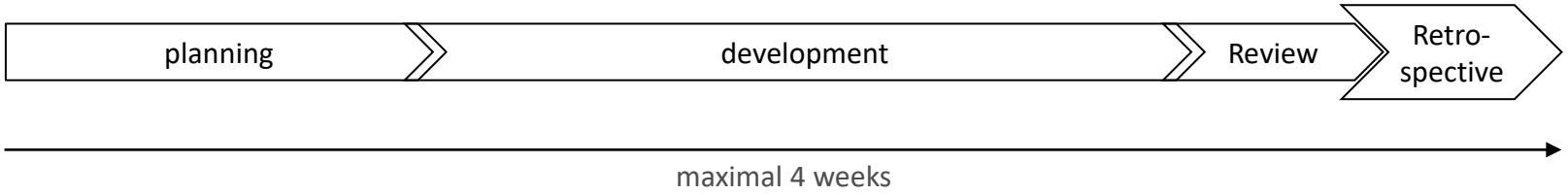
Scrum: events in the sprint process





Increment

An increment is only ready and potentially ready for delivery if it corresponds to the definition of done!



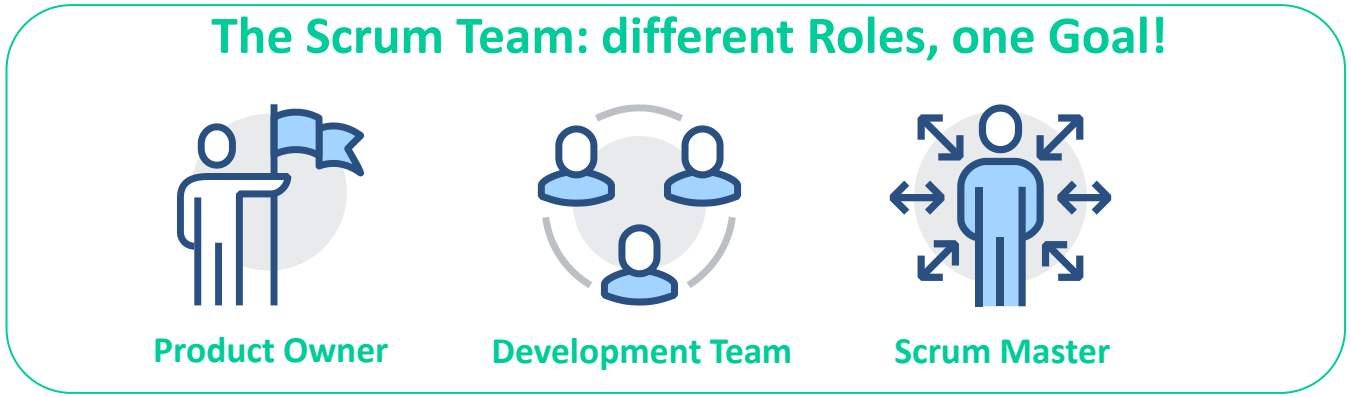
Advantages of Scrum

- Organizations that engage in Scrum benefit from the learning curve of all stakeholders and regularly inspire their customers by delivering what they want and not “only” what was set once on the first day.

Additional advantages:

- Improved returns through more frequent smaller releases
- Reduced costs by improving poor organizational processes
- Quicker use thanks to operational increments
- Confidence in being successful in a complex world
- Enjoy working through close collaboration, which leads to improved interpersonal relationships and greater mutual trust

The Scrum Team



Scrum Teams …

- … **are self-organizing and interdisciplinary.**
- … **decide for them self how best to do your job.**
- … **have all the skills required to get the job done.**
- … **are able to optimize flexibility, creativity and productivity.**
- … **deliver products iteratively and incrementally, maximizing the opportunity for feedback.**

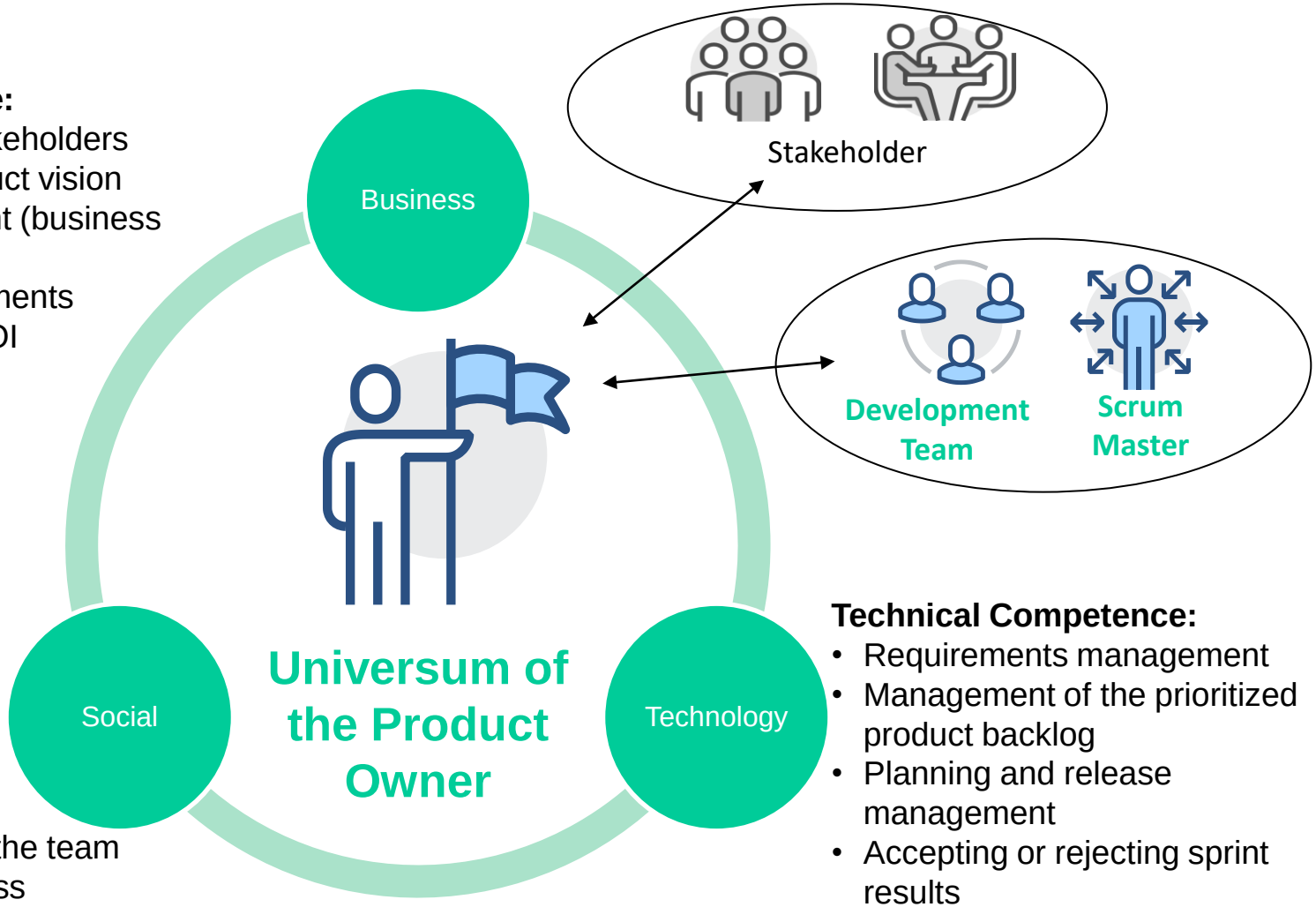
The Product Owner

- Is accountable for the economic success and value maximization of the product.
- Is accountable for the value-adding work (Product Backlog) of the development team.
- Is accountable for the product backlog.
- Is a single person, not a committee.
- Can only be successful if its decisions are respected in the organization.

➢ The decisions of the Product Owner are visible in the content and order of the Product Backlog.

The Product Owner

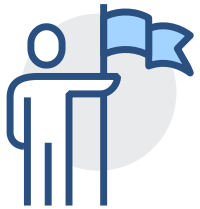
Business expertise:
- represents all stakeholders
- conveys the product vision
- strategic alignment (business / development)
- Prioritizing investments
- responsible for ROI



Technical Competence:
- Requirements management
- Management of the prioritized product backlog
- Planning and release management
- Accepting or rejecting sprint results
- Monitor progress
- Conducting reviews

Social Skills:
- Motivation of the team
- responsiveness
- ensures that decisions can be made at the lowest possible hierarchical level
- Resolving team doubts about the project

Main tasks of the Product Owner



- Organization of economic concerns
- Participate in the planning
- Maintenance of the Product Backlog
- Definition of acceptance criteria and verification of their compliance
- cooperation with …

  … Development Team

  … all Stakeholders

The Development Team

... is able to implement all types of activity
that are necessary to create an increment.

Code programming

Data Base design

Code testing

Architecture design

peparing Documentation

User Interface (UI) design

Server management

The Development Team

Development Team

- … consists of professionals …
  - … who hand over a finished increment at the end of each sprint,
  - … which is potentially deliverable.
  - … is structured and empowered to organize and manage its own work.
  - ➢ The resulting synergy optimizes the overall efficiency and effectiveness of the development team.
- … is always accountable as a whole.
- … knows no dedicated titles like architect, tester etc.: „**No ranks, no titles**“.
- … is small enough to stay nimble and big enough to get the job done in a sprint.
- … consists of 3–9 members.

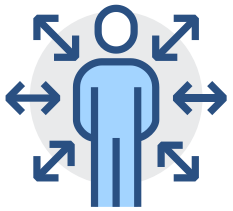
If the Product Owner and Scrum Master also do work from the Sprint Backlog, they are part of the development team.

Main tasks of the Development Team

- Carry out sprint
- Participation in the daily scrum (daily examination and adjustment)
- Maintenance of the product backlog (refine, estimate, if necessary, create and prioritize)
- Plan sprint
- Examine and adapt product (sprint review) and process (retrospective)

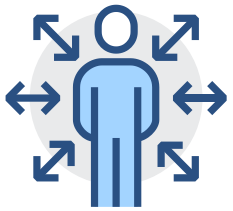
Development Team

The Scrum Master



- … is responsible for understanding and executing Scrum.
- … does this by ensuring that the Scrum team understands
- … and follows Scrums …
  - … **Theory**,
  - … **Practices**
  - … **Rules**
- … helps to optimize the collaboration in such a way that the value generated by the Scrum team becomes maximum.
- … Acts as a Servant Leader for the Scrum team.

The Scrum Master: Servant Leader for



Scrum Master

… the Product Owner

- Providing techniques for effective product backlog management
- Create an understanding of product planning in an empirical work environment
- Provide the right understanding of agility and its application
- Support in the execution of Scrum events

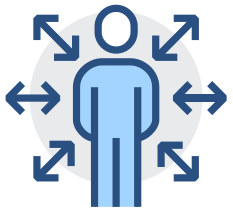
… the Development Team

- Coaching and Support …
  - … in everyday work.
  - … when introducing Scrum
- Remove obstacles (impediments)
- Support in the execution of Scrum events

… the organization

- Leading and coaching the introduction of Scrum
- Planning Scrum implementations
- Helping stakeholders understand Scrum

The main tasks of the Scrum Master



Scrum Master

- Coach for development team and product owner
- Servant Leader
- Process authority (Scrum values, principles and practices)
- protects the development team from disruptive influences
- Eliminate obstacles unless the development team is able to do so
- Organizational development consultant for the Scrum team and stakeholders (ensures the success of Scrum in the organization)

The Product Backlog

Collection of all requirements for the product to be created

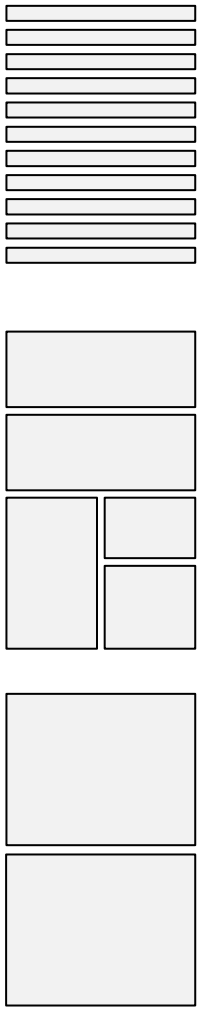
Consists of differently detailed requirements

Evolves throughout the project

Requirements that are to be implemented in one of the next sprints are more detailed than requirements that will only be implemented later.

The requirements are described in the form of user stories.

As <Role>, I want <Target>, so that I <Advantage>



detailed functions

More detailed request packages („function groups“)

rough requirement packages („Modules“)

User Stories (Product Backlog) from Brainstorming

Events:
As ... I want ... so that I ...

Service-Chat-Bot:
As ... I want ... so that I ...

Order Manager:
As ... I want ... so that I ...

Newsletter-Management:
As ... I want ... so that I ...

Homepage (Look & Feel):
As ... I want ... so that I ...

Reviews:
As ... I want ... so that I ...

Articles:
As ... I want ... so that I ...

Area for premium customers:
As ... I want ... so that I ...

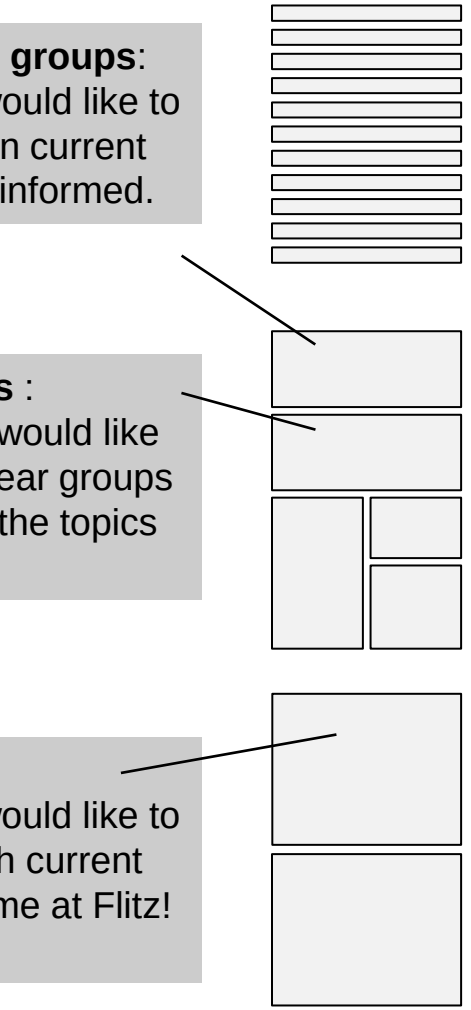
FAQs:
As ... I want ... so that I ...

News:
As ... I want ... so that I ...

News - Subscribe to topic groups:
As a visitor to the website, I would like to be kept up to date on certain current topics by email so that I am informed.

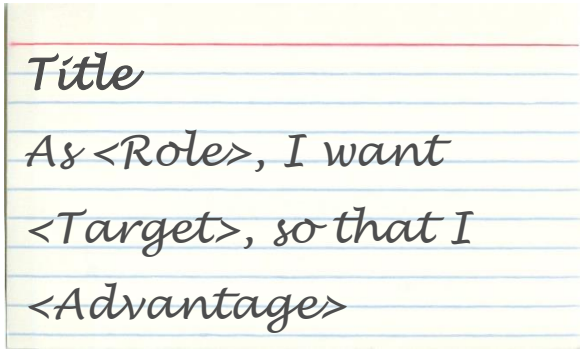
News - topic groups :
As a visitor to the website, I would like the topics to be grouped in clear groups so that I can quickly access the topics that interest me.

News:
As a visitor to the website, I would like to be able to quickly see which current information is interesting for me at Flitz! Auto.



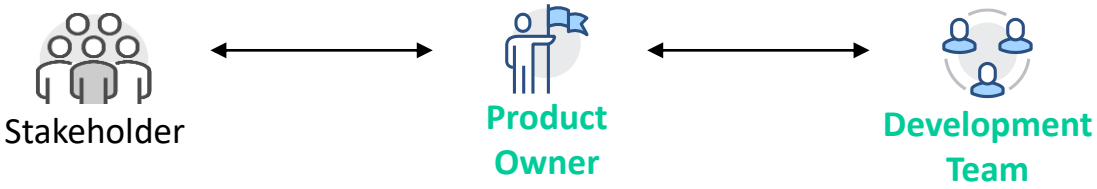
User Story = 3Cs

Card



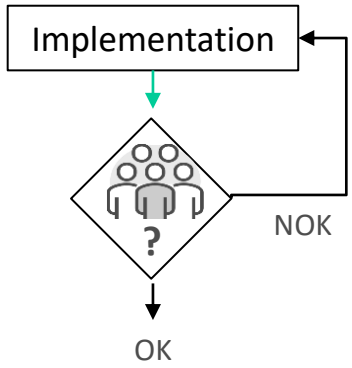
Conversation

Dialog => necessary details!



Confirmation

Conditions of Satisfaction (COS)



Good User Stories are „INVEST“

- **I**ndependent: A story is independent of other stories.
- **N**egotiable: User stories are not a written law; if necessary, these can be changed in the consensus of the stakeholders.
- **V**aluable: Stories should deliver recognizable added value.
- **E**stimable: A story must be so clear that the developers can estimate the effort of the implementation.
- **S**mall: The team has to decide on the specific scope of stories. As a rule of thumb, the implementation of a story should be between 0.5 and 10 person days.
- **T**estable: Stories must be testable.

+ not technically formulated

Product Backlog Items (BPIs): User Stories

The Product Backlog consists of user stories …

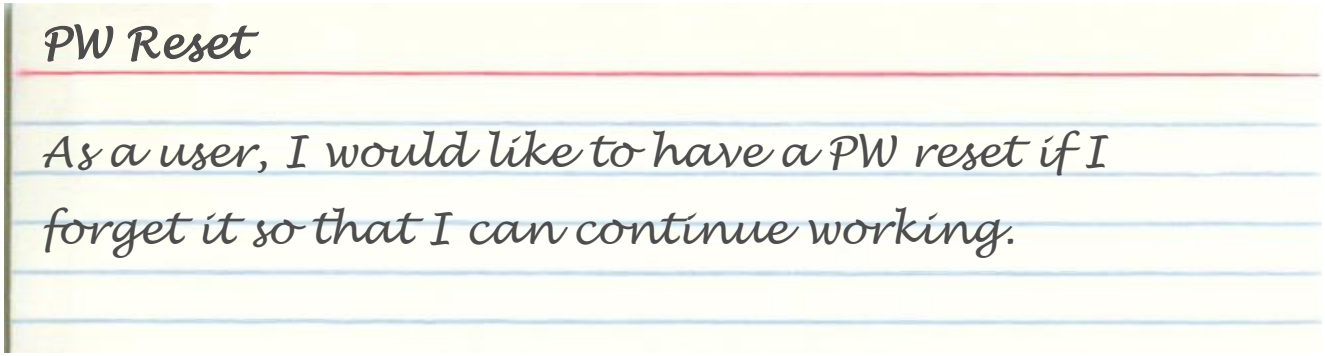
… with functional requirements

… with changes (improvements, concretizations, etc.) to already implemented functional requirements

… with technical improvements / adjustments (necessary updates)

… for troubleshooting (tickets)

… to acquire knowledge



Stories to acquire knowledge (“Spikes“)

What we don't know, we have prototypes, concepts, experiments or studies, etc. to find out.

User Story

As a developer, I want to know alternative solutions for the service chat bot to know which solution is the better choice in the long run.

Confirmation

Clarity should be achieved as to which solutions have a good price-performance ratio for our requirements.

User story and possible other attributes

- **Name/Title** (if necessary, plus a unique ID)
- **User Story**
  As a <Role> I need a <Functionality> so that I get the <Benefit>.
- **Acceptance Criteria**
  The acceptance criteria answer the question of what should be tested and ensure that a story can be accepted.
- **technical (non-functional) requirements**
  Mandatory requirements regarding architecture, implementation, monitoring requirements (heartbeat), dependencies. Dealing with code refactoring.
- **Test Coverage** (Environment (test, dev, prod), test data, test scenarios …)
- **Additional Information**
  Framework conditions to be observed, assumptions and exclusions, description of the initial situation, click paths, process / flow diagram, URL …
- **Risks**, that are associated with the (non) implementation (=> test concept)
- **Dependencies** to other stories / requirements / ideas
- **Further Information**
  Screenshots, mockups, documentation, interfaces ... as well as work processes, such as the following :
  „ Each new story must go through at least one round between the author and the developer in order to enable a uniform understanding. The user story is expanded, i. H. Details are described, acceptance criteria expanded, missing data (such as screens, test data etc.) added, answers to questions (in the description section) documented etc.“

Product Backlog Structure

small User Stories

- As an editor, I need an area on the homepage in order to be able to place current announcements clearly visible to the visitor.
- ...

Days

n

1

(larger User Stories) **Epics**

- As an editor, I need an easy-to-use editing system (web content management system) that gives me the maximum freedom to place content.
- ...

Weeks

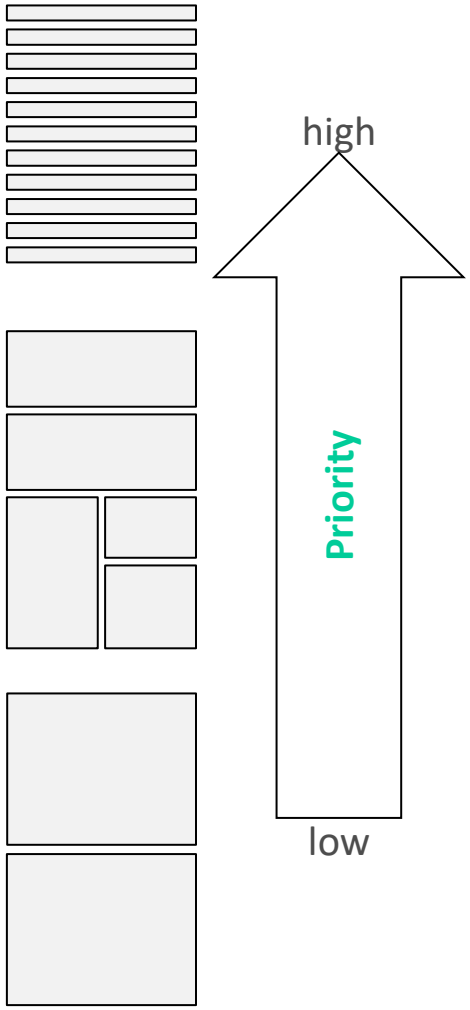
n

1

(large User Stories) **Themes**

- As a company, we need an attractive and modern homepage that fits our new image so that we appear credible to the outside world.
- ...

Month



The Sprint: The Heart of Scrum

The Sprint is a time box of a maximum of one month, within which a finished ("Definition of Done"), usable and potentially deliverable product increment is produced.

All sprints within a development project should be of the same length.

The new sprint starts immediately after the previous sprint is completed.

Each sprint includes the following events :

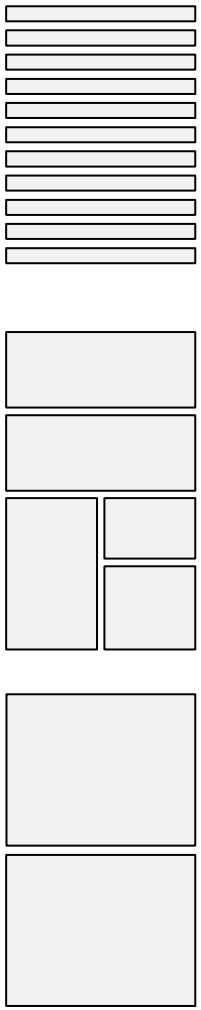
- Sprint Planning
- Daily Scrums
- Sprint Review
- Sprint Retrospective

Sprints are sometimes referred to as iteration.

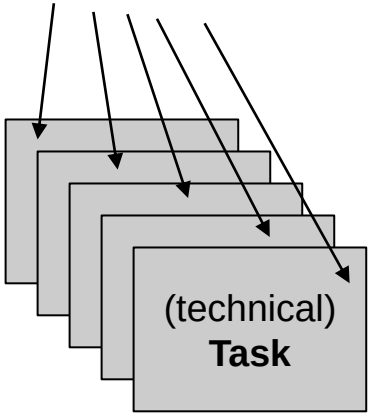
During the sprint …

- … no changes are made that jeopardize the sprint goal.
- … the quality standard is not reduced.
- … the scope of requirements can be renegotiated based on new knowledge between the product owner and the development team.

Sprint Planning: Decomposition of User Stories → Tasks



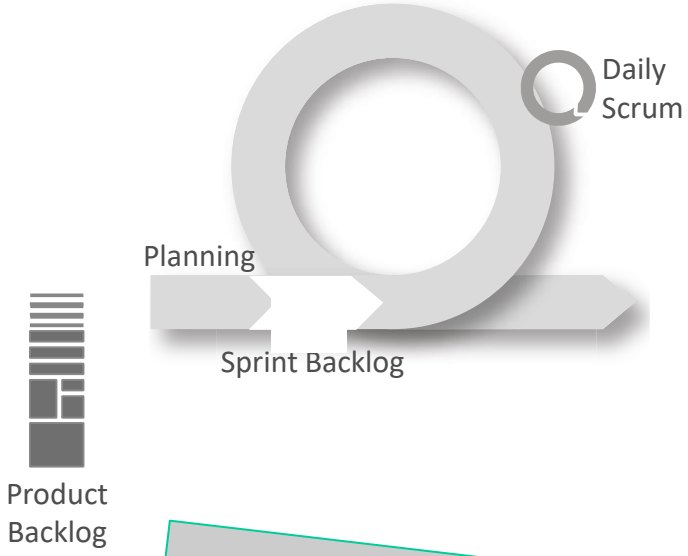
As <Role>, I want <Target>, so that I <Advantage>



Examples for Tasks:

- Creating a database table
- Creation of a layout for the login screen
- Creating an HTML page to login
- Creating a CSS file for HTML page
- Creating a Java script for login
- Test the table and its links
- Test the layout of the HTML page
- …

Daily Scrum – „the small planning meeting“

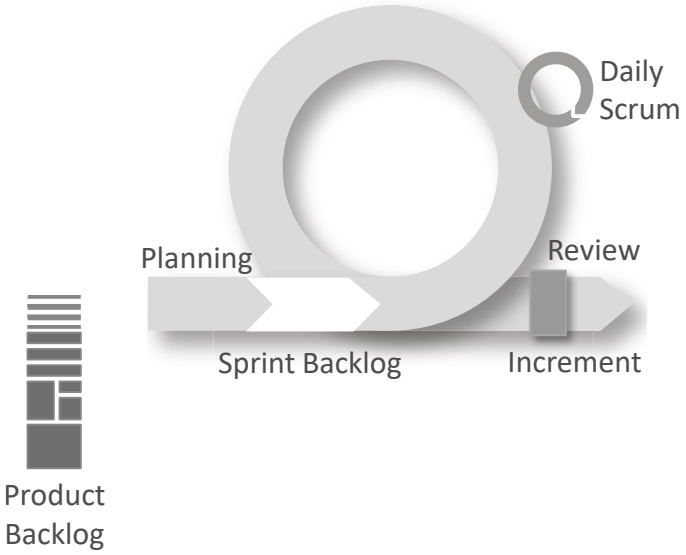


- daily meeting of the development team
- 15 minutes maximum
- Attendees:
  - Development Team
  - Scrum Master
  - possibly silent (!) guests
- Purpose
  - synchronization of the activities of the Development Team
  - checking the progress towards the sprint goal
  - helps the team achieve the sprint goal

Three questions to be answered by each member of the Development Team:

- *What have I achieved since the last Daily Scrum?*
- *What will I achieve by the next Daily Scrum?*
- *What obstacles do I see that could prevent me / us from reaching the sprint goal?*

Sprint Review



- The goal is to review what was created in the sprint (the product increment) and adjust the product backlog if necessary.
- Attendees:
  - Product Owner (also invites stakeholders)
  - Development Team
  - Scrum Master
  - important stakeholders

The sprint review is an informal meeting (no status report) in which the increment is demonstrated to the participants.

The demonstration of the increment is intended as a suggestion for feedback and as a basis for cooperation.

Elements of the Sprint Review

Product Owner

- invites stakeholders
- explains which product backlog entries are "done"
- represents the current status of the Product Backlog

Development Team

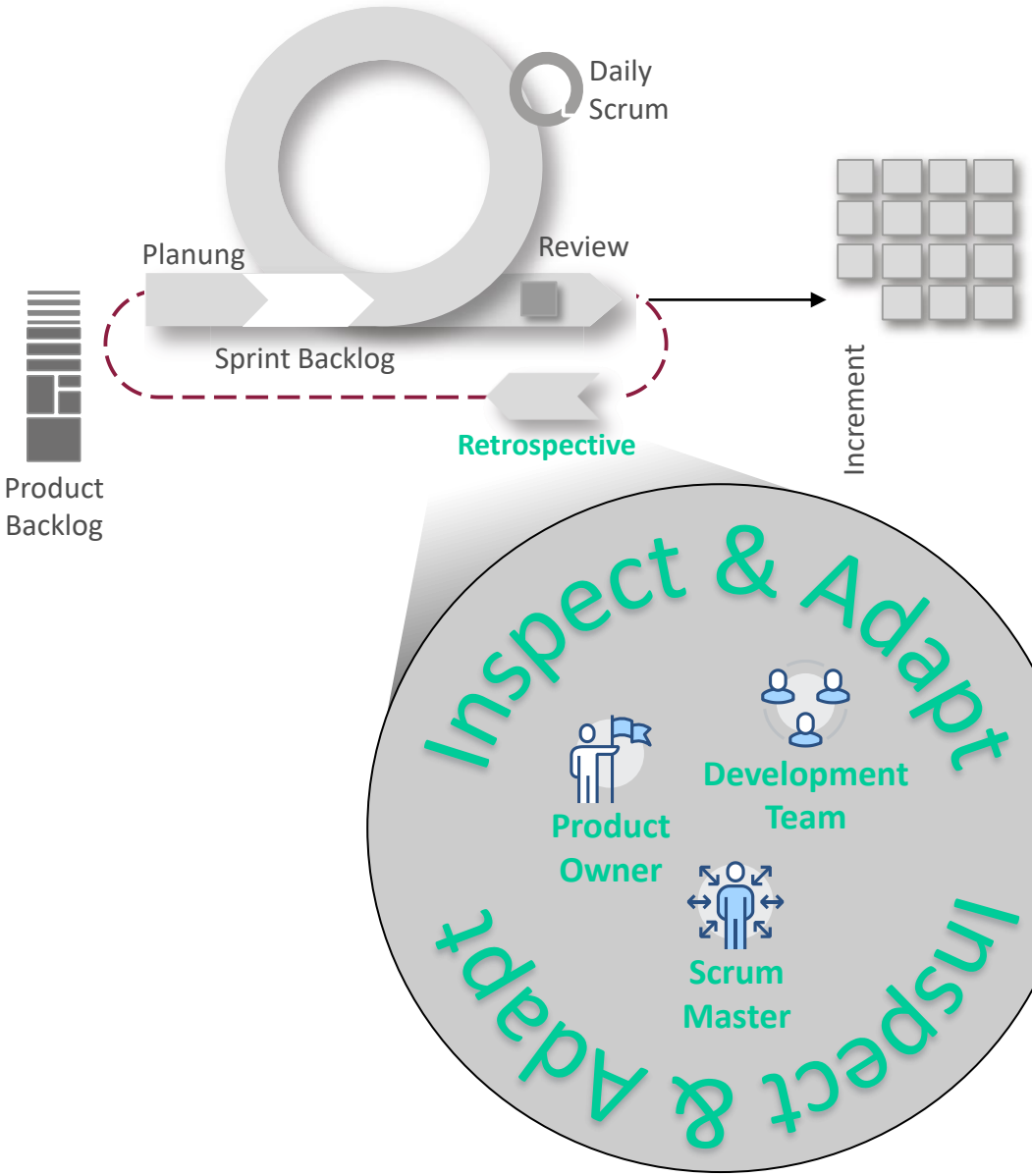
- shows what went well during the sprint, what problems occurred and how they were solved
- demonstrates the “done” work and answers questions

Scrum Master

Stakeholder

- Assessment of the current (market) situation about new knowledge
- all participants work together on what to do in the next sprint (=> valuable input for sprint planning)
- Review the schedule, budget and potential characteristics of the next expected product release

Sprint Retrospective



the aim is, …

- … to review how the sprint went in terms of people, relationships, processes, ways of working and tools.
- … recognize the most important elements that went well.
- … identify possible improvements and put them in order.
- … create a plan for implementing the improvements.

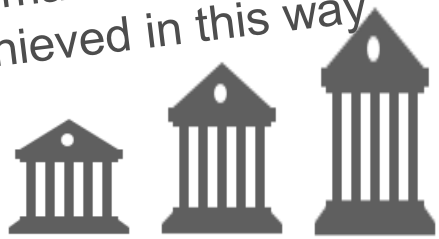
Estimation concepts

All Team Members

The estimates should be based on the expertise of all members of the Development Team.

use relative sizes

As a rule, better estimation results are achieved in this way



focus on sufficient correctness / accuracy, not precision

Estimates are not commitments

Estimating the workload

The development team estimates workload for backlog entries.

Estimations can be made in:

- **ideal times** (= Net working hours, without breaks and other interruptions)
  - ideal hours or
  - ideal days
- **Story Points**: relative units

Time / relative size-based Estimation

Specifying a time measure (= real time) results in the following problems:

- The given estimate indicate a precision they don't have.
- It seems that the workload could be calculated exactly.

Both result in unjustified expectations and unnecessary pressure in the development team.

The way out: estimate in story points

Story points are relative sizes

To determination the size of a story you need:

- one (or more) reference (s) as fixed point (s)
- a unit (=Story Point) that describes the metric we use to measure
- a scale to indicate a quantitative gradation
- periodic redefinition of the reference

Time / relative size-based Estimation

Number-based scales

- 2-power series: 1, 2, 4, 8, 16, 32, 64, 128 … 2^k
- Based on the Fibonacci series of numbers: 0, 1, 2, 3, 5, 8, 13, 20, 40, 100

The Fibonacci sequence describes u. a. numerous growth processes of plants and animals. It seems like it is a kind of growth pattern in nature.

[Quelle: Der goldene Schnitt, golden-section.eu, Dr. Dr. Ruben Stelzner in Zusammenarbeit mit Prof. Dr. Wolfgang Schad, abgerufen am 26. Oktober 2015]1

Other scales

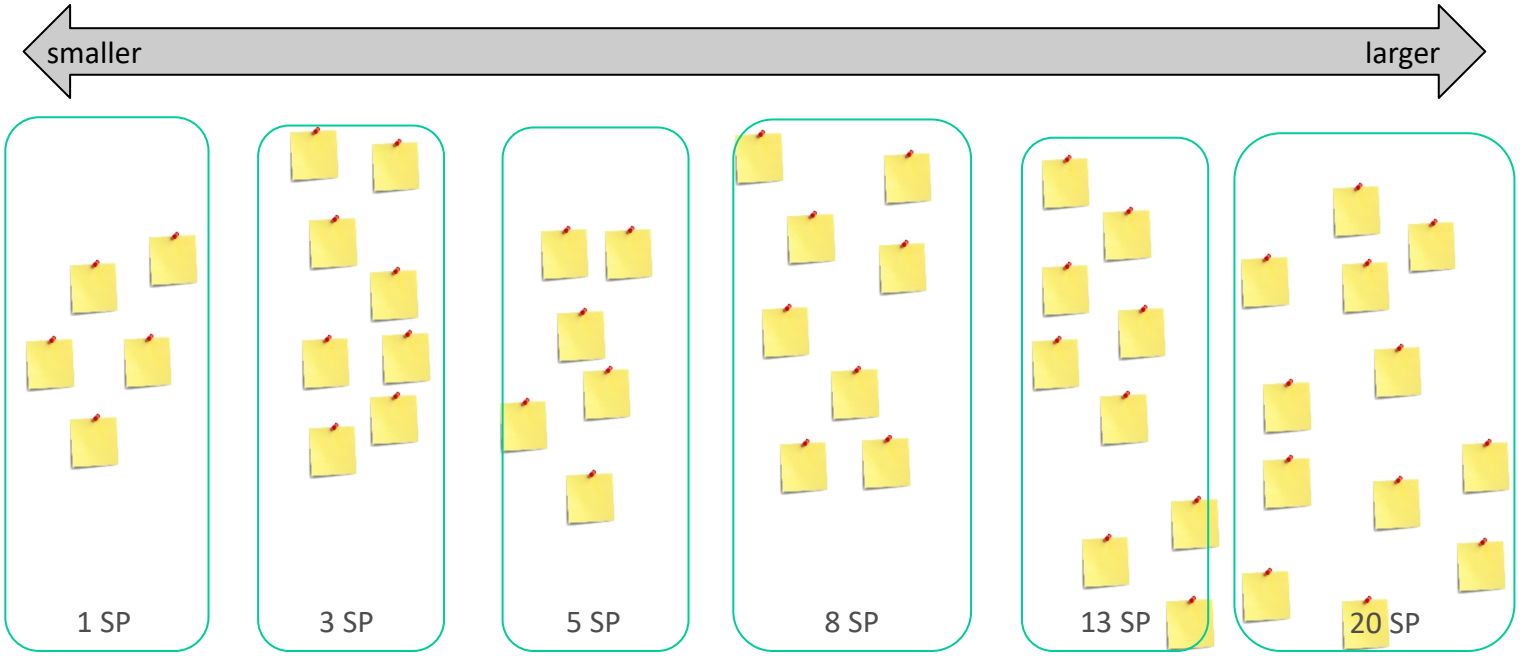
- T-shirt sizes: XXS – XS – S – M – L – XL – XXL

Note:

If many smaller stories are processed in a sprint, the overall estimation error is smaller than if a few large stories are processed!

Estimation method 1: Affinity Estimation

- **quick estimation of a large number (> 20) of user stories**
- Release planning or portfolio planning
- Preparation: List of user stories on post-its, sufficient space for grouping the stories
- Participants: Scrum team and, if necessary, other participants who can answer questions



Estimation method 1: Affinity Estimation

1. **Silent Sizing**: Sort quietly by relative size: Each member of the development team receives a set of stories. Stories that cannot be grouped through inquiries are put aside. (Time: approx. 5-20 minutes)

2. **Wikipedia-like Editing**: Discussion of "questionable" placements in the development team and with the product owner in order to arrive at a coordinated placement of the stories in the development team. (Time: approx. 20–60 minutes)

3. **Place Stories into Sizing Buckets**: Size assignment (clothing sizes, power of two or “Fibonacci numbers” etc.) to the individual story groups. (Time: approx. 10–30 minutes)

4. **Product Owner Review**: The Development Team can now take a break while the product owner looks at the result to see if he has any questions for the team. (Time: 15 minutes)

5. **Wrap-up**: The team answers the questions of the product owner and explains and / or revises his sizing decision.

6. **Documentation**: The result is documented by the Product Owner.

Estimation method 2: Planning Poker

- Each estimator receives a set of cards with one card value each.
- After a user story has been read out by the customer or product owner, it is briefly discussed in the team.
- Afterwards each appraiser evaluates the story for himself (!) and chooses a card.
- When each appraiser has made his assessment, the cards are revealed and the deviations are discussed.
- The rounds of estimates are repeated until the estimated values have sufficiently approximated.

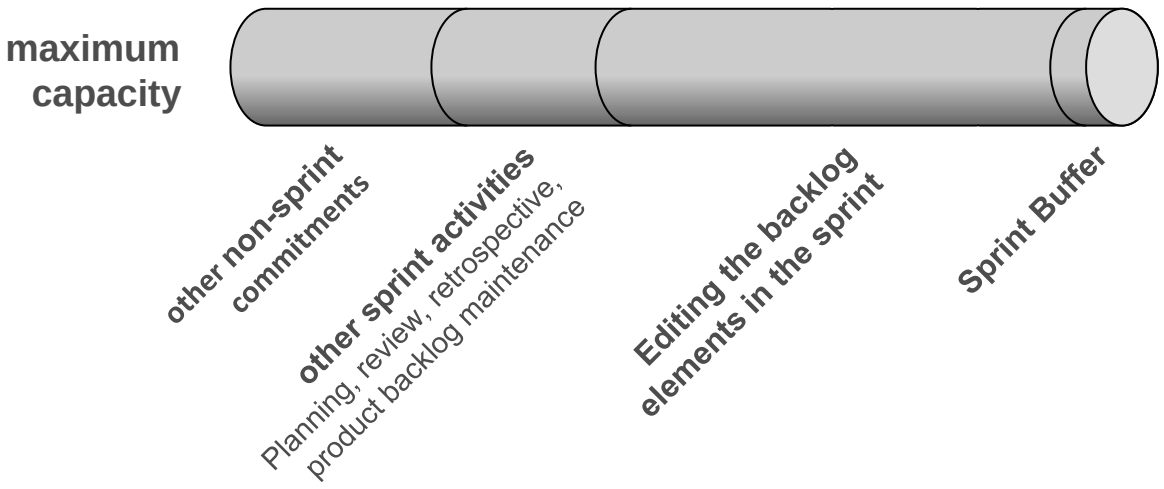
Schätzer	Runde 1	Runde 2	Runde 3
Lucy	3	5	5
Tim	8	8	5
Cathleen	20	10	8
Miles	5	8	5

Estimation of the Teams Capacity

Person	Working Hours / Day (h)		Net Working Time 20 Day Sprint (h)
	gross	net*	
Lucy	8	4	80
Miles	4	3	60
Hans	4	3	60
Bertram	8	6	120
Gabi	8	5	100
Ursel	8	6	120
Hans-Günther	4	2	40
total			**580**

How does a team know how many user stories can be processed in the next sprint?

Identify development Team capacity



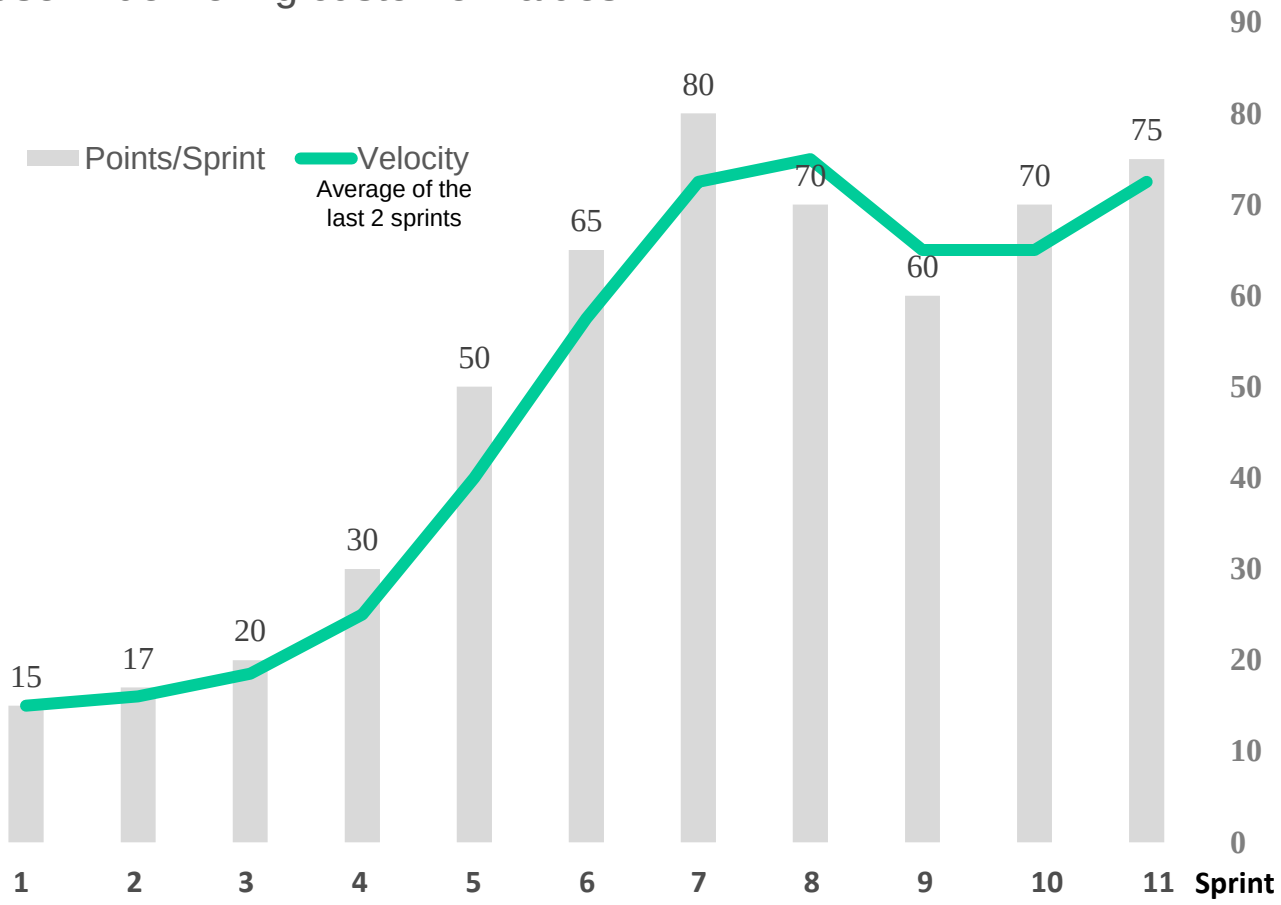
The **Sprint Puffe**r …

- … serves to catch up with the unforeseen.
- … can usually only be determined with increasing experience.

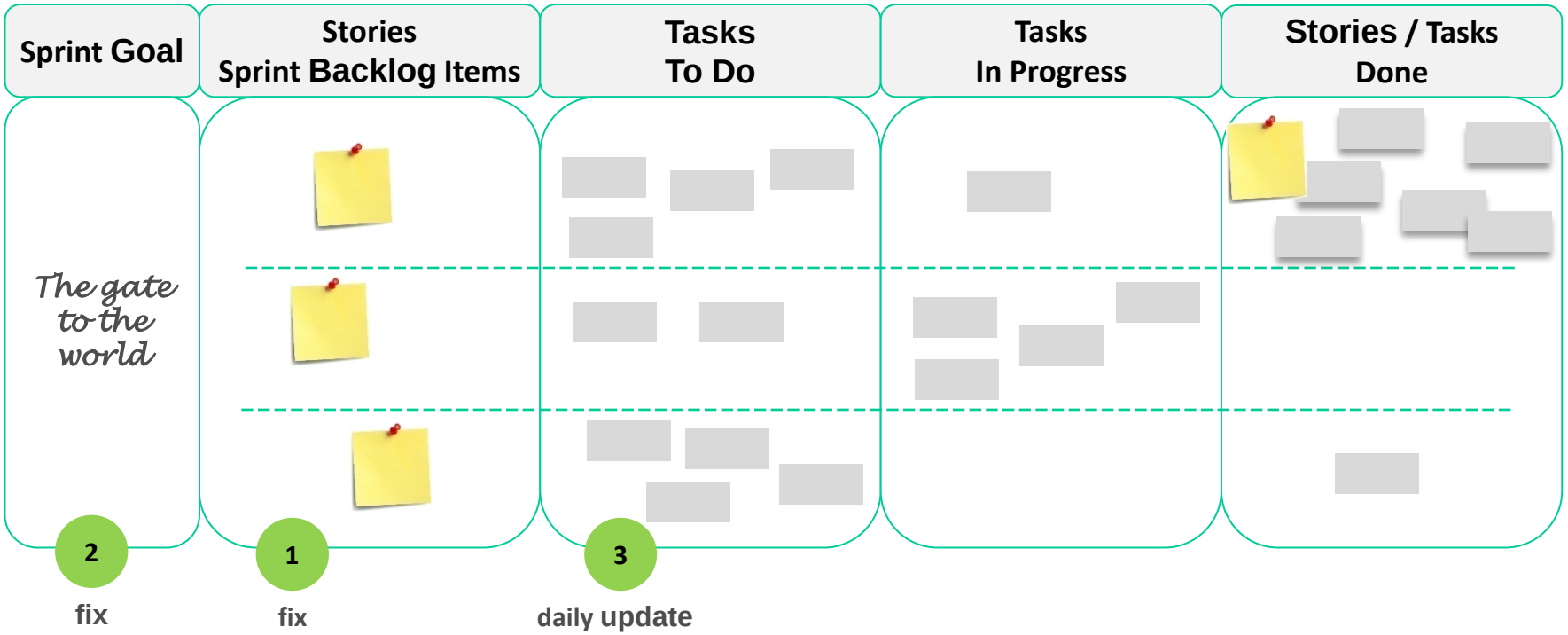
The Velocity

... of a Scrum team indicates the amount of work that was completed in a sprint.
... serves two important purposes :

1. It is the essential concept for Scrum planning.
2. It provides a yardstick by which the Scrum team can assess and improve its use in delivering customer values.



The Task Board



Summary

- Usage of different development methodologies and how Scrum evolved
- Roles in Scrum methodology
- Product Backlog and User Stories
- The Sprint
  - Sprint Planning
  - Daily Scrum
  - Sprint Review
  - Sprint Retrospective
- Estimation methods
- Scrum task board

References and literature

[GI, 2019] Gesellschaft für Information (2019): „Wie interagieren UX-Professionals mit ihrem Umfeld und ihren Kollegen“, GI

[Schweizer, 2003] Schweizer, Raffael (2003): „Agile Software Entwicklung mit Scrum“, https://files.ifi.uzh.ch/rerg/amadeus/teaching/seminars/seminar_ws0304/07_Schweitzer_Scrum_Folien.pdf

[Starke, 2002]: Starke, Gernot (2002): „Effektive Softwarearchitekturen“, Hanser-Verlag