Scrum Master Handguide

Document name: Scrum Master Handguide.docx
Last update: 16.09.2019
Autor: A. Balogh

Content

Introduction 3
Other Sources 6
Variants of Agile Software Development 6
Certification 6
Scrum 6
  Scrum-Framework 6
  Scrum Values 6
  Scrum Artifacts 7
Scrum Events 7
  Sprint Planning 8
  Daily Scrum 9
  Sprint Review 9
  Sprint Retrospective 9
Scrum-Team 10
  Stakeholder 10
  Management 10
  Sponsor 10
  Product Owner (PO) 10
  Scrum Master (SM) 11
  Development Team (Dev-Team) 12
Sprint 13
  Sprint Backlog 13
  Sprint Goal 13
  Sprint Cancellation 14
GLOSSARY 15
  Manifesto For Agile Software Development 15
  Burn-Down Chart 15
  Burn-Up Chart 15
  Candle Problem 16
  Cone of Uncertainty 16
  CFD - Cumulative Flows 16
  DoD - Definition of Done 16
  Dysfunction of a Team 16
  Increment / Product Increment 17
  Emergence 17
  Empiricism / Empirical Process 17
  Engineering standards 17
  Feedback Loops 17
  Forecast (of Functionality) 17
  Grooming 17
  Product Backlog 17
  Product Backlog Refinement 18
  Ready 18

RUP – Rational Unified Process 18
Self-Organization 18
Slack 18
Spike 18
Story 19
Task 19
Tracer Bullet 19
Velocity 19
Tools 20
Confluence 20
Jira 20
Mingle 20
Rally 20
RTC 20
Scrumban 20
TFS 20
Version One 20
Agile Softwareentwicklung 21
Agile Werte 21
Agiles Prinzip 21
Agile Methode 22
Agiler Prozess 22
DEFINITIONEN 24
SAFe - Scaled Agile Framework 24
Table of Figures 26
Index 27

Introduction

Links:

- ***Scrum*** is not a process or a technique for building products.
- ***Scrum*** is a framework for developing and sustaining complex products.
- ***Scrum*** consists of ***Scrum-Team***, ***roles***, ***events***, ***artifacts*** and ***rules***.
- Scrum is founded on ***empirical process control theory***, or ***empiricism***.
- Scrum is focused on maximize the Teams ***ability to deliver quickly***.
- The term Scrum was defined by ***Hirotaka Takeuchi*** and ***Ikujiro Nonaka*** in ***1986*** (see article "New New Product Development Game").
- ***Rugby*** approach.
- See also ***Ken Schwaber*** and ***Jeff Sutherland*** 1990.
- There ***is no centralized project management*** in Scrum, no Project Manager.
- Scrum ***increases*** the opportunity to ***control risk*** and ***optimizes*** the ***predictability of progress***.

Scrum ist ein Vorgehensmodell mit Meetings, Artefakten, Rollen, Werten und Grundüberzeugungen, das beim Entwickeln von Produkten im Rahmen agiler Softwareentwicklung hilfreich ist. Teammitglieder organisieren ihre Arbeit weitgehend selbst und wählen auch die eingesetzten Software-Entwicklungswerkzeuge und -Methoden. Ken Schwaber, Jeff Sutherland und Mike Beedle haben Scrum erfunden und etabliert. Als Software-Entwicklungsmethode wird Scrum das erste Mal in dem Buch *"Wicked Problems, Righteous Solutions"* beschrieben. Scrum in Produktionsumgebungen wird zum ersten Mal in dem Artikel *"The New New Product Development Game"* erläutert und später in *"The Knowledge Creating Company"* weiter ausgeführt von *Ikujiro Nonaka* und *Hirotaka Takeuchi*

Limitations:

- Teams whose members are geographically disperse or part-time.
- Teams whose members have very specialized skills.
- Products with many external dependencies.
- Products that are mature or legacy or with regulated quality control.

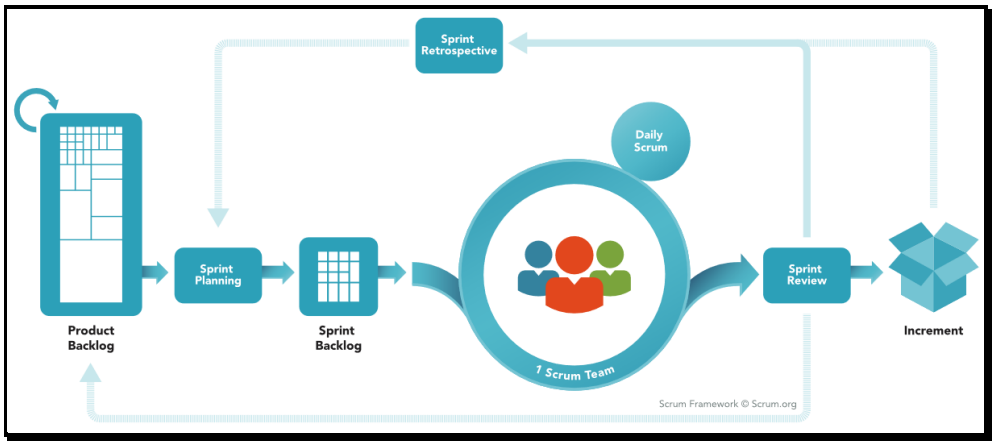


Figure 1: Scrum Framework I

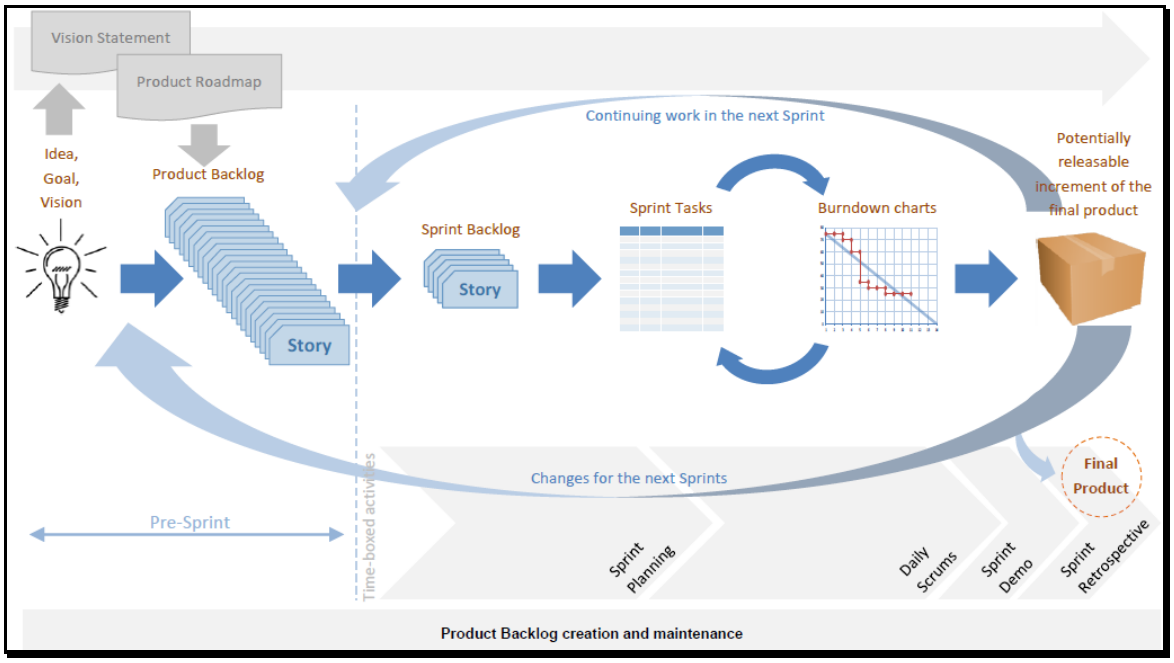


Figure 2: Scrum Framework II

When to use Scrum	When to use traditional methods
Scope is not clearly defined The product will gradually appear during the project	Scope is clearly defined upfront Clear product description is available upfront Similar projects were done before
Requirements change frequently Customer learns more about what they want as the project goes on	Requirements are well defined up front Few changes are expected during the project Requirements are not expected to change much
Activities cannot be well defined upfront Estimating (planning) is difficult	Activities can be well defined upfront Estimating is possible and reliable
Process is iterative (numerous cycles) Each cycle heavily depends on the previous one	Process is more long term Project might be split into phases
Success is mostly measured by customer satisfaction	Success is mostly measured by achieving the project goals for time, cost, scope…
Incremental results have value and can be used by users (put into production)	Users cannot normally start using the products until the project is complete (e.g. a bridge)

Figure 3: When to use Scrum?

Fiction	Fact
Developers are free to do what they want.	Developers work in a productive and predefined framework and the Scrum Master makes sure they are following Scrum.
Scrum gets rid of all paper work and allows the team to start developing right away.	There are certain planning steps involved in every Scrum project and development can only start when the Sprint Backlog has been defined.
All requirements (in the form of stories) must be agreed before the Development Team is allowed to start working on the product.	The Development Team can start working as soon as the initial stories of the Product Backlog are ready.
Scrum is very easy to implement, even without training.	Using Scrum is a big change; It might seem easy to implement Scrum compared to other project approaches, but people must still have a good understanding of Scrum to be able to run their projects well.
Scrum is just a set of simple rules.	Scrum is a set of rules and a framework, plus a compatible work culture and ethic.
The Scrum Master is like a project manager.	There is no one similar to a traditional project manager in a Scrum project. The Scrum Master makes sure the Scrum framework is followed.
Scrum does not require you to have a Business Case.	There should be a justified reason to spend any money in any company and this should be documented. The Product Owner is responsible for ensuring that there is a feasible reason for performing the project and aligning the project with it.
Scrum allows the Development Team to decide what will be delivered.	A Team only decides on how to deliver; it is up to the Product Owner to determine what will be delivered.
The Product Owner is the project manager.	The Product Owner only creates and maintains the Product Backlog, but does not manage the day to day activities of the Team.
Scrum tells us everything about managing projects.	Scrum mostly deals with the definition and delivery of the products. Many of the business oriented aspects of the project are done outside Scrum.
The Product Owner is a representative from the customer.	The Product Owner is one of the people from the performing organization (the organization in charge of producing the final product of the project; a contractor in many cases), and the contact point with the customer.

Figure 4: Fiction and Facts

Other Sources

Founder of Scrum:

- Ken Schwaber
- Jeff Sutherland

Daniel H. Pink	Author: Work Management, Behavioral Science A whole New Mind Drive To sell is Human Candle Problem
Patrick Lencioni	Author: Business Management
Lyssa Adkins Diana Larsen Geoff Watts John Kotter Holger Rathgeber	
Alex Armstrong	Comedian?

Variants of Agile Software Development

- Scrum
- Agile Unified Process (AUP)
- Dynamic Systems Development Model (DSDM)
- Extreme Programming (XP)

Certification

PSM 1

- People who have passed PSM I, achieving certification, demonstrate a fundamental level of Scrum mastery.
- PSM I certificate holders prove that they understand Scrum as described in the Scrum Guide and the concepts of applying Scrum.
- PSM I holders have a consistent terminology and approach to Scrum.

Scrum

Scrum-Framework

- Roles
- Events
- Artifacts
- Rules

Scrum Values

- **Focus**
  On what is the most important.
- **Respect**
  Cross-functioning, self-organized team.
- **Openness**
  Frequently inspecting through delivery
- **Courage**
  We admit we do not know everything.
- **Commitment**
  Dedicated to delivering working software

Scrum Artifacts

- Scrum's artifacts represent ***work*** or ***value*** to provide ***transparency*** and ***opportunities*** for ***inspection and adaption***.
- The ***Scrum Master*** must work with the Product Owner, Development Team and other involved parties to understand if the artifacts are completely ***transparent***.
- The ***Scrum Master*** is ***responsible*** for coping with incomplete artifact transparency.
- The Scrum Artifacts should be ***frequently inspected*** by the Scrum users to detect undesirable variances. The frequency should ***not get in the way of the work***.
- Scrum are specifically designed to ***maximize transparency of key information*** so that everybody has the ***same understanding of the artifact***.

The Scrum Artifacts are:

- Product Backlog
- Sprint Backlog
- Increment

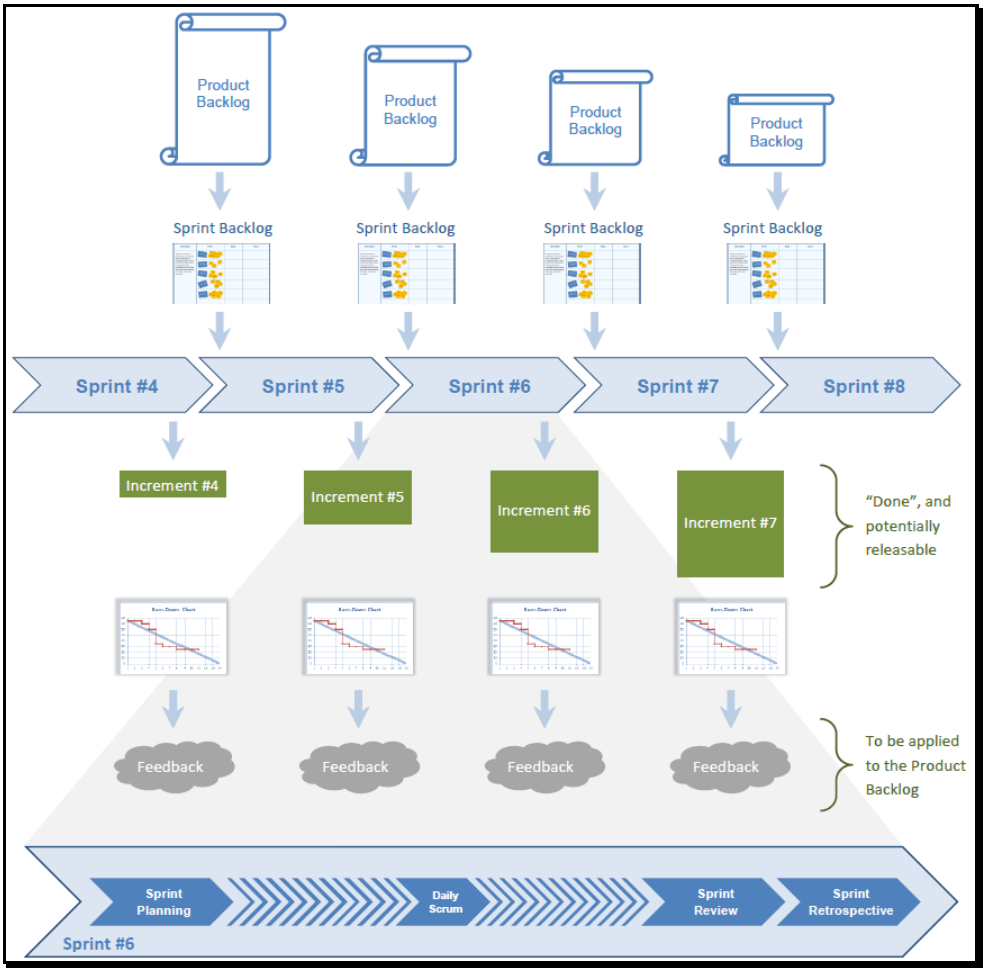


Figure 5: Scrum Artifacts

Scrum Events

- 4 Prescribed events are used in Scrum to create regularity and to minimize the need for meetings not defined in Scrum.
- All Events are ***time-boxed***, such that every event has a ***maximum duration***.

Scrum prescribes four formal events for inspection and adaption:

- Sprint Planning
- Daily Scrum

- Sprint Review
- Sprint Retrospective

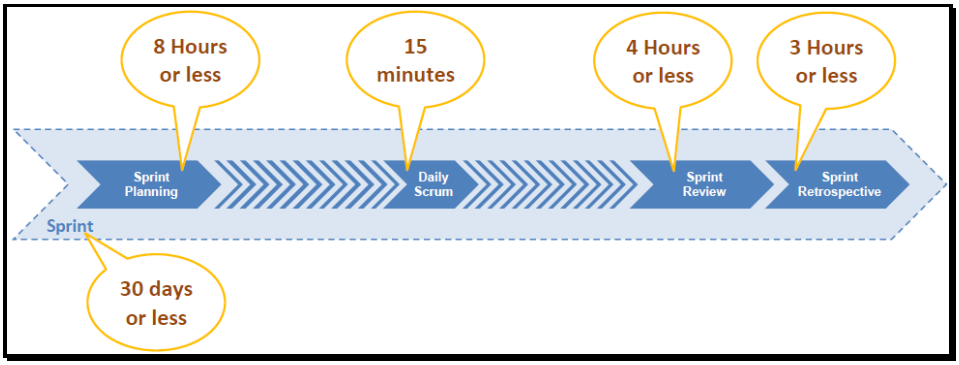


Figure 6: Events

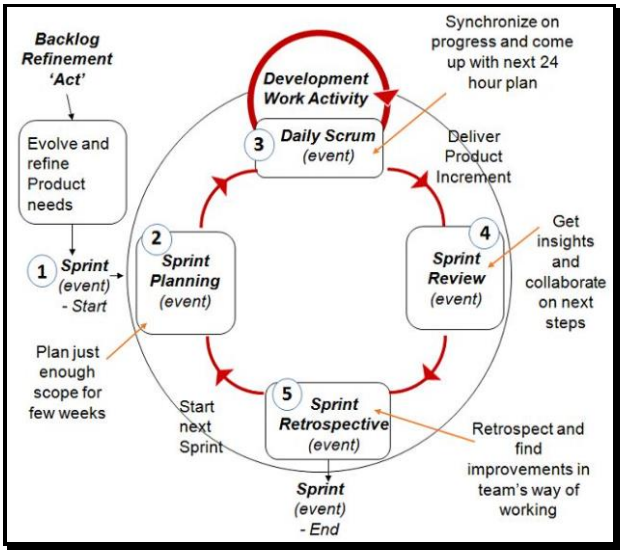


Figure 7: Scrum Events

Sprint Planning

- Collaborative work of the ***entire Scrum Team***.
- Sprint Planning is time-boxed to a maximum of ***eight hours*** for a one-month Sprint.
- The Development Team may also ***invite other people*** to attend in order to provide technical or domain advice.
- ***Selecting Product Backlog items*** that can be completed in one Sprint.
- The Scrum Guide requires ***only the work planned for the first days*** of the Sprint is decomposed by the end of the Sprint Planning.

Input:

- Product Backlog
- Latest Product Increment
- Project capacity of the Development Team during the Sprint
- Past performance of the Development Team

Process:

1. Creation of the ***Sprint Backlog*** during the first half of the Sprint Planning meeting.
2. Creation of a ***Collection of Tasks*** during the second half of the Sprint Planning meeting.

Target: What can be done and how to do it.

- ***What*** can be delivered in the Increment resulting from the upcoming Sprint.

- ***How*** will the work needed to deliver the Increment be achieved.
- Crafting a ***Sprint Goal***.
- By the end of the Sprint Planning, the ***Development Team should be able to explain*** to the Product Owner and Scrum Master how it intends to work as a self-organizing team to accomplish the Sprint Goal and create the anticipated ***Increment***.

Daily Scrum

- The Daily Scrum is a ***15-minute time-boxed event*** for the Development Team to synchronize activities and create a plan for the ***next 24 hours***.
- The Daily Scrum is held at the ***same time and place each day*** to reduce complexity.
- The Development Team chooses the ***starting time*** (recommendation at beginning of the day).
- ***Only Development Team members*** participate at the Daily Scrum.
- ***Updates*** are reflected in the Sprint Backlog.
- The ***Product Owner*** should not speak in Daily Scrum!

The 3 Questions (Process):

- ❑ What did I do yesterday that helped the Development Team meet the Sprint Goal?
- ❑ What will I do today to help the Development Team meet the Sprint Goal?
- ❑ Do I see any impediment that prevents me or the Development Team from meeting the Sprint Goal?

OUTCOM:

- The plan for next 24 hours, having the Sprint goal in mind.
- Updated Sprint Backlog.
- New impediments/updated impediments status.

Sprint Review

- A Sprint Review is held at the end of the Sprint to ***inspect the Increment*** and ***adapt the Product Backlog*** if needed and to figure out ***what to do next***.
- This is a ***four-hour time-boxed meeting*** for one-month Sprints.
- ***Incomplete work*** cannot be demonstrated.

Elements of the Sprint Review meeting:

- ❑ Attendees: Scrum Team and key stakeholders invited by the Product Owner.
- ❑ The ***Product Owner*** explains:
  - What Product Backlog items have been “Done”
  - What has not been “Done”.
  - Discusses the Product Backlog as it stands.
  - He projects likely completion dates based on progress to date (if needed).
- ❑ The ***Development Team***:
  - Discusses what went well during the Sprint, what problems it ran into, and how those problems were solved.
  - Demonstrates the work that it has “Done” and answers questions about the Increment.
- ❑ The ***Entire Group***:
  - Collaborates on what to do next, so that the Sprint Review provides valuable input to subsequent Sprint Planning.
  - Review of how the marketplace or potential use of the product might have changed what is the most valuable thing to do next.
  - Review of the timeline, budget, potential capabilities, and marketplace for the next anticipated release of the product.

Result:

- Revised Product Backlog, as input to the next Sprint.

Sprint Retrospective

- This is a ***three-hour time-boxed meeting*** for one-month Sprints.
- The Sprint Retrospective is an opportunity for the Scrum Team to inspect itself and create a plan for ***improvements to be enacted during the next Sprint***.
- The Sprint Retrospective occurs ***after the Sprint Review*** and ***prior the next Sprint Planning***.
- During the Sprint Retrospective, the ***DoD*** is reviewed and changed if necessary.

Purpose:

- ❑ Inspect how the last Sprint went with regards to people, relationships, process, and tools.
- ❑ Identify and order the major items that went well and potential improvements.
- ❑ Create a plan for implementing improvements to the way the Scrum Team does its work.

Scrum-Team

- Scrum Teams are ***self-organized*** and ***cross-functional***.
- The team is designed to optimize ***flexibility***, ***creativity*** and ***productivity***.
- Proposed working hours for a Scrum Team member is ***7 – 8 hours per day***.

Consists of:

- Product Owner
- Development Team
- Scrum Master

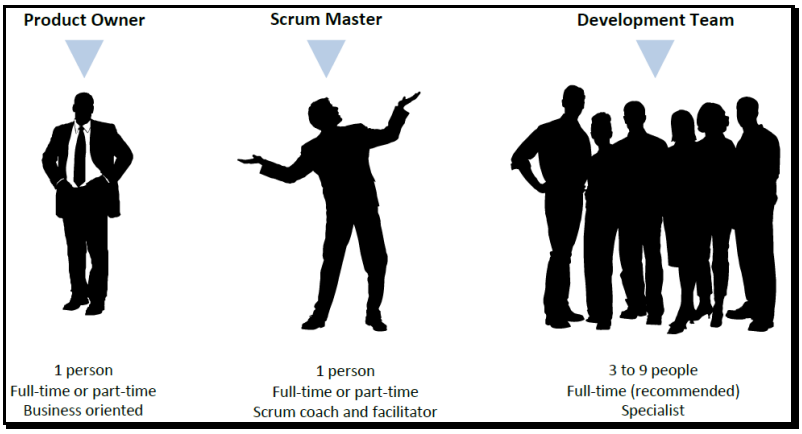


Figure 8: The Scrum Team

Stakeholder

- Are roles ***outside Scrum***.
- A person ***external to the Scrum Team*** with a specific interest in and knowledge of a product that is required for incremental discovery.
- ***Represented by the Product Owner*** and actively engaged with the Scrum Team at Sprint Review.
- The Key Stakeholders are allowed to participate ***only in the Sprint Review meeting***.

Management

- Management has ***no active role*** in the actual product development through Scrum.
- However, management external to the Scrum team is incredible ***important in setting the vision and strategy*** to guide the overall direction of the organization.

Sponsor

- A ***Business Representative*** with ***budget power***.

Product Owner (PO)

- Is the ***Product CEO***.
- The Product Owner is ***one person***, not a committee.
- The ***Product Owner*** is responsible for ***maximizing the value of the product*** and the ***work of the Development Team***.
- The sole person ***responsible*** for managing the ***Product Backlog***.
- To succeed, the entire organization must ***respect his or her decisions***.
- Represents the ***product's stakeholders*** and the ***voice of the customer***.
- Empowered to ***Make Decisions about the product***.

- ***Available to the Development Team*** to answer questions.
- ***Product Value Maximizer***
- ***Lead facilitator of Key Stakeholder involvement***.
- ***Product Marketplace expert***.
- ***Tracks/monitors Total Work Remaining*** at least every Sprint Review. He compares this amount with work remaining at previous Sprint Reviews to assess progress toward completing projected work by the desired time for the goal. This information is made transparent to all stakeholders.

Tasks:

- Demonstrates the solution to key stakeholders who were not present at a Sprint Review.
- Defines and announces releases.
- Communicates team status.
- Organizes milestone reviews.
- Educates stakeholders in the development process.
- Negotiates priorities, scope funding and schedule.
- Ensures that the Product Backlog is visible, transparent and clear.
- Tracks total work remaining.

Attributes

- Empathy
- Communicate effectively

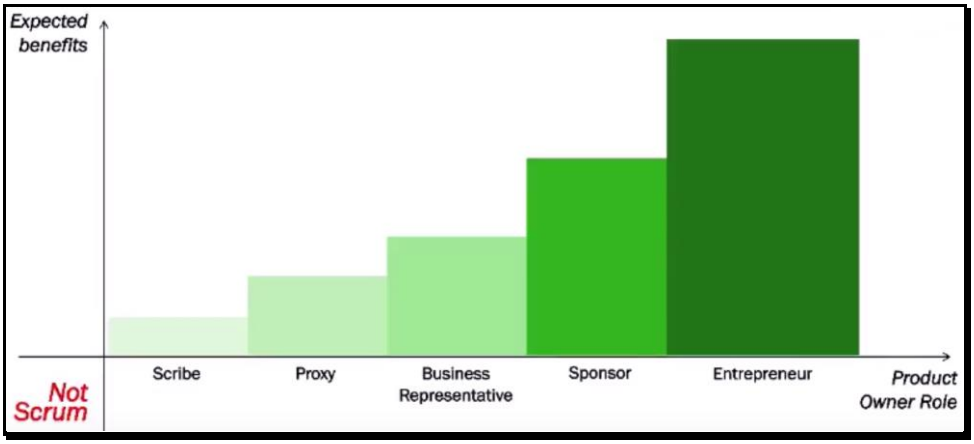


Figure 9: Product Owner

Scrum Master (SM)

- ***Management Role***.
- The Scrum Master is responsible for ensuring Scrum is ***understood and enacted***.
- The Scrum Master teaches the Development Team to ***keep the meetings within the defined time-boxes***.
- The Scrum Master does not have to attend the Daily Scrum, he must assure that the ***Development Team participates***.
- ***Accountable*** for removing impediments to the ability of the team to deliver the product goals and deliverables.
- Does not have ***people management responsibilities***.
- The role within a Scrum Team accountable for ***guiding***, ***coaching***, ***teaching*** and ***assisting*** a Scrum Team and its environments in a proper understanding and use of Scrum.
- Scrum Masters understand the impact of the ***transparency*** that Scrum brings and requires, and how Scrum is best used ***to increase an organization's agility***.
- Scrum Masters have the ***tools*** and ***ideas*** to get Scrum teams going, guide them along the way and help them continuously self-develop to become successful through ***better collaboration***.

Scrum Master Service to the Product Owner:

- Finding techniques for effective Product Backlog management.
- Helping the Scrum Team understand the need for clear and concise Product Backlog items.
- Understanding product planning in an empirical environment.

- Ensuring the Product Owner knows how to arrange the Product Backlog to maximize value.
- Understanding and practicing agility
- Facilitating Scrum events as requested or needed.

Scrum Master Service to the Development Team:

- Coaching the Development Team in self-organization and cross-functionality.
- Helping the Development Team to create high-value products.
- Removing impediments to the Development Team's progress.
- Facilitating Scrum events as requested or needed
- Facilitation and removing impediments serves a team in achieving the best productivity possible.
- Coaching the Development Team in organizational environments in which Scrum is not yet fully adopted and understood.
- Servant-leader for the Development Team.

Scrum Master Service to the Organization:

- Leading and coaching the organization in its Scrum adoption.
- Planning Scrum implementations within the organization.
- Helping employees and stakeholders understand and enact Scrum and empirical product development.
- Causing change that increases the productivity of the Scrum Team.
- Working with other Scrum Masters to increase the effectiveness of the application of Scrum in the organization.

Development Team (Dev-Team)

- The Development Team consists of professionals who do the work of delivering a potentially releasable Increment of ***"Done"*** product at the end of each ***Sprint***.
- Are ***self-organized*** and ***cross-functional***.
- All ***Development Team members*** are ***Developers***.
- Scrum recognizes ***no sub-teams***.
- ***Accountability*** belongs to the Development Team as a whole.
- ***Development Team Size*** should be between ***3 - 9 members***. ***Product Owner*** and ***Scrum Master*** roles are not included in the count.
- Development Team members ***may change as needed*** while taking into account a ***short-term reduction in productivity***.
- The Development Team is ***responsible for managing the progress of work*** during a Sprint.
- Responsible for delivering ***Potentially Shippable Increments (PSIs)*** at the end of each Sprint.
- ***Accountable*** for ***managing***, ***organizing*** and ***doing*** all development work required to create a releasable Increment of product every Sprint.
- ***Responsible for the estimates in the Product Backlog.***

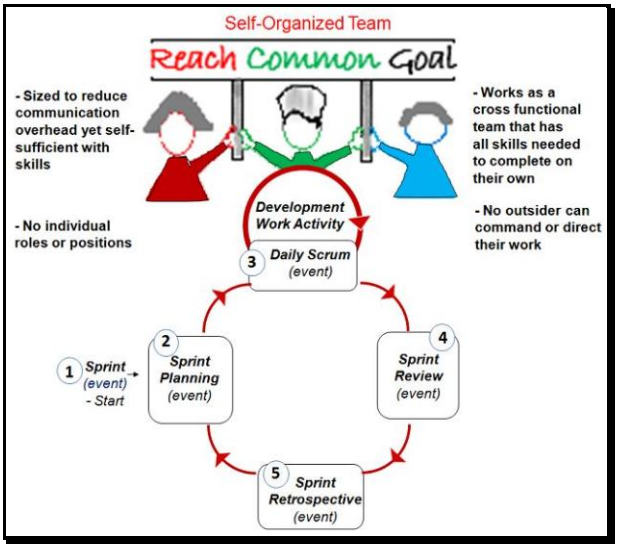


Figure 10: Development Team

Sprint

- The heart of Scrum is a Sprint, a ***time-box*** of ***one month or less*** during which a "Done", useable and potentially releasable product increment is created.
- The length of a Sprint should keep the ***business risk acceptable*** to the Product Owner.
- ***Short enough*** to be able to synchronize the development work with ***other business events***.
- A ***new Sprint*** starts immediately after the conclusion of the previous Sprint.
- Once a ***Sprint*** begins, its duration is fixed and cannot be shortened or lengthened.
- Each Sprint may be considered a ***project*** with no more than a one-month horizon.
- ***During every Sprint useable*** and ***potentially releasable Product Increment*** is created.
- During the Sprint, scope may be clarified and re-negotiated between the ***Product Owner*** and ***Development Team***.
- Sprints are done consecutively, ***without intermediate gaps***.
- Scrum does not require having aligned ***Sprint lengths*** for multiple teams.

Sprint contains:

- Sprint Planning
- Daily Scrums
- Development work
- Sprint Review
- Sprint Retrospective

Figure 11: Sprint

Sprint Backlog

- Managed by the ***Development Team***.
- Only the ***Development Team*** can change its Sprint Backlog during a Sprint.
- The Development Team ***modifies*** the Sprint Backlog throughout the Sprint and the Sprint Backlog ***emerges during the Sprint***.
- The ***Product Backlog*** items selected for this ***Sprint*** plus the ***plan for delivering them*** is called the ***Sprint Backlog***.
- ***Set of Product Backlog items*** selected for the Sprint, ***plus*** a ***plan for delivering the Product Increment*** and realizing the Sprint Goal.
- Ii is a ***forecast*** by the Development Team about what functionality will be in the next Increment and the work needed to deliver that functionality into a "Done" Increment.
- Makes visible all the ***Tasks*** that the Development Team identifies as necessary to meet the Sprint Goal.
- At any point in time in a Sprint, the total work remaining in the Sprint Backlog can be ***summed***.
- These stories (features, functionalities, or deliverables) make up the Sprint Backlog, so the Sprint Backlog is a ***list of all stories*** that will be developed in the next Sprint.
- The Sprint Backlog consist of a Collection of ***Tasks***.

Sprint Goal

- The ***Scrum Team is crafting the Sprint Goal*** at the Sprint Planning.
- The Sprint Goal is an objective set for the Sprint that can be met through the implementation of ***Product Backlog items***.
- At any time, the total work remaining to ***reach a Goal*** can be summed.
- The ***Product Owner*** tracks total work remaining at least every Sprint Review.

This is a sample Sprint Goal:

We are going to enable all the essential parts of the website store to set up a complete purchase process. This makes other features of the website more meaningful to the customer.

Figure 12: Sample Sprint Goal

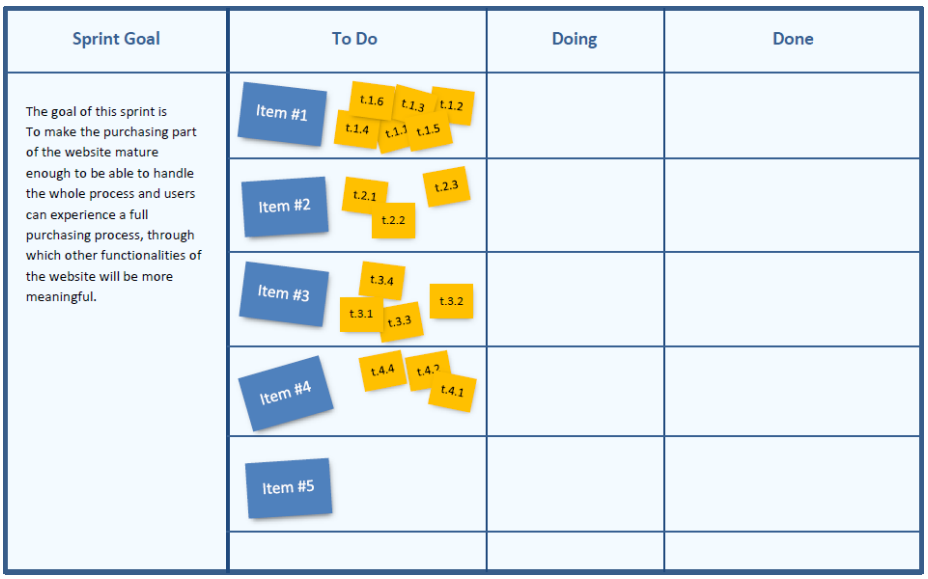


Figure 13: Sprint Goal Management

Sprint Cancellation

- Only the ***Product Owner*** has the authority to cancel a Sprint.
- A Sprint would be cancelled if the ***Sprint Goal*** becomes ***obsolete***. This might occur if the ***company changes direction*** or if ***market or technology conditions change***.
- When a Sprint is cancelled, ***any completed and "Done" Product Backlog items are reviewed***.
- If part of the work is potentially releasable, the Product Owner typically ***accepts it***.
- All incomplete Product Backlog Items are ***re-estimated and put back on the Product Backlog***.

GLOSSARY

Manifesto For Agile Software Development

MANIFESTO FOR AGILE SOFTWARE DEVELOPMENT

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.

PRINCIPLES OF THE AGILE MANIFESTO

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Burn-Down Chart

- The Sprint Burn-Down Chart is a public displayed chart showing ***remaining work/work hours*** in the Sprint Backlog.
- Update, every day.
- A chart showing the ***evolution*** of ***remaining effort against time***.
- Burn-down charts are an optional implementation within Scrum ***to make progress transparent.***
- The theory states that each member of the ***Development Team*** updates the Chart, but in real life it's usually the ***Scrum Master***.

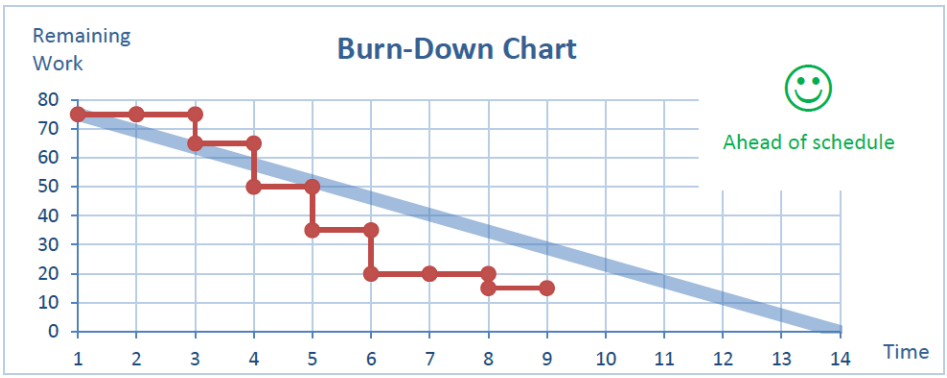


Figure 14: Burn-Down Chart

Burn-Up Chart

- A chart showing the ***evolution*** of an ***Increase in a measure against time***.
- Burn-up charts are an optional implementation within Scrum ***to make progress transparent***.

Candle Problem

See: Daniel H. Pink

1. As long as a task involves only mechanical skill, bonus works as it would be expected: "The higher the pay, the better the performance".
2. Once the task called for "even rudimentary cognitive skill", a larger reward "led to poorer performance".
3. In eight of nine tasks we examined across the three experiment: "higher incentives lead to worse performance".

Autonomy
Mastery
Purpose

Cone of Uncertainty

- Describes the evolution of the ***amount of uncertainty*** during a project.

CFD - Cumulative Flows

- Management report.
- It gives you an “at a glance” picture of key process variables such as velocity, WIP and ticket cycle times.
- It can help you release more features faster by identifying bottlenecks and problems in your development process.

DoD - Definition of Done

- Managed by the ***Development Team***.
- Work ***completed*** and ***transparent***, this is Done.
- A ***shared understanding*** of expectations that software must live up to in order to be releasable into production.
- To ***assess*** when work is completed on the Product Increment.
- ***Guides*** the Development Team in knowing ***how many Product Backlog items*** it can select during a Sprint Planning.
- Assures ***artifact transparency.***
- ***Conventions***, ***standards*** and ***guidelines of the Organization*** have to be taken in account.

Dysfunction of a Team

See: Patrick Lencioni

Figure 15: Dysfunction of Teams

Dysfunction #1: Absence of Trust

The fear of being vulnerable with team members prevents the building of trust within the team.

Dysfunction #2: Fear of Conflict
The desire to preserve artificial harmony stifles the occurrence of productive ideological conflict.

Dysfunction #3: Lack of Commitment
The lack of clarity or buy-in prevents team members from making decisions they will stick to.

Dysfunction #4: Avoidance of Accountability
The need to avoid interpersonal discomfort prevents team members from holding one another accountable.

Dysfunction #5: Inattention to Results
The pursuit of individual goals and personal status erodes the focus on collective success.

Increment / Product Increment

- The Product Increment should be usable and releasable at the end of every Sprint, but it does not have to be released.
- The ***Increment*** is the sum of all Product Backlog items completed during a Sprint and the value of the Increments of all previous Sprints.
- A ***piece of working software*** that adds to previously created Increments, where the sum of all Increments - as a whole - form a ***product***.

Emergence

- The process of the coming into existence or prominence of ***new facts*** or ***new knowledge*** of a fact, or knowledge of a fact becoming visible unexpectedly.

Empiricism / Empirical Process

- Process control type in which ***only the past is accepted as certain*** and in which decisions are based on ***observation***, ***experience*** and ***experimentation***.

The three pillars of Empiricism:

- Inspection
- Transparency
- Adaption

Engineering standards

- A shared ***set of development and technology standards*** that a Development Team applies to create releasable Increments of software.

Feedback Loops

- Daily Scrum, Sprint Review and Sprint Retrospective are typical feedback loops in Scrum.

Forecast (of Functionality)

- The ***selection of items*** from the Product Backlog a Development Team deems feasible for implementation in a Sprint.

Grooming

- Due to the increasingly negative connotation of the word grooming, this activity is increasingly known as ***Backlog Refinement*** or ***Backlog Management.***

Product Backlog

- Managed by the ***Product Owner***.
- ***Development Team*** is responsible for all ***estimates*** in the Product Backlog.
- Is the ***single source*** of requirements for any changes to be made to the product.
- The Product Backlog is an ***ordered list*** of everything that might be needed in the product and is the single source of requirements for any changes to be made to the product.

- The Product Backlog Is ***never complete, it's dynamic***.
- The Product Backlog ***evolves*** as the product and the environment in which it will be used evolves.
- The Product Backlog lists all ***features***, ***functions***, ***requirements***, ***enhancements*** and fixes that constitute the changes to be made to the product in future releases.
- Contains also ***Non-Functional Requirements (NFRs)***.
- ***Less valuable*** and ***most unclear items*** are at the bottom.
  The Product Owner decides based upon: ***Value, risk, priority and necessity***
- The ***Product Owner*** is responsible for the ***content***, ***availability*** and ***ordering***.

Product Backlog Management:
- Clearly expressing Product Backlog items.
- Ordering the items in the Product Backlog to best achieve goals and missions.
- Optimizing the value of the work the Development Team performs.
- Ensuring that the Product Backlog is visible, transparent and clear to all and shows what the Scrum Team will work on next.
- Ensuring the Development Team understands items in the Product Backlog to the level needed.

Product Backlog Refinement
- The activity in a Sprint through which the Product Owner and the Development Team ***add granularity to the Product Backlog***.
- Refinement shouldn't consume more than ***10%*** of the capacity of the Development Team.

Ready
- A ***shared understanding*** by the ***Product Owner*** and the ***Development Team*** regarding the preferred level of description of Product Backlog items introduced at Sprint Planning.
- Product Backlog items that ***can be "Done"*** by the Development Team within one Sprint are deemed ***"Ready" for selection in a Sprint Planning***.

RUP – Rational Unified Process
- Is an iterative software development process framework.
-

Self-Organization
- The management principle that teams autonomously organize their work.
- Self-organization happens within boundaries and against given goals.
- Teams choose how best to accomplish their work, rather than being directed by others outside the team.

Slack
- It does not matter how much we work.
- What we produce is important.
- We should be product-oriented, rather than activity-oriented.
- One way of being productive, is to limit the work time to a reasonable amount, and have frequent off times.
- That is why it is recommended (but not necessary) to have a slack between each two Sprints.
- Let’s have a day or two off to recharge your batteries, read some relevant articles, and check out what other teams are doing.

Spike
- A time boxed period used to research a concept or create a simple prototype.

Story

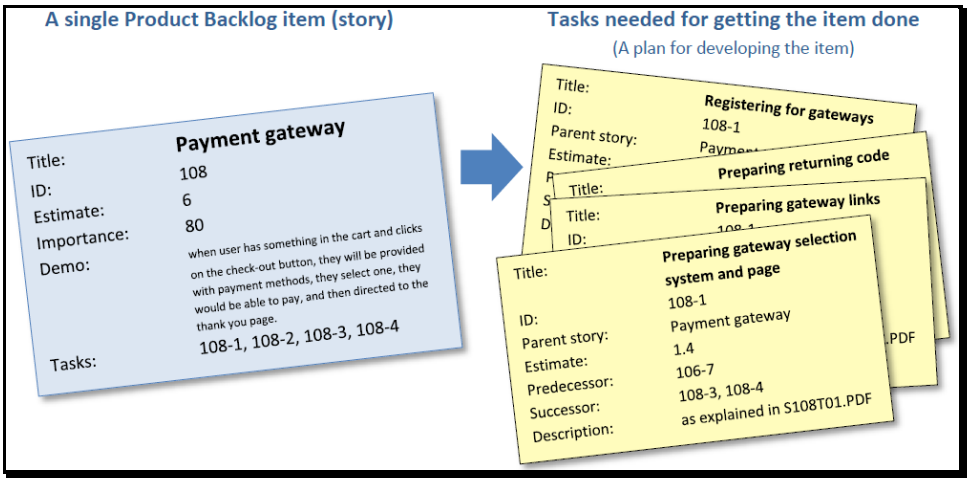


Figure 16: Story

Task

- An item in the Sprint Backlog.

Tracer Bullet

Velocity

- The total effort a team is capable of in a Sprint.
- Indication of the average amount of Product Backlog turned into an Increment of product during a Sprint by a Scrum Team, ***tracked by the Development Team*** for use within the Scrum Team.

Tools

Confluence

- Collaboration Software

Jira

- ???

Mingle

- ???

Rally

- ???

RTC

- ???

Scrumban

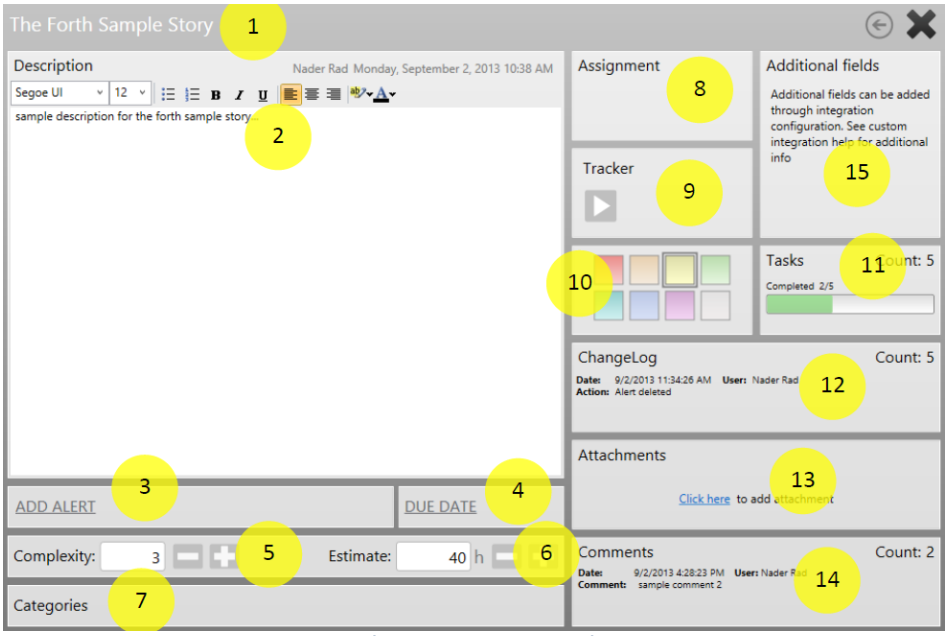


Figure 17: Scrum Tool

TFS

- ???

Version One

- ????

Agile Softwareentwicklung

Agile Softwareentwicklung
Im Kern geht es bei agiler Softwareentwicklung um möglichst häufige Rückkopplungsprozesse und zyklisches (iteratives) Vorgehen auf allen Ebenen: bei der Programmierung, im Team und beim Management.
Die entstehende Software wird so schnell und so oft wie möglich vom Kunden bewertet. Schliesslich kann nur der Kunde beurteilen, was er wirklich benötigt.
Anders als in der klassischen Vorgehensweise bietet agiles Vorgehen die Möglichkeit, noch während des Entwicklungsprozesses bereits definierte Anforderungen zu ändern und neue Anforderungen hinzuzufügen.

Abbildung 1: Agile Softwareentwicklung

Agile Werte

Englisch

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

Deutsch

- **Menschen und Interaktionen** stehen über Prozessen und Werkzeugen
- **Funktionierende Software** steht über einer umfassenden Dokumentation
- **Zusammenarbeit mit dem Kunden** steht über der Vertragsverhandlung
- **Reagieren auf Veränderung** steht über dem Befolgen eines Plans

Das heisst, obwohl wir die Werte auf der rechten Seite wichtig finden, schätzen wir die Werte auf der linken Seite höher ein.“

Agiles Prinzip

Ein *Agiles Prinzip* ist ein Leitsatz für die agile Arbeit.

Manchmal werden agile Prinzipien auch als Methode bezeichnet. Die eigentlich falsche Verwendung des Begriffs *Methode* für Prinzipien wird von den Vertretern agiler Methoden bewusst eingesetzt.
Zufriedenstellung des Kunden durch frühe und kontinuierliche Auslieferung von wertvoller Software

- Agile Prozesse nutzen Veränderungen (selbst spät in der Entwicklung) zum Wettbewerbsvorteil des Kunden.
- Lieferung von funktionierender Software in regelmäßigen, bevorzugt kurzen Zeitspannen (wenige Wochen oder Monate)

- Nahezu tägliche Zusammenarbeit von Fachexperten und Entwicklern während des Projektes (Bsp.: Gemeinsamer Code-Besitz (Collective Code Ownership))
- Bereitstellung des Umfeldes und der Unterstützung, welche von motivierten Individuen für die Aufgabenerfüllung benötigt wird
- Informationsübertragung nach Möglichkeit im Gespräch von Angesicht zu Angesicht
- Als wichtigstes Fortschrittsmaß gilt die Funktionsfähigkeit der Software
- Einhalten eines gleichmäßigen Arbeitstempos von Auftraggebern, Entwicklern und Benutzern für eine nachhaltige Entwicklung
- Ständiges Augenmerk auf technische Exzellenz und gutes Design
- Einfachheit ist essenziell (KISS-Prinzip)
- Selbstorganisation der Teams bei Planung und Umsetzung
- Selbstreflexion der Teams über das eigene Verhalten zur Anpassung im Hinblick auf Effizienzsteigerung

Agile Methode

- Streng genommen bezeichnet *agile Methode* eine an Agilität ausgerichtete Methode zur Softwareentwicklung.
- Ein Kennzeichen agiler Methoden ist, dass sie in einem Prozess dazu dienen können, die Aufwandskurve möglichst flach zu halten. Als Leitsatz gilt: Je mehr du nach Plan arbeitest, desto mehr bekommst du das, was du geplant hast, aber nicht das, was du brauchst. Daraus resultieren einige Prinzipien des *agile Modelling* und des *extreme Programmings.*
- Agile Methoden lassen sich in zwei Gruppen unterteilen: die tatsächlichen Methoden und die den Methoden zu Grunde liegenden Prinzipien.

Beispiele für agile Methoden:

- **Paarprogrammierung**
- **Testgetriebene Entwicklung**
- **ständige Refaktorierungen**
- **Story-Cards**
- **schnelle Codereviews**

Agiler Prozess

Ziel der Vorgehensweise ist es, den Softwareentwicklungsprozess durch Abbau der Bürokratie und durch die stärkere Berücksichtigung der menschlichen Aspekte effizienter zu gestalten.
Den meisten agilen Prozessen liegt zu Grunde, dass sie versuchen, die reine Entwurfsphase auf ein Mindestmass zu reduzieren und im Entwicklungsprozess so früh wie möglich zu ausführbarer Software zu gelangen, die dann in regelmässigen, kurzen Abständen dem Kunden zur gemeinsamen Abstimmung vorgelegt werden kann. Auf diese Weise soll es jederzeit möglich sein, flexibel auf Kundenwünsche einzugehen, um so die Kundenzufriedenheit insgesamt zu erhöhen. Sie stellen damit einen Gegensatz zu den klassischen Vorgehensmodellen wie dem V-Modell oder dem Rational Unified Process (RUP) dar.

Allen agilen Prozessen ist gemeinsam, dass sie sich zahlreicher Methoden bedienen, die Aufwandskurve möglichst flach zu halten. Inzwischen gibt es eine Vielzahl von agilen Prozessen. Zu den bekanntesten zählen unter anderem:

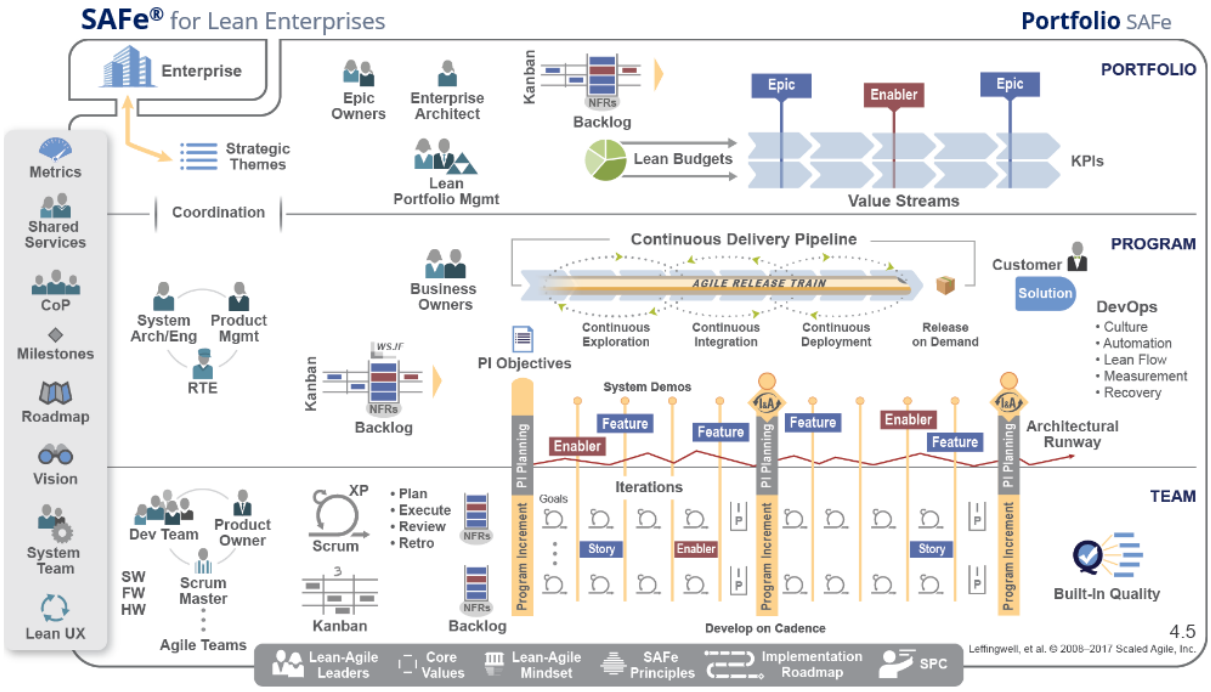
- **Adaptive Software Development (ASD)**
- **Crystal**
- **Extreme Programming (XP)**
- **Feature Driven Development (FDD)**
- **Kanban**

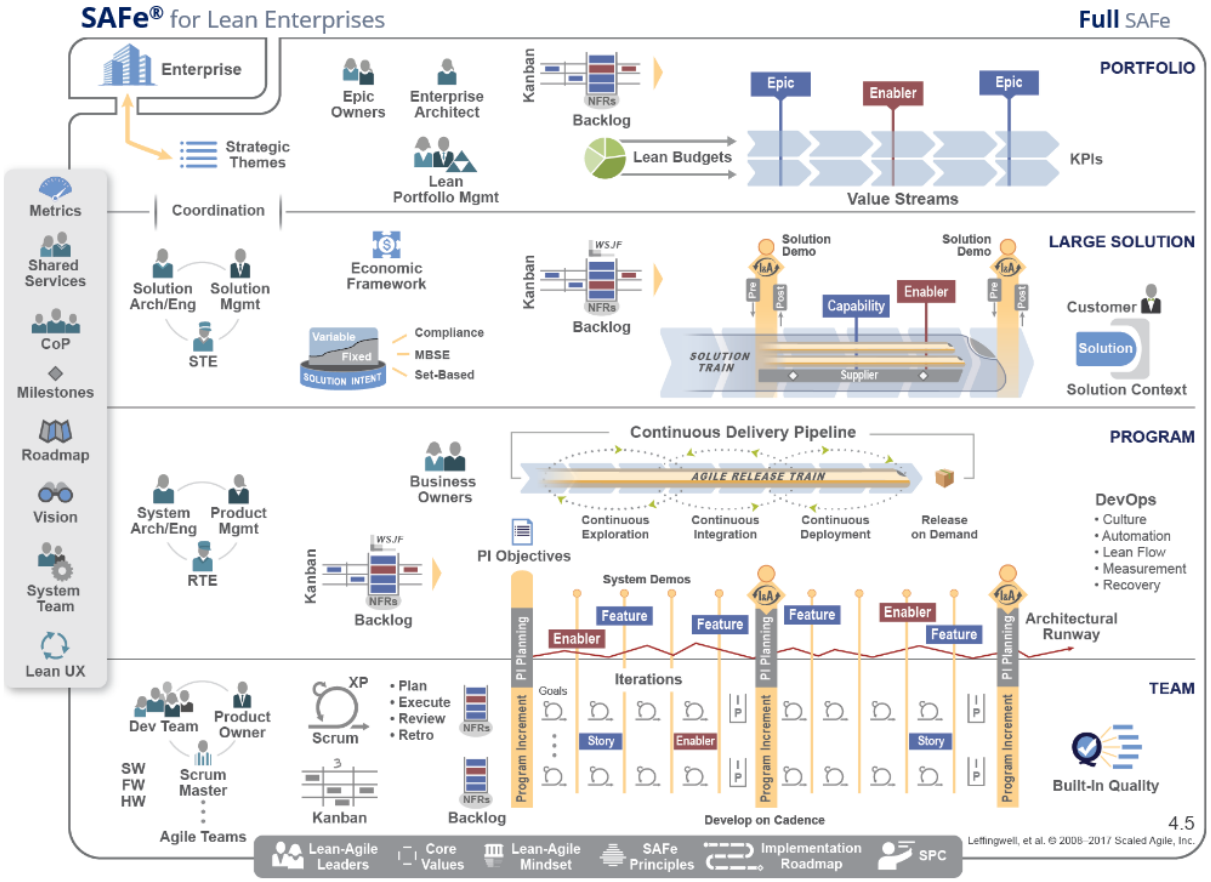
- **Scrum**
- **Agiles Testen**
- **Behavior Driven Development (BDD)**

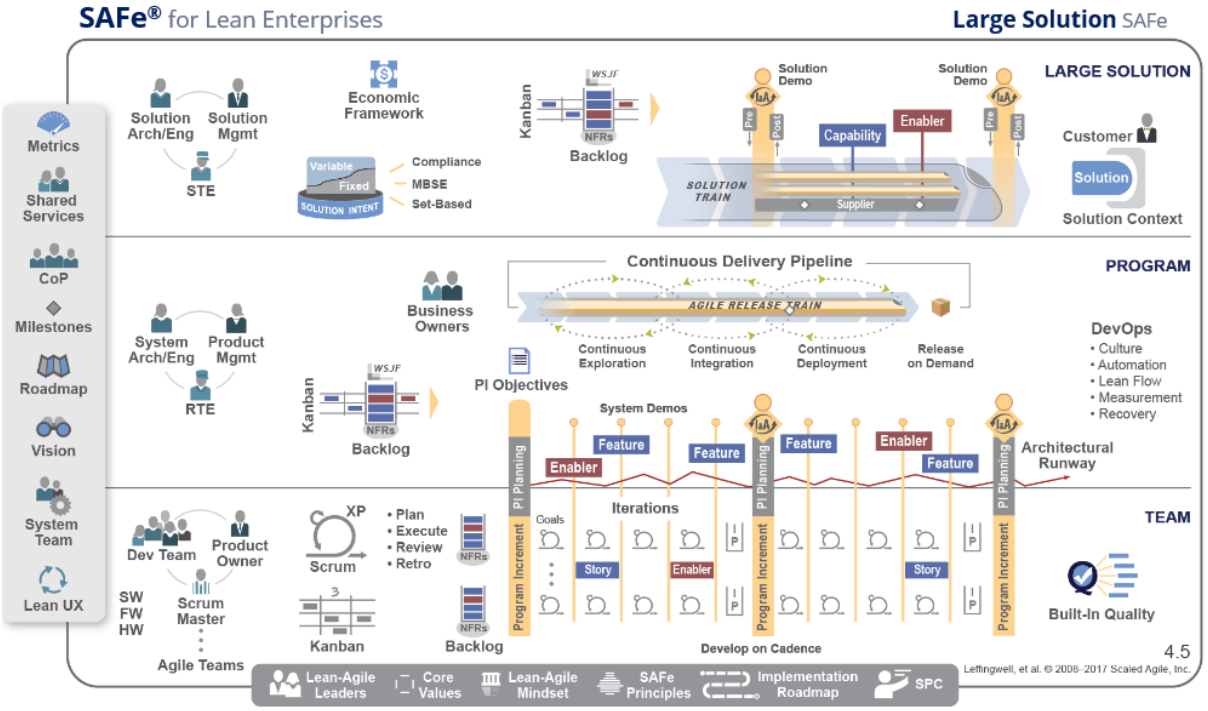
DEFINITIONEN

SAFe - Scaled Agile Framework

- Was SAFe zur Koordination mehrerer solcher Teams etabliert, ist ein «***Team of Teams***», also ein Zusammenschluss mehrerer Teams zu einem sogenannten ***Agile Release Train***.
- Dieser Zusammenschluss mehrerer agiler Teams trifft sich dann zu einem ***Big Room Planning***, um die Prioritäten eines jeden Teams zu alignieren und Abhängigkeiten direkt mit denen zu besprechen, die sie betreffen: den Teams.






SAFe® for Lean Enterprises
Large Solution SAFe
Metrics
Shared Services
CoP
Milestones
Roadmap
Vision
System Team
Lean UX
Solution Arch/Eng
Solution Mgmt
STE
Economic Framework
Variable
Fixed
SOLUTION INTENT
Compliance
MBSE
Set-Based
Kanban
WSJF
NFRs
Backlog
Solution Demo
Pre
Post
Capability
Enabler
SOLUTION TRAIN
Supplier
LARGE SOLUTION
Customer
Solution
Solution Context
Continuous Delivery Pipeline
AGILE RELEASE TRAIN
Continuous Exploration
Continuous Integration
Continuous Deployment
Release on Demand
PROGRAM
Business Owners
System Arch/Eng
Product Mgmt
RTE
PI Objectives
System Demos
Feature
Enabler
PI Planning
Architectural Runway
DevOps
• Culture
• Automation
• Lean Flow
• Measurement
• Recovery
Dev Team
Product Owner
Scrum Master
SW
FW
HW
Agile Teams
XP
Scrum
Kanban
• Plan
• Execute
• Review
• Retro
Iterations
Goals
Story
Program Increment
TEAM
Built-In Quality
Develop on Cadence
4.5
Lean-Agile Leaders
Core Values
Lean-Agile Mindset
SAFe Principles
Implementation Roadmap
SPC
Leffingwell, et al. © 2008–2017 Scaled Agile, Inc.

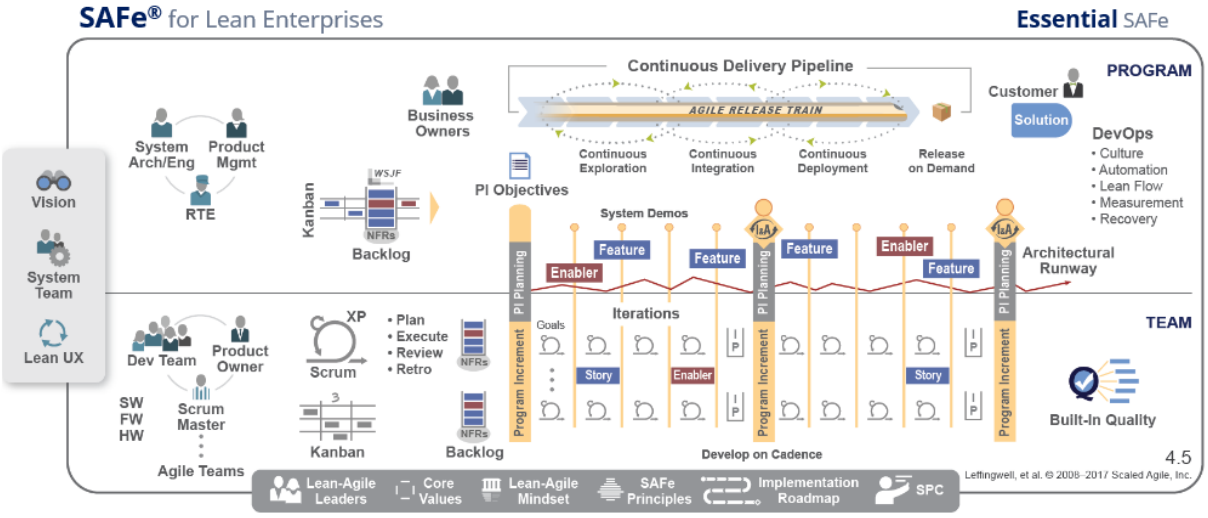

SAFe® for Lean Enterprises
Essential SAFe
Vision
System Team
Lean UX
System Arch/Eng
Product Mgmt
RTE
Business Owners
Continuous Delivery Pipeline
AGILE RELEASE TRAIN
Continuous Exploration
Continuous Integration
Continuous Deployment
Release on Demand
Customer
Solution
PROGRAM
DevOps
• Culture
• Automation
• Lean Flow
• Measurement
• Recovery
Kanban
WSJF
NFRs
Backlog
PI Objectives
System Demos
Feature
Enabler
PI Planning
Architectural Runway
Dev Team
Product Owner
Scrum Master
SW
FW
HW
Agile Teams
XP
Scrum
Kanban
• Plan
• Execute
• Review
• Retro
Iterations
Goals
Story
Program Increment
TEAM
Built-In Quality
Develop on Cadence
4.5
Lean-Agile Leaders
Core Values
Lean-Agile Mindset
SAFe Principles
Implementation Roadmap
SPC
Leffingwell, et al. © 2008–2017 Scaled Agile, Inc.

Table of Figures

Figure 1: Scrum Framework I 4
Figure 2: Scrum Framework II 4
Figure 3: When to use Scrum? 5
Figure 4: Fiction and Facts 5
Figure 5: Scrum Artifacts 7
Figure 6: Events 8
Figure 7: Scrum Events 8
Figure 8: The Scrum Team 10
Figure 9: Product Owner 11
Figure 10: Development Team 12
Figure 11: Sprint 13
Figure 12: Sample Sprint Goal 13
Figure 13: Sprint Goal Management 14
Figure 14: Burn-Down Chart 15
Figure 15: Dysfunction of Teams 16
Figure 16: Story 19
Figure 17: Scrum Tool 20

Index

Agile Softwareentwicklung.....21

Scrum..... 3