

Ein kompakter Überblick zur Agilität nach Scrum

 **bridging IT**

Menschen Methoden Lösungen

## Einleitung

Agilität ist mittlerweile ein weitläufig bekannter Begriff, der auch außerhalb der IT zunehmend ein Thema ist und immer mehr Anklang findet. Agiles Management bezieht sich nicht nur auf agiles Projektmanagement und hat weitreichende Auswirkungen auf Führungsprinzipien, Wertekultur und Organisationsstrukturen eines gesamten Bereichs oder ganzer Unternehmen.

Um das Thema Agilität einzuführen und umzusetzen, gibt es mittlerweile verschiedene Methoden und Good Practices als bekannte Frameworks am Markt. Eines der etabliertesten und bekanntesten Frameworks hierfür ist unbestritten Scrum.

Daher haben wir bei bridgingIT dieses Booklet als handlichen Guide zum Verständnis und zur Unterstützung auf dem agilen Weg zu Scrum entwickelt, um das Good Practice Framework zugänglich und umsetzbar zu machen. Immer mit der Zielsetzung im Hinterkopf, nicht nur nach agilen Regeln zu arbeiten („do agile“), sondern auch nach agilen Grundwerten zu denken und zu handeln („be agile“).

# Inhaltsverzeichnis

**Einleitung** .....2

## **Grundlagen**

Agilität..... 4  
Was ist Scrum?..... 10

## **Scrum-Team/-Rollen**

Product Owner ..... 12  
Development-Team..... 14  
Scrum Master ..... 16  
Agile Coach ..... 18

## **Scrum-Anforderungen**

Product Vision..... 20  
Themes..... 22  
Roadmap..... 24  
Epics ..... 26

User Stories ..... 28  
Tasks ..... 30

## **Scrum Events**

Sprint..... 32  
Sprint Planning (I und II)... 34  
Daily Scrum ..... 38  
Sprint Review..... 40  
Sprint Retrospective..... 42

## **Scrum-Artefakte**

Product Backlog ..... 44  
Sprint Backlog ..... 46  
Inkrement..... 48

## **Essentielle Hilfsmittel**

Definition of Ready ..... 50

Definition of Done..... 52  
Backlog Refinement..... 54  
Scrum/Sprint Board ..... 56  
Burndown Charts..... 58  
Commitment & Delivery... 60

## **Weitere Hilfsmethoden**

Planning Poker..... 62  
Timeboxing ..... 64  
Grenzen und  
Erweiterungen..... 66

## **Über bridgingIT..... 70**

OpenneSs

Courage

Respect

FocUs

CommitMent

## Die agilen Grundwerte

**Openness:** Das Team ist offen gegenüber Veränderungen, anderen und neuen Ansichten und nutzt diese für seine Arbeit.

**Courage:** Das Team steht zu seiner Arbeit, aber auch zu Fehlern und kommuniziert diese offen und transparent.

**Respect:** Der Umgang mit anderen Teammitgliedern ist von gegenseitigem Respekt und Vertrauen geprägt.

**Focus:** Die Teammitglieder setzen klare Prioritäten und arbeiten Aufgaben nacheinander und vollständig ab.

**Commitment:** Die Teammitglieder stellen die Teamziele stets über ihre individuellen Bedürfnisse und Ansichten.



## Agile Methoden

Individuen &  
Interaktion

funktionierende  
Lösung

Zusammenarbeit  
Business & IT

auf Änderungen vor-  
bereitet sein & eingehen

mehr Mehrwert



## Wasserfallmethode

Prozesse &  
Werkzeuge

allumfassende  
Dokumentation

Vertragsverhandlungen  
& -abgrenzungen

strikte Planverfolgung  
von A bis Z

Mehrwert

## Agile Prinzipien

Die rechts beschriebenen Prinzipien haben auch in agilen Methoden ihren berechtigten Platz, sind den links abgebildeten Prinzipien aber untergeordnet.

**Individuen & Interaktion** sind wichtiger als Prozesse & Werkzeuge, weil Lösungen immer von Personen entwickelt und umgesetzt werden.

**Funktionierende Lösungen** sind wichtiger als eine umfassende Dokumentation, denn nur funktionierende Lösungen werden auch eingesetzt.

Die **Zusammenarbeit von unterschiedlichen Stakeholdern** ist Voraussetzung für die Entwicklung gelungener Lösungen.

Anforderungen und Wünsche **verändern sich mit der Zeit** oder müssen neu definiert werden. Dadurch ergeben sich neue, innovative Lösungsmöglichkeiten.

## Wasserfall

## Agil

Constraints  
„Limitierung“

Scope

Kosten

Termin

PLAN  
DRIVEN

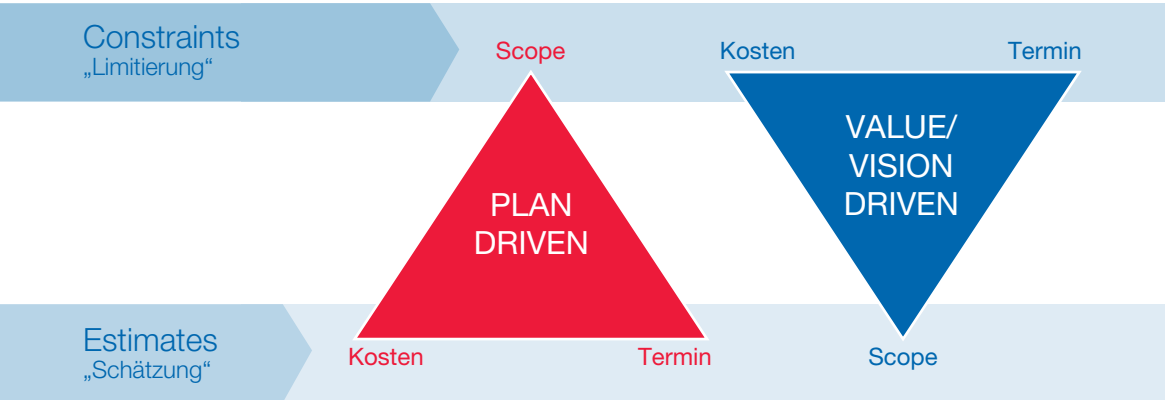
VALUE/  
VISION  
DRIVEN

Estimates  
„Schätzung“

Kosten

Termin

Scope





# Agile Zielsetzung

Agile Ziele folgen immer einer vorher definierten Vision eines erwünschten Endzustands, sind jedoch niemals fix.

Im Gegensatz zu klassischen Methoden werden zu Beginn bestimmte Constraints festgesetzt, die das Erreichen des Ziels aber so wenig wie möglich beeinflussen, sondern nur dazu dienen, den Projektraum abzustecken.

Aus der Vision leiten sich Ziele ab, die sich im Laufe eines Projekts verändern können und dadurch das weitere Vorgehen beeinflussen. Im Fokus steht nicht, ob ein Ziel auf eine bestimmte Art und Weise erreicht wurde, sondern nur, welchen Mehrwert das Ziel zur Vision liefert.

Daher ist es in agilen Methoden essenziell, regelmäßig Zwischenziele zu erreichen und ihren Wertbeitrag zur Vision zu überprüfen.



# Was ist Scrum?

Scrum ist ein agiles Framework, bei dem der Mehrwert des Ergebnisses im Mittelpunkt steht und dem Projektteam größtmögliche Freiheiten bei der Erreichung dieses Ziels gegeben werden. Dies wird durch eine fixe Mischung aus Arbeitsphasen und regelmäßigen Treffen erreicht.

Ursprünglich wurde Scrum für die Softwareentwicklung konzipiert. Mittlerweile wird es jedoch in vielen sehr unterschiedlichen Bereichen erfolgreich angewendet.

Immer dann, wenn Projekte unvorhersehbare Ereignisse berücksichtigen müssen, komplexe Probleme erstmals gelöst werden sollen, oder sich Projektanforderungen im Laufe der Zeit wahrscheinlich verändern, kommen die Stärken von Scrum zum Tragen.

Alle agilen Methoden bauen auf einem gemeinsamen Mindset auf, das im agilen Manifest in Form von Werten, Prinzipien und Zielen beschrieben wird.

Product Owner  
„Demand“



Development-Team  
„Supply“



Scrum Master  
„Service“

## Der Product Owner (PO)

Der Product Owner ist der „Mini-CEO“ des Produkts und somit auch für dessen Erfolg verantwortlich. Er ist immer eine einzelne Person, kann bzw. sollte jedoch durch weitere Personen in seinen Aufgaben im Sinne eines PO-Teams oder aus dem Scrum-Team heraus unterstützt werden.

Er entwickelt als „Servant Leader“ eine Vision für das Produkt und zeigt in einer Roadmap auf, wie diese erreicht werden soll. Der PO hat die Hoheit über das Product Backlog und entscheidet, was umgesetzt und freigegeben wird. Er definiert und priorisiert Features, verwaltet Änderungsbedarfe und formuliert die User Stories für die unterschiedlichen Funktionalitäten. Durch die Priorisierung von Arbeitsschritten maximiert er den Return of Investment des Projekts.

Zusätzlich vertritt der Product Owner die Projektgruppe gegenüber Stakeholdern im Unternehmen und fungiert als Schnittstelle zum Kunden. Während des Sprints steht er dem Team zur Verfügung, um Fragen zu beantworten und die Arbeit zu koordinieren.

Product Owner  
„Demand“



Development-Team  
„Supply“



Scrum Master  
„Service“

## Das Development-Team (DEV)

Das Development-Team bearbeitet die Einträge im Product Backlog und erstellt Lösungen. Dabei trägt es die Verantwortung für Analyse, Design, Entwicklung, Test und Dokumentation.

Es erstellt das Sprint Backlog im Rahmen des Sprint Plannings aus ausgewählten Einträgen des Product Backlogs und dokumentiert den eigenen Fortschritt im Sprint Board und Backlog.

Das DEV-Team ist (komplett) selbstorganisiert und kann dadurch seine Arbeitsweisen auf die eigenen Wünsche und Vorstellungen ausrichten, entwickelt beispielsweise eigene Teamregeln und -prozesse. Idealerweise besteht es aus 3–9 Personen, die so organisiert sind, dass ein einfacher Austausch untereinander möglich ist (z. B. durch Co-Location am selben Ort).

Das Team sollte multidisziplinär aufgestellt sein und die unterschiedlichen Kompetenzen der Teammitglieder kombinieren. Dadurch kann spätestens am Ende eines Sprints ein vollständiges Produktinkrement vorgestellt und ausgeliefert werden.

Product Owner  
„Demand“



Development-Team  
„Supply“



Scrum Master  
„Service“



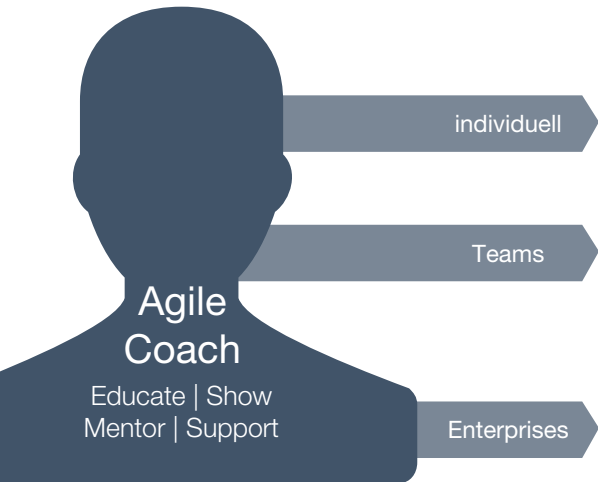
## Der Scrum Master (SM)

Der Scrum Master ist „Lehrer“ für das restliche Scrum-Team, wobei er das Scrum Framework implementiert, Fragen rund um Scrum beantwortet und durch Beobachten und gezieltes Eingreifen sicherstellt, dass der Prozess von allen Teammitgliedern eingehalten, akzeptiert und kontinuierlich verbessert wird.

Er unterstützt die Organisation somit auch als Methoden- und Verhaltenscoach bei der übergreifenden Scrum-Befähigung, macht aber keine direkten Vorgaben zur Art der Umsetzung außerhalb der Methodiken von Scrum.

Das Scrum-Team unterstützt und schützt er durch Koordination, Moderation und die Beseitigung von Hindernissen (Impediments).

Des Weiteren dient er als Vermittler zwischen Development-Team, Product Owner und dem Management und als Mediator bei Konflikten.



Product Owner

Developer

Scrum Master



# Agile Coach

Der Agile Coach stellt eine Erweiterung der klassischen Scrum-Rollen dar.

Er ist Experte in agilen Methoden und unterstützt Einzelpersonen, gesamte Scrum-Teams und sogar ganze Einheiten als externe Kraft. Insbesondere sorgt er auch für die Anerkennung der Arbeitsweisen von Scrum-Teams in der umgebenden Organisation und regelt, strukturiert und koordiniert die Arbeit zwischen klassischen und agilen Teams.

Neben Hilfestellungen zur Umsetzung von Scrum im jeweiligen Umfeld kann der Agile Coach auch die Teamzusammensetzung und Kommunikationsstrukturen beobachten und beurteilen und so für Konfliktprävention und eine optimale Arbeitsverteilung sorgen. Dabei greift er jedoch nur sehr eingeschränkt in die operative Arbeitsweise ein und wirkt vielmehr anleitend und unterstützend.

Zusätzlich fördern und kommunizieren Agile Coaches die grundlegenden agilen Werte und setzen diese zusammen mit den Teams um.



# Product Vision

Die Product Vision zeichnet ein klares Bild des Zielprodukts. Sie wird am Anfang des agilen Prozesses vom Product Owner formuliert und stellt seine aktuelle Wunschvorstellung des fertigen Produkts dar. Die Vision ist dabei eine ausführliche Beschreibung der gewünschten Eigenschaften, Bedienarten oder Abläufe des finalen Produkts.

Je genauer die Vision beschrieben wird, desto besser kann sie vom Scrum-Team aufgenommen und verstanden werden. Themes können dabei helfen, die Vision in thematische, unabhängige Bereiche zu unterteilen.

Die Product Vision dient als Grundlage für das Product Backlog und kann in Verbindung damit zur Messung des Grads der Zielerreichung verwendet werden.

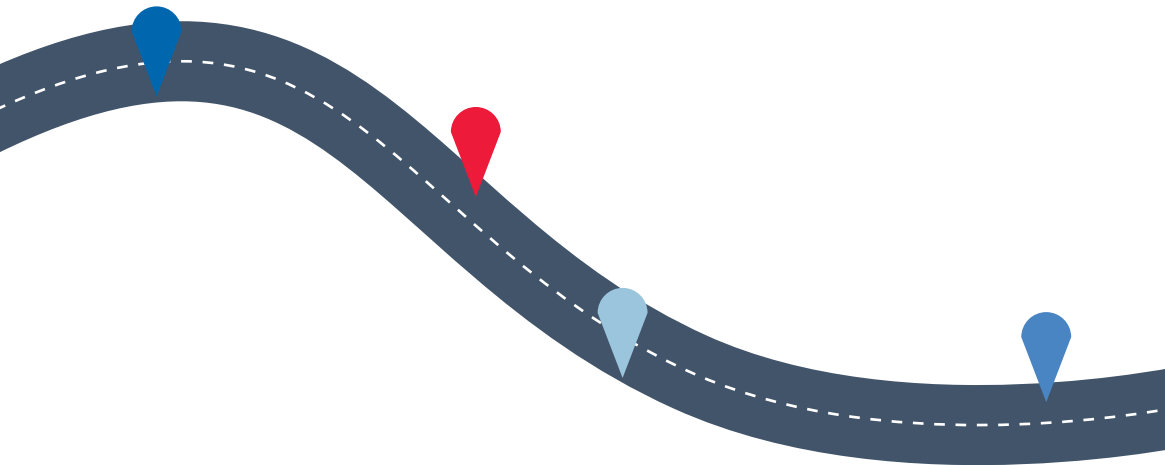


# Themes

Eine Product Vision beschreibt oftmals Veränderungen in sehr unterschiedlichen thematischen Bereichen, beispielsweise in verschiedenen Unternehmensabschnitten oder eine Software mit vielfältigen Funktionen. Da es sehr aufwändig und unübersichtlich ist, alle erwünschten Änderungen in allen Bereichen ungeordnet im Backlog zu vermerken, wird die Vision zur besseren Übersicht in Themes aufgeteilt, die jeweils einen Bereich abdecken und unabhängig voneinander bearbeitet werden können.

Themes werden weiterführend durch mehrere Epics und wiederum User Stories detailliert, die zu einem ähnlichen (Aufgaben- oder Themen-)Bereich gehören, aber im Gegensatz zu Epics unabhängig voneinander bearbeitet werden können. So wie jedes Epic und jede User Story einen Teil zu einem Theme beiträgt, trägt jedes Theme einen Teil zur Umsetzung der Vision bei.

Durch die Reihenfolge, in der Themes bearbeitet werden, lässt sich eine erste Roadmap für die Umsetzung der Product Vision und damit der Fertigstellung des Endprodukts entwerfen.





# Roadmap

Aus dem Umfang und der Komplexität der Product Vision kann eine Roadmap entwickelt werden, die die Bearbeitung der unterschiedlichen Bereiche der Vision zeitlich einordnet.

Die Roadmap kann sich dabei an den Themes orientieren, aber auch anderen Schemata folgen. Mithilfe der Roadmap lässt sich die Anzahl an benötigten Sprints erstmalig schätzen und damit auch ein Überblick über die benötigten Ressourcen und unterschiedlichen Veröffentlichungszeiträume erstellen.

Die Roadmap ergänzt die Scrum-Methode und dient vor allem zu Beginn des Projekts zur Aufwandsschätzung. Nach den Ergebnissen der ersten Sprints kann sie weiter verfeinert werden, um die Aufwandsschätzung zu präzisieren.



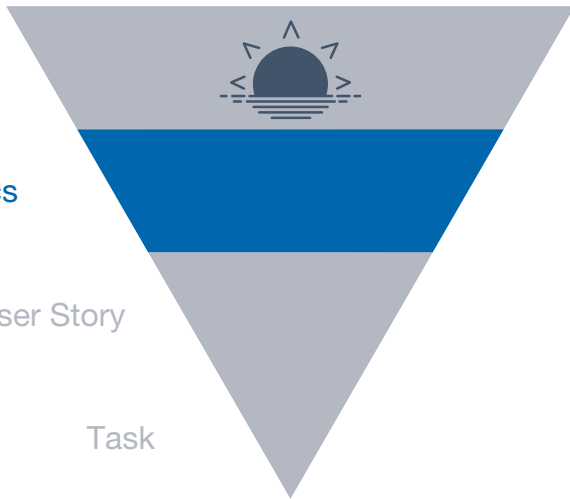
Vision & Themes



Epics

User Story

Task



# Epics

Ein Epic ist eine Gruppierung von einzelnen User Stories, die erst nach vollständiger Bearbeitung einen umfassenden Nutzen für das Inkrement liefern. Damit ist ein Epic vergleichbar mit einer User Story, aber zu umfangreich, um in einem Sprint abgeschlossen zu werden.

Um trotzdem innerhalb eines Sprints an einem Epic arbeiten zu können, wird es in direkt voneinander abhängige User Stories aufgeteilt, die in aufeinanderfolgenden Sprints bearbeitet werden.

Im Gegensatz zu anderen User Stories liefern einzelne User Stories eines Epics somit nicht zwangsläufig einen direkten Beitrag zum aktuellen Inkrement, solange die übrigen, dem Epic zugeordneten User Stories nicht abgeschlossen sind. In der Regel wird erst mit Abschluss aller zugehörigen User Stories das Epic in das Inkrement implementiert und der Nutzen für das Endprodukt sichtbar.

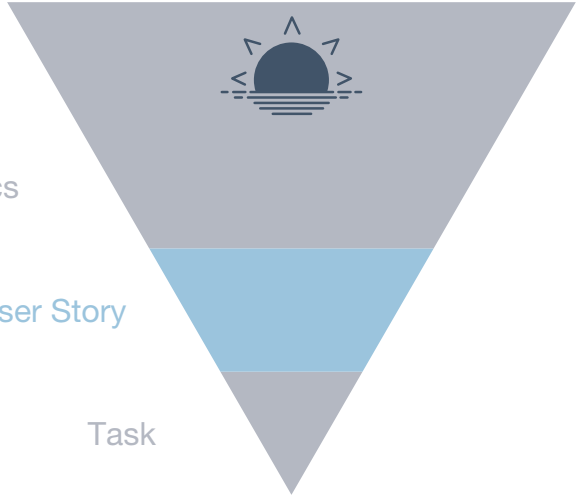


Vision & Themes

Epics

User Story

Task



# User Stories

Eine User Story beschreibt eine einzelne benötigte Funktionalität aus der Perspektive eines bestimmten Anwenders oder einer bestimmten Anwendergruppe. Einträge in dem Product Backlog sind normalerweise entweder als User Stories oder als Epics hinterlegt.

User Stories müssen klein genug sein, um innerhalb eines Sprints vollständig bearbeitet werden zu können. Der genaue Umfang einer Story wird durch die Anzahl und Komplexität der enthaltenen Tasks bestimmt. Je komplexer eine User Story und je aufwändiger die Umsetzung ist, desto weniger unterschiedliche Stories können in einem Sprint erledigt werden.

Die Komplexität und der damit verbundene Zeitaufwand für die Bearbeitung einer einzelnen User Story wird vor Beginn eines Sprints geschätzt und am Sprintende im Review mit dem realen Aufwand verglichen. Dabei werden die enthaltenen Tasks sowohl auf Komplexität untersucht als auch betrachtet, welche Tasks weggefallen sind und welche nachträglich hinzugefügt wurden.

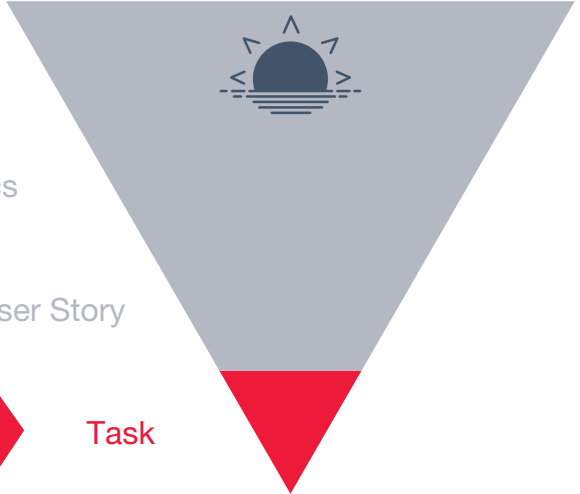


Vision & Themes

Epics

User Story

Task



# Task

Ein Task ist die kleinste Aufgabeneinheit in Scrum. Aus jeder User Story ergeben sich unterschiedliche Tasks. Diese können im Gegensatz zu User Stories auch erst im Verlauf eines Sprints entstehen. Wichtig ist jedoch, dass sie klar einer User Story zugeordnet sind und von einer Einzelperson erledigt werden können.

So wie sich das Sprint Backlog aus den User Stories zusammensetzt, setzt sich jede User Story aus unterschiedlichen Tasks zusammen.

Tasks können verschiedene Arten von Arbeiten sein (beispielsweise Aufgabe, Meeting oder Termin), solange das Scrum-Team weiß, welchen Inhalt der Task hat und er dazu beiträgt, eine User Story erfolgreich abzuschließen.



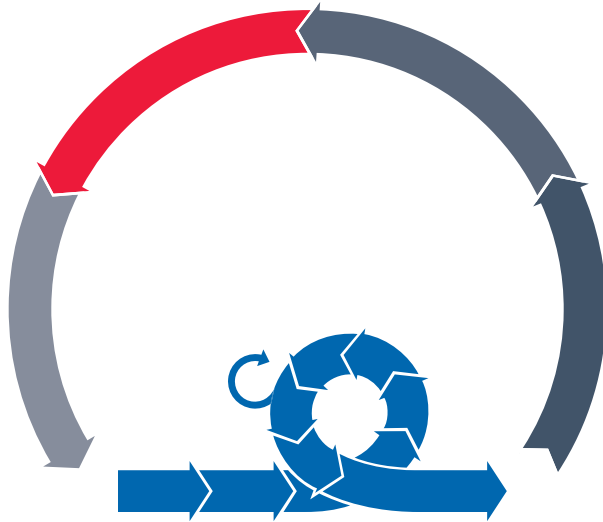


# Sprint

Eines der zentralen Merkmale von Scrum ist die Aufteilung eines Großprojekts in eine Vielzahl an vollständigen Kleinprojekten, den Sprints. Je nach Umfeld, agilem Reifegrad und persönlichen Vorlieben besitzt ein Sprint eine feste Dauer von 1–4 Wochen. Ein Sprint darf nachträglich nicht verlängert oder verkürzt werden (vgl. Timeboxing).

Im Sprint werden die im Sprint Planning besprochenen und vereinbarten User Stories umgesetzt und an dessen Ende zu einem funktionierenden (potentiell auslieferbaren) Inkrement zusammengesetzt.

Im Laufe eines Sprints finden unterschiedliche, fest vorgeschriebene Treffen statt, die den Sprintverlauf gliedern. Diese sind das Sprint Planning 1 + 2, die Daily Scrums sowie das Review und die Retrospective.



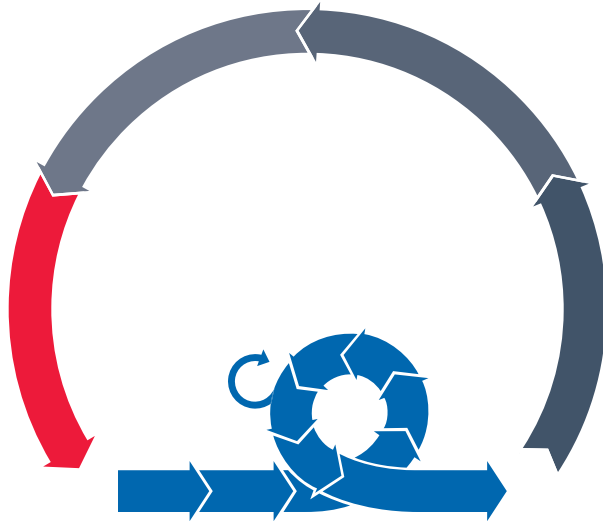
# Sprint Planning 1

Die Sprint-Planung zu Beginn jedes Sprints setzt sich aus zwei Treffen zusammen: dem Sprint Planning 1 und dem Sprint Planning 2.

Im Sprint Planning 1 stimmt sich das DEV-Team gemeinsam mit dem Product Owner ab, welche Stories aus dem priorisierten Backlog im kommenden Sprint umgesetzt werden sollen und können. Dabei folgt das Sprint Planning 1 klaren Regeln:

- Der PO stellt die User Stories entlang ihrer Priorität vor und beantwortet Rückfragen des Teams, um Unklarheiten zu beseitigen.
- Danach entscheidet das DEV-Team, ob die vorgestellte User Story im Sprint entwickelt werden kann, und bewertet den Aufwand der Story und die dahinter liegende Komplexität anhand von Story Points.\*
- Mit dem Sprint Planning 1 und der Übernahme der User Stories ins Sprint Backlog wird die Verantwortung für die User Story an das DEV-Team übergeben. D. h. der PO darf von sich aus die Userstory ab jetzt nicht mehr inhaltlich ändern!

\*Es existieren Techniken, die bei der Schätzung/Bewertung der User Stories helfen können; siehe dazu bspw. das Unterkapitel „Planning Poker“.

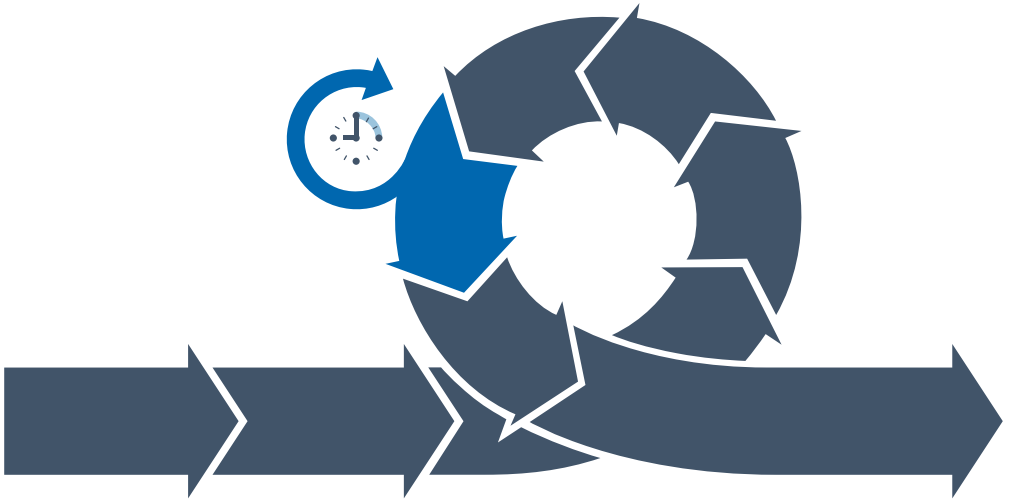


## Sprint Planning 2

Im Sprint Planning 2 werden die für den kommenden Sprint ausgewählten User Stories durch das DEV-Team in einzelne Tasks heruntergebrochen und eine erste Aufteilung auf die Teammitglieder vorgenommen.

Dabei ist insbesondere wichtig, dass die Tasks in ihrer Länge und Komplexität so exakt wie möglich eingeschätzt werden, sodass das Team als Ziel alle ausgewählten User Stories innerhalb des Sprints abarbeiten kann.

Zusätzlich definiert das Team ein konkretes Sprint-Ziel, das sie mit der Erledigung der User Stories erreichen möchten. Anhand dieses Ziels kann das DEV-Team die ausgewählten User Stories noch einmal genauer für einen optimalen Sprintablauf priorisieren und am Sprintende den eigenen Erfolg messen.



## Daily Scrum

Das Daily Scrum ist eine tägliche Besprechung des Scrum-Teams, meist am Anfang des Tages.

In dem maximal 15 Minuten dauernden Treffen werden die wichtigsten Aufgaben des Tages und Fragen der Teammitglieder untereinander abgestimmt. Ebenso wissen die Teammitglieder nach dem Treffen, mit wem sie sich im Laufe des Tages weiter absprechen müssen bzw. wer ihre Unterstützung benötigt.

Hierzu werden typischerweise drei Fragen durch jedes Teammitglied beantwortet: Was habe ich seit gestern zum Sprintziel beigetragen? Was plane ich heute zum Sprintziel beizutragen? Wo brauche ich Unterstützung von wem bzw. was behindert mich aktuell?

Die Daily Scrums helfen dem Team dabei, den täglichen Workload einzuschätzen und schon früh ein Gefühl dafür zu entwickeln, ob der laufende Sprint erfolgreich abgeschlossen werden kann. Zusätzlich werden durch die regelmäßigen Absprachen Konflikte an Schnittpunkten vermieden und ein Erfahrungsaustausch zwischen den Teammitgliedern angeregt.



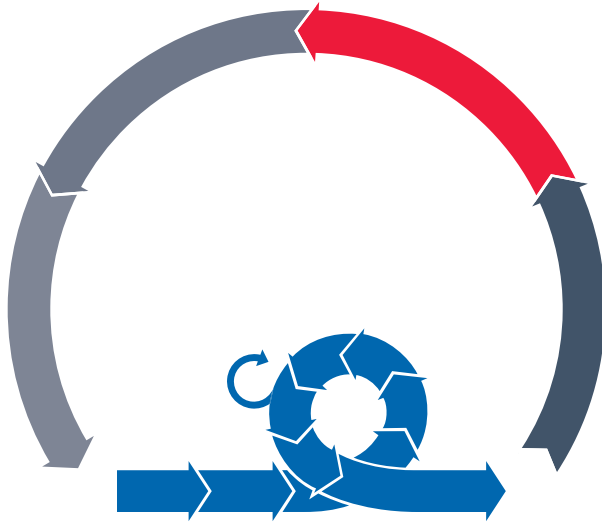


## Sprint Review

Das Sprint Review findet am Ende jedes Sprints statt. Hier treffen der PO, der SM, das DEV-Team und – soweit vorhanden – weitere Stakeholder zusammen.

In diesem Meeting wird das durch den Sprint erstellte Inkrement vorgestellt und ausprobiert. Des Weiteren wird besprochen, welche Aufgaben erledigt wurden und welche Themen offen (geblieben) sind. Ist der Product Owner mit dem Zustand des Inkrements zufrieden, erfolgt die offizielle Abnahme und Freigabe des Inkrements.

Da der Product Owner das fertige Produkt, das Inkrement, selbst sieht und es ausprobieren kann, bekommt er ein Gefühl dafür, ob es seinen Wünschen entspricht. Je nachdem, wie zufrieden er mit der momentanen Entwicklungsrichtung ist, kann er anschließend Änderungen am Product Backlog vornehmen, um es an seine geänderte Vorstellung des fertigen Produkts anzupassen.



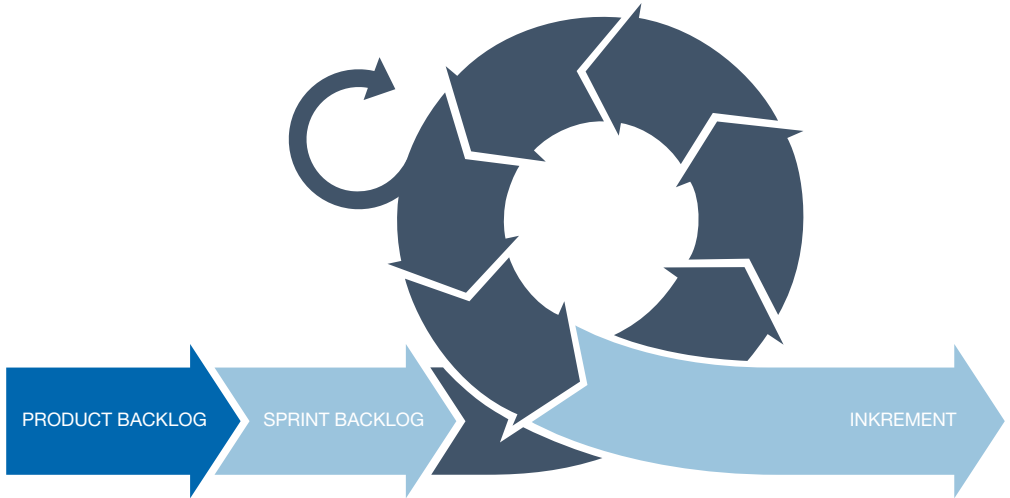
## Sprint Retrospective

Die Retrospective dient als Steuerungsmeeting im Sinne der kontinuierlichen Verbesserung des agilen Prozesses (KVP).

In diesem Meeting spricht das Scrum-Team über Probleme sowie Erfolge im vergangenen Sprint und identifiziert Stärken und Verbesserungspotenziale. Der Scrum Master übernimmt meist die Rolle des Moderators und hilft dem Team dabei, die Verbesserungen im nächsten Sprint umzusetzen.

Das regelmäßige Feedback zur eigenen Aufwandsschätzung und den Arbeitsweisen ist eine weitere große Stärke von Scrum. Durch die Reflexion werden Ineffizienzen schnell entdeckt und können vom Team in kurzer Zeit behoben werden.

Zum erfolgreichen Einsatz der Retrospective sind eine gewisse Offenheit und das Eingestehen eigener Fehler unbedingt nötig. Hierfür ist Vertrauen ein wichtiger Schlüsselfaktor.



# Product Backlog

Das Product Backlog enthält alle aktuellen Anforderungen an bzw. Bedarfe für das Produkt und wird vom Product Owner verwaltet. In ihm werden alle Funktionen als einzelne User Stories beschrieben und nach Relevanz sortiert. Je wichtiger eine Funktion ist, desto detaillierter wird sie beschrieben und desto weiter oben steht sie im Backlog.

Dementsprechend ist es für den Product Owner sehr hilfreich, sich bei der Detaillierung und Ausgestaltung der User Stories an der Priorität bzw. aktuelle Position im Product Backlog zu orientieren.

Im Sprint Planning 1 stimmen sich Development-Team und Product Owner darüber ab, welche Einträge (User Stories) im kommenden Sprint bearbeitet und dadurch ins Sprint Backlog übertragen werden.

Das Product Backlog existiert so lange, wie das Produkt selbst existiert, und verwaltet jegliche Änderungen, Fehler und nötige (zukünftige) Entwicklungsschritte.

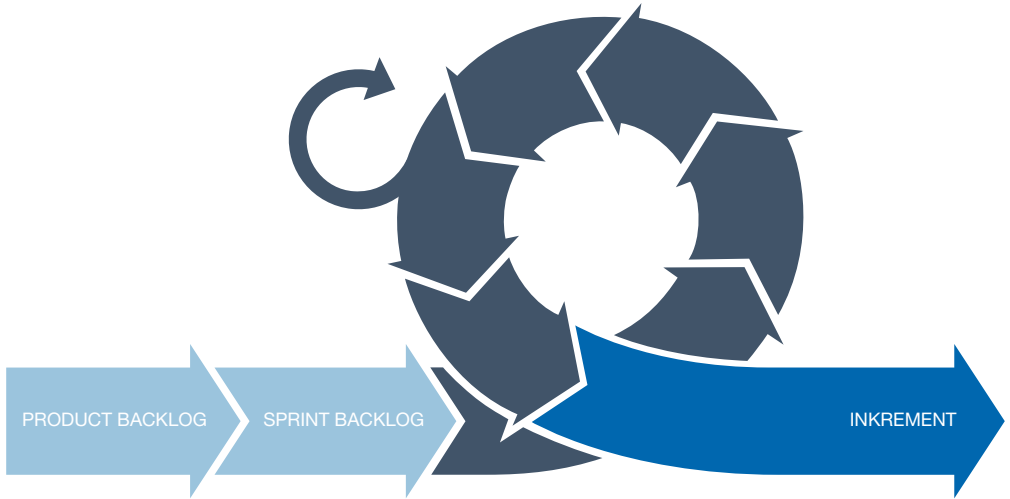


## Sprint Backlog

Das Sprint Backlog enthält alle für den jeweiligen Sprint ausgewählten User Stories, die aus dem Product Backlog im Rahmen des Sprint Planning 1 übernommen wurden. Die Stories werden durch das Team über das Sprint Planning 2 in einzelne Arbeitsschritte (Tasks) aufgebrochen, die ebenfalls im Backlog hinterlegt werden.

Im Laufe des Sprints wird das Backlog entsprechend des tatsächlichen Arbeitsfortschritts regelmäßig, meist im Rahmen des Dailies oder sogar mehrmals am Tag, angepasst. Zusätzlich benötigte Arbeitsschritte (Tasks) werden zum Backlog hinzugefügt, unnötige entfernt. Der Bearbeitungszustand einzelner Tasks und die Zuständigkeiten werden ebenfalls im Backlog festgehalten. Zur visuellen Darstellung wird in der Regel ein Sprint Board verwendet (siehe Unterkapitel „Scrum/Sprint Board“).

So entsteht eine detaillierte Übersicht der aktuellen Arbeit des Teams und des zu erwartenden Fortschritts während des Sprints.





# Inkrement

Das Inkrement stellt einen potentiell „fertigen“ Teil des Produkts (spätestens) am Ende eines Sprints dar. In ihm sind alle abgearbeiteten Backlog-Einträge und alle Inkremente vorheriger Sprints enthalten.

Jedes Inkrement ist an sich ein potenziell auslieferungsfertiges Produkt, auch wenn es nicht ausgeliefert werden soll.

Im Sprint Review entscheidet dann der Product Owner, ob die Funktionalität seinen Ansprüchen entspricht und das Inkrement somit Teil des Endprodukts wird bzw. bleibt.

Durch die Unterteilung des Produkts in eine Vielzahl an Inkrementen können Funktionalitäten im Nachhinein hinzugefügt oder wieder entfernt werden. Hierdurch entsteht der Vorteil, dass das Produkt „lebendig“ ist und sehr wahrscheinlich nur Dinge enthält, die wirklich funktionieren bzw. sogar genutzt werden.

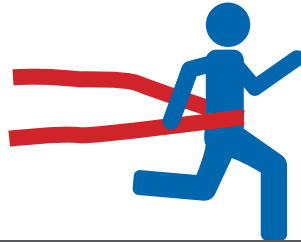


## Definition of Ready (DoR)

Bevor eine User Story in einem Sprint bearbeitet werden kann, muss sie bestimmte Kriterien erfüllen, um richtig verstanden und damit auch korrekt umgesetzt werden zu können. Diese Kriterien sind in der Definition of Ready festgeschrieben. Hierbei handelt es sich um eine verbindliche Vereinbarung zwischen Product Owner und DEV-Team.

Um eine vollständige (= „ready“) User Story zu erstellen, müssen u. a. die Anforderungen inklusive Akzeptanzkriterien, der bereits vorhandene Projektfortschritt, die technischen Möglichkeiten und auch die Fähigkeiten des Scrum-Teams berücksichtigt werden. Dementsprechend kann die Definition of Ready während des Projektes angepasst werden.

Die Definition of Ready wird von vielen äußeren Faktoren beeinflusst und kann sich daher von Team zu Team stark unterscheiden.

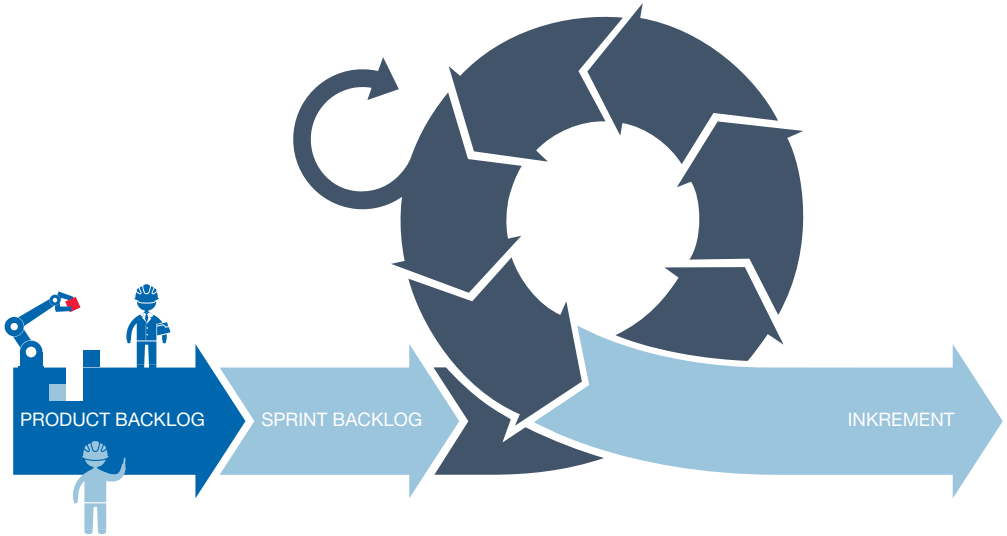


## Definition of Done (DoD)

Die Definition of Done ist eine individuelle Vereinbarung des Scrum-Teams darüber, wann die Arbeit an einer User Story abgeschlossen ist. Jedes Teammitglied muss ein genaues Verständnis dieser Definition entwickeln und dieses auf unterschiedliche User Stories anwenden können.

Durch eine gemeinsame Festlegung, wann eine User Story abgeschlossen ist, kann der Arbeitsfortschritt im Sprint Review und bereits während der Daily Scrums exakt betrachtet werden. So wird sichergestellt, dass ein Inkrement wirklich auslieferungsfertig ist. Ebenso werden Konflikte über den Grad der Fertigstellung vermieden. Je genauer die Definition of Done festgelegt ist, desto effektiver ist die Konfliktvermeidung.

So einfach diese Konvention klingen mag, so schwierig ist es, sie konsequent umzusetzen.



## Backlog Refinement

Das Backlog Refinement bezeichnet die kontinuierliche Überarbeitung und Kontrolle des Product Backlogs durch den Product Owner zusammen mit dem restlichen Scrum-Team und ggf. weiteren Personen/Stakeholdern.

Hierbei evaluieren die Teilnehmer, ob das Product Backlog noch alle relevanten Produktfeatures enthält, die User Stories noch korrekt priorisiert sind, und ob die Ergebnisse der vergangenen Sprints wirklich die wichtigsten User Stories behandelt haben.

Im Zuge dessen ist es möglich, User Stories zu entfernen, hinzuzufügen oder neu zu priorisieren. Ebenso können Annahmen zum Arbeitsaufwand hinzugefügt oder geändert werden.

Das Backlog Refinement sollte regelmäßig durchgeführt werden, ist jedoch kein fest vorgeschriebenes Treffen, da die Tätigkeiten teilweise durch den PO begleitend ausgeführt werden. Findet kein Backlog Refinement statt, besteht das Risiko, dass die fachliche Klärung in das Planning verlagert wird und damit das Ergebnis des Plannings negativ beeinträchtigt.



 Sprint Backlog

Story #1

Story #2

Story #3

 To Do

Task A

Task B   Task D

 In Progress

Task C

Done

Task A   Task B

Task C

Task B

Task A

Task C



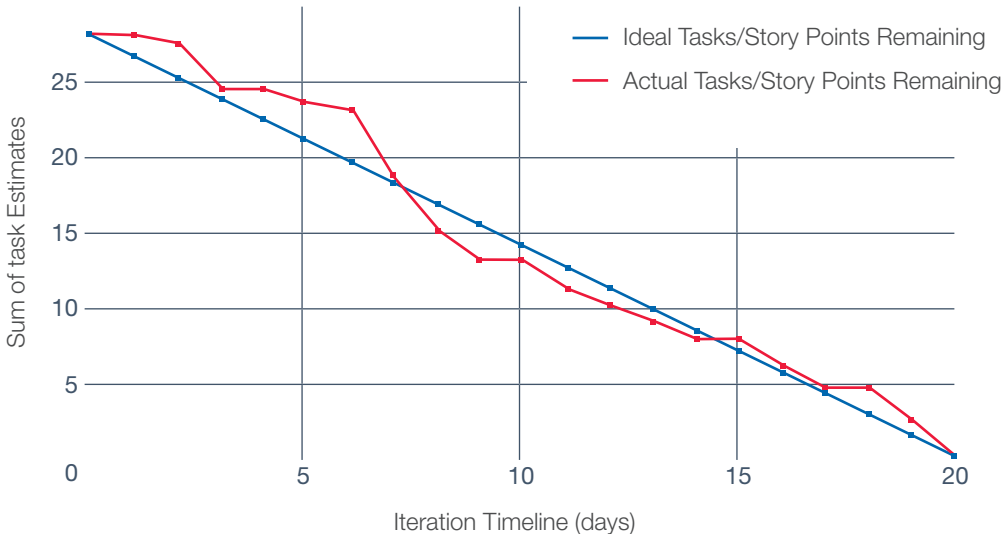
## Scrum/Sprint Board

Das Scrum bzw. Sprint Board ist eine visuelle Darstellung des aktuellen Sprints mit allen User Stories, Tasks, Zuständigkeiten und Arbeitsfortschritten.

Bei kleineren Projekten kann es ein physisches Board sein, an das Zettel geheftet werden. Häufig wird aber eine Software verwendet, die das Sprint Board digital abbildet.

Durch die für alle einsehbare Visualisierung wird es dem Team ermöglicht, den Sprintfortschritt im Auge zu behalten und die eigenen Aufgaben zu überblicken. Ebenso wird – je nach Ausgestaltung und Nutzung – die teaminterne Kommunikation gefördert.

Das Board ist ebenfalls ein wichtiger Bestandteil des Daily Scrums. Mithilfe des Boards kann einerseits das Daily Scrum thematisch geordnet werden, andererseits nutzen viele Teams das Daily Scrum, um das Board zu aktualisieren.



## Burndown Charts

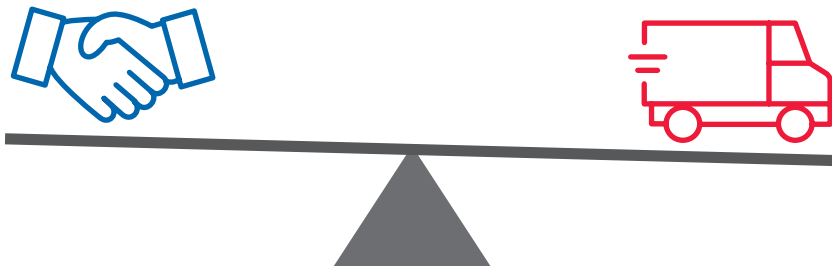
Im Burndown Chart wird der konkrete zeitliche Verlauf eines Sprints festgehalten und mit dem ideal angenommenen Verlauf verglichen.

Auf der x-Achse wird meist die Zeit (in Tagen) eingetragen, auf der y-Achse die Anzahl der im Sprint zu bearbeitenden Tasks bzw. die Story Points der zu bearbeitenden User Stories.

Die Tasks bzw. Story Points werden gleichmäßig auf den Sprint verteilt, sodass im Idealfall am Sprintende alles bearbeitet sein sollte.

Durch das Burndown Chart ist zu jedem Zeitpunkt des Sprints ersichtlich, inwiefern der geplante dem tatsächlichen Fortschritt entspricht.

Ein Burndown Chart kann zusätzlich für Epics erstellt werden. Ebenfalls kann das Diagramm mit einer steigenden statt einer fallenden Kurve visualisiert werden. In diesem Fall spricht man vom einem „Burnup Chart“.



## Commitment & Delivery

Während Burndown Charts die Geschwindigkeit innerhalb eines Sprints darstellen, geben die Commitment und Delivery Charts einen Überblick über die Leistung in mehreren (konsekutiven) Sprints.

Sie basieren auf den Ergebnissen der Burndown Charts der einzelnen Sprints und visualisieren, wie sich die Aufwandsschätzungen (Commitment) im Vergleich zu den tatsächlich gelieferten Ergebnissen (Delivery) entwickeln.

Idealerweise gleichen sich die beiden Werte mit fortschreitender Projektdauer aneinander an. Werden die Schwankungen jedoch mit der Zeit stärker, oder die Abweichungen von Commitment und Delivery bleiben gleich, sollte das Scrum-Team (Product Owner, DEV-Team und Scrum Master) nach Ursachen suchen und diese beheben.



bridgingIT Planning Poker App  
kostenlos herunterladen



iOS



Microsoft



Android

# Planning Poker

Das Planning Poker ist eine etablierte Methode im Rahmen des Sprint Planning 1 zur Einschätzung von Zeitaufwänden und Komplexität von User Stories.

Das DEV-Team, eventuell unterstützt durch externe Experten, bekommt die User Stories durch den Product Owner vorgestellt und schätzt – jeder für sich – die Größe der Story anhand vorgegebener Werte ein. Diese können z. B. auf Spielkarten abgedruckt oder mithilfe einer App visualisiert werden.

Treten große Abweichungen zwischen den Einschätzungen auf, werden die minimalen und maximalen Einschätzungen in einer kurzen Diskussionsrunde erläutert und anschließend neu geschätzt. Dieser Vorgang wird so lange wiederholt, bis ein Konsens über Komplexität und Aufwand entsteht (in der Praxis haben sich zwei bis maximal drei Runden bewährt). Kommt keine Konsensmeinung zustande, wird ein Mittelwert aus allen Schätzungen gebildet.





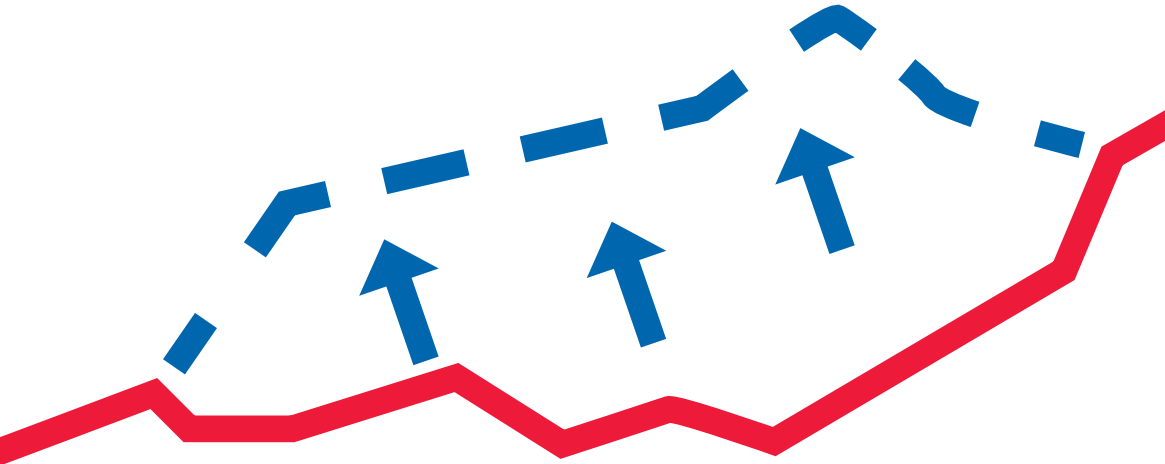
# Timeboxing

Timeboxing ist ein grundlegendes Scrum-Prinzip, bei dem vereinbarte Zeitfenster (Timeboxes) strikt eingehalten werden. Hierbei ist wichtig, zu verstehen, dass eine Timebox lediglich die maximale Dauer definiert.

Dies gilt sowohl für die Länge von Sprints als auch für die Länge aller Treffen, wie z. B. der Daily Scrums oder dem Review.

Die strikte Einhaltung dieser Zeiträume erfordert Übung und Hingabe aller beteiligten Mitglieder und führt vor allem am Anfang der Arbeit mit Scrum zu vielen Problemen. Bei erfahrenen Teams kommt es durch die konsequente Einhaltung zu einer deutlichen Effizienzsteigerung in Meetings.

Der Scrum Master ist für die Einhaltung der Zeitfenster hauptverantwortlich. Aber auch die restlichen Teammitglieder müssen sich der Einhaltung der Timeboxes verschreiben, und durch Priorisierung und kurze Diskussionsrunden zur Umsetzung beitragen.



## Grenzen und Erweiterungen

Scrum ist als Methode für kleine Teams mit maximal neun Teammitgliedern entwickelt worden und wurde ursprünglich für die Softwareentwicklung konzipiert. Es zeigt sich jedoch zunehmend, dass agile Arbeit mit Scrum in sehr vielen Bereichen die Motivation und Produktivität steigern kann.

Um Scrum auch in anderen Bereichen erfolgreich einsetzen zu können, werden je nach Branche unterschiedliche Anpassungen am Framework vorgenommen.

Werden für ein Projekt mehr Personen benötigt, oder soll Scrum großflächig in einem Unternehmen eingesetzt werden, kommen weitere organisatorische Positionen und Meetings dazu. Die Anzahl der zusätzlichen Ebenen und die Menge an weiteren Managern und Meetings ergeben sich nach der Menge der zu organisierenden Teams und dem verwendeten Framework.

„Scrum is like inviting your disliked mother in law.“

– Ken Schwaber

## Scrum

### Hauptmerkmale

- neuer Ansatz der Projektorganisation
- selbstständige, interdisziplinäre Teams
- Einteilung des Projekts in kurze Unterabschnitte

### Stärken

- höhere Motivation der Mitarbeiter
- erfolgreichere Projekte
- permanente Klarheit über Projektstand, Ressourcen und aktuelle Aufgaben
- schnell gelernt ...

### Schwächen

- ... aber schwierig zu meistern!
- große Anzahl an persönlichen Treffen
- Voraussetzung: Verankern des Agilen Manifests im Unternehmen

## Nexus

### Hauptmerkmale

- Skalierung über Teams
- Integrationsteam: Nexus Integration Team
- ideal für 3–9 agile Scrum-Teams
- 1 Produkt, 1 Backlog, 1 PO, 1–10 SM
- minimale Änderungen an Scrum

### Stärken

- schnell zu lernen weil leichtgewichtig
- wenige explizite Rollen und Events
- initiale Skalierung und Beschleunigung variabel

### Schwächen

- keine Adressierung des Managements
- nur Sprint-, keine Release-Orientierung

## SAFe

### Hauptmerkmale

- mehrere Ebenen (Portfolio, Large Solution, Programm, Teams)
- verbindet mehrere agile & klassische Methoden
- Organisations-, Unternehmensansatz

### Stärken

- wird von hierarchischen Organisationen ggf. eher akzeptiert
- Vergleichbarkeit und viele Ressourcen

### Schwächen

- gilt als wenig agil, da Agilität hauptsächlich auf Team-Ebene erfolgt
- schwergewichtig, aufwändig zu implementieren



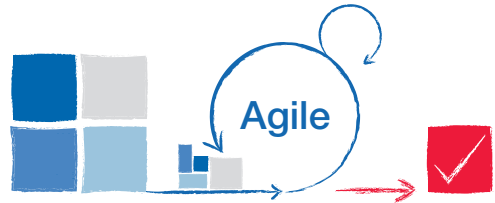
# Über bridgingIT

Menschen stehen bei bridgingIT im Mittelpunkt, denn sie entwickeln nach bewährten und innovativen Methoden spezifische Lösungen für unsere Kunden. Dabei setzen wir seit unserer Gründung 2008 gezielt auf höchste Qualität, Professionalität und Innovationskraft. Als Full Service Provider garantieren wir zudem eine ganzheitliche Betrachtung – von der Anforderung über die Projektierung bis zum Betrieb. Dabei steht die Umsetzung von fachlichen sowie technischen Anforderungen im Fokus, somit schlagen wir die Brücke zwischen Fachseite und IT.

Wir sind darauf ausgerichtet, hochqualifizierte und unternehmerisch ausgerichtete Persönlichkeiten für unser Unternehmen zu begeistern und sie langfristig zu binden. Durch unsere geringe Mitarbeiterfluktuation ergeben sich nachhaltige Partnerschaften und es entsteht eine offene Zusammenarbeit. Unsere Kunden sind für uns ein starker Partner, mit dem wir an einem Strang ziehen - für die erfolgreiche Durchführung und Gewährleistung des Projekterfolges.

Unsere mittlerweile mehr als 500 festangestellten Mitarbeiter beraten an den Standorten Mannheim, Stuttgart, Frankfurt, Karlsruhe, Köln, München, Zug und Nürnberg Kunden des gehobenen Mittelstandes und Konzernkunden bei der Umsetzung von Unternehmensstrategien und dem Einsatz modernster Technologien mit einem herstellerunabhängigen Beratungsansatz.

## Passende Navigation und Lösungsfindung für die zu agilisierende Organisation in Form von:



- Organisations-, Programm- & Projektberatung, -begleitung, -unterstützung
- Vermittlung agiler Arbeitsweisen durch bspw. LEGO-Scrum-Simulation
- im Kontext BPM bspw. Workshops zur Prozesszieldefinition, Festlegung Prozessmethodenframework, Ist-Prozessaufnahme, Erarbeitung Rollen- und Gremienkonzepte, Soll-Prozess-Entwicklung, Prozessmessung, -analyse & -optimierung
- im Kontext Management of Change bspw. Auftragsklärung (Vision & Ziele), systemische Organisationsdiagnose, Aufbau Interventionskatalog, Stakeholdermanagement, Teamentwicklung, Konfliktmanagement, Blended Learning, Mentoring & Coaching



## Agilität@bridgingIT

Wer Scrum bereits in einem Team eingeführt hat, kennt einige der Stolpersteine auf dem agilen Weg. Menschen müssen sich auf andere Methoden einstellen und in der täglichen Arbeit zunächst viele kleine Schritte bis zur „finalen“ Lösung machen. Ein einfaches copy & paste eines Standards wird hierbei nicht funktionieren. Im Kontext der Wege unserer Kunden, integrieren wir daher bereits initial und begleitend verschiedenste Methoden und Instrumente des Prozess- und Changemanagements.

Durch die von uns angewandten Methoden des Prozessmanagements wird eine Transparenz für den gesamten Projektkontext erzeugt. Die so geschaffene Transparenz bildet die Grundlage z. B. für Anforderungsdefinitionen, für eine Steigerung der Reaktionsfähigkeit auf systemische innere und äußere Einflussfaktoren, für eine Komplexitätsreduktion und eine Risikominimierung und ermöglicht dadurch eine zielgerichtete Steuerungsfähigkeit in Projekten.

Das organisatorische Change-Management hingegen ermöglicht es, den Veränderungsprozess in Transformationsprojekten mit zielgerichteter Kommunikation zu begleiten und zu beschleunigen, Widerstände in der Organisation im Projektkontext zu erkennen und durch konkrete Change-Maßnahmen abzubauen und auftretende Konflikte zu lösen.



## Standorte bridgingIT Familie

▶▶ **fastahead**

[www.fastahead.com](http://www.fastahead.com)

Berlin



Heidelberg Mobil

[www.heidelberg-mobil.com](http://www.heidelberg-mobil.com)

Heidelberg

 **bridging IT**

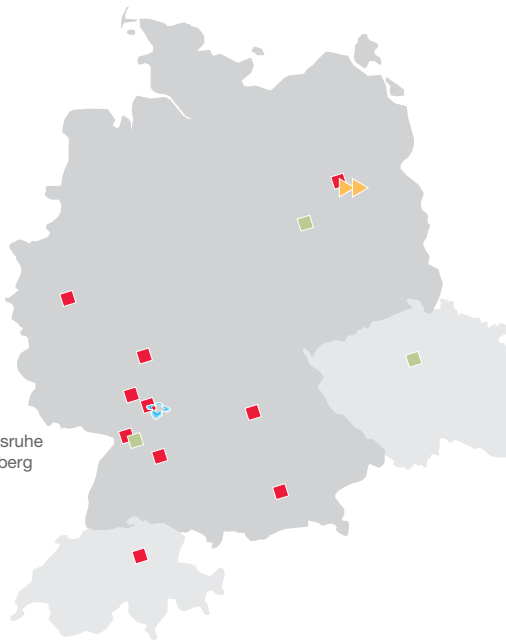
[www.bridging-it.de](http://www.bridging-it.de)

Berlin | Frankfurt | Heidelberg | Karlsruhe  
Köln | Mannheim | München | Nürnberg  
Stuttgart | Zug (CH)

 **crafting IT**

[www.crafting-it.de](http://www.crafting-it.de)

Karlsruhe | Magdeburg | Prag





bridging IT

Menschen Methoden Lösungen