

Wieso agiles Projektmanagement?

Die heutige Arbeitswelt ist von sich ständig ändernden Anforderungen geprägt. Dies betrifft natürlich auch Projekte, weswegen das klassische Projektmanagement durch seine starren und unflexiblen Prozesse bzw. Strukturen oftmals an seine Grenzen gerät. In dieser häufig als VUCA*-Welt beschriebenen Umgebung werden Produkte und Dienstleistungen durch die zunehmende Digitalisierung umfangreicher und komplexer. Zudem sind Märkte und Branchen wesentlich schnelllebiger geworden.

Der Erfolg agiler Methoden basiert u.a. auf hervorragender Teamarbeit, kurzen Feedbackschleifen, hoher Reaktionsfähigkeit bei Änderungswünschen und kontinuierlicher Prozessverbesserung.

Agiles Projektmanagement bietet daher den optimalen Rahmen, um sich an die Schnelllebigkeit und Komplexität der äußeren Umstände jederzeit anzupassen.

*VUCA ist ein Akronym, das sich auf "volatility" ("Volatilität"), "uncertainty" ("Unsicherheit"), "complexity" ("Komplexität") und "ambiguity" ("Mehrdeutigkeit") bezieht.

Agiles PM	Klassisches PM
<ul style="list-style-type: none">• Veränderungen zur Laufzeit werden akzeptiert und antizipiert• angepasste Planung (vom Groben ins Feine)• iterative Wiederholung von Projektphasen• betont Selbstorganisation von Teams• frühe Lieferung von nutzbaren (!) Teilergebnissen	<ul style="list-style-type: none">• Veränderungen zur Laufzeit sollten vermieden werden• starke Vorausplanung• einmalige Abfolge von Projektphasen• im Regelfall auf Befehlssteuerung ausgelegt• späte Lieferung des Gesamtergebnisses

Frameworks

In diesem Abschnitt geben wir Dir einen Einblick in die gängigsten agilen Frameworks und Vorgehensweisen.

Diese basieren im Wesentlichen auf einem agilen Mindset, dessen Grundlage wiederum das Manifest für Agile Softwareentwicklung bildet.

Besonders wichtig ist hierbei das **iterativ-inkrementelle Vorgehen** (d.h. schrittweise und wiederholend; Sprints). Nach jeder wiederholten Anwendung des gleichen Prozesses auf bereits gewonnene Zwischenergebnisse wird ein potentiell lieferfähiges Inkrement fertiggestellt.

Fest verankert ist auch das **Inspect and Adapt**-Prinzip zur permanenten Verbesserung. Im Gegensatz zum klassischen Projektmanagement, bei dem häufig am Ende des Projektes eine Lessons-Learned-Sitzung durchgeführt wird, inspizieren (inspect) die Teams im agilen Projektmanagement regelmäßig den aktuellen Stand (Review) und die gemeinsame Zusammenarbeit (Retrospektive). Daraus leiten sie Maßnahmen zur möglichen Verbesserung ab und setzen diese um (adapt).

Frameworks

Agiles Mindset



Individuen und Interaktionen sind wichtiger als Prozesse und Tools



Zusammenarbeit mit dem Kunden ist wichtiger als die Vertragsverhandlung



Funktionierende Software ist wichtiger als eine umfassende Dokumentation



Reagieren auf Veränderung ist wichtiger als das Festhalten an einem Plan

Diese vier Kernaussagen werden in zwölf Prinzipien genauer ausgeführt.

12 Prinzipien

Diese machen deutlich, dass die Anfänge des Agilen Projektmanagements geprägt waren von einer Ablehnung des traditionellen Projektmanagements mit intensiver Planung, Überwachung und Steuerung.

1. Unsere höchste Priorität ist es, Kundinnen und Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software (Produkte) zufrieden zu stellen.
2. Radikale Anforderungsänderungen sind selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil der Kundinnen und Kunden.
3. Liefere funktionierende Software (Produkte) regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.

4. **Fachkundige und Entwickelnde müssen während des Projektes täglich zusammenarbeiten.**
5. **Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen.**
6. **Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.**
7. **Funktionierende Software ist das wichtigste Fortschrittsmaß.**
8. **Agile Prozesse fördern eine nachhaltige Entwicklung. Die Auftraggebenden, Entwickelnden und Benutzenden sollten ein gleichmäßiges Tempo auf unbegrenzte Zeit halten können.**
9. **Ständiges Augenmerk auf technische Exzellenz und gutes Design fördert Agilität.**
10. **Einfachheit – die Kunst, die Menge nicht getaner Arbeit zu maximieren – ist essenziell.**
11. **Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams.**
12. **In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann und passt sein Verhalten entsprechend.**

Scrum

Einführung

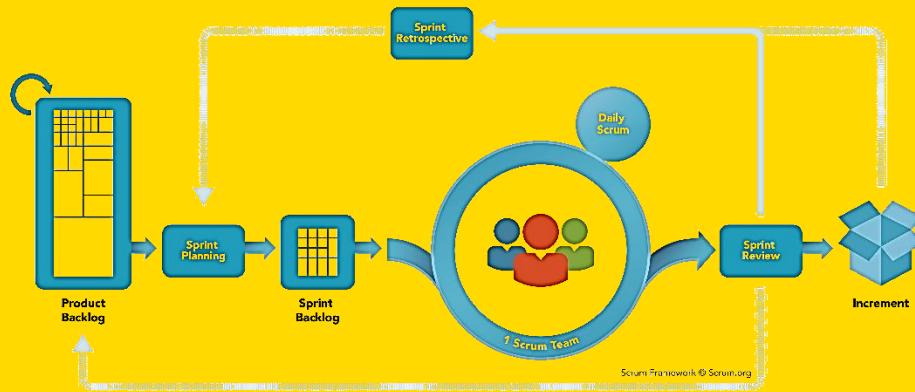
Der Begriff stammt aus dem Rugby. Dort bezeichnet "Scrum" einen dichten Haufen von Spielenden, die sich um den Ball scharen (deutsch: Gedränge).

Dieses Bild wird auf das Projektmanagement übertragen. Scrum soll ausdrücken, worum es beim Projektmanagement vor allem gehen soll: Flexibilität, Dynamik und tägliche Meetings, in denen die Projektmitarbeitenden ihre Aufgaben abstimmen.

Ursprünglich war Scrum ein Vorgehensmodell bei der Softwareentwicklung. Die Vorgehensweise und die damit verbundenen Methoden und Werkzeuge wurden allgemein auf das Projektmanagement übertragen. Daraus entwickelte sich der Begriff agiles Projektmanagement als Abgrenzung zum klassischen Projektmanagement.

Scrum Prozess im Überblick

SCRUM FRAMEWORK



Scrum.org

Grundlagen des Sprints

Scrum zeichnet sich vor allem durch regelmäßige und sich wiederholende Arbeitsabläufe aus. Diese Zyklen werden meist Iteration oder Sprint genannt und sind zeitlich beschränkt. Ursprünglich wurden Sprints in der agilen Software-Entwicklung auf 30 Tage terminiert. Mittlerweile geht der Trend verstärkt in die Richtung von zwei- oder dreiwöchigen Sprints.

Ziel eines jeden Sprints ist es, ein funktionsfähiges Zwischenprodukt, das auch Product-Increment genannt wird, zu entwickeln. Da die Zeit eines Sprints beschränkt ist, kann sich das Scrum Team jeweils nur auf die Weiterentwicklung der grundlegenden Funktionalität des Zwischenprodukts und auf kurzfristige Ziele konzentrieren. Dies motiviert auch die Product Owner, sich auf die wichtigsten Funktionen des Zwischenprodukts zu beschränken.

Die einzelnen Sprints bauen immer aufeinander auf, wobei sehr oft Entwicklungen aus vorherigen Sprints ersetzt oder als „*Lessons Learned*“ wieder verworfen werden.

Ein Sprint besteht, unabhängig von seiner Dauer, aus folgenden fünf Aktivitäten:

- Sprint Planning, dem Planen des Sprints
- Daily Scrum, dem täglichen Treffen des Entwicklungsteams
- Umsetzen der Backlog Items für das Inkrement
- Sprint Review, der Präsentation des erstellten Inkrements
- Sprint Retrospektive, der Analyse und Bewertung des methodischen Vorgehens während des Sprints sowie der Planung und Vereinbarung von Verbesserungsmaßnahmen

Scaled Agile Framework - SAFe

Warum Scaled Agile Framework?

Von Scrum hast Du sicherlich bereits gehört. Dahinter verbirgt sich ein agiles Vorgehensmodell für das Projekt- und Produktmanagement, welches ursprünglich für die Umsetzung von Softwareprojekten entwickelt wurde. Daraus hat sich der Ansatz zum de facto Standard für agiles Projektmanagement entwickelt. Scrum ist bekannt, gut dokumentiert und erfreut sich international einer wachsenden Beliebtheit.

Nur Scrum hat eine Schwäche. Es macht keine Aussage darüber, wie mehrere voneinander abhängige Teams parallel an einem Projekt bzw. Produkt arbeiten können. Diese Schwäche behebt der Scaled Agile Framework, kurz SAFe. Bei SAFe handelt es sich um einen Framework, um große, agile Initiativen und Projekte zu harmonisieren.

SAFe führt Praktiken aus dem Lean und Agile Management zusammen. Der Framework fungiert als Wissensbasis für Organisations- und Arbeitsmuster. Er fördert das Abgleichen, die Zusammenarbeit und das Ausliefern von Ergebnissen einer großen Anzahl von parallel arbeitenden agilen Projektteams.

SAFe wurde für den Einsatz unter ganz unterschiedlich komplexen Bedingungen konzipiert. Dafür gibt es vier verschiedene Konfigurationen mit jeweils dazu passenden Rollen, Artefakten und Aktivitäten.

[Unter diesem Link findest Du immer die aktuellste Version dieses Frameworks](#)

Wie ist SAFe aufgebaut?

Aufbau – Grundprinzipien und Werte von SAFe

Dem Framework liegen neun lean und agile Prinzipien zu Grunde, welche die Rollen, Aktivitäten und (Zwischen-)Ergebnisse des Ansatzes definieren:

1. Eine wirtschaftliche Perspektive einnehmen
2. Systemisches Denken anwenden
3. Variabilität annehmen und Optionen vorhalten
4. Inkrementell mit schnellen integrierten Lernzyklen entwickeln
5. Meilensteine basieren auf der objektiven Bewertung von funktionierenden Systemen
6. Unfertige Ergebnisse visualisieren und limitieren, Losgrößen reduzieren sowie Warteschlangen managen
7. Einen kontinuierlichen Rhythmus anwenden, mit der übergreifenden Planung synchronisieren
8. Die intrinsische Motivation von Wissensarbeitern erschließen
9. Fällen von Entscheidungen dezentralisieren

Diese neun Prinzipien sind angelehnt an Ideen und Prinzipien aus dem Agilen Manifest.

Aufbau – Grundprinzipien und Werte von SAFe

Die vier Kernwerte von SAFe lauten:

- alignment
- built-in quality
- transparency
- program execution

Konfigurationen von SAFe

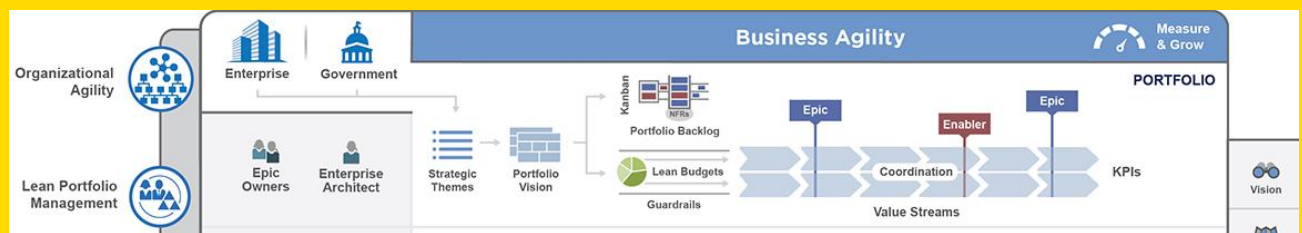
SAFe bietet vier Konfigurationen mit verschiedenen Vorgaben für Projekt- bzw. Teamgröße sowie Rollen (zur Detailansicht dieser bitte auf die Links klicken):

- Essential SAFe – Basiskonfiguration für parallel arbeitende Scrum Teams
- Portfolio SAFe – zusätzlich zu Essential SAFe mit Strategie-, Investitions-, Governance- und Portfolio Management-Aufgaben.
- Large Solution SAFe – zusätzlich zu Portfolio SAFe für Unternehmen, die große und komplexe Lösungen entwickeln.
- Full SAFe – zusätzlich zu Large Solution SAFe für Unternehmen, die große und integrierte Lösungen in Teams von mehreren hundert Personen entwickeln.

**Je umfangreicher die Skalierung und je komplexer die Anforderungen, desto mehr Ebenen sollten zur Erledigung der Aufgaben pro Stufe einbezogen werden:
Die Ebenen im Einzelnen:**

Portfolioebene:

Auf dieser Ebene erfolgt die strategische Sicht auf Produktentwicklung, Budgetzuordnung und Abgleich mit Unternehmenszielen.



Large-Solution-Ebene: Die allergrößten Produktentwicklungsunterfangen erhalten mit dieser Ebene einen eigenen „Release Train“ für die Kontrolle des Wertstroms. So soll SAFe den Anforderungen besonders großer und komplexer Programme gerechter werden.

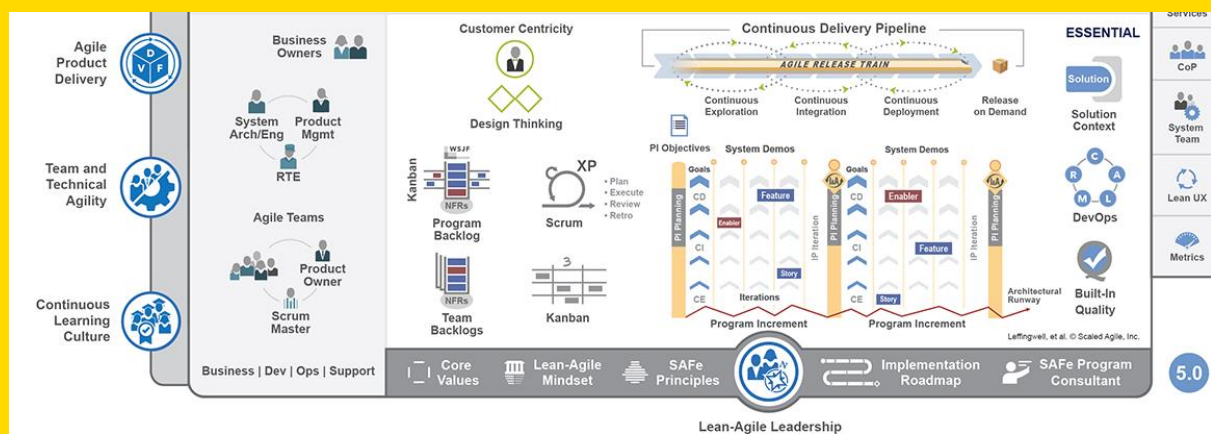
Die Large Solution Ebene und Portfolio Ebene sind für große Organisationen bzw. Vorhaben geeignet.



Essential: SAFe nutzt das normale Scrum Team Setup und wandelt dies ein wenig ab. Pro Team kümmern sich fünf bis neun (statt in Scrum drei bis neun) Entwickler um die Umsetzung, begleitet durch einen Product Owner und einen Scrum Master für je circa 1-2 Umsetzerteams. Außerdem arbeitet der PO dem übergeordneten Produktmanagement auf Programmebene zu. Die Teams arbeiten dabei funktionsübergreifend und weitestgehend selbstorganisiert.

Die Essential-Ebene enthält die identischen Rollen wie das Vorgehensmodell Scrum, daher Product Owner, Scrum Master und Entwicklungsteam. Auf der Program-Ebene agieren zudem weitere Rollen:

**Je umfangreicher die Skalierung und je komplexer die Anforderungen, desto mehr Ebenen sollten zur Erledigung der Aufgaben pro Stufe einbezogen werden:
Die Ebenen im Einzelnen:**



Möchte man SAFe als Rahmenwerk einführen, sollte man sich zunächst die nachfolgenden zwei Fragen beantworten:

- Wie stufen wir unsere Vorhaben im Hinblick auf Komplexität und Größe ein? Hier kannst Du die untenstehende Tabelle zu Rate ziehen. Bei Unsicherheit gilt: lieber erst einmal die Umsetzung der Prinzipien in der einfachsten Stufe ausprobieren, bevor ein Ansatz unternehmensweit ausgefahren wird.
- Welche Ebenen sollen einbezogen werden? Geht es uns mehr um die Einzelteam- und Programmebene? Oder betrachten wir unsere Arbeitsweisen gleich auf Portfolioebene? Oder gar für einen ganzen Konzern (Large Solution und bei international agierenden auch Full SAFe)?

SAFe bietet je nachdem, wie Du diese Frage für deine Situation beantwortest, verschiedene Ausbaustufen hinsichtlich Projektkomplexität und -größe. Einen guten Überblick erhältst Du mithilfe der Grafik auf [scaled agile framework](#) – klicke dort auf die einzelnen Reiter, um das Diagramm für jede der Stufen zu sehen.

Ausbaustufe	Ebenen	Ziel/Anforderung
Essential	Team und Programmebene	Geeignet für Unternehmen, die schnell starten wollen.
Portfolio	Team-, Programm- und Portfolioebene	Geeignet für Unternehmen, die ihre Programme mit der Unternehmensstrategie abgleichen wollen
Large Solution	Team-, Programm-, Portfolio-, und Large-Solution-Ebene	Für Unternehmen mit großem Skalierungsbedarf
Full	Team-, Programm-, Portfolio-, und Large-Solution-Ebene	Für Global Player mit großem Skalierungsbedarf – sowohl für Projekte als auch Mitarbeiter

Im Agilen Projektmanagement gibt es drei verschiedene Kernrollen



Product Owner

- Für die Wertmaximierung des Produkts verantwortlich
- Repräsentant der Stakeholder
- Eine einzelne Person (kein Komitee)
- Die Organisation muss seine Entscheidungen respektieren
- Definiert die Produktvision
- Entscheidet über Produktanforderungen und priorisiert deren Abarbeitungsreihenfolge
- Ist für die Beschreibung und die Verfeinerung der Anforderungen verantwortlich
- Macht das Product Backlog sichtbar und transparent
- Stellt das Verständnis des Teams bezogen auf Backlog Einträge und Produktvision sicher
- Definiert die Releases
- Ist fachlich versiert und überlässt die Technik (das Wie) dem Team
- Ist für das Team verfügbar aber nicht „Chef“ des Teams
- Verantwortet das dem Product zugeteilte Budget
- Vertritt die Interessen des Kunden bzw. der Stakeholder



Scrum Master

- Ist verantwortlich für das Verständnis und die Durchführung von Scrum
 - Sorgt für die Arbeitsfähigkeit der Product Owner, der Entwicklungsteams und der Organisation
 - Sorgt für Informationsfluss zwischen Product Owner und Team
 - Ist ein versierter Moderator (z.B. Daily und Retro)
 - Kann gut organisieren und Regeln durchsetzen
 - Nimmt sich selbst zurück und hat Spaß am Erfolg Anderer
 - Hält Spannungen aus und kann Konflikte lösen
 - Ist Vermittler und Unterstützer
- dient dem Product-Owner durch das:**
- Vermitteln von Techniken/Methoden für die optimale Gestaltung des Product-Backlogs
 - Vermitteln der Notwendigkeit klarer, prägnanter Backlog-Einträge
- dient dem Entwicklungsteam durch das:**
- Coachen zur Selbstorganisation
 - Beseitigen von Hindernissen (Impediments)
 - Unterstützen bei der Durchführung von Scrum Ereignissen bei Bedarf bzw. auf Anfrage
- dient der Organisation durch das:**
- Leiten und Coachen bei der Einführung von Scrum
 - Planen von Scrum-Implementierungen innerhalb der Organisation
 - Zusammenarbeiten mit anderen Scrum-Mastern



Development Team

- Ist immer als Ganzes verantwortlich
- Sorgt für die Umsetzung der zu erledigenden Aufgaben
- Besteht aus mindestens drei, maximal neun Profis
- Organisiert sich selbst und entscheidet - innerhalb der vom Product Owner gegebenen strategischen Rahmenbedingungen - autonom
- Ist interdisziplinär (cross-functional)
- Hat keine Titel außer „Entwickler“
- Hat keine Unterteilungen innerhalb des Teams

Artefakte

Backlog

- Das **Product Backlog** enthält alle Anforderungen an das Produkt formuliert in sogenannten User Stories
- Es wird im Laufe der Projektlaufzeit vom Product Owner weiterentwickelt und gepflegt
- Aus dem Product Backlog zieht sich das Team User Stories für den nächsten Sprint in das **Sprint Backlog**
- Die User Stories im Sprint Backlog werden im Sprint Planning nach Aufwand geschätzt (meist in Storypoints) und gegebenenfalls weiter ausdefiniert (z.B. in Tasks zerlegt). Der Product Owner ist dabei anwesend

Definition of Done

- Legt fest welche Anforderungen erfüllt sein müssen, damit eine Anforderung bzw. User Story abgeschlossen werden kann (Akzeptanzkriterien). Beispiel: Die Funktion wurde auf einer Testumgebung installiert und getestet.
- Sie wird gemeinsam vom Team und dem Product Owner festgelegt.

Inkrement

- Die Summe aller nach einem Sprint gemäß **Definition of Done** fertiggestellten Anforderungen (bzw. User Stories) sowie der bereits zuvor fertiggestellten (aktueller Stand des Produktes)
- Das **Inkrement** muss „done“, also potentiell nutzbar und veröffentlichbar („releasable“) sein

Kanban

Einführung

Kanban ist neben Scrum eine der am weitesten verbreiteten agilen Managementmethoden. Wann wählt man am besten die eine, wann die andere Methode? Was sind überhaupt die Vorteile von Kanban und wie findest Du die für dein Projekt beste Kanban Software? Ein Überblick.

Kanban wurde ursprünglich als Methode für die Produktionssteuerung entwickelt, mittlerweile ist die agile Methode auch im Projektmanagement verbreitet. Kanban ermöglicht eine bessere Auslastung, höhere Transparenz und schnellere Iterationen im Projekt.

Definition

Wird in der Produktion der Materialfluss visualisiert, werden im Projektmanagement die Aufgaben dargestellt. Jede Aufgabe wird auf einer farbigen Karte notiert. Je nach Bearbeitungsstatus wandert diese Karte auf einem Board von links nach rechts bis zu ihrer Fertigstellung.

Im klassischen Kanban System wird das Kanban Board in drei Spalten eingeteilt:

To Do: In die linke Spalte werden Aufgaben eingetragen, deren Bearbeitung noch nicht begonnen wurde.

In Progress/Doing: Sobald die Bearbeitung einer Aufgabe beginnt, wird die Karte in die mittlere Spalte verschoben.

Done: Ist die Aufgabe erledigt, wandert die Karte in die rechte Spalte.

Kanban ist ein sehr flexibler Ansatz. Die einzelnen Spalten können an die Anforderungen des Projekts angepasst werden. In manchen Fällen eignen sich vier- oder fünfspaltige Boards besser, um alle Stufen des Entwicklungsprozesses zu erfassen oder müssen für die eigenen Arbeitsweisen passendere Bezeichnungen gefunden werden.

Kanban Boards können analog oder digital geführt werden. Auch eine kombinierte Nutzung von digitalen Tools und physischem Board ist denkbar.

Regeln

Das Kanban System kennt nur wenige Regeln. Um dennoch sicherzustellen, dass die Vorteile der Methodik ausgeschöpft werden, hat David Anderson sechs Prinzipien beschrieben, die Unternehmen beachten sollten:

Regeln

1. Klare Regeln: Alle Regeln für den Prozess sollten explizit gemacht werden, also zum Beispiel sollte definiert werden, was darunter zu verstehen ist, wenn eine Aufgabe den Status "Done" erhält.

2. Aufgabenlimit: Begrenzen Sie die Zahl der zur Verfügung stehenden Karten. So stellen Sie sicher, dass nicht zu viele Aufgaben begonnen, sondern fokussiert Ergebnisse erzielt werden. David Anderson empfiehlt zudem, Karten-Limits pro Station festzulegen.

3. Workflow: Notieren Sie die Aufgaben im Backlog in absteigender Reihenfolge und stellen Sie sicher, dass immer Aufgaben in Bearbeitung sind und ein stetiger Workflow gegeben ist.

4. Kontinuierlicher Verbesserungsprozess (Kaizen): Analysieren Sie Ihre Kanban-Prozesse regelmäßig und bemühen Sie sich darum, die Effizienz Ihrer Arbeitsweise zu verbessern.

5. Leadership: Kanban kann nur funktionieren, wenn Mitarbeitende auf allen Ebenen Verantwortung übernehmen und auch die ausführenden Teammitglieder sich aktiv für die Verbesserung von Abläufen einsetzen.

6. Modelle: Die Verwendung von Modellen wie die der Engpasstheorie, der Komplexitätstheorie oder der Lean IT können dabei unterstützen, ein besseres Verständnis für Prozesse zu erlangen und effizientere Lösungen zu

Scrum vs. Kanban

Neben Kanban ist Scrum eine zweite weit verbreitete agile Projektmanagement-Methode. Wann eignet sich welche Methode besser? Dazu lohnt es sich, die Eigenheiten der beiden Methoden näher zu betrachten. Es wird deutlich, dass beide in keinem Ausschlussverhältnis stehen. Kanban und Scrum können kombiniert eingesetzt werden.

Scrum ist die umfassendere Methode. Das Kanban System bietet zwar nicht nur das häufig verwendete Kanban Board, sondern hat weiteren theoretischen Overhead, doch es lässt sich auf diesen visuellen Aspekt des Aufgabenmanagement reduziert, flexibel mit anderen Methoden kombinieren.

Gemeinsamkeiten	Unterschiede	
<ul style="list-style-type: none"> • agil • nutzen eine Pull-Logik (Teammitglieder wählen ihre Aufgaben aus dem Backlog) • Teams erhalten mehr Verantwortung • begrenzen den Umfang des Work in Progress • zielen auf transparente Prozesse • Projektergebnisse werden in Inkrementen erstellt • regelmäßige Prozess-Analyse, um Abläufe zu optimieren 	<p>Kanban:</p> <ul style="list-style-type: none"> • wenige feste Regeln • für Projekte und Routineaufgaben • auch für die Arbeitsorganisation von Einzelpersonen • ab einer gewissen Projektgröße besteht die Gefahr, dass Kanban unübersichtlich wird (Aufteilung in kleinere Einheiten notwendig) • kontinuierliche Pflege des Kanban Boards • kennt keine festen Rollen • Priorisierung im Backlog optional • Iterationen sind optional 	<p>Scrum:</p> <ul style="list-style-type: none"> • komplexes Regelwerk • nur für Teams ab drei Personen geeignet • nur Projekte, nicht für Routineaufgaben geeignet • Scrum Board wird nach jedem Sprint neu aufgesetzt • feste Rollen (Product Owner, Scrum Master, Team) • Priorisierung im Backlog vorgeschrieben • Iterationen sind vorgeschrieben

Vorteile	Nachteile
<ul style="list-style-type: none"> • Der Vorteil für die Projektarbeit liegt darin, dass der Managementaufwand reduziert und die Abarbeitungsgeschwindigkeit steigt. • Kanban kann mit traditionellen genauso wie mit agilen Managementmethoden eingesetzt werden. • Als agile Methode profitieren Teams auch bei Kanban von kurzen Sprints, sodass Fehlentwicklungen schneller bemerkt werden. • Kanban erhöht die Arbeitsmotivation, da die einzelnen Teams autonomer arbeiten und 	<ul style="list-style-type: none"> • Teammitglieder müssen überlappende Kompetenzen haben, damit keine Engpässe entstehen, wenn einzelne Mitarbeiter ausgelastet sind, sondern verschiedene Kollegen einspringen können, um das Backlog auf Null zu bringen. • Kanban funktioniert nur, wenn sich Arbeitspakete klar in voneinander abgrenzbare Aufgabenschritte gliedern lassen. • Projekte, die mit einer festen Deadline planen, lassen sich besser über Methoden wie Scrum managen, die ein größeres

Vorteile	Nachteile
Mitglieder sich ihre Aufgaben aus dem vorhandenen Pool selbst „holen“	Augenmerk auf das Zeitmanagement legen.

Hybrides Projektmanagement: Das Beste aus Agil und Klassisch

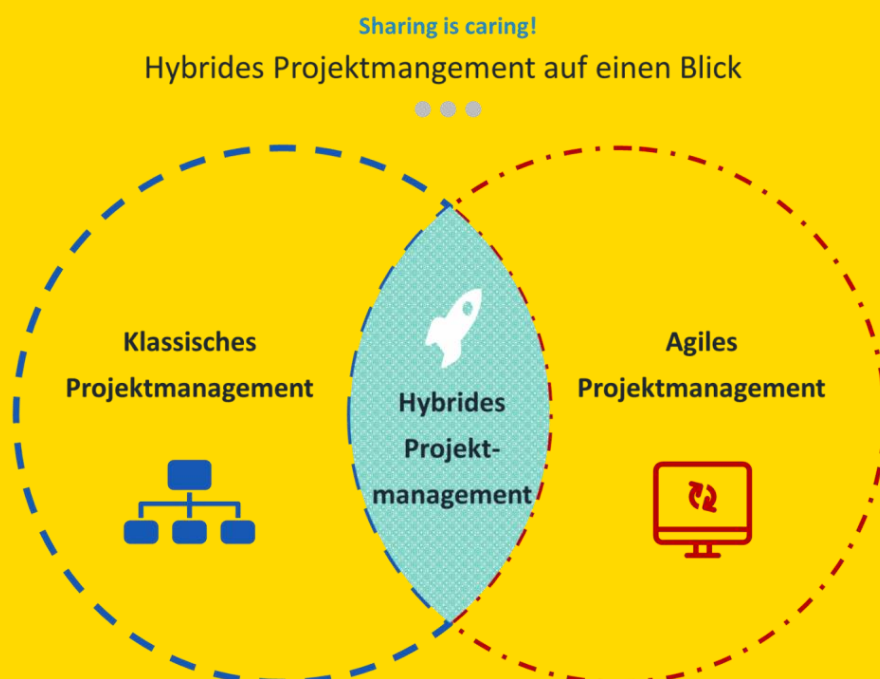
Unter dem hybriden Projektmanagement wird eine Kombination unterschiedlicher Projektmanagement-Methoden oder die Verwendung einzelner Elemente von verschiedenen Ansätzen verstanden. Deutlich wird das, wenn der Begriff „Hybrid“ näher betrachtet wird. Darunter ist eine Kombination oder Vermischung aus etwas zu verstehen. Zum Beispiel:

- In der Automobilindustrie bezeichnen Hybrid-Systeme z.B. eine Kombination aus Benzin- und Elektromotor.
- Eine Hybridsoftware ist eine Software, die auf unterschiedlichen Betriebssystemen läuft.
- Hybriduhren besitzen sowohl eine analoge als auch eine digitale Zeitanzeige

Auf das Projektmanagement übertragen, bezeichnet das hybride Projektmanagement eine Kombination folgender PM-Methoden:

- Klassisches Projektmanagement
- Agiles Projektmanagement

Doch wann sollten welche Elemente genutzt werden und wie wird das hybride Projektmanagement in der Realität bereits praktiziert?



Warum ist eine Kombination der Projektmanagement-Methoden sinnvoll?

Wendet ein Unternehmen das hybride Projektmanagement an, dann finden sich in der Praxis oft die klassischen Methoden ergänzt durch agile Ansätze. Einzelne Vorgänge eines Projekts werden dann zum Beispiel agil nach Scrum durchgeführt. Grund des Einsatzes der agilen Techniken ist meist, vermeintliche Schwächen des klassischen Projektmanagements zu umgehen. Der Nutzen des agilen Projektmanagements liegt dabei insbesondere in folgenden Aspekten:

- Die Mitarbeitenden sollen sich selbst organisieren und damit weniger überlastet sein
- Es gibt sowohl Eigen- als auch Teamverantwortung. Wer für was verantwortlich ist, ist jedoch ganz klar definiert
- Das Team hat gemeinsame Werte und Prinzipien
- Im Vordergrund steht eine iterative Arbeitsweise. Dabei wird zunächst eine vorläufige Version oder ein Prototyp entworfen, für welchen sich das Team ein Feedback einholt. Unter Einbindung des Feedbacks entsteht eine Nachfolgeversion. Es werden so viele Feedback- und Nachbesserungsschleifen wiederholt, bis die Auftraggebenden mit dem Ergebnis zufrieden sind.
- Das Team muss auf den Input der Auftraggebenden reagieren, was eine besonders flexible Arbeitsweise voraussetzt beziehungsweise überhaupt erst ermöglicht

Tools

Auf dieser Seite möchten wir Dir einige hilfreiche Tools für den Alltag im agilen Projektmanagement vorstellen.

Diese können teilweise auch im klassischen oder hybriden Projektmanagement hilfreich sein.

Fokussiert haben wir uns dabei auf Projektmanagement Tools, um die eigentliche Projektarbeit zu managen (z.B. mittels Kanban-Boards), Informationen transparent zu machen und um Retrospektiven durchzuführen.

Hilfreiche Werkzeuge für die tägliche Projektarbeit

Folgende Werkzeuge unterstützen Euch in der täglichen Projektarbeit und sind die Grundlage dafür, alle Aufgaben transparent im Blick zu behalten:

- **JIRA - Confluence**

JIRA ist ein Projektmanagement Tool des Herstellers Atlassian. Es hilft Dir bei der agilen Zusammenarbeit in deinem Team. Du kannst von vielen Vorteilen, wie der Workflow und der agilen Organisation profitieren.

JIRA stellt euch einen Projektraum mit zahlreichen Funktionen zur Verfügung, darunter zählen zum Beispiel Priorisierungen, Schätzungen,

Hilfreiche Werkzeuge für die tägliche Projektarbeit

Kategorisierungen, Berichte, Workflows und Beobachtungsfunktionen. Kombiniert mit den integrierten Benachrichtigungen könnt ihr so eine maximale Transparenz in der Planung erreichen und dadurch die Produktivität im Team steigern.

MS Planner

Microsoft hat uns mit Teams und Planner zwei ausgezeichnete Werkzeuge an die Hand gegeben, mit denen wir nicht nur agile Projektmanagement-Methoden, sondern auch generell agile Arbeitsprozesse einfach abbilden können.

- **Physisches Kanban Board**

- **Timebox**

Timeboxing ist eine Methode des Zeitmanagements bzw. der Organisation, in denen feste Zeitblöcke (die Timebox) für Aufgaben reserviert werden. Neben der physischen Variante "Timekeeper" kann dir folgendes online Tool behilflich sein.

Folgende Werkzeuge haben sich in der Planung und Durchführung von Retrospektiven bewährt:

- **Retromat**

Inspirationen für gute Retrospektiven finden sich im Retromat Du planst Deine nächste Retrospektive? Fang mit einem zufällig generierten Ablauf an, pass ihn an, druck ihn aus und teile die URL mit anderen. Oder schau Dich, falls Du bereits Erfahrungen mit Retrospektiven hast, einfach nach neuen Ideen um.

- **Retrospected**

Ein gutes und einfaches Tool, um Retrospektiven remote durchzuführen ist Retrospected. Es kann selbstverständlich zum Beispiel zur methodischen Abwechslung auch alles Element in einer Retrospektive vor Ort eingesetzt werden.

- **Funretro**

Einfaches Tool für die virtuelle Durchführung einer Retrospektive.

Literatur:

- **Heimrath, Markus: *Agiles Projektmanagement: Scrum für Einsteiger*, 2018**
- **Schwaber, Ken; Sutherland, Jeff: *The Scrum Guide*, 2017**
- **Schwaber, Ken: *Agile Project Management with Scrum (Microsoft Professional)*, 2004**
- **Sutherland, Jeff; Sutherland, J.J.: *Scrum: The Art of Doing Twice the Work in Half the Time*, 2015**

- **Leopold, Klaus:** *Agilität neu denken: Warum agile Teams nichts mit Business Agilität zu tun haben*, 2018
- **Rasmusson, Jonathan:** *The Agile Samurai: How Agile Masters Deliver Great Software*, 2010
- **Sunden, Joakim; Hammarberg, Marcus:** *Kanban in Action*, 2014
- **Kniberg, Henrik:** *Lean from the Trenches: Managing Large-Scale Projects with Kanban*, 2012
- **Kniberg, Henrik:** *Scrum and Xp from the Trenches 2nd Edition*, 2015