



4.5.1 Projektdesign

Teil 2

Agenda

1. Projektdesign
 - Was ist das? / Definition
 - Methodische Vorgehen
 - Komplexität: Stacey Matrix
2. Projektmanagementansätze & Vorgehensmodelle
3. Traditionelle Vorgehensmodelle
 1. Sequentiell
 2. Nebenläufig
 3. Inkrementell
 4. Iterativ
4. Agile Vorgehensmodelle
 1. Scrum
 2. Kanban



Projektdesign

- Das Projektdesign beschreibt den grundsätzlichen Handlungsansatz und die Art und Weise mit dem das Projekt angegangen und bearbeitet werden soll („Wie soll das Projekt geplant und umgesetzt werden?“).

Die ist abhängig u.a. von:

- Projektart, Projektgröße
- Erfolgskriterien
- Erfolgsfaktoren
- Komplexität des Vorhabens
- Projektkultur
- Erfahrungen aus abgeschlossenen Projekten
- Wünschen & Bedürfnissen der Organisation
- Projektmanagementsystem & Standardvorgehensmodellen der Organisation
- PM-Kenntnisse der Projektbeteiligten

Projektdesign

- Das Projektdesign wird zu Beginn des Projekts festgelegt, um den optimalen Ansatz für die Durchführung des Projekts zu definieren.
- Es bildet die Basis für alle Aktivitäten im Projekt. Aus diesem Grund muss der Ansatz noch vor Beginn der Planungen, Organisation und Durchführung festgelegt werden.
- Zitat ICB, 4.5.1: *"Die Wahl des Ansatzes und der Designaktivitäten muss erfolgen, bevor mit den Planungen, der Organisation und der Durchführung des Projekts begonnen wird."*

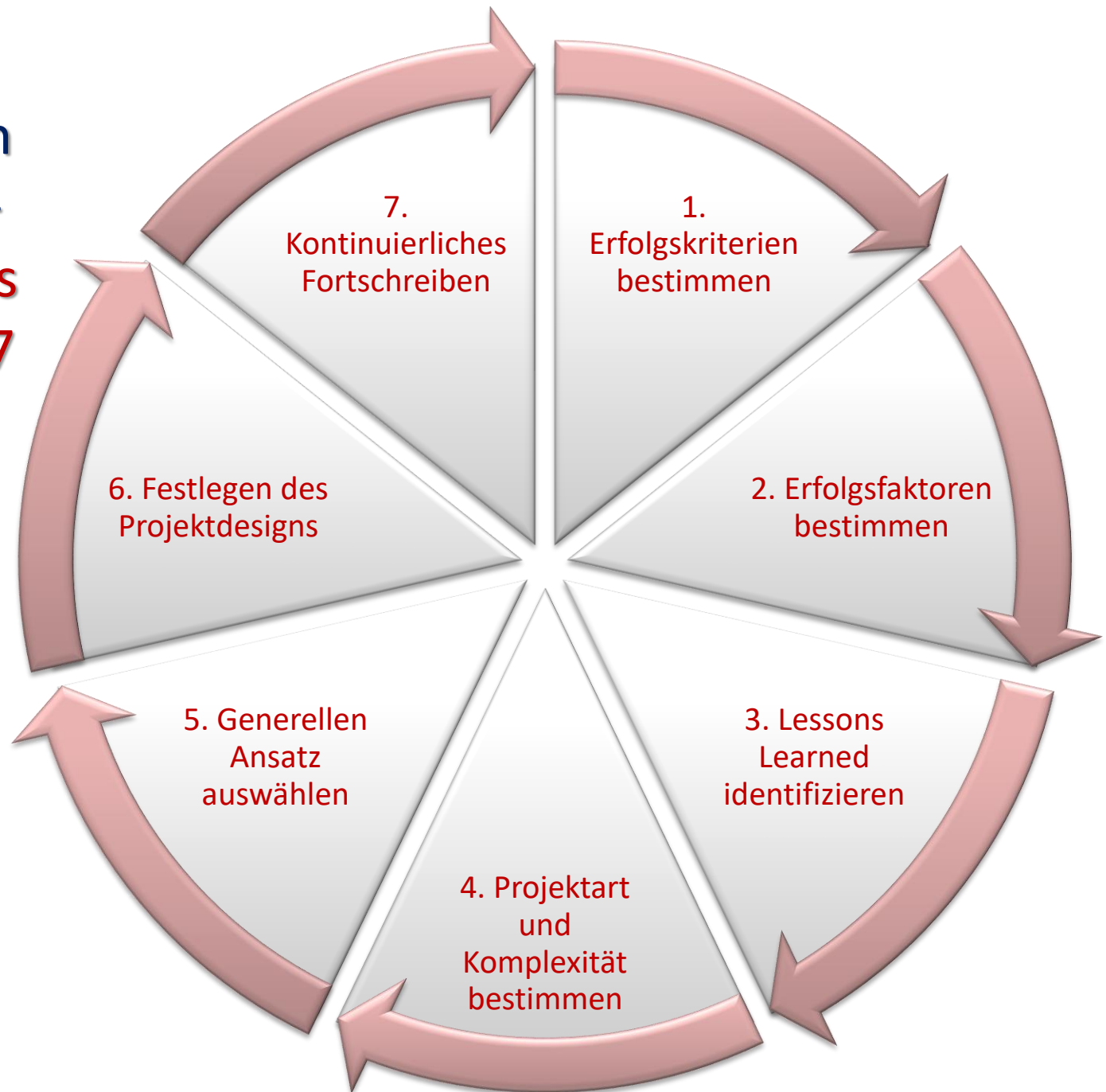
Projektdesign

- Die Projektdesign ist jedoch nicht unabänderlich.
- Da sich Anforderungen und Rahmenbedingungen ändern können, ist es wichtig, das Design regelmäßig neu zu bewerten und wenn nötig anzupassen.
- Zitat ICB, 4.5.1: *"Da sich alle externen Faktoren und Erfolgskriterien (und / oder die Wahrnehmung für deren Relevanz) im Projektverlauf häufig verändern, muss das Projektdesign in regelmäßigen Abständen neu bewertet und wenn nötig angepasst werden."*

Projektdesign

- Beim Projektdesign dreht sich alles um die Wahl des richtigen Ansatzes & Vorgehensweise für ein Projekt.
- **Das Ziel:** Den Erfolg des Projekts und aktive Beteiligung sicherstellen.
- Um dies zu ermöglichen, werden alle Einflüsse auf das Projekt identifiziert und der optimale Ansatz ausgewählt.
- Generell ist der Ansatz der sinnvollste, der die höchste Wahrscheinlichkeit für das Erreichen des Projekterfolgs bietet.
- Bei der Festlegung des Projektdesigns empfiehlt die ICB 4 ein **Methodisches Vorgehen in 7 Schritten.**

Projektdesign entwickeln – Methodisches Vorgehen in 7 Schritten



Projektdesign entwickeln – Methodisches Vorgehen

1. Analysieren der Aufgabenstellung und Zielsetzung des Projekts und Herausarbeiten der Erfolgskriterien
2. Analysieren der Erfolgsfaktoren des Projekts und Ableiten einer Strategie zur positiven Beeinflussung der Erfolgsfaktoren
3. Identifizieren der vorliegenden Erfahrungen mit ähnlichen Projekten (Lessons Learned)
4. Bestimmen der Projektart und Projektkomplexität und Ableiten der Konsequenzen für das Projektdesign
5. Den generellen Ansatz für das Projektmanagement auswählen und festlegen
6. Festlegen des Projektdesigns und Entwerfen einer Strategie für die Information und Kommunikation des Projektdesigns
7. Kontinuierliche Fortschreibung des Projektdesigns während des Projektverlaufs

Komplexität

- Ein Projekt kann unterschiedlich komplex sein: Sowohl das angestrebte Projektergebnis als auch die Prozesse auf dem Weg dahin können unterschiedlich komplex sein. Beides fließt in die Wahl des Projektmanagementansatzes ein.
- Zitat ICB, 4.5.1: *"Bei der Wahl eines geeigneten Ansatzes muss der Einzelne die spezifische Komplexität des Projekts berücksichtigen - die Komplexität der vereinbarten Ergebnisse und / oder der erforderlichen Prozesse des Projekts."*

Folgende Faktoren führen zu einer hohen Komplexität des Projektes:

- Innovative oder technisch komplexe Ergebnisse oder Prozesse
- Viele Beteiligte (mehrere Teams, Abhängigkeiten usw.)
- Unterschiedlichen Interessen der Stakeholder
- Viele Schnittstellen zu anderen Projekten, Prozessen oder Programmen
- Starke Einschränkungen, z. B. extrem knappe Zeitpläne oder Budgets

Komplexität - Stacey Matrix

- Mit Hilfe der Stacey-Matrix kann die Komplexität eines Projekts beurteilt werden.

Auf Basis der zwei Faktoren

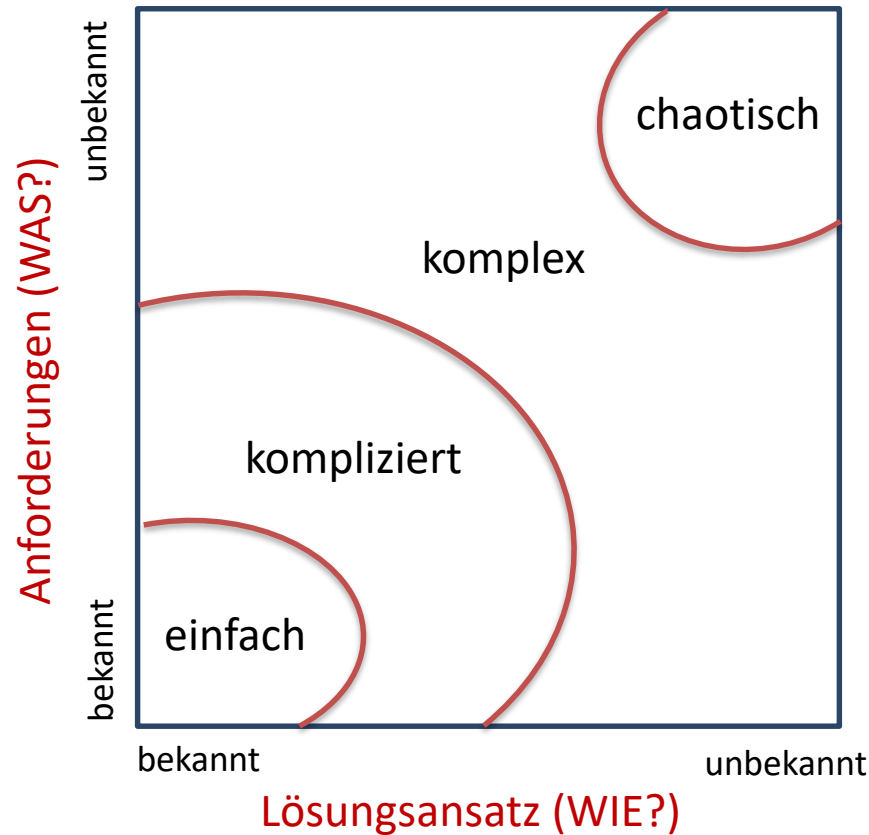
- "Anforderungen (WAS?)" (bekannt bis unbekannt) uns
- „Lösungsansatz (WIE?)" (bekannt bis unbekannt)

werden die Projekte in vier Kategorien eingeordnet:

- einfach,
- kompliziert,
- komplex oder
- chaotisch.

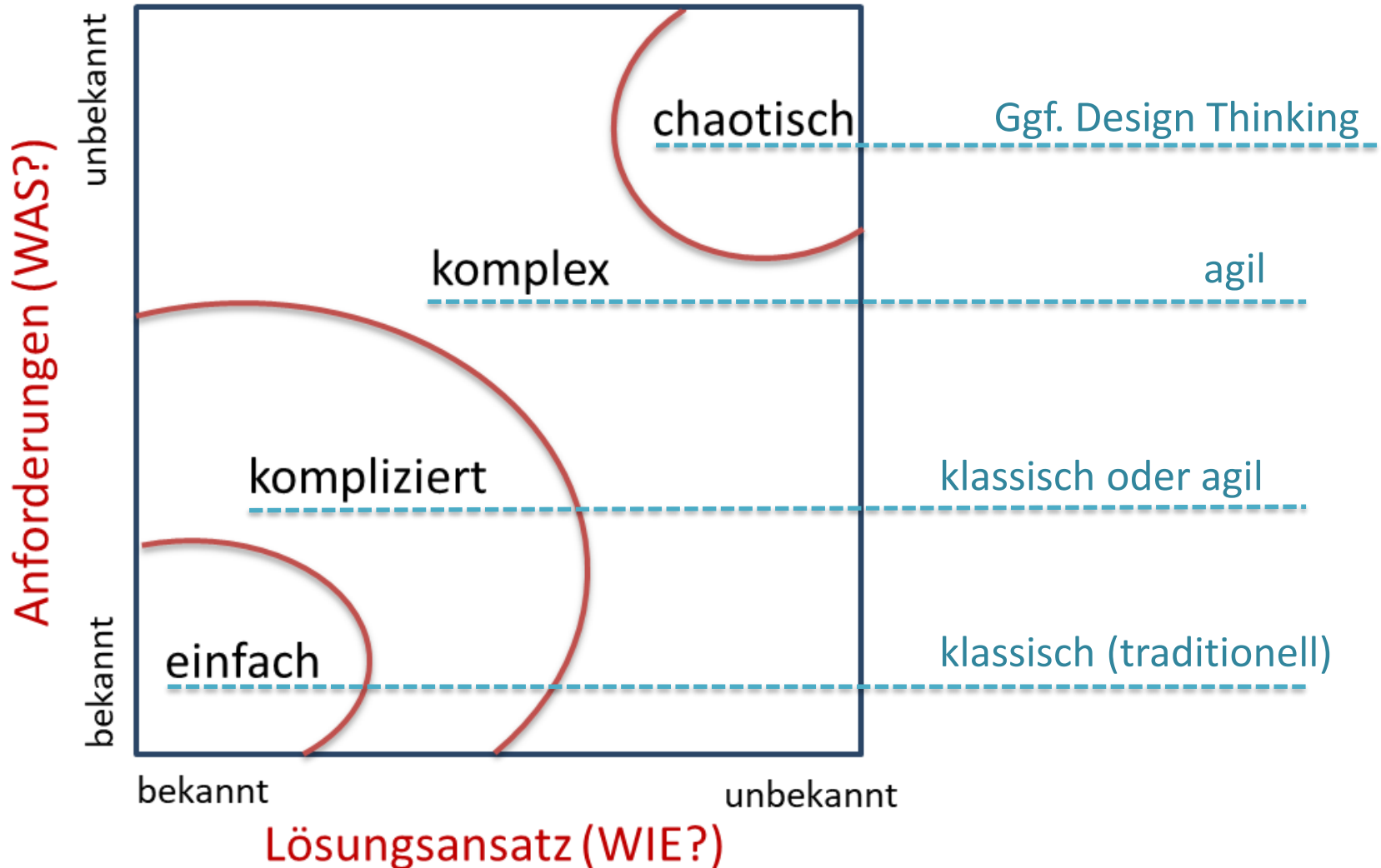
Erkenntnisse dieser Einordnung können in das Projektdesign einfließen und der passende Projektmanagementansatz gewählt werden.

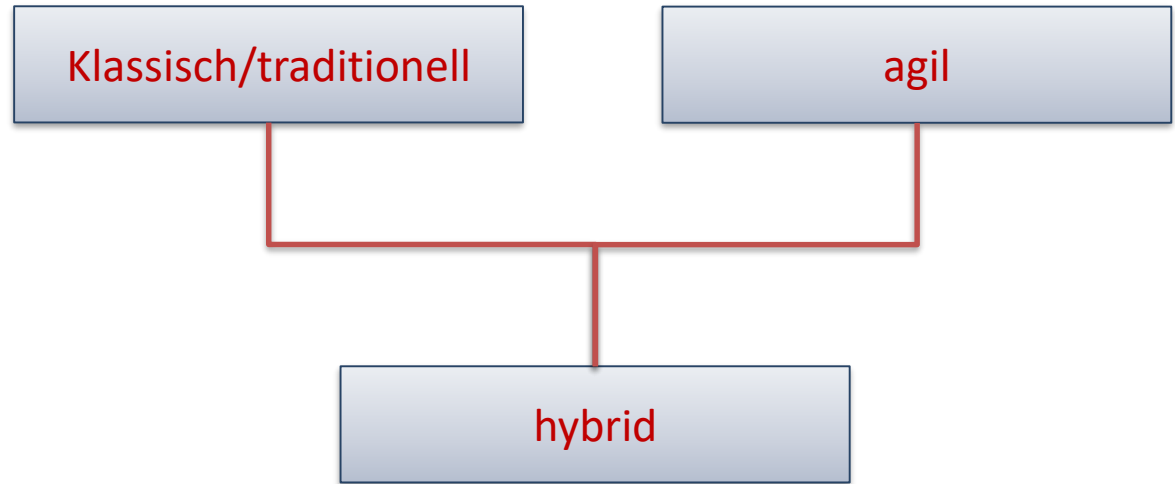
Komplexität – Stacey-Matrix



Die Stacy-Matrix stellt auf den Achsen "Anforderungen (WAS?) bekannt / unbekannt" und "Lösungsansatz (WIE?) bekannt / unbekannt" Komplexität in insgesamt vier Bereichen dar: **einfach**, **kompliziert**, **komplex** und **chaotisch**. Sie gibt Hinweise zur grundsätzlichen Art und Weise des Handlungsansatzes an die Projektaufgabe.

Komplexität – Stacey-Matrix





Projektmanagementansätze & Vorgehensmodelle

Projektmanagementansätze & Vorgehensmodelle

Projektmanagementansätze:

- beschreiben die grundlegende strategische Herangehensweise und Prinzipien, wie ein Projekt gemanagt und geführt wird.

Beispiele sind:

- Traditionell (mit Projektphasen: z.B. sequentiell, nebenläufig, wiederholend)
- Agil
- Hybrid
- Design-to-cost
- Timeboxed
- Engpassorientiert
- Evolutionär

Vorgehensmodelle:

- geben eine konkrete Methodik und Schritte vor, wie das Projekt abgewickelt wird. (praktische Anleitung, „Wie gehen wir vor?“)
- Standardvorgehensmodelle haben Vorteile: Sie sind bewährt, erprobt und sparen Zeit.

Projektmanagementansätze & Vorgehensmodelle

Design-to-Cost:

- zu Beginn wird festgelegt, wie viel das zu erstellende Produkt oder Projektergebnis am Ende kosten darf. Die gesamte Planung und Durchführung des Projektes muss dann darauf ausgerichtet werden.
- **Beispiel:** Bau eines Einfamilienhauses mit festgelegtem Budget.

Engpassorientiert:

- Konzentration auf kritische Aspekte/Ressourcen (Bottlenecks) des Projekts, um Engpässe zu vermeiden.
- **Beispiel:** Ein Event-Planungsprojekt, bei dem die Buchung der Veranstaltungslocation als Hauptengpass betrachtet wird, die Zusammenarbeit mit einem Spezialisten, der sehr angefragt ist.

Projektmanagementansätze & Vorgehensmodelle

Timeboxed/Timeboxing:

- feste Vorgabe des Zeitrahmens für ein Projekt. Bis zu diesem fixen Endtermin soll so viel Leistung, wie möglich, erbracht werden.
- **Beispiel:** Konzert mit festem Veranstaltungstermin. Aber auch der Sprint im Scrum Framework.

Evolutionär:

- Fortlaufende Weiterentwicklung basierend auf sich entwickelnden Anforderungen & Feedback (Produkt in neuen Versionen/Updates)
- **Beispiel:** Betriebssystementwicklung - Ein neues Betriebssystem wird mit grundlegenden Funktionen veröffentlicht und erhält dann regelmäßig Updates und Upgrades, um es zu erweitern und zu verbessern.

Projektmanagementansätze & Vorgehensmodelle

Agil:

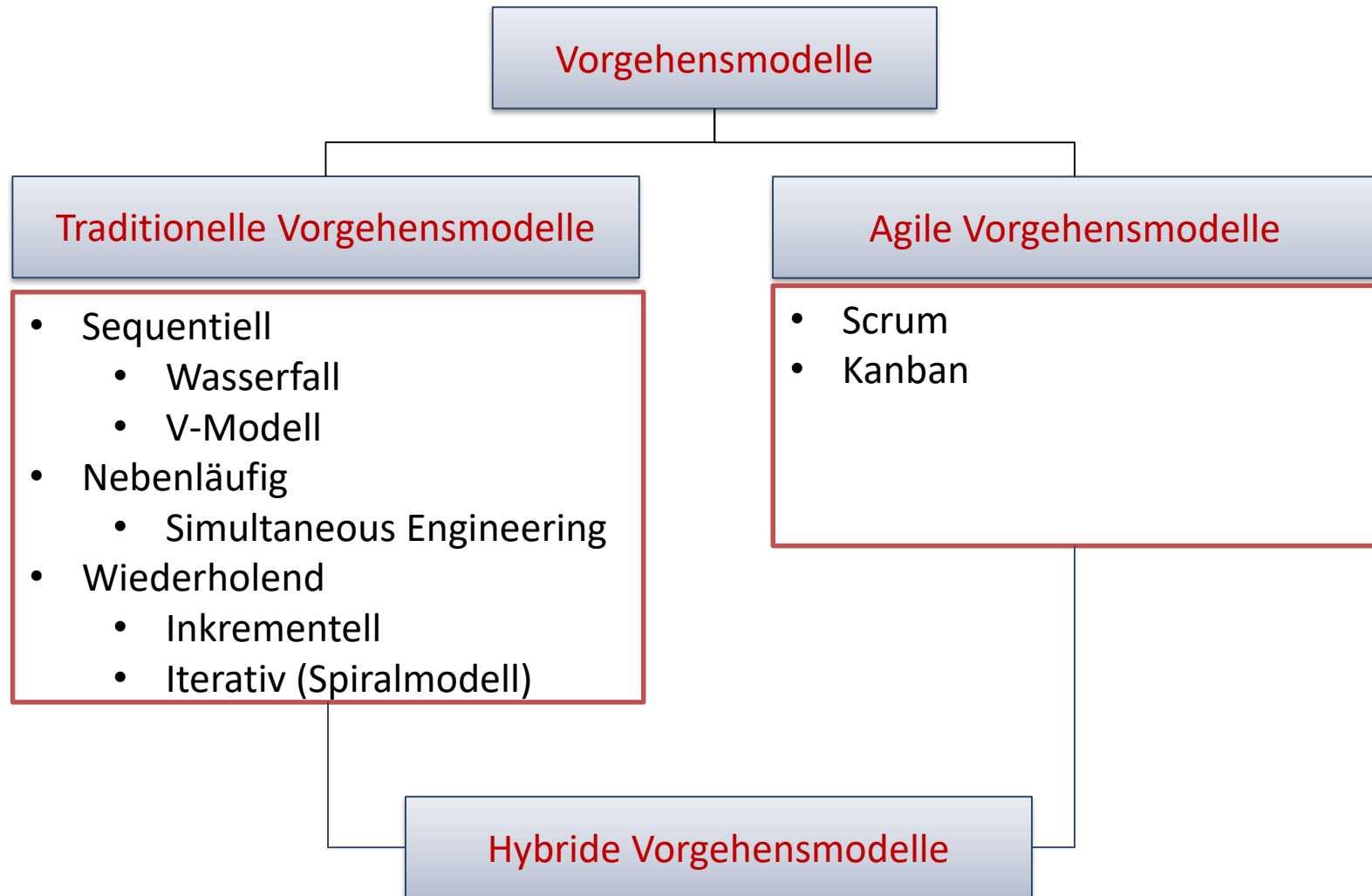
- Flexible Herangehensweise mit Fokus auf Zusammenarbeit, Anpassung und rasche Ergebnisse.
- **Beispiel:** Die Entwicklung einer mobilen App mit agilen Methoden wie Scrum oder Kanban.

Hybrid:

- Kombination verschiedener Ansätze, um die Vorteile mehrerer Methoden zu nutzen.
- **Beispiel:** Ein IT-Projekt, das Elemente des agilen und sequenziellen Ansatzes kombiniert.

Der Projektmanagementansatz beeinflusst die Wahl des geeigneten Vorgehensmodells

Vorgehensmodelle



Vorgehensmodelle

- Vorgehensmodelle geben eine **konkrete Methodik und Schritte** vor, wie das Projekt abgewickelt wird. (praktische Anleitung, „Wie gehen wir vor?“)
- Standardvorgehensmodelle reduzieren den Planungsaufwand.
- Sie sind **zeitsparend, bewährt (erprobt)** und **anpassbar** an Projektanforderungen.

Wenn es kein Standardvorgehensmodell gibt:

- Wasserfall-Modell für den ersten Überblick nutzen.
- Grobplanung (z.B. der Phasen, Hauptaktivitäten, Meilensteine) mittels Schätzungen / Brainstorming im Team durchführen.
- Ggf. Erfahrungen aus abgeschlossenen Projekten nutzen und auf das aktuelle Projekt übertragen.

Traditionelle Vorgehensmodelle

Sequentiell:

- linearer Ansatz, bei dem Phasen nacheinander abgearbeitet werden.
- **Beispiel:** Konstruktion eines Gebäudes - Die Bauphasen erfolgen in einer festgelegten Reihenfolge: Grundlagen legen, Rohbau, Installationen, Innenausbau, Endkontrolle.

Nebenläufig:

- Phasen können sich überlappen, um Zeit zu sparen.
- **Beispiel:** Softwareentwicklung: Design, Programmierung und die Qualitätssicherung laufen parallel in getrennten Teams

Traditionelle Vorgehensmodelle

Inkrementell:

- Schrittweise Entwicklung, wobei Teile des Projekts sukzessive hinzugefügt werden.
- **Beispiel:** Die Entwicklung einer E-Commerce-Website, bei der nach und nach neue Produktkategorien hinzugefügt werden.

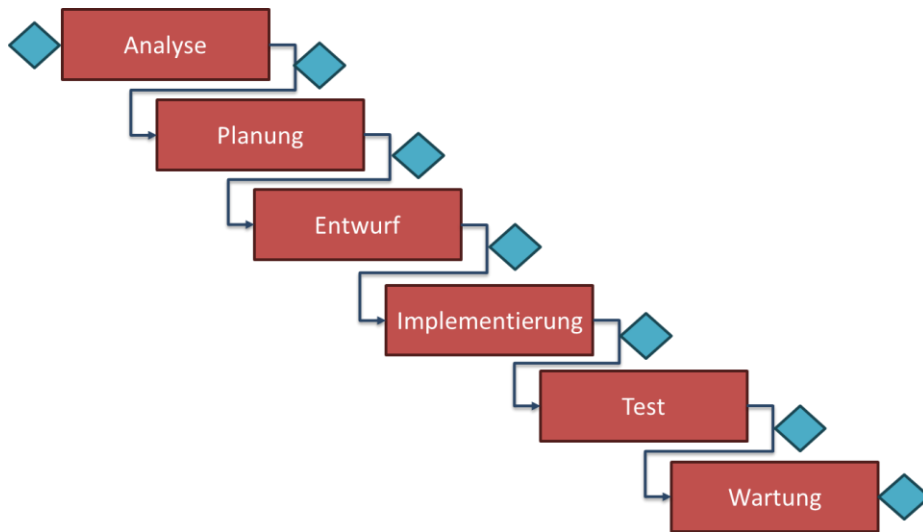
Iterativ:

- Wiederholung von Phasen zur schrittweisen Verbesserung/Verfeinerung.
- **Beispiel:** Die Entwicklung einer Software, bei der die Funktionalitäten in mehreren Iterationen verfeinert werden.

Sequentielle Vorgehensmodelle

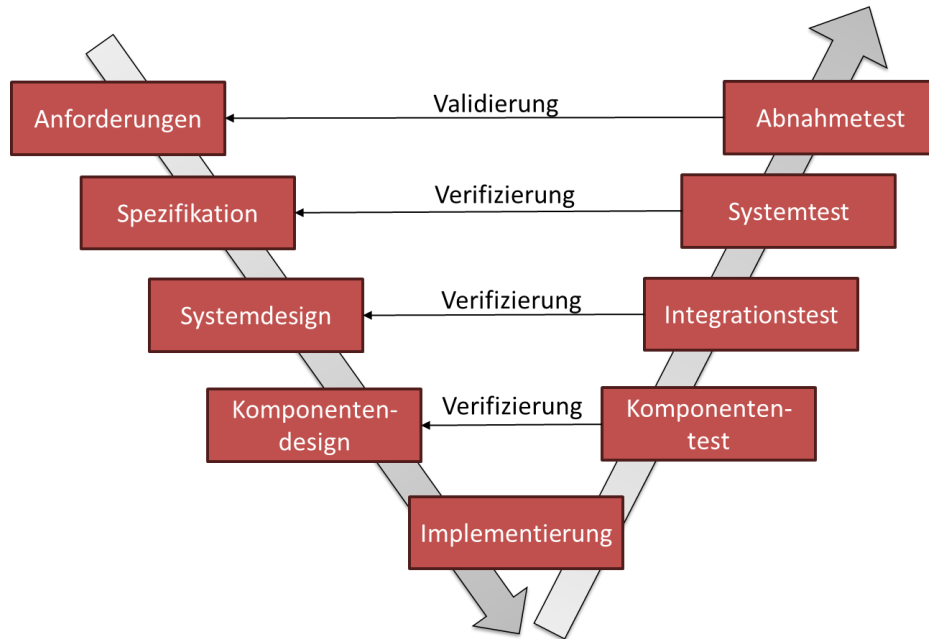
- Sequentielle Vorgehensmodelle basieren auf einer strikt linearen Abfolge von Phasen (linearer Phasenverlauf)
- Eine Phase beginnt erst wenn die vorherige abgeschlossen ist.
- Beim Wechsel von einer abgeschlossenen Phase in die nächste wird ein Meilenstein durchlaufen.
- Ein Rücksprung auf den vorherigen Projektabschnitt ist nicht möglich.
- Klare Übergänge zwischen den Phasen führen zu einer strukturierten, geordneten Projektbearbeitung
- Zu den sequentiellen Vorgehensmodellen gehören:
 - Wasserfallmodell
 - V-Modell

Sequentielle Vorgehensmodelle: Wasserfallmodell



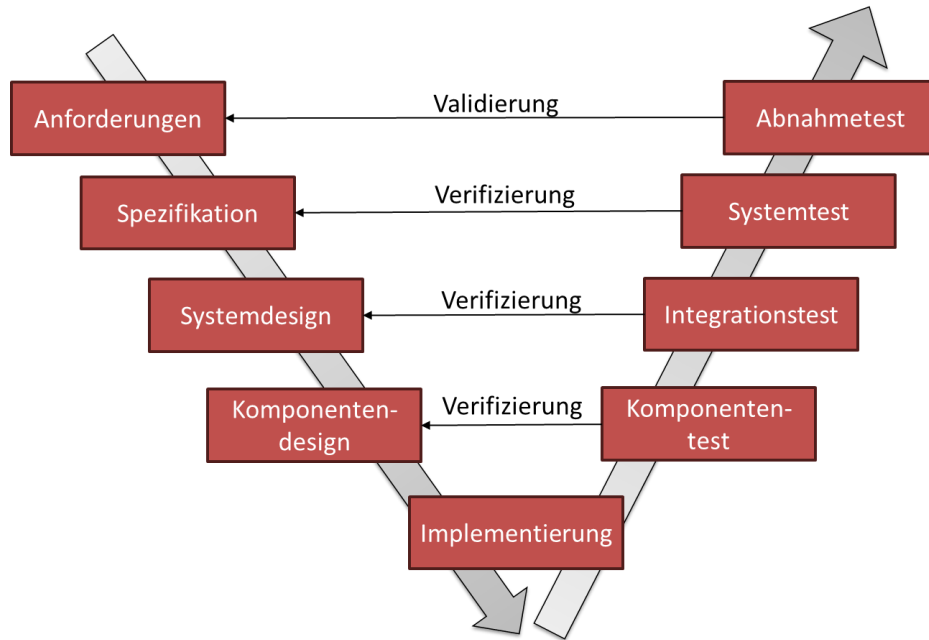
- Projektphasen werden sequentiell durchlaufen.
- Eine Phase muss abgeschlossen sein, bevor die nächste beginnen kann
- Jede Phase beginnt und endet mit einem Meilenstein.
- Am häufigsten eingesetztes Modell.
- Eignet sich gut für Projekte mit stabilen Anforderungen und klaren Projektzielen.
- **Vorteile:**
 - Klare Strukturen und Abläufe
- **Nachteile:**
 - Recht starr und unflexibel

Sequentielle Vorgehensmodelle: V-Modell / V-Modell-XT



- Hat seinen Ursprung in der Softwareentwicklung
- Fokus auf Qualitätssicherung
- **Linker Ast:** erst werden die Anforderungen vom Grobentwurf zum Feinentwurf immer detaillierter spezifiziert
- **Rechter Ast:** Nach der Implementierung wird jeder Entwicklungsschritt mit einem zugehörigen Test verknüpft

Sequentielle Vorgehensmodelle: V-Modell / V-Modell-XT



Verifizierung:

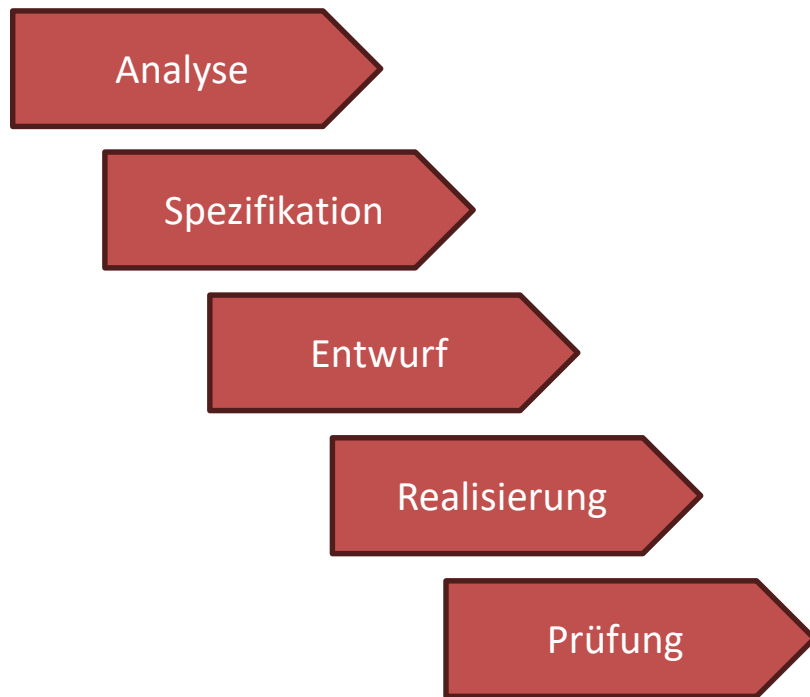
- Nachweis, ob Implementierung korrekt ist
- Prüft, ob technische Anforderungen erfüllt sind
- Fokussiert auf die Konformität mit den Spezifikationen
- Antwortet auf die Frage: "Ist es richtig entwickelt?"

Validierung:

- Nachweis, ob Kundenwünsche erfüllt sind
- Prüft, ob Produkt den beabsichtigten Zweck erfüllt
- Fokussiert auf Kundenzufriedenheit und Bedürfnisse
- Antwortet auf die Frage: "Ist das Richtige entwickelt worden?"

Nebenläufige Vorgehensmodelle

Beispiel: Simultaneous Engineering



- Bei den nebenläufigen Vorgehensmodellen können Projektphasen überlappend oder parallel abgearbeitet werden
- D.h. eine Folgephase darf beginnen, bevor die vorherige abgeschlossen ist.
- Ziel ist es, die Projektdauer zu verkürzen
- Bekanntestes Modell ist das **Simultaneous Engineering**: hier arbeiten verschiedene Teams parallel an unterschiedlichen Teilen des Projekts.
- Erfordert gute Koordination und Kommunikation

Wiederholende Vorgehensmodelle

iterativ vs. inkrementell

Wiederholende Modelle betonen die wiederholte Durchführung von Prozessen. Das Projekt wird in Inkremente (kleinere Einheiten/Teilergebnisse) oder Iterationen (Verbesserungen) unterteilt, die schrittweise abgearbeitet werden.

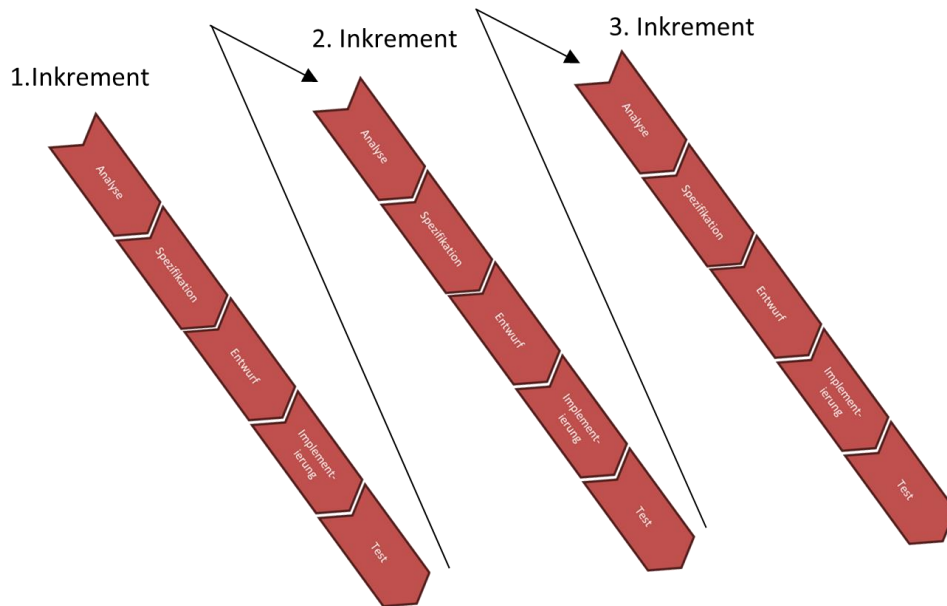
Iterativ



inkrementell



Wiederholende Vorgehensmodelle: Iterativ



- Beim inkrementellen Modell wird das Projekt in überschaubare Inkremente unterteilt.
- Die Schritte innerhalb eines Inkrements werden als Wasserfall getätigt.
- Nach jeder Wiederholung ist ein Inkrement entstanden.

Wiederholende Vorgehensmodelle: Iterativ - Spiralmodell

- Beim iterativen Modell wird das Produkt als Ganzes in wiederholenden Iterationen schrittweise verfeinert/verbessert
- man nähert sich schrittweise an eine endgültige Lösung an
- Die Iterationen werden dazu genutzt, Anforderungen zu konkretisieren und von Iteration zu Iteration die Ziele zu verfeinern.
- Am einfachsten kann man es sich als Spirale vorstellen (Spiralmodell)



Agile Methoden

- Agile Vorgehensmodelle betonen Flexibilität und enge Zusammenarbeit mit den Kunden.
- Sie sind auf schnelle Anpassungen an sich ändernde Anforderungen ausgelegt.
- Einen Phasenplan gibt es nicht. Eine Projektleitung gibt es nicht.
- Agile Methoden beruhen auf den agilen Werten und Prinzipien (Agiles Manifest, 2001)

Die 4 Werte des Agilen Manifests

INDIVIDUEN UND INTERAKTIONEN	sind wichtiger als	PROZESSE UND WERKZEUGE
FUNKTIONIERENDE SOFTWARE	ist wichtiger als	UMFASSENDE DOKUMENTATION
ZUSAMMENARBEIT MIT DEM KUNDEN	ist wichtiger als	VERTRAGS- VERHANDLUNGEN
REAKTION AUF VERÄNDERUNG	ist wichtiger als	DAS BEFOLGEN EINES PLANS

Agiles Projektmanagement

- Agiles Projektmanagement ist ein iterativer Ansatz zur Planung und Steuerung von Projekten.
- In kurzen Iterationszyklen werden regelmäßig Ergebnisse (Inkremente) fertiggestellt, Feedback eingeholt und das weitere Vorgehen entsprechend angepasst.

Typische Merkmale von agilen Projekten:

- Sie werden iterativ in kurzen Zyklen abgewickelt.
 - Jede Iteration liefert ein fertiges Inkrement.
 - Der Fokus liegt auf der Orientierung an Kundenwünschen und dem Erschaffen von Werten.
 - Regelmäßiges Feedback und Kommunikation spielen eine große Rolle.
 - Es wird schnell und proaktiv auf geänderte Anforderungen reagiert.
 - Die Teams arbeiten selbstorganisiert (nach dem Pull-Prinzip).
- Agile Methoden sind z.B. Scrum, Kanban und Extreme Programming

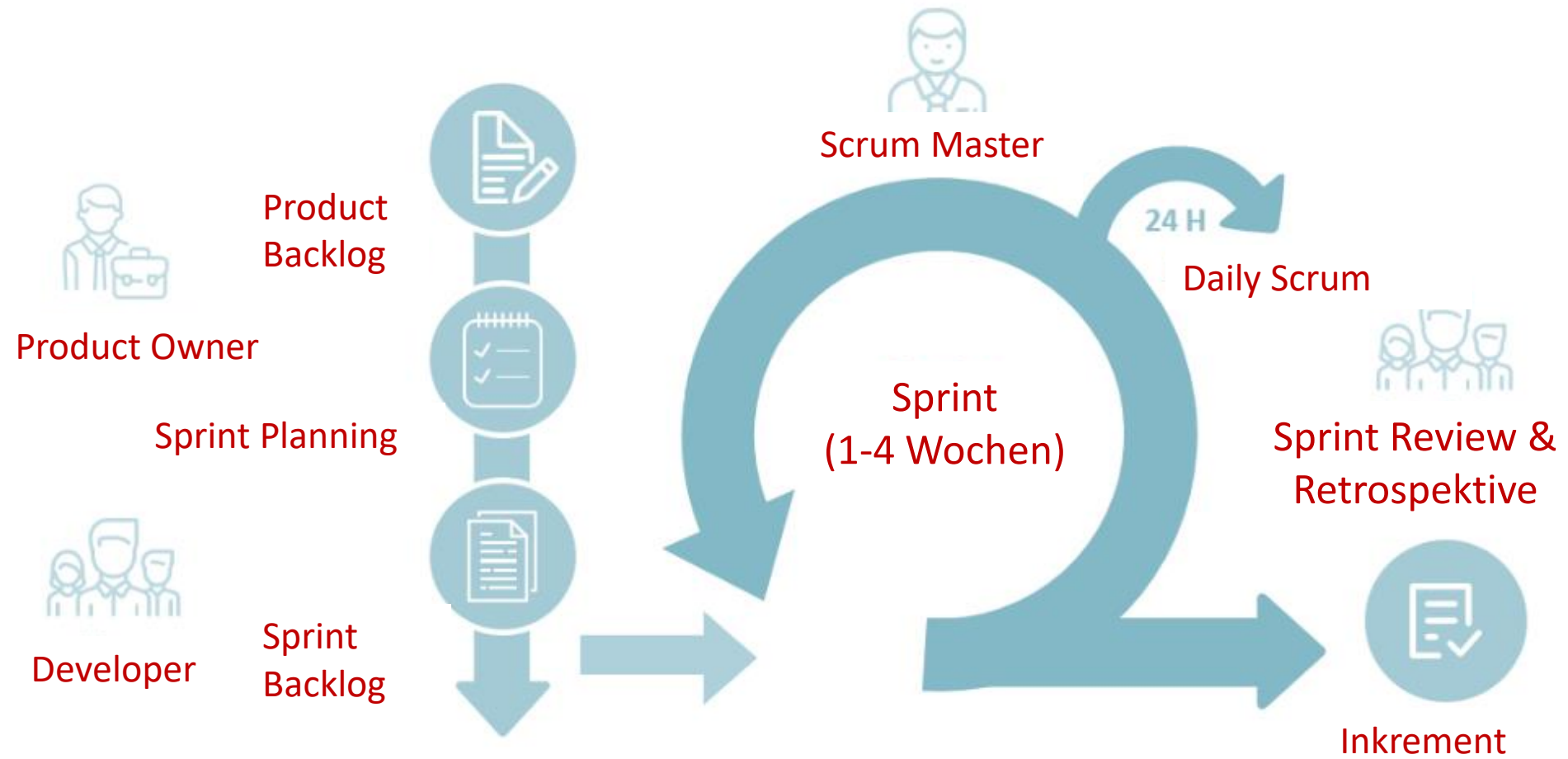
Agiles Projektmanagement – Voraussetzungen /Erfolgsfaktoren

- Agile Werte & Prinzipien werden akzeptiert und praktisch gelebt.
- Das Team arbeitet selbstorganisiert
- Einbinden des Kunden in das Projekt (enge Zusammenarbeit)
- Häufige Feedbackschleifen
- Flexibler Umgang mit Änderungen
- Regeln agiler Frameworks werden konsequent eingehalten.
(Einhalten von Timeboxing, befolgen der Prozesse)
- Alle Beteiligten kennen ihre Rollen und Befugnisse.
- Führungskräfte sehen sich mehr als Unterstützer und Coach statt als „Entscheider von oben“.
- Möglichst viele Beteiligte verfügen über ein agiles Mindset.
- Es stehen geeignete Räume mit Visualisierungsmöglichkeiten (z. B. Kanban-Boards) zur Verfügung.



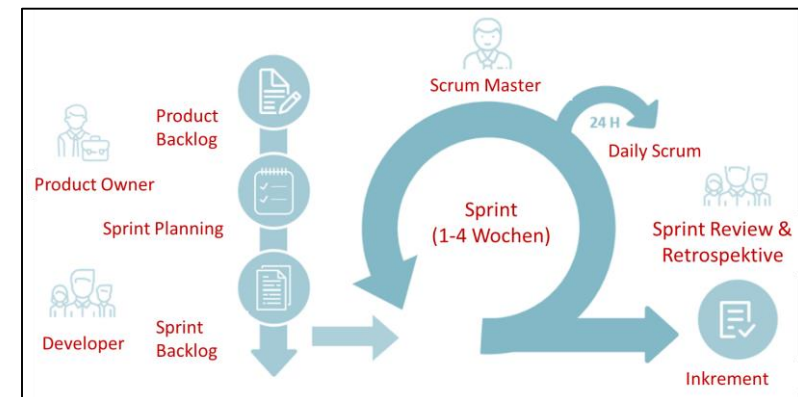
Scrum

Scrum Framework



Scrum

- Scrum ist ein agiles Framework, das in kurzen Entwicklungszyklen (Iterationen) sogenannten Sprints arbeitet.
- Der Name Scrum stammt aus dem Rugby-Sport (engl. Gedränge)
- In jedem Sprint werden aus dem Product Backlog ausgewählte Anforderungen umgesetzt
- Jeder Sprint liefert ein fertiges Inkrement
- Scrum betont Selbstorganisation und kontinuierliches Feedback zur Verbesserung
- Das Team arbeitet selbstgesteuert, nach dem Pull-Prinzip
- Seinen Ursprung hat Scrum in der Softwareentwicklung
- Heute wird es auch für andere Projekte eingesetzt, eignet sich aber nicht für jede Projektart



Scrum

Rollen, Artefakte und Ereignisse

Rollen	Artefakte	Ereignisse
<ul style="list-style-type: none">• Product Owner• Scrum Master• Developer	<ul style="list-style-type: none">• (Product) Backlog• Sprint Backlog• Inkrement• DoR - Definition of Ready• DoD - Definition of Done	<ul style="list-style-type: none">• Sprint Planning• Sprint• Daily Scrum• Sprint Review• Sprint Retrospektive

Scrum

Rollen - Scrum Team

- Das **Scrum Team** (5-9 Mitglieder) besteht aus drei Rollen, eine Projektleitung ist nicht vorgesehen

Product Owner:

- Repräsentiert die Kundenseite.
- Vertritt die kaufmännischen Interessen des Projekts.
- Verantwortlich für das Product Backlog.
- Priorisiert Anforderungen und gibt Freigaben für die Auslieferung.

Scrum Master:

- Coach, Mentor für das Team und gewährleistet die Anwendung von Scrum-Prinzipien (verantwortlich für den Scrum-Prozess)
- Beseitigt Hindernisse, fördert Selbstorganisation und kontinuierliche Verbesserung.

Developer:

- Cross-funktionales Team, das das Produkt entwickelt und liefert.
- Organisiert sich selbst, entscheidet über Arbeitsweise und Aufgabenverteilung (Pull-Prinzip)

Scrum – Rollen

Artefakte

Product Backlog:

- Gesamte Liste von Anforderungen und Funktionalitäten.
- Priorisiert nach Wert und Notwendigkeit.
- Product Backlog Items sind in Form von User Stories formuliert.
 - „Als (ROLLE) möchte ich (FUNKTION), damit (NUTZEN)“
- Quelle der Arbeiten, die durch das Scrum Team erledigt wird.

Sprint Backlog:

- Ausgewählte Anforderungen aus dem Product Backlog für den aktuellen Sprint.
- Detaillierte Aufgaben (Tasks) für die Umsetzung.

Inkrement:

- Das Arbeitsergebnis eines abgeschlossenen Sprints.
- Potenziell auslieferbares Teilergebnis (vorzeigbar und lauffähig)

Scrum – Rollen

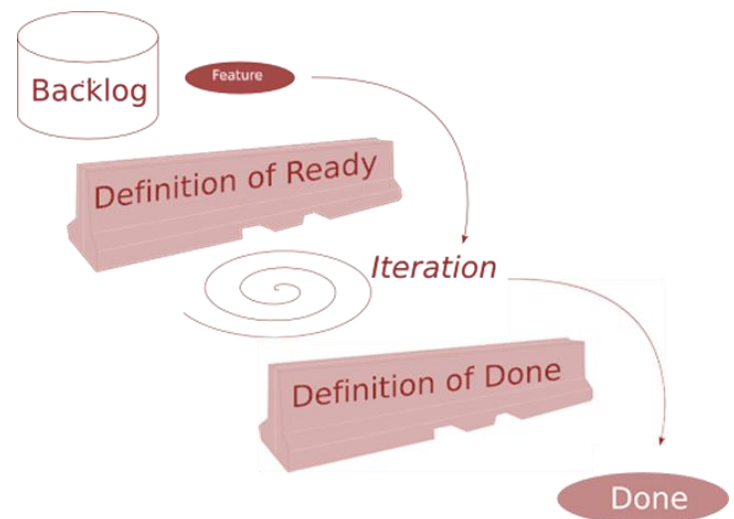
Artefakte

Definition of Ready (DoR):

- Die Definition of Ready beschreibt, wann eine Aufgabe bereit ist, begonnen zu werden.

Defintion of Done (DoD):

- Die Definition of Done beschreibt, unter welchen Bedingungen eine Aufgabe wirklich fertig ist.



Scrum – Rollen

Ereignisse

Sprint:

- Iterative Entwicklungsperiode von konstanter Länge (1-4 Wochen).
- Die Sprintdauer ist festgelegt und darf nicht überschritten werden (Timebox)
- Ziel: Fertigstellung eines auslieferbaren Inkrements.

Sprint Planning:

- Besprechung, in der das Team den Umfang und Inhalt des nächsten Sprints festlegt.
- **Teil 1:** Auswahl von Aufgaben aus dem Product Backlog in das Sprint Backlog („WAS“ wird gemacht?)
- **Teil 2:** Festlegen „WIE“ erfolgt die Umsetzung der Tasks des Sprint Backlogs
- Teilnehmer: Scrum Team (Teil 1), Developer & Scrum Master (Teil 2)

Scrum – Rollen

Ereignisse

Daily Scrum:

- Kurzes tägliches Meeting, um die Arbeit zu synchronisieren (maximal 15 Minuten, oft als Stand-Up-Meeting)
- Austausch über Fortschritt, Herausforderungen und Planung.
- Teilnehmer: Scrum Master, Developer
- Jeder Developer hat drei Fragen zu beantworten:
 - „Was wurde seit gestern erreicht?“
 - „Was ist für heute geplant?“
 - „Welche Hindernisse gibt es bzw. welche Unterstützung wird gebraucht?“

Scrum – Rollen

Ereignisse

Sprint Review:

- Treffen am Ende des Sprints, um das abgeschlossene Inkrement zu präsentieren.
- Feedback vom Product Owner und ggf. Kunden/Stakeholdern

Sprint Retrospektive:

- Reflexion am Ende des Sprints über
 - den Prozess und
 - die Zusammenarbeit
- Identifikation von Verbesserungsmöglichkeiten.
- Teilnehmer: Scrum Team

Scrum - Timeboxing

- Timeboxing bedeutet das strikte Einhalten vorgegebener Termine bzw. Zeiten.
- Bevor die festgelegte Dauer (Timebox) überschritten wird, wird der Inhalt gekürzt.
- Es ist eine grundsätzliche Technik der agilen Methoden.
- Das Prinzip findet Anwendung bei allen Scrum Ereignissen.

Sprint	4 Wochen	3 Wochen	2 Wochen	1 Woche
Sprint Planning	8 Stunden	6 Stunden	4 Stunden	2 Stunden
Daily Scrum	15 Minuten	15 Minuten	15 Minuten	15 Minuten
Review	4 Stunden	3 Stunden	2 Stunden	1 Stunde
Retrospektive	3 Stunden	135 Minuten	90 Minuten	45 Minuten

User Story

- Eine User Story ist eine kurze, prägnante Beschreibung einer Anforderung aus Sicht des Benutzers.
- Sie folgt einem bestimmten Aufbau:
"Als [Rolle] möchte ich [Ziel, Funktion], damit [Nutzen]"

Beispiele:

1. Als Benutzer möchte ich mich mit meinem Benutzernamen und Passwort anmelden, damit ich auf meine persönlichen Einstellungen zugreifen kann.
2. Als Kunde möchte ich Artikel in meinen Warenkorb legen können, um später alle ausgewählten Produkte auf einmal kaufen zu können.
3. Als Reisender möchte ich die Verfügbarkeit von Flügen an meinem bevorzugten Reisedatum überprüfen, damit ich meine Reise entsprechend planen kann.
4. Als Moderator möchte ich Beiträge von Benutzern löschen können, um unangemessene Inhalte zu entfernen und die Community sicher zu halten.

Starfish Retrospektive



Retrospektive (= Rückblick):

- moderiertes Meeting,
- um die Zusammenarbeit zu reflektieren
- und mögliche Verbesserungsmaßnahmen für die Zukunft abzuleiten (und umzusetzen!)
- Ohne Schuldzuweisungen („no fingerpointing“, „no shaming, no blaming“), sondern konstruktiv, lösungsorientiert und wertschätzend
 - „Lasst uns zurückblicken und aus den positiven und negativen Erkenntnissen lernen, damit wir uns ständig verbessern.“
- Eine einfache Methode zur Strukturierung der Retrospektive ist die **Starfish (= Seestern) Retrospektive**

Starfish Retrospektive



MORE OF

Was sollten wir mehr tun, weil es sich als nützlich erweist?

KEEP DOING

Was sollten wir beibehalten?



START DOING

Was sollten wir Neues ausprobieren?
Was sind die Wünsche?

LESS OF

Was sollten wir reduzieren, weil es nicht nützlich und hilfreich ist?

Womit sollten wir aufhören?

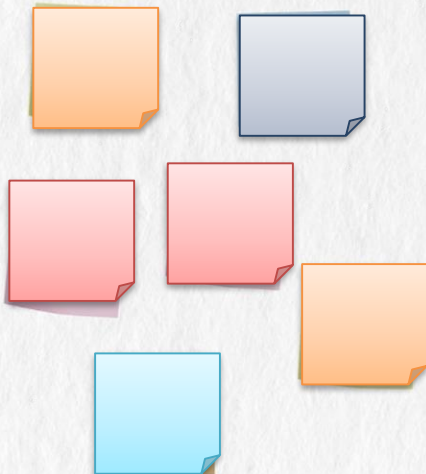
STOP DOING



Alternative: START-STOP-CONTINUE-Retrospektive

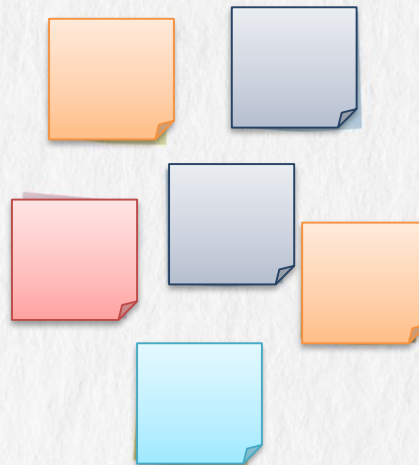
START

WOMIT SOLLTEN WIR
BEGINNEN?



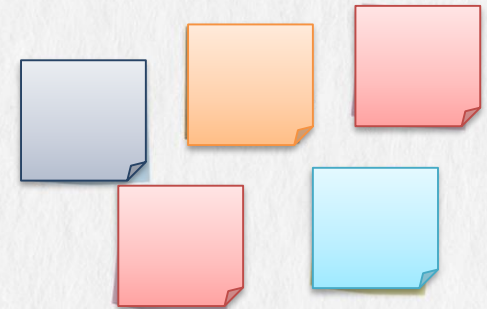
STOP

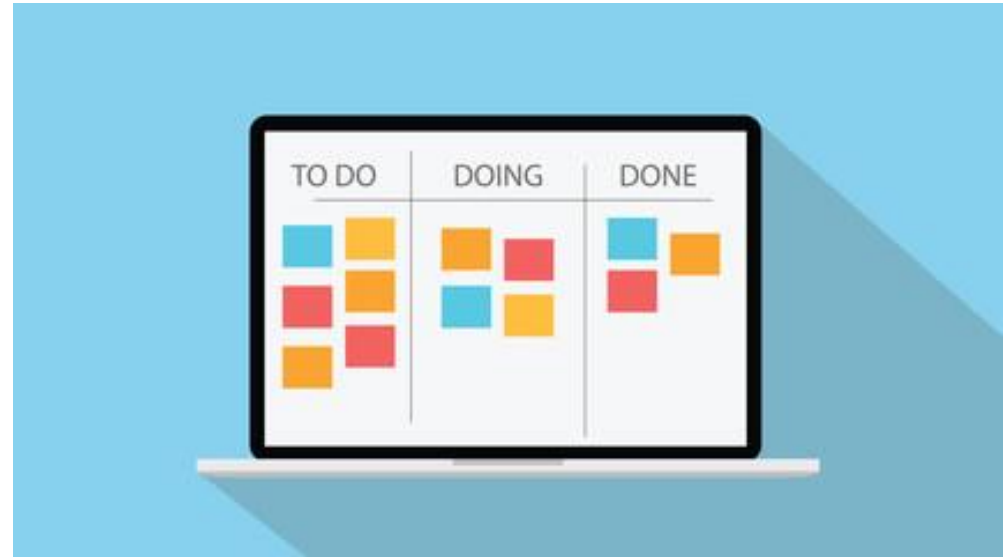
WOMIT SOLLTEN WIR
AUFHÖREN?



CONTINUE

WOMIT SOLLTEN WIR
WEITERMACHEN?





Kanban

Kanban

- Kanban jap. für „Karte, Signalkarte“
- Ursprünglich 1947 von Taiichi Ohno im Toyota-Produktionssystem entwickelt.
- Nicht denselben Ursprung wie die meisten agilen Methoden, aber häufig damit verbunden.
- Schreibt keine festen Prozesse, Strukturen oder Rollen vor.

Fördert Selbstorganisation durch das
Pull-Prinzip:

- Teammitglieder ziehen Aufgaben vom Kanban-Board anstatt sie zugewiesen zu bekommen.
- Der Kern für die Umsetzung von Kanban ist das **Kanban-Board**, mit mindestens drei Spalten (z.B. „to do“, „doing“, „done“).
- **Das Kanban-Board visualisiert den Arbeitsfluss.**



Kanban

Vier Grundprinzipien von Kanban:

- Beginne dort, wo du dich im Moment befindest.
- Strebe nach inkrementellen, evolutionären Veränderungen.
- Respektiere den bestehenden Prozess, Rollen und Verantwortlichkeiten.
- Fördere Führungsverhalten auf allen Ebenen in der Organisation.

Sechs Kerneigenschaften von Kanban:

- Visualisiere den Workflow. (Kanban-Board)
- Begrenze die Menge der begonnenen Arbeit (Work in Progress – WIP, kein schädliches Multitasking)
- Messe und kontrolliere den Fluss des Workflows. (Vorlaufzeit, Durchlaufzeit, Durchsatz.)
- Mache die Prozessregeln explizit. (Mach Regeln für alle öffentlich)
- Verwende Modelle zur Identifizierung von Verbesserungschancen.
- Etabliere Feedbackschleifen (z.B. Daily-Stand-Ups, Review-Meetings).

