

Projektmanagement



Was der Kunde wollte



Was der Projektleiter verstand



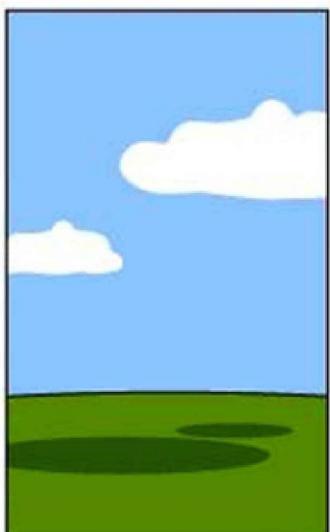
Wie es der Analytiker entwarf



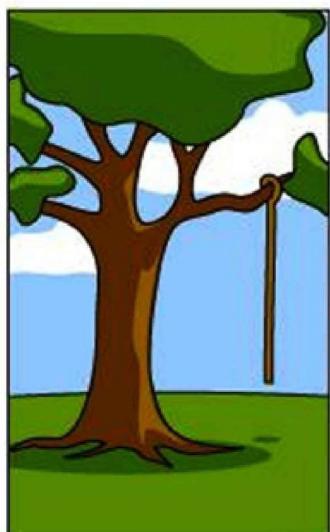
Was der Programmierer schrieb



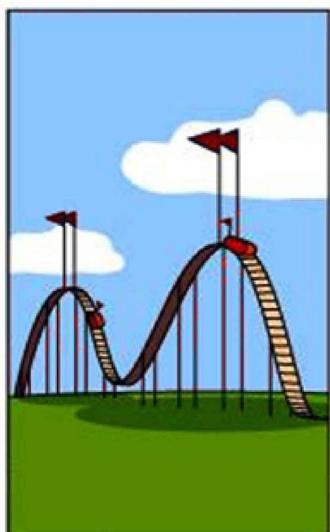
Wie es der Berater definierte



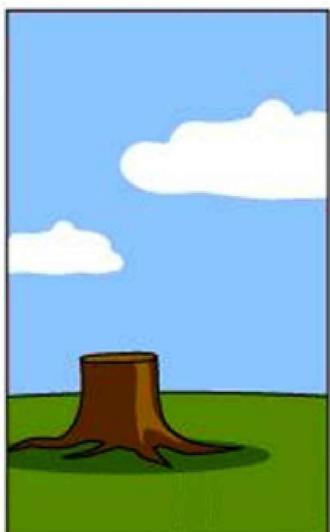
Wie das Projekt dokumentiert wurde



Was installiert wurde



Was dem Kunden in Rechnung gestellt wurde



Wie der After Sale Support aussah



Was der Kunde wirklich gebraucht hätte

Frühere Entwicklung

- „Projektplanung gibt es, seit Menschen größere Vorhaben gemeinschaftlich durchführen. Weder ein militärischer Feldzug, noch die Errichtung großer Gebäude (Tempel, Festungen), noch beispielsweise eine lange Seereise zur Entdeckung der Westpassage nach Indien sind vorstellbar, ohne dass die Verantwortlichen diese Projekte detailliert geplant hätten. Doch geschah dies lange Zeit formlos, allein aufgrund der Erfahrungen und Kenntnisse der Verantwortlichen; erst im 20. Jahrhundert sollten diese informellen Verfahren zusammengetragen, systematisiert und in die wissenschaftlich aufbereitete Form gebracht werden, unter der heute Projektmanagement betrieben wird.“
- Henry Gantt entwickelte 1910 den Balkenplan (auch **Gantt-Diagramm** genannt). Gants Methode kam erstmals bei einem größeren Bauvorhaben, der Errichtung des 1935 fertiggestellten Hoover Dam, zum Einsatz. Die erste Dokumentation der Vorgehensweise beim Projektmanagement wurde vermutlich im Rahmen des Manhattan-Projekts vorgenommen.
- Eine weitere Entwicklung des Projektmanagements war dann für den Wettlauf ins All erforderlich – vor allem für das Apollo-Programm

Geschichte des Projektmanagement

1930

Walter Shewhart entwarf *Projekte mit kurzen Zyklen* als Möglichkeit, die Qualität zu verbessern. Seine Methode ist als PDSA (Plan-Do-Study-Act) bekannt.

1950er/frühe 1960er

Das Militär setzte IID (*Iterative & Incremental Development*) erfolgreich bei der Entwicklung des Überschalljets X-15 ein. Für das Mercury-Projekt nutzte die NASA ebenfalls IID für die Softwareentwicklung, einschließlich des Einsatzes von Timeboxing, Test first und Top-down-/Stub-Development.

1970er

Dr. Royce veröffentlicht „Managing the Development of Large Software Systems“; er wies darauf hin, dass die *Wasserfallmethode* für sich allein *ineffektiv* ist & dass sie mindestens 2x wiederholt werden muss, um erfolgreich zu sein.

1940er

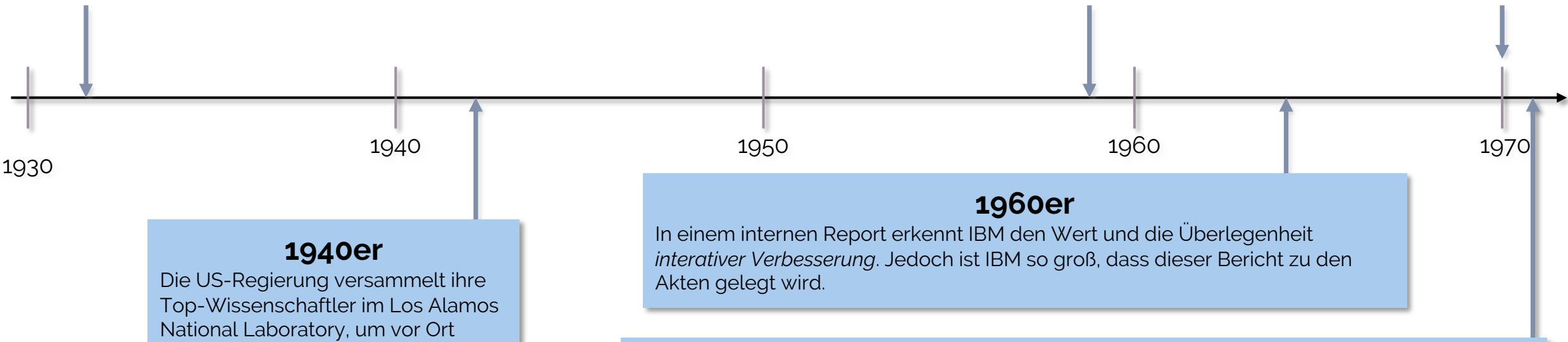
Die US-Regierung versammelt ihre Top-Wissenschaftler im Los Alamos National Laboratory, um vor Ort rasch die *theoretische Atombombe* zu entwickeln.

1960er

In einem internen Report erkennt IBM den Wert und die Überlegenheit *interaktiver Verbesserung*. Jedoch ist IBM so groß, dass dieser Bericht zu den Akten gelegt wird.

Frühe 1970er

Die IBM-Abteilung Federal Systems und die Firma TRW stellen für mehr als 100 Millionen US-Dollar Projekte mittels iterativer Methoden fertig, einschließlich der Befehls- und Steuersoftware für das Raketenabwehrprogramm.



Geschichte des Projektmanagement

1980er

Grady Booch, Guru für objektorientierte Programmierung, *formuliert das Spiralmodell* für die Softwareentwicklung.

1986

Harvard Business Review veröffentlicht „The New Product Development Game“

1980

1990

2000

2010

2020

2001

17 Experten für DSDM, XP, Scrum, FDD und andere Lean-Methoden treffen sich, um die Zukunft der iterativen und inkrementellen Softwareentwicklung (IID) zu diskutieren. Dieses Treffen führte zu dem Begriff *agile Methoden* und zur Gründung der Agile Alliance.

2004

Gründung der Scrum Alliance.

1990er

Jeff Sutherland und Ken Schwaber entwickeln einen auf Zeitfenster mit fester Dauer basierten Ansatz, bei dem es sich um eine Mischung aus japanischer IDD Techniken, die bei Honda eingesetzt werden (sashimi), und Konzepten aus „The New Product Development Game“ handelt. Sie nennen diesen Ansatz **Scrum**.

2010

Das US-Verteidigungsministerium verlangt bei allen IT-Projekten *agile Ansätze* (National Defense Authorization Act, Section 804):



Information

1910: Henry Gantt entwickelte den Balkenplan (auch Gantt-Diagramm genannt), das noch heute seine Anwendung findet



Projektplanung

Kenntnisse über wesentliche Schritte einer Projektplanung

1. Schritt:

Der Projektstrukturplan (PSP)

2. Schritt:

Der Projektlaufplan

3. Schritt:

Der richtige Terminplan

4. Schritt:

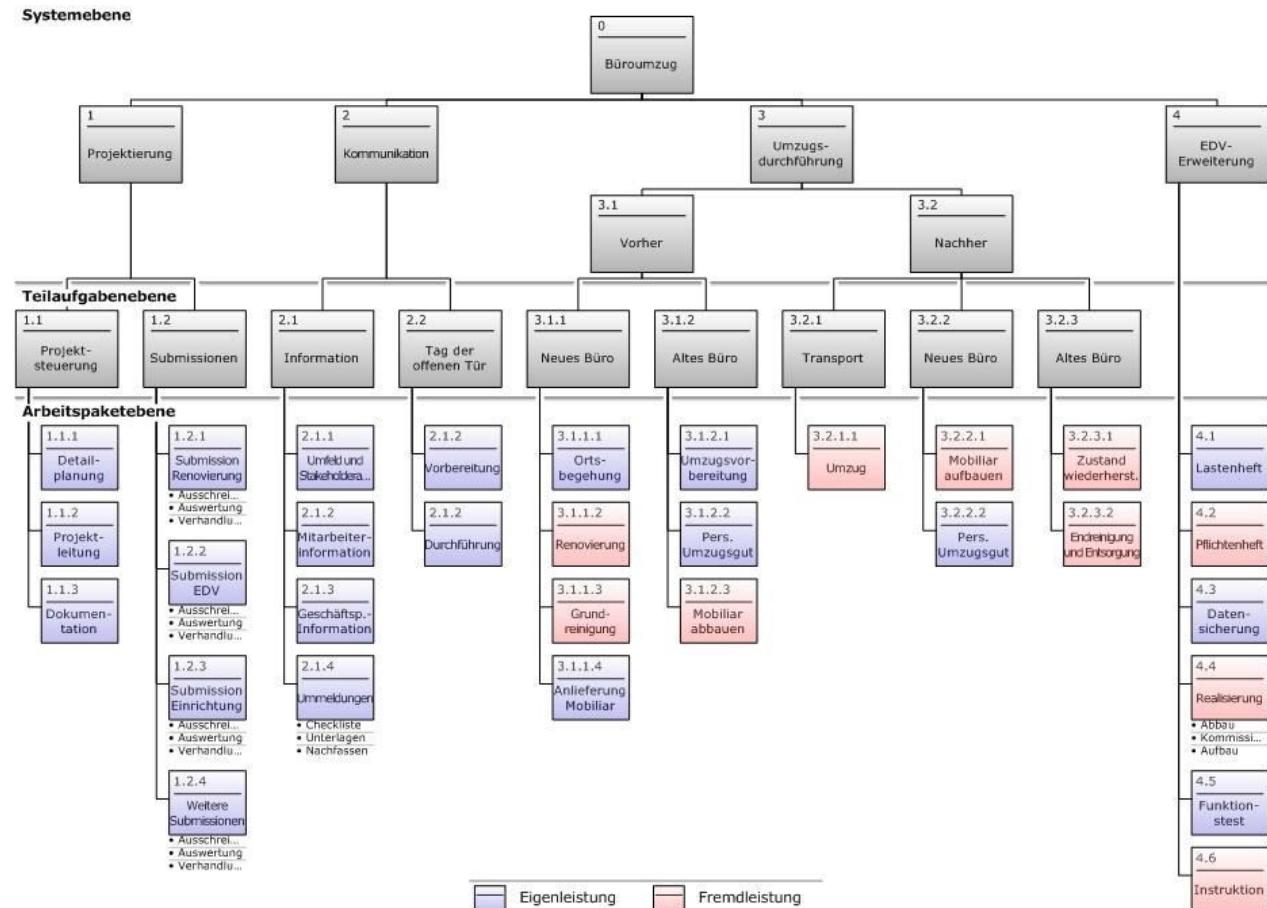
Ressourcenplanung

5. Schritt:

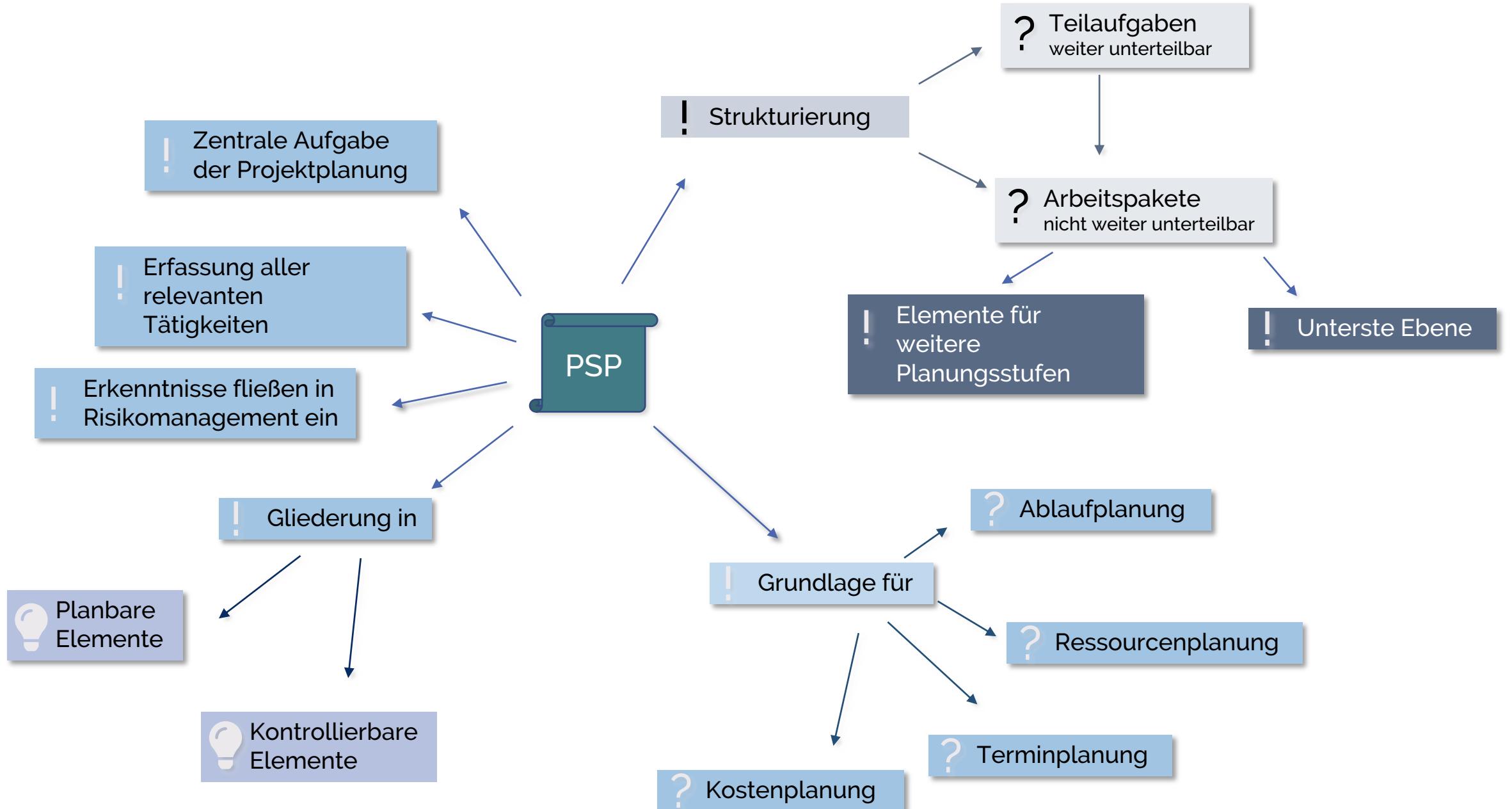
Kosten- und Budgetplanung

Fachbegriff Projektstrukturplan

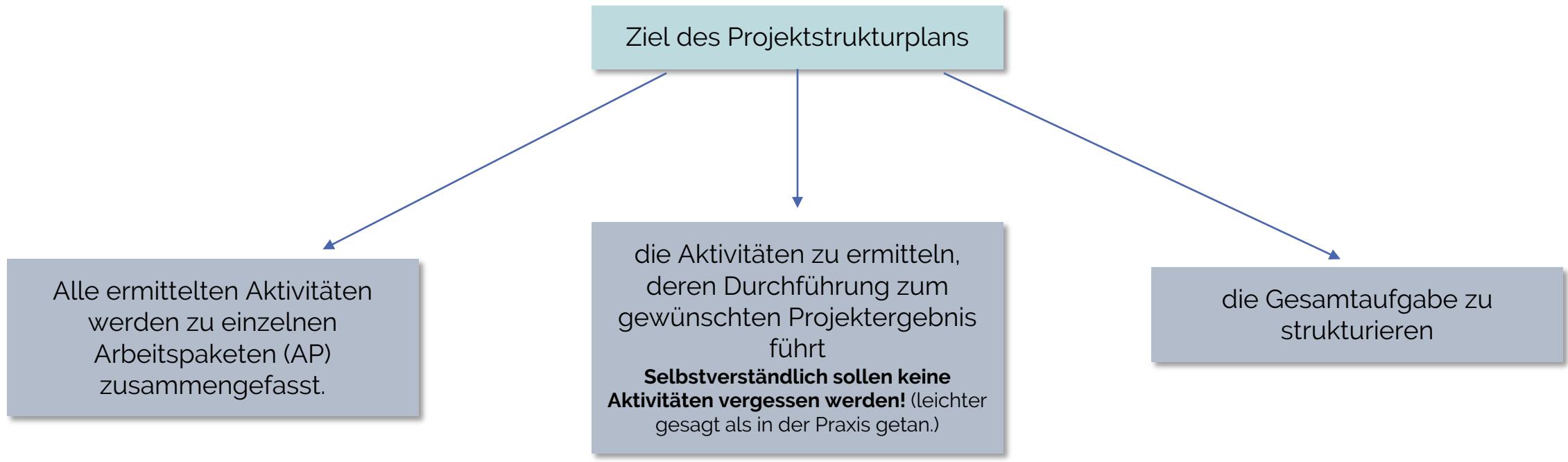
Der Projektstrukturplan ist das Ergebnis einer Gliederung des Projekts in plan- und kontrollierbare Elemente. Ein Projekt wird im Rahmen der Strukturierung in Teilaufgaben und Arbeitspakete unterteilt. Wikipedia



Projektstrukturplan „Plan der Pläne“

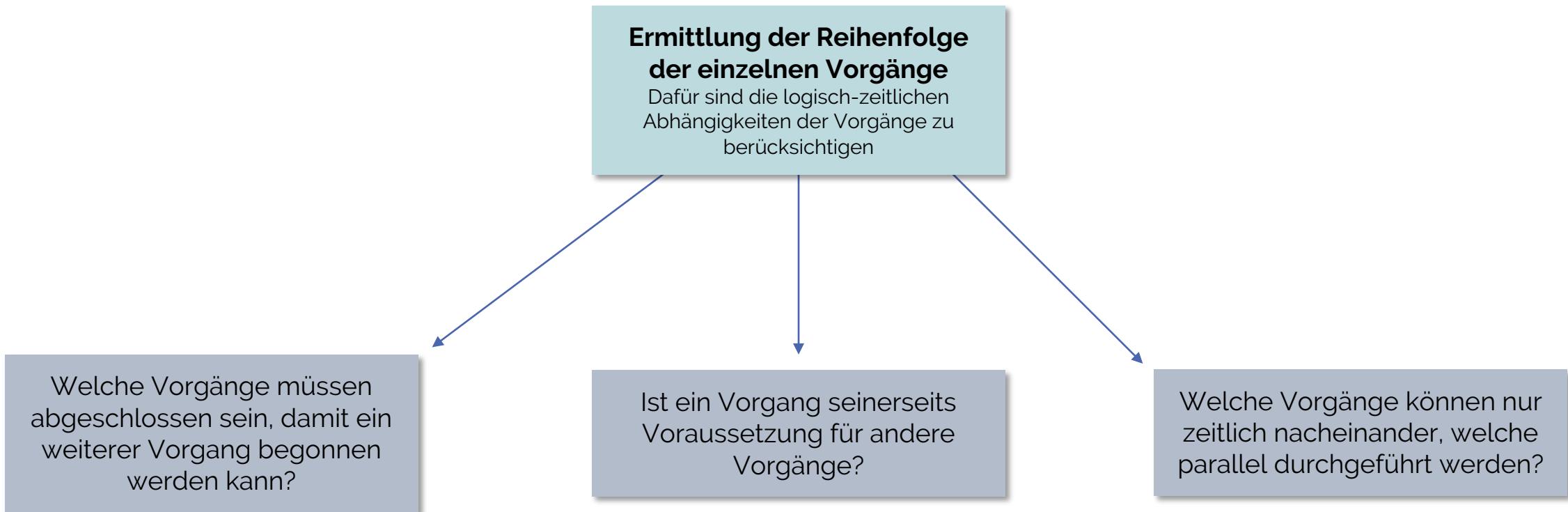


1. Schritt der Projektplanung: Der Projektstrukturplan (PSP)



Bedenke: Bei Großprojekten sollte unbedingt ein PSP erstellt werden. Bei Kleinprojekten kann darauf verzichtet werden und sofort mit dem nächsten Planungsschritt begonnen werden.

2. Schritt der Projektplanung: Der Projektablaufplan



- Beispielsweise können die Arbeiten an der Elektronik und der Mechanik parallel erfolgen. Der Test des Gerätes kann aber erst nach dem Zusammenbau durchgeführt werden. Bei komplexeren Projekten sind diese Abhängigkeiten oft nicht mehr so offensichtlich.
 - *Die Arbeitspakete sind die Grundbausteine des Projektes.*
 - Für jedes Arbeitspaket sind die Tätigkeiten detailliert zu beschreiben.
 - Jedes Arbeitspaket ist einem Verantwortlichen zuzuordnen.
- **Nur so ist in der Durchführung ein effizientes Projekt-Controlling möglich.**

3. Schritt der Projektplanung: Der richtige Terminplan

Für die Terminplanung gibt es 3 unterschiedliche Methoden

die Vorgangsliste

- Die Vorgangsliste erfordert den geringsten Aufwand.
- Sie eignet sich für einfache, überschaubare Projekte.

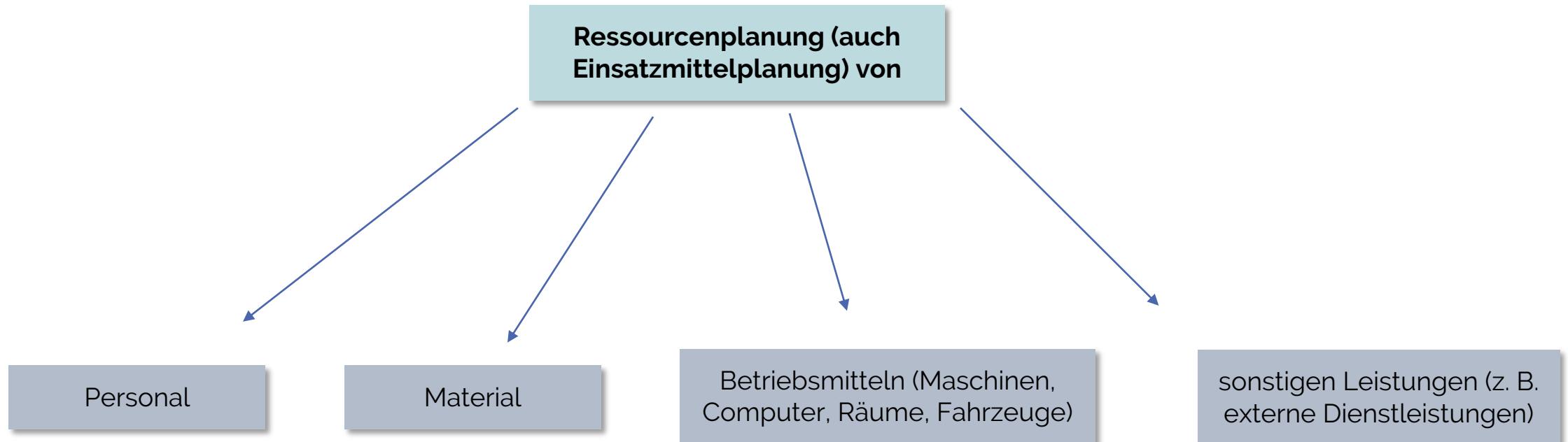
das Balkendiagramm

- Für Projekte mit wenigen Vorgängen reicht das Balkendiagramm für die Terminplanung völlig aus.

die Netzplantechnik

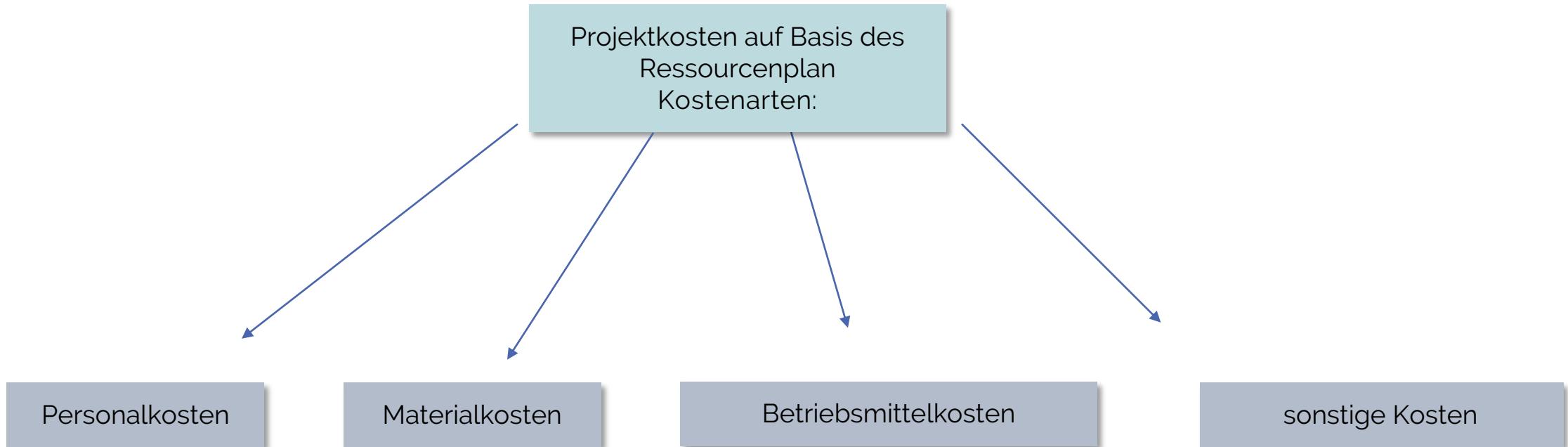
- Für komplexere Projekte mit einer größeren Zahl an Arbeitspaketen (oder Vorgängen) wird die Netzplantechnik eingesetzt.
- Die einzelnen Vorgänge werden dabei als Kästen dargestellt (in der Fachsprache auch als Knoten bezeichnet) und mit den Terminangaben versehen.

4. Schritt der Projektplanung: Ressourcenplanung



- Nicht nur von welcher Ressource wie viel, sondern auch benötigte Kapazitäten von Mitarbeitern, Materialien, Geräten oder Räumen für die geplanten Zeiträume (steht das wirklich zur Verfügung?)
- Achtung:
 - Bedarf an Mitarbeitertagen oder -stunden einer bestimmten Qualifikation (z. B. Entwicklungsingenieure, Techniker, Mechaniker, Sekretariat) ermitteln
 - Urlaubs- und Feiertage, Betriebsferien und andere absehbare Ausfalltage berücksichtigen
 - Mitarbeiter haben oft noch andere Aufgaben und stehen somit nicht mit ihrer gesamten Arbeitszeit für das Projekt zur Verfügung

5. Schritt der Projektplanung: Kosten- und Budgetplanung



- Die Personalkosten sind bei den meisten Projekten der größte Posten. Sie werden üblicherweise anhand von Stunden- oder Tagessätzen errechnet; für Betriebsmittel lassen sich entsprechend Maschinenkostensätze verwenden.
- Für Materialien und Zukaufteile sind meist die Preise bekannt, und für den Bezug interner Leistungen existieren Verrechnungspreise. Mit den Zahlen der Kostenplanung kann das Projektbudget ermittelt werden.

Was ist der Unterschied zwischen Lastenheft und Pflichtenheft?

Lastenheft

- erstellt der **Auftraggeber** um zu verdeutlichen was das Projekt beinhaltet
- Beschreibt genau was der Auftraggeber vom Auftragnehmer erwartet

Pflichtenheft - Die Antwort auf das Lastenheft

- Wird vom **Auftragnehmer** erstellt
- beinhaltet Informationen darüber, wie der Auftragnehmer das Geforderte umsetzen will
- Inhalte sind für beide Seiten rechtlich bindend
- Alle zuvor zwischen Auftraggeber und Auftragnehmer getroffenen Absprachen verlieren in der Regel durch das Pflichtenheft ihre Gültigkeit

Das Lastenheft

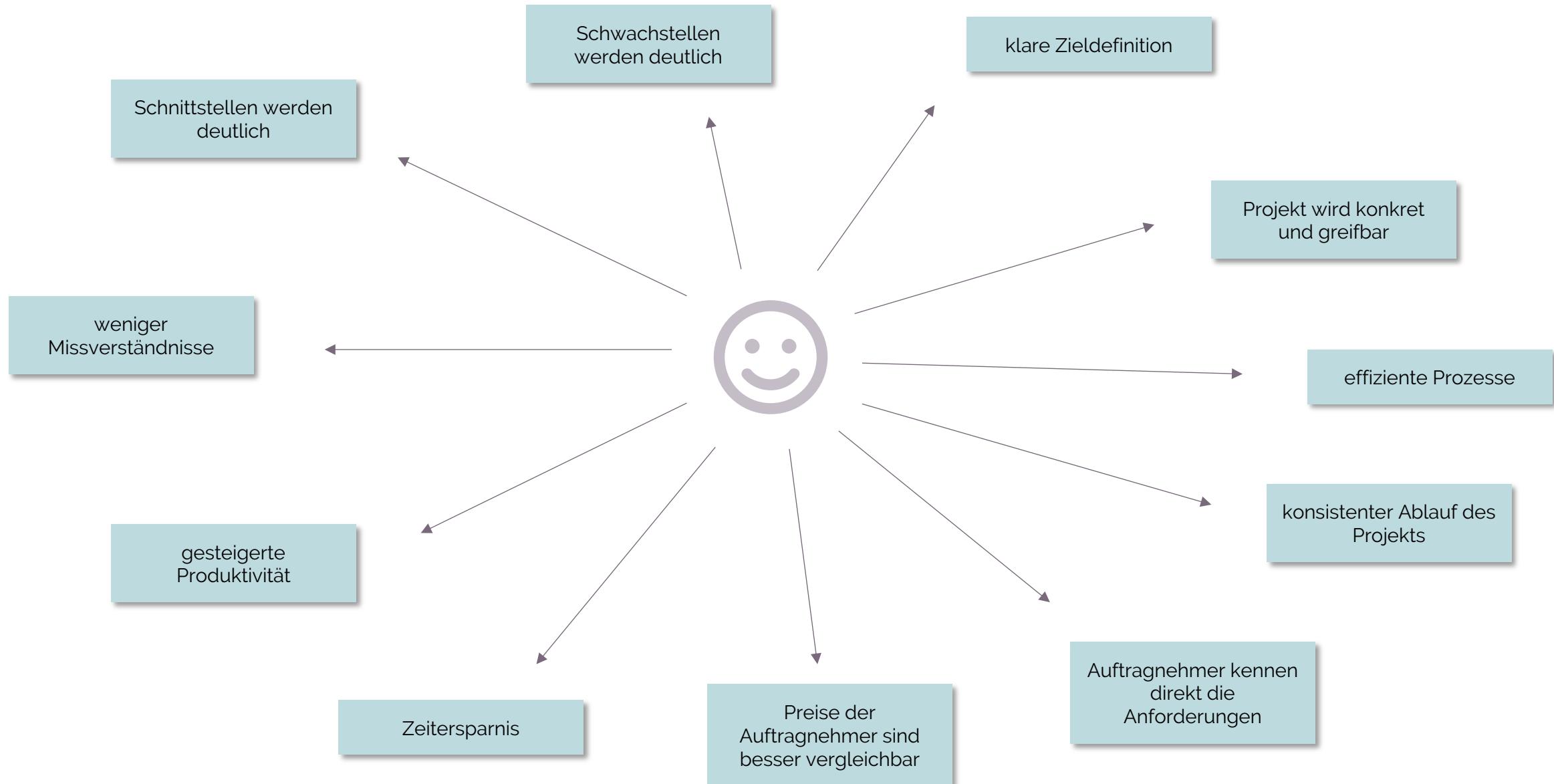
- Die Website gruenderszene.de definiert es wie folgt:
 - Der Begriff **Lastenheft** bezeichnet ein Dokument, in welchem ein Auftraggeber seine Anforderungen gegenüber dem Auftragnehmer festhält. Die genauen Anforderungen werden beschrieben, die der Auftraggeber an die Erreichung des Projektziels stellt.

Orientierungshilfe für Auftragnehmer

- Oft bezeichnet als **Anforderungs- oder Kundenspezifikation**
- detailliertes Lastenheft zu erstellen ist viel Arbeit
- ein gutes Lastenheft ist für ein Projekt **wie eine Wegbeschreibung**, die klar festlegt, wo es losgeht und wo es hingehen soll
- gut strukturiertes Lastenheft für Auftragnehmer attraktiv – es zeigt, dass Sie als Auftraggeber sich genau im Klaren darüber sind, wie Sie Ihr Projekt umsetzen wollen

- Planen Sie ausreichend Zeit für die Erstellung Ihres Lastenheftes ein. Wird es unter Zeitdruck erstellt, kommt es schnell zu Fehlern, die im späteren Verlauf des Projekts viel Zeit und Geld kosten können.

die wichtigsten Vorteile eines Lastenheftes im Überblick



Inhalte Lastenheft

Die Ausgangssituation

Beschreibung der Ist-Situation. Deutlich machen, warum das Projekt überhaupt zustande kommt. Welches Problem soll behoben werden? Deutlich machen, warum Handlungsbedarf besteht.

Das Ziel

Definiere genau, was mit dem Projekt oder dem fertigen Produkt erreicht werden soll. Achte darauf, quantifizierbare Ziele zu setzen – z. B. eine Umsatzsteigerung um 30 Prozent. Termine und einzuhaltende Zeitfenster sollten an dieser Stelle ebenfalls vermerkt werden

Das gewünschte Ergebnis

Definiere, was das Produkt können soll. Welche Funktionen muss es erfüllen? Dieser Punkt ist natürlich auch auf Prozesse übertragbar: Beschreibe z. B., wie diese nach dem Projekt aussehen sollen und welchen Mehrwert das für das Unternehmen haben wird

Offene Fragen

Konnte etwas bisher nicht geklärt oder festgelegt werden? Dafür gibt es diesen Teil des Lastenheftes. Schreibe alle offenen Fragen auf. Am besten merke an dieser Stelle auch an, wer sich um die Beantwortung der Fragen kümmert und welches Zeitfenster es dafür gibt

Der Lieferumfang

Halte so genau wie möglich fest, was wann geliefert werden soll. Wenn es mehrere Auftragnehmer gibt, beschreibe, wer was zu welchem Zeitpunkt liefert

Abnahme des Projekts bzw. Produkts

Wann gilt das Projekt als abgeschlossen und wer definiert, ob die Qualität stimmt? In vielen Unternehmen gibt es dafür spezielle Gremien. Lege klar fest, wer für die Abnahme verantwortlich ist

Der spätere Einsatz

Beschreibe, wie und wo das Produkt nach der Fertigstellung eingesetzt wird. Gehe ggf. auch auf Rahmenbedingungen wie klimatische Bedingungen ein, wenn diese eine Rolle bei der späteren Nutzung des Produktes spielen.

Die wichtigsten Phasen und Meilensteine

Projekte werden üblicherweise in verschiedene Phasen unterteilt. Beschreibe, was in den einzelnen Phasen passiert? Die Übergänge von einer Phase in die nächste sind Meilensteine. Halte fest, wann mit der Erreichung dieser Meilensteine gerechnet wird und definiere, was bis zum jeweiligen Zeitpunkt erreicht sein soll.

Das Pflichtenheft

beschreibt in konkreter Form, wie der Auftragnehmer die Anforderungen des Auftraggebers zu lösen gedenkt – das sogenannte *wie und womit*

Der Auftraggeber beschreibt vorher im Lastenheft möglichst präzise die Gesamtheit der Forderungen – was er entwickelt oder produziert haben möchte. Erst wenn der Auftraggeber das Pflichtenheft akzeptiert, sollte die eigentliche Umsetzungsarbeit beim Auftragnehmer beginnen

Es gibt **keine rechtlich vorgeschriebene Gliederung** für ein Pflichtenheft

Wichtig ist darüber hinaus, dass der **gesamte Ablauf des Projektes** beschrieben wird – mit allem, was dazugehört

Entscheidend ist, dass das Pflichtenheft **so genau wie möglich** ist. Beschreibe alle Abläufe und sorge dafür, dass man als Auftragnehmer einen detaillierten Überblick haben. Gibt es offene Fragen, suche nach Antworten. Gibt es Engpässe, kläre dies mit dem Auftraggeber. Sinn und Zweck des Pflichtenheftes ist, dass die Anforderungen des Lastenheftes geprüft werden. Ebenso sollte die Umsetzung genau geplant werden, damit es keine bösen Überraschungen gibt.

Beachte jedoch auch, dass immer etwas dazwischen kommen kann. Lasse also hin und wieder ruhig ein bisschen **Raum für Änderungen und Anpassungen**.

Wähle bei der Formulierung eine **verständliche Sprache** und verzichte, wenn möglich, auf Fachbegriffe. Denn das Pflichtenheft wird von verschiedenen Leuten gelesen – nicht alle haben ein tiefergehendes technisches Verständnis.

Um komplexe Inhalte verständlich zu vermitteln, bieten sich **grafische Darstellungen** an. Arbeitet mit Diagrammen, Tabellen oder Mind-Maps, um das Wichtigste so verständlich wie möglich zu machen.

Vermeide allgemeine Aussagen – etwa zu Zeiträumen oder Kosten. Das kann später zu schweren Meinungsverschiedenheiten führen. Je weniger Interpretationsspielraum Ihre Formulierungen bieten, desto besser.

Inhalte Pflichtenheft

Einleitung

Neben der expliziten Benennung des Auftragnehmers und des Auftraggebers sollte an dieser Stellung auch eine grobe Kurzbeschreibung des Projektes erfolgen. Gehe darauf ein, was das Projekt beinhaltet und wie das Endergebnis aussehen soll. Wichtig ist, dass auch eine Person, die das erste Mal von dem Projekt hört, versteht, worum es geht

Auftrag

Nun folgt die Formulierung des Auftrags. Anknüpfen an der Beschreibung aus der Einleitung an und Vertiefung durch Erläuterungen, beschreibe die Etappenziele sowie Rahmenbedingungen und gegebenenfalls auch die speziellen Besonderheiten des Projekts

Problemanalyse

Fasse die wichtigsten Probleme zusammen, die erwartet werden. Wichtig ist vor allem, dass für die wahrscheinlichsten Probleme bereits ein Lösungsansatz formuliert ist, um später Zeit zu sparen. Mach dir auch über unwahrscheinliche Probleme Gedanken

Bereits bestehende Systeme oder Produkte

Bestehen eventuell projektbezogene Systeme oder Produkte, die bereits vom Auftraggeber erledigt wurden und nun nicht mehr vom Auftragnehmer umgesetzt werden müssen? Sofern diese trotzdem für das Projekt relevant sind, erwähne sie ebenfalls im Pflichtenheft

Teams und Schnittstellen

Benenne möglichst genau, wer an dem Projekt mitwirkt und wo die Schnittstellen sind. Auch die Kommunikationswege in diesem Abschnitt auflisten.

Qualität

Welche Anforderungen stellst du an die Qualität? Beschreibe auch, wie die Qualitätssicherung, -kontrolle und -abnahme aussieht

Rahmenbedingungen

Hier wird zum Beispiel auf die gesamte Bearbeitungszeit eingegangen. Beschreibe ruhig auch die geplanten Betriebs- und Arbeitszeiten

Technische Anforderungen

Hier halte fest, was du für die Umsetzung brauchst – zum Beispiel Hard- und Software. An dieser Stelle ist es sinnvoll auszuführen, welches Equipment für welche Aufgabe benötigt wird

Projektentwicklung

Beschreibe genau, wann, was passiert und wie das gesamte Projekt organisiert ist.

Ziele einer Projektdokumentation

Projektdokumentation (PDO, gelegentlich auch Projektabschlussbericht) ist gem. [DIN 69901](#) die

- *Zusammenstellung ausgewählter, wesentlicher Daten über Konfiguration, Organisation, Mitteleinsatz, Lösungswege, Ablauf und erreichte Ziele des Projektes.*

Die Projektdokumentation sollte demnach Aufschluss darüber geben, welches Problem zu lösen war (IST-Zustand) und welche Lösung man angewendet hat (SOLL-Konzept). Darüber hinaus sollte auch geklärt werden, aus welchen Gründen man diesen Lösungsweg beschritten hat, sowie, wo der Gewinn liegt und die Kosten des Projektes (Projektverlauf). Sie ist unabdingbar für eine lückenlose, aussagekräftige Dokumentation, da sie eine sehr wichtige Informationsfunktion übernimmt. Es empfiehlt sich, parallel zur Projektabwicklung die PDO zu schreiben.

Typischer Aufbau einer Projektdokumentation

- Projektvorfeld (IST-Analyse)
 - *Aufgabenstellung (SOLL-Konzept)*
 - *Anforderungen*
- Projektziele (SOLL-IST-VERGLEICH)
- Zeitplanung / Meilenstein (Projektmanagement)
- Wirtschaftlichkeit
- Projektverlauf
- Ausblick / Folgeaktivitäten
- Fazit
- Anlagen / Anhang: (eigentlich Bestandteil der Projektakte)
 - *Pflichtenheft*
 - *Angebot*
 - *Preiskalkulation/Nutzwertanalyse*
 - *Kundeneinweisung*
 - *Abnahme- und Testprotokoll*
 - *Rechnung*

Perspektive

Auch wenn oftmals Projektdokumentationen in der "Ich-Form" geschrieben werden, sollte man dies möglichst vermeiden. Es empfiehlt sich hingegen eine personen neutrale, passive Darstellung.

Fachbegriff Prototyp

- **Prototyping** bzw. **Prototypenbau** ist eine Methode der [Softwareentwicklung](#), die schnell zu ersten Ergebnissen führt und *frühzeitiges Feedback* bezüglich der Eignung eines Lösungsansatzes ermöglicht. Dadurch ist es möglich, Probleme und Änderungswünsche frühzeitig zu erkennen und mit weniger Aufwand zu beheben, als es nach der kompletten Fertigstellung möglich gewesen wäre.
- Ein Prototyp steht für ein lauffähiges Stück Software oder eine anderweitige konkrete Modellierung (z. B. [Mock-up](#)) einer Teilkomponente des Zielsystems. Dieser Prototyp dient anschließend oft als Basis für eine bessere Kommunikation mit den Kunden oder auch innerhalb des Entwicklungsteams über konkrete Dinge (statt abstrakte Modelle).

Vor- und Nachteile von Prototyping

Vorteile

- Die Anforderungen der Anwender können laufend präzisiert und verifiziert werden. Damit sinkt das Risiko einer Fehlentwicklung.
- Unbeabsichtigte Wechselwirkungen zwischen einzelnen Komponenten des Produkts können früher erkannt werden.
- Der Fertigstellungsgrad ist besser verifizierbar.
- Die Qualitätssicherung kann frühzeitig eingebunden werden.

Nachteile

- Prototyping verführt oft dazu, Anforderungen weder korrekt zu erheben noch sauber zu dokumentieren. Der Entwicklungsprozess kann sich dadurch erheblich verlangsamen.
- Es entstehen während der Entwicklung zusätzliche Kosten, weil der Prototyp nur als Basis für die folgende eigentliche Entwicklung des Produktes dient. Diese Kosten und Zeitaufwand können durch weniger Nacharbeit am Endprodukt wieder ausgeglichen werden.

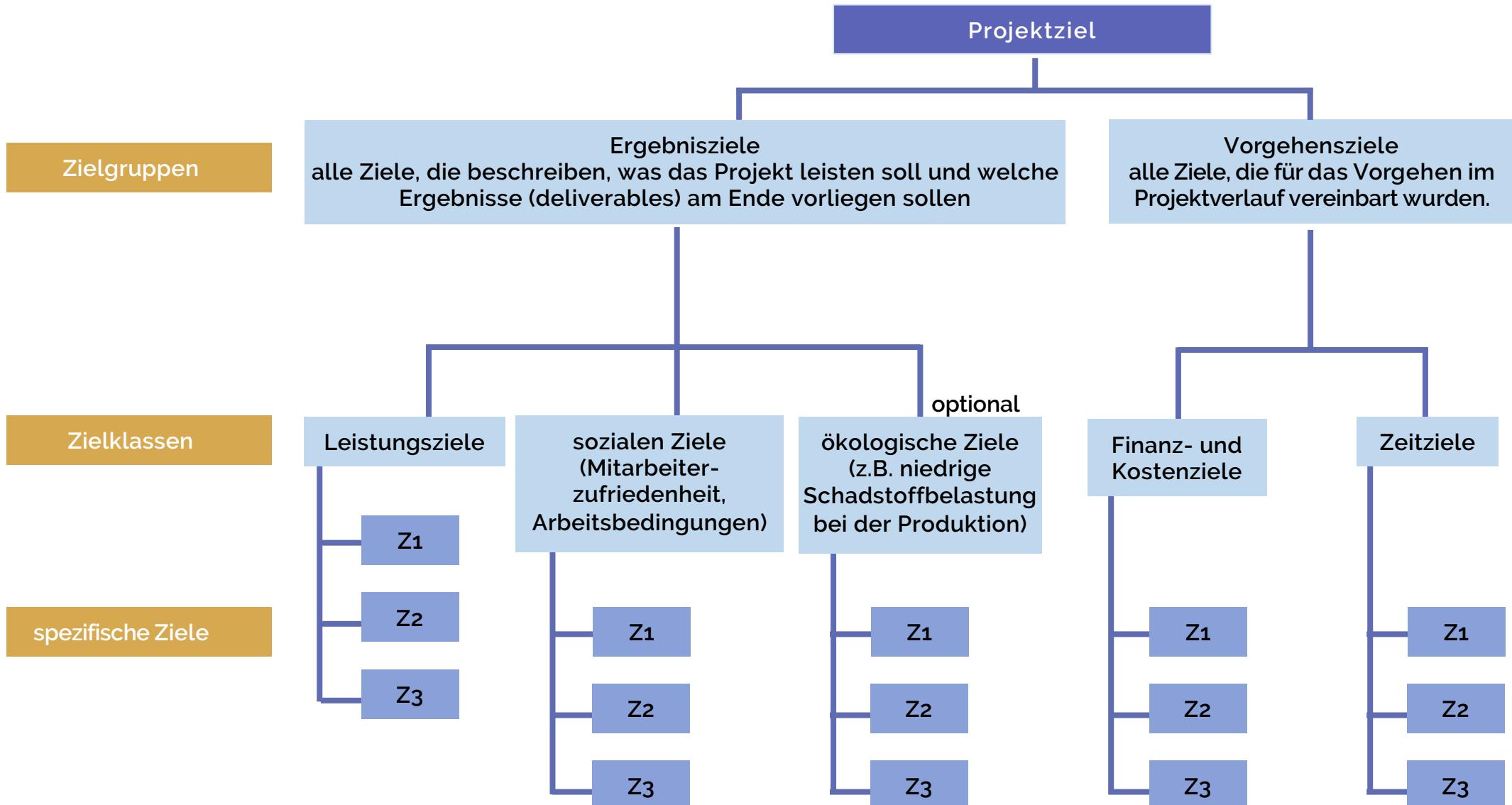
Arten von Prototyping

Art	Ziel	Wichtigstes Ergebnis
Rapid Control Prototyping	Rapid Control Prototyping bezeichnet die Softwareentwicklung von Regelungen und Steuerungen mit Hilfe grafischer Werkzeuge. Diese ist nicht zu verwechseln mit dem aus dem Maschinenbau bekannten <u>Rapid Prototyping</u> .	
Exploratives Prototyping	Ziel ist es nachzuweisen, dass Spezifikationen oder Ideen tauglich sind.	Eine übersichtliche und belastbare <u>Anforderungsspezifikation</u>
Evolutionäres Prototyping	Anhand der Grundfunktionalitäten die Akzeptanz beim Nutzer und die Notwendigkeit ergänzender Funktionen zu überprüfen	Ein Programm mit den Grundfunktionalitäten
Experimentelles Prototyping	Sammeln von Erfahrungen mit dem Prototyp	ein erster experimenteller Prototyp
Vertikales Prototyping (Durchstich)	Die Entwicklung eines funktionalen Ausschnitts eines Programmes	Ein ausgewählter Teil des Systems ist durch alle Ebenen hindurch implementiert.
Horizontales Prototyping	Eine funktionierende Ebene, die vorgestellt werden kann, oder an der sich andere Ebenen orientieren können.	Eine ausgewählte Ebene des Gesamtsystems ist fertiggestellt



Ziele und Nicht Ziele

klassische Methode: Zielgruppen, Zielklassen, spezifische Ziele



weiterf黨render Link

<https://erfolgreich-projekte-leiten.de/zielhierarchie/>

Muss-, Soll-, Kann-Ziele



Muss-Ziele



Soll-Ziele



Kann-Ziele



unveränderlich



veränderbar



nice-to have



vertraglich fixiert



gewünschte Ziele



für Erfolg nicht wichtig



bei nicht erreichen: Projekt
teilweise/ganz gescheitert

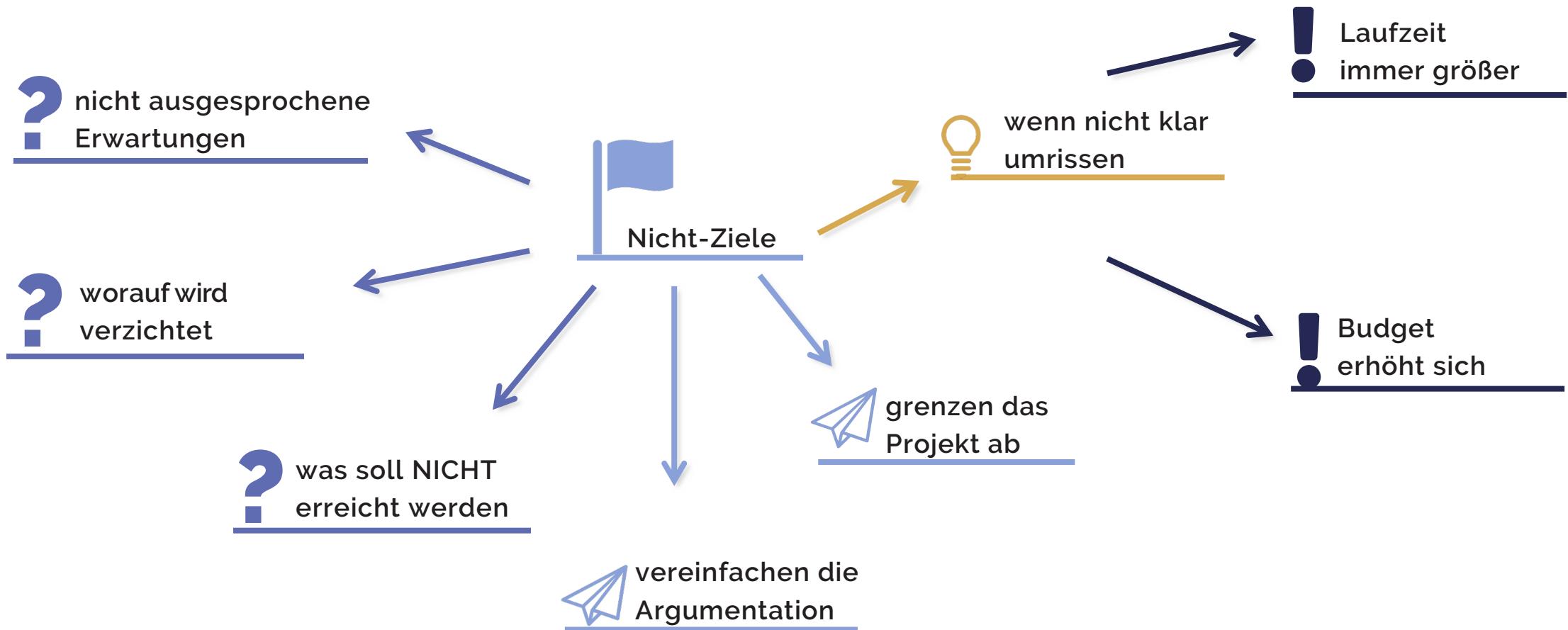


können gestrichen
werden bei Bedarf



werden als erstes
gestrichen bei Bedarf

Nicht-Ziele



Info

- verhindern ein Ausufern des Projektes
- gehören nicht zum Projektinhalt



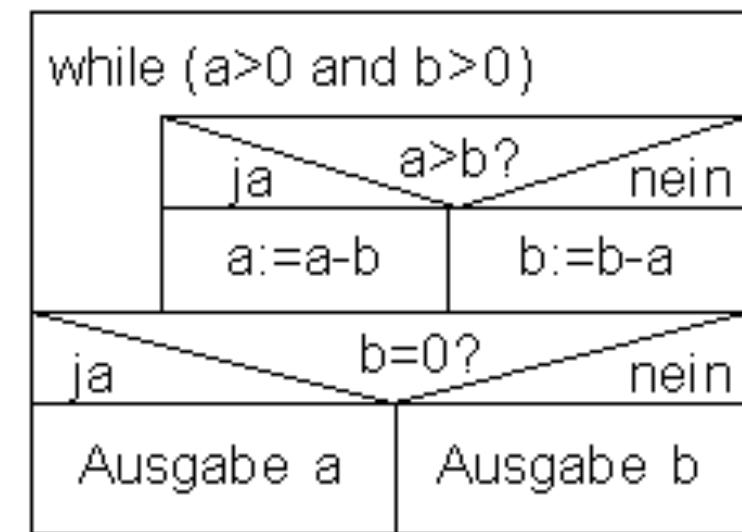
Projektstrukturplan Struktogramm Ablaufdiagramm

Fachbegriff Softwareentwurf

- Der Begriff *Softwareentwurf* bezeichnet eine Tätigkeit im Rahmen der Entwicklung eines Softwaresystems, die auf den Ergebnissen der Anforderungsdefinition (etwa dem Pflichtenheft oder dem Lastenheft) aufsetzt und sich als eigene Phase - oder aufgeteilt auf verschiedene, spezielle Entwurfsphasen (z. B. Architekturentwurf oder Komponentenentwurf) – in typischen Vorgehensmodellen der Systementwicklung wiederfindet.
- In Anlehnung an [IEEE 1990, S. 25] versteht man unter dem Softwareentwurf den
 - Prozess der Definition der Architektur, Komponenten, Schnittstellen und anderer Merkmale eines Software(teil)systems.

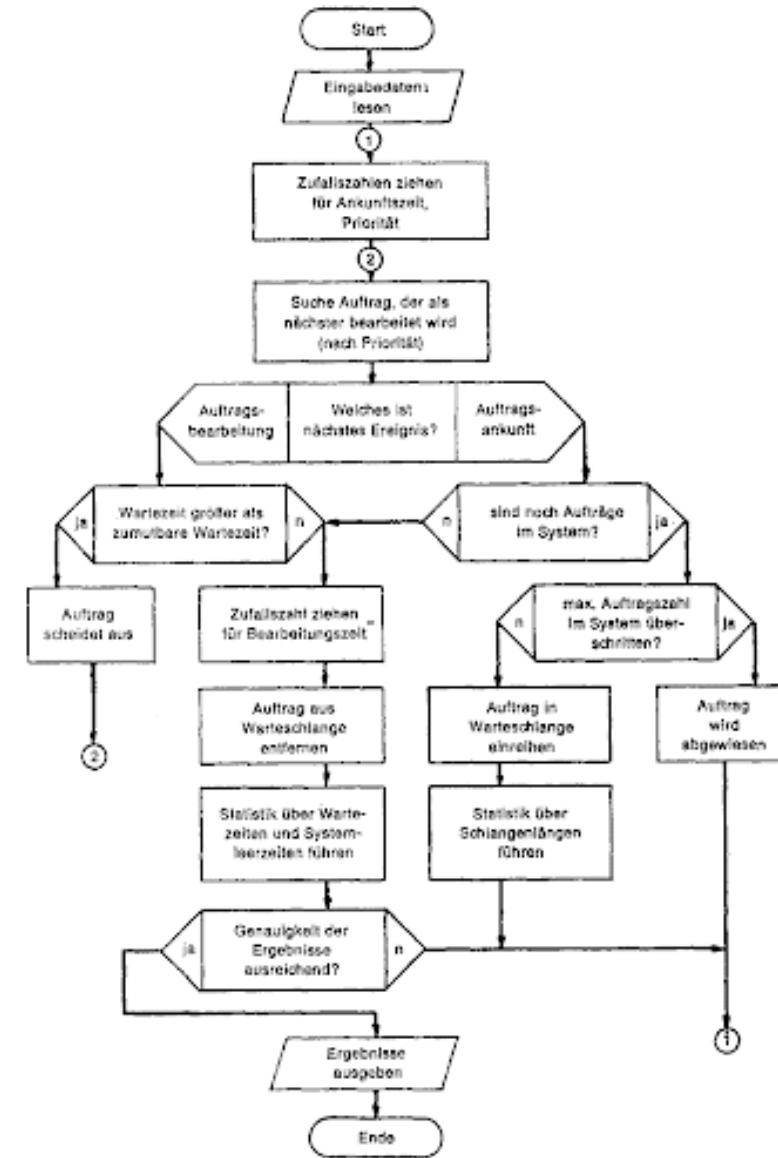
Fachbegriff Struktogramm

Es wurde 1972/73 von Isaac Nassi und Ben Shneiderman entwickelt und ist in der DIN 66261 genormt. Da Nassi-Shneiderman-Diagramme Programmstrukturen und Kontrollstrukturen darstellen, werden sie auch als **Struktogramme** bezeichnet.



Fachbegriff Ablaufdiagramm (Flowchart)

Die **Ablaufdiagramme**, auch Gantt-Diagramme genannt, geben als Auftragsfolgediagramme und Maschinenfolgediagramme die zeitliche Struktur des Problems an.





Der Netzplan, Meilensteinplan und der Balkenplan

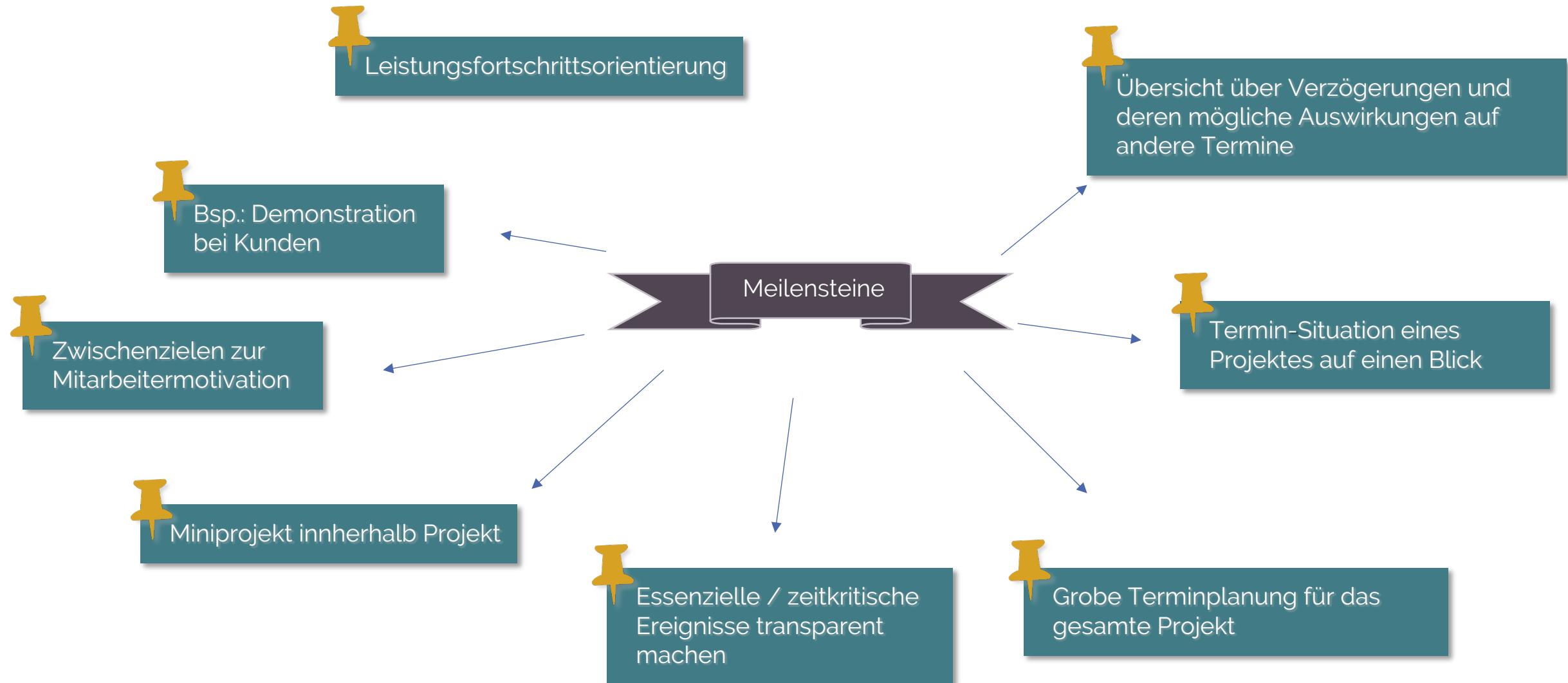
Netzplan

- Instrument zur Planung, Überwachung und Steuerung von einzelnen Projekten.
- Vorläufer des Netzplanes ist das Balkendiagramm
- Ziel: zeitl. Abfolge der einzelnen Vorgänge und deren Abhängigkeiten darzustellen; kritischen Weg ermitteln; Gesamtaufzeit des Projekts ermitteln

- Erkennen des "kritischen Wegs"
- ermöglicht Vergleich verschiedener Planungsvarianten
- fördert rechtzeitige Entscheidungen, da mögliche Konsequenzen im Netzplan ersichtlich sind
- geeignet für:
 - Strukturplan
 - Zeitplan
 - Einsatzmittelplan
 - Kostenplan



Meilensteine



Info

Meilensteine sind wesentliche Zeitpunkte in einem Projekt, an die üblicherweise Ergebnisse (= Deliverables) und Entscheidungen (= Stop-or-Go) geknüpft werden.



Meilensteinplan

Erfasst insbesondere zeitkritische Ereignisse, die bei Verzögerung oder Nichteinhaltung den Projektverlauf negativ beeinflussen. Komplexe Projekte können so übersichtlich strukturiert werden

Ziel der Meilensteinplanung:

- Grobe Terminplanung für das gesamte Projekt
- Essenzielle Ereignisse im Projekt transparent machen
- Übersicht über Verzögerungen und deren mögliche Auswirkungen auf andere Termine
- Leistungsfortschrittsorientierung
- Schaffen von Zwischenzielen zur Mitarbeitermotivation

Balkenplan

Stellt die geplanten Tätigkeiten auf einer Zeitachse in Form von Balken dar.
Dies ermöglicht einen schnellen Überblick über die zeitliche Planung sowie Dauer der Arbeitspakete.

Was man herauslesen kann:

- Arbeitspakete (vertikale Achse)
- zeitlicher Ablauf (horizontale Zeitachse)
- Parallelle Tätigkeiten
- Abhängigkeiten (durch Pfeile gegenzeichnet)
- Meilensteine (schwarze Raute)
- Kritischer Pfad (in Rot)
- Pufferzeiten (blaue Balken haben Pufferzeiten)



Klassische Methode: Die Smart-Formel



Beispiel:

„Ich werde bis zum 01.01.2020 (*terminiert*) durch meine selbstständige Arbeit als [...] (*spezifisch*) im Monat € 5.000,- brutto (*messbar*) verdienen. (Gesamtziel ist *realistisch* und *attraktiv*).“



Identifizierung von Funktionen

Was hast du vor und warum ist es wichtig?

Mach dir erstmal ein Bild davon, um was es bei dem Projekt überhaupt geht.

Ordne die genannten Ziele nach Priorität und Zeitraum

Du willst Resultate liefern und dafür brauchst du messbare Ergebnisse. Deshalb ist es wichtig zu wissen, welche Ziele höhere Prioritäten haben

Wer ist die Zielgruppe in diesem Projekt?

Welche Personen muss das Produkt später ansprechen? Jung oder alt, beeinträchtigt usw. Muss ich also mit bestimmten Farben arbeiten, großer Typo oder ähnliches.

Wie sieht Erfolg für dieses Projekt aus?

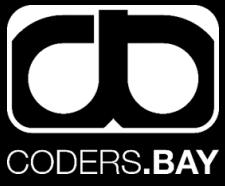
Ab wann ist man zufrieden, was ist der Optimalfall, der erreicht werden kann?

Was sind die Ziele?

Sie sind da um zu wissen auf was man sich fokussieren muss und was man erreichen möchte.

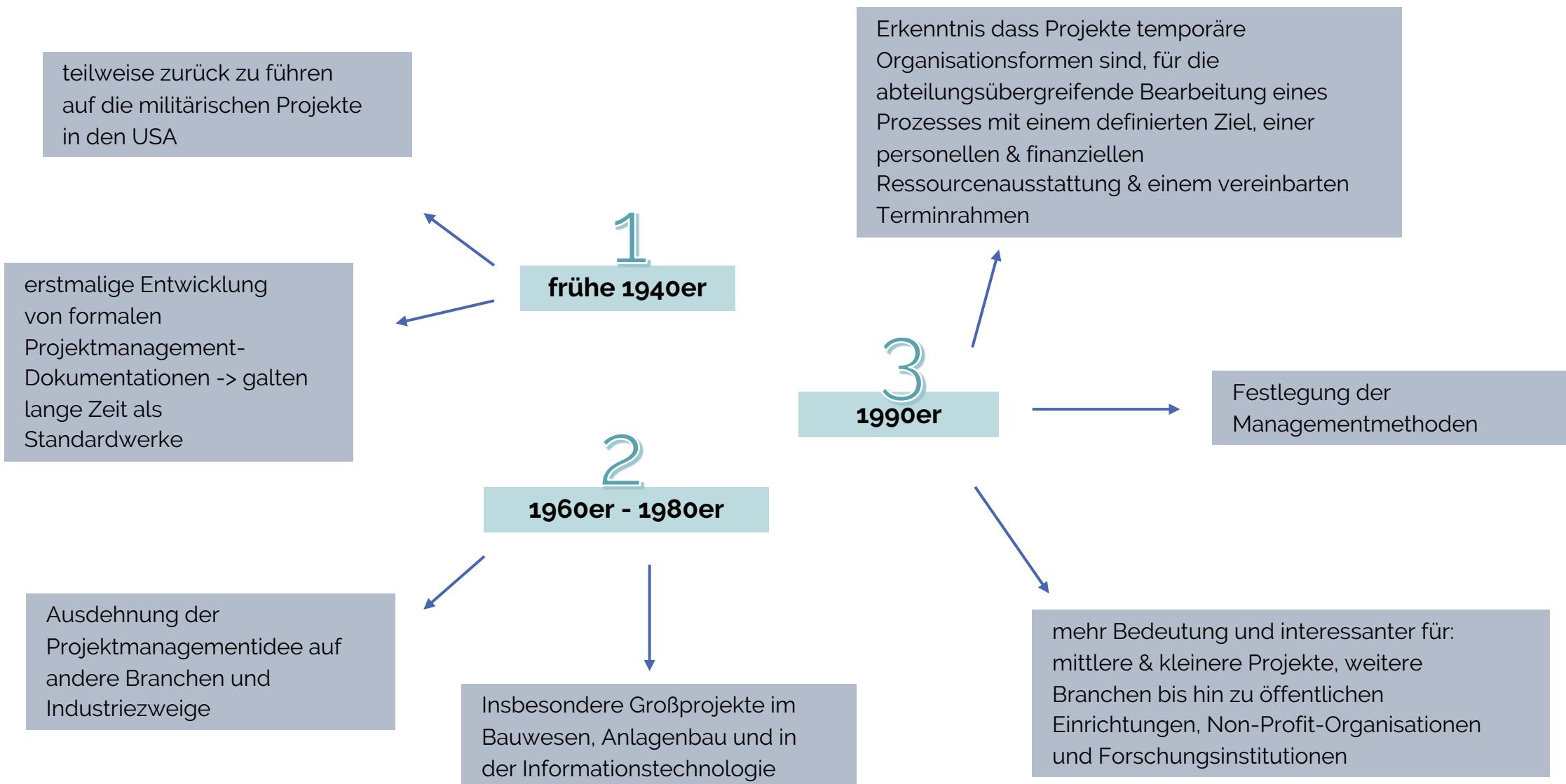
Was ist, wenn das Projekt scheitert?

Es könnte auch sein, dass etwas nicht die genannten Ziele erreicht. Was passiert dadurch auf Kundenseite? Wie stark würde es ihm/mir schaden? Wie viel Risiko steckt hier im Projekt



Agilität

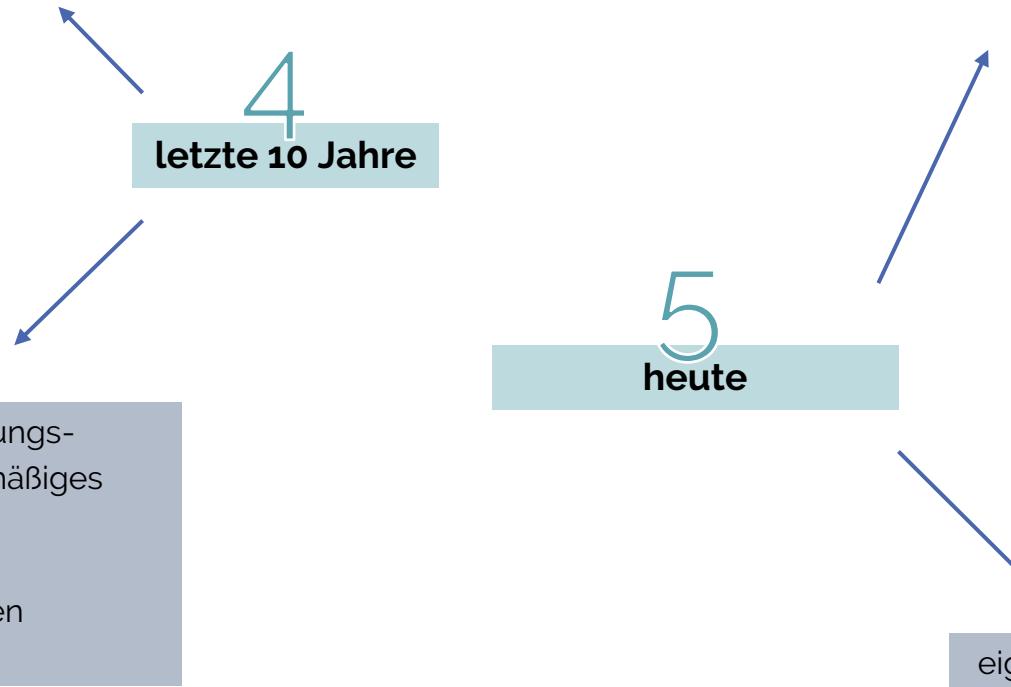
Die 5 Entwicklungsstufen des Projektmanagements



Die 5 Entwicklungsstufen des Projektmanagements

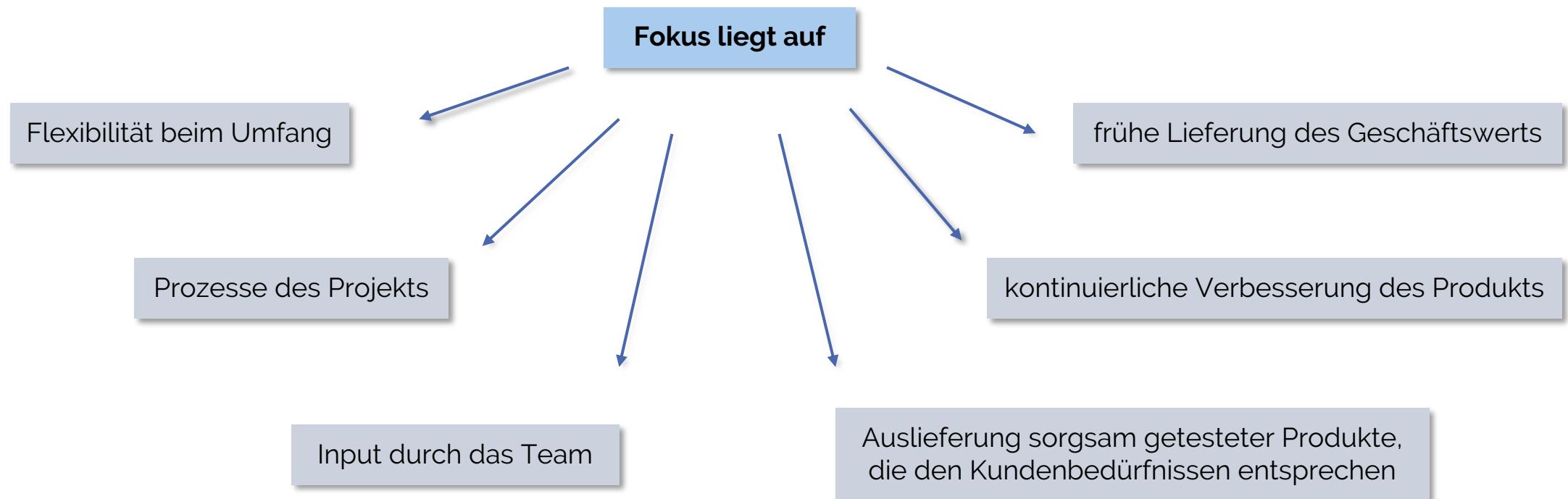
Entwicklung & Etablierung von alternativen Projektmanagementmethoden mit Ursprung in der Entwicklung von Software und in IT-Projekten -> agilen Projektmanagementmethoden

Hybridform aus klassischen und agilen Managementmethoden gekoppelt mit zentralen Elementen wie (Team) Führung, (Fach)Wissen oder systemischen Ansätzen



Projektmanagement modernisieren

Agile Methoden im Projektmanagement => Modell für das Projektmanagement



Kenntnisse über agiles Projektmanagement/Methoden

Der Begriff Agilität steht für Wendigkeit oder Beweglichkeit. Wer also nach dem Prinzip der Agilität handelt, reagiert flexibler und schneller. Das hilft, sich auf unvorhergesehene Ereignisse und Veränderungen besser vorzubereiten und anzupassen. Im Projektmanagement spricht man von einer innovativen Form der Arbeitsorganisation und Überarbeitung bisheriger, klassischer Strukturen. Eine Prozessoptimierung und Vereinfachung von Workflows spart Ressourcen und ermöglicht langfristig gesehen eine effizientere Arbeitsweise.

Agile Vorteile



Schnellere Auslieferung von Produkten



Besseres Management wechselnde Prioritäten



Steigerung der Produktivität



Bessere interne Zusammenarbeit mit der IT-Abteilung



Reduzierung der Projektrisiken



Bessere Projektübersicht



Steigerung der Teammoral



Vorhersagbarkeit der Produktauslieferung verbessern



Verbesserung der Entwicklungsdisziplin



Verteilte Teams besser managen



Reduzierung der Projektkosten

Zitat

Agilität ist eine Haltung, eine Unternehmenskultur mit gelebten Werten. Es gibt klare Vorgehensweisen und definierte Werkzeuge, die die Selbstorganisation von Einzelpersonen, Teams und Organisationen unterstützen.



Das agile Manifest (2001)



„Wir erschließen bessere Wege, Software zu entwickeln, indem wir es selbst tun und anderen dabei helfen. Durch diese Tätigkeit haben wir diese Werte zu schätzen gelernt

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

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

– Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland und Dave Thomas

die 4 Werte des agilen Manifest



Individuen &
Interaktionen

stehen über Prozessen
und Werkzeugen



Zusammenarbeit
mit dem Kunden

steht über der
Vertragsverhandlung



Funktionierende
Software

steht über einer umfassenden
Dokumentation



Reagieren auf
Veränderung

steht über dem Befolgen eines
Plans



Info

Formuliert Februar 2001 bei einem Treffen in Utah.

die 12 Prinzipien des agilen Manifest

Als wichtigstes Fortschrittsmaß gilt die Funktionsfähigkeit der Software

Einhalten eines gleichmäßigen Arbeitstempos von Auftraggebern, Entwicklern und Benutzern für eine nachhaltige Entwicklung

Ständiges Augenmerk auf technische Exzellenz und gutes Design

Lieferung von funktionierender Software in regelmäßigen, bevorzugt kurzen Zeitspannen (wenige Wochen oder Monate)

Selbstreflexion der Teams über das eigene Verhalten zur Anpassung im Hinblick auf Effizienzsteigerung

Bereitstellung des Umfeldes und der Unterstützung, welche von motivierten Individuen für die Aufgabenerfüllung benötigt wird

Zufriedenstellung des Kunden durch frühe und kontinuierliche Auslieferung von wertvoller Software

Agile Prozesse nutzen Veränderungen (selbst spät in der Entwicklung) zum Wettbewerbsvorteil des Kunden.

Einfachheit ist essenziell

Nahezu tägliche Zusammenarbeit von Fachexperten und Entwicklern während des Projektes (Bsp.: Gemeinsamer Code-Besitz (Collective Code Ownership))

Selbstorganisation der Teams bei Planung und Umsetzung

Informationsübertragung nach Möglichkeit im Gespräch von Angesicht zu Angesicht

Der agile Lackmustest

Unterstützt das, was wir aktuell tun, die frühe und kontinuierliche Lieferung wertvoller Software?



Heißt unser Prozess Veränderungen willkommen und macht es sich deren Vorteile zunutze?



Führt unser Prozess zur Funktionalität und unterstützt er die Lieferung einsatzfähiger dieser?



Arbeiten die Entwickler und der Product Owner täglich zusammen?
Arbeiten der Kunde und die Business Stakeholder eng mit dem Projektteam zusammen?



Stellt unsere Arbeitsumgebung dem Entwicklungsteam die Unterstützung bereit, die es braucht, um seine Arbeit zu erledigen?



Kommunizieren wir persönlich anstatt über das Telefon oder per Mail?



Messen wir den Fortschritt anhand der Menge der erstellten, einsatzfähigen Funktionalität?



Können wir das Tempo auf unbegrenzte Zeit beibehalten?



Unterstützen wir technisches Qualität und ein gutes Design, dass zukünftige Änderungen möglich macht?



Maximieren wir die Menge der nicht getanen Arbeit, das heißt, machen wir nur so wenig wie nötig, um das Ziel des Projekts zu erreichen?



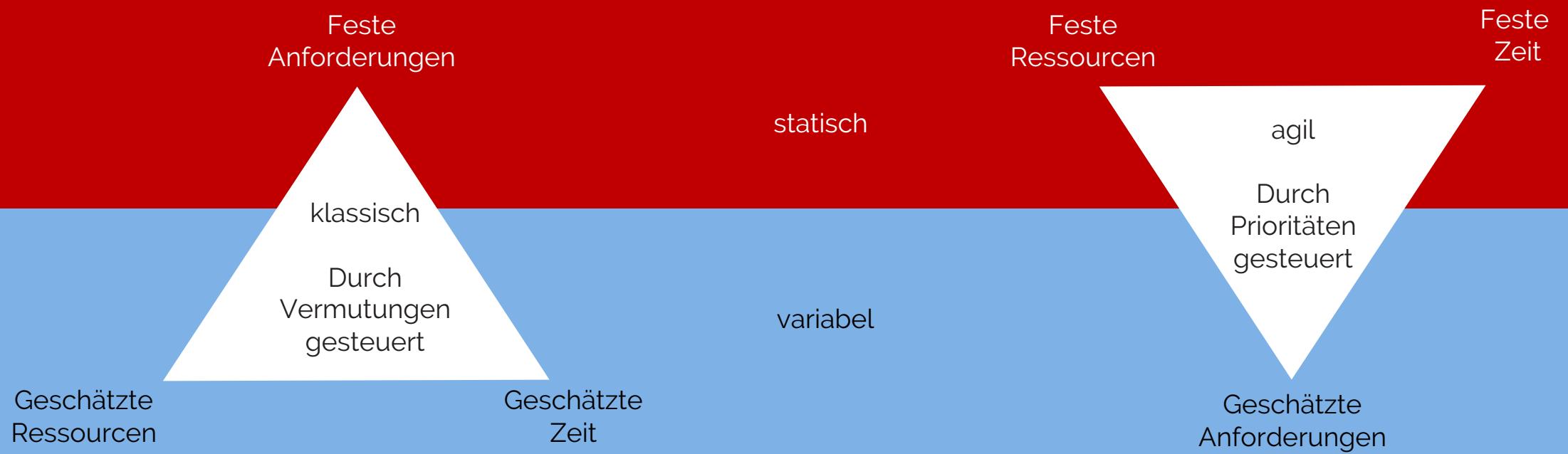
Organisiert und steuert sich das Entwicklungsteam selbst? Besitzt es die Freiheit erfolgreich zu sein?



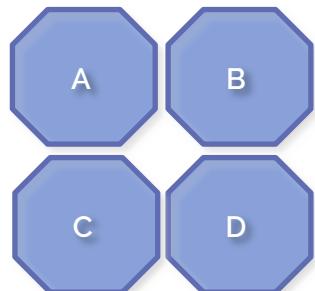
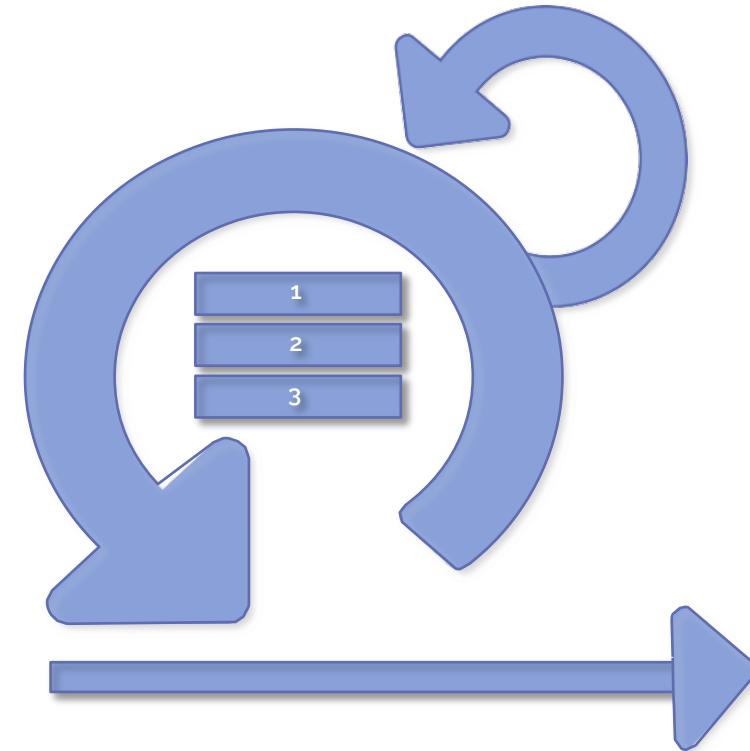
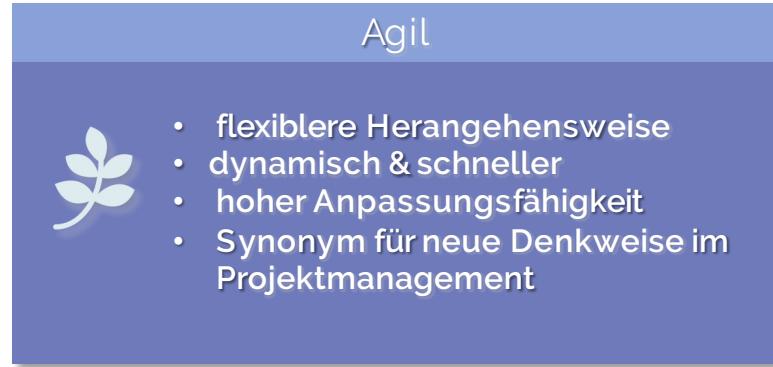
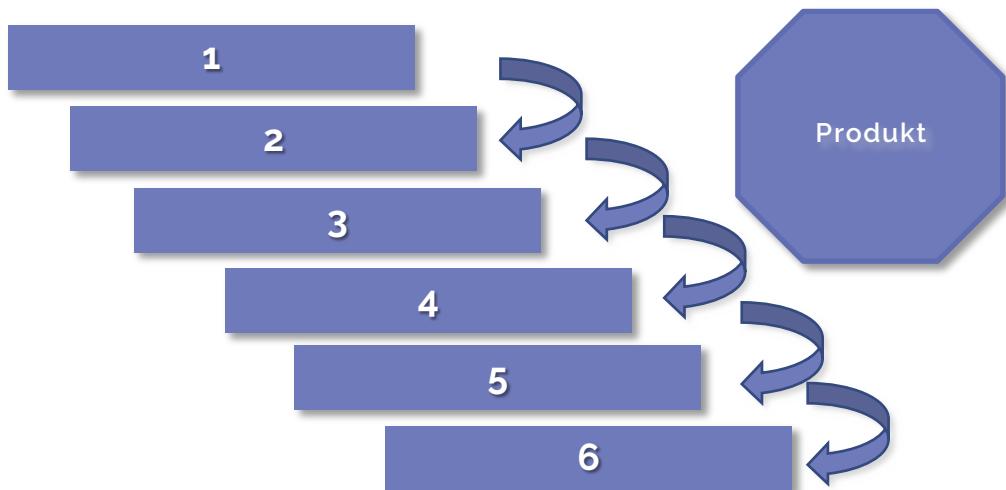
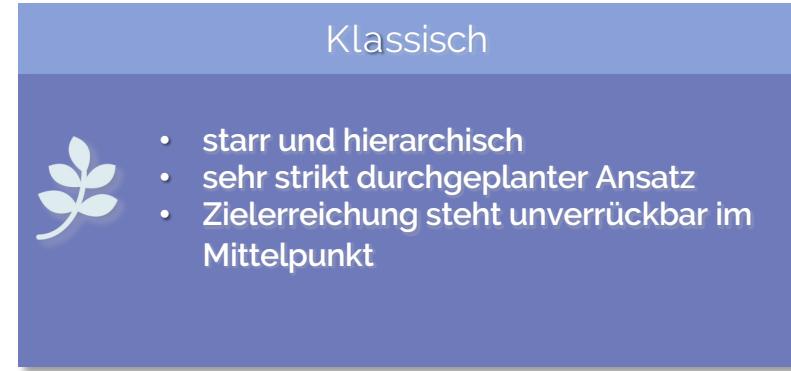
Reflektieren wir in regelmäßigen Intervallen und passen wir unser Verhalten dementsprechend an?



klassisch vs. agil



klassisch vs. agil





Projektmanagementmethoden

Kenntnisse über Softwareprozessmodelle

- Softwareprozesse sind die Aktivitäten zum Entwerfen, Implementieren und Testen eines Softwaresystems. Der Softwareentwicklungsprozess ist kompliziert und beinhaltet viel mehr als technisches Wissen.
- Hier bieten sich Software-Prozessmodelle an. Ein Softwareprozessmodell ist eine **abstrakte Darstellung** des Entwicklungsprozesses.
- Ein Softwareprozessmodell ist eine Abstraktion des Softwareentwicklungsprozesses. Die Modelle geben die Phasen und die Reihenfolge eines Prozesses an. Stellen Sie sich dies als eine Darstellung der **Reihenfolge der Aktivitäten** des Prozesses und der **Reihenfolge vor**, in der sie ausgeführt werden.
- **Ein Modell definiert Folgendes:**
 - Die auszuführenden Aufgaben
 - Die Eingabe und Ausgabe jeder Aufgabe
 - Die Vor- und Nachbedingungen für jede Aufgabe
 - Der Ablauf und die Reihenfolge jeder Aufgabe
- Es gibt viele Arten von Prozessmodellen, um unterschiedliche Anforderungen zu erfüllen. Wir bezeichnen diese als **SDLC-Modelle** (Software Development Life Cycle-Modelle). Die beliebtesten und wichtigsten SDLC-Modelle sind:
 - Wasserfall-Modell
 - V-Modell
 - Inkrementelles Modell
 - RAD-Modell
 - Agiles Modell
 - Iteratives Modell
 - Prototypmodell
 - Spiralmodell
- Die Auswahl des richtigen Softwareprozessmodells für Ihr Projekt kann schwierig sein. Wenn Sie Ihre Anforderungen gut kennen, ist es einfacher, ein Modell auszuwählen, das Ihren Anforderungen am besten entspricht. Bei der Auswahl Ihres Software-Prozessmodells müssen Sie die folgenden Faktoren berücksichtigen:
 - Projektanforderung
 - Projektgröße
 - Projektkomplexität
 - Kosten für Verspätung
 - Einbeziehung der Kunden
 - Vertrautheit mit Technologien
 - Projektressourcen

Arten von Softwareprozessmodellen

Wasserfall
Modell

Agiles Modell

Spiralmodell

V-Modell

RAD-Modell

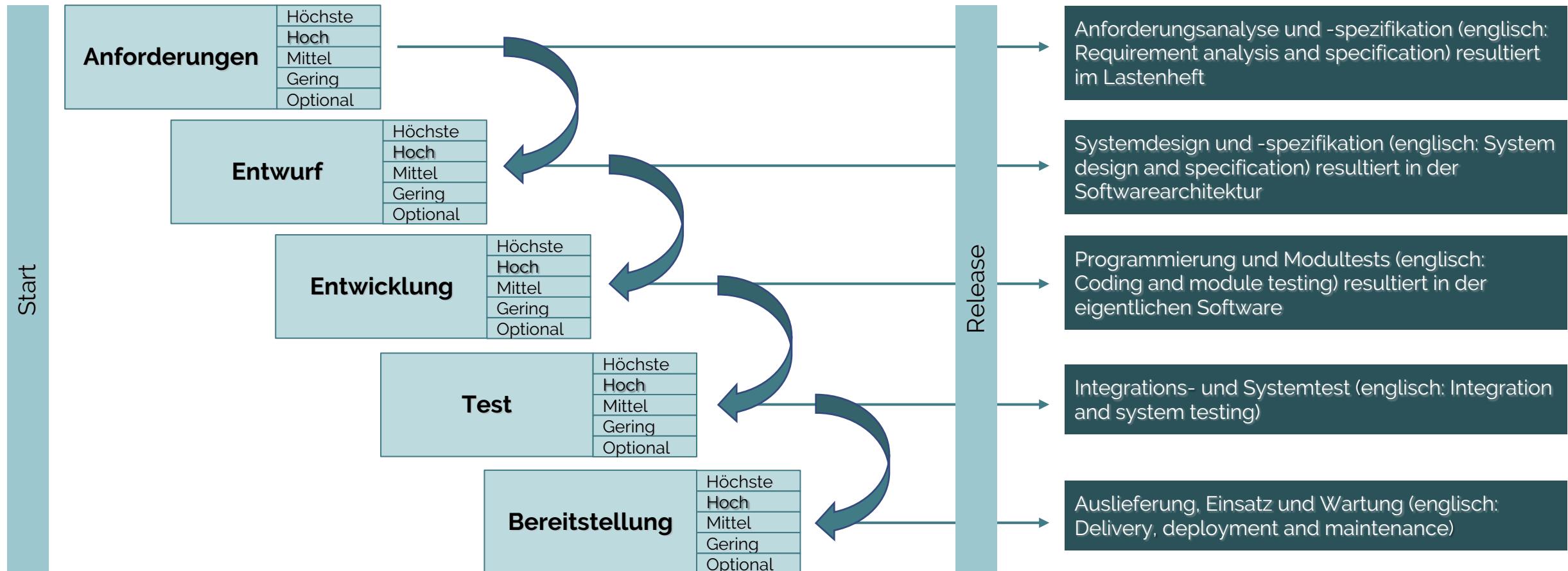
Inkrementelles
Modell

Iteratives Modell

Das Wasserfallmodell

- Ein **Wasserfallmodell** ist ein lineares (nicht iteratives) Vorgehensmodell, das insbesondere für die Softwareentwicklung verwendet wird und das in aufeinander folgenden Projektphasen organisiert ist. Wie bei einem Wasserfall mit mehreren Kaskaden "fallen" die Ergebnisse einer Stufe nach unten in die nächste und sind dort verbindliche Vorgaben.
- In einem Wasserfallmodell hat jede Phase vordefinierte Start- und Endpunkte mit eindeutig definierten Ergebnissen. Meist beschreibt das Modell auch einzelne Aktivitäten, die zur Herstellung der Ergebnisse durchzuführen sind. Zu bestimmten Meilensteinen und am jeweiligen Phasenende werden die vorgesehenen Entwicklungsdocumente im Rahmen des Projektmanagements verabschiedet.
- Der Softwareprozess **ist nicht linear**, daher müssen die erstellten Dokumente möglicherweise geändert werden, um Änderungen widerzuspiegeln.
- Das Wasserfallmodell ist leicht zu verstehen und zu befolgen. Es erfordert nicht viel Kundenbeteiligung, nachdem die Spezifikation fertig ist. Da es **unflexibel** ist, kann es sich nicht an Änderungen anpassen. Es gibt keine Möglichkeit, die Software bis zur letzten Phase zu sehen oder zu testen.
- Das Wasserfallmodell hat eine **starre Struktur** und sollte daher in Fällen verwendet werden, in denen die Anforderungen vollständig verstanden werden und sich wahrscheinlich nicht radikal ändern.
- Bei der Umsetzung schrittweise ablaufender Projektmanagement-Methoden, zeigen sich Fehler häufig erst am Ende eines Projektes. Daher ist die Wasserfall Projektmanagement-Methode für Projekte mit vielen unvorhersehbaren Faktoren wenig geeignet.

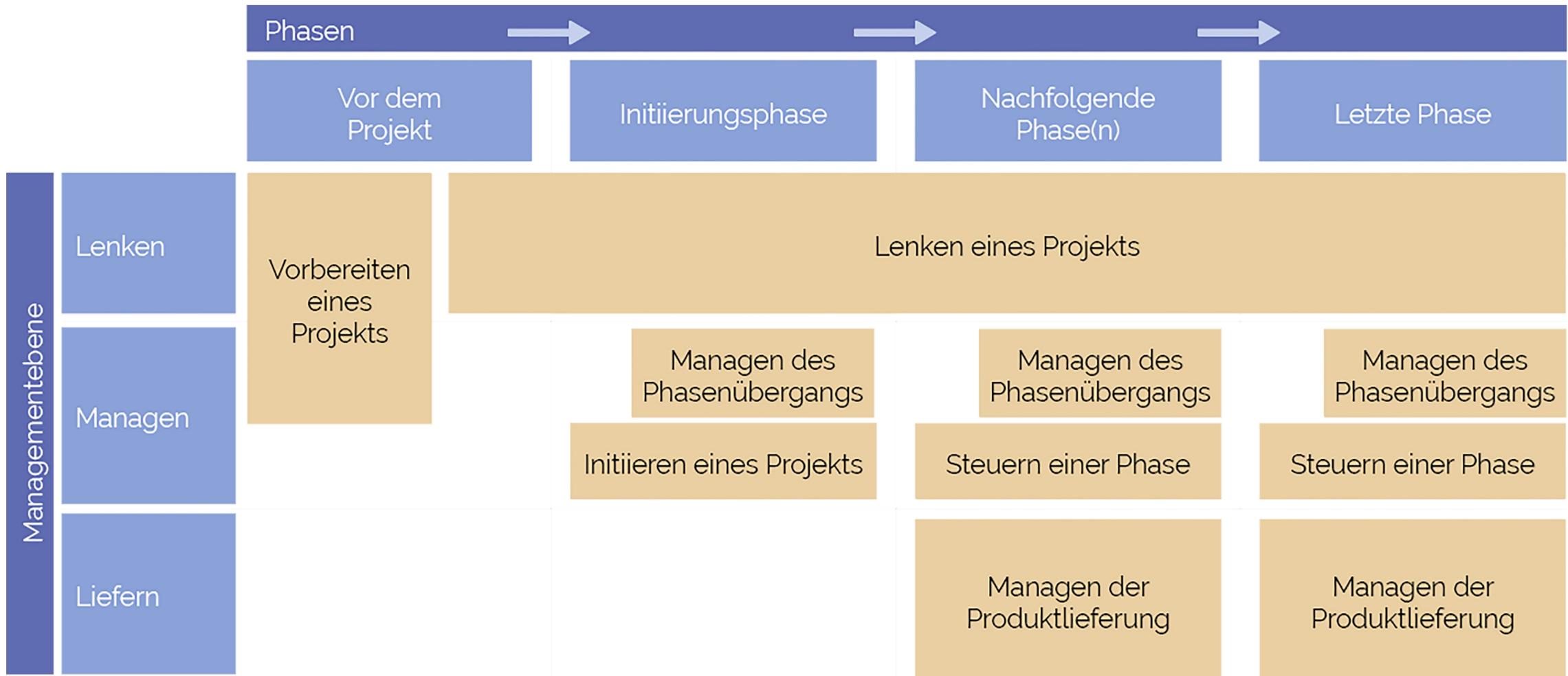
klassische Wasserfallmethode



Info

erste bekannte Beschreibung in der Softwareentwicklung von Herbert D. Benington beim Symposium on advanced programming methods for digital computers am 29. Juni 1956

Prince2

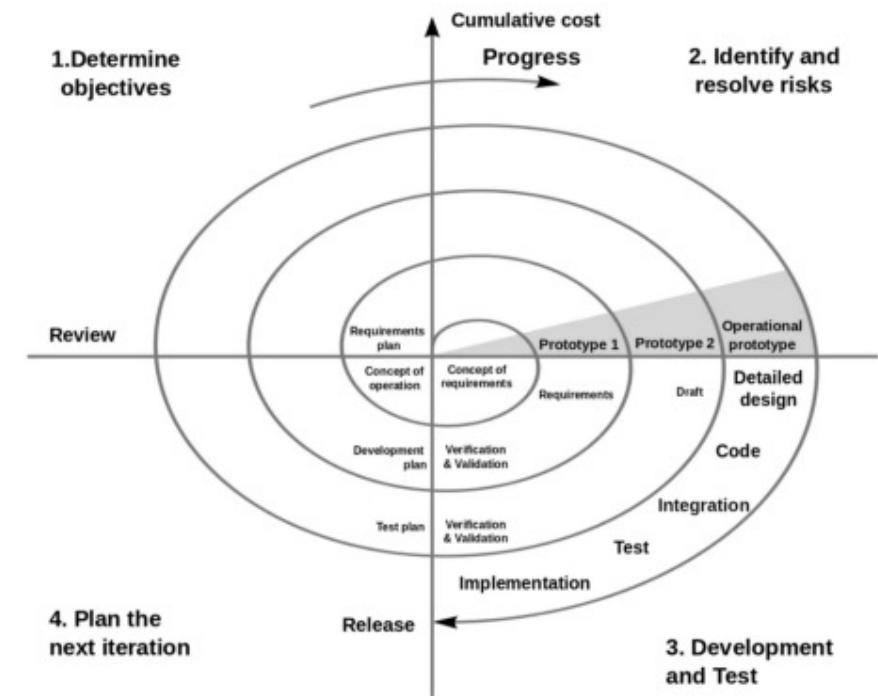


Info

1989 von britischen Central Computer and Telecommunications Agency (CCTA) als Regierungsstandard für Projektmanagement im Bereich der Informationstechnik (IT) entwickelt

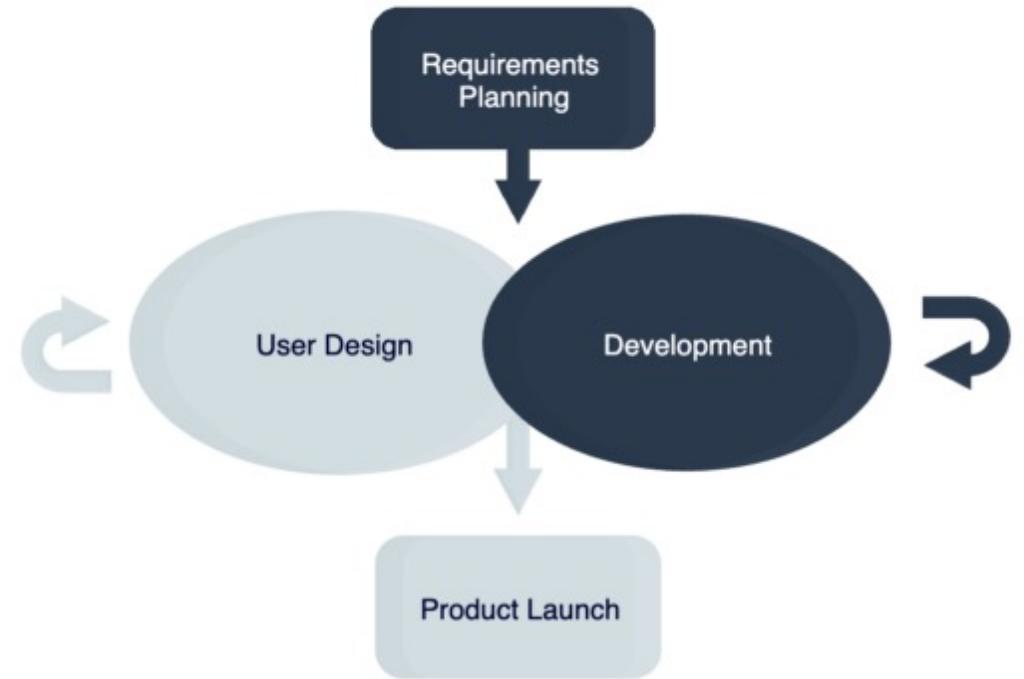
Spiralmodell

- Das Spiralmodell ist ein risikogesteuertes iteratives Softwareprozessmodell. Das Spiralmodell liefert Projekte in Schleifen. Im Gegensatz zu anderen Prozessmodellen sind die Schritte keine Aktivitäten, sondern **Phasen** zur Behebung des Problems, bei dem das größte Risiko besteht, dass ein Fehler auftritt.
- Es wurde entwickelt, um die besten Eigenschaften des Wasserfalls zu berücksichtigen und eine Risikobewertung einzuführen.*
- Sie haben die folgenden Phasen für jeden Zyklus:
 - Gehen Sie das Problem mit dem höchsten Risiko an und bestimmen Sie die objektiven und alternativen Lösungen
 - Bewerten Sie die Alternativen und identifizieren Sie die damit verbundenen Risiken und möglichen Lösungen
 - Entwickeln Sie eine Lösung und überprüfen Sie, ob sie akzeptabel ist
 - Planen Sie den nächsten Zyklus



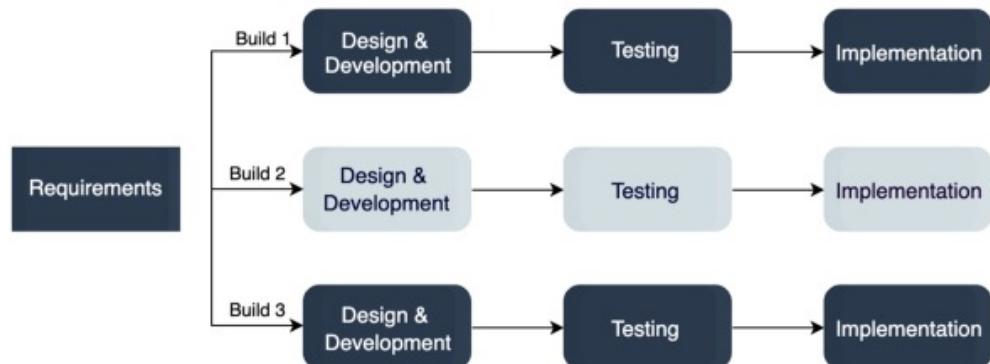
RAD-Modell

- Das Rapid Application Development (RAD-Modell) basiert auf iterativer Entwicklung und Prototyping mit **wenig Planungsaufwand**. Parallel dazu entwickeln Sie Funktionsmodule für eine schnellere Produktlieferung. Es umfasst die folgenden Phasen:
 1. Geschäftsmodellierung
 2. Datenmodellierung
 3. Prozessmodellierung
 4. Anwendungsgenerierung
 5. Testen und Umsatz



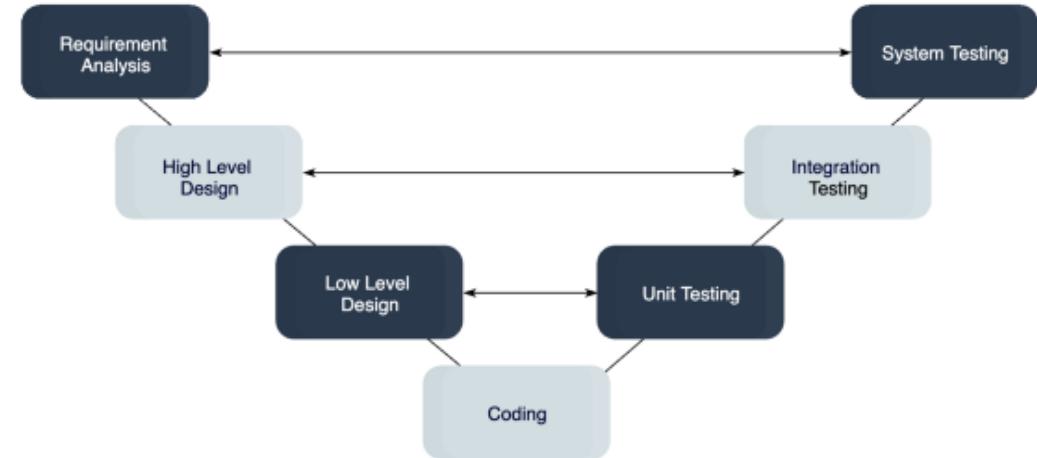
Iteratives Modell

- Das iterative Entwicklungsmodell entwickelt ein System, indem **kleine Teile** aller Merkmale erstellt werden. Dies hilft, den anfänglichen Umfang schnell zu erfüllen und ihn für Feedback freizugeben.
- Im iterativen Modell implementieren Sie zunächst einen kleinen Satz der Softwareanforderungen. Diese werden dann in den sich entwickelnden Versionen **iterativ erweitert**, bis das System fertiggestellt ist. Dieses Prozessmodell beginnt mit einem Teil der Software, die dann implementiert und überprüft wird, um weitere Anforderungen zu ermitteln.



V-Modell

- Das V-Modell (Verifizierungs- und Validierungsmodell) ist eine Erweiterung des Wasserfallmodells. Alle Anforderungen werden zu Beginn gesammelt und können nicht geändert werden. Sie haben für jede Stufe eine entsprechende Testaktivität. Für jede Phase im Entwicklungszyklus gibt es eine **zugehörige Testphase**.
- Die entsprechende Testphase der Entwicklungsphase ist parallel geplant, wie Sie oben sehen können.*
- Das V-Modell ist sehr diszipliniert, leicht zu verstehen und erleichtert das Projektmanagement. Es ist jedoch nicht gut für komplexe Projekte oder Projekte mit unklaren oder sich ändernden Anforderungen. Dies macht das V-Modell zu einer guten Wahl für Software, bei der Ausfallzeiten und Ausfälle nicht akzeptabel sind.



Info

Vorgeschlagen wurde dieses Vorgehen zuerst von dem US-amerikanischen Softwareingenieur Barry Boehm im Jahre 1979 und basiert auf dem Wasserfallmodell. In Deutschland kommt das V-Modell inzwischen sehr häufig bei allen möglichen öffentlich finanzierten Projekten zum Einsatz und ist dabei so etwas, wie ein inoffizieller Standard geworden.



Die einzelnen Phasen des V-Modells

- Das „V“ entsteht deshalb, weil das Modell **die Entwicklungsphasen den korrespondierenden Qualitätssicherungsphasen gegenüberstellt**. Der linke Arm des Buchstabens V enthält die Aufgaben zur Konzipierung und Entwicklung des Systems, der rechte Arm die zugehörigen Maßnahmen zur Qualitätssicherung. In der Mitte dieser beiden Arme, eingebettet zwischen den Entwicklungsphasen und den Qualitätssicherungsphasen, liegt die Implementierung des Produkts. Im Fall eines Softwareprojekts wäre dies die Programmierung der Software.
- Die korrekte Umsetzung der geplanten **Softwarearchitektur** wird durch **Unit Tests** abgefragt. Hier wird im Detail geprüft, ob einzelne Module der Software exakt die geforderten Funktionen erfüllen und auch wirklich die erwarteten Ergebnisse liefern. Idealerweise finden diese Modultests möglichst parallel zur Entwicklung statt, um Fehler zu vermeiden.
- Dem **Systementwurf** stehen die **Integrationstests** gegenüber. Hier wird darauf geprüft, ob die einzelnen Komponenten so miteinander zusammenarbeiten, wie es geplant war – ob beispielsweise alle Abläufe die erwarteten Ergebnisse liefern. Fehlerhafte Ergebnisse können an dieser Stelle u. a. auf Probleme mit Schnittstellen hinweisen.
- Der **Systemtest** prüft, ob die allgemeinen Anforderungen an das System erfüllt wurden, die beim Konzipieren der **Systemarchitektur** festgelegt wurden. Solche Tests finden in der Regel in einer Testumgebung statt, die die realen Bedingungen beim Kunden möglichst exakt nachstellt.
- Der **Anforderungsanalyse** des Gesamtsystems steht am Ende des Projekts die Abnahme des fertigen Produkts gegenüber. Bei der **Endabnahme** prüft der Kunde, ob die Vorgaben im laufenden Betrieb erfüllt werden. In der Regel wird hier nur das Verhalten der Software an der Oberfläche getestet – sprich: das, was der Auftraggeber bei der täglichen Nutzung zu sehen bekommt. Man spricht dabei auch von einem Akzeptanztest.

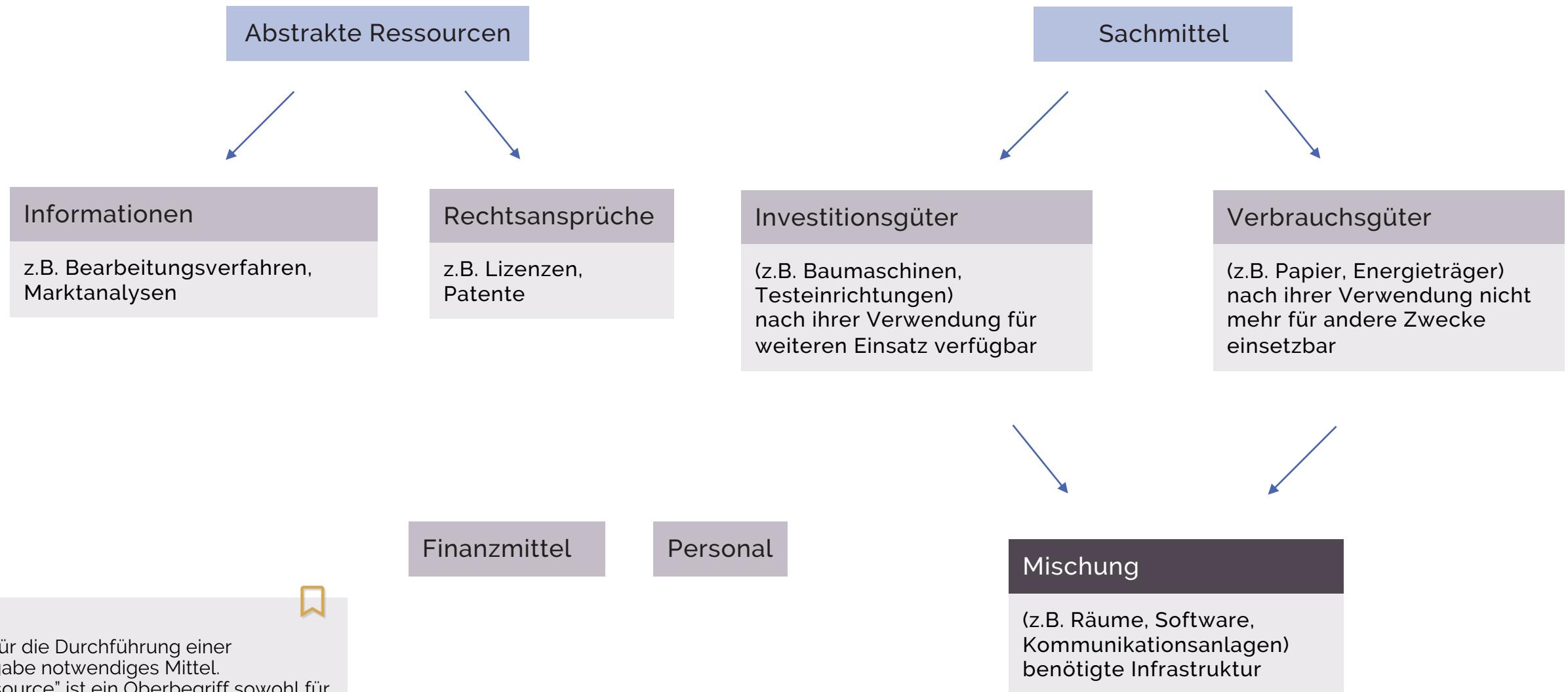
Die Vorteile des V-Modells

- Optimierung der Kommunikation zwischen den Beteiligten durch fest definierte Begriffe und Zuständigkeiten
- Minimierung von Risiken und bessere Planbarkeit durch fest vorgegebene Rollen, Strukturen und Ergebnisse
- Verbesserung der Produktqualität durch fest integrierte Maßnahmen zur Qualitätssicherung
- Kosteneinsparung durch transparente Aufarbeitung des gesamten Produktlebenszyklus
- Insgesamt kann das Modell dabei helfen, **Missverständnisse und unnötige Arbeiten zu vermeiden**. Außerdem sorgt es dafür, dass alle Aufgaben zum richtigen Zeitpunkt und in sinnvoller Reihenfolge erledigt werden und dass möglichst keine Leerlaufzeiten entstehen.

Die Nachteile des V-Modells

- Das Vorgehensmodell ist teils zu simpel, um den Entwicklungsprozess aus Sicht der Entwickler vollständig abzubilden. Der Fokus liegt verstärkt auf dem **Projektmanagement**. Zudem erlaubt die relativ starre Struktur kaum, flexibel auf Änderungen während der Entwicklung zu reagieren, und fördert somit einen vergleichsweise linearen Projektverlauf. Dennoch ist es möglich, mit dem V-Modell agile Entwicklung zu betreiben, wenn das Modell richtig verstanden und genutzt wird.

Was sind Ressourcen



Info

Ein für die Durchführung einer Aufgabe notwendiges Mittel.
Ressource" ist ein Oberbegriff sowohl für Mitarbeiter als auch für Sachmittel oder Finanzmittel.

klassische Methode: Das magische Dreieck



Zusammenhänge zwischen den Ressourcen eines Projektes darstellen



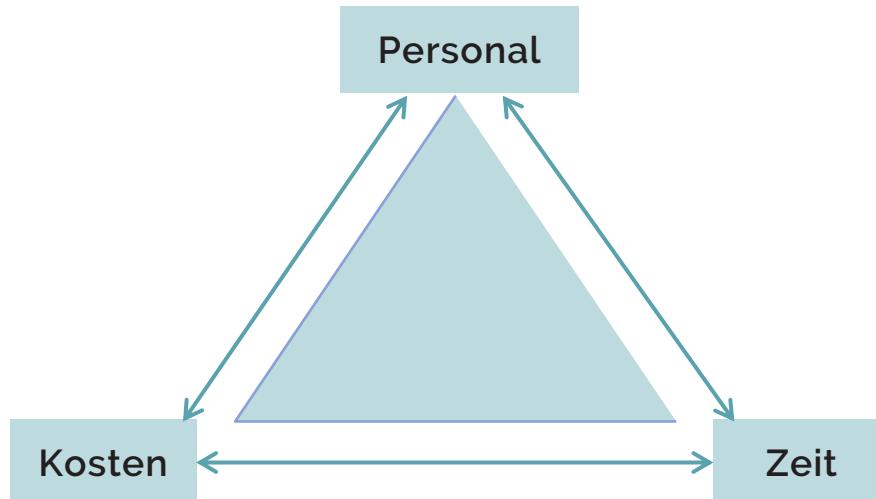
drei Zielgrößen



beeinflussen sich gegenseitig und konkurrieren miteinander



untereinander ausbalanciert



Bsp weitere Ressourcen

Werkzeuge
Maschinenlaufzeiten
Verbrauchsmittel

Info

Ressourcen: alles was zur Planung, Umsetzung und schließlich zum erfolgreichen Abschluss eines Projektes benötigt wird

Der Projektsteckbrief

 In späteren Projektphasen hilft er, sich noch einmal an das ursprüngliche Vorhaben zu erinnern.

 definiert und quantifiziert die Ziele

 Vielzahl an Varianten, unterscheiden sich in Breite und Tiefe der Projektbeschreibung

 Der Projektsteckbrief legt u. a. Anfang und Ende des Projekts fest

 beschreibt Ausgangslage sowie Abhängigkeiten

 gibt das Projektbudget vor

 Budget- & Ressourcenplan

 Projektbeteiligte

 Projektbeschreibung

 Projekttyp

 Risiken

 Projektziele

 Projektnummer

 Auftraggeber

 Termine

 Projektnutzen

 Projektleiter

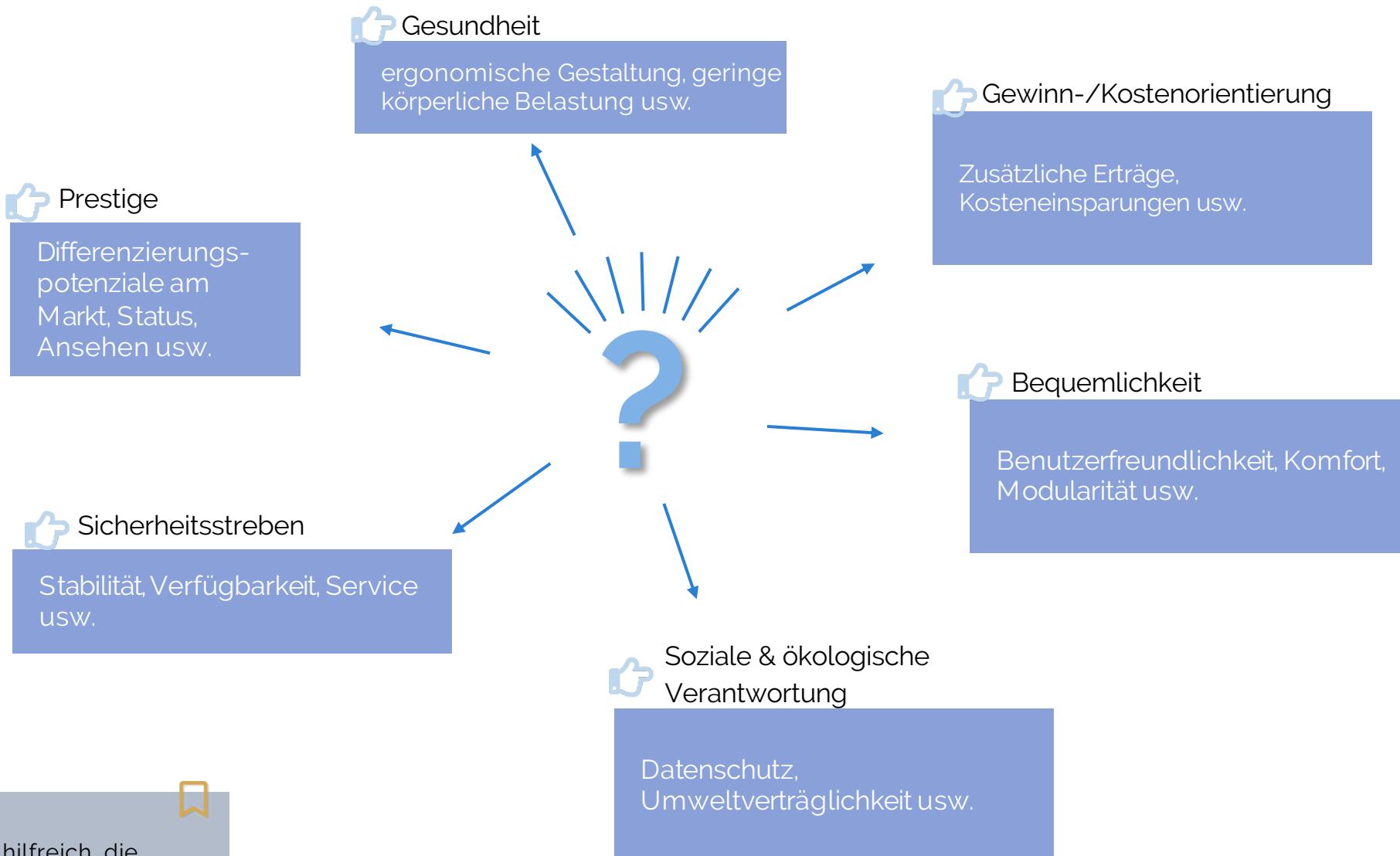
 Projekttitle



Info

Der Projektsteckbrief bleibt bis zum Ende des Projekts möglichst unverändert und wird im Idealfall nur in der Potenzialabschätzung und den notwendigen Investitionen aktualisiert.

Projektsteckbrief: Projektnutzen



Info



Bei der Formulierung ist es hilfreich, die Überleitung durch folgende Prädikate herzustellen: schafft, fördert, stärkt, sorgt für, spart, ermöglicht, befreit von, verringert, schützt vor usw.

Nützliche Tools

Wo stehen wir?

Nützliche Werkzeuge (WO)

- Situationsanalyse
- SWOT-Analyse
- Kontextanalyse (sachlich, zeitlich & sozial)
- Anforderungsanalyse
- Lastenheft

Wie können wir die Ziele erreichen?

Nützliche Werkzeuge (WIE)

- Aufgabenplanung
- Projektstrukturplan
- Risikoplanung

Wieviel wird das kosten?

Nützliche Werkzeuge (WIEVIEL)

- Finanz- & Projektbudgetplanung
- Personal- & Ressourcenplanung
- Sachmittelplanung

Wer ist beteiligt

Nützliche Werkzeuge (WER)

- Kommunikationsmatrix
- Projektorganigramm
- Stakeholder-Analyse

Wann sollen die Ziele erreicht werden?

Nützliche Werkzeuge (WANN)

- Projektterminplanung
- Meilensteinplanung

Was soll erreicht werden?

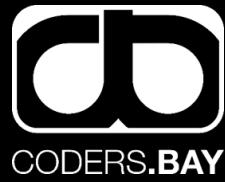
Nützliche Werkzeuge (WAS)

- Gesamt- und Teilzielbildung
- SMART-Methode zur Zielformulierung
- Nicht-Zielformulierung

Warum machen wir das Projekt?

Nützliche Werkzeuge (WARUM)

- Ziel-Break-Down (normative Zielbildung)
- Return on Investment (ROI)
- Business Case



Kreativitätstechniken

Brainstorming

Einfach anzuwenden, keinerlei Schulung der Teilnehmer notwendig

- Gleichberechtigung der Teilnehmer
- Dauer der Sitzung nach Problemerörterung
- Freie Ideenäußerung i Aufnahme bzw. Abwandlung fremder Ideen ist erlaubt
- Verbot von Kritik und Killerphrasen
- Wahl eines Schriftführers
- Aufzeichnung der Ideen durch Schriftführer oder Tonband



Die Bewertung bzw. Auswahl geeigneter Ideen erfolgt später, entweder durch die Teilnehmer oder durch Dritte

Mindmapping

Ziel: **Aufbau eines qualitativen Strukturmodells** auf der Grundlage assoziativer Verknüpfungen oder Über- und Unterordnungsrelationen zwischen verschiedenen Konzepten und / oder deren Bestandteilen

Ma **sammelt** verschiedenen Konzepte oder Konzeptbestandteile und **stellt** relative oder assoziative **Verknüpfungen** graphisch dar
Anhand dieses Strukturmodells lassen sich **Beziehungen** besser **erkennen** oder auch **neue Beziehungen entdecken**

Sowohl für Einzelpersonen geeignet, die schnell einen Überblick bekommen möchten
Auch für eine Gruppe, die ein unübersichtliches Problem strukturieren und unter einem neuen Blickwinkel betrachten möchte





Kommunikation und Feedback

Effiziente Projekt-Kommunikation

- › schafft Klarheit in den Aussagen
- › verkleinert den Abstimmungs-Aufwand
- › reduziert unproduktive Zeit
- › vermeidet Missverständnisse
- › verringert die Demotivation

Sich gut informiert zu fühlen, ist Grundlage für reibungslose Abläufe und Kooperation im Projektmanagement. Ebenso wichtig wie die Information selbst ist dabei die Weitergabe an sich: Wer informiert wen und wann?

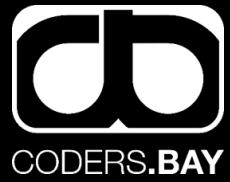
Feedback

Feedback ist keine Wertung und keine Interpretation sondern eine Information an den Sender der Nachricht, was ich gehört, gesehen und verstanden habe

Feedback kann Missverständnisse klären und damit die Zusammenarbeit erleichtern und nachhaltig verbessern

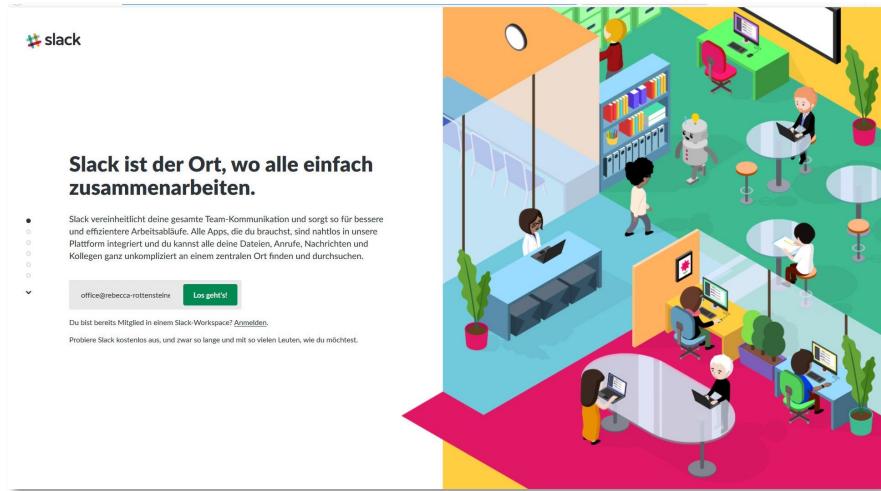
Sowohl zwischen Projektleiter und Mitarbeiter als auch zwischen Mitarbeitern.

Auch hilfreich um mehrere Perspektiven zu haben.



Nützliche Tools

nützliche Tools – Kommunikation

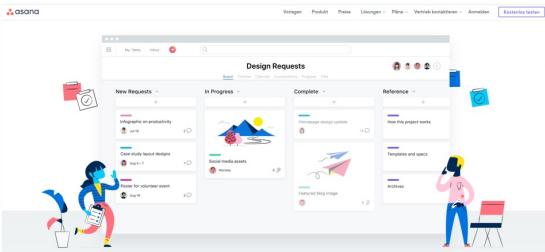


<https://slack.com>

The screenshot shows the Franz application interface. It has a sidebar with a tree view of different messaging channels. The main area displays messages from various platforms: WhatsApp, Facebook Messenger, HipChat, and Telegram. Each platform's messages are color-coded and timestamped. On the right side, there's a sidebar with links for "About", "Download", "Pricing", "Support", "Sign in", and "Create account". Below that is the Franz logo and the tagline "Get back on top of your messaging.". At the bottom, there are download links for Windows and other platforms.

<https://meetfranz.com/>

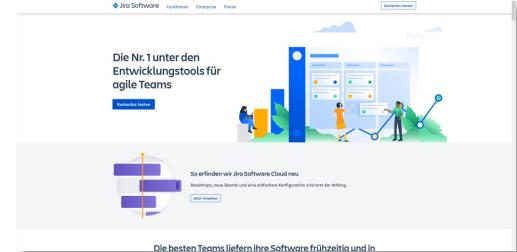
Nützliche Tools - Taskmanagement



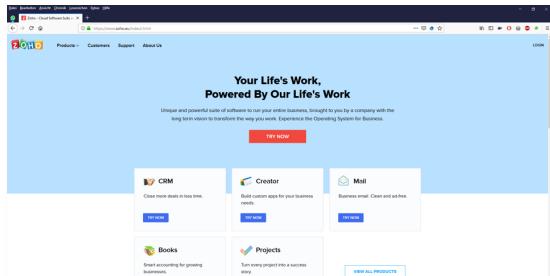
<https://asana.com/de>



<https://wunderlist.com/de>



<https://de.atlassian.com/software/jira>



<https://www.zoho.eu>



<https://www.meistertask.com>



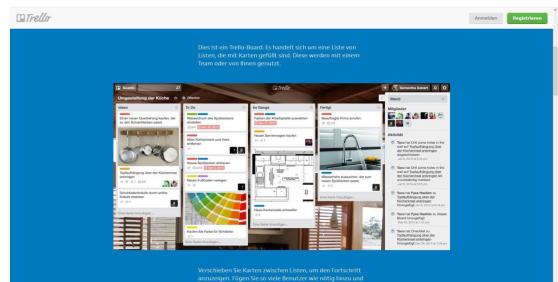
<https://basecamp.com>



<https://www.targetprocess.com>

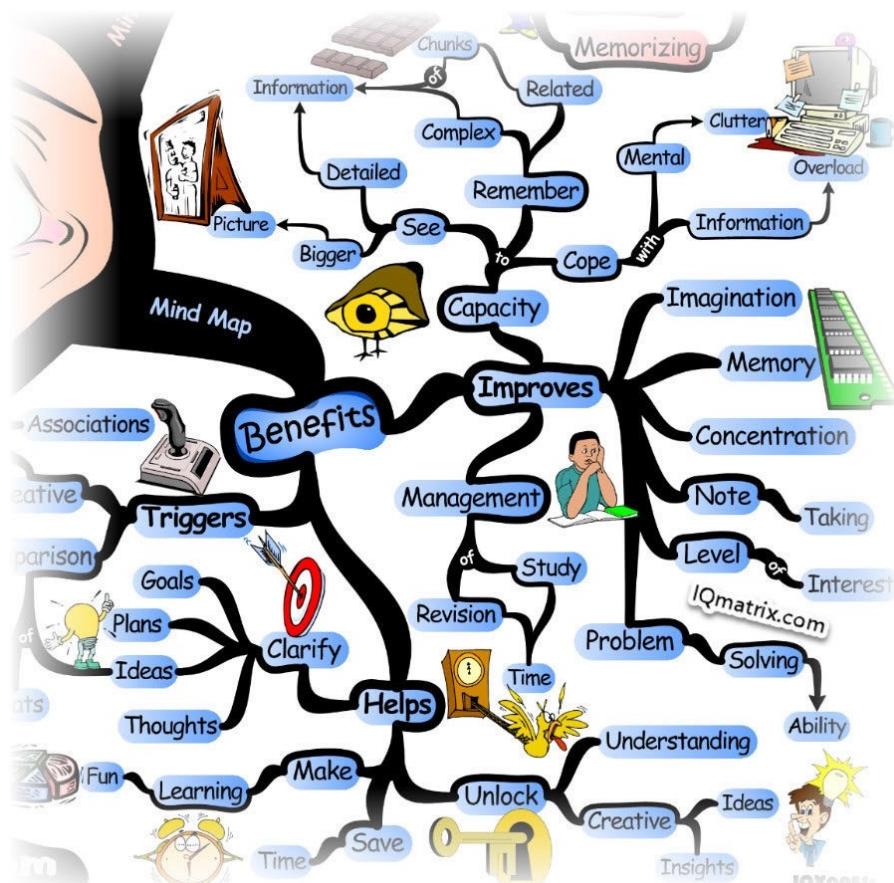


<https://try.wrike.com/de/>



<https://trello.com/>

Nützliche Tools - MindMapping



<https://blog.iqmatrix.com/wp-content/uploads/2009/03/what-are-the-benefits-of-mind-mapping-mind-map.jpg>

Verwendung empfiehlt sich am Beginn einer Problemdefinition, bei der ersten Strukturierung von neuen Ideen



Gedanken frei entfaltet und weiterentwickelt



verschiedene Aspekte eines Themas zu ordnen



fördert die Kreativität

gesamtheitliche Erfassung, Erforschung, Strukturierung und Visualisierung von Problemen & Themenfeldern



Tools im Vergleich
<https://t3n.de/news/mind-mapping-online-tools-568258/>



Einführung in SCRUM

Was ist Scrum?

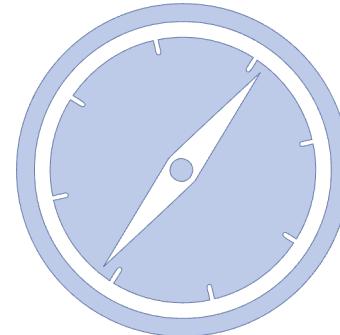
- Der Begriff „Scrum“ bedeutet „Gedränge“. Er entstammt dem Rugby-Sport, wo er das Zusammenstehen des Teams beschreibt.
- Scrum wird erstmalig 1986 im Artikel „The New New Product Development Game“ von Ikujiro Nonaka und H. Takeuchi erwähnt.
- Darauf basierend entwickeln Ken Schwaber, Jeff Sutherland, Mike Cohn u. a. im Lauf der 90er Jahre eine Methodik, die bei der Softwareentwicklung eingesetzt wird.
- Im Fokus steht das Liefern von funktionierender Software in kurzen Iterationen, damit möglichst schnell Feedback eingeholt und auf Änderungen reagiert werden kann.
- Mit wenigen einfachen Regeln und Rollen ermöglicht Scrum einen schnellen Einstieg in die Anwendung der Methodik
- 2001 wird das „Agile Manifest“ veröffentlicht, welches die wichtigsten agilen Prinzipien und Grundwerte zusammenfasst.

Einführung & Grundlagen von Scrum

empirisches Expositionsmodell

für Menschen, die das Scrum-Modell anwenden, Wissen aus der Praxis gewonnen haben und auf der Grundlage dieser Erfahrungen Entscheidungen treffen

**Scrum ist keine Methode
sondern eine neue Denkweise**



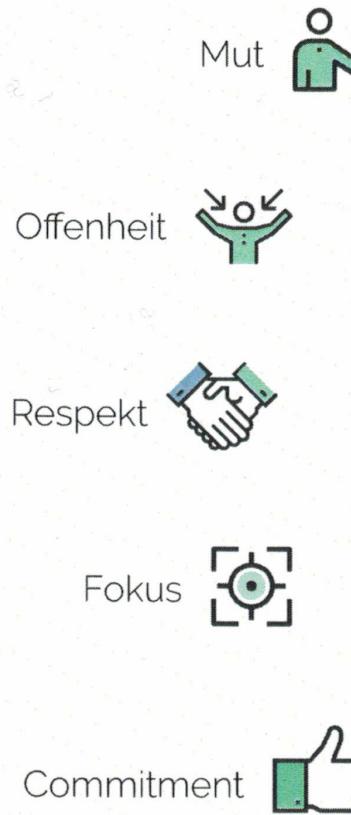
**Scrum stellt Struktur bereit, die
zu größerer Effizienz und
schnelleren Ergebnissen führt**

**Man konzentriert sich auf das,
was heute getan werden kann,
mit dem Ziel die zukünftige
Arbeit in überschaubare Teile
aufzuteilen.**

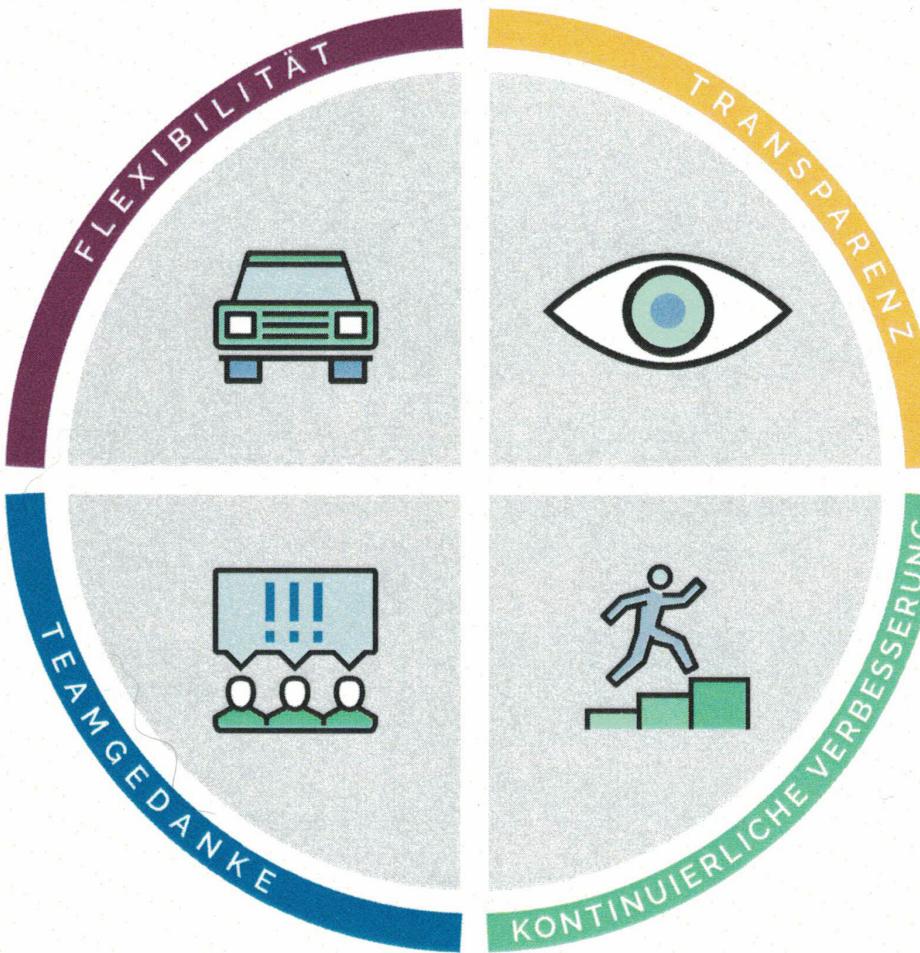
**Möglichkeit Projekte zu
organisieren**

Werte und Vorteile

Agile Werte



Vorteile von Scrum

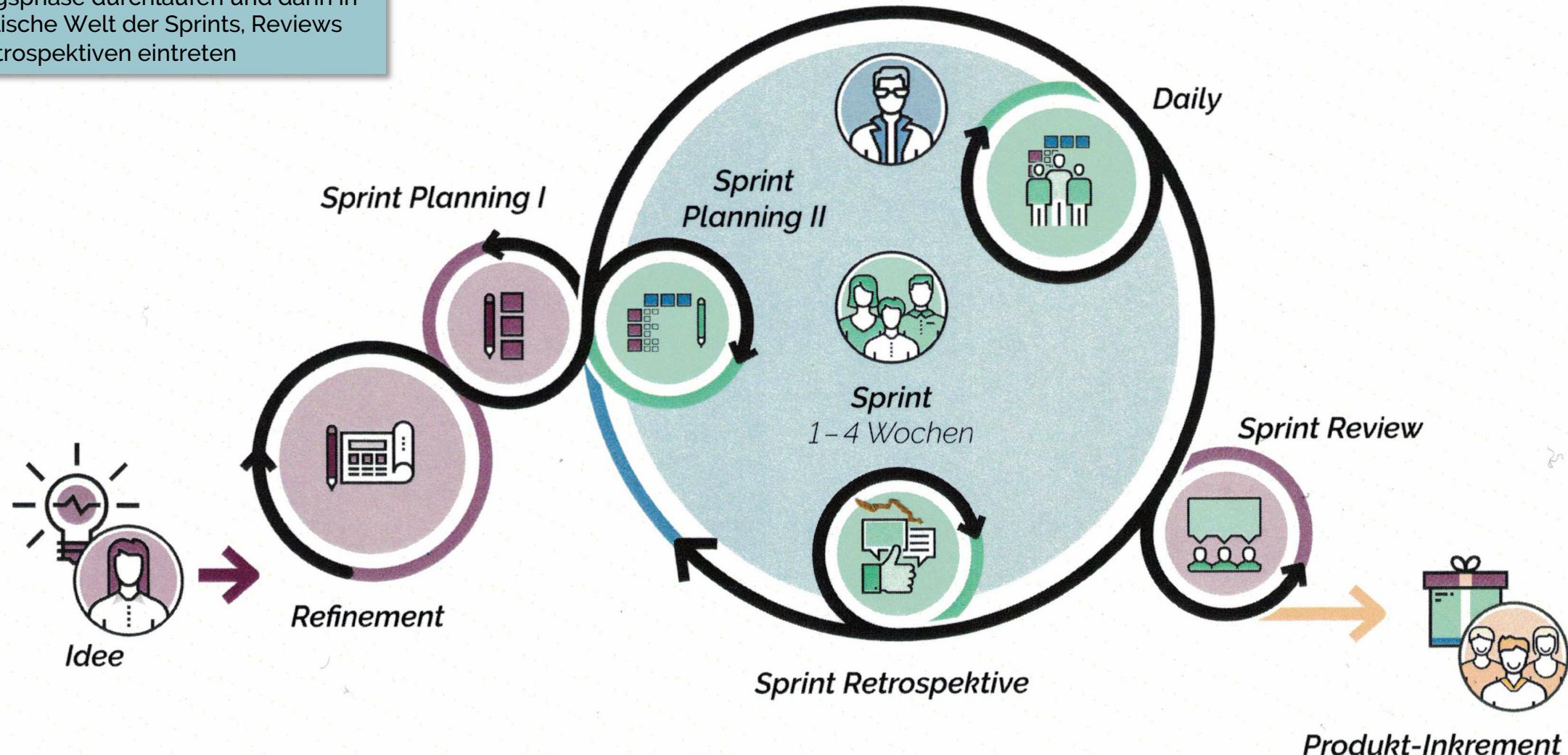


Agile Prinzipien

-
- Priorisierung (Icon: List with upward arrow)
 - Selbstorganisation (Icon: Group of people with upward arrows)
 - Empirismus (Icon: Person with circular arrows)
 - Emergent (Icon: Bar chart)
 - Zusammenarbeit (Icon: Two hands holding a lightbulb)
 - Time-Boxed (Icon: Clock)

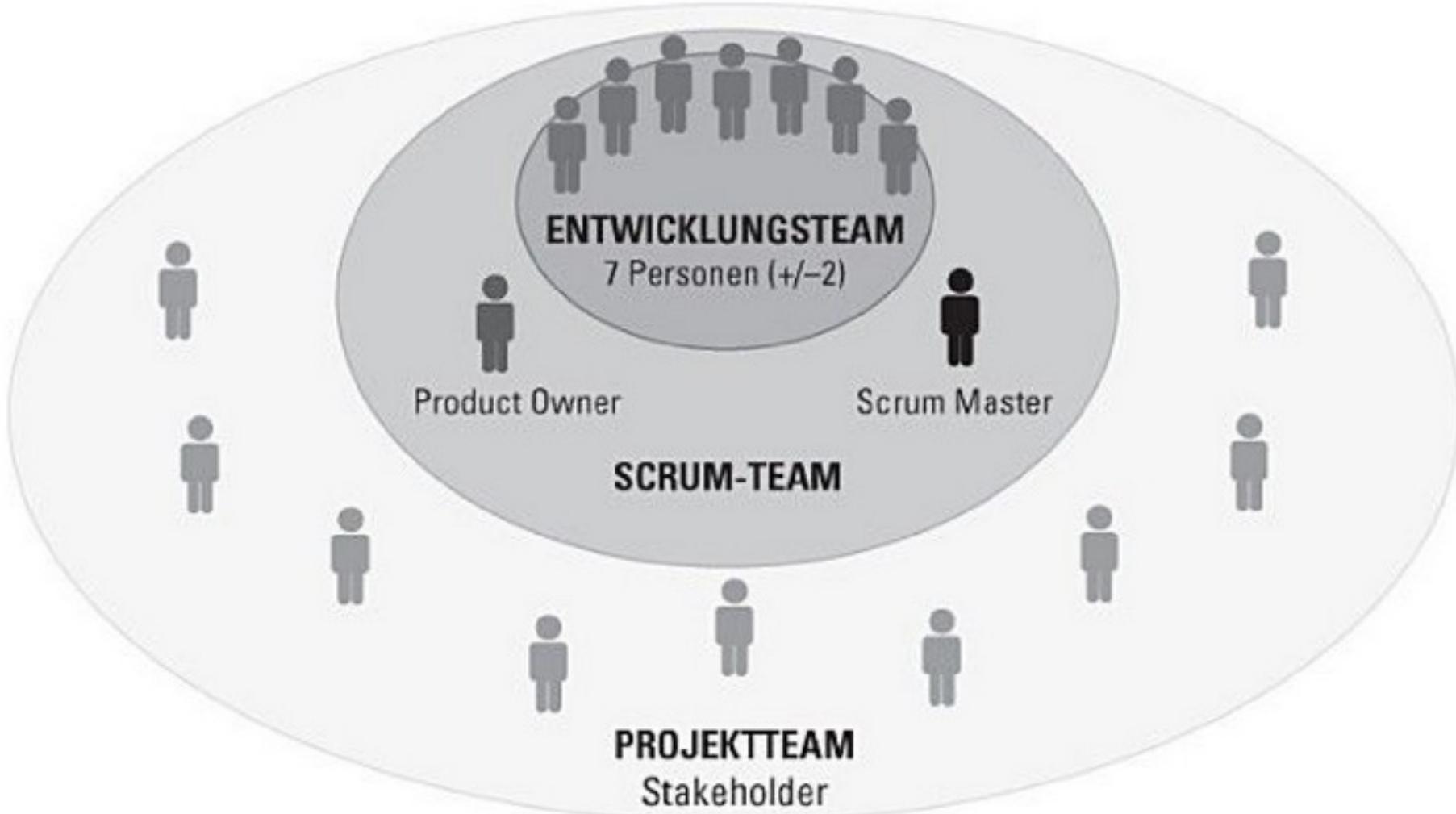
Übersicht Scrum Ansatz

mit der Produktvision beginnen, die Planungsphase durchlaufen und dann in die zyklische Welt der Sprints, Reviews und Retrospektiven eintreten



Mit "Sprint" bezeichnet Scrum den wertschöpfenden Projektprozess, bei dem das Entwicklungsteam innerhalb eines Vorgangs mit fixierter Dauer Anforderungen aus dem Sprint Backlog in ein Inkrement umsetzt

Das Scrum Team



Gängige Rollen außerhalb von Scrum

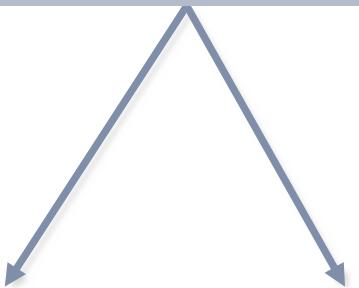
Stakeholder

Menschen, die das Projekt beeinflussen oder von ihm betroffen sind. Interne Stakeholder sind innerhalb des Unternehmens oder der Organisation.

Externe Stakeholder können Investoren oder Nutzer sein.

Scrum-Mentoren

auch Scrum-Coach, arbeitet mit Team zusammen um Scrum-Praxis weiterzuentwickeln.



Stakeholder auf geschäftlicher Seite

Bsp.: Kunde, Verkaufsteam,
Marketing

Product Owner ist
verantwortlich

Stakeholder auf nicht geschäftlicher Seite

Bsp.: Lieferanten, Auftragnehmer

Scrum Master ist
verantwortlich

Product Owner

Der Product Owner hat die Vision, WAS umgesetzt werden soll

Eigenschaften:
Visionär, Erwartungsmanager, Entscheider



Aufgabe



- Bestimmt WAS umgesetzt wird
- Verantwortlich für die Produktvision
- Ziel die Stakeholder glücklich zu machen
- Betreibt Stakeholder-Management
- Hält das Product Backlog priorisiert
- Stellt sicher, dass dem Entwicklungsteam alle notwendigen Informationen zur Umsetzung zur Verfügung stehen

Der Scrum Master

Der Scrum Master hilft dem Team und der Organisation dabei, auf der Basis der agilen Werte zu arbeiten und sich kontinuierlich zu verbessern.

Eigenschaften:

Coach, Mentor, Diener, Beschützer des Entwicklungsteams – mit dem Ziel, generierten Wert zu maximieren

	Aufgabe	
	<ul style="list-style-type: none">- Verantwortlich für das Verständnis und die Durchführung von Scrum- Coach für das Entwicklungsteam- Berater für den Product Owner- Moderator- Vertritt die agilen Werte innerhalb der Organisation- Beseitigt Hindernisse, die das Entwicklungsteam aufhalten und von diesem nicht selbst gelöst werden können	

Entwicklungsteam

Das Entwicklungsteam verfügt über die Fähigkeiten, die notwendig sind, um die Produktvision des Product Owners zu verwirklichen

Eigenschaften:
Interdisziplinär, Umsetzer,
selbstorganisierend



Aufgabe



- Bestimmt **WIE** das Produkt umgesetzt wird
- Organisiert sich selbst
- Arbeitet eigenverantwortlich
- Ist interdisziplinär aufgestellt und verfügt über alle benötigten Fähigkeiten, um ein Produkt-Inkrement zu erstellen
- Entscheidet, wie viele Features im Sprint umgesetzt werden (Pull-Prinzip)

Stakeholder

Stakeholder sind am Produkt interessierte oder von diesem betroffene Personen, wie z.B. Geschäftsführung, Endkunden oder Dienstleister

Eigenschaften:
Interessiert am Produkt, häufig Geldgeber

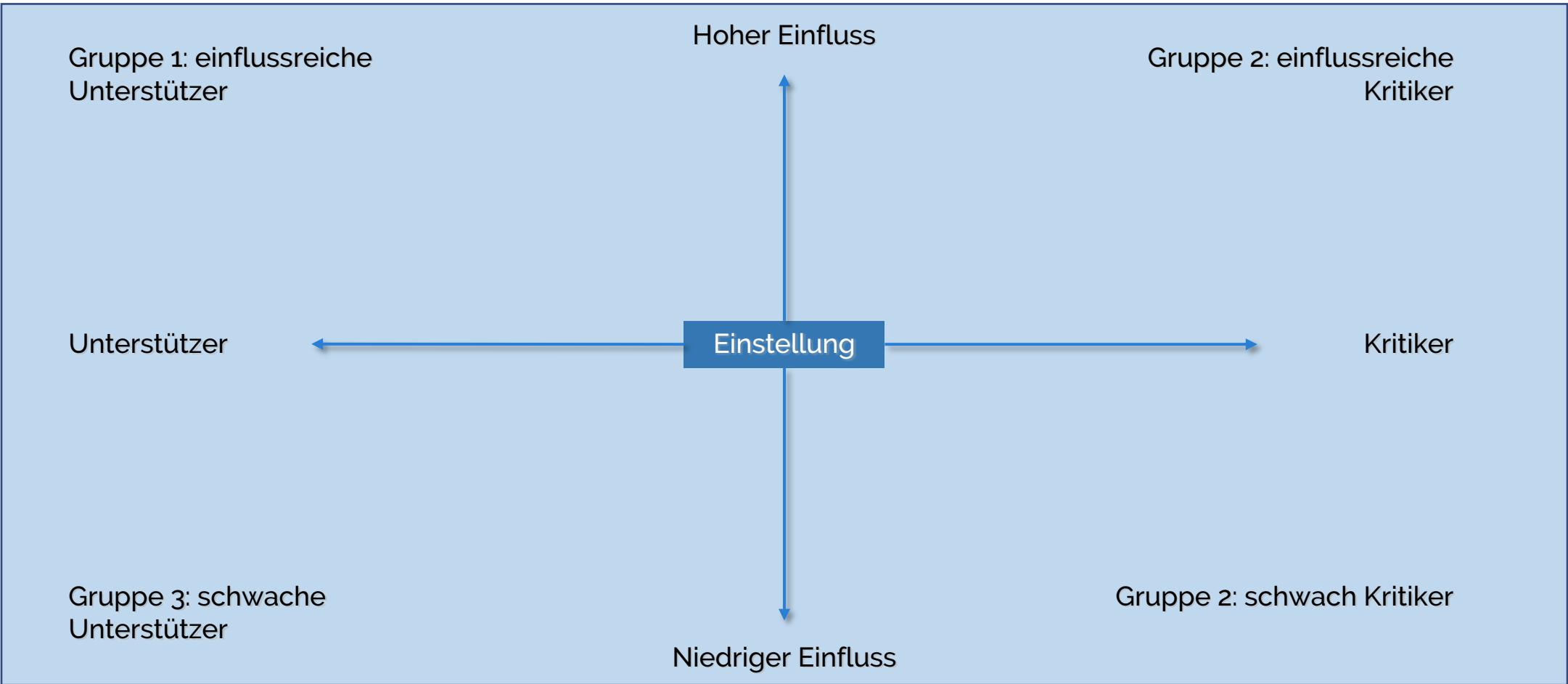


Aufgabe



- Kommunizieren ihre Produkt-Wünsche an den Product Owner
- Geben Feedback an den Product Owner und das Entwicklungsteam weiter und helfen dadurch dem Product Owner zu beurteilen, welche Features den höchsten Business-Wert haben

Konkrete Stakeholder-Analyse



Stakeholder Feedback

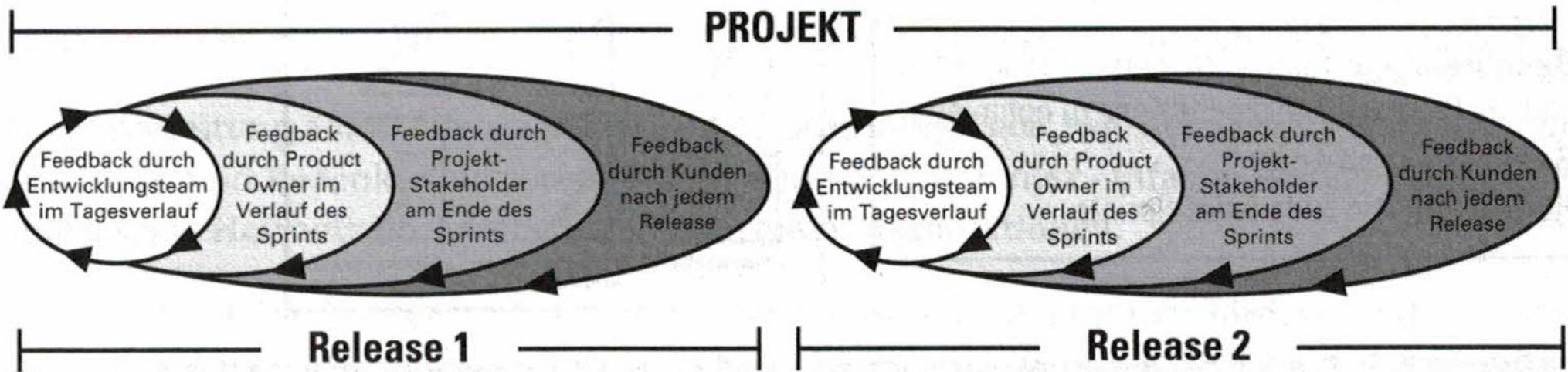


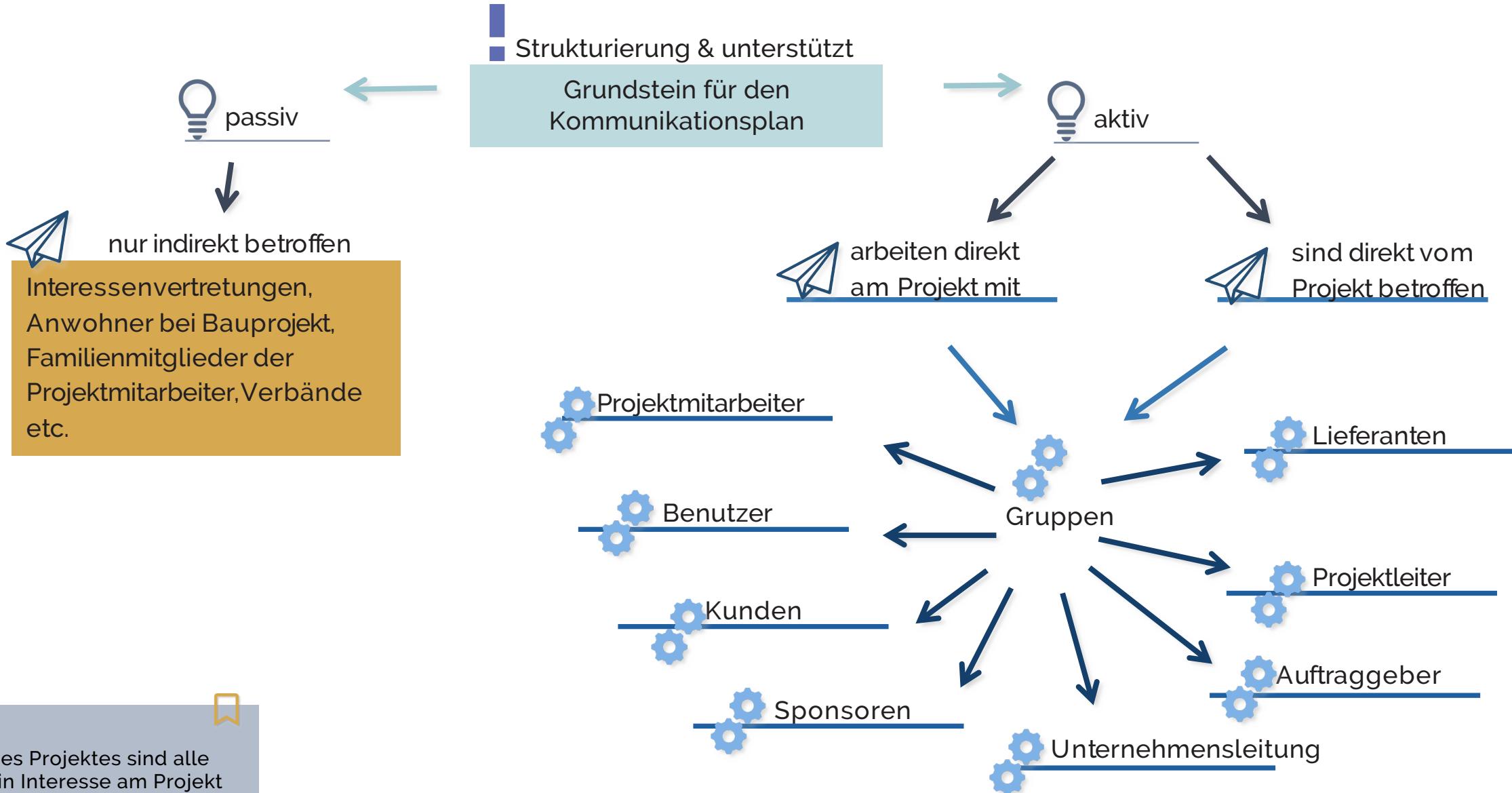
Abbildung 6.4: In einem typischen Scrum-Projekt gibt es mehrere Feedback-Ebenen.



Info

Feedback spielt in Scrum an unterschiedlichen Stellen eine wichtige Rolle. Jedes Mal, wenn Feedback eintrifft, wird es in das Produkt-Backlog eingespeist und in den Sprint-Planungssitzungen berücksichtigt – echte Überprüfung und Anpassung.

weitere Stakeholder Definition



Die Meetings im Überblick

Refinement (Verfeinerung)

Zeit: 60 – 120 min pro Woche

Das Product Backlog wird aktualisiert, priorisiert und geschätzt.

Sprint Planning I

Zeit: 60 min pro Sprintwoche

Der Product Owner stellt seine wichtigsten (d.h. am höchsten priorisierten) Features aus dem Product Backlog vor, die er gerne im nächsten Sprint umgesetzt hätte. Das Entwicklungsteam entscheidet, wie viel es davon in diesem Sprint umsetzen kann.

Sprint Planning II

Zeit: 60 min pro Sprintwoche

Das Entwicklungsteam plant den nächsten Sprint und füllt das Sprint Backlog mit den zu erledigenden Tasks, um die im Sprint Spanning I festgelegten Features erfolgreich umzusetzen

Daily Scrum

Zeit: 15 min pro Tag

Das Entwicklungsteam trifft sich am Sprint Backlog, um den aktuellen Tag zu planen. Jedes Team beantwortet 3 Fragen:

1. Was habe ich gestern für das Team gemacht?
2. Was plane ich heute für das Team zu machen
3. Was behindert mich bei meiner Arbeit, um für das Team Wert zu schaffen?

Review

Zeit: 60 min pro Sprintwoche

Das Entwicklungsteam stellt dem Product Owner und den Stakeholdern die umgesetzten Features vor.

Retrospektive

Zeit: 45 min pro Sprintwoche

Das Team lässt den letzten Sprint Revue passieren und evaluiert, wie es in Zukunft noch besser zusammenarbeiten kann (kontinuierliche Verbesserung).

Sprint Planning I



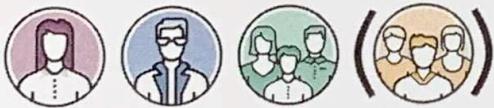
Der Product Owner stellt seine wichtigsten (d.h. am höchsten priorisierten) Features aus dem Product Backlog vor, die er gerne im nächsten Sprint umgesetzt hätte.

→ **WAS** soll gemacht werden

Zeit:

60 min pro Sprintwoche

Teilnehmer:



Wann:

Beginn des Sprints



Das Entwicklungsteam entscheidet, wie viel es in diesem Sprint umsetzen kann!

Sprint Planning II

Das Entwicklungsteam plant den Sprint. → **WIE** wollen wir diese Features umsetzen?

Zeit:
60 min pro Sprintwoche

Teilnehmer:
  

Wann:
am Anfang des Sprints,
direkt nach dem
Sprint Planning I

User Stories nach Business-Wert sortiert.
→ Verwendung der rechteckigen Post It's®

Alle **To-Do-Tasks** nach dem Sprint Planning II,
also alle geschriebenen Tasks.
→ Verwendung der quadratischen Post It's® Super Sticky

User Story **To-Do** **WIP** **Impediments** **Done**

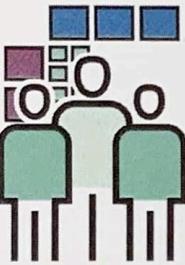
WIP (Work in Progress):
Tasks, die zwischen den Daily Scrums bearbeitet werden.

Hier hängen Zettel mit **Impediments**, die das Entwicklungsteam bei der Erfüllung ihrer Arbeit behindern.

Alle **erledigten Tasks**.

 Um in einen Arbeitsflow zu kommen, hat es sich gezeigt, dass die Aufteilung einiger weniger großer Tasks in viele kleine Tasks hilfreich ist.

Daily Scrum



Zeit: 15 min pro Tag

(Time-boxed! Nach 15 min ist das Meeting vorbei – dies hilft dem Entwicklungsteam, sich zu fokussieren)

Teilnehmer:



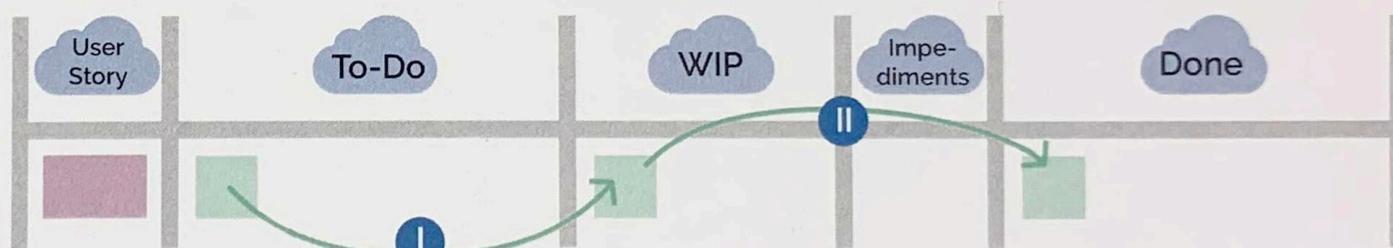
Wann: idealerweise am Morgen, jeden Tag zur gleichen Zeit (das Entwicklungsteam ist selbstorganisierend und entscheidet dies)

Das Entwicklungsteam trifft sich am Taskboard, um den aktuellen Tag zu planen. Jedes Teammitglied beantwortet 3 Fragen:

1 Was habe ich gestern für das Team gemacht?

2 Was plane ich heute für das Team zu machen?

3 Was behindert mich an meiner Arbeit um für das Team Wert zu schaffen?



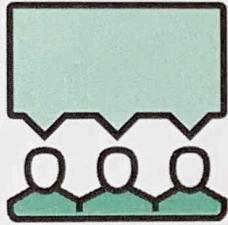
I Wenn heute daran gearbeitet wird.

II Wenn die Aufgabe erledigt ist.



Falls Behinderungen auftreten, die niemand aus dem Entwicklungsteam lösen kann, werden sie unter *Impediments* an das Taskboard gehängt, damit sich der Scrum Master um deren Behebung kümmert.

Review



Zeit:

60 min pro Sprintwoche

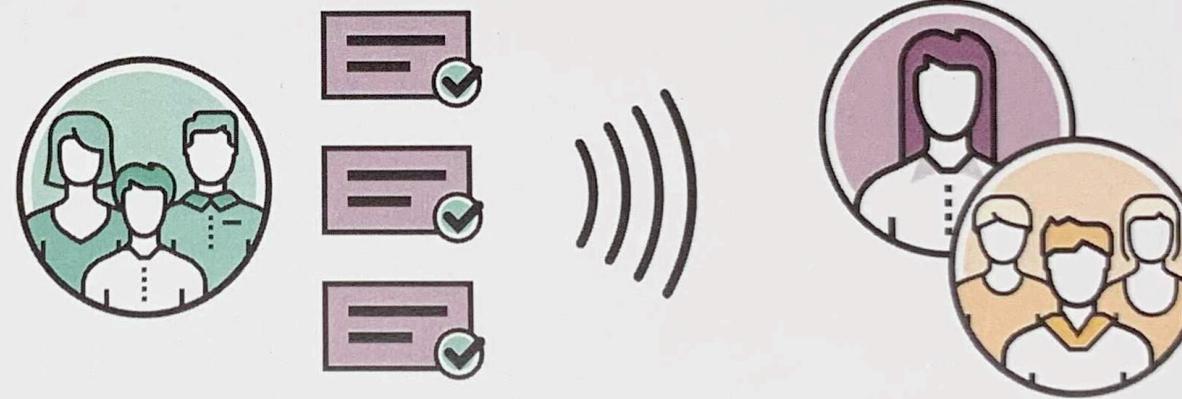
Teilnehmer:



Wann:

am Ende des Sprints

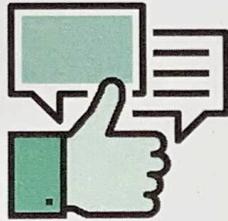
Im Review steht das **Feedback des Product Owners** und der **Stakeholder** im Vordergrund. Dabei stellt das Entwicklungsteam dem Product Owner und den Stakeholdern alle in diesem Sprint fertiggestellten Features (User Stories) vor.



Das Entwicklungsteam ist **selbst-organisierend** und entscheidet daher, wer aus dem Entwicklungsteam präsentiert und wie die Vorführung abläuft.

Das Meeting ist ein **offenes Meeting**. Der Product Owner kann Stakeholder einladen, die Interesse an den in diesem Sprint umgesetzten Features haben.

Retrospektive



Zeit:

45 min pro Sprintwoche

Teilnehmer:



(Das Entwicklungsteam entscheidet, ob der PO an dem Termin teilnehmen soll)

Wann:

am Ende des Sprints nach dem Review-Meeting

Das Team reflektiert mit der Unterstützung des Scrum Masters den letzten Sprint und erarbeitet, wie es in Zukunft noch besser zusammenarbeiten kann

Start to...	Stop to...	Continue to...
Three blue rectangular boxes representing items to start with.	Three blue rectangular boxes representing items to stop doing.	Three blue rectangular boxes representing items to continue doing.



In diesem Meeting passieren sehr wichtige Dinge im Team: Mit jedem Sprint verbessert sich das Team, so dass es sich schrittweise zu einem High-Performance-Team entwickelt!

Beispiele und Ideen rund um das Thema Retrospektive:

www.plans-for-retrospectives.com



Roadmap of Value – Der Wertschöpfungsplan

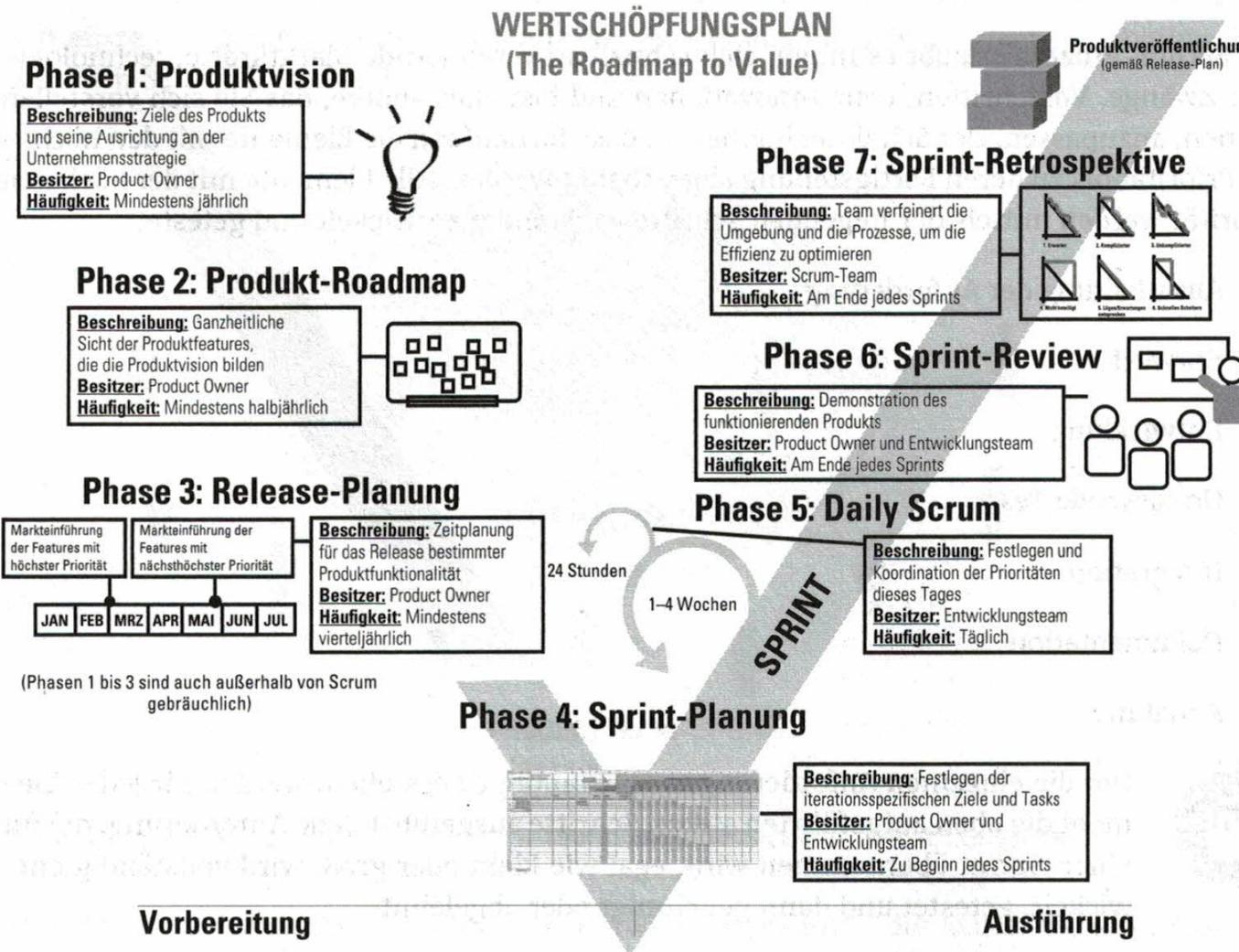


Abbildung 1.1: Die sieben Phasen der Roadmap to Value

mit der Produktvision beginnen, die Planungsphase durchlaufen und dann in die zyklische Welt der Sprints, Reviews und Retrospektiven eintreten

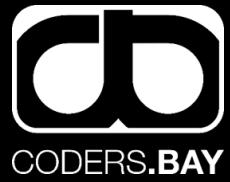
Phasen helfen zu erkennen was sie erreichen wollen

7 Phasen: angefangen bei der Visionsphase bis zur Aufgabenebene und wieder zurück

ist eine Ansammlung von Scrum und geprüften gängigen Praktiken

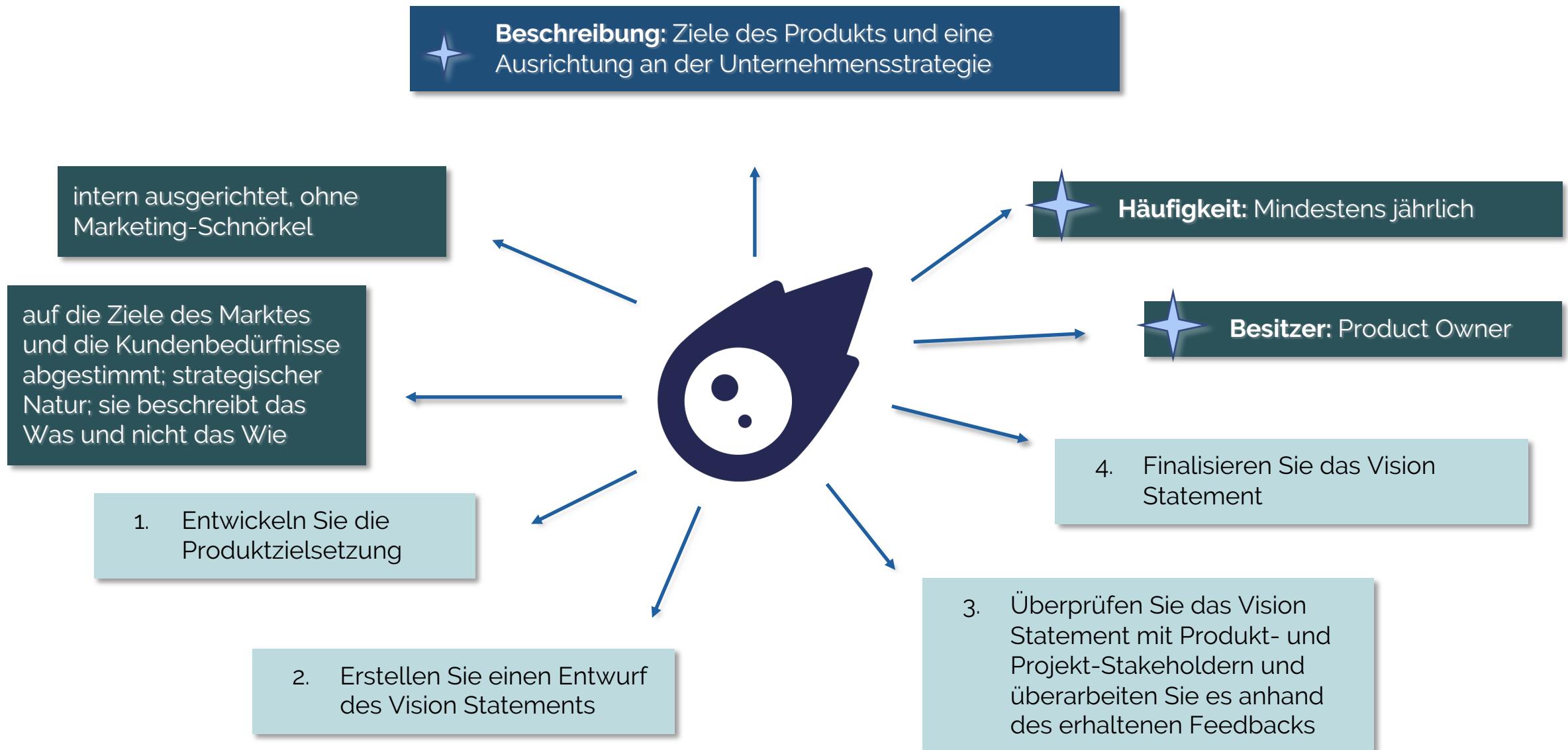


Information
<https://www.dummies.com/careers/project-management/the-agile-roadmap-to-value/>



Produktvision

Phase 1 - Produktvision



Phase 1

Produktzielsetzung entwickeln



Aspekte

- **Kunde:** Wer wird das Produkt verwenden? Mehrere Antwortmöglichkeiten
- **Wichtigste Produktziele:** Wie profitiert das Unternehmen, dass das Produkt erstellt, vom Produkt? Zu den Zielen können Vorteile für eine bestimmte Abteilung in Ihrem Unternehmen gehören, wie den Kundensupport oder die Marketingabteilung, oder auch Vorteile für die Firma als Ganzes. Welche spezifischen Unternehmensstrategien unterstützt das Produkt?
- **Notwendigkeit:** Warum benötigt der Kunde das Produkt? Welche Features sind für den Kunden entscheidend?
- **Wettbewerb:** Wie steht das Produkt im Vergleich mit ähnlichen Produkten dar?
- **Primäres Unterscheidungskriterium:** Was unterscheidet dieses Produkt vom Status quo oder dem der Konkurrenz oder von beidem

Phase 1 – Entwurf des Vision Statements erstellen

Für <Zielkunden>

die <Bedürfnisse>

ist <Produktname>

ein/eine <Produktkategorie>

mit dem/der <Produktnutzen, zwingender Grund für die Anschaffung>

im Unterschied zu <primären Alternativen von Wettbewerbern>

ist unser Produkt <Wettbewerbsvorteil, Nutzenversprechen>

Wir empfehlen, diese Schlussfolgerung hinzuzufügen:

was unsere Strategie zur <Unternehmensstrategie> unterstützt

Phase 1 – Beispiel

Für Hausbesitzer, **die** einen kontinuierlichen Warmwasserfluss und eine optimale Energienutzung wünschen, **ist** der Acme Tankless Water Heater ein bei Bedarf arbeitender und an der Verbrauchsstelle installierter Durchlauferhitzer, **der** das Wasser effizient erwärmt, während Sie es nutzen. **Im Unterschied** zu Boilern, die einen Tank aufweisen, **bietet unser Produkt** eine kontinuierlicher Warmwasserversorgung bei konstanten Temperaturen und niedrigeren Betriebskosten bei einem Wirkungsgrad von 94 Prozent, **was unsere Strategie unterstützt**, für Nachhaltigkeit zu sorgen, indem wir heute die Vergeudung natürlicher Ressourcen reduzieren.

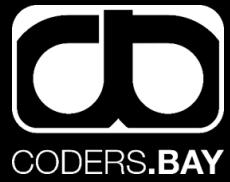
Phase 1

Überprüfen und überarbeiten



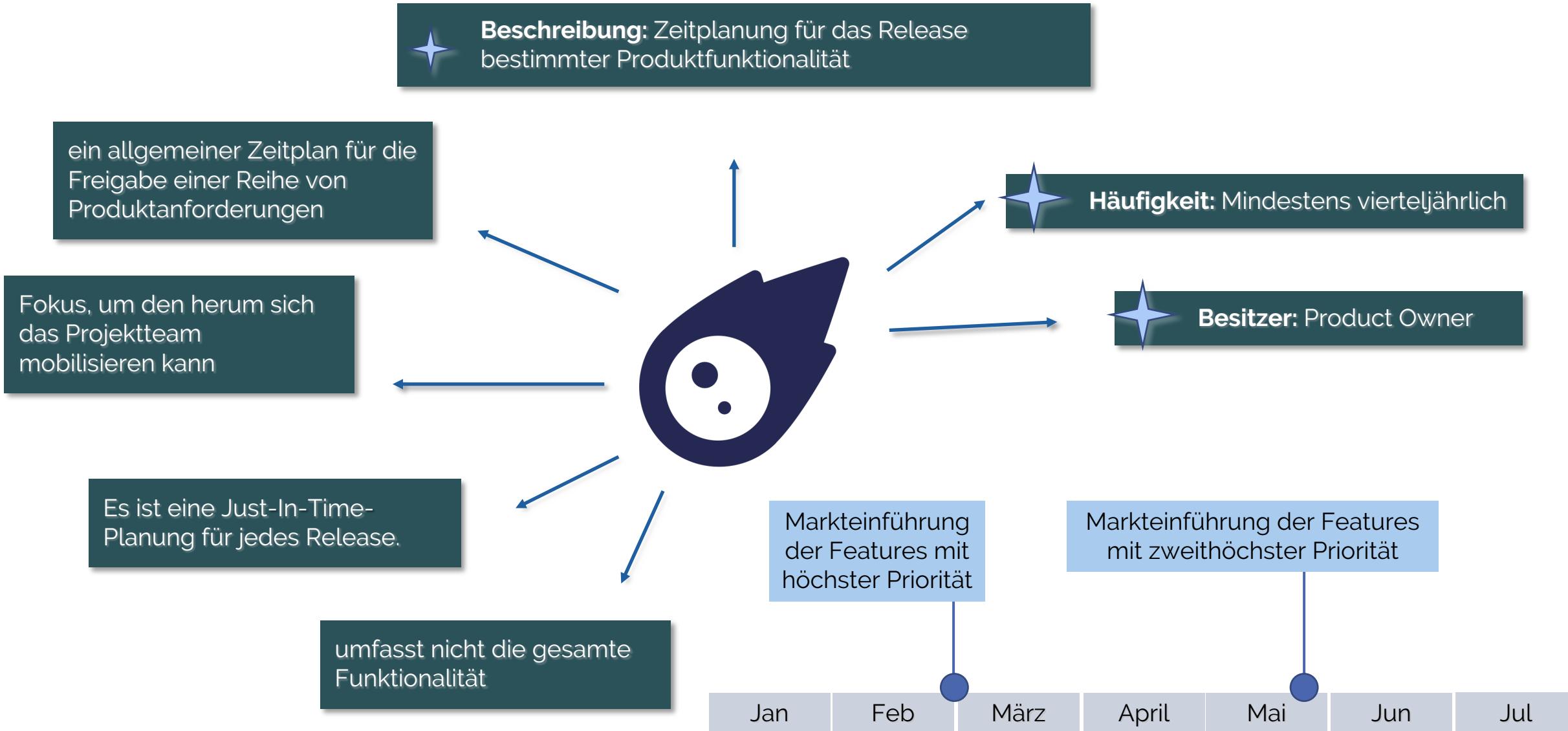
Qualitätscheckliste

- Ist das Vision Statement **deutlich, fokussiert** und für eine **interne Zielgruppe** geschrieben?
- Enthält das Statement eine **überzeugende Beschreibung**, wie das Produkt die Kundenwünsche erfüllt?
- Beschreibt die Vision das **bestmögliche Ergebnis**?
- Ist das **Geschäftsziel spezifisch genug**, damit das Ziel erreicht werden kann?
- Liefert das Statement **Werte**, die mit den Unternehmensstrategien und Zielen übereinstimmen?
- Ist das Product Vision Statement **überzeugend**?
- Ist die Vision **prägnant**?



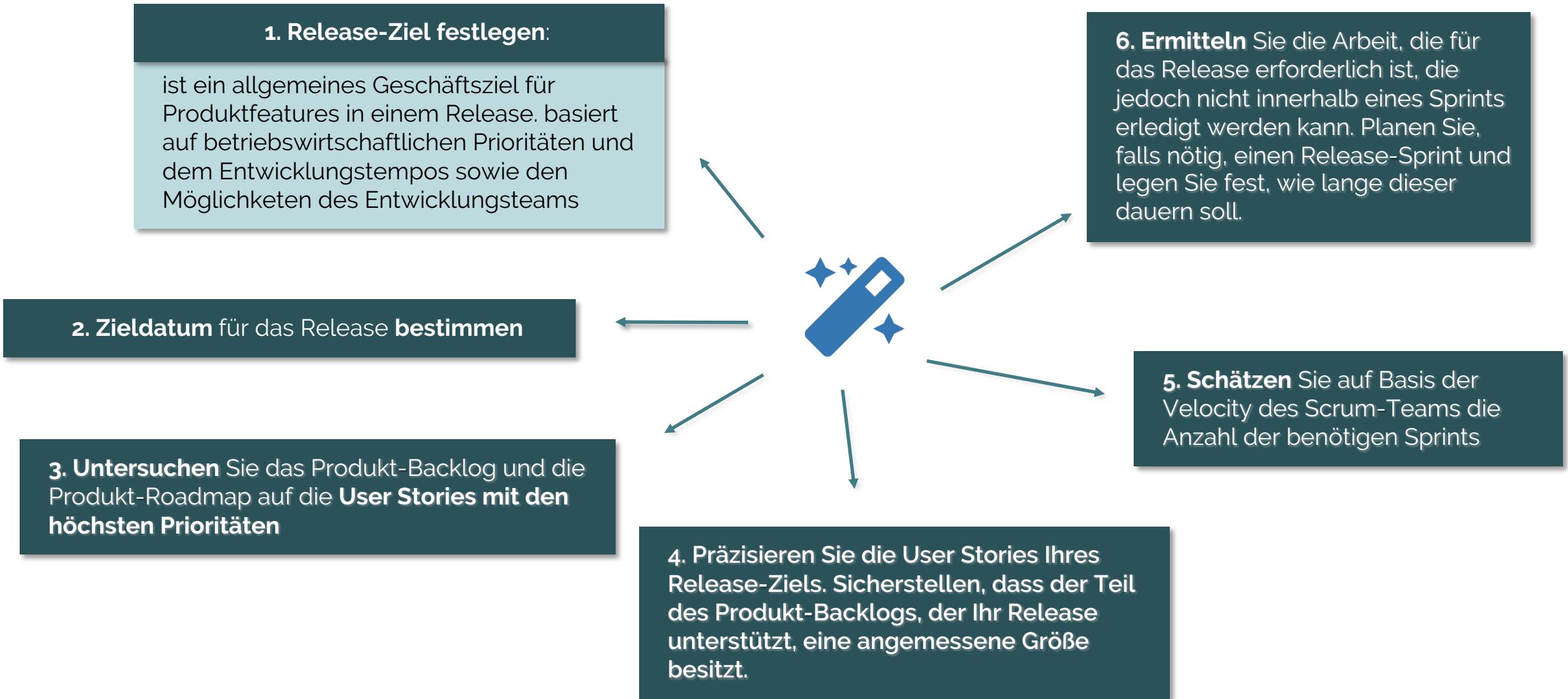
Release Planung

Phase 3 – Release-Planung



Release-Plan in der Praxis

6 Schritte

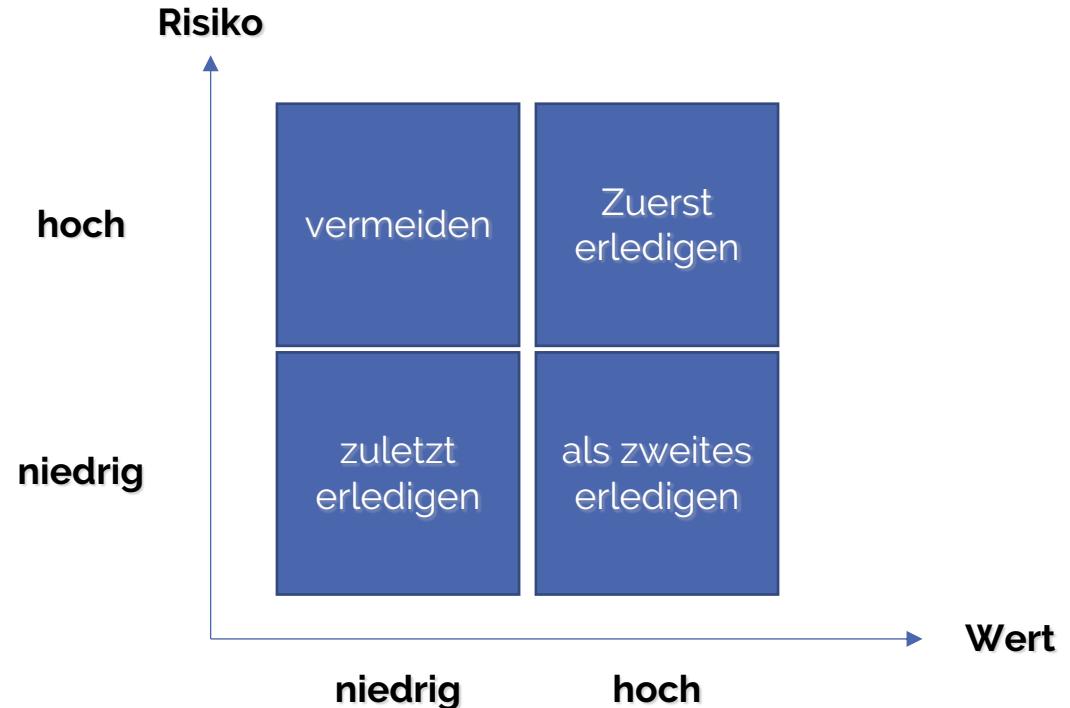


Phase 3 & 4: Release & Sprint Planung

Release	Release	Release	Release
Release1: 15.8.2019 Feature1, Feature2,	Release2: 15.11.2019 Feature3,	Relese3: 15.2.2020 Feature4, Feature5,	Release4: 15.5.2020 Feature6; Feature7,
Sprint1	Sprint4	Sprint7	Sprint10
User Story Datenbank für Profile erstellen	Sprint5	Sprint8	Sprint11
User Story Feature1 Eingabefeld erstellen	Sprint6	Sprint9	Sprint12
User Story Persönliche Daten editieren	+		
User Story Feature2 Eingabefeld erstellen	+		
Sprint2			
Sprint3			
Sprint			
User Story			
User Story Speicherfunktion			

Release-Ziele

Jeder Release-Plan hat ein **übergeordnetes Geschäftsziel**, das als **Release-Ziel** bezeichnet wird. Dieses Ziel wird vom Product Owner erstellt und knüpft direkt an das Endziel des Produktes an, das im Product Vision Statement definiert ist.

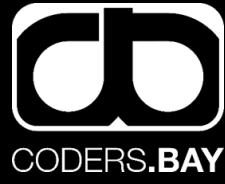


Backlog-Prioritätsmatrix: hilft, die Prioritäten der Storys zu bestimmen



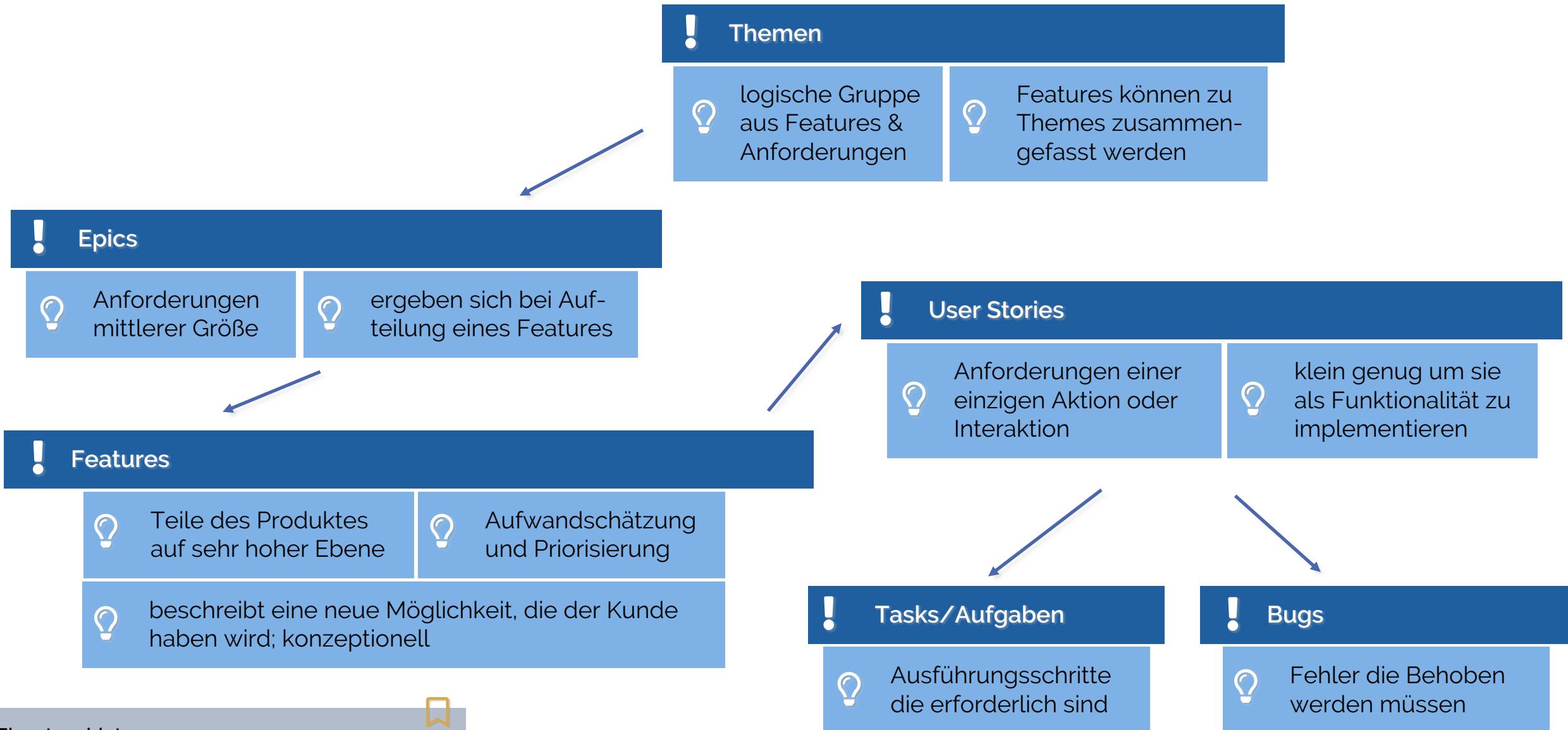
Info

- mittelfristige Grenzen für bestimmte Funktionalitäten
- beschleunigt Priorisierungsprozess



Zerlegung der Anforderungen

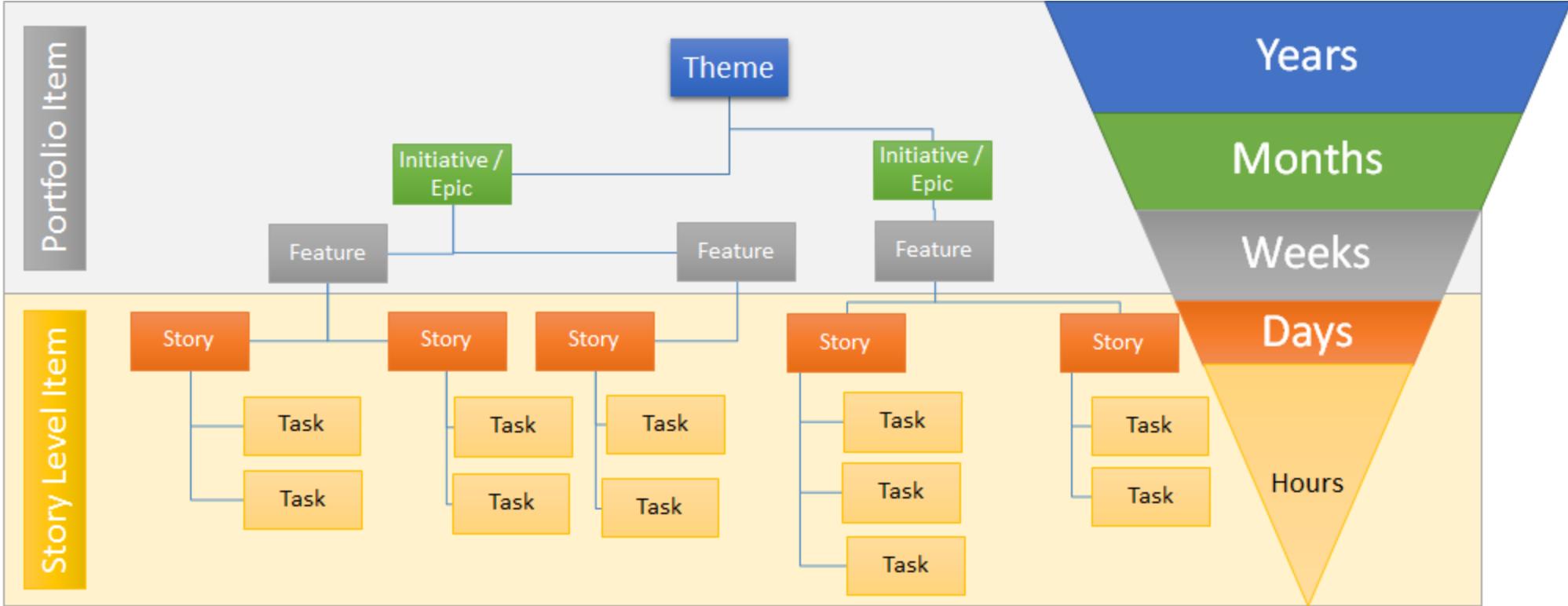
Zerlegung der Anforderungen



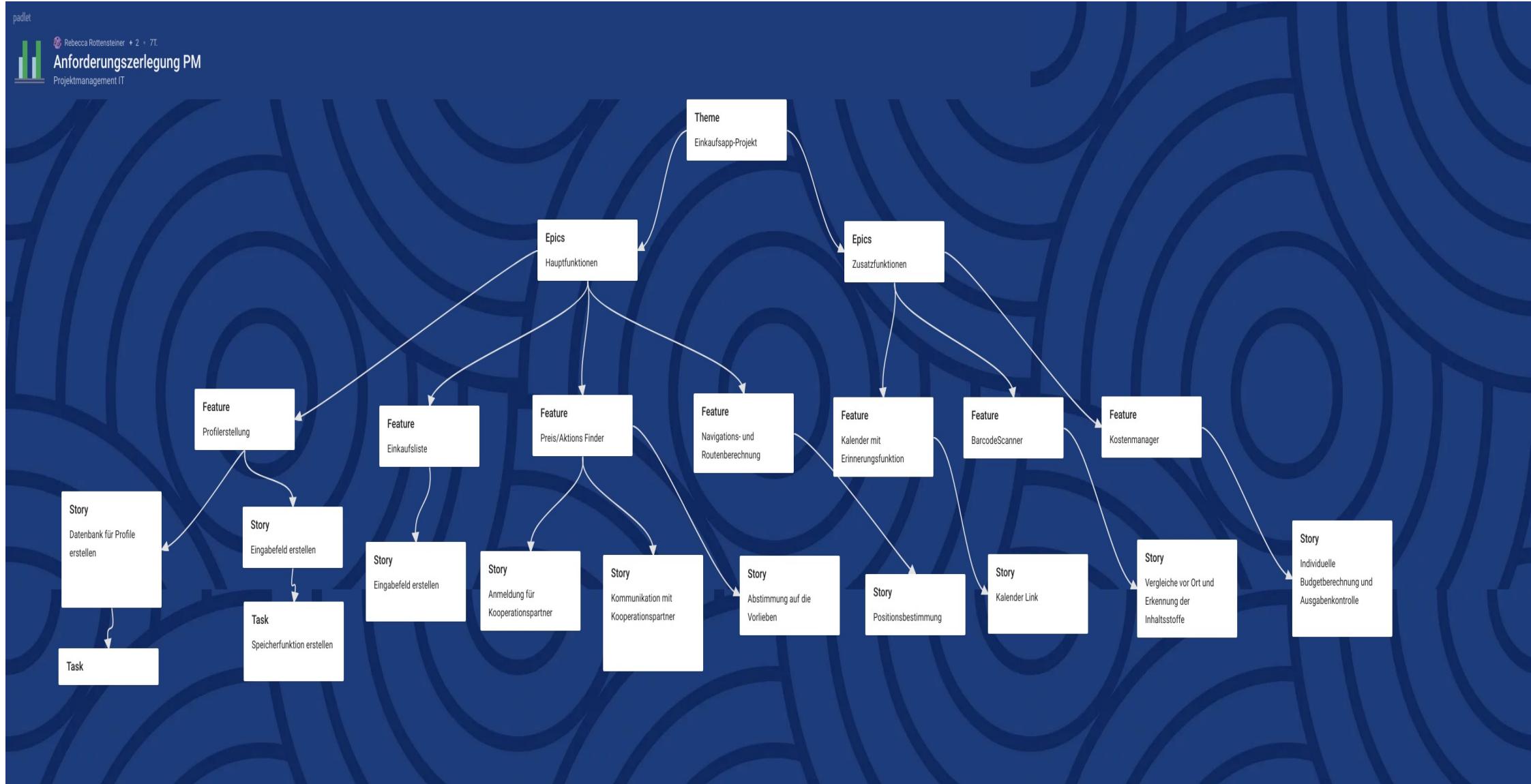
Einsatzgebiet

Bei kleineren Projekten kann auf die Ebene der Epics und User Stories verzichtet werden. Dann unterstehen Tasks und Bugs den Features direkt.

Zerlegung der Anforderungen



Zerlegung der Anforderungen BSP



Features

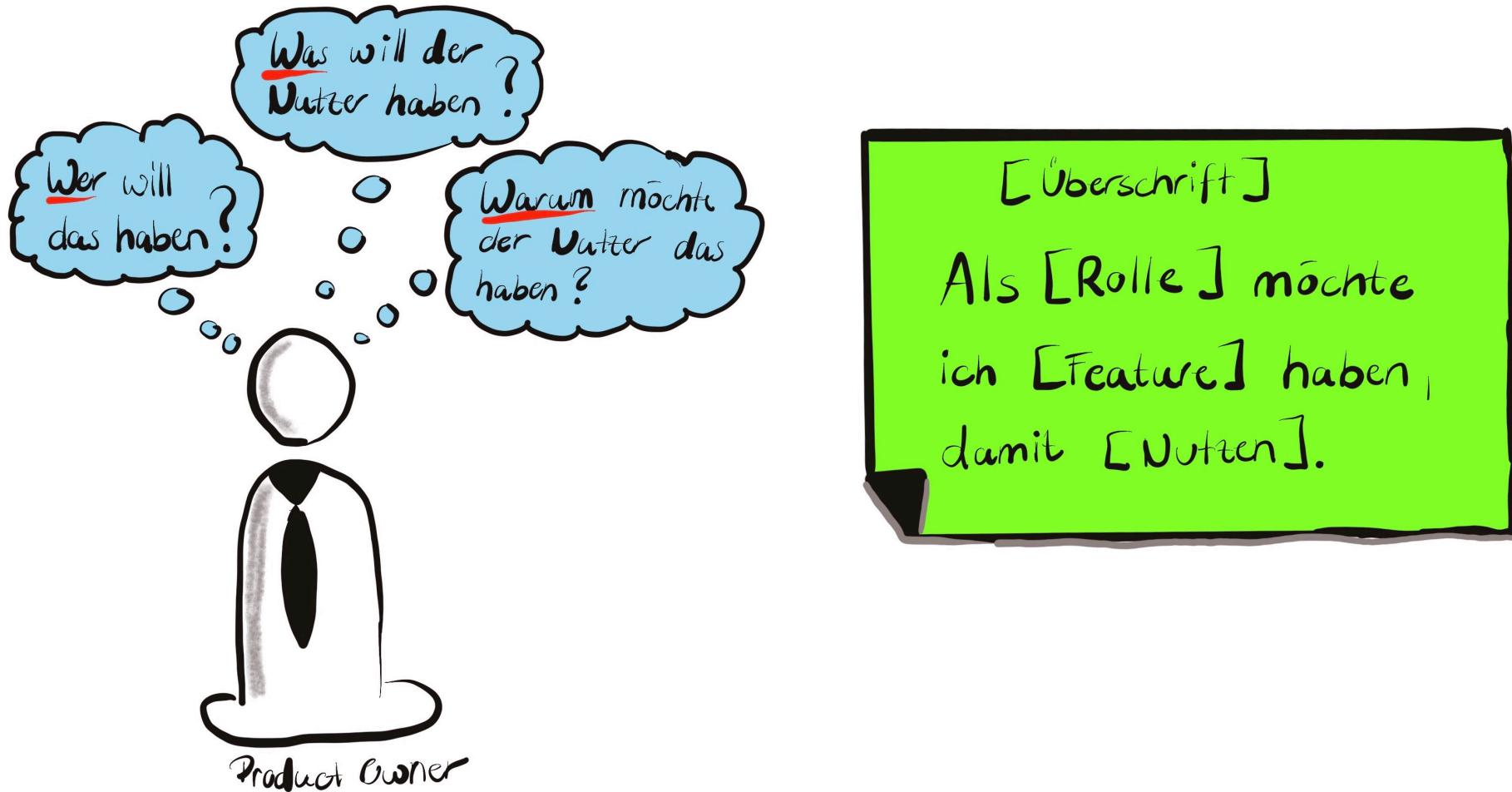
Der Unterschied zwischen User Story und Features:

ein Feature beschreibt die Funktionalität

eine User Story, welche Aktion ein Benutzer auf der Website ausführt

Es ist also wesentlich einfacher sich vorzustellen, welche Handlungen durchgeführt werden müssen, um die Anforderung an einen Webservice zu erfüllen, als welche Features eine Website enthält. Unserer Erfahrung nach lässt die Auflistung von Features zu viel Spielraum und oftmals gehen die Vorstellungen in der Umsetzung dann einfach zu weit auseinander. Das kann schlussendlich zu - für beide Seiten - ungewollten Mehraufwänden führen.

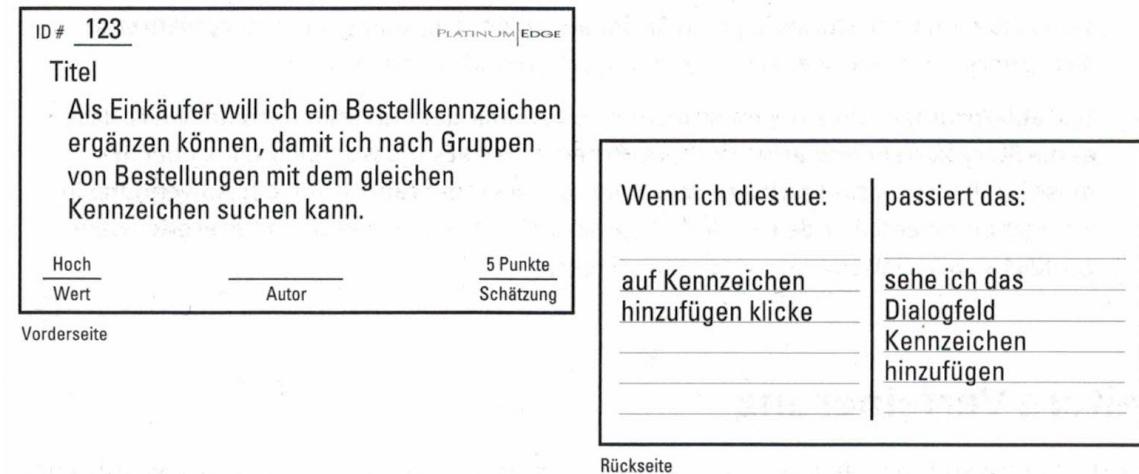
User Stories



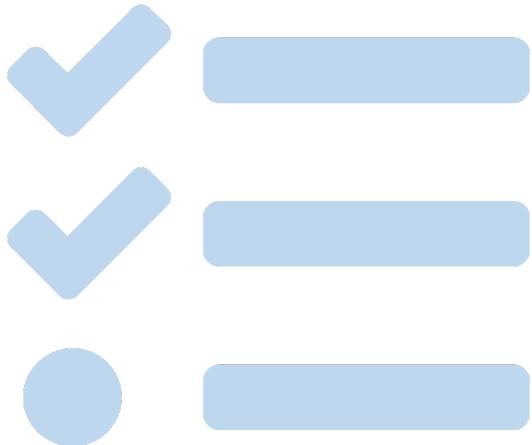
User Story

- Eine ID-Nummer, die die gleiche wie im Produkt-Backlog ist und die vom Product-Owner vergeben wird, wenn er einen Eintrag in das Produkt-Backlog aufnimmt
- Einen kurzen Titel
- Eine Beschreibung des Benutzers
- Was der Anwender erreichen will
- Der Grund, warum der Benutzer die Aufgabe erfüllen will

Abbildung 3.6 zeigt ein Beispiel für eine fertige Karte.

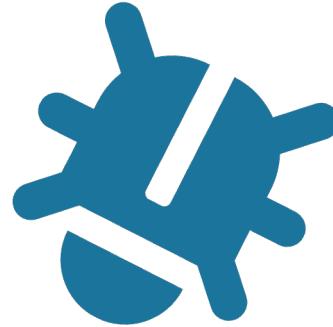
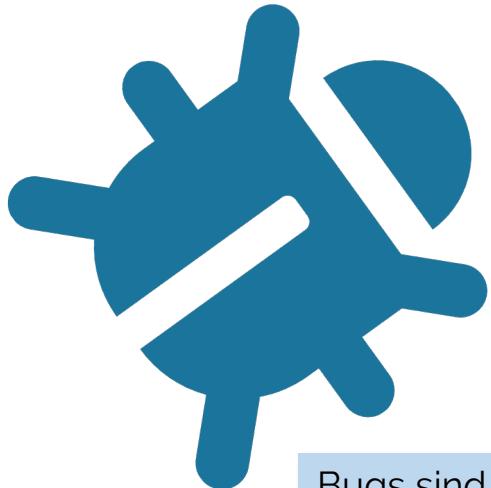


Tasks

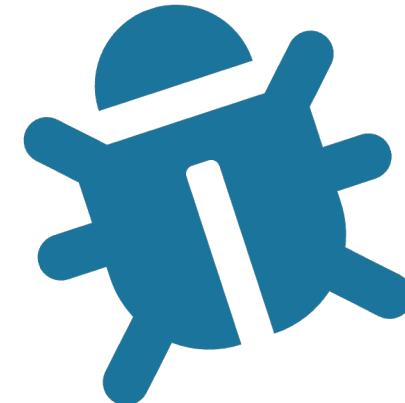
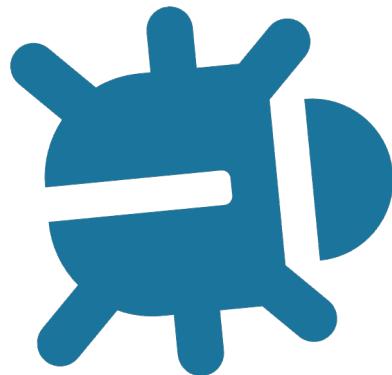


Die Anforderungen in konkrete Aufgaben (Tasks) zerlegt.
Diese sollten innerhalb eines Tages bearbeitet werden können.
Das Ergebnis ist der Sprint Backlog.

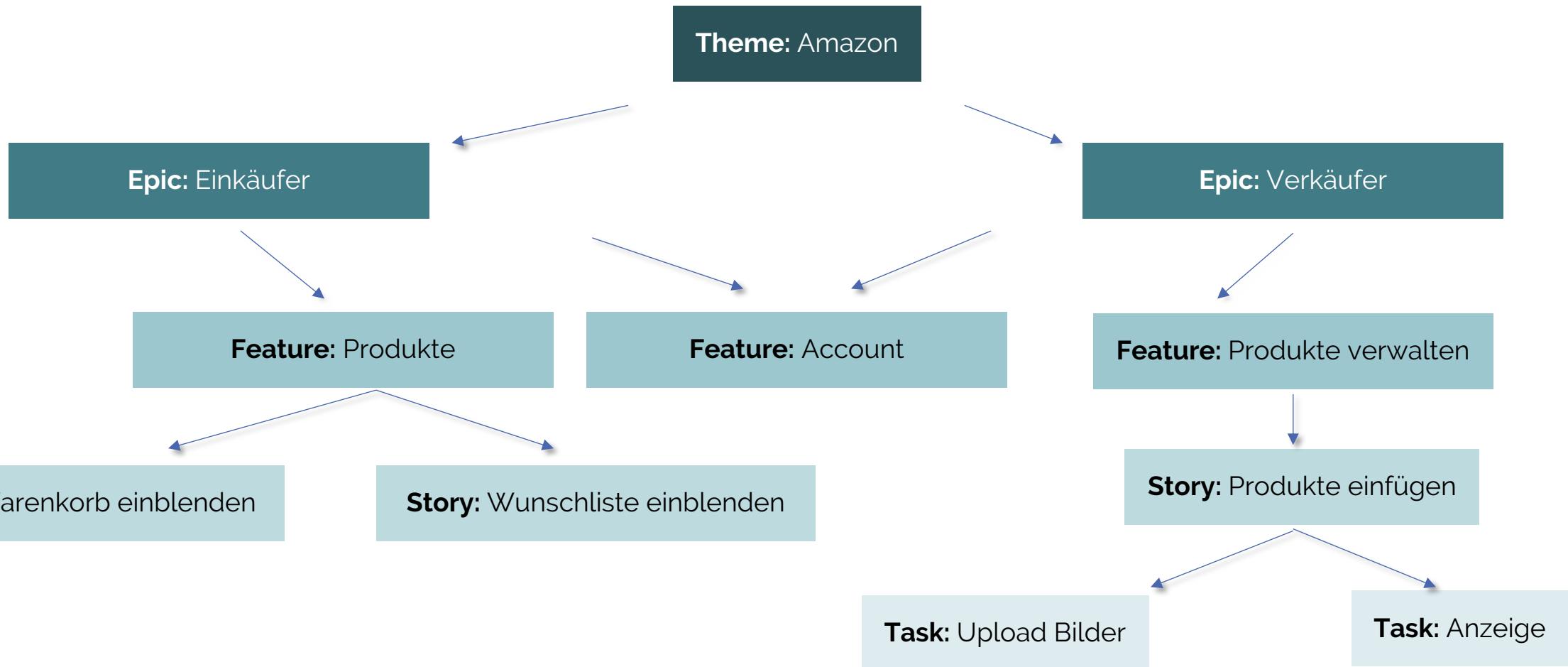
Bugs



Bugs sind Fehler in einem Programm. Es ist umstritten ob es Sinn macht diese vorher zeitlich zu schätzen oder sie zu priorisieren. Das kommt sowohl auf das Projekt als auch auf den Aufwand an, sollte man für jedes Projekt bzw. jeden Bug selbst entscheiden.

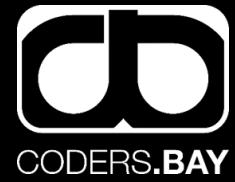


Beispiel Zerlegung Amazon



Übungshinweis

Jedes Produktfeature passt auf eine Haftnotiz. Diese Methode ist in der Praxis gelebte Einfachheit.



Taskboard

Das Taskboard

Die Visualisierung der Artefakte und die Darstellung der Fortschritte fördert die Zusammenarbeit in Teams. Oft ist ein Taskboard ein zentraler Anlaufpunkt, an dem sich Mitarbeiter treffen, um über Projekte, Prioritäten, Anforderungen etc. zu sprechen.

Darauf lässt sich jederzeit erkennen, welche Product Backlog Einträge für den Sprint ausgewählt wurden, welche Aufgaben dazu zu bearbeiten sind, und in welchem Bearbeitungszustand diese Aufgaben sind.

Es reizt zum Denken und Handeln an, weil es sich im Arbeitsbereich des Scrum-Teams befindet, wo jeder das Board sehen kann.

- Das Taskboard – Fortschritt verfolgen

To do	In Arbeit	Abnahme	Fertig
Die User Stories und Aufgaben, die noch erledigt werden müssen, befinden sich in der Spalte ganz links	User Stories und Tasks, an denen das Entwicklungsteam derzeit arbeitet, befinden sich in der Spalte In Arbeit. In dieser Spalte sollte sich nur eine User Story befinden. Wenn sich mehrere User Stories in Arbeit befinden, ist dies ein Hinweis darauf, dass die Mitglieder des Entwicklungsteams nicht cross-funktional arbeiten, sondern stattdessen beliebte Tasks horten. Sie gehen das Risiko ein, dass am Ende des Sprints mehrere User Stories teilweise fertig sind und nicht mehr User Stories komplett fertiggestellt wurden.	Nachdem das Entwicklungsteam eine User Story fertiggestellt hat, verschiebt es sie in die Spalte Abnahme. User Stories in der Spalte Abnahme können vom Product Owner überprüft werden; entweder gibt er dem Team Feedback, oder er akzeptiert die User Story	Wenn der Product Owner eine User Story überprüft und verifiziert hat, dass die User Story fertiggestellt ist, kann der Product Owner diese User Story in die Spalte Fertig verschieben

Taskboard

OPEN	PLANNED	IN PROGRESS	IN TESTING	DONE
TASK 13 Teammitglied 1.5 h	TASK 12 Teammitglied 1.75 h	TASK 10 Teammitglied 1.5 h	TASK 7 Teammitglied 1.5 h	TASK 1 TASK 2
TASK 14 Teammitglied 2.5 h	TASK 11 Teammitglied 0.5 h	SUPPORT 1 Teammitglied 2.75 h	BUG 3 Teammitglied 0.25 h	TASK 3 TASK 4
BUG 5 Teammitglied 4 h		BUG 4 Teammitglied 0.75 h	TASK 8 Teammitglied 1.5 h	TASK 5 BUG 1
SUPPORT 2 Teammitglied 0.5 h			TASK 9 Teammitglied 3.5 h	BUG 2 TASK 6

Leitfaden für die Erstellung: Produkt-Roadmap

1. pro Haftnotiz 1 Produktanforderung. So viele wie möglich
2. Anforderungen in verwandte Kategorien oder Gruppen einordnen
3. Priorisierung der Anforderungen auf der Roadmap. Rechte Seite höchste Priorität, linke Seite niedrigste Priorität.
4. Größenordnung des Aufwand der einzelnen Haftnotizen schätzen
5. Reihenfolge entsprechend den Abhängigkeiten und Schätzungen anpassen

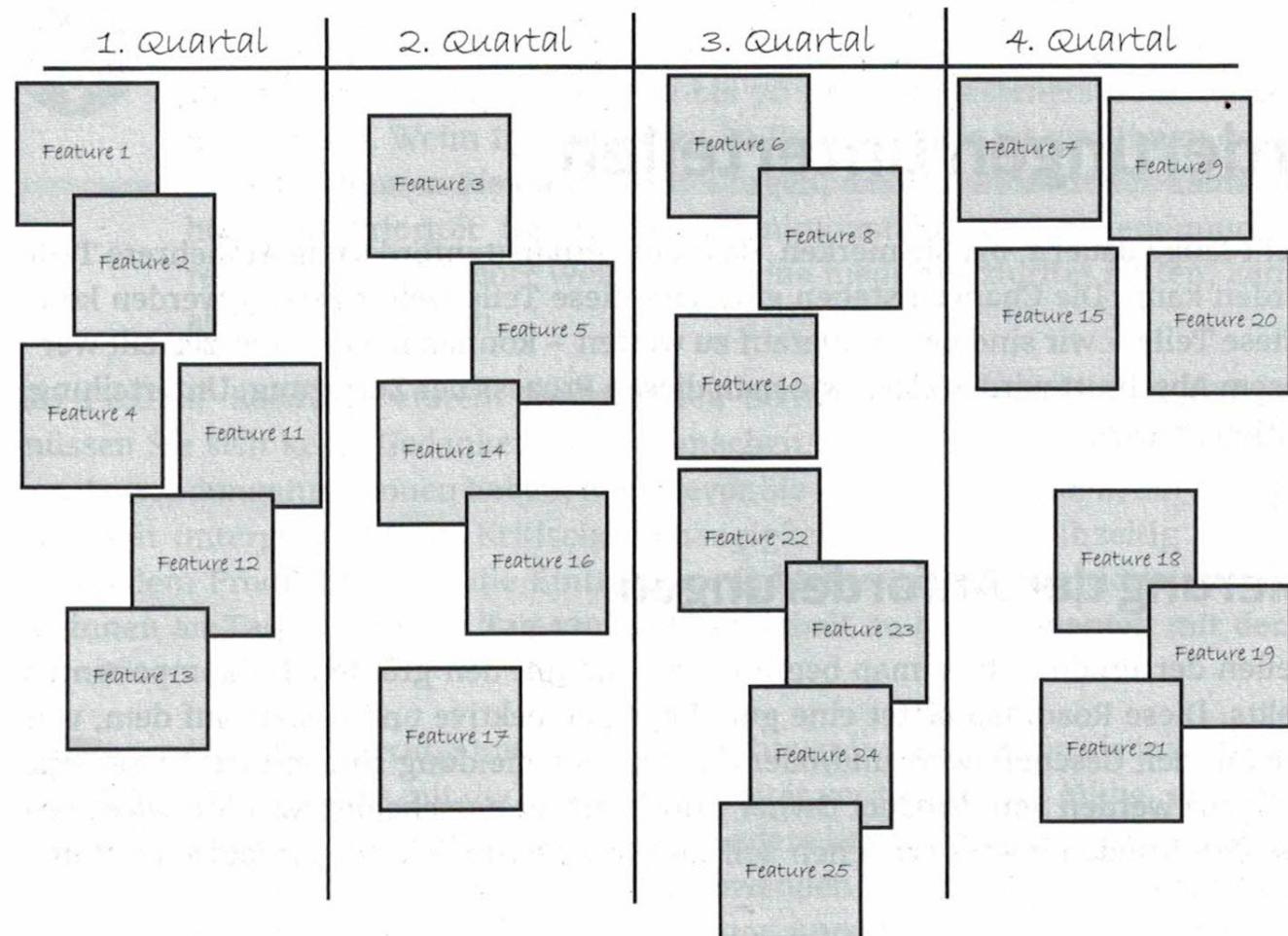
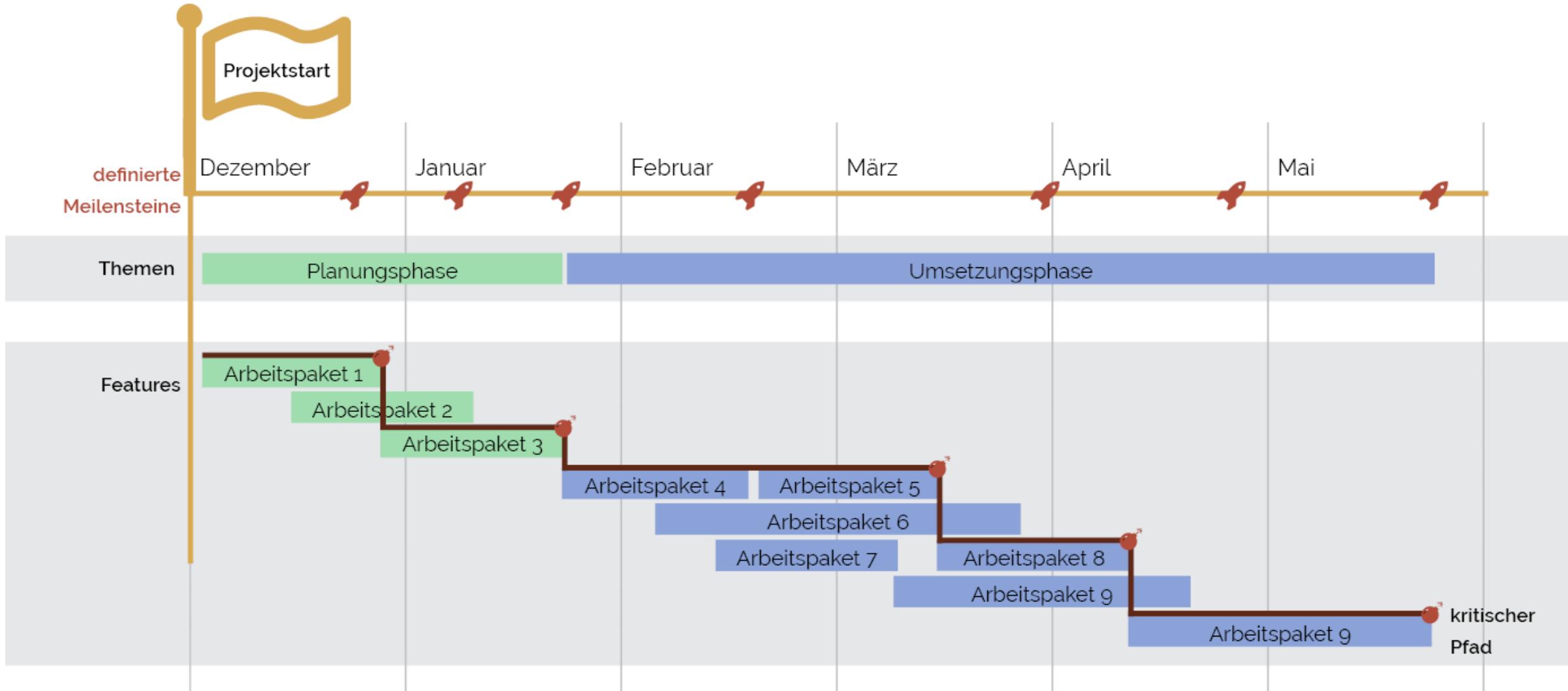


Abbildung 3.2: Ein Beispiel für eine Produkt-Roadmap nach Quartalen

Roadmap Beispiel



Produkt Road Map

Roadmap

Gemacht mit Frohsinn



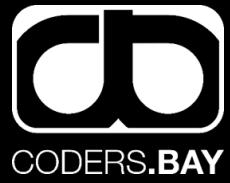
Eintrag im Produkt-Backlog

1. Nummer, die die Priorität des Eintrags im Produkt-Backlog angibt
2. Beschreibung
3. Aufwandsschätzung für die Fertigstellung des Elements
4. ID-Nummer (optional)
5. Status (optional)
6. Wert des Elements für das Unternehmen oder für das Produkt
7. Typ des Eintrags (optional), bei dem es sich um eine Anforderung, Overhead, eine Wartung oder eine Verbesserung handeln kann.

Nr.	ID	Element	Typ	Status	Schätzung
1	121	Als Administrator möchte ich Konten mit Profilen verknüpfen, damit Kunden auf neue Konten zugreifen können.	Anforderung	Nicht begonnen	5
2	113	Traceability Matrix der Anforderungen aktualisieren	Overhead	Nicht begonnen	2
3	403	Schulung Testautomatisierung für Michael	Verbesserung	Nicht begonnen	3
4	97	Refactoring der Login-Klasse	Wartung	Nicht begonnen	8
5	68	Als Besucher der Website will ich Filialen suchen können, damit ich Bankdienste nutzen kann.	Anforderung	Nicht begonnen	8

Abbildung 3.4: Das Produkt-Backlog ist die sortierte To-do-Liste Ihres Projekts.





Sprint-Planung

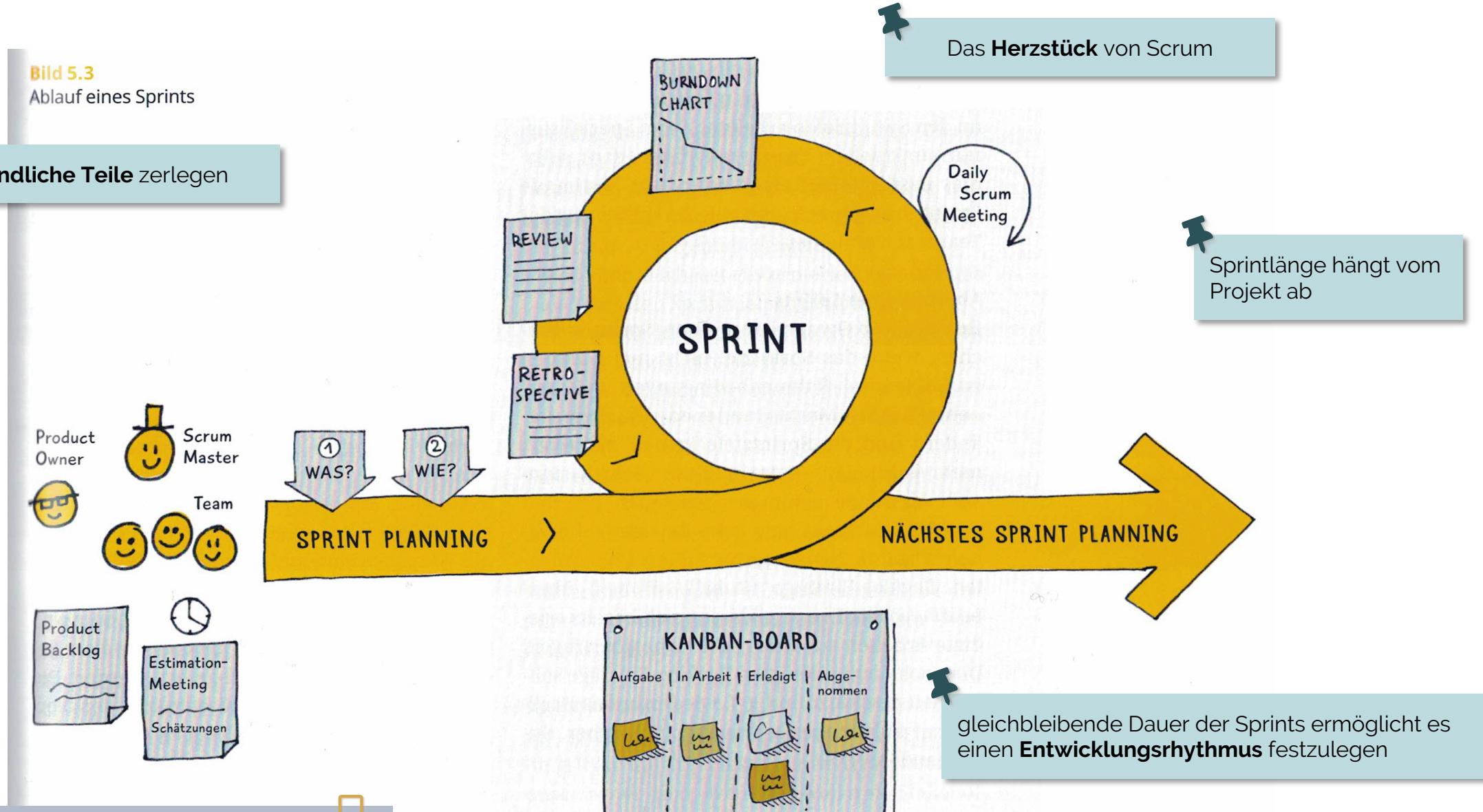
Ablauf eines Sprints

Bild 5.3

Ablauf eines Sprints



Projekt in **handliche Teile** zerlegen



Info

Wenn während eines Projekts die Sprintdauer geändert wird, entstehen erhebliche Kosten: frühere Velocity ist nicht mehr relevant

Phase 4 – Sprint-Planung

eine sich konsistent wiederholende Iteration einer bestimmten Dauer, in der das Entwicklungsteam eine bestimmte Gruppe von Produkteigenschaften von Anfang bis Ende implementiert.

Am Ende sollte die Funktionalität einsatzfähig sein, demonstriert werden und ausgeliefert werden können



 **Beschreibung:** Festlegen der interationsspezifischen Ziele und Tasks

 **Häufigkeit:** Zu Beginn jedes Sprints

 **Besitzer:** Product Owner & Entwicklungsteam

Enthält

1. Sprint-Planung zu Anfang des Sprints
2. Daily-Scrum-Meetings
3. Entwicklungszeit, der größte Teil des Sprints
4. Ein Sprint-Review und eine Sprint-Retrospektive am Ende des Sprints

Sprint-Backlog

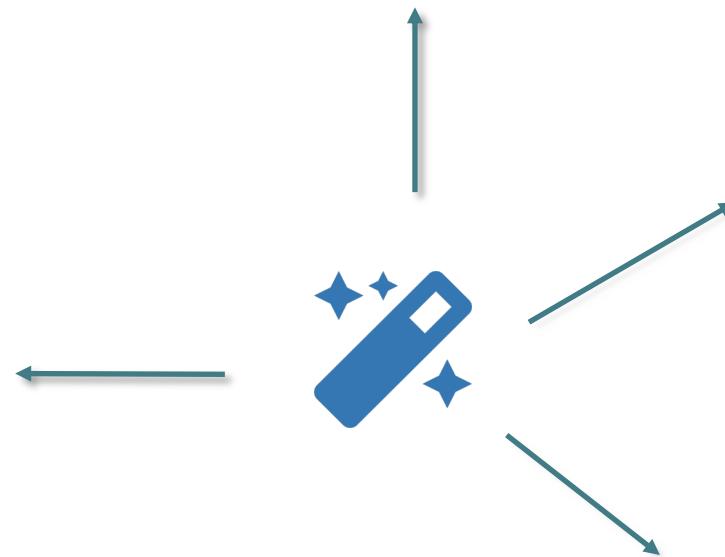
- Sprint Planen -

1. Legen Sie die **Ziele** für Ihren Sprint fest

4. Erstellen Sie ein **Sprint-Backlog**. Sprint-Backlog besteht aus:

- Die **Liste der User Stories** nach ihrer Priorität sortiert
- Die **relative Aufwandsschätzung** für jede User Story
- Die **Aufgaben (Tasks)**, die für die Entwicklung jeder User Story erforderlich sind
- Der **Aufwand, in Stunden**, um jede der Aufgaben zu erledigen
- Ein **Burndown-Chart**, das den Status der Arbeit darstellt, die das Entwicklungsteam fertig gestellt hat

2. Wählen sie die **User Stories** aus, die diese Ziele unterstützen



3. Teilen Sie die User Stories in **spezifische Entwicklungsaufgaben** auf



Info

Liste der User Stories für den aktuellen Sprint sowie damit zusammenhängende Aufgaben.

Burndown-Diagramm

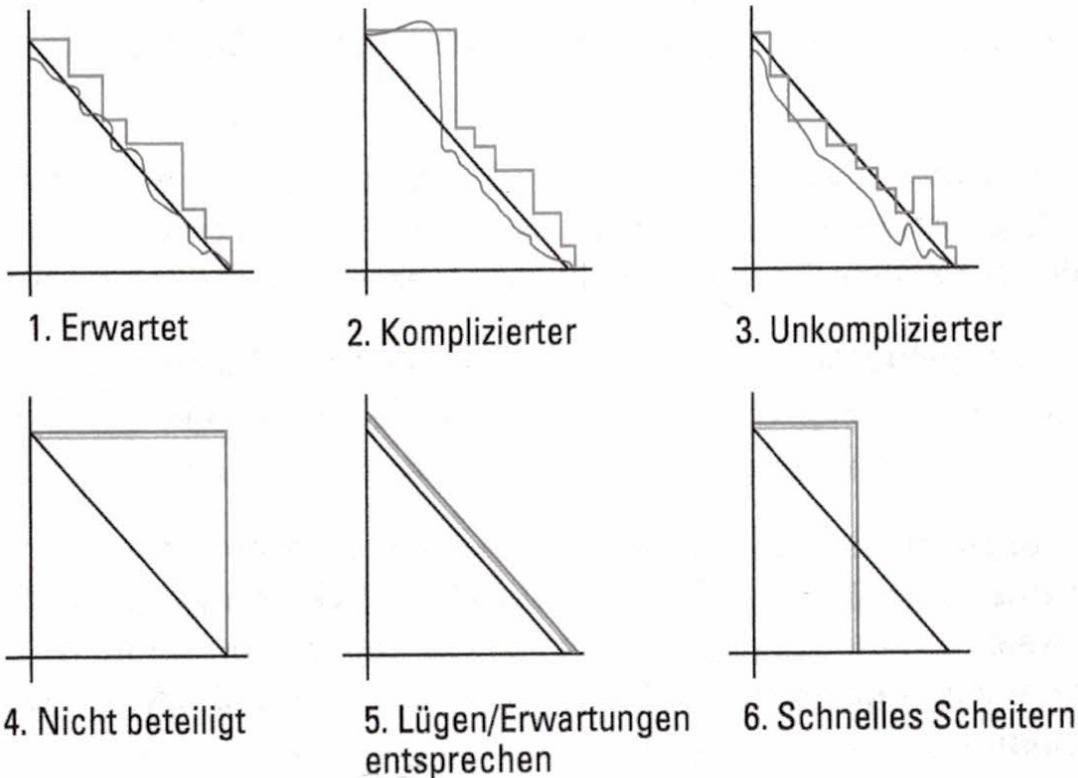


Abbildung 9.4: Typische Muster von Burndown-Diagrammen

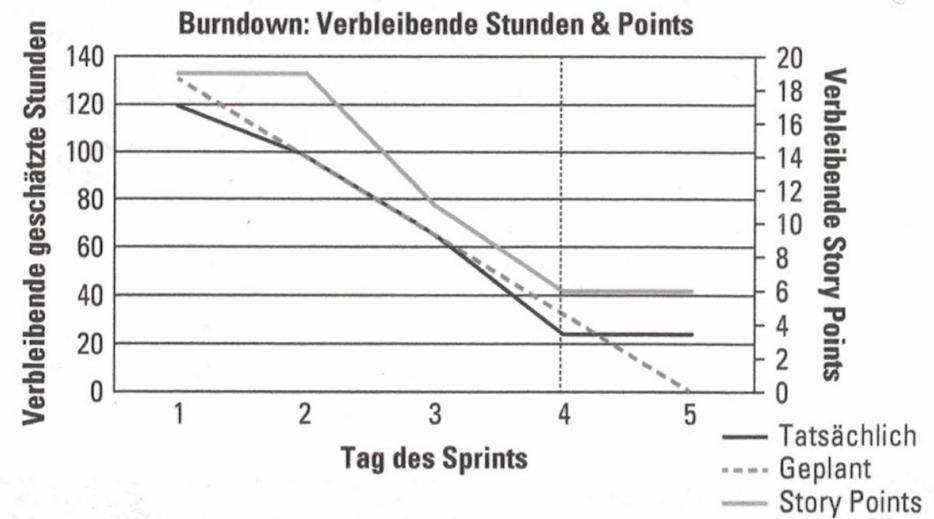


Abbildung 9.3: Ein Burndown-Diagramm

- Ein Burndown-Diagramm ist ein mächtiges Werkzeug, um den Fortschritt und die verbleibende Arbeit zu visualisieren. Das Chart stellt die folgenden Informationen dar:
- Die ausstehende Arbeit (in Stunden) auf der ersten vertikalen Achse
- Zeit, in Tagen, entlang der horizontalen Achse
- Manche Charts zeigen außerdem ausstehende Story Points auf einer zweiten y-Achse
- ermöglicht es jedem auf einen Blick, den Status des aktuellen Sprints zu erkennen.

Info

Burndown-Diagramme ermöglichen die visuelle Darstellung des Fortschritts eines Sprints. Sie zeigen den Umfang der geleiteten Arbeit im Vergleich zu dem, was noch zu tun ist.

Verhältnis Sprint Planning Meeting & Sprint Dauer

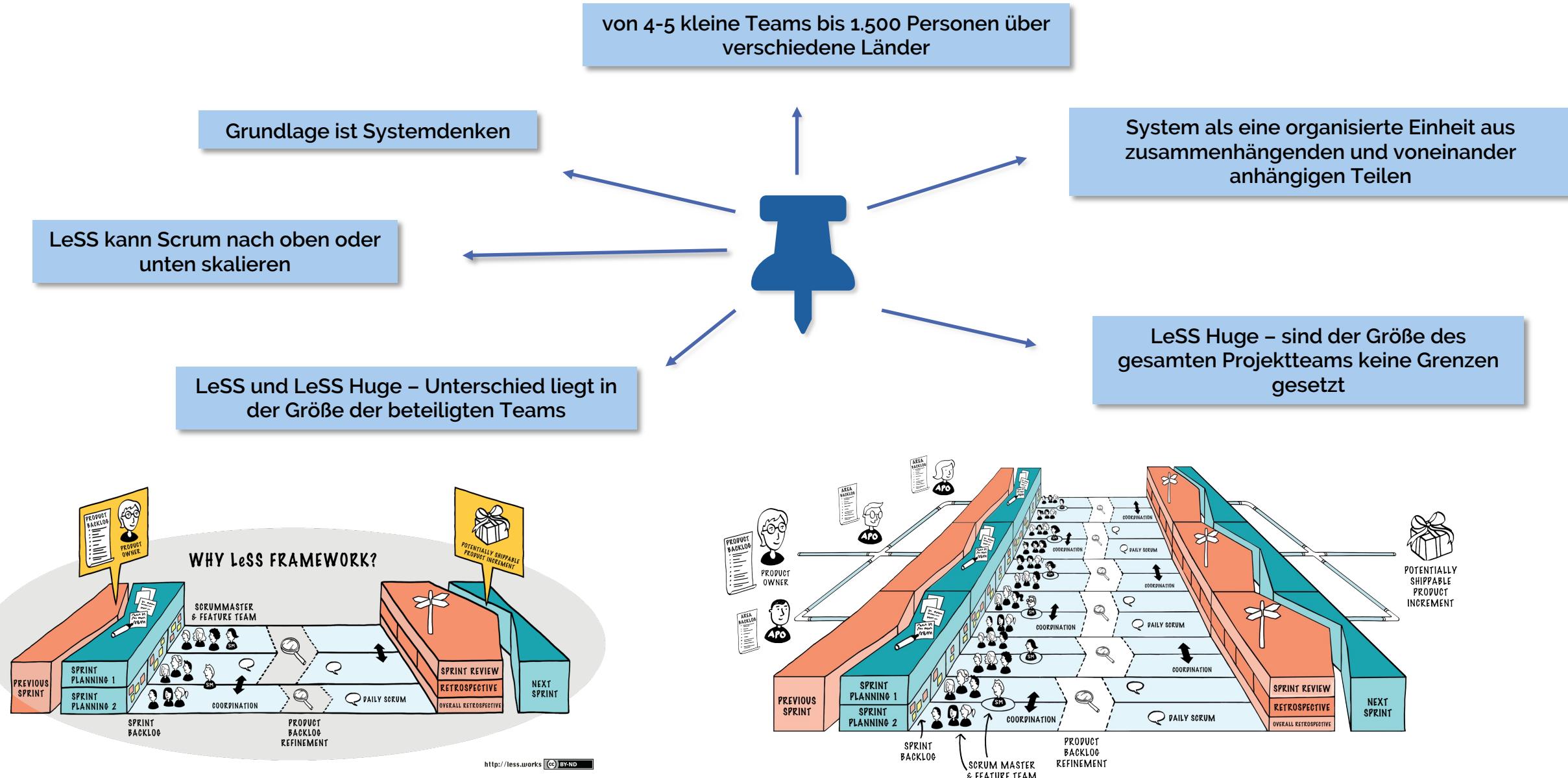


Wenn der Sprint so lang ist	Sprint Planning Meeting sollte nicht länger dauern als
Eine Woche	Zwei Stunden
Zwei Wochen	Vier Stunden
Drei Wochen	Sechs Stunden
Vier Wochen	Acht Stunden

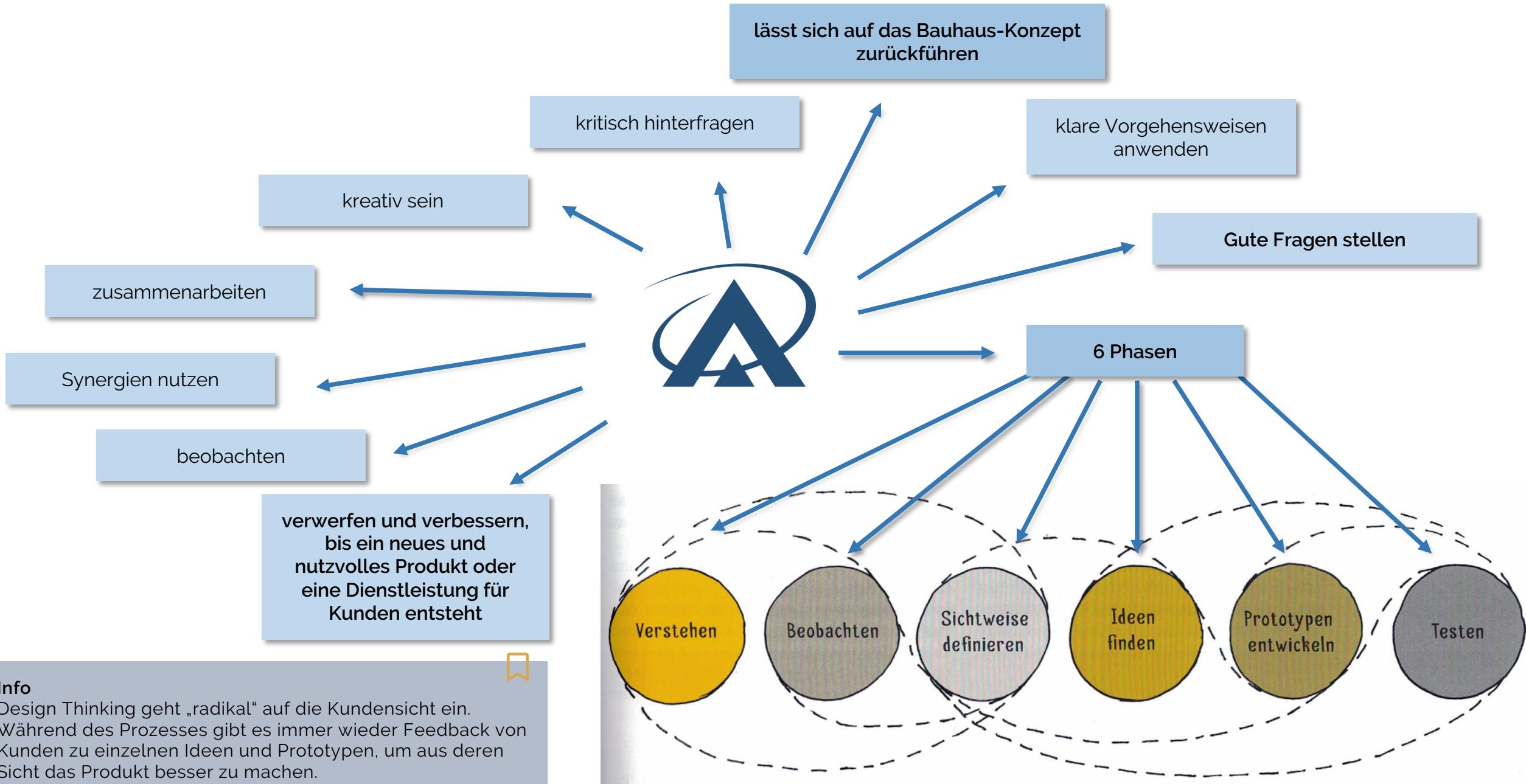


Agile Frameworks

LeSS - Large-Scale Scrum



Design Thinking



Kanban

Technik aus dem Toyota-Produktionssystem

kleinen Schritten, die regelmäßig überprüft und ggf korrigiert werden

japanische Wort Kanban bedeutet visuelles Signal, Tafel

Methode um Verschwendungen zu vermeiden

Arbeitsabläufe und den Durchsätze des Systems zu verbessern

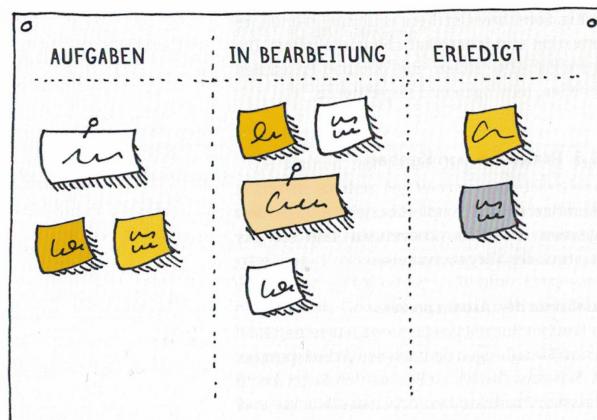


Bild 5.4
Einfaches Kanban Board

verbessern Effizienz

- ✓ Visualisieren
- ✓ Menge angefangener Arbeit begrenzen
- ✓ Arbeitsablauf steuern
- ✓ Regeln für den Prozess explizit machen
- ✓ Feedbackschleifen implementieren
- ✓ Zusammenarbeit verbessern, experimentell weiterentwickeln

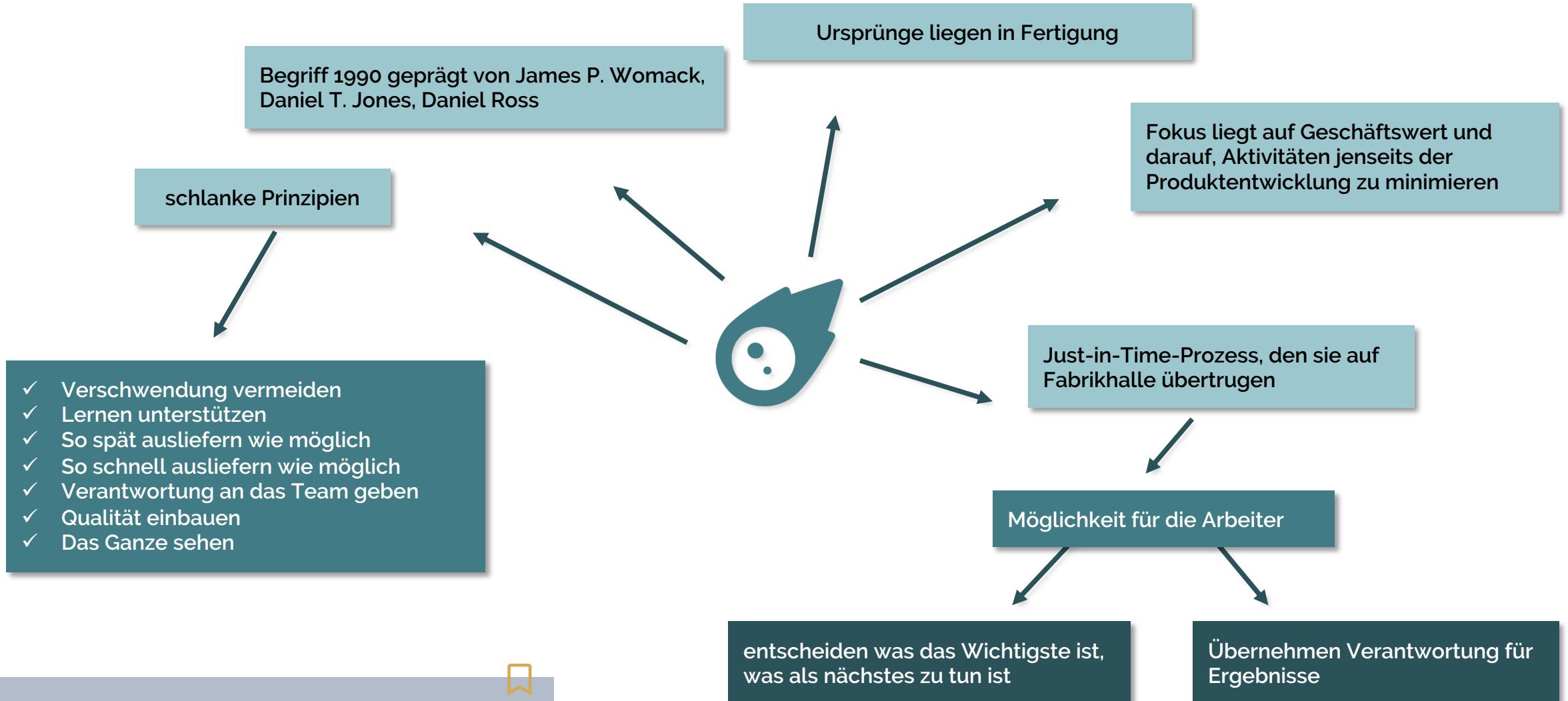
Wasserfall das gesamte Produkt durchläuft seriell die einzelnen Produktionsphasen

Kanban zerlegt dagegen in einzelne „Werkstücke“ bzw. Produktanforderungen

Info

Kanban ist eine Methode, bei der die Anzahl paralleler Arbeiten, der Work in Progress (WiP), begrenzt und somit kürzere Durchlaufzeiten erreicht und Probleme – insbesondere Engpässe – schnell sichtbar gemacht werden sollen.

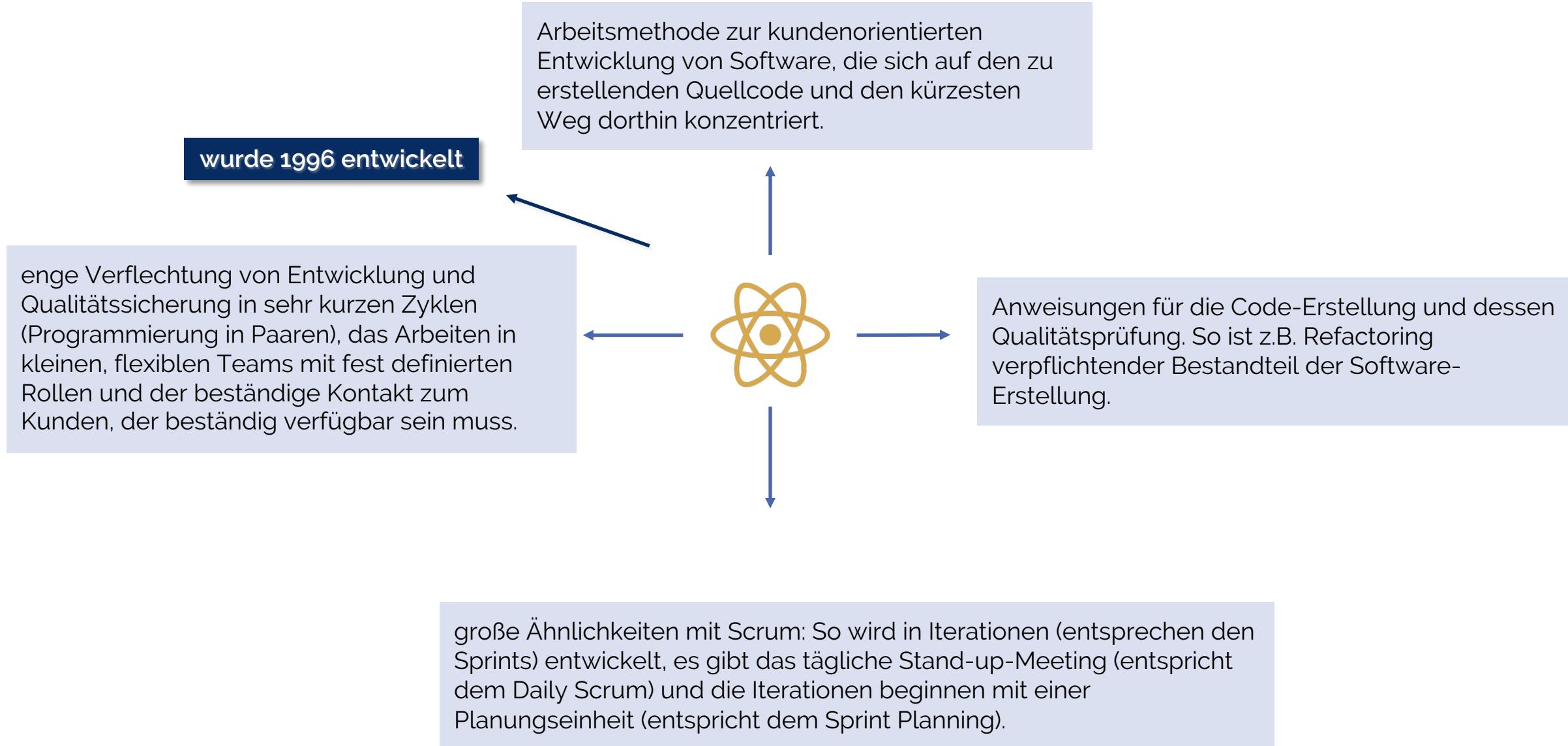
Lean



Info

Das Konzept dahinter ist, die Maschinen und Leute arbeiten zu lassen, Lagerbestände aufzubauen und so sehr effizient zu sein.

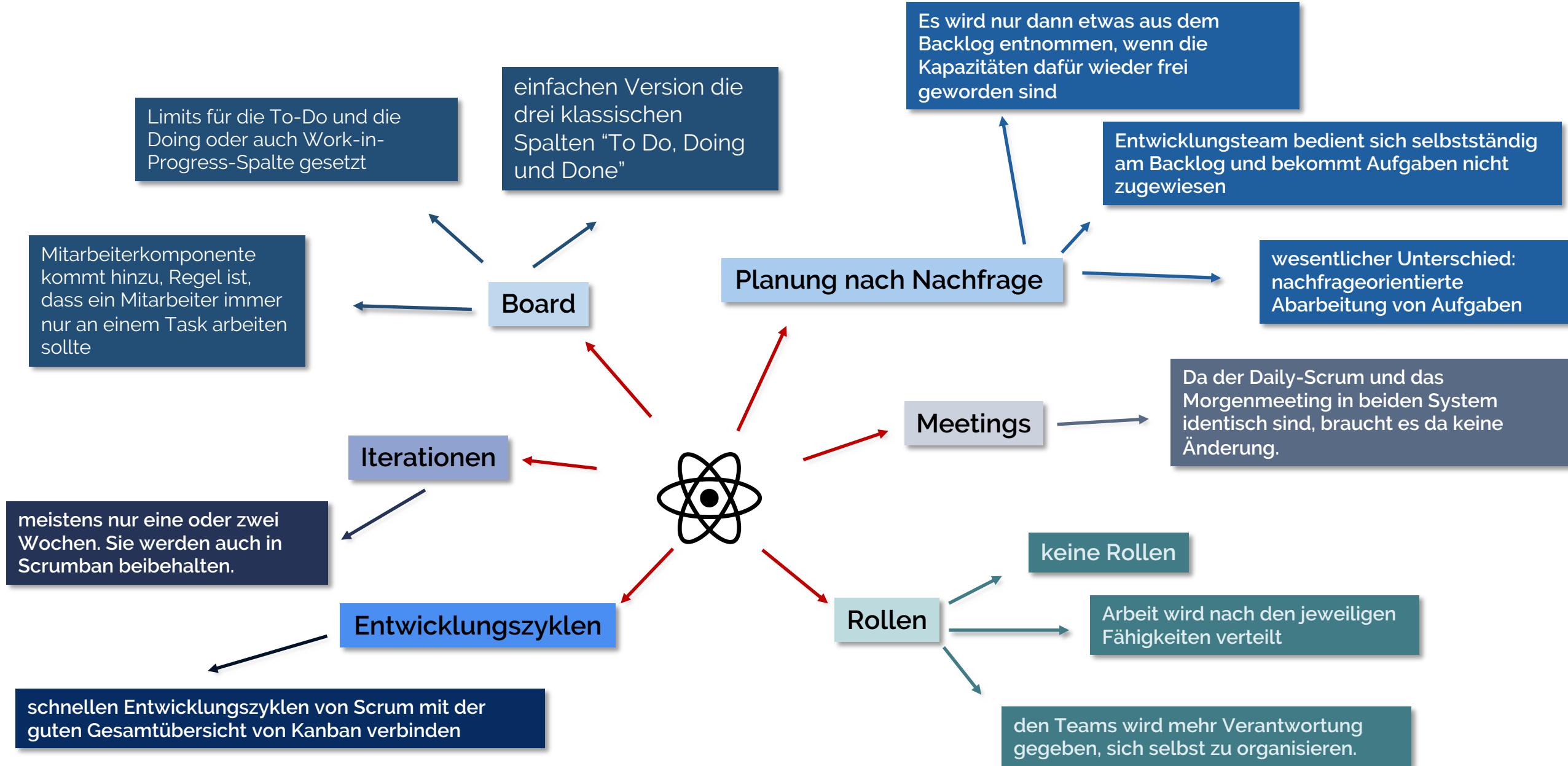
Extreme Programming



Exkurs Refactoring

Refactoring ist eine Methode zur systematischen Restrukturierung von Programm-Code, bei der sich das Verhalten der Software aus Anwendersicht nicht verändert. **Refactoring** beeinflusst somit die Qualität, nicht aber den Umfang des im Projekt erstellten Werks. Ob bei einer iterativen Software-Entwicklung ein abschließendes **Refactoring** zum Leistungsumfang des Projekts gehört, ist somit davon abhängig, ob Qualitätskriterien wie z.B. Wartungsfähigkeit oder Weiterentwickelbarkeit gefordert sind.

Scrumban



Fazit



Design Thinking

ist eine kreative Herangehensweise, zusammen mit dem Kunden und einem interdisziplinären Team in sechs Phasen ein innovatives Produkt bzw. eine Dienstleistung zu kreieren. Herausforderung besteht in einer guten und wirtschaftlichen Idee



Es werden Produkte und Dienstleistungen entwickelt, die für Kunden von Bedeutung sind. Die Interessen des zukünftigen Kunden werden hoch priorisiert.



Scrum

ist die am besten ausdifferenzierte agile Vorgehensweise und verbindet prozess- und teamorientierte Vorgehensweisen. Viele Werkzeuge von Scrum werden heute in andere agile Arbeitsweisen adaptiert.



ist das differenzierteste agile Framework, lebt von einer klaren Rollendefinition sowie selbstorganisierten Teams und plant in Sprints. Sehr komplex und benötigt intensive Schulung.



LeSS

hilft, komplexe Produktentwicklung in eine einfache Struktur zu bringen, damit diese gut bearbeitbar wird.



ist eine Adaption von Scrum, die helfen soll, sehr große Gruppen zielführend zu steuern. Komplexe Produktentwicklung kann so in eine einfache Struktur gebracht werden. Anfang eine zeitintensive Schulungsphase



Kanban



hilft, Prozesse im Unternehmen effizienter und kalkulierbarer zu gestalten. Im Fokus steht immer eine Prozessverbesserung, jedoch keine sozialen oder persönlichen Entwicklungsprozesse



macht Prozesse stabiler, berechenbarer und effizienter im Unternehmen und schafft damit eine wesentliche Wertschöpfung

Beispiel: Scrum und das FBI

Cyberangriffe aller Art, von Kreditkarten-Hacking bis hin zu Regierungs- und militärischen Sicherheitsbedrohungen, nehmen weltweit zu. Militär- und Cyberkrieg-Experten sind sich einig, dass die Kontrolle dieser wachsenden Bedrohung ein großes Anliegen ist. Programme für die Bekämpfung von Cyberbedrohungen, die sich an die sich ständigen Herausforderungen anpassen können, werden benötigt – und zwar schnell. In vielen Fällen ist es eine Frage der nationalen Sicherheit.

Das FBI hat zunehmend aktiv an Lösungen gearbeitet, um auf diese Cyberangriffe auf nationaler und krimineller Ebene reagieren zu können. Die Führung weiß, dass es nicht ausreicht, alte Techniken zum Umgang mit diesen übermodernen Bedrohungen einzusetzen. Neue Wege des Technologieeinsatzes, der Personalgestaltung und der Zusammenarbeit mit Partnern sind gefragt.

Kein Wunder, dass sie sich dieser schwierigen Aufgabe mit Scrum nähern.

Nach den Terroranschlägen vom 11. September 2001 begann das FBI damit, den Informationsfluss und die Koordination mit allen relevanten Stellen zu verbessern. Nach einigen Fehlstarts entwickelte man das Programm Sentinel: ein umfassendes Software Case Management System. Ziel ist es, ein bestehendes System aus einer Mischung aus digitalem und papiergebunden

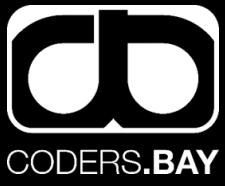


Informationsfluss durch ein rein digitales zu ersetzen.

Zehn Jahre lang wurde das Projekt mit den vom FBI vorgeschriebenen Wasserfallmethoden durchgeführt, einschließlich umfangreicher Vorentwürfe und festgelegter Anforderungen. Ein neuer CIO, Chad Fulgham, verlangte von den beauftragten Entwicklern eine inkrementelle Lieferung. Hier sind einige der Symptome des bis dahin verwendeten Wasserfallprozesses:

Nur 4 von 18 Workflows waren live, aber mit Fehlern. Weniger als 1 Prozent der mehr als 14.000 Formulare wurden im neuen System erstellt. Die bisherigen Kosten beliefen sich auf 405 Millionen Dollar, mit geschätzten weiteren 350 Millionen für die folgenden sechs Jahre. Alle gelieferten Funktionen wurden innerhalb der Organisation als optional angesehen.

Scrum wurde drei Monate lang eingesetzt und man konnte erreichen, was mit dem Wasserfallansatz in 10 Jahren nicht möglich war: 5% des Budgets und 80% weniger Personal. Insgesamt wurde der Personalüberhang um mehr als 50 Prozent reduziert, es wurden User Stories erstellt und 21 Sprints von zweiwöchiger Dauer geplant – was zu einer Verkürzung des Projektzeitplans um 85% führte.



Daily SCRUM

Daily Scrum

Entwicklungsteam

Im Daily-Scrum-Meeting macht jedes Mitglied des Entwicklungsteams die folgenden drei Aussagen

- ✓ **Gestern habe ich** (nennen Sie die fertiggestellten Elemente)
- ✓ **Heute werde ich** (nennen Sie die Aufgabe)
- ✓ **Meine Hindernisse sind** (zählen Sie die Hindernisse auf, falls vorhanden)

Scrum Master

Stellungnahme in Bezug auf die Hindernisse des Teams

- ✓ **Gestern konnte ich Folgendes lösen** (nennen Sie die Hindernisse, die beseitigt wurden)
- ✓ **Heute arbeite ich an der Beseitigung von** (nennen Sie das Hindernis)
- ✓ **Die Hindernisse, die ich eskalieren werde, sind** (nennen Sie die Hindernisse, bei denen Sie Unterstützung benötigen, falls vorhanden)

Beschreibung

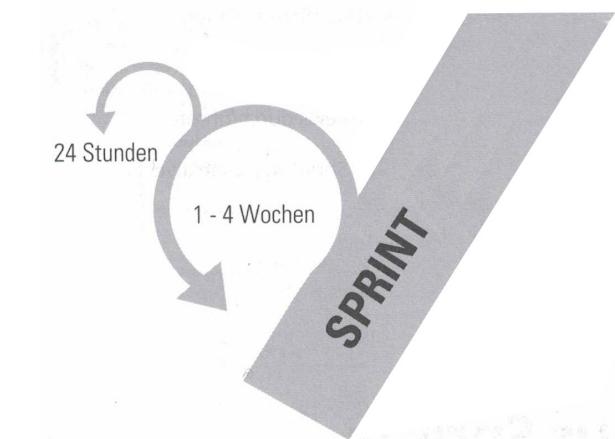
Festlegen und Koordinieren der Prioritäten dieses Tages

Besitzer

Entwicklungsteam

Häufigkeit

Täglich

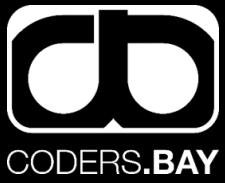


Daily Scrum

Damit es kurz und effektiv bleibt, können verschiedene Regeln befolgt werden

- ✓ Alle dürfen am Daily Scrum teilnehmen, reden jedoch dürfen nur das Entwicklungsteam, der Scrum Master und der Product Owner.
- ✓ Der Fokus liegt auf den bevorstehenden Prioritäten
- ✓ Daily-Scrum-Meetings dienen der Koordination, nicht der Problemlösung
- ✓ Pünktlich beginnen





Sprint Review

Sprint-Review



Richtlinien

Dauer ca. 1h

- ✓ **Entwicklungsteam** bereitet fertiggestellte Anforderungen für Präsentation vor
- ✓ keine formale Folien
- ✓ Nur Anforderungen demonstriert, die gemäß Definition of Done als fertig gelten und vom Product Owner genehmigt wurden
- ✓ Entwicklungsteam zeigt nutzbare Funktionalität der Anforderungen



Rollen

Stellungnahme in Bezug auf die Hindernisse des Teams

- ✓ **Scrum Master:** unterstützt Meeting und stellt sicher, dass es fokussiert bleibt und die Zeit eingehalten wird
- ✓ **Product Owner:** stellt Sprint-Ziel vor und wie gut es erreicht wurde. Informiert über fertiggestellte Backlog Elemente und fasst zusammen, was für anstehende Releases noch zu tun ist
- ✓ **Entwicklungsteam:** Zeigt und erläutert fertiggestellte Anforderungen
- ✓ **Stakeholder:** Stellen Fragen und geben Feedback



Beschreibung

Demonstration des funktionierenden Produkts



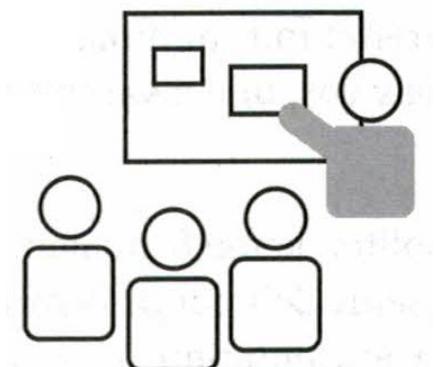
Besitzer

Product Owner, Entwicklungsteam



Häufigkeit

Am Ende jedes Sprints



Sprint Review

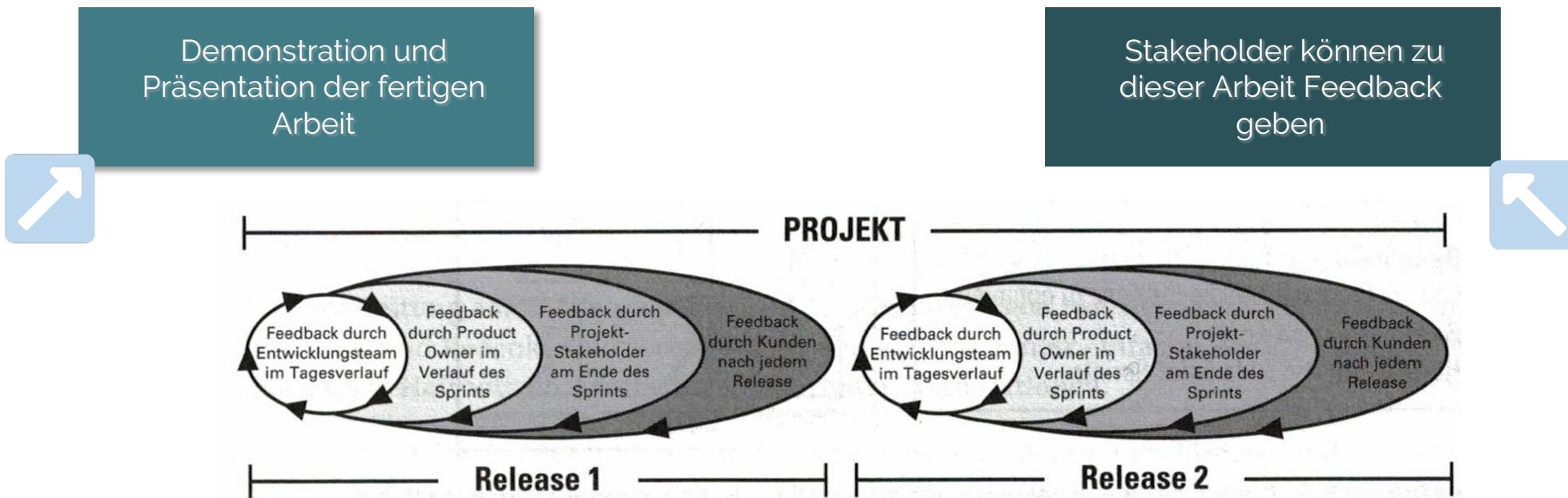


Abbildung 6.4: In einem typischen Scrum-Projekt gibt es mehrere Feedback-Ebenen.

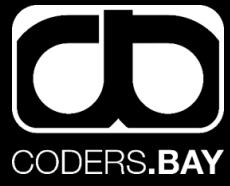
- ✓ **Keine PowerPoint-Folien!** Funktionalität im Einsatz zeigen
- ✓ Das **gesamte Scrum-Team** sollte an dem Meeting teilnehmen
- ✓ **Alle**, die an dem Meeting interessiert sind, **dürfen teilnehmen**.
- ✓ Der Product Owner führt in das **Release Ziel**, das **Sprintziel** und die neu **eingeführten Möglichkeiten** ein
- ✓ Das Entwicklungsteam **demonstriert**, was es während des Sprints fertiggestellt hat. Typischerweise präsentiert das Entwicklungsteam neue Features oder Architekturen
- ✓ Die Demonstration sollte auf Geräten stattfinden, die so gut wie möglich der geplanten Produktionsumgebung entsprechen.
- ✓ Stakeholder können **Fragen** zum demonstrierten Produkt stellen und Feedback geben
- ✓ Keine nicht offengelegte, manipulierte Funktionalität, die der Anwendung den Eindruck verleihen sollen, ausgereifter zu sein, als sie es tatsächlich ist.
- ✓ Der Product Owner kann eine **Diskussion** darüber leiten, was basierend auf den gerade präsentierten Features und neuen Elementen, die während des aktuellen Sprints in das Product-Backlog aufgenommen wurden, als Nächstes kommt



Sprint Review - Dauer



Wenn mein Sprint so lang ist	Sollte mein Sprint-Review-Meeting nicht länger dauern als
Eine Woche	Eine Stunden
Zwei Wochen	Zwei Stunden
Drei Wochen	Drei Stunden
Vier Wochen	Vier Stunden



Sprint Retrospektive

Sprint-Retrospektive



Richtlinien

ist ein Meeting, in dem Product Owner, Entwicklungsteam und Scrum Master diskutieren, wie der Sprint verlief und was sie tun können, um den nächsten Sprint zu verbessern.



Die drei primären Fragen

- ✓ Was lief gut im Verlauf des Sprints?
- ✓ Was würden wir gern ändern?
- ✓ Wie können wir diese Veränderung implementieren?



Beschreibung

Team verfeinert die Umgebung und die Prozesse, um die Effizienz zu optimieren



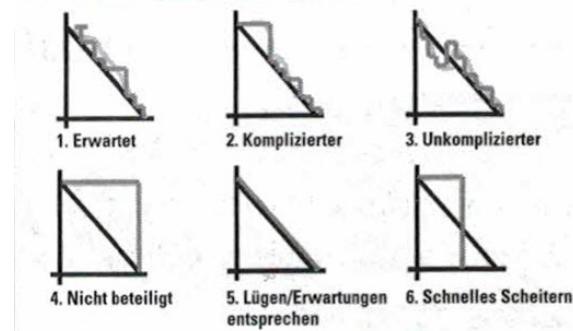
Besitzer

Scrum-Team



Häufigkeit

Am Ende jedes Sprints



Sprint Retrospektive

Weitere Bereiche die diskutiert werden könnten

Ergebnisse

Vergleichen Sie die Menge der geplanten Arbeit damit, was das Entwicklungsteam fertiggestellt hat. Schauen Sie sich das Sprint-Burndown-Diagramm an und was es dem Entwicklungsteam darüber sagt, wie es arbeitet

Leute

Diskutieren Sie die Teamzusammenstellung und die Übereinstimmung

Beziehungen

Reden Sie über die Kommunikation, die Zusammenarbeit und das Arbeiten in Zweiergruppen

Prozesse

Behandeln Sie die Unterstützungs-, Entwicklungs- und Peer-Review-Prozesse

Werkzeuge

Wie gut sind die verschiedenen Werkzeuge für das Scrum-Team geeignet? Besprechen Sie Artefakte, elektronische Werkzeuge, Kommunikationswerkzeuge und technische Werkzeuge

Produktivität

Wie kann das Team die Produktivität verbessern und im nächsten Sprint so viel Arbeit wie möglich erledigen

Sprint Retrospektive

Dauer



Wenn mein Sprint so lang ist	Sollte mein Sprint-Review-Retrospektive-Meeting nicht länger dauern als
Eine Woche	45 Minuten
Zwei Wochen	1,5 Stunden
Drei Wochen	2,25 Stunden
Vier Wochen	3 Stunden

Fibonacci-Zahlen & Story-Points

Methode zur relative Schätzung

Anforderungen von 1 bis 8 können in einem Sprint untergebracht werden = User Story

Anforderungen von 13 bis 34 fließen in ein Release ein. müssen weiter aufgeschlüsselt werden, bevor sie in einen Sprint einfließen.

Anforderungen von 55 bis 144 sind zu groß für ein Release, spiegeln typischerweise Features wider.

Kriterien für die Größe der einzelnen User Stories sind die Komplexität der Aufgabe, Risiken, Kenntnisse der Teammitglieder und externe Abhängigkeiten

die letzten beiden Zahlen in der Folge werden addiert, um die nächste Zahl zu erzeugen: 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, usw.



Storypoints sind eine Einheit, die die Größe einer User Story beschreiben.



Info

- Ein Story-Point ist die Fibonacci-Nummer, die einer individuellen Anforderung (das heißt einer User-Story) zugeordnet ist.
- Leonardo Pisano Bigollo, auch bekannt als Fibonacci, lebte von etwa 1170 bis zirka 1250 in der Nähe von Pisa, Italien

Schätzmethoden

1. Fist-of-Five-Methode



5 = LOVE IT!

4 = Good idea.

3 = Yeah, I can support it.

2 = I have reservations, let's discuss further.

1 = Opposed. Do not move forward.

2. Planungs-/Schätzpoker





Moderation und Präsentation in Projekten

Präsentation

- » Erzählen Sie eine Geschichte und packen Sie ruhig auch Emotion mit hinein
- » Halten Sie Ihre Präsentation einfach und überladen Sie diese nicht mit Inhalt
- » Arbeiten Sie mit anschaulichen Bildern



- » Eine gute Vorbereitung ist die halbe Miete
- » Orientieren Sie sich an Ihrer Zielgruppe
- » Wer von sich selbst überzeugt ist, überzeugt auch seine Zuhörer
- » Weniger ist mehr

Moderation

Eine gut vorbereitete Moderation spart Zeit und Nerven!

Der Moderator führt die Gruppe

Muss seine eigenen Interessen und Meinungen zurückstellen

Aufgaben eines Moderators:

- Klärung des Auftrags und der Ziele der Moderation
- Erstellung der Dramaturgie
- Organisatorische Vorbereitung der Sitzung
- Einführung in die Thematik
- Setzen der Moderationsinhalte
- Steuerung des Diskussionsprozesses
- Pointierung der Inhalte und inhaltliche Klärung verschwommener Beiträge
- Verantwortung für die Visualisierung und Dokumentation der Ergebnisse

Projektleiter

menschliche Schnittstelle
Schnittstelle zwischen Auftraggeber,
Projektteam und all jenen die vom
Projekt betroffen sind

Diese vier Eigenschaften muss ein Projektleiter haben

- Generalist mit Fachkompetenz. ...
- Feingefühl und Kommunikationsstärke. ...
- Flexibler Menschenkenner. ...
- Motivierend, mutig, leidenschaftlich.

Aufgaben und Tätigkeiten eines Projektleiters

- Zeit- und Budgetplanung.
- Überwachung der Durchführung.
- Protokollierung und Dokumentation der Projektumsetzung.
- Präsentation des Fortgangs vor Entscheidungsträgern.
- Zusammenstellung und Koordination von Teams und **Aufgaben**.

Fachbegriff

Definition Projektmanagement

Als Projektmanagement wird das Initiiieren, Planen, Steuern, Kontrollieren und Abschließen von Projekten bezeichnet.

Definition Projekt

Ein Projekt ist ein zielgerichtetes, einmaliges Vorhaben, das aus einem Satz von abgestimmten, gesteuerten Tätigkeiten besteht und durchgeführt werden kann, um unter Berücksichtigung von Vorgaben wie etwa Zeit, Ressourcen und Qualität ein Ziel zu erreichen.

Definition Projektauftrag

Der Projektauftrag ist ein Dokument, das die Existenz eines Projektes formell bestätigt. Der Projektauftrag ist Teil des Umfangsmanagements und wird in der Regel von einer Managerin oder einem Manager einer höheren Ebene unterzeichnet.

Bei einem **externen** Projekt im Kundenauftrag muss Geld verdient werden. Ein kommerzielles Unternehmen möchte als Auftragnehmer mit dem Projekt einen Gewinn machen, d.h. es soll mehr einbringen als es kostet.



Bei einem **internen** Projekt entstehen unmittelbar keine Einnahmen

Fachbegriff Soll-Ist-Analyse

Die Abweichungsanalyse wird in Teilabweichungen zerlegt. Dabei sind die folgenden Ablaufschritte zu beachten. Zuerst werden die erreichten Ist-Werte den Soll-Werten gegenübergestellt und mit Hilfe der Abweichungsanalyse werden die Korrekturentscheidungen und Gegenmaßnahmen vorgeschlagen. Während des Korrekturprozesses lässt diese Analyse entweder ursprüngliche Unternehmensziele realisieren oder führt zu neuen Unternehmenszielen. Dabei werden die Abweichungen erklärt und somit eine Grundlage für die Korrekturmaßnahmen geschaffen.

Angebot

Mit dem Angebot oder einer Offerte reagiert ein Anbieter auf die Anfrage eines potentiellen Kunden und legt die Bedingungen fest, unter denen er bereit ist, Produkte zu liefern oder Dienstleistungen zu erbringen.

Nutzwertanalyse

bewertet die Nutzwertanalyse die Effektivität bzw. den Outcome. Die Nutzwertanalyse findet u. a. Anwendung im Controlling, im Projektmanagement, in der Volkswirtschaftslehre und sogar im Vergaberecht, eben überall dort wo eine Beurteilung auf Basis mehrerer quantitativer und qualitativer Kriterien, Zielen oder Bedingungen getroffen werden muss. Die NWA ist die „Analyse einer Menge komplexer Handlungsalternativen mit dem Zweck, die Elemente dieser Menge entsprechend den Präferenzen des Entscheidungsträgers bezüglich eines multidimensionalen Zielsystems zu ordnen. Die Abbildung der Ordnung erfolgt durch die Angabe der Nutzwerte (Gesamtwerte) der Alternativen.“

Eine NWA wird häufig erstellt, wenn „weiche“ Kriterien vorliegen, anhand derer zwischen verschiedenen Alternativen eine Entscheidung gefällt werden muss.

Kenntnis Projektkostenplan

Kostenplan ist das Ergebnis der Projektkalkulation. Er gibt an, welche Kosten zu welchem Zeitpunkt für welche Leistungen voraussichtlich anfallen. Der Kostenplan kann als Matrix aus Projektstruktur (vertikale Gliederung) und zeitlichem Ablauf (horizontale Gliederung) dargestellt werden.

Fachbegriff DevOps

Der Begriff **DevOps** setzt sich aus „Dev“ (Development, Entwicklung) und „Ops“ (Operations, Vorgänge) zusammen und vereint Menschen, Prozesse und Technologien, damit Kunden kontinuierlich hochwertige Produkte erhalten.

Personas

- prototypische Beschreibung von Personen, die es allen Beteiligten leichter macht, sich diese Personen vorzustellen und Empathie für sie zu entwickeln
- Kann mithilfe eines fiktiven Steckbriefs dargestellt werden
 - Inhalt: Name, Alter, Familienstand, Beruf, Interessen, Hobbys, typische Probleme, Wünsche oder Bedürfnisse, eventuell Passendes Foto und typisches Motto
- Einfache Methode, um plastisches Bild von Zielgruppe zu erzeugen
- Sicherstellung das Designer, Entwickler, Texter, Agentur an gleichen Typus Mensch denkt bei Erstellung
- Im Steckbrief können bereits Hinweise auftauchen, welche Inhalte oder Funktionen die Seite aus Sicht der Zielgruppe haben sollte damit Bedürfnisse erfüllt sind
- Tools für Gestaltung von Persona-Steckbriefen:
 - <https://xtensio.com/user-persona/>
 - Templates für Darstellung der Eigenschaften
- Bei mehreren Zielgruppen: pro Zielgruppe eine Persona
- Kommt nicht auf Differenziertheit an, sondern dass ein Mensch gezeichnet wird, der als Repräsentant für die Personen steht, für die die Webseite gedacht ist

Fachbegriff

Backlog

Ganz allgemein bezieht sich der Begriff Backlog auf eine Menge an Arbeit, die sich im Laufe der Zeit angesammelt hat oder einen Bestand an Aufträgen, die auf ihre Abarbeitung warten. Im Projektmanagement bezeichnet ein Backlog generell projektbezogene Aufgaben, die noch zu erledigen sind.

Arbeitspaket

Das Arbeitspaket ist ein plan- und kontrollierbares Element in einem Projekt, das nicht weiter untergliedert wird. Arbeitspakete werden innerhalb eines Projektstrukturplans erstellt.

Normen für Projekte

- **internationale Regelwerke:** geben vor nach welchem Schema und Standards Projekte geplant werden
- **Standardisierung von Projektplanung und Umsetzung** für die internationale Zusammenarbeit um Prozesse zu vereinfachen und effektiver zu gestalten.
- **Umsetzung** der Normen **eher in größeren Unternehmen** bei größeren Projekten

DIN-Norm (DIN 69901-5:2009-01)

- Deutsche Industrienorm vom Deutschen Institut für Normung
- Grundlagen, Prozessen, Prozessmodellen, Methoden, Daten, Datenmodellen und den Begriffsdefinitionen des Projektmanagement
- Kostenpflichtig

PMI-Definition (USA Institution: „Project Management Institut“)

- Organisation weltweit tätig
- bemüht sich eigene Standards für das Projektmanagement durchzusetzen & zu etablieren
- Basis bildet dabei der Project Management Body of Knowledge, auch PMBOK-Guide abgekürzt.

ISO-Norm (ISO 21500)

- International gültige Norm für Projekte & Projektmanagement
- für jede Art von Projekten gültig
- Verwendung für alle organisatorischen Einrichtungen
- Alle Eventualitäten der Projektplanung abgedeckt

IPMA (International Project Management Association)

- Deutsche Industrienorm vom Deutschen Institut für Normung
- Regelwerk besteht aus 5 Teilen
- Grundlagen, Prozessen, Prozessmodellen, Methoden, Daten, Datenmodellen und den Begriffsdefinitionen des Projektmanagement
- Kostenpflichtig



Information

Ein Projekt ist ein geplantes Arbeitsprogramm, dessen Fertigstellung eine bestimmte Menge an Zeit, Aufwand und Planung erfordert



Erfolge & Misserfolg von Softwareprojekten

Erfolg und Misserfolg

American Heritage Dictionary definiert Scheitern als "den Zustand des Nicht-Erreichens eines gewünschten Endzustandes", den Erfolg als "das Erreichen von etwas Geplantem, Gewünschtem oder Versuchten"

Üblicherweise wird ein **Softwareprojekt und -produkt** als **erfolgreich** wahrgenommen, wenn es im geplanten Budget- und Zeitrahmen abgeschlossen wurde und die zuvor definierten **Anforderungen zur Zufriedenheit des Auftraggebers umgesetzt** sind

Jeder muss sein Ziel selber definieren und im Endeffekt entscheiden ob dieses Ziel gelungen ist oder nicht, ob es ein Erfolg war oder nicht.

Scheitern des Projektes



Überschreiten bestimmter Zeiträume



Überschreiten eines maximalen Kostenrahmens



Unverzichtbare Mitarbeiter verlassen Projekt



Zeitrahmen nicht einhaltbar



Legen Sie fest, unter welchen Bedingungen man das Projekt als gescheitert betrachten muss.



Finanzrahmen gesprengt



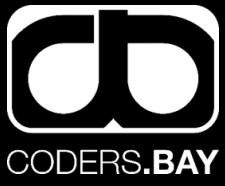
Auftraggeber stellt Projekt aus strategischen Gründen ein



Projekt erweist sich aus sachl. wie techn. Problemen als nicht durchführbar



Auftraggeber geht Insolvenz & kann Fortführung nicht bezahlen



Ende