

Software Engineering

Urheberrechtlich geschütztes Skript

Rechtsvermerk:

Dieses Dokument ist urheberrechtlich geschützt und beinhaltet Informationen des aus dem Unternehmen des Lehrenden. Das Dokument ist als begleitendes Lehrmaterial für die Kursteilnehmer (Studierenden) an einer Hochschule freigegeben.

Das Dokument darf im Sinne der einfachen Nutzungsrechte lediglich für die private Weiterbildung und als Lernunterlage zur Prüfungsvorbereitung verwendet werden.

Eine Verwertung, Vervielfältigung, Veröffentlichung oder Weitergabe – auch in Auszügen – ist nicht erlaubt und bedarf der ausdrücklichen Zustimmung des Urhebers.

Die Angabe/Verwendung der Quellen je Folie sind im Original PowerPoint-Dokument dieses Skripts im Notizenbereich vermerkt. In der PDF-Ausfertigung dieses Skripts sind die Quellen je Folie nicht ablesbar. Auf Anfrage werden die Quellen bereitgestellt.

Eigenschaften / Merkmale

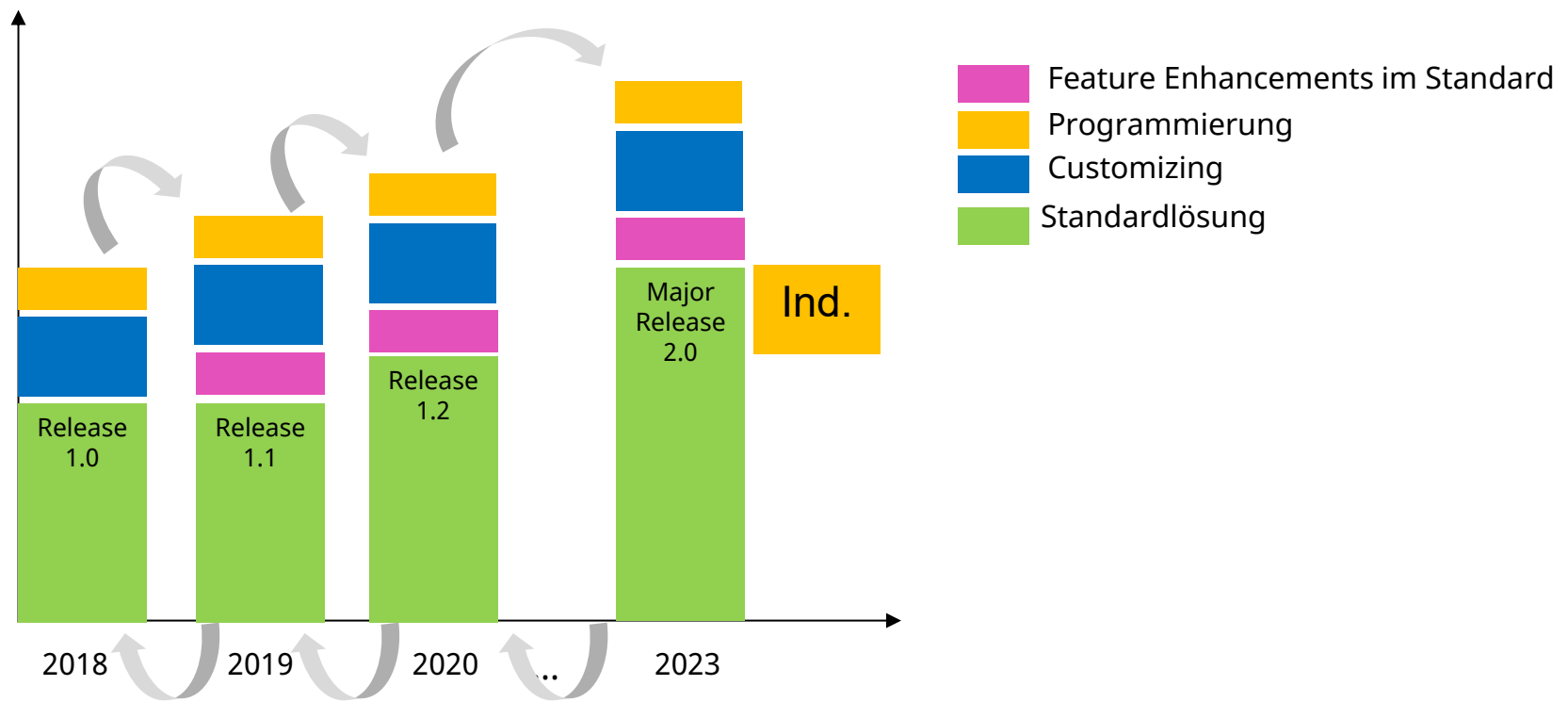
- Stark individualisierte Unternehmensprozesse erfordern eine individuelle Software
- Hat den Anspruch einer maßgeschneiderten SW-Lösung
- Soll die Alleinstellungsmerkmale des Unternehmens stabilisieren und individuelle Verfahrensweisen abbilden
- Wird in Unternehmen i.d.R. zur Abbildung der primär wertschöpfenden Aufgaben eingesetzt
- Erfordert eine durchdachte **Systemvision** und detaillierte Festlegung der Anforderungen
- Die Anforderungserhebung basiert auf einer fundierten **Geschäftsprozessanalyse**
- Oftmals wird eine hohe Administrierbarkeit der SW erwartet
- Die Weiterentwicklung wird durch den Auftraggeber angestoßen

Eigenschaften / Merkmale

- Deckt den Branchenstandard ab und orientiert sich an den Bedürfnissen des Markts
- Ist erprobt und bereits qualitätsgesichert
- Wird in Unternehmen i.d.R. zur Abbildung der sekundär wertschöpfenden Aufgaben eingesetzt
- Stärkung der Alleinstellungsmerkmale steht nicht im Vordergrund
- Geeignet für stark standardisierbare Prozesse und Aktivitäten
- Individualität bei der Ausgestaltung der Software spielt eine geringe Rolle
- Anpassungsbedarf wird über Customizing abgedeckt
- Nutzer profitiert von der kontinuierlichen Weiterentwicklung des Anbieters (sustaining Innovation)
- Standardsoftware ist meist das Ergebnis einer Produktentwicklung

Kriterien bei der Auswahl einer Standardsoftwarelösung

- Releasefähigkeit
- Releasehoheit



Gegenüberstellung Open Source vs. Proprietäre Lösungen

Open Source	Proprietär
Quellcode offen	Quellcode „verschlossen“ (oftmals hauseigene Programmiersprache)
Flexibel anpassbar	Stark begrenzte Anpassbarkeit (80% Customizing)
Weiterentwicklung über Community	Weiterentwicklung durch Hersteller
Lizenzkostenfrei	Lizenzkostenpflichtig
Eigenverantwortlicher Betrieb	Trend geht zu Cloudbetrieb SaaS → Gefahr des Vendor-Lock-Ins
Stabilität abhängig von eigenen Kompetenzen und Aufwand	Stabilität hoch, da erprobt und Verantwortung beim Hersteller

Praxisauszug

Was die EU-Kommission prüft

In einer vorläufigen Bewertung des Falls legt die Kommission ihre wettbewerbsrechtlichen Bedenken dar. Demnach verlange der deutsche Konzern von seinen Kunden, die Software nur von SAP warten zu lassen. Zudem müssten Kunden dieselbe Art von Wartungs- und Supportleistungen zu denselben Preisbedingungen wählen. Dies könne Kunden daran hindern, Wartungs- und Supportleistungen verschiedener Anbieter zu unterschiedlichen Preis- und Supportniveaus zu kombinieren, obwohl dies für sie günstiger wäre. Auch wirft die EU-Kommission SAP vor, dass Kunden Wartungs- und Supportleistungen für ungenutzte Softwarelizenzen nicht kündigen können – sie also für ungewünschte Leistungen weiterzahlen müssten. Kunden, die nach einer Auszeit wieder Dienstleistungen von SAP abonnieren, müssten außerdem eine Wiederaufnahmegebühr und Nachzahlungen leisten, heißt es weiter. Die Brüsseler Behörde verdächtigt das Unternehmen daher, seine Kunden auszubeuten und Drittanbieter im Wettbewerb unlauter einzuschränken.

Handelsblatt Artikel vom 25.09.2025

[Quelle](#)

Anforderungsdokumentation

Zentrale Bedeutung von Anforderungen und deren Dokumentation

- Dienen als Ausgangspunkt für nachfolgende Phasen
 - Systementwicklung
 - Testing/ Abnahme
- Anforderungen sind rechtlich relevant (verbindlich im Sinne des Vertrags und können eingefordert werden)
- Anforderungen sind komplex (Umfang und Vernetzung)
- Anforderungen sollen allen Beteiligten zur Verfügung stehen

Anforderungsdokumentation

Verwendung von Anforderungsdokumenten

- Planung
 - Definition der Arbeitspakete und Meilensteine für die Systemrealisierung
 - Projekt- und Ressourcenplanung
 - Input für die Aufwandsschätzung
 - Anforderungen sind oftmals Teil der Verträge (Festpreis)
- Architekturentwurf
 - Grundlage für den Architekturentwurf
- Implementierung
 - Auf Basis des Architekturentwurfs werden die Anforderungen umgesetzt

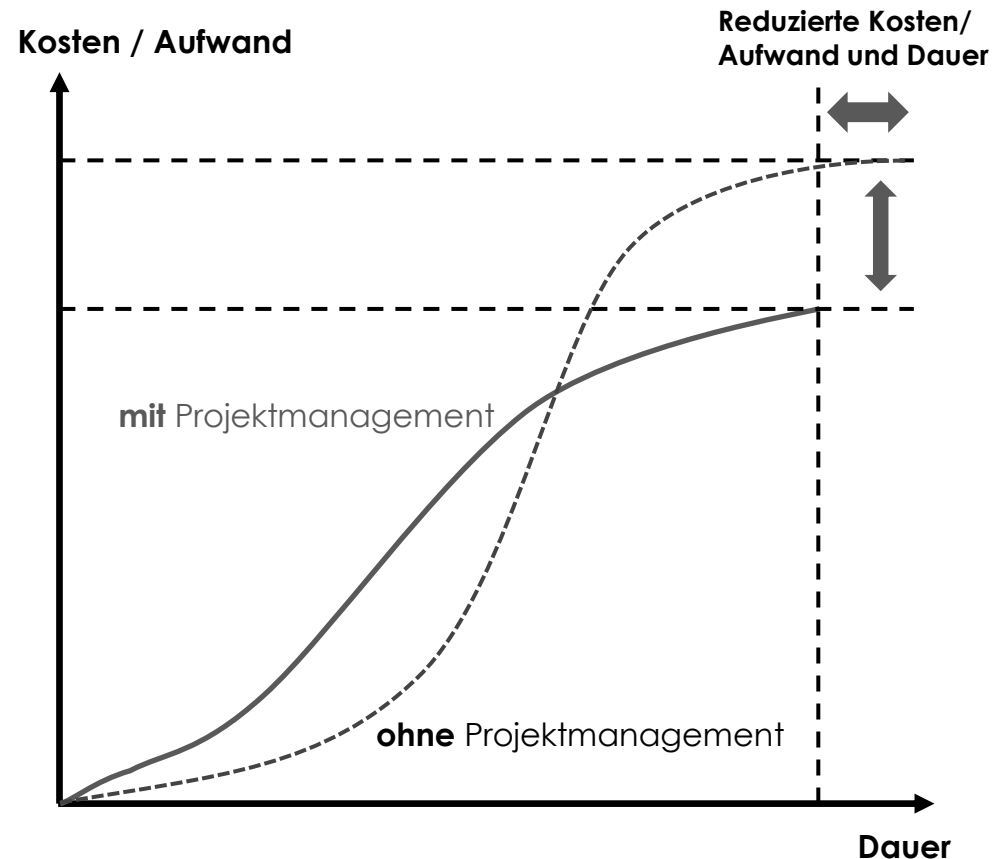
Anforderungsdokumentation

Verwendung von Anforderungsdokumenten

- Testing
 - Input zur Erstellung der Testfälle
 - Definition der Abnahmekriterien
 - Beschaffung valider Testdaten
- Änderungsmanagement
 - Basis für Abschätzung zu Aufwand und Risiko von Änderungsanforderungen
- Systemnutzung/-wartung
 - Dient zur Voranalyse, ob es sich um einen Anwendungs- oder ggf. Bedien- oder Implementierungsfehler handelt

Gründe für Projektmanagement

- Geringere Projektkosten
- Weniger Zeitbedarf und bessere Einhaltung der Vertragstermine sowie interner Zwischentermine
- Frühe Erkennung von Planabweichungen
- Bessere Zusammenarbeit zwischen den Projektbeteiligten
- Erreichen der aufgestellten inhaltlichen Ziele

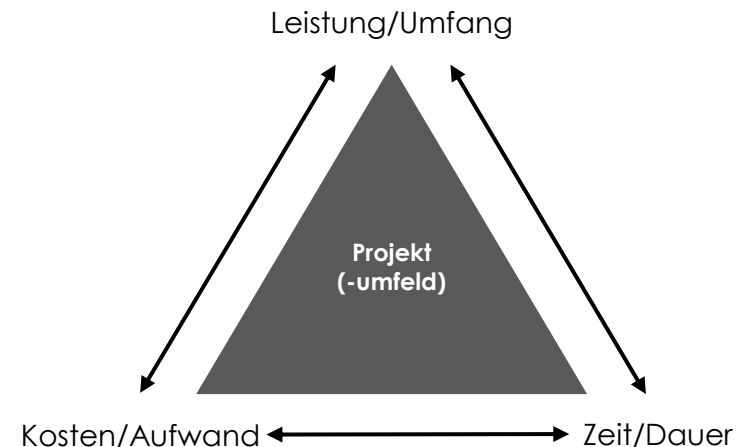


Ziele des Projektmanagements

- **Planbarkeit von Projekten:** Durch effektives Projektmanagement werden Unsicherheiten reduziert und Projekte vorhersagbar, wodurch sich Risiken besser managen lassen.
- **Erleichterte Führbarkeit:** Klare Strukturen und definiertes Vorgehen verbessern die Führungseffizienz und erleichtern die Steuerung komplexer Projekte.
- **Steigerung der Produktivität:** Optimierung des Vorgehens in Projekten führt zu einer erhöhten Effizienz und Produktivität.
- **Entwicklungsmöglichkeiten für Mitarbeiter:** Projekte bieten Mitarbeitern Gelegenheiten, neue Fähigkeiten zu erlernen und vorhandene Kompetenzen durch die Übernahme verantwortungsvoller Aufgaben zu erweitern.
- **Reduzierung und Transparenz von Projektkosten:** Durch detaillierte Kostenplanung und -kontrolle werden Projektkosten transparenter und lassen sich effektiver reduzieren.
- **Verkürzung der Projektdauer:** Effizientes Projektmanagement minimiert Zeitverluste durch optimierte Abläufe und verbesserte Koordination aller Beteiligten.
- **Erhöhung der Objektqualität:** Qualitätsmanagement als integraler Bestandteil des Projektmanagements sichert und steigert die Qualität der Projektergebnisse.

Zielkonflikte im Projektmanagement

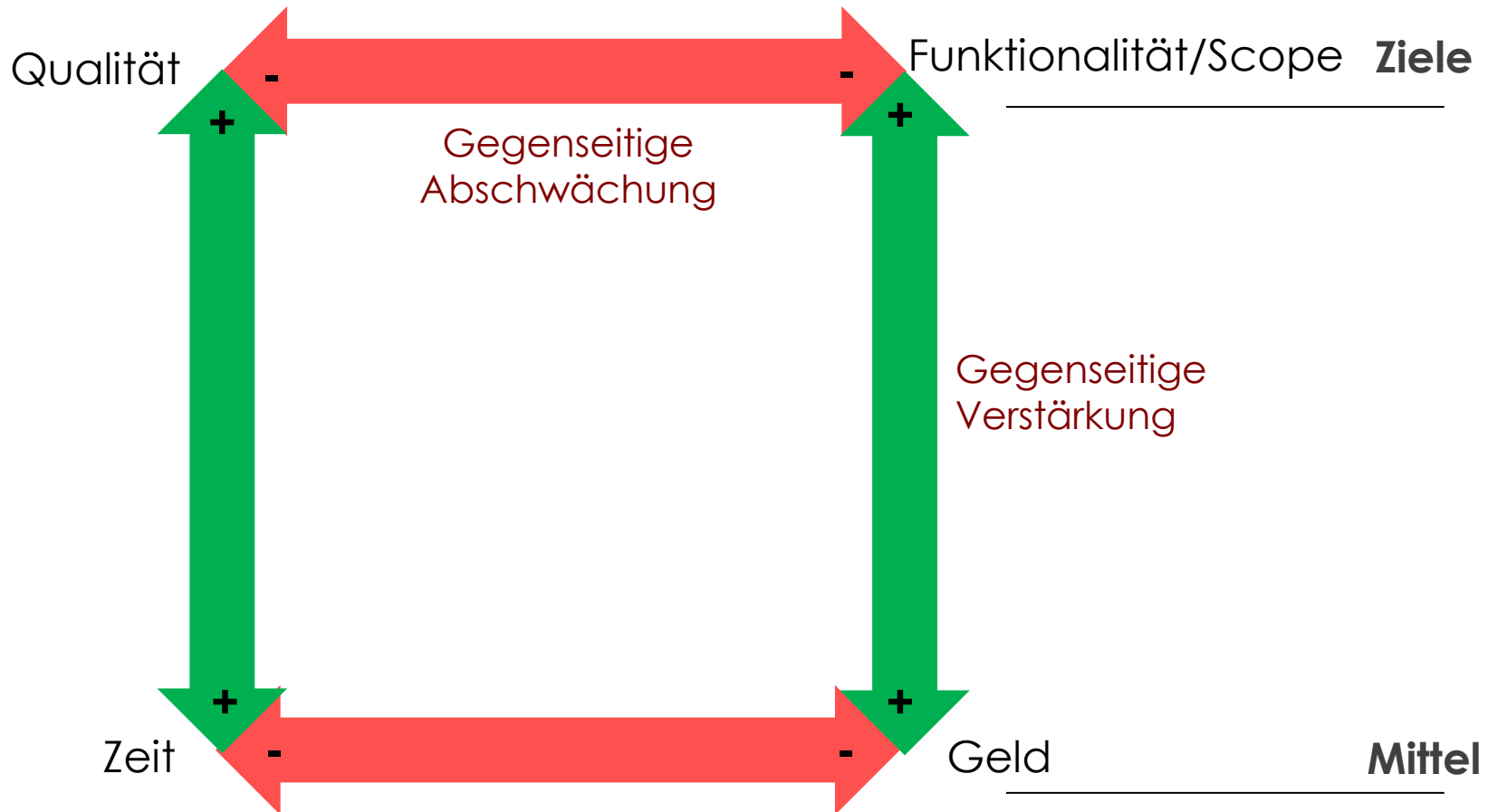
- Konstantes Spannungsverhältnis zwischen Zeit, Kosten und Leistung erfordert ständige Abwägungen im Projektmanagement
- Wechselwirkungen zwischen Zielen zeigen, dass Änderungen in einem Zielbereich wie Zeit, Kosten oder Leistung direkte Auswirkungen auf andere Bereiche haben und oft Kompromisse erfordern
- Balance der Projektziele ist Hauptziel effektiven Projektmanagements, um Projekterfolg zu sichern
- Das sog. „Magische Dreieck“ als Darstellung wesentlicher Projektziele unterstreicht die Bedeutung der wechselseitigen Abhängigkeiten
- Erhalt der optimalen Balance zwischen den Parametern ist eine kontinuierliche Herausforderung, die eine ständige Anpassung und Feinabstimmung der Projektmanagementstrategien erfordert



Bei der Planung und Steuerung von Projekten stehen vier Zielgrößen im besonderen Blickpunkt des Auftraggebers und der Projektleitung

- Erreichen einer bestmöglichen **Qualität** in der Projektausführung sowie bei den Projektergebnissen
- Erreichen bzw. Sicherung des geplanten Leistungsumfangs (**Quantität/Scope**) im Abschluss des Projekts
- Erreichen einer möglichst kurzen **Projektdauer**
- Erreichen möglichst niedriger **Projektkosten**

Das magische Viereck



Kosten

- Das Projekt wird mit einem festen Budget gestartet
- Dieses Budget wird nachträglich nicht mehr gekürzt. Auch bei Anbietern von Festpreisprojekten wird in der Regel nicht während der Projektlaufzeit das für das Projekt verfügbare Budget gekürzt, um einen höheren Gewinn zu erzielen
- Das Budget wird nicht ohne Grund erhöht

Zeit

- Das Projekt wird mit einem festen Zeitrahmen gestartet. Oft gibt es einen fixen Zieltermin
- Oft kommt es zu einer faktischen Kürzung der Projektlaufzeit, wenn sich der Projektstart verzögert, z. B. aus organisatorischen Gründen wie fehlenden Ressourcen oder nicht freigegebenen Budgets

Qualität

- Ist der am schwierigsten zu messende Faktor und ist nicht so klar vorgegeben wie Budget und Zeit
- Qualitätskriterien müssen im Projekt konkretisiert werden. Über einzelne, konkrete Punkte kann dann ggf. gesprochen werden, ob dieses Qualitätskriterium gehalten werden muss

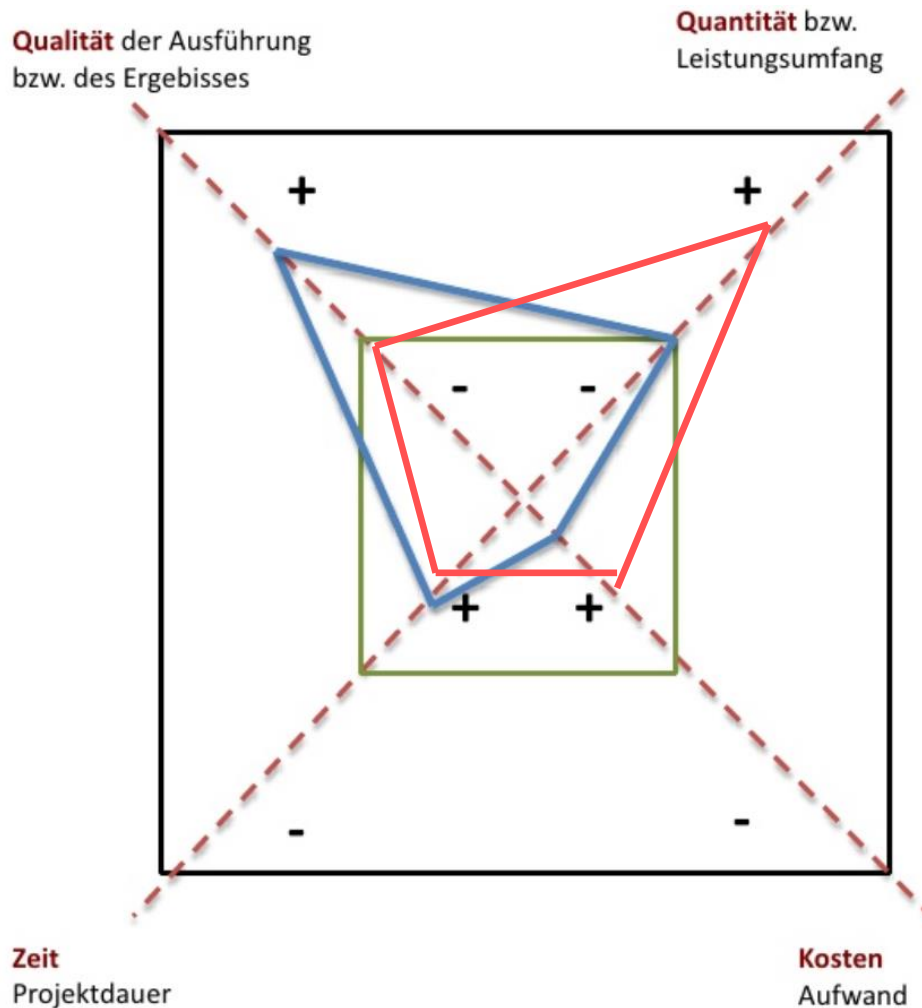
Leistung/Scope

- Der Leistungsumfang tendiert während eines Projekts dazu, immer größer zu werden. Ursache dafür sind zu Projektbeginn zwangsläufig unscharfe Anforderungen, bei deren Konkretisierung immer neue Implementierungsaspekte entstehen können
- In konkreten Punkten lässt sich der Leistungsumfang auch im Projektverlauf kürzen. Dabei verschwimmt die Abgrenzung zu den Qualitätsmerkmalen

Beispiele

- Soll zum Beispiel die Dauer D des Vorhabens – bei gleichbleibender „Qualität“ und „Quantität“ – verkürzt werden, dann führt dies i.d.R. zu steigendem Aufwand (Kosten), da eine Verkürzung der Projektdauer i.d.R. nur mit steigenden Ressourceneinsatz möglich ist
- Soll dagegen eine höhere Qualität im Endergebnis – bei gleichbleibender „Quantität“ - erreicht werden, dann führt dies konsequenterweise zu einer Verlängerung der Projektdauer und zu steigendem Aufwand in der Projektrealisierung

Das Teufelsquadrat nach Harry Sneed



- Grünes Quadrat entspricht der Planung
- Blaues „Quadrat“ beschreibt den Projektabschluss: Die höhere Qualität bei gleichbleibender Quantität hat sich in gestiegenen Kosten und längere Projektdauer niedergeschlagen

Das Teufelsquadrat nach Harry Sneed - Beschreibung

Zielgrößen auf den Diagonalen:

- Zeit: Projektlaufzeit
- Kosten: Budget
- Qualität: z. B. Funktionalität, Nutzbarkeit, Wartbarkeit
- Leistungsumfang: Anzahl ausgelieferter Funktionen

Die Fläche (Produktivität) eines Projekts ist invariant

- Die aufgespannte Fläche kann verschoben werden, wirkt sich aber unmittelbar auf die anderen Dimensionen aus
- Wenn ein Projekt z. B. in weniger Zeit und zu geringeren Kosten abgeschlossen werden soll, verringern sich auch Leistungsumfang und Qualität

Das Teufelsquadrat nach Harry Sneed - Interpretation

Je weiter außen, desto besser das Ergebnis:

- „Mehr“ Qualität und Leistung führen zu besserem Ergebnis (daher „plus“ außen)
- Kürzere Entwicklungsdauer und geringere Kosten führen zu besserem Ergebnis (daher „minus“ außen)

„Chinesenprinzip“

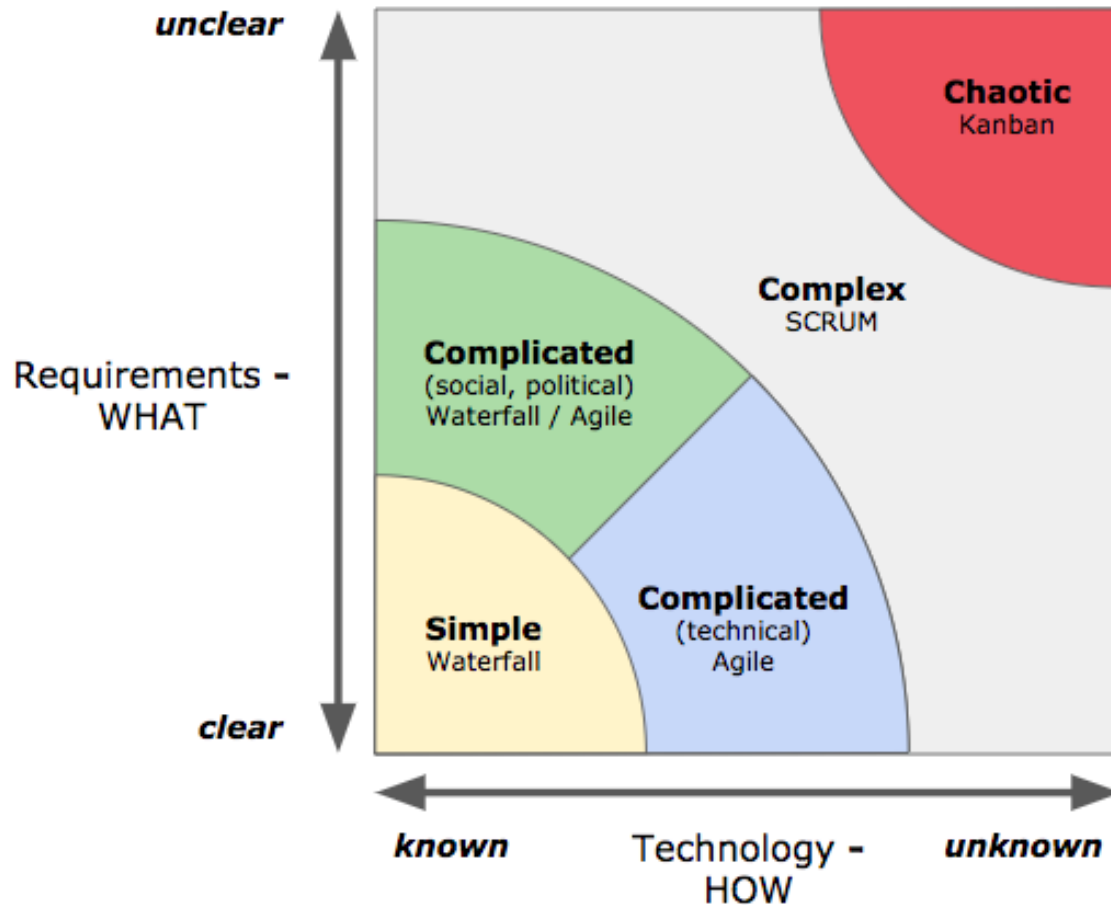
- Idee: Ein Projekt wird beschleunigt, indem massiv Personen in das Projekt entsandt werden
- Funktioniert nur bei stark parallelisierbaren, voneinander unabhängigen Tätigkeiten, die keine größere Einarbeitung erfordern

Phasen, die ein Projekt durchläuft – organisatorisches Versagen

1. **Enthusiasmus** („wir schaffen das locker“)
2. **Desillusionierung** („wir haben uns überschätzt“)
3. **Panik** („wie kriegen wir den Wildwuchs wieder in den Griff“)
4. **Massenflucht der Verantwortlichen** („es warten neue Projekte auf mich, ich bin nicht mehr verantwortlich...“)
5. **Suche nach den Schuldigen** („wer hat's verursacht?“)
6. **Bestrafung der Unschuldigen** („ein Sündenbock muss gefunden werden!“)
7. **Ruhm und Ehre für die Unbeteiligten** („Ruderherumreisser“)

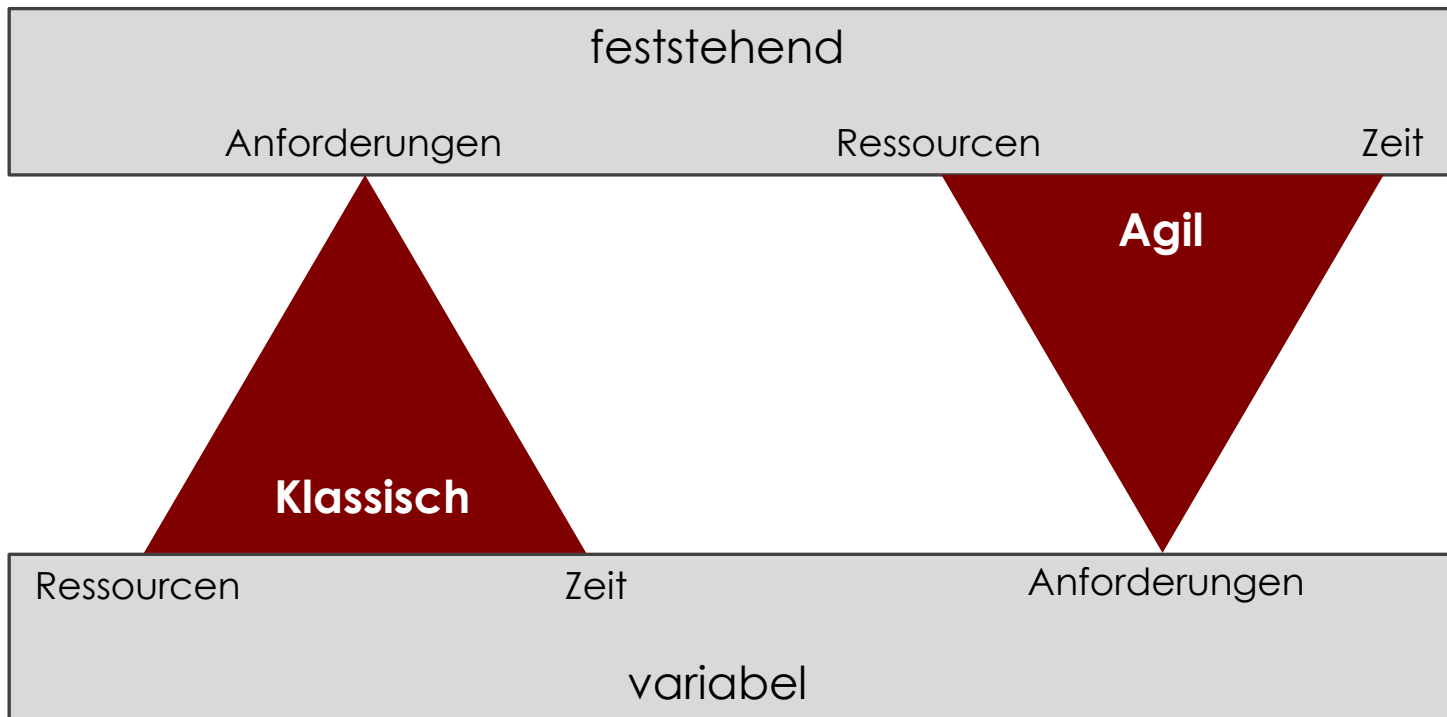
Vorgehensmodelle / Frameworks

- Wasserfall
- Scrum



<https://www.agile-minds.com/wp-content/uploads/2019/03/image-1.png>

Zielausrichtung der Vorgehensmodelle



Wasserfallmodell - Grundidee

- Linear
- Nicht iterativ
- Jede Phase hat definierte Ziele sowie Start- und Endpunkte (sog. Meilensteine)
- Wichtige Dokumente
 - Lastenheft
 - Pflichtenheft
- Erweitertes Wasserfallmodell



Wasserfallmodell - Grundidee

- Jede Phase wird dokumentiert
- Die Dokumentation ist das Ziel
- Reihenfolge der Aktivitäten bzw. Phasen steht fest
- Einfach und verständlich
- Geringer Aufwand für das Management
- Benutzer/Auftraggeber sind hauptsächlich in der Analyse- und Designphase eingebunden
- Wechsel der Phase nur möglich, wenn alle Aktivitäten abgeschlossen sind

Wasserfallmodell - Vorteile

- Klare Abgrenzungen
- Einfach zu planen und kontrollieren
- Leicht verständlich
- Einsatz ohne aufwändige Schulungen möglich
- Sehr effektiv, wenn
 - Die Kosten klar sind
 - Der Umfang klar ist
- Größe des Teams spielt keine Rolle
- Je größer das Team, desto schwieriger die Synchronisation
- Frühe Verfügbarkeit von Gesamtbudget und Funktionsumfang

Wasserfallmodell - Nachteile

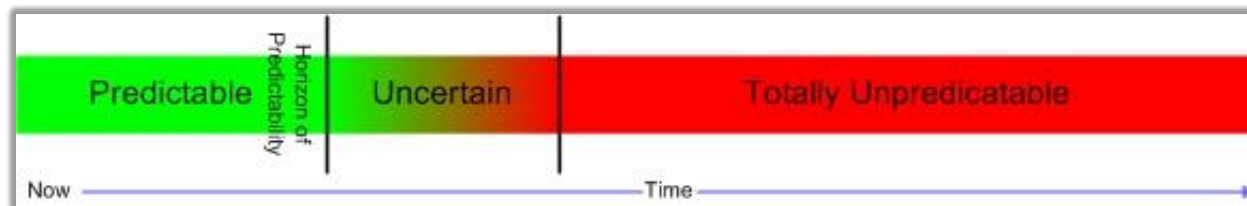
- Problem der Abgrenzung
- Nur für einfache Projekte
- Unflexibel gegenüber Änderungen
- Fehler werden nicht oder spät erkannt
- Rückschritte in der Praxis nicht vermeidbar
- Gefahr durch Verzettelung in den frühen Phasen
- Ergebnisse sind erst am Projektende verfügbar
- Anwender sind in den späteren Phasen nicht mehr beteiligt

Wasserfallmodell - Fazit

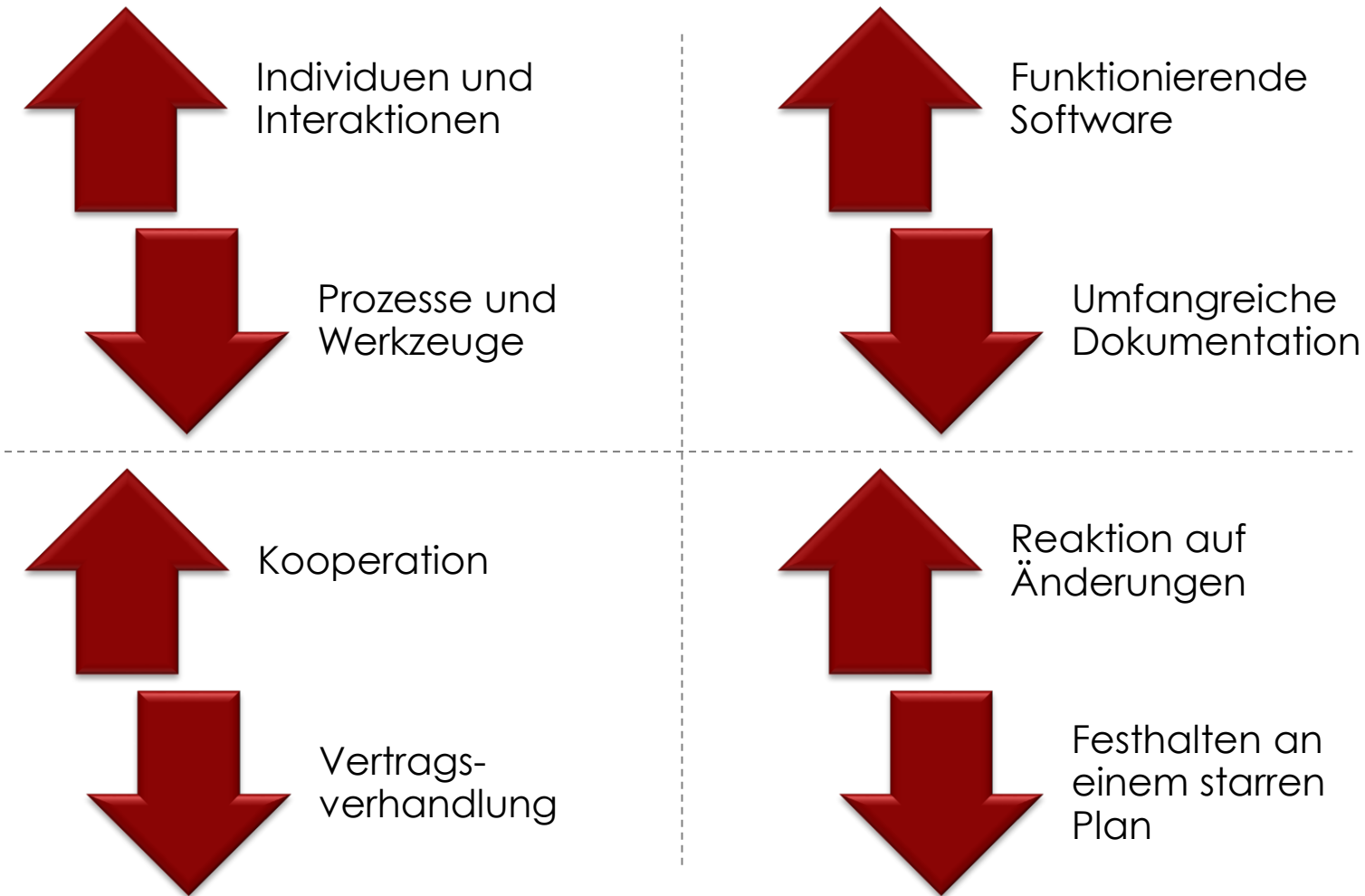
- Nachteile überwiegen, da
 - Im Projektalltag auf Änderungen reagiert werden muss
 - Kein Projekt (besonders in der Softwareentwicklung) schon zu Beginn definiert werden kann
 - Auch Rückschritte im erweiterten Wasserfallmodell nicht reichen, da diese Phasen erneut abgeschlossen werden müssen
 - Einbindung und Kommunikation mit wichtigen Stakeholdern fehlt (Kunde, Anwender)
 - Zeitliche Verzögerungen in den ersten Phasen das gesamte Projekt verzögern

Agile Vorgehensmodelle

- Konsequenz sind steuernde Verfahren
 - sog. agile Vorgehensmodelle
- Reduzierter Planungsaufwand zu Beginn
- Geringerer Dokumentationsaufwand
 - So viel Dokumentation wie nötig
 - So wenig Dokumentation wie möglich
- Horizon of Predictability



Agile Vorgehensmodelle – Paradigmen



Agile Vorgehensmodelle – Terminologie

■ Vision

- Kurze Beschreibung des Projekts (500-700 Wörter)
- Um was handelt es sich?
- Wer führt das Projekt durch?
- Wer gehört zur Zielgruppe?
- Was ist das Alleinstellungsmerkmal?

■ User Story

- Funktionale oder nichtfunktionale Anforderung
- Nicht detailliert, sondern soll Diskussion über Anforderung anregen
- Jede User Story stellt einen bestimmten Mehrwert dar
- Jede User Story ist aus Sicht des Anwenders beschrieben
- Jede User Story hat einen bestimmten Aufwand

Agile Vorgehensmodelle – Terminologie

■ Story Point

- Repräsentieren den Aufwand der User Story
- Konsolidiert Risiko, Wissen, Komplexität, Zeit, Qualität und Ressourcen

■ Epic

- Repräsentiert eine umfangreiche User Story
- Dient als Platzhalter für User Stories, deren Umsetzung in der Zukunft liegt

■ Product Backlog

- Enthält alle User Stories und Epics eines Projekts
- Einträge sind grobgranular
- Priorisiert

■ Task

- Entsteht aus einer User Story
- Ist eine konkrete Aufgabe
- Aus einer User Story werden mehrere, detailliert beschriebene Tasks

Agile Vorgehensmodelle – Terminologie

■ Definition of Done

- Beschreibt, wann ein Task abgeschlossen ist
- Kann je nach Projekt unterschiedliche Ausprägungen annehmen
- Meistens zählen hierzu auch erfolgreiche Tests

■ Definition of Ready

- Beschreibt, wann ein Task ausreichend spezifiziert ist, um in die Umsetzung übernommen zu werden

■ Sprint

- Synonym für Iteration
- Meistens zwei- bis vierwöchig
- Sprints haben eine festgelegte Dauer

■ Sprint Backlog

- Enthält alle Tasks für einen Sprint
- Team verpflichtet sich dazu, alle Tasks im nächsten Sprint umzusetzen
- Sprint Backlog darf sich nicht verändern

Agile Vorgehensmodelle – Rollen

■ **Product Owner**

- Schnittstelle zwischen Team und Kunde
- Füllt das Product Backlog mit User Stories
- Priorisiert das Product Backlog je nach Mehrwert und Aufwand

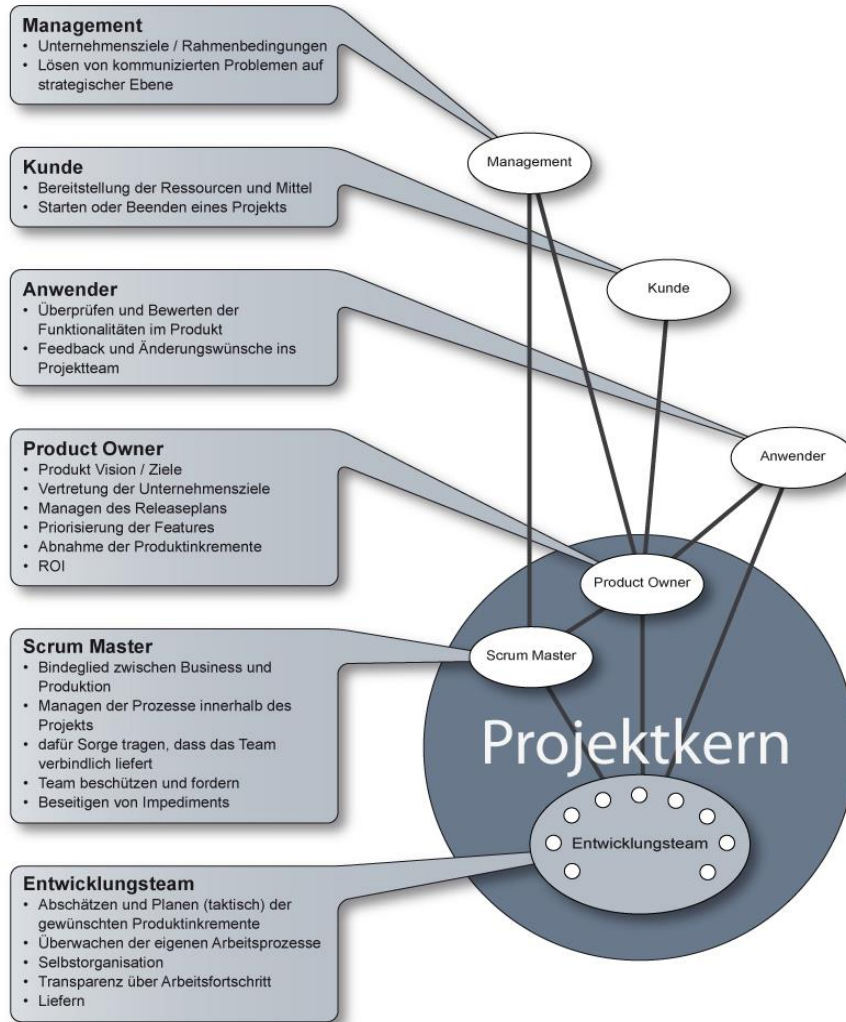
■ **Impediment Backlog**

- Besteht aus sog. Blocks
- Schwierigkeiten, die das Team daran hindern den Aufgaben nachzugehen

■ **Scrum Master**

- Sorgt dafür, dass der Scrum Prozess korrekt eingehalten wird
- Unterstützt Team und Product Owner gleichermaßen
- Moderator

Agile Vorgehensmodelle – Rollen



■ Product Owner

- Stimme des Kunden
- Verfügungsgewalt

■ Team

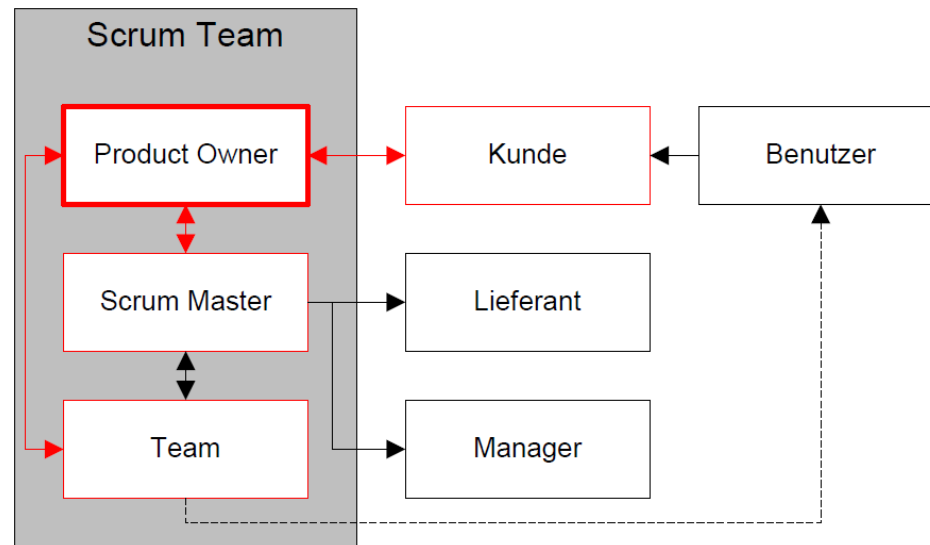
- Selbstorganisiert
- Unbeeinflusst

■ Scrum Master

- Scrum Prozess
- Normales Teammitglied

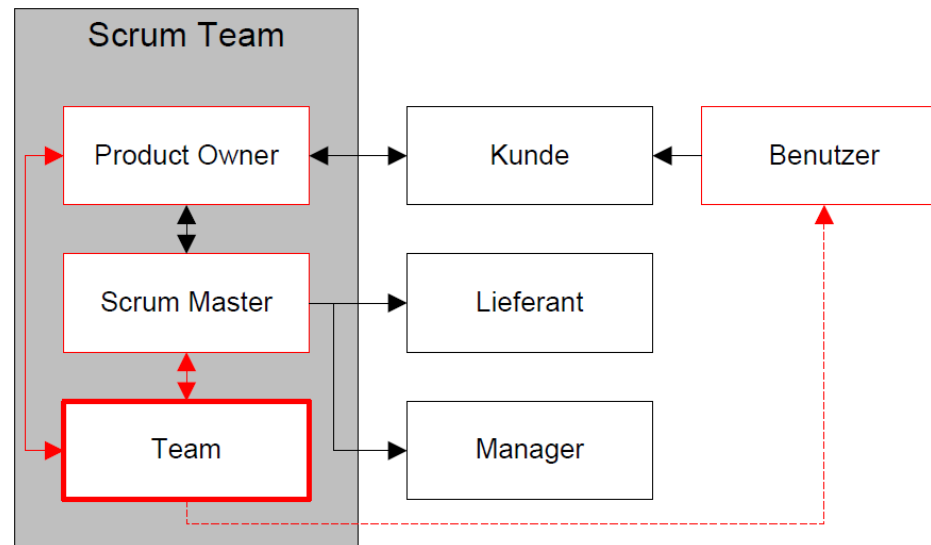
Agile Vorgehensmodelle – Product Owner (Verantwortung)

- Erstellt Epics und User Stories
- Verwaltet das Product Backlog
 - Legt Prioritäten fest
 - Legt Business Value fest
- Respektiert Eigenorganisation des Teams
- Darf das Sprint Backlog nicht beeinflussen



Agile Vorgehensmodelle – Team (Verantwortung)

- Ideale Größe 7 ± 2
- Aufwandsschätzungen
- Entscheidet selbst, welche User Stories im nächsten Sprint umgesetzt werden
- „Committed“ sich für den nächsten Sprint auf einen definierten Inhalt
- Wird während des Sprints nicht gestört



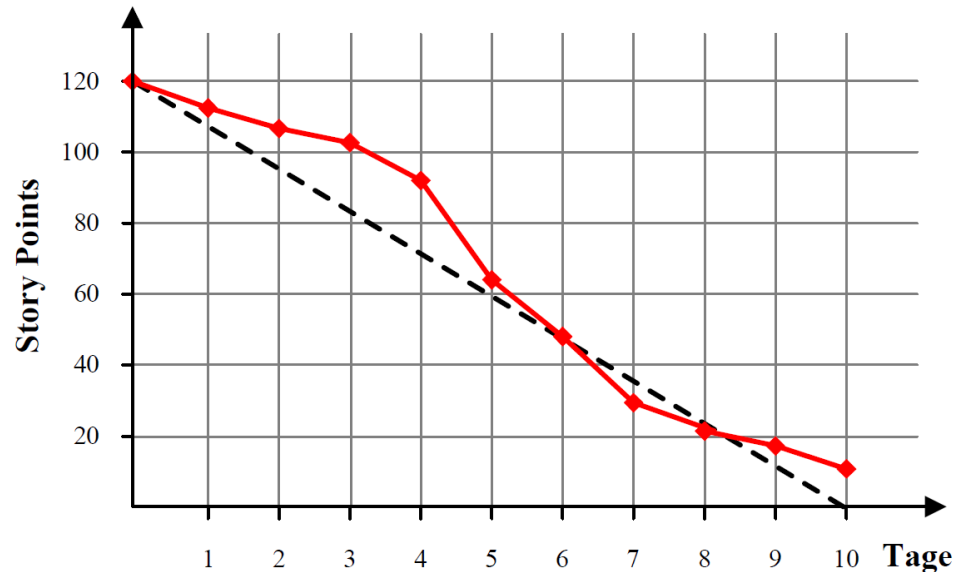
Product Backlog

- Offen
- Frei veränderbar
- Inhalte
 - Sprint
 - Priorität
 - Name
 - Status
 - Aufwandsschätzung
 - Business Value



Burndown Chart und Velocity

■ Burndown-Chart



■ Velocity

- Anzahl an Story Points die ein Team während des Sprints umsetzen kann

User Story und Task

- **INVEST** – bezieht sich auf die User Stories
 - Independent, d.h. unabhängig von anderen User Stories (atomar)
 - Negotiable, d.h. die User Story ist verhandelbar und nicht in Stein gemeißelt
 - Valuable, d.h. die User Story generiert einen Mehrwert
 - Estimable, d.h. die User Story ist geschätzt (und abschätzbar)
 - Small, d.h. die User Story ist nicht zu groß
 - Testable, d.h. es kann überprüft werden, ob die User Story umgesetzt ist
- **SMART** (nach Gloger) – bezieht sich auf die Tasks
 - Specific, d.h. der Task ist konkret spezifiziert
 - Measureable, d.h. es ist messbar, ob der Task erfolgreich beendet wurde
 - Attractive, d.h. der Task ist herausfordernd und anspruchsvoll
 - Realistic, d.h. es ist möglich den Task zu beenden
 - Time-boxed, d.h. es gibt einen klaren Zeitrahmen

Visionsplanung, Roadmap und Releaseplanung

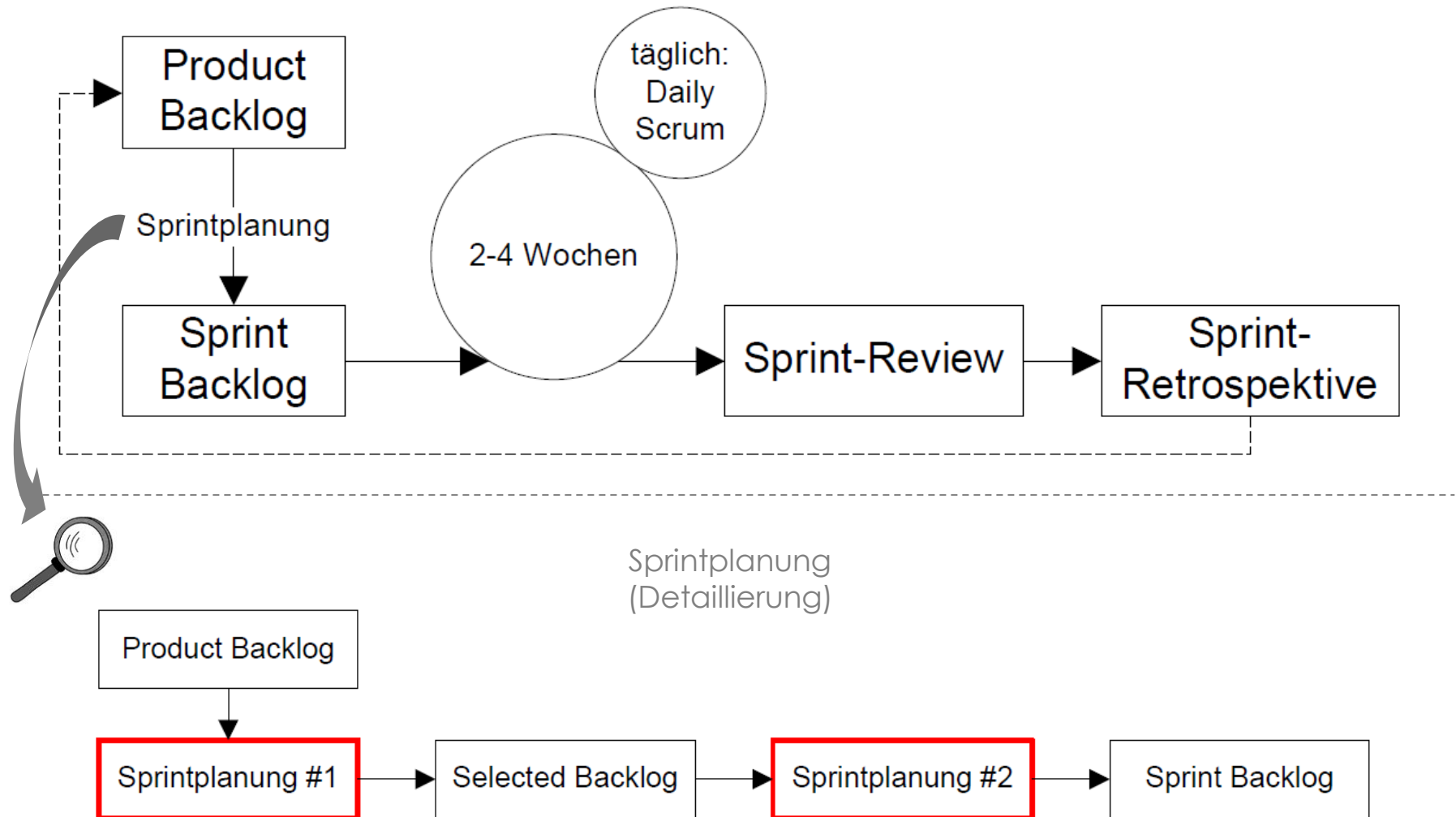
■ Visionsplanung

- »I believe that this nation should commit itself to achieving the goal, before this decade is out, of landing a man on the moon and returning him safely to the earth.«
- Identifizierung der Mitarbeiter ist ein wichtiger Aspekt

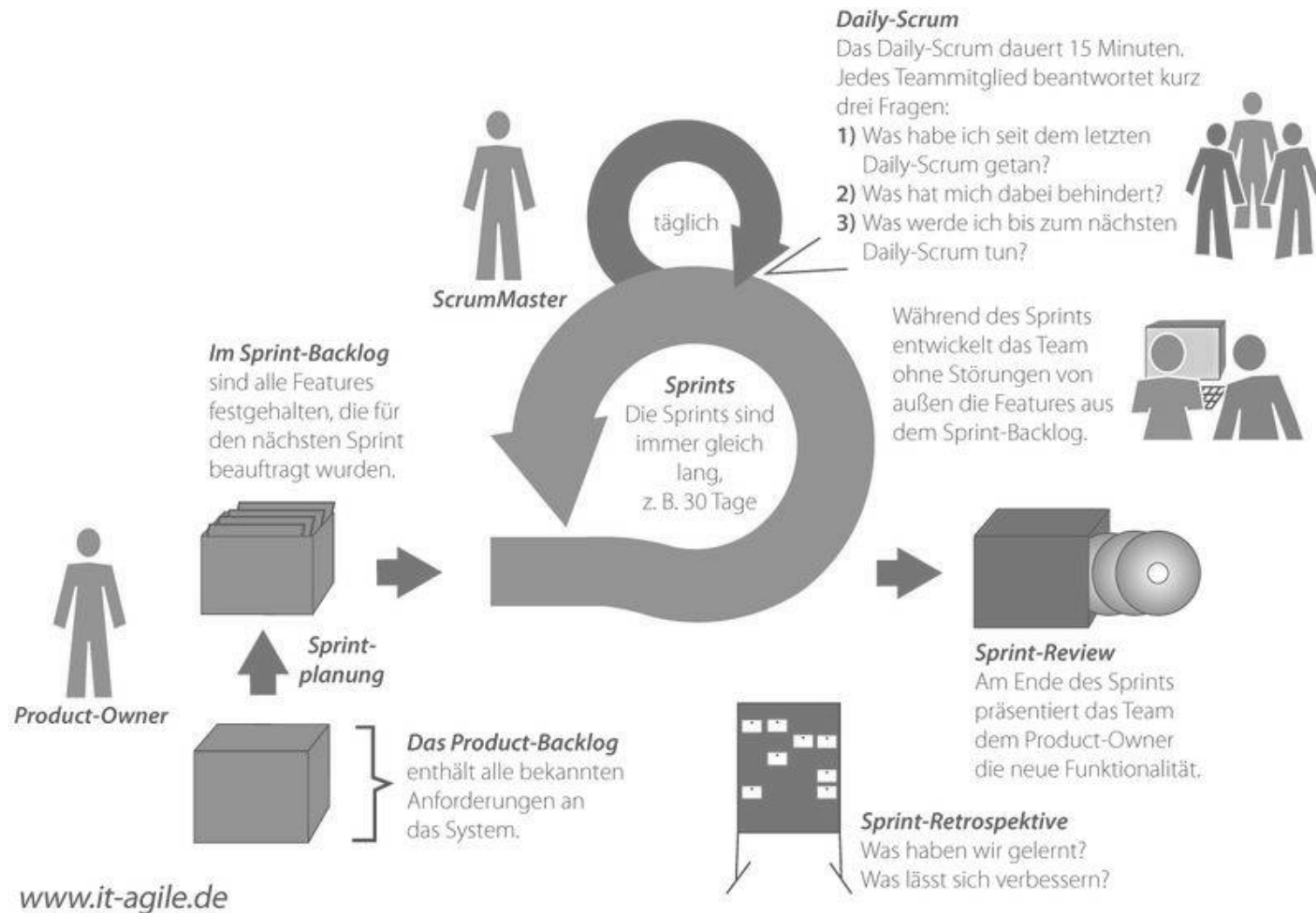
■ Roadmap- und Releaseplanung

- Themen- oder Datumsreleases
- Verantwortlich: Product Owner
- Orientiert sich an der Velocity

Sprintplanung



Scrum Prozess



Sprintplanung

■ Daily-Scrum Meeting

- Bist du gestern mit dem fertig geworden, was du dir vorgenommen hast?
- Welche Aufgaben machst du bist zum nächsten Meeting (morgen)?
- Gibt es ein Problem, das dich an der Weiterarbeit hindert?

■ Fakten

- Max. 20 Minuten
- Moderiert, nicht geleitet
- Keine Problemlösung
- Impediments pflegen (Hindernisse, die beim Scrum Master angesiedelt sind)
- Burndown-Chart
- Sprint Backlog

Ergebnis

- Intensive, tägliche Kommunikation ohne hohen Zeitaufwand
- Koordination ohne permanente Führung
- Eigenorganisation durch Ausgleich zwischen Freiheit und Vorschrift
- Vertrauen basierte Zusammenarbeit

Klassische und agile Vorgehensmodelle – ein Widerspruch?

