

# Software Engineering in der industriellen Praxis

Modul 10: Large-Scale Project Management  
Christian Schmitz

## Referent



**Christian Schmitz**  
**Geschäftsbereichsleiter**

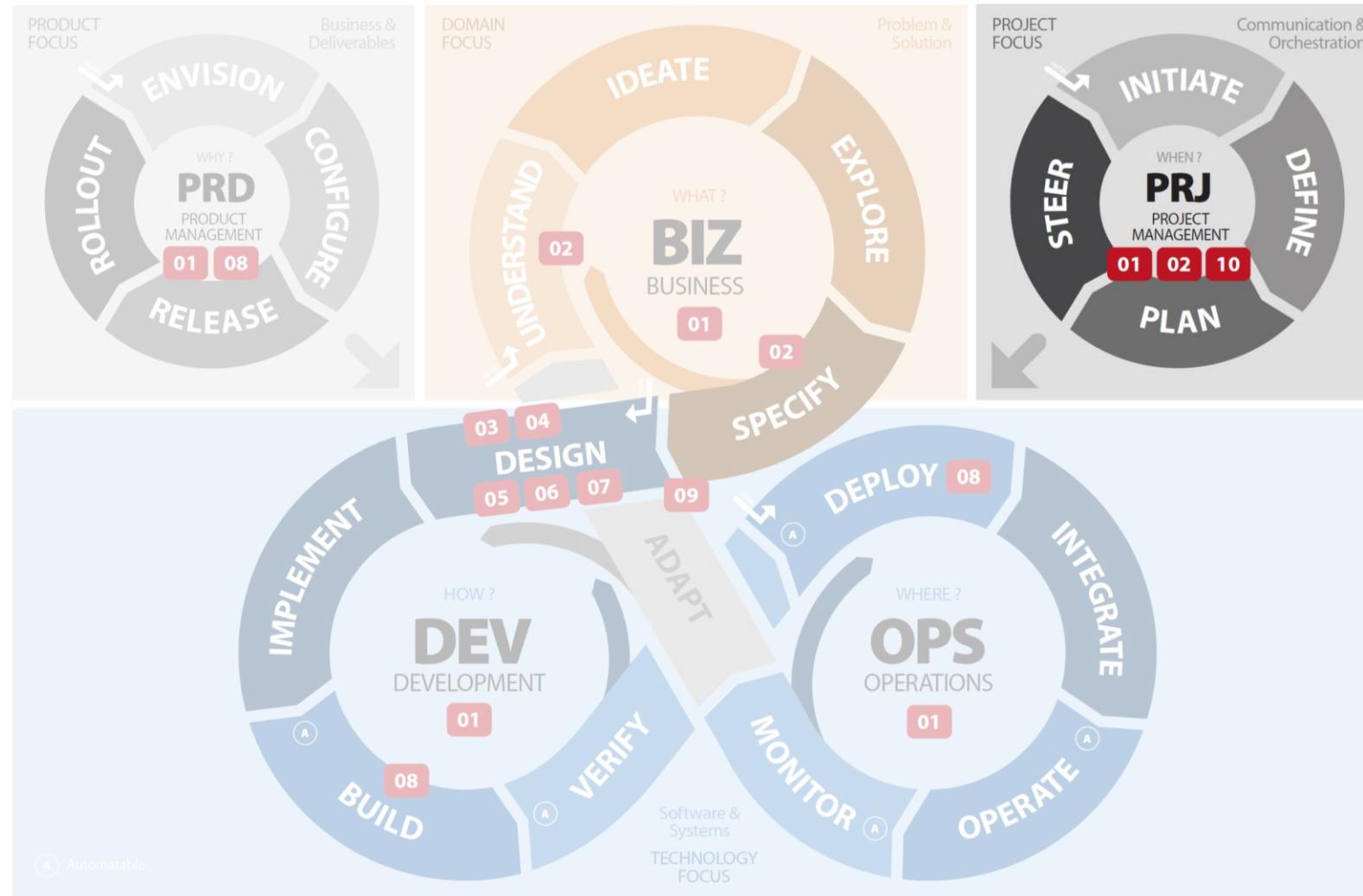
Ihr möchten in **Kontakt** bleiben?  
Sprecht mich gerne auf **XING** oder **LinkedIn** an, Stichwort: Vorlesung SEIP



# Unsere Regeln für ein optimales Online-Meeting

- |   |   |
|---|---|
|  Bitte schalte das Micro stumm, wenn du nicht sprichst |  Mute your mic when not speaking       |
|  Bitte Kamera einschalten 😊                            |  Please turn your camera 😊             |
|  Wenn möglich, Headset benutzen                        |  Please use a headset if possible      |
|  Bei Fragen – bitte Hand heben über den Button         |  Use “raise hand” button for questions |
|  Oder alternativ im Chat kommentieren                  |  Alternatively, comment in chat        |

# Software Engineering Workflow & Vorlesung Software Engineering in der industriellen Praxis (SEIP)





## Large-Scale Project Management Geplanter Ablauf

<b>15:10 - 15:25</b>	Enterprise Agile Frameworks
<b>15:25 - 15:40</b>	Wirtschaftlichkeit von IT Projekten
<b>15:30 - 16:30</b>	Aufwandsschätzung und Kalkulation von Großprojekten

# Large-Scale Project Management

Enterprise Agile Frameworks



# SCRUM at a glance



## Scrum Guide One Pager

- 1 Transparency
- 2 Überprüfung
- 3 Anpassung

**Scrum Team (ST)**

- Zentrales Team (1 PO, 1 SM und n DEV).
- Managt sich selbst (Teamstärke ≤ 10).
- Ist interdisziplinär, alle notwendigen Kompetenzen sind intern vorhanden.
- Verantwortlich für die Erreichung der Sprint-Incremente und somit die Erreichung der jeweiligen Sprint-Ziele; orientiert sich strikt am übergeordneten Produkt-Ziel.

**Product Owner:in (PO)**

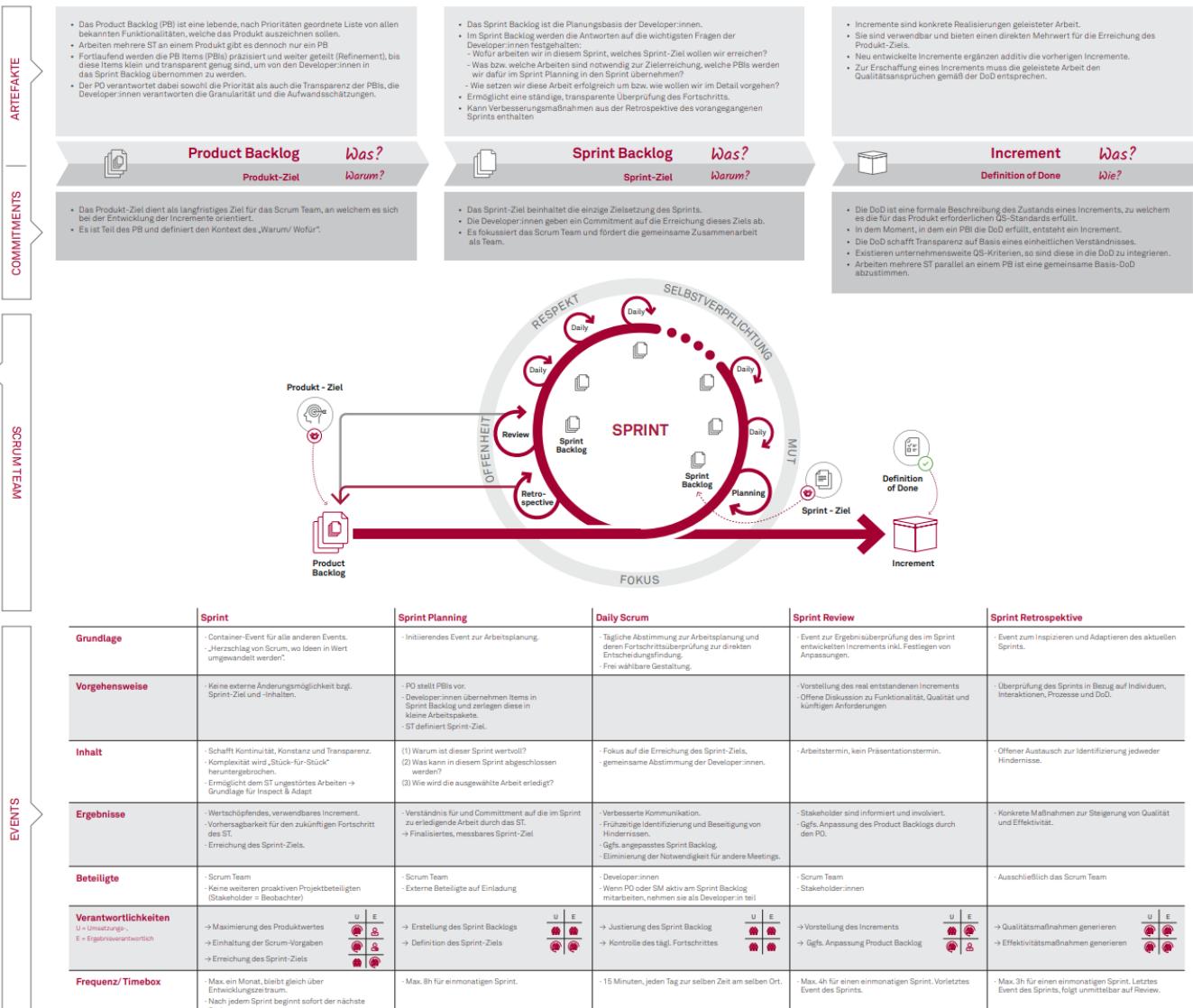
- Ist genau 1 Person, kein Gremium.
- Ist verantwortlich für das Produkt-Ziel, die Wertesetzung des Produktes und das Product Backlog.
- Stellt sicher, dass das Product Backlog sichtbar, transparent, und dessen Inhalte priorisiert und für alle verständlich oder von allen verstanden.
- Die gesamte Organisation muss die Entscheidung des POs respektieren.

**Developer:innen (DEV)**

- Verantwortlich für die Planung und Abarbeitung des Sprint Backlogs.
- Gemeinsam verantwortlich für die Erstellung der Incremente und deren Qualität (DoD).
- Geben ein Commitment zur Erreichung der Sprint-Ziele ab.
- Stimmen sich diesbezüglich täglich ab.

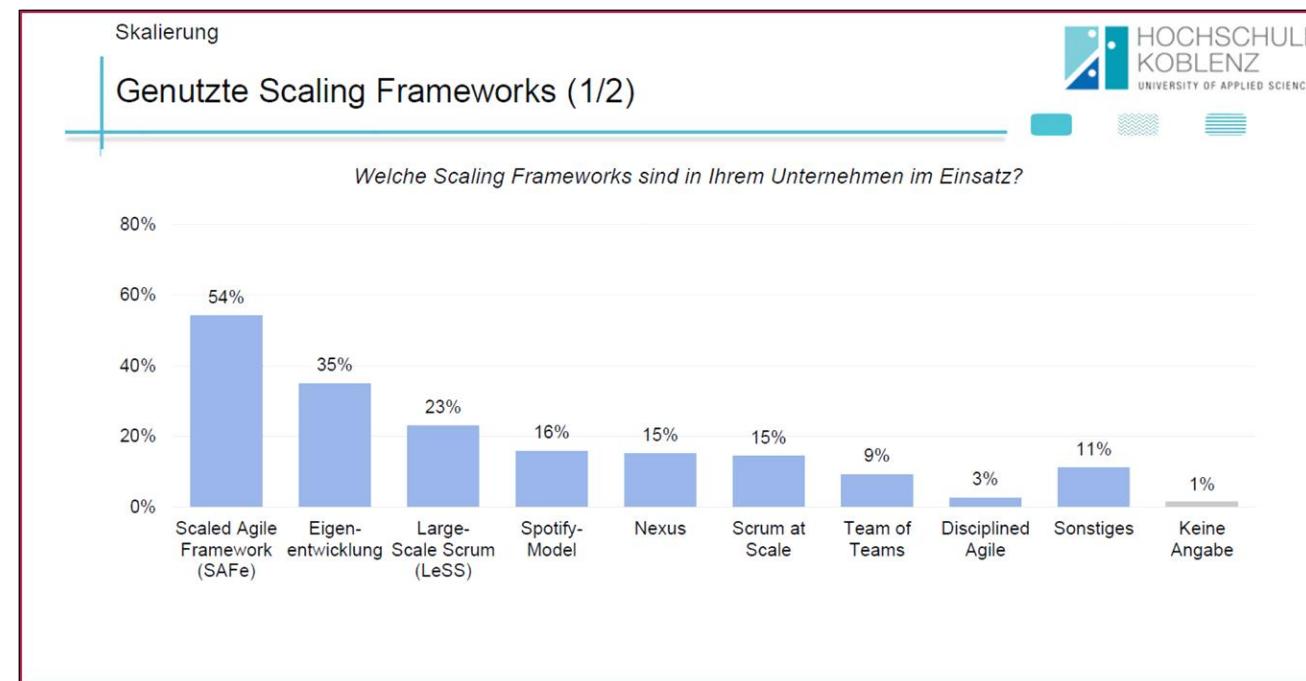
**Scrum Master:in (SM)**

- Sorgt dafür, dass jeder (auch außerhalb des ST) die Theorie, Praktiken und Regeln von Scrum kennt und einhält.
- Verantwortlich für die Arbeitsfähigkeit des ST und dessen Effektivität.
- Ist ein wahrer Leader; moderiert, coacht und berät ST und Organisation.
- Beseitiert Impedimenta und kommuniziert mit allen Stakeholdern, um das Produktwert zu maximieren.
- Lebt und verbreitet die Scrum-Lehre.
- Sorgt dafür, dass die Scrum Events stattfinden und Mehrwert stiften.



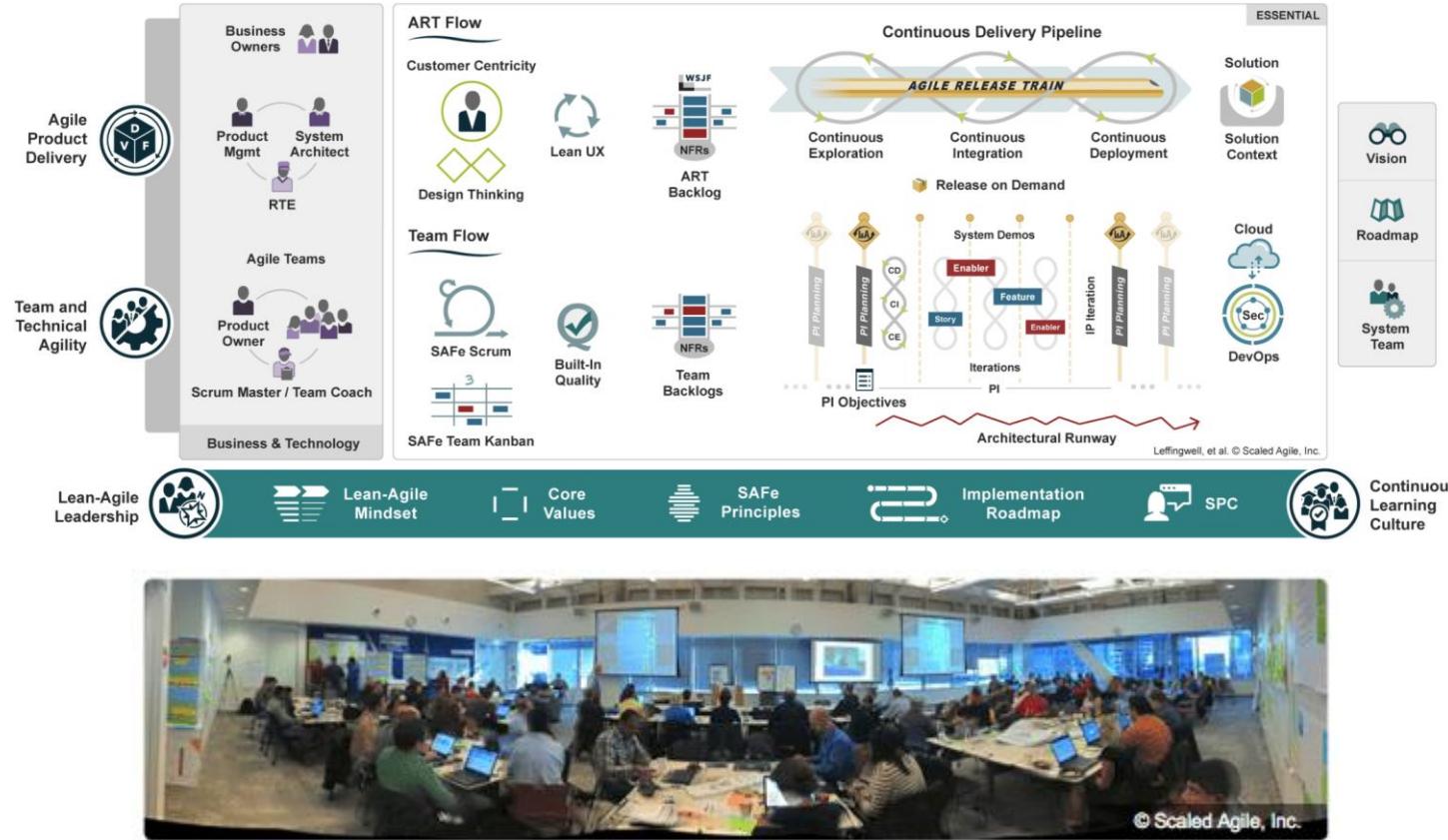
# Overview of different enterprise agile frameworks

- SAFe and LeSS are the predominantly used enterprise agile frameworks
- Both frameworks are widely used
- Both frameworks have been implemented successfully or could be implemented incorrectly



# SAFe at a glance (1/2)

## SAFe 6.0



## Summary

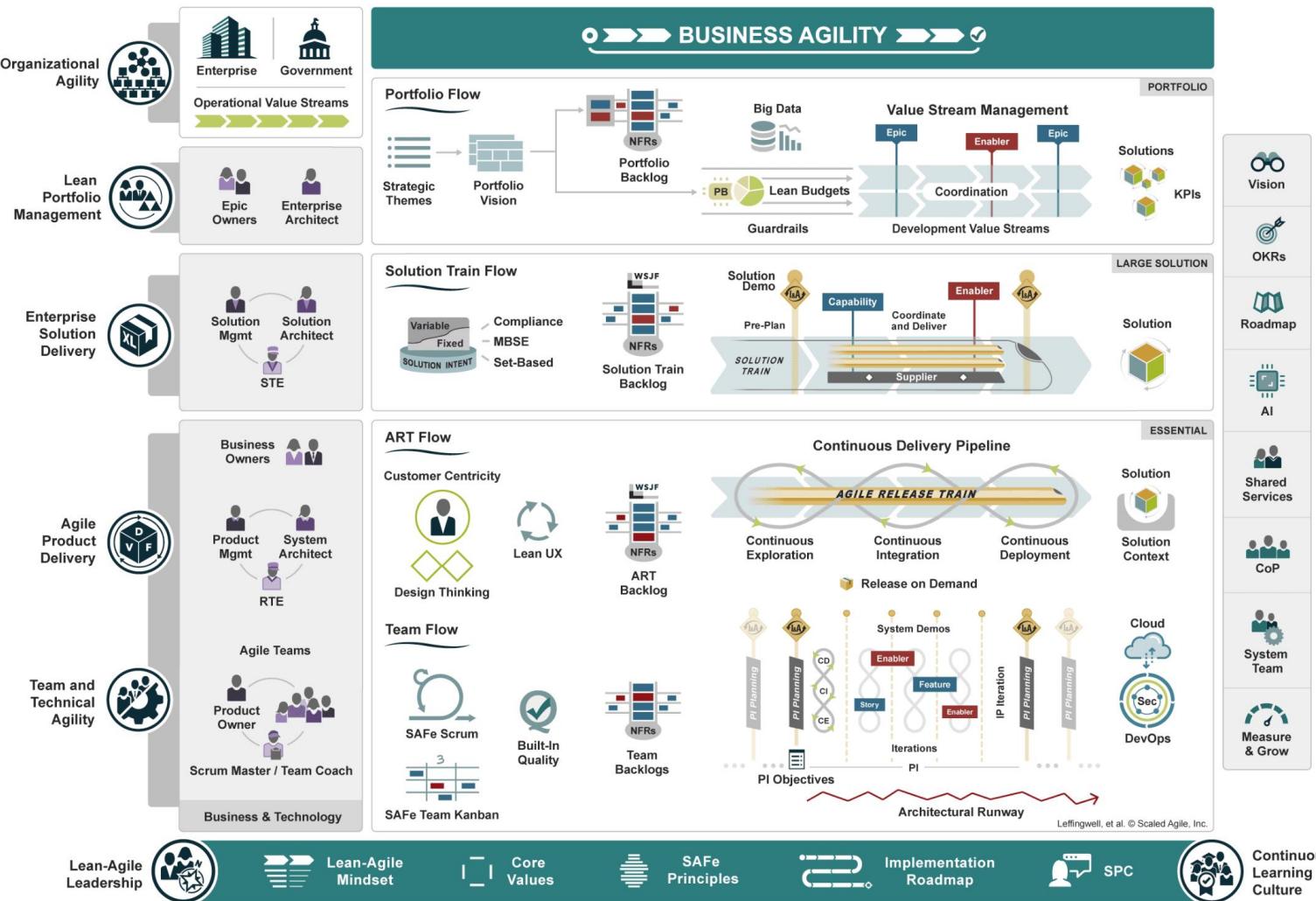
- Several roles (and maybe hierarchies)
- Big room Meetings with multiple teams
  - PI-Planning,
  - Inspect & Adapt (Review, Retro)
- Team Meetings (with one team only)
  - Refinement
  - Iteration Planning Review, Retro
- Same Cadence with 5 Iterations and with Exploration, Integration, Deployment in a Iteration and PI
- Overall Consistent Approach on more levels e.g. Design Thinking or Portfolio Management (see next image)

# SAFe at a glance (2/2)

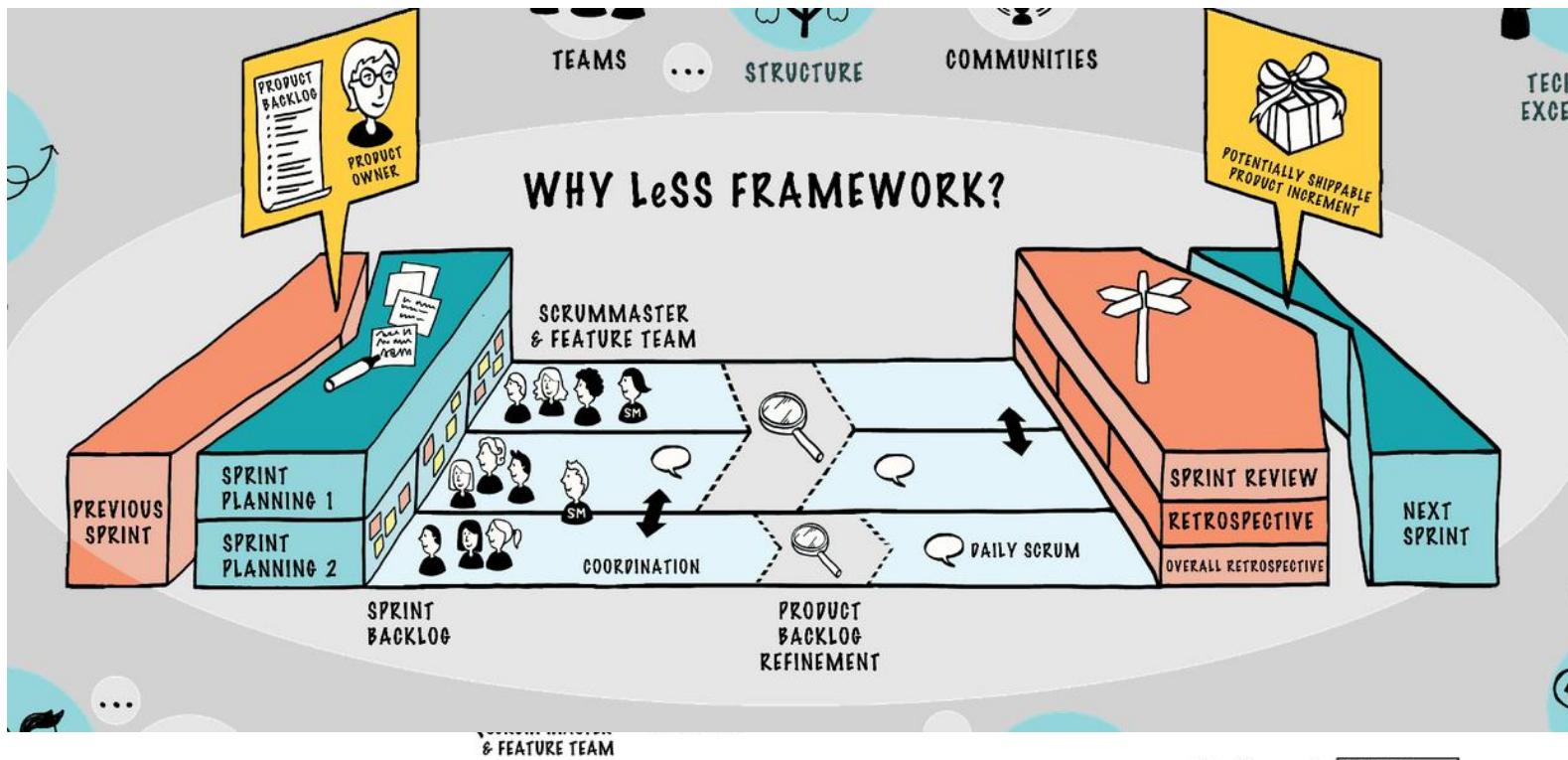
## SAFe 6.0

Select SAFe configuration

OVERVIEW ESSENTIAL LARGE SOLUTION PORTFOLIO FULL



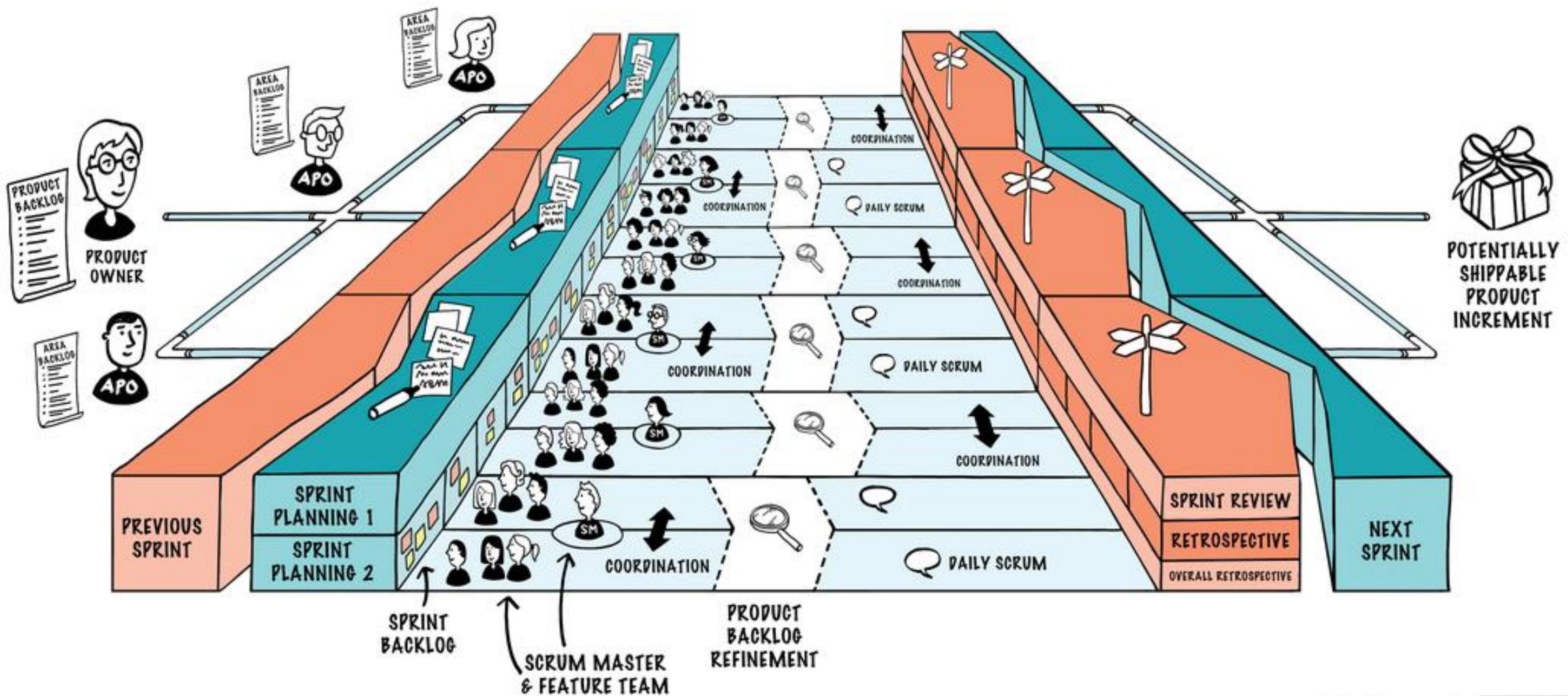
# LeSS & LeSS Huge at a glance (1/2)



## Summary

- LeSS talks not much about hierarchies or roles
- Big room meetings with multiple team
  - Planning 1
  - Refinement
  - Overall Retro
  - Review
- Team meetings (with one team only)
  - Planning
  - Refinement
  - Retro
- Focus on technical Software Development (not Portfolio)

## LeSS &amp; LeSS Huge at a glance (2/2)

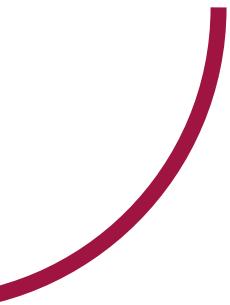


# Comparison chart of selected enterprise agile frameworks

Kosten / Aufwand						
Kategorie	Kriterien	LeSS	Nexus	Spotify	Scrum @ Scale	SAFe
Allgemeine Info	Von wem und wann?	Craig Larman, Bass Vodde, 2005	Ken Schwaber, 2015	Henrik Kniberg, Anders Ivasson, 2012	Jeff Sutherland, 2018	Dean Leffingwell, 2011
	Dokumentation	LeSS.works	scrum.org (Nexus Guide)	labs.spotify.com	scrumalliance.org	scaledagileframework.com
	Basis (Scrum/ Kanban,...)	Scrum	Scrum	offen für alle agilen Methoden	Scrum	offen für alle agilen Methoden
Referenz / Einsatz	Referenzkunden	Telekommunikation, BMW, UPS, >30	keine bekannt	Spotify, ING, Deutsche Telekom, Rewe Digital, >4	SAP, Intel, Comcast, Allianz	Hoch / viele >60
	Teamgröße	LeSS: bis ca. 8 Teams; LeSS Huge: ab ca. 8 Teams	3 bis 9 Teams	Squad < 8, ab 4 Squads	Team SoS: 2-5 ; SoSoS 6-25 Mitarbeiter SoS <45; SoSoS <225	ab 7 Teams
Einsatz	Unterstützt standortübergreifende Arbeiten	anwendbar Co-located	anwendbar	nicht anwendbar (Huddle Rooms)	anwendbar	anwendbar aber nicht empfohlen (Big Room Planning)
Bewertung	Verbindlichkeit/ Strukturierungsgrad	niedrig bis mittel	mittel	mittel bis hoch	hoch	hoch
	Abhängigkeiten zwischen den Teams	LeSS: möglich; LeSS Huge: möglichst keine Abhängigkeiten	möglichst keine Abhängigkeiten (oder minimieren)	möglichst gering	offen => eigene Governance möglich	viele Abhängigkeiten möglich => wird gesteuert
	Artefakte	Inkrement, 1 Produkt Backlog, n Sprint Backlog, m Area Backlogs, 1 DoD	Nexus Sprint Backlog, Product Backlog, Increment	Squad, Backlog, Roadmaps	Scrum Backlog, SoS Impediment Backlog, Release Plan	Program Board, PI Objectives
	Events und Zeremonien	Overall Retro, großes und kleines Refinement, Planning 1 (which), Planning 2 (how), Daily (oftter mal woanders teilnehmen), Communities	Nexus Sprint Planning, Nexus Sprint Retro, Nexus Sprint Review, Nexus Daily Scrum	Scrum Kanban Meetings, Hackdays, Quartalsstudie, Chapter Meeting, Gilde Treffen	Scrum (Daily, ..); Scaled Daily Scrum, Exec. Meta Scrum, PO Team Release Planning	PI Planning, SoS, Syst. Demo; (epic, feature, story) Dependency Map Enabler
Einführung	Kosten / Lizenz / Schulung	agiles Mindset	gering	abhängig von aktuellem kulturellen Umfeld	Lizenz: durchschnittlich; Change: hoch; Training: niedrig	mittel bis hoch
	Was muss „ontop“ getan/ entschieden werden?	Mindset organisatorische Änderungen	Agiles Mindset	ggf. Agile / Spotify Mindset Schulung	Change Mgmt: Transformation, Setup; FW: Fehlende Vorgaben in SoS entwickeln	Implementation Roadmap durchlaufen: v.a. Schulung für alle, LACE Team bevollmächtigen
	Organisationskultur Ist-Bild => Wunschkörper	—	—	Agiles Mindset, willing to fail, failure culture	Startup	eher geeignet für traditionelle Unternehmen, für alle offen
	Risiken und Hindernisse	—	—	kein Blick auf das Product Backlog als Ganzes durch zu viel Autonomie	Change Aufwand hoch, Freiheit => Chaos?	kann fehlinterpretiert werden und über bestehende Organisation übergestülpt werden ohne viel zu ändern
Besonderheiten	Besonderheiten	LeSS Test, Area Product Owner, Travelers	Integration Team	Autonomie Squads, Huddle-rooms, Fehler-Kultur, Gemeinschaft von Hierarchie, ständige Motivation, langfristige Ziele bei Squads	ganze Organisation, methodische Governance im SoS, nicht FW; leichtgewichtig	Selfassessment (Essential SAFe, Lean Enterprise, Dev Ops Health,...), Implementation Map, geeignet für (Weiter-) Entwicklung von Programmen (mehrere Produkte)
	Rollen (extra)	—	Integration Team	Agile Coach	Chief Product Owner	Release Train Engineer, Product Manager, SPC, System Architect, Business Owner, System Team
Strukturierungsgrad / Verbindlichkeit						

# Large-Scale Project Management

Wirtschaftlichkeit von IT Projekten



# Software Engineering in der industriellen Praxis

## Projektmanagement: Wirtschaftlichkeit

1. Grundlagen und Begriffsdefinitionen

---

2. Wirtschaftlichkeit von IT-Projekten

---

3. Literatur

# Software Engineering in der industriellen Praxis

## Projektmanagement: Wirtschaftlichkeit

1. Grundlagen und Begriffsdefinitionen
  2. Wirtschaftlichkeit von IT-Projekten
  3. Literatur
- 
-

# IT-Wirtschaftlichkeit

## IT-Investitionen



- IT-Vorhaben und IT-Projekte stellen oftmals große Investitionen im Unternehmen dar.
- Einem häufig nicht transparenten Nutzen stehen hohe Kosten gegenüber.
- Begrenzte IT-Budgets erfordern eine objektive Priorisierung dieser Investitionen.

## IT-Organisation



- Wirtschaftlichkeit ist Kernfrage jedes Unternehmens.
- Die IT-Organisation wird als Kostentreiber gesehen.
- Früher wurde IT im Gegensatz zu „klassischen“ Unternehmensfunktionen als „Black Box“ für Entscheidungsträger angesehen, seit es vermehrt digitale Geschäftsmodelle ist der Nutzen der IT klarer.

## Zielsetzung



- Messbare, vollständige und nachhaltige Kriterien als Grundlage für unternehmerische Entscheidungen
- Laufende Kontrolle der Zielerreichung



Wirtschaftlichkeit  
durch IT

=   
Nutzen durch IT  
IT-Kosten

# Software Engineering in der industriellen Praxis

## Projektmanagement: Wirtschaftlichkeit

1. Grundlagen und Begriffsdefinitionen

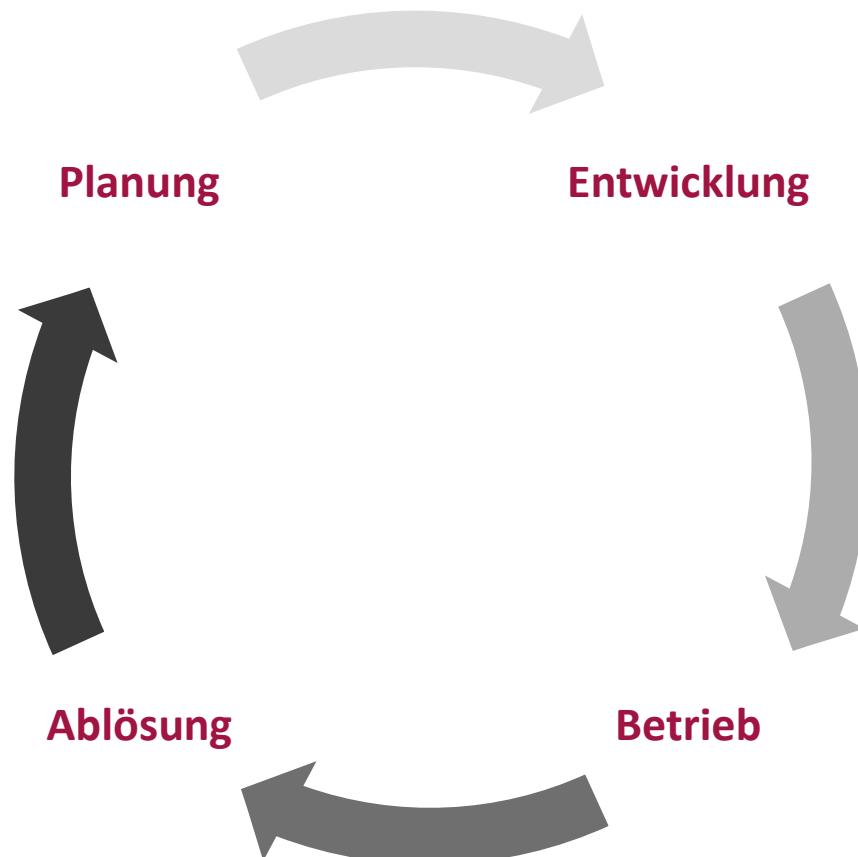
---

2. Wirtschaftlichkeit von IT-Projekten

---

3. Literatur

# Die Phasen eines IT-Vorhabens



# Vernachlässigte oder nicht berücksichtigte Kostentreiber

## Planung

- Machbarkeitsstudie / Business Case
- Grobkonzept / Anforderungsanalyse
- Make or Buy Entscheidung
- Fachkonzept / DV-Konzept, Design

## Häufig vernachlässigte Kostentreiber

## Ablösung

- Entsorgung
- Datensicherung
- Datenmigration

## Planung

## Entwicklung

## Ablösung

## Betrieb

## Entwicklung / Einführung

- Implementierung / Customizing
- Test / Integration
- Hardware, Software
- SLA-Erstellung
- Inbetriebnahme / Roll-out
- Organisationsänderungen

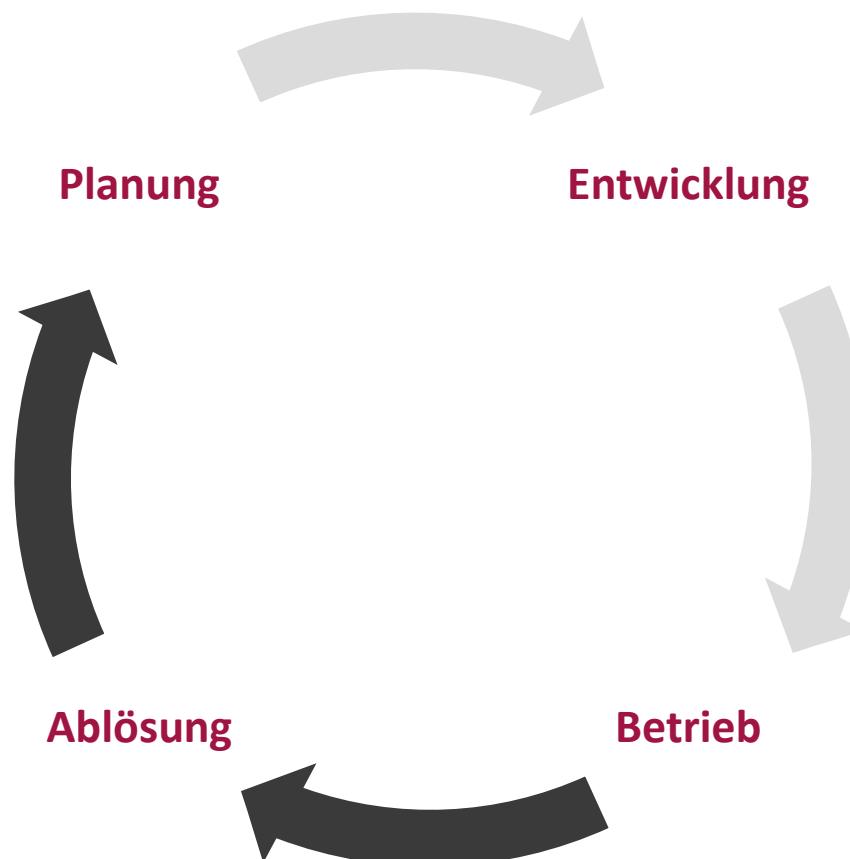
## Betrieb

- Anpassungen
- Wartung und Pflege
- Betriebsmittel
- Weiterentwicklung
- Releasewechsel

# Kostenverteilung für IT-Vorhaben

60-80 % 

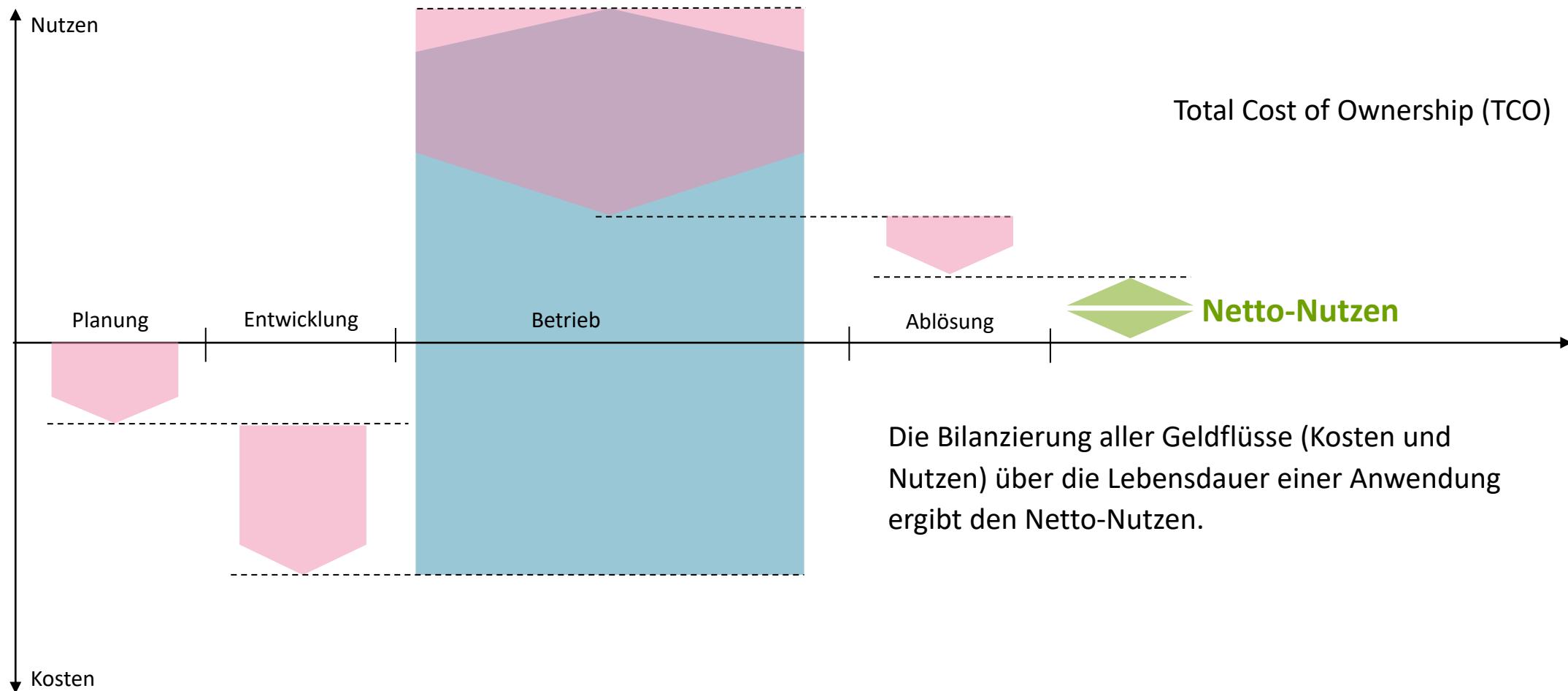
der Kosten fallen dagegen während des **Betriebs** der Anwendung an.



20-40 % 

der Kosten für eine typische IT-Lösung entfallen nur auf die **Erstellung** der Applikation.

# Kosten und Nutzen über den Software-Lebenszyklus



# Software Engineering in der industriellen Praxis

## Projektmanagement: Wirtschaftlichkeit

1. Grundlagen und Begriffsdefinitionen

---

2. Wirtschaftlichkeit von IT-Projekten

---

3. Literatur



## Literatur

- Bernotat J., Stein J., “10 Tipps & Tricks zum Business Case”, GPM-Magazin PMAktuell, 2/2007, S. 43-47
- Stein J, “Mit dem Business Case Wirtschaftlichkeit von Projekten nachweisen – der Business Case sichert den Erfolg von IT-Projekten”, GI/ACM-Regionalgruppe, Karlsruhe, 25.09.2007
- Brugger R., “Der IT Business Case”, Springer, 1. Aufl., 2005

# Large-Scale Project Management

Aufwandsschätzung und Kalkulation von Großprojekten

# Software Engineering in der industriellen Praxis

## Projektmanagement: Aufwandsschätzung

1. Grundlagen und Begriffsdefinitionen

---

2. Bottom-Up Schätzung (Expertenschätzung)

---

3. Top-Down Schätzung (Use Case Points)

---

4. Literatur

# Software Engineering in der industriellen Praxis

## Projektmanagement: Aufwandsschätzung

### 1. Grundlagen und Begriffsdefinitionen

---

### 2. Bottom-Up Schätzung (Expertenschätzung)

---

### 3. Top-Down Schätzung (Use Case Points)

---

### 4. Literatur

# Wenn Projekte „Groß“ werden!

## Was ist der Unterschied zwischen einem Projekt und einem Großprojekt?

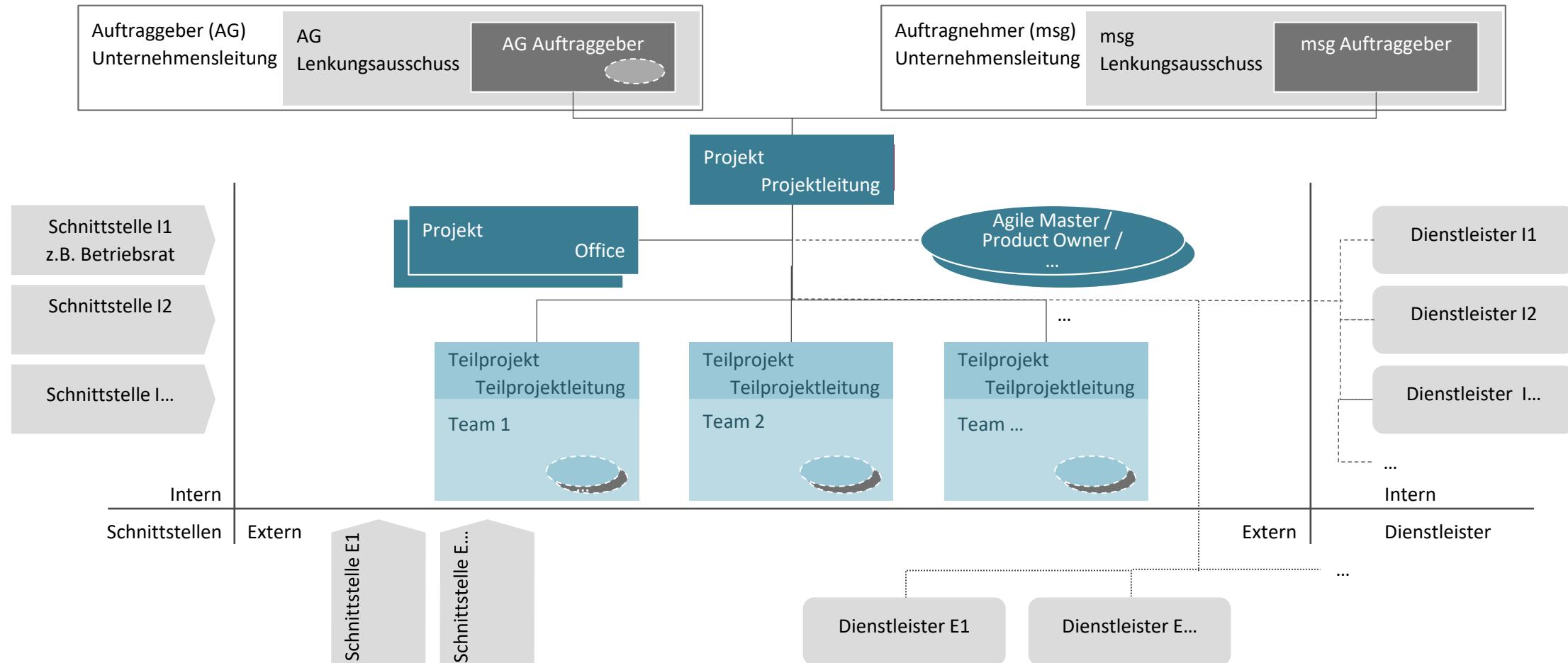
- Projekt und Großprojekte haben die gleiche Definition
  - Ein Projekt ist ein „Vorhaben, das im Wesentlichen durch Einmaligkeit der Bedingungen in ihrer Gesamtheit gekennzeichnet ist, wie z. B. Zielvorgabe, zeitliche, finanzielle, personelle oder andere Begrenzungen, projektspezifische Organisation.<sup>1</sup>
- Projekte und Großprojekte unterscheiden sich „nur“ in der Ausprägung von einigen Kriterien, in der Regel der hohen Quantität der Ziele und/oder Lieferergebnisse (Spezifikation) und der Umfang in Zeit und Kosten.



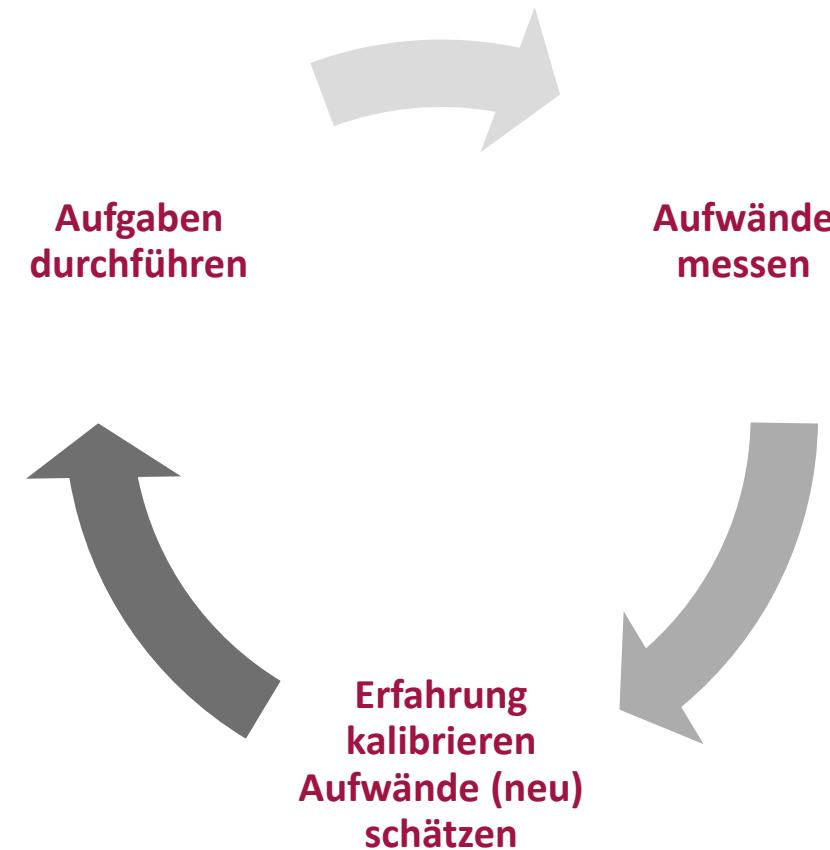
Projekt (IPMA Checkliste)	Großprojekt
Einmalig und neuartig	„Einmaliger“, es gibt weniger vergleichbare Projekte
Zeitlich begrenzt	Lange Laufzeit (> 2 Jahre)
Interdisziplinäre Zusammenarbeit	Und sehr viele Disziplinen
Festgelegte Ergebnisverantwortung	Projektleitung, ggf. Co-Projektleitung
Komplexität	Sehr hohe Komplexität
Zielvorgabe	Zielvorgaben umfangreich, instabil
Begrenzte Ressourcen	Hohe Quantität an Ressourcen

1. Definitionen: © GPM (E-Book) | PM3 | DIN 69901-5 (DIN, 2009c)

Interne und externe Schnittstellen und Dienstleister müssen genauso betrachtet werden, wie doppelte Auftraggeber und weitere (agile) Rollen!

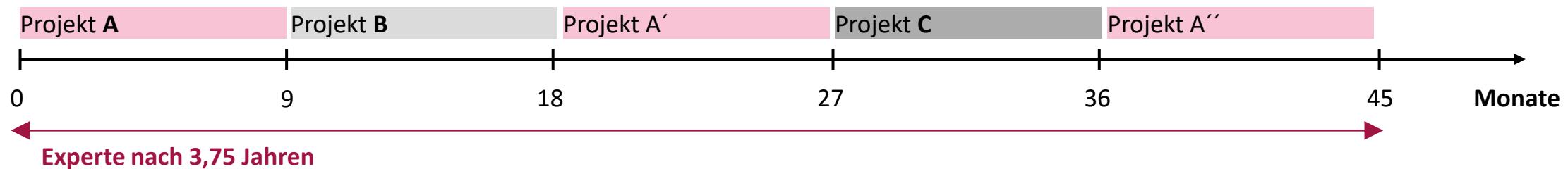


# Aufwandsschätzungen beruhen immer auf praktischer Erfahrung und Intuition

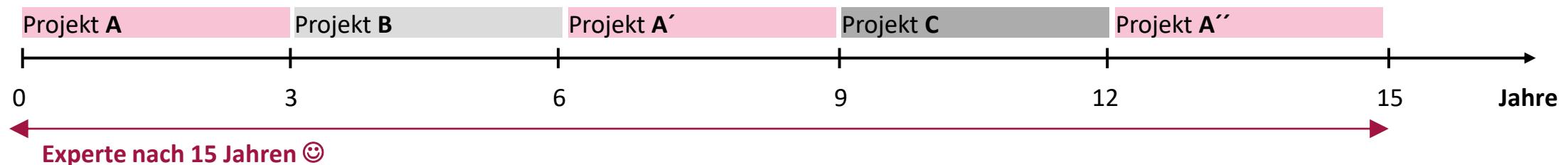


# Die Grenzen der Intuition sind in Großprojekten erreicht

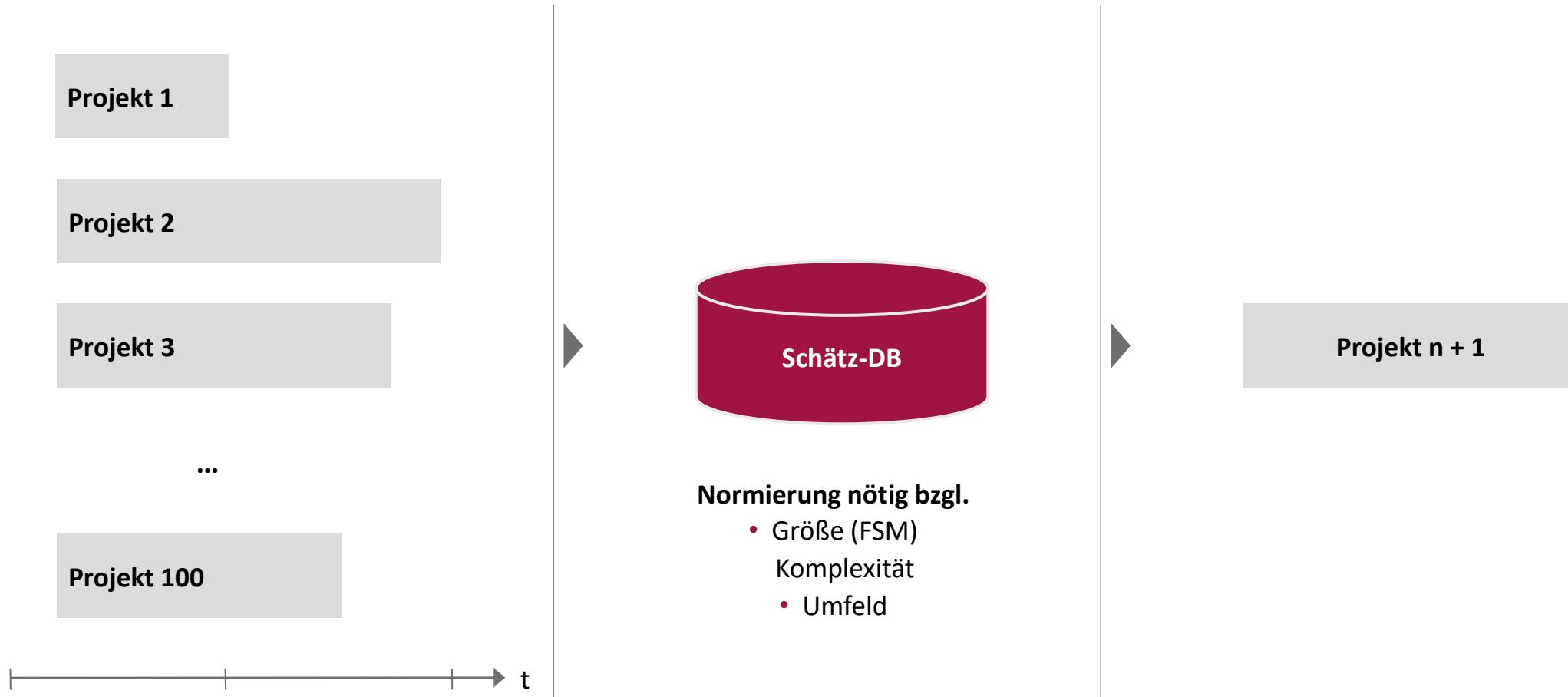
- Expertenschätzungen beruhen auf Erfahrungen von Experten:  
Jedes Element der Stückliste wird individuell vom Experten taxiert
- Experte: Mindestens 3 x eine vergleichbare Aufgabe/Projekt selber durchgeführt
- Annahme: **ein typisches (kleines) Projekt** dauert 9 Monate:



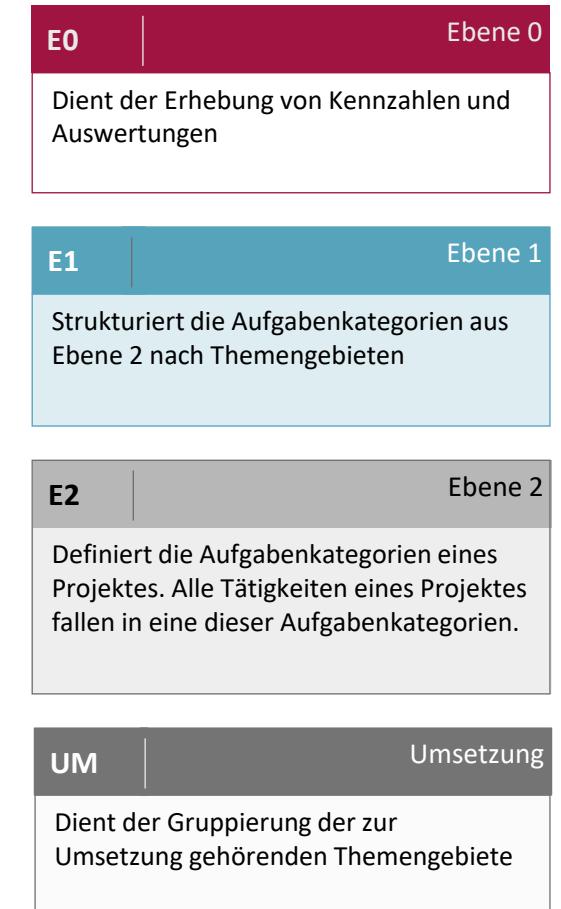
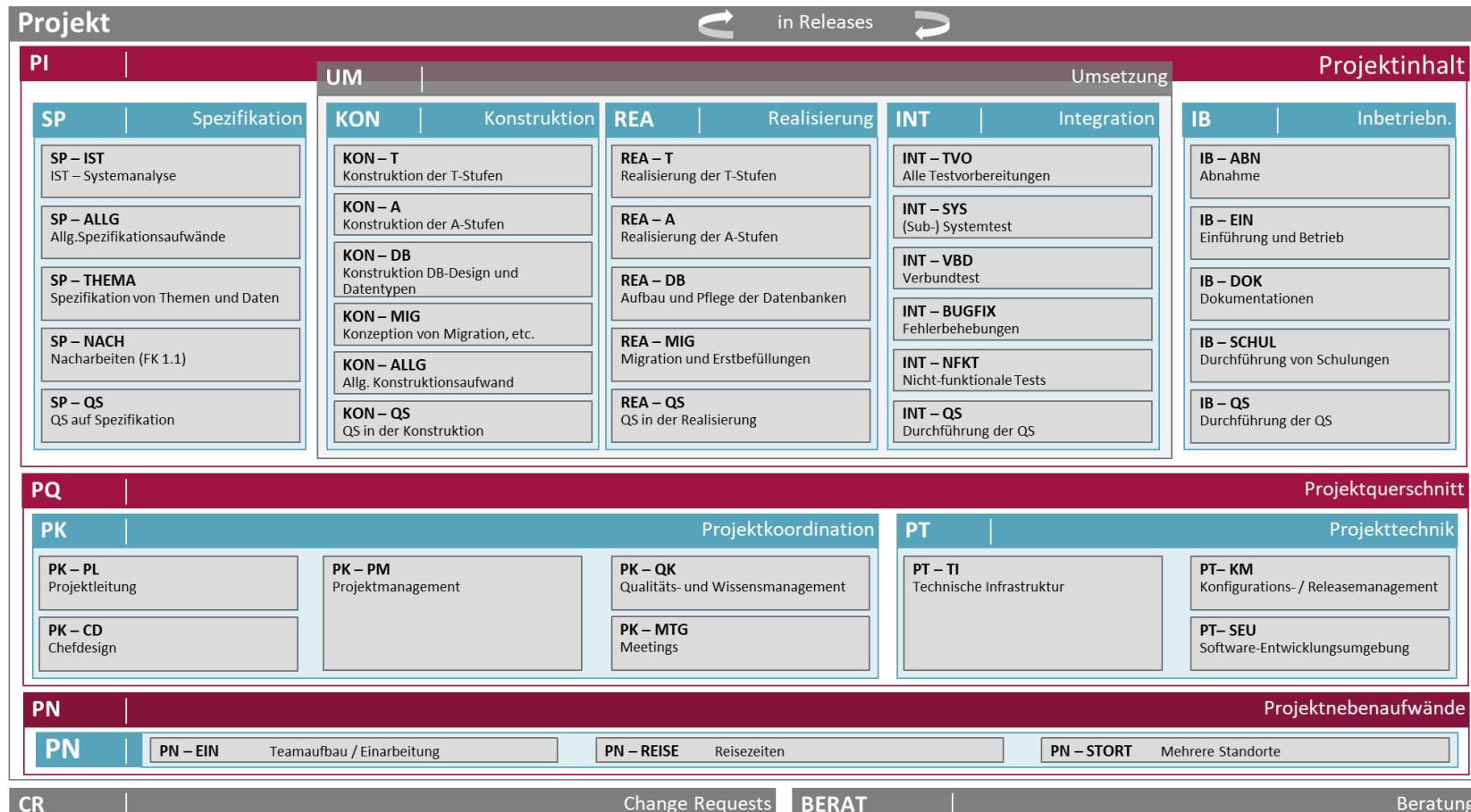
- Annahme: ein **Großprojekt** bzw. **Programm** dauert 3 Jahre:



# Schätzdatenbanken mit FSM (Funktional Size Measurement) überwinden die Grenzen der Intuition bei Großprojekten



# Das Aufwandsmodell<sup>1</sup> strukturiert Projekttätigkeiten nach Aufgabenkategorien → Alle Tätigkeiten in einem Projekt lassen sich eindeutig zuordnen!

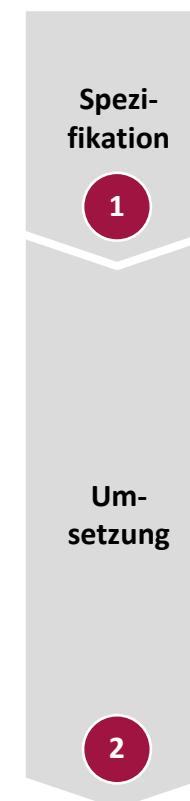
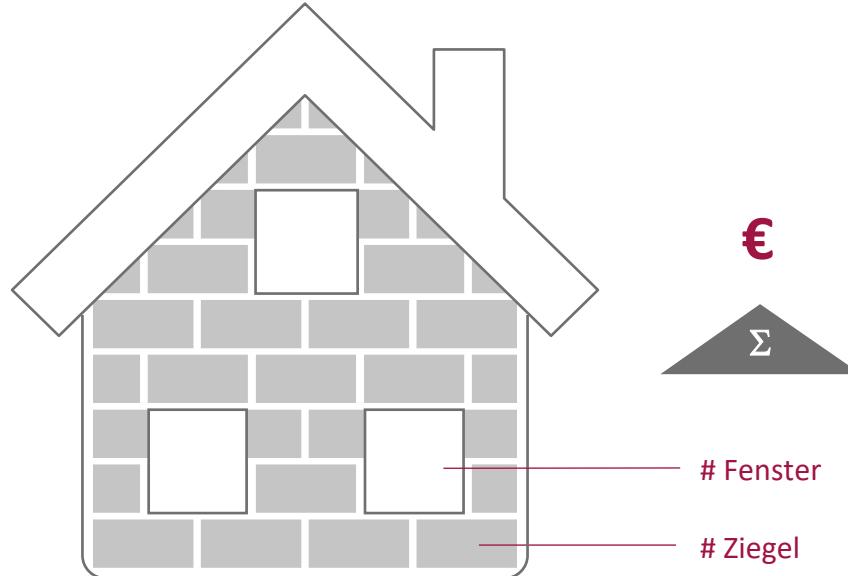


1. Quelle: Dissertation „Use Case Points 3.0“ von Dr. Stephan Frohnhoff, Universität Paderborn, 2009

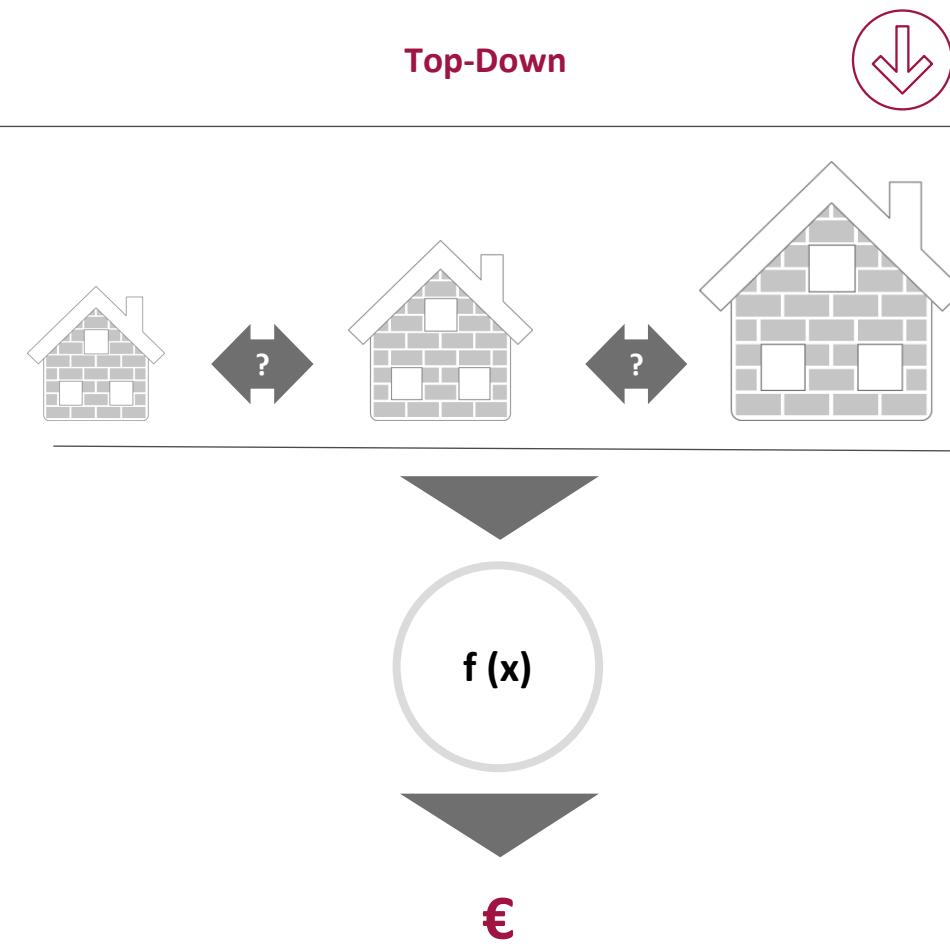
# Wir unterscheiden Bottom-Up und Top-Down Schätzverfahren



Bottom-Up



Top-Down



# Bottom up ist die bevorzugte Schätzstrategie

## Schätzstrategien



Top-Down

- Gesamthaftes Schätzen des Projektaufwandes mit Hilfe von **mathematischen Algorithmen** auf Basis der funktionalen Anforderungen. Verwendet msg in der Regel nur zur Plausibilisierung.



Bottom-Up

- Aufwände jedes Aufwandspostens werden getrennt ermittelt und zum Gesamtprojektaufwand **summiert**.
- Im typischen msg Projekt gehen wir Bottom-Up vor.

# Schätzverfahren im Überblick

Algorithmische Methoden	Vergleichsmethoden	Kennzahlenmethoden	Expertenschätzungen
<b>COCOMO</b> <b>Function Points</b> <b>Use Case Points</b>	<b>Analogiemethode</b> <ul style="list-style-type: none"> <li>Stellt Bezug zu durchgeführten Entwicklungsprojekten her</li> <li>Keine messbaren Produktgrößen wie LoC nötig</li> <li>Nachkalkulationen alter Projekte nötig</li> </ul>	<b>Multiplikatormethoden</b> <b>Prozentsatzmeth.</b> <ul style="list-style-type: none"> <li>Ähnlich Analogiemethode, allerdings braucht man Messdaten abgeschlossener Projekte</li> </ul>	<b>Einzelschätzung</b> <b>Delphi-Methode</b> <b>Schätklausur / PERT-Methode</b> <ul style="list-style-type: none"> <li>Greifen wenn möglich auf Analogiemethode zurück</li> <li>Erstmalige Schätzung neuer Anforderungen durch Expertenwissen</li> </ul>
<b>Top-Down</b>			<b>Bottom-Up</b>

# Software Engineering in der industriellen Praxis

## Projektmanagement: Aufwandsschätzung

1. Grundlagen und Begriffsdefinitionen

---

2. Bottom-Up Schätzung (Expertenschätzung)

---

3. Top-Down Schätzung (Use Case Points)

---

4. Literatur

## Experten-Schätzungen stellen ein weit verbreitetes Verfahren für alle Arten von Entwicklungsprojekten dar

- Systematische Bottom-Up Schätzung von Experten, basierend auf ihrem Erfahrungsschatz
- Schätzposten werden als Aufwandsposten projektspezifisch abgeleitet
- Für „inhomogene“ oder stark kundenspezifische Projekte häufig der einzige gangbare Weg
- Verschiedene Varianten der Experten-Schätzung unterscheiden Systematik und Umfang der Einbindung von Experten:
  - Einzelschätzung: Ein einziger Experte legt die Schätzwerte für einen bestimmten Aufwandsposten fest
  - Delphi-Methode: Mehrere Experten führen ihre Schätzung anonym und getrennt voneinander durch
  - Schätzklausur: Mehrere Experten schätzen im Rahmen eines gemeinsamen Schätzworkshops



# Experten-Schätzung – Vertiefende Informationen: Gegenüberstellung der Varianten



## Einzelgeschätzung

- Ein einziger Experte legt die Schätzwerte für einen bestimmten Aufwandsposten fest.
- Genauigkeit hängt wesentlich von der Erfahrung dieses Experten ab.
- Hohe Verantwortung für eine Person
- Einseitige Beurteilung von Aufwänden und Unsicherheiten

Pragmatisch aber leicht ungenau

## Delphi-Methode

- Mehrere Experten führen ihre Schätzung anonym und ohne Abstimmung untereinander durch.
- Zusammenführung der Schätzung durch den PL ggf. in Iterationen bei (großen) Abweichungen.
- Korrektur-Möglichkeiten der Schätzzahl ohne Gesichtsverlust
- Keine Gruppenzwänge

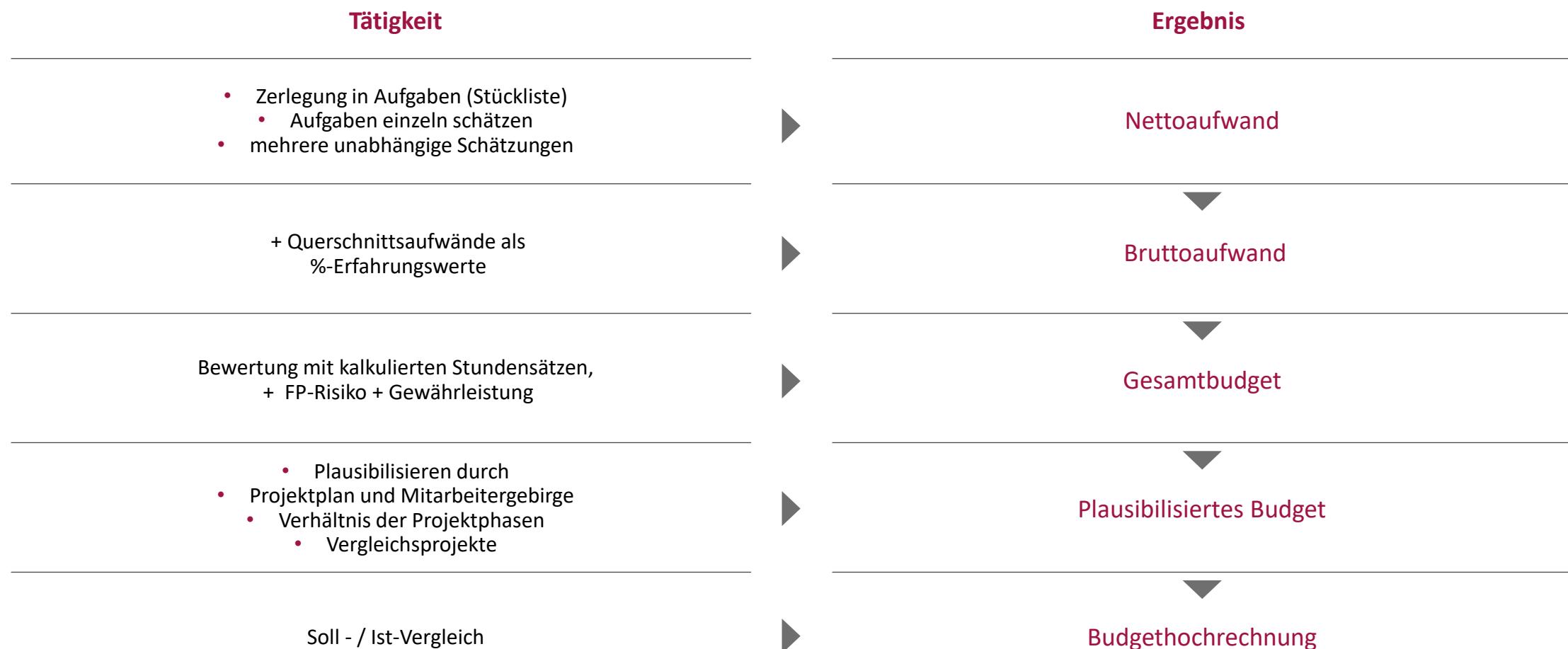
Verlässlich aber sehr zeitaufwändig

## Schätzklausur / Planning Poker

- Mehrere Experten schätzen „online“ im Rahmen eines gemeinsamen Workshops.
- Sofortige Zusammenführung von Aufwänden und Plausibilisierung
- Inhaltliche Diskussionen bei großen Abweichungen
- Gruppe einigt sich auf Aufwand pro Schätzposten
- Risiken werden bewusst
- Gleichmäßiger Informationsstand im Team

Besser als Mittelwerte aber ebenfalls zeitaufwändig

# Die Aufwandsschätzung besteht aus mehreren Schritten



# Alles, was Aufwand macht, ...

## Aufwandsposten (Schätzposten)

- Alle aufwandstragenden Tätigkeiten im Projekt
- Die Liste aller Aufwandsposten gibt die Stückliste
- Nicht jeder Aufwandsposten muss 1:1 ein Arbeitsergebnis sein
- Aufwandsposten müssen nicht mit den späteren Planungseinheiten übereinstimmen

## Arbeitsergebnis Deliverable Ergebnis

- z. B. Fachkonzept, Dialog, Systemdokumentation

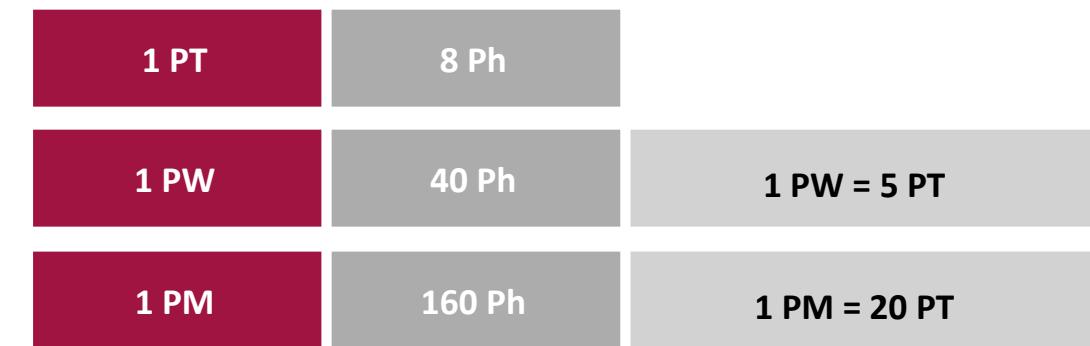
## Sonstige Tätigkeiten

- z. B. Review durchführen, Projektleitung, Meeting, Kickoff-Veranstaltung

# Wir schätzen Aufwände in Personentagen (PT) zu 8 h

- Ein Aufwand von 1 Personentag (PT) muss in 8 Stunden (!) erbracht werden können – nicht in einem 10-Stunden-Tag (oder 24h-Tag ☺).
- Wir schätzen Rüstzeiten nicht extra, d.h. in jeder Aufwandszahl sind also auch Zeiten für Kaffeetrinken, kleinere Pausen, Mails lesen, Schreibtisch aufräumen etc. enthalten

## Planungs- und Schätzungssicht



# Für jeden Schätzposten wird Aufwand und Schätzunsicherheit ermittelt

$$\text{Gesamtaufwand} := \text{Schätzung} + \text{Aufwandsrisiko}$$

---

## Schätzung

Vorgehen zur Ermittlung des Aufwands und des Schätzrisikos unter Zuhilfenahme eines Schätzverfahrens.

**Grundlage sind in jedem Fall feststehende Anforderungen oder mindestens als Prämissen dokumentierte Annahmen über Projektinhalt und Rahmenbedingungen.**

Das Ergebnis der Schätzung ist der vollständige Aufwand des Projekts in PT (im Gegensatz zur Kalkulation: €).

---

## X% der Schätzunsicherheit.

## Aufwandsrisiko

Die Schätzunsicherheiten wird nicht bei jedem Aufwandsposten zuschlagen, die Festlegung hängt von der Einschätzung des Angebotsverantwortlichen ab.

# Was bedeutet „Festpreis“?

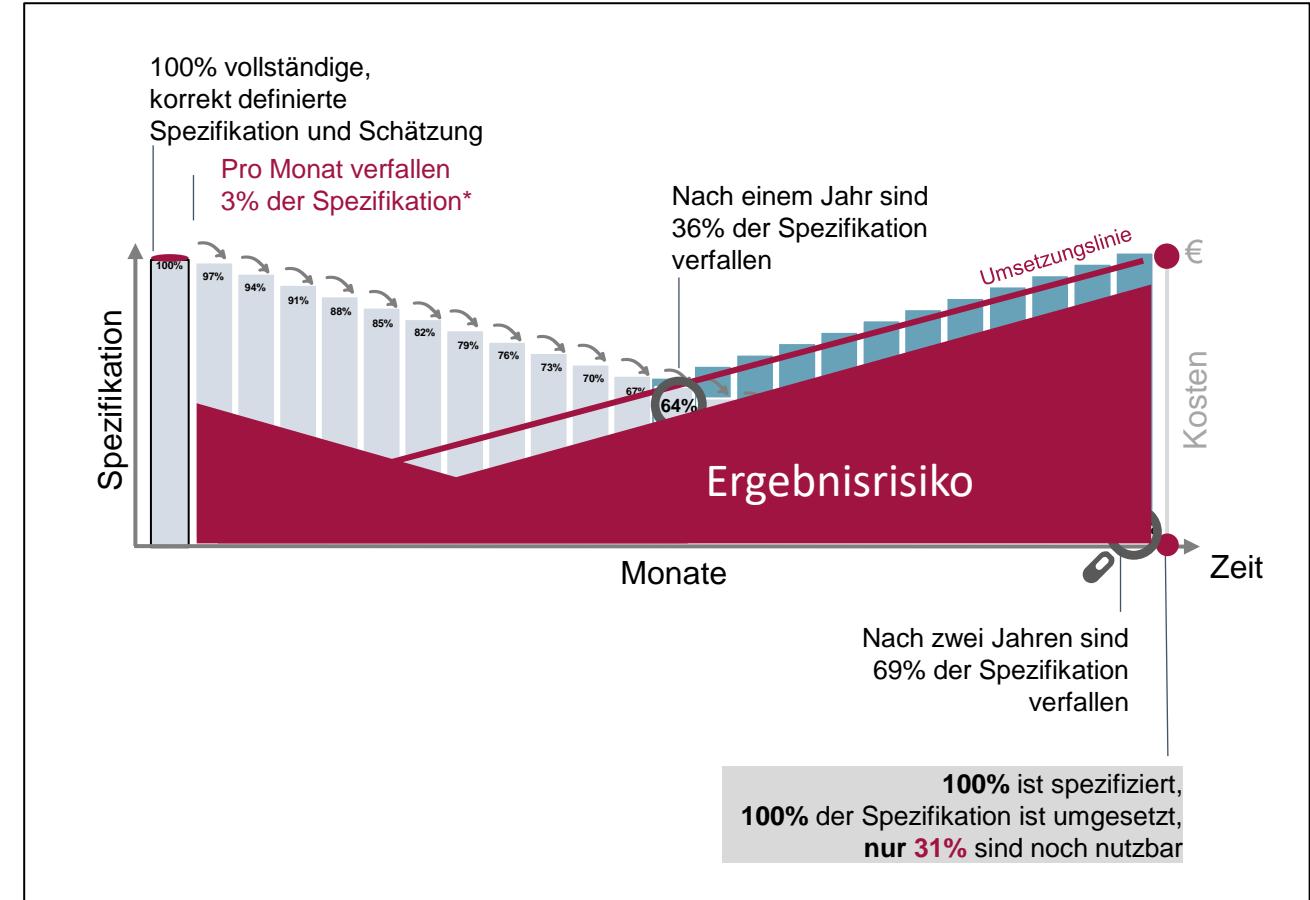
## Definition Festpreis:

Lieferung ...

- ✓ eines definierten **Ergebnisses**
- ✓ zu einem bestimmten **Zeitpunkt** &
- ✓ für einen festgelegten **Preis**

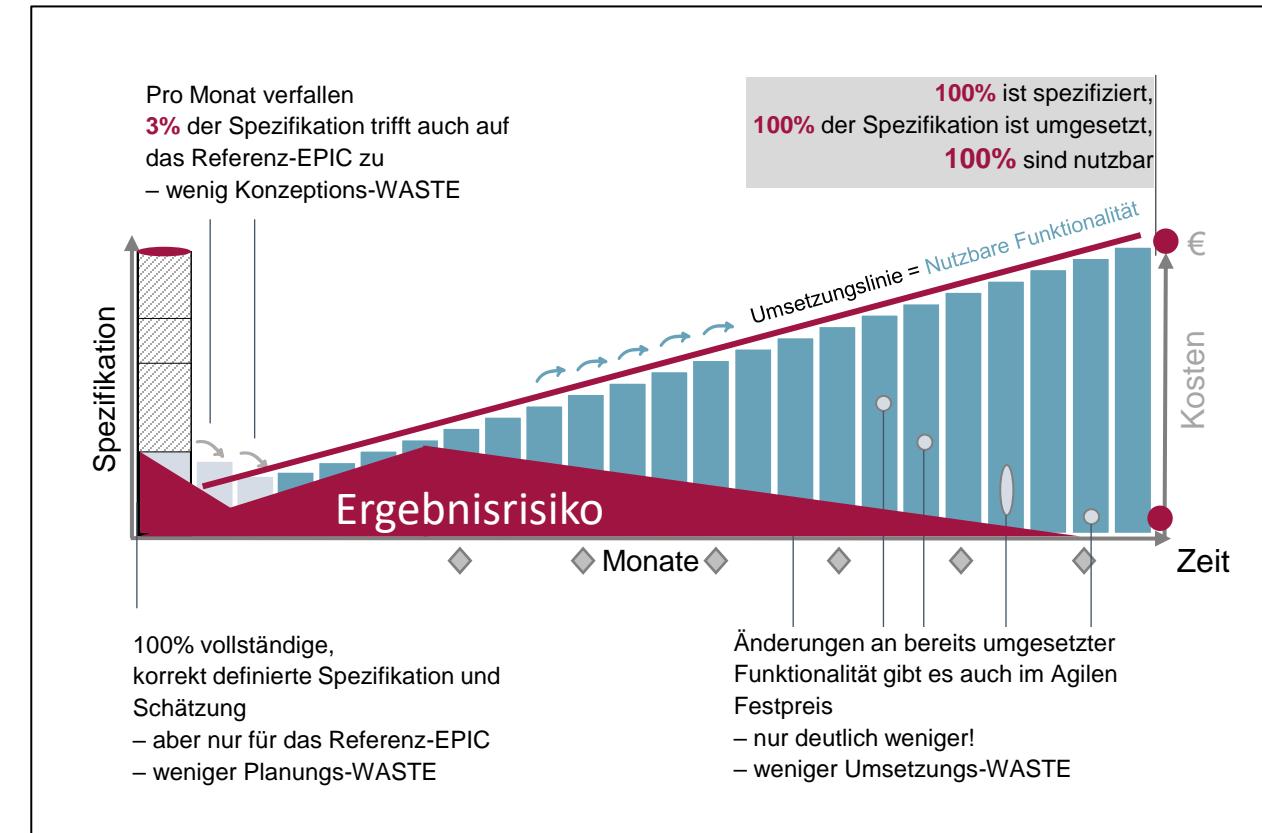
## Probleme des Festpreises (Vorgehensmodell nach Wasserfall)

- Dynamische Veränderungen erwirken einen „Spezifikationsverfall“ des definierten Ereignisses  
z.B. Änderungen ...
  - des Marktes
  - der Technik
  - des Produktes
  - des Umfelds
  - der Stakeholder / Ansprechpartner
  - der Unternehmensprioritäten
  - ...
- Kunde & Dienstleister arbeiten nicht mittelbar zusammen
- Bürokratischer Aufwand für Vertragsänderungen aufgrund der oben genannten Änderungen



# Wie funktioniert ein „Agiles Festpreisprojekt“?

- 1.**  
Vison/Ziel
  - Gemeinsame Definition des Gesamtziels
  - Identifizierung eines EPICs als Referenz-EPIC (hohe Priorität, frühe Entwicklung)
  
- 2.**  
Referenz-EPIC
  - Vollständige Runterbrechung in User Stories & Schätzung (z.B. mit Planning Poker & Story Points) auf Basis einer / mehrerer Referenz-User-Stories  
-> Kunde & Dienstleister zusammen
  - Story Points werden in Kosten umgerechnet
  
- 3.**  
Weitere EPICs
  - Gemeinsam durchgeführte Schätzung auf Grundlage des Referenz-EPICs unter Verwendung von Magic Estimation
  
- 4.**  
Weitere EPICs
  - Spezifikation & Schätzung wird kurz vor Umsetzung der Funktionalität erstellt:  
„Änderungen“, „Erfahrungen“ & „Feedback“ fließen zeitnah mit ein, dadurch ist
    - der Spezifikationsverfall ist gering &
    - weniger Konzeptions-WASTE entsteht



# Praxisbeispiel: Strukturierung einer Stückliste

BEISPIEL  
AUS  
DER PRAXIS

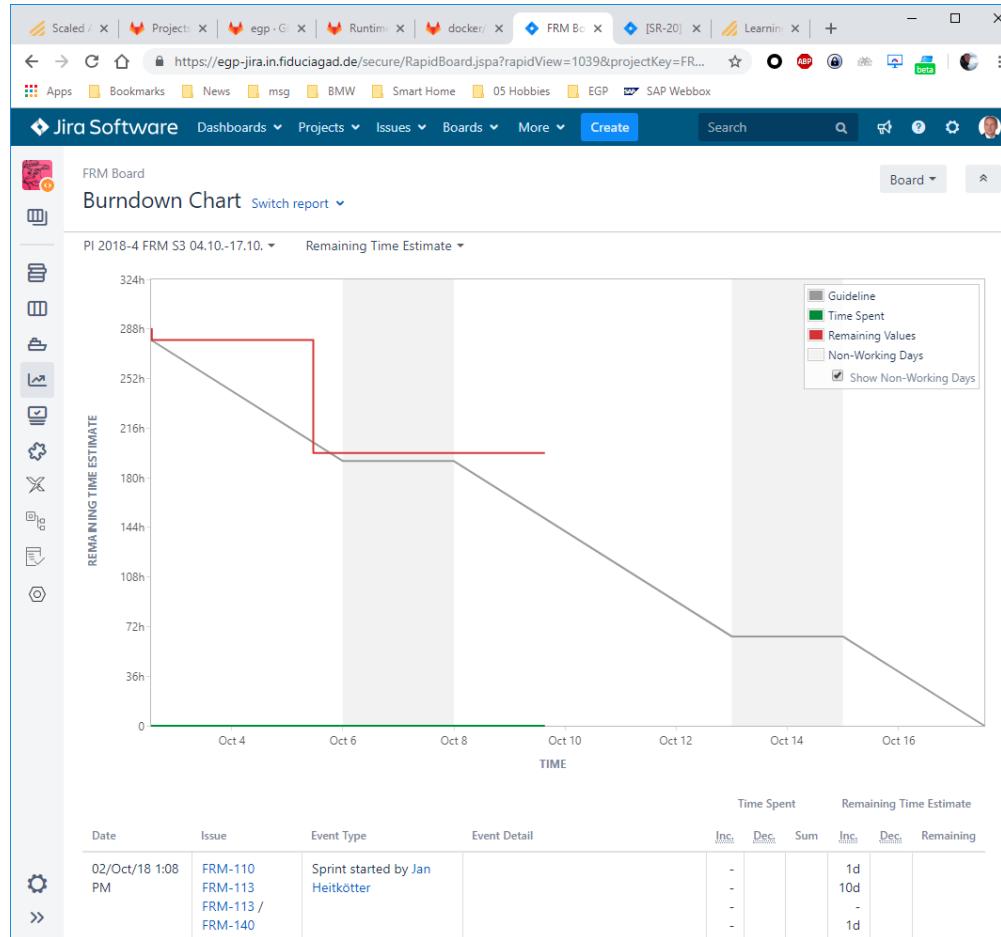
Strukturierung nach den Requirements des Kunden

Abstrakte, projektspezifische Strukturierung des Projektinhalts nach Komponenten & Themen

Projektspezifische Detaillierung in einzelne Aufwandsposten (Schätzposten)

Aufwandsschätzung in Bearbeitertagen (PT)

# Praxisbeispiel: Projektcontrolling der einzelnen Backlog-Positionen / Program Increments / Sprints



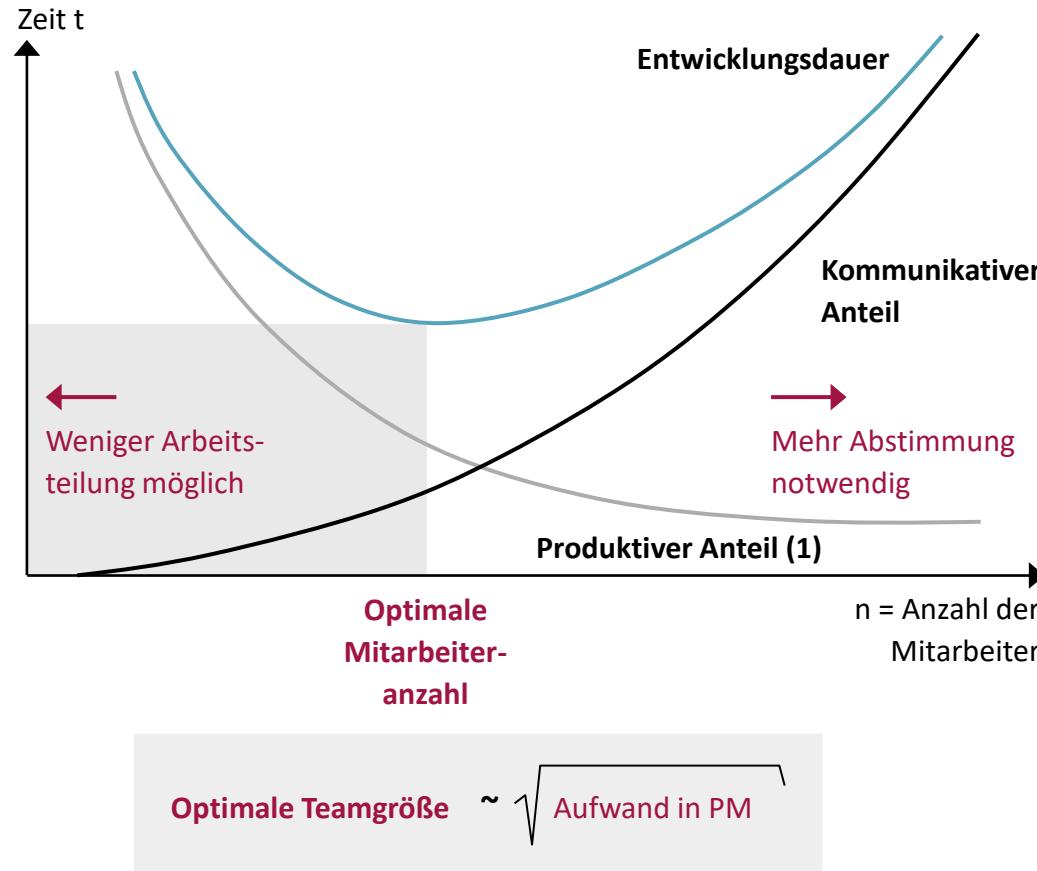
- Im Rahmen der Umstellung auf ein agiles Vorgehen wurde die zentrale Steuerung maßgeblich gestärkt („ART-Team“).
- Fortschritts- und Budgetcontrolling findet auf Sprint- und PI-Ebene sowie global statt. Eine Reihe von automatisierten Auswertungen liegen dazu schon vor oder sind in Entstehung. Gemessene KPIs umfassen u.a.:
  - Team-Performance
  - Einhaltung des Sprint-Sscopes und des geplanten Aufwands
  - Budgetverbrauch, Burn-Down-Charts, regelmäßige Restaufwandschätzungen

# Die Kennzahlen des Aufwandsmodells dienen der Plausibilisierung einer Schätzung

Kennzahlenplausibilisierung					
Kennzahlen	Schätzung	Erfahrungswerte aus Aufw.-Modell (~1σ-Bereich) von	bis	Median	Kommentar
SP / PI	0%	8%	28%	19%	Detailspezifikation fertig bzw Rest nach Aufwand bis v1.0
KON / UM	13%	9%	25%	17%	
REA / UM	53%	35%	65%	52%	
INT / UM	34%	17%	40%	32%	
INT-BUGFIX / UM	8%	5%	19%	13%	
PK / PI	45%	15%	40%	28%	
PK-PL / PI	16%	6%	18%	12%	PL & Entwicklungsleitung, daher mehr als normal
PK-PM / PI	3%	2%	6%	4%	
PK-CD / PI	12%	4%	12%	8%	
PT / PI	17%	3%	10%	6%	DevOps sind mit eingeplant (zusätzliche Resource)
PN-EIN / PI	1%	2%	7%	4%	
QS / PI	0%	3%	8%	5%	wurde nicht separat geschätzt, sollte aber abgedeckt sein

- Unter einer Kennzahl verstehen wir jeden Quotienten aus zwei Aufgabenkategorien des Aufwandsmodells. Damit steht hinter jeder Kennzahl eine klare inhaltliche Bedeutung, was die Voraussetzung für einen unternehmensweiten Gebrauch von Kennzahlen darstellt.
- Mit der Kennzahlenplausi kann plausiliert werden,
  - in der Schätzung alle Aufwandskategorien umfassend abgedeckt wurden.
  - die Verteilung des Aufwandes entsprechend der Erfahrung aus bisherigen Projekten sinnvoll erscheint.
- Die Erfahrungswerte zur prozentualen Verteilung dienen dabei nur als Richtwerte, da jedes Projekt individuell ist. Größere Abweichungen sollten aber zumindest hinterfragt und begründet werden.

# Als erster Anhaltspunkt für Teamgröße und Projektlaufzeit dient Brooks Faustformel



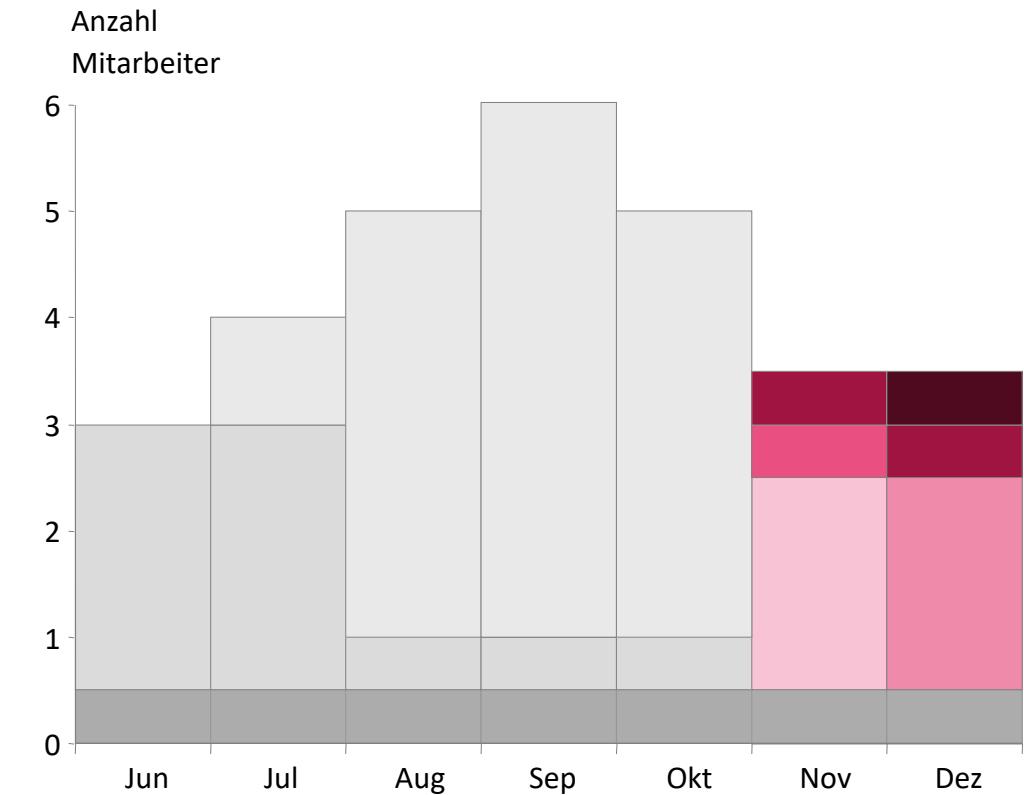
„Der Mann-Monat als Maßstab für den Umfang des Arbeitsaufwandes ist ein gefährlicher und irreführender Mythos. Der Begriff will uns glauben machen, Bearbeiter und Monate seien austauschbare Faktoren“

Fred Brooks in „Vom Mythos des Mann-Monats“



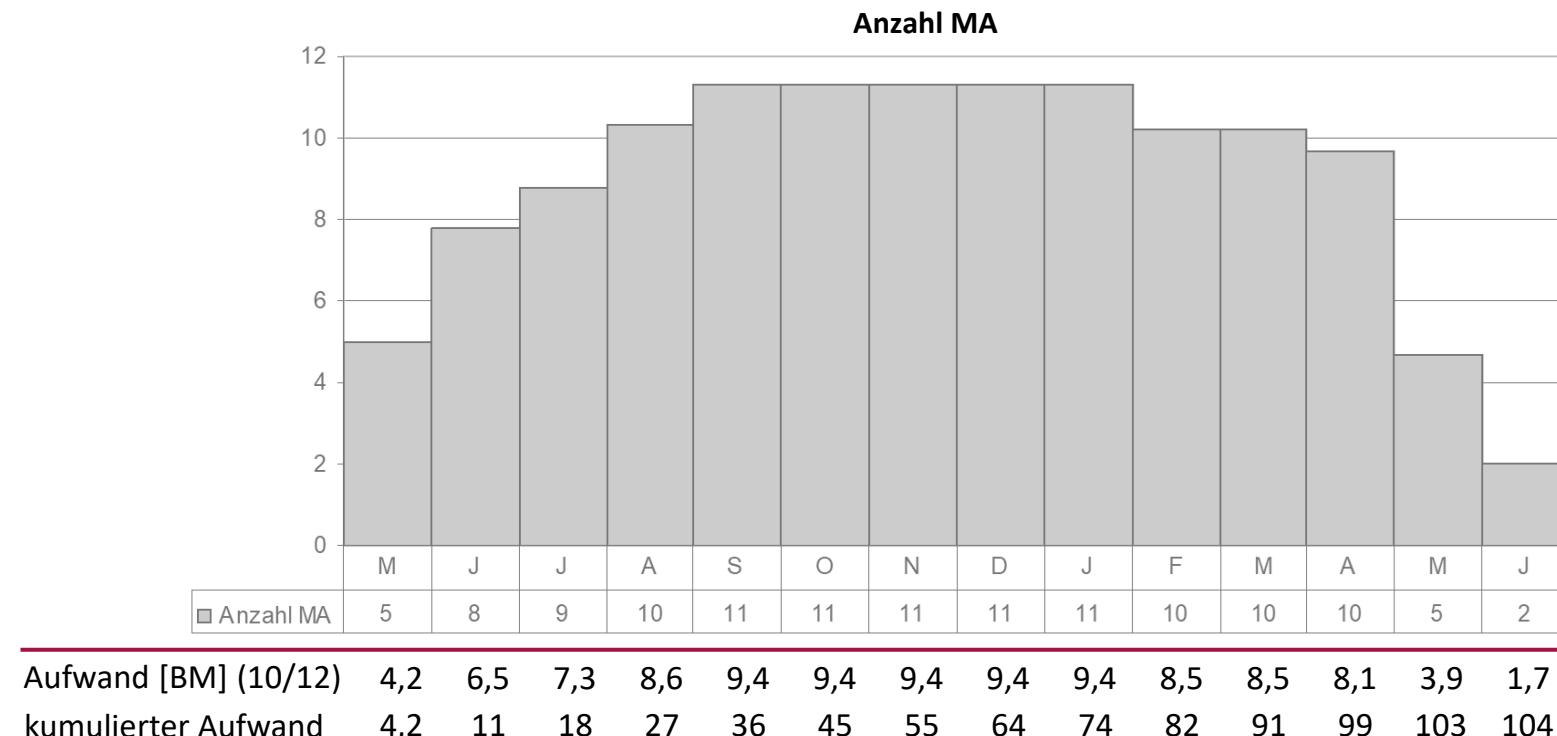
# Die Aufwandsschätzung wird durch ein Mitarbeitergebirge plausibilisiert

- Den Projektablauf mit geschätzter Dauer und Teamgröße skizzieren
- Fläche ausrechnen, hier: 30 Zeitmonate (ZtM)
- 1 ZtM = 0,8 BM wegen Feiertagen, Fortbildung, Krankheit, Meetings, etc.
- Hier ergibt die Umrechnung von ZtM auf PM:  $30 * 0,8 = 24$  PM
- Passt das zur Aufwandsschätzung?



# Aus dem Mitarbeitergebirge und dem Gesamtaufwand kann die Projektdauer ermittelt werden

In diesem Beispiel wurde der Gesamtaufwand von 104 PM auf 14 Monate verteilt:  
Maximum 11 Mitarbeiter, im Schnitt 8,9 Mitarbeiter bzw. 7,4 PM, Teamaufbau und maximale Teamgröße sind vernünftig.



# In die Budgetierung des Projekts fließen – neben dem Aufwand – weitere Parameter ein

Parameter	Methode	Erfahrungswert
Alle Parameter	Kalkulatorische Vorgaben	Konkret oder % von Bruttoaufwand
Stundensatz	Festlegung durch Management; nach Qualifikation oder Mischstundensatz	
Bruttoaufwand * Stundensatz	Durchschnittliche Stunden / Tag festlegen Überstunden kalkulieren	8 - 9 h / Tag
Reisekosten	Anzahl Reisen * durchschn. Kosten	bis zu 14 %
Festpreis Risikozuschlag		10 - 25 %
Gewährleistung		3 - 10 %
sonstige Kosten	Anschaffungskosten für Hardware, Software per Einkaufsliste	Nur „Durchreichen“ oder mit Zuschlag

# Zusammenfassung der Grundsätze

## Konkret

- Möglichst viele Aufwandsposten werden konkret geschätzt; möglichst wenige durch prozentuale Aufschläge bestimmt.
- 

## Schätzunsicherheit

- Zu jedem geschätzten Aufwandsposten wird die Schätzunsicherheit festgehalten. Für jeden Schätzposten wird dann aber nur eine Aufwandszahl festgehalten, welche die Grundlage für die spätere Projektplanung und die Kalkulation bildet.
- 

## Aufwandsblatt

- Das Ergebnis der Schätzung wird im so genannten Aufwandsblatt dokumentiert.
- 

## Vollständigkeit

- Über das Aufwandsblatt wird die Vollständigkeit und Plausibilisierung der Zahlen zueinander sichergestellt.
- 

## Prämissen

- Häufig stößt man an Grenzen (weil etwas nicht sauber spezifiziert ist, weil etwas unklar ist, weil etwas vergessen wurde). In diesem Fall formuliert man Prämissen für die Schätzung, die Grundlage des Angebots werden.
-

# Software Engineering in der industriellen Praxis

## Projektmanagement: Aufwandsschätzung

1. Grundlagen und Begriffsdefinitionen

---

2. Bottom-Up Schätzung (Expertenschätzung)

---

3. Top-Down Schätzung (Use Case Points)

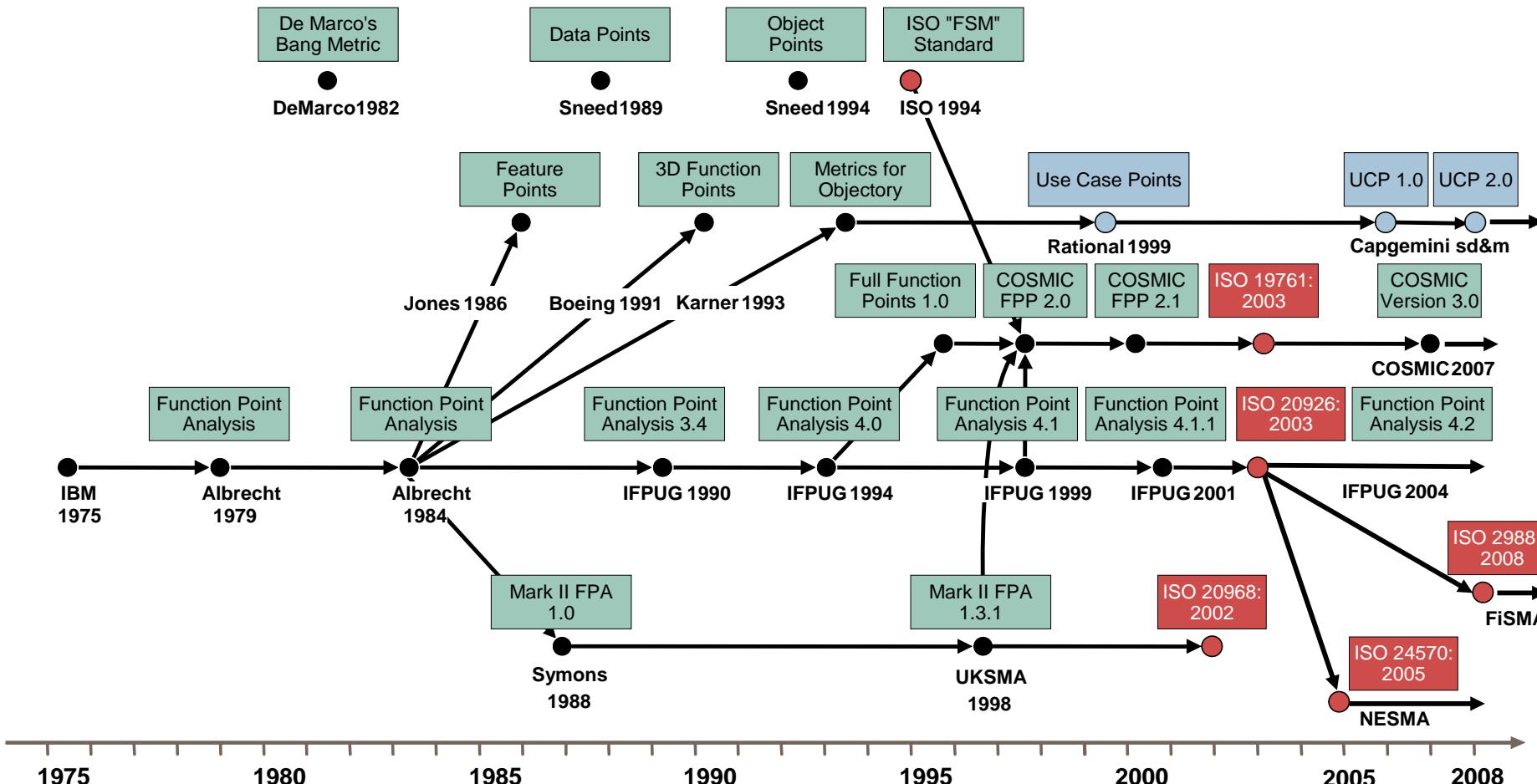
---

4. Literatur

## Motivation zum Einsatz von Use Case Points

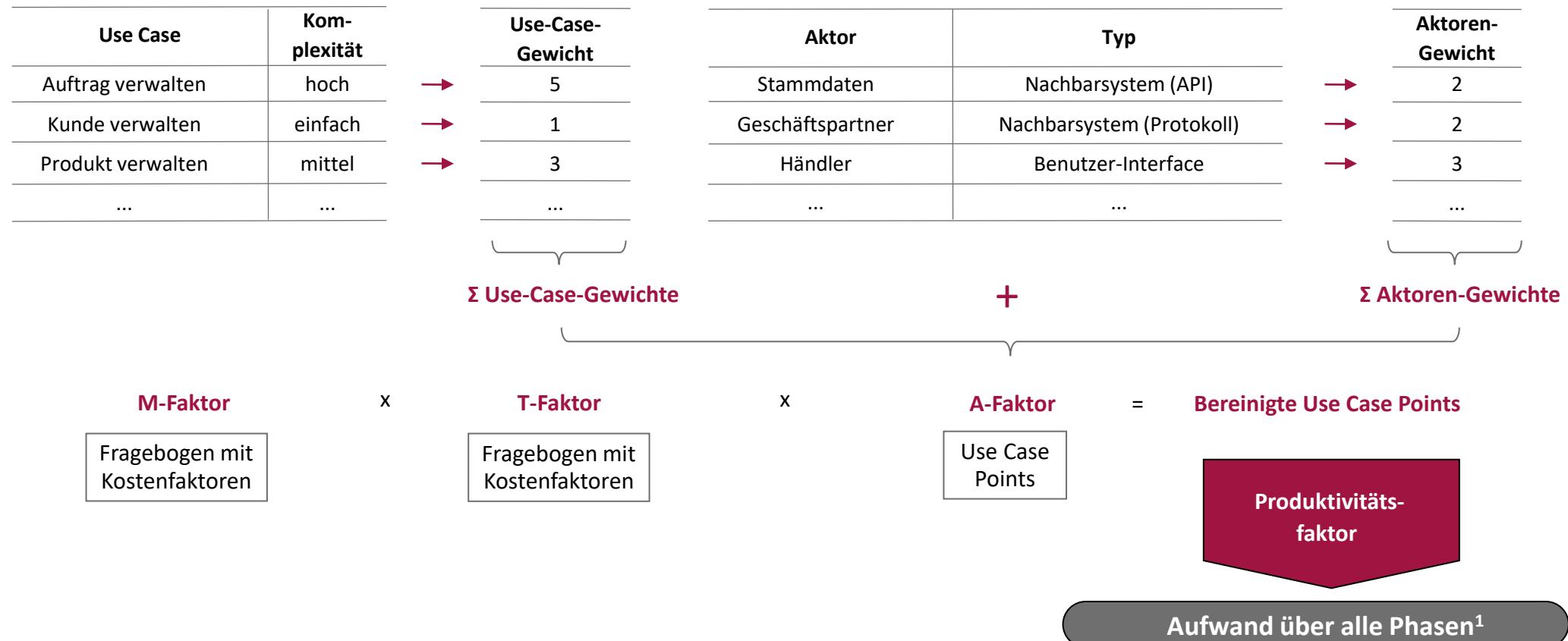
- **Top-Down Schätzmethode** zur schnellen Schätzung von Projekten
- Gesamthafte Schätzung des Projektaufwandes mit Hilfe von **mathematischen Algorithmen**
- Schätzung basiert auf **funktionale Anforderungen** (Use Cases)
- Einsatz in der Regel nur zur **Plausibilisierung der Expertenschätzung**
- Ermöglicht aber auch **schnelle (grobe) Budgetabschätzungen**
- Je mehr Schätzungen in einem Umfeld gemacht werden, je genauer kann Methode werden

# Entwicklung der funktionalen Größenmessung



Quelle: Lother, M.; Dumke, R.: Points Metrics - Comparison and Analysis. in: Dumke et al (Eds.): Current Trends in Software Measurement – Proceedings of the 11th IWSM, Montréal, Shaker Verlag. Aachen. pg: 228-267. 2001; ergänzt durch S. Frohnhoff, sd&m AG

Die Use Case Points (UCP) Methode setzt direkt auf einer Use Case basierten Spezifikation auf und ist sehr einfach anzuwenden



ABC Individuelle Analyse

→ Berechnung nach Standard-Metrik  
(einfach, mittel, komplex)

■ Berechnung nach firmeneigener Metrik

1) gemäß Mapping auf Aufwandsmodell

# Die UCP-Methode setzt eine fachliche Größenbestimmung voraus

Geeignet	Nicht geeignet
<ul style="list-style-type: none"><li>• Individualentwicklung</li><li>• Neuentwicklung</li><li>• Neuentwicklung fachlicher Geschäfts-prozesse in betrieblichen Anwendungen</li><li>• Stammdaten-Pflegesysteme</li></ul>	<ul style="list-style-type: none"><li>• Produktanpassungen</li><li>• Wartung, d.h. geringfügige Anpassung bestehender Systeme</li><li>• Technikstufen, Steuerungssysteme</li></ul>



**Methode ist ungeeignet,  
wenn Umfang von System-Anpassungen nur schlecht durch Use Cases erfasst wird,  
z. B. bei technischen Stufen, in denen sich die Fachlichkeit (A-Faktor) nur wenig ändert**

# Software Engineering in der industriellen Praxis

## Projektmanagement: Aufwandsschätzung

1. Grundlagen und Begriffsdefinitionen

---

2. Bottom-Up Schätzung (Expertenschätzung)

---

3. Top-Down Schätzung (Use Case Points)

---

4. Literatur



## Literatur

- [https://www.msg.group/images/msggroup/services/techrefresh/Dissertation\\_Use\\_Case\\_Points\\_3.0\\_Frohnhoff\\_msg.pdf](https://www.msg.group/images/msggroup/services/techrefresh/Dissertation_Use_Case_Points_3.0_Frohnhoff_msg.pdf) → Suche nach „UCP“
- Balzert, H.: Lehrbuch der Software-Technik, Band 1, Software-Entwicklung. Spektrum Akademischer Verlag, 2. Auflage, 2000.
- Siedersleben, J.: "Softwaretechnik - Praxiswissen für Software-Ingenieure" 2. überarbeitete und aktualisierte Auflage, Hanser Verlag, 2003.
- Frohnhoff, S.; Jung, V.; Engels, G.: "Use Case Points in der industriellen Praxis" In "Applied Software Measurement - Proceedings of the International Workshop on Software Metrics and DASMA Software Metrik Kongress", Abran, A. et al. Eds. Shaker Verlag, 2006, pp. 511-526
- Cockburn, A.: "Writing Effective Use Cases", Addison-Wesley, 2001.
- Smith, J.: „The Estimation of Effort Based on Use Cases“, Rational Software, Cupertino, CA.TP-171, October 1999.  
<http://whitepapers.zdnet.co.uk/0,39025945,60071904p-39000629q,00.htm>

# Kontakt



Christian Schmitz  
+49 (170) 9241329  
christian.schmitz@msg.group

msg systems ag  
Robert-Bürkle-Straße 1  
85737 Ismaning

+49 89 96101-0  
+49 89 96101-1113

info@msg.group

value – inspired by people