

# Projektmanagement

---

*Planung Teil II*

Teil 5 - Projektmanagement - WS 2015/16

Jörg Pechau  
Department Informatik, Uni Hamburg

# Themen heute

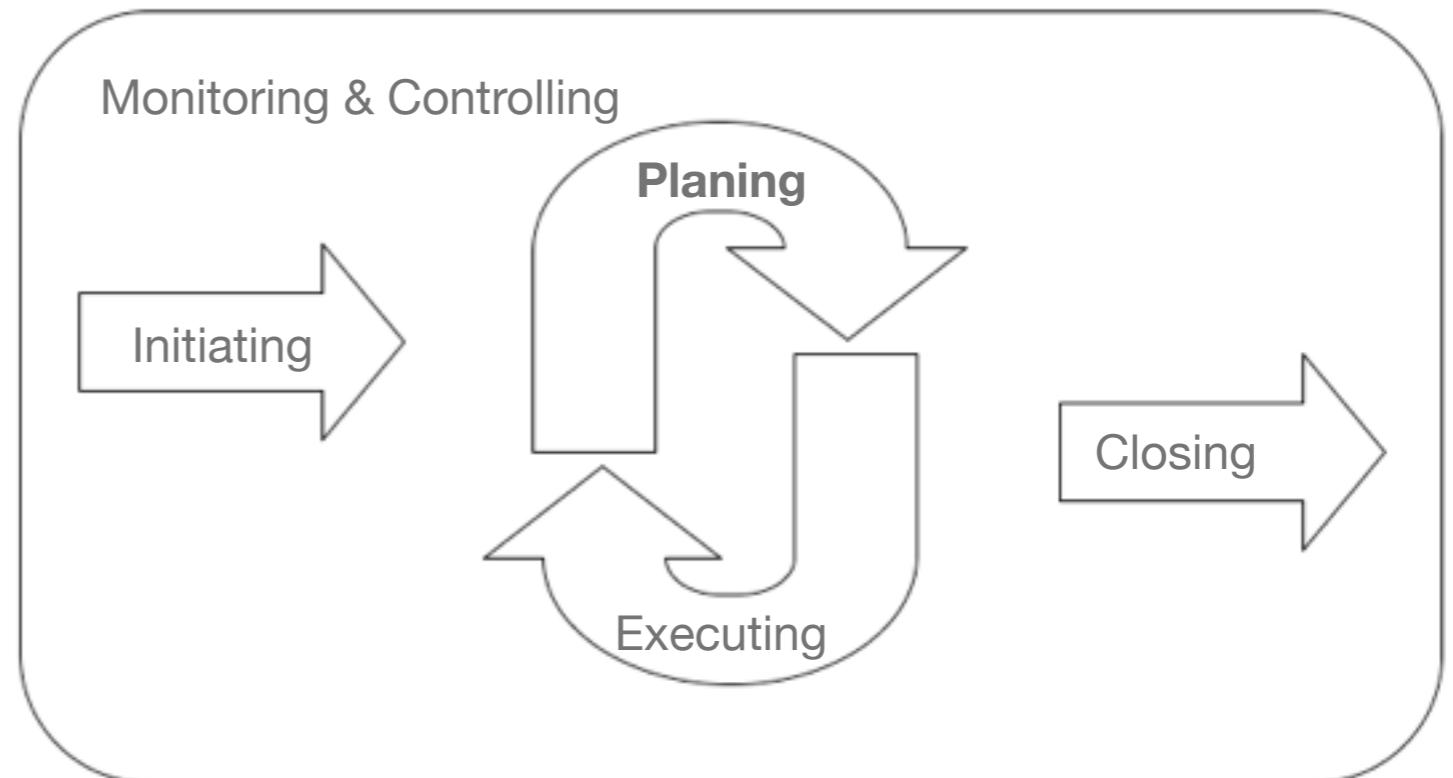
---

- „Planung“ - Aufwand feststellen
  - Was ist Aufwand? Was ist Dauer?
  - Was bedeutet Aufwand schätzen?
  - Wie schätze ich?
  - Potenzielle Fehlerquellen

# Kurze Erinnerung

# Situation & Ziel der Prozessgruppe „Planing“

- Situation
  - Ein Projektauftrag liegt vor oder ist im Entstehen
  - Ein Projekt oder eine -phase soll gestartet, ausgeführt und gesteuert werden
- Ziele
  - Entwickeln eines Projektplans
  - Aktualisieren eines Projektplans



# Planen

---

- Projektplanung ist ein „Dreikampf“ aus

- Zerlegen des Projekts in Strukturelemente
- Schätzen des Aufwands



- Anordnen der Strukturelemente

**Planen ist Team-Aufgabe und ist Kernverantwortung des Project Managements.**

- Aus Planung & Rahmenbedingungen leiten wir Aussagen ab



- Projektdauer
- Projektkosten, Ressourcen- und Personalbedarf
- Über diese Schritte iterieren, bis als Ergebnis eine sinnvoller Projektplan entstanden ist

# Projektstrukturplan (PSP)

- „Aka“ **Workbreakdown Structure (WBS)**

- **Zerlegung**

Ein PSP ist keine vollständige Liste aller einzelnen Arbeitsschritte, sondern eine umfassende Gliederung des Projektumfangs!

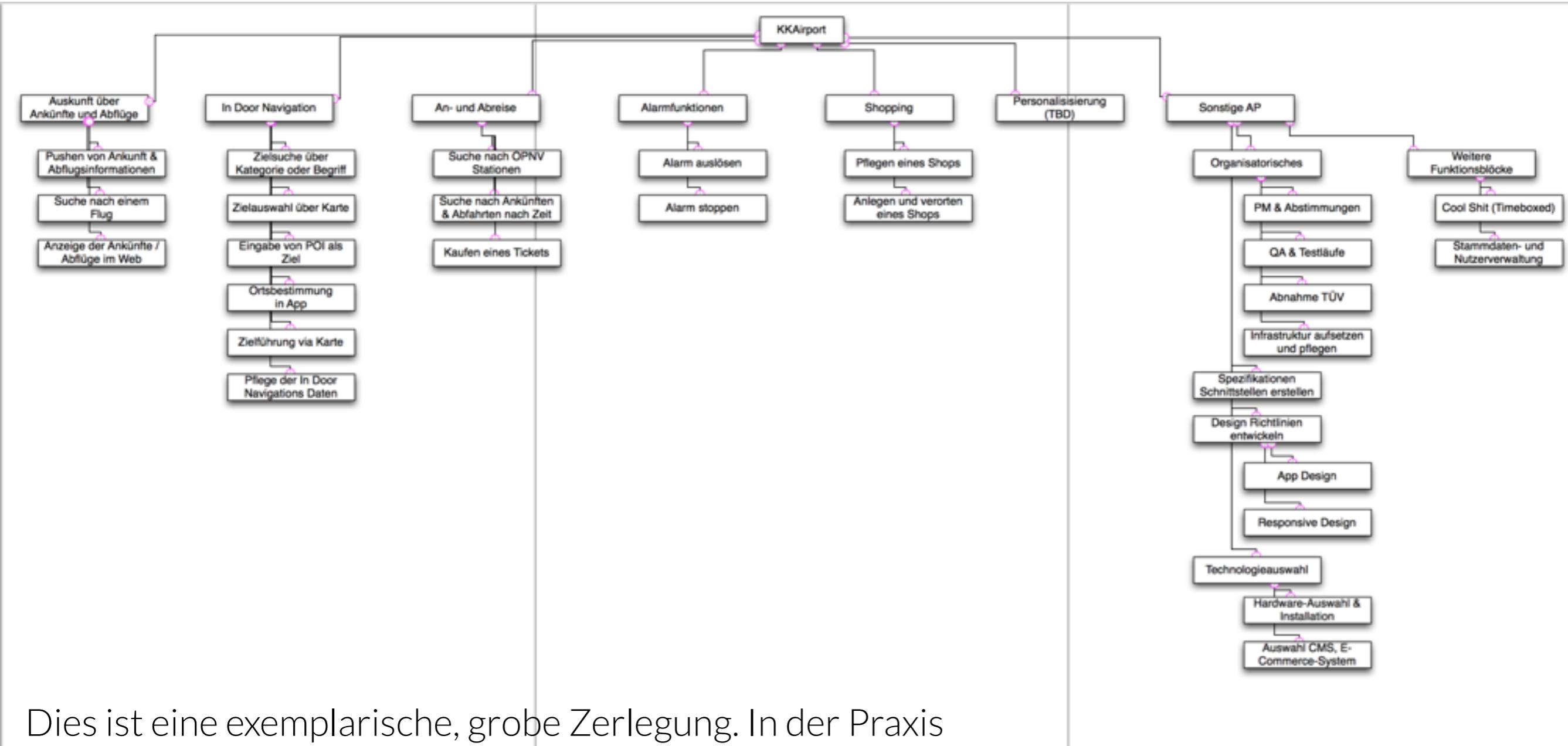
- „Was“ ist wichtig, nicht „Wie“

- Balance zwischen zu viel und zu wenigen Details



# Musterlösung

# Ein fachlich/funktional strukturierter PSP



Dies ist eine exemplarische, grobe Zerlegung. In der Praxis  
bräuchten wir noch weitere Ebenen und müssten noch mehr  
Elemente ergänzt werden.

# Snapshot Projektakte

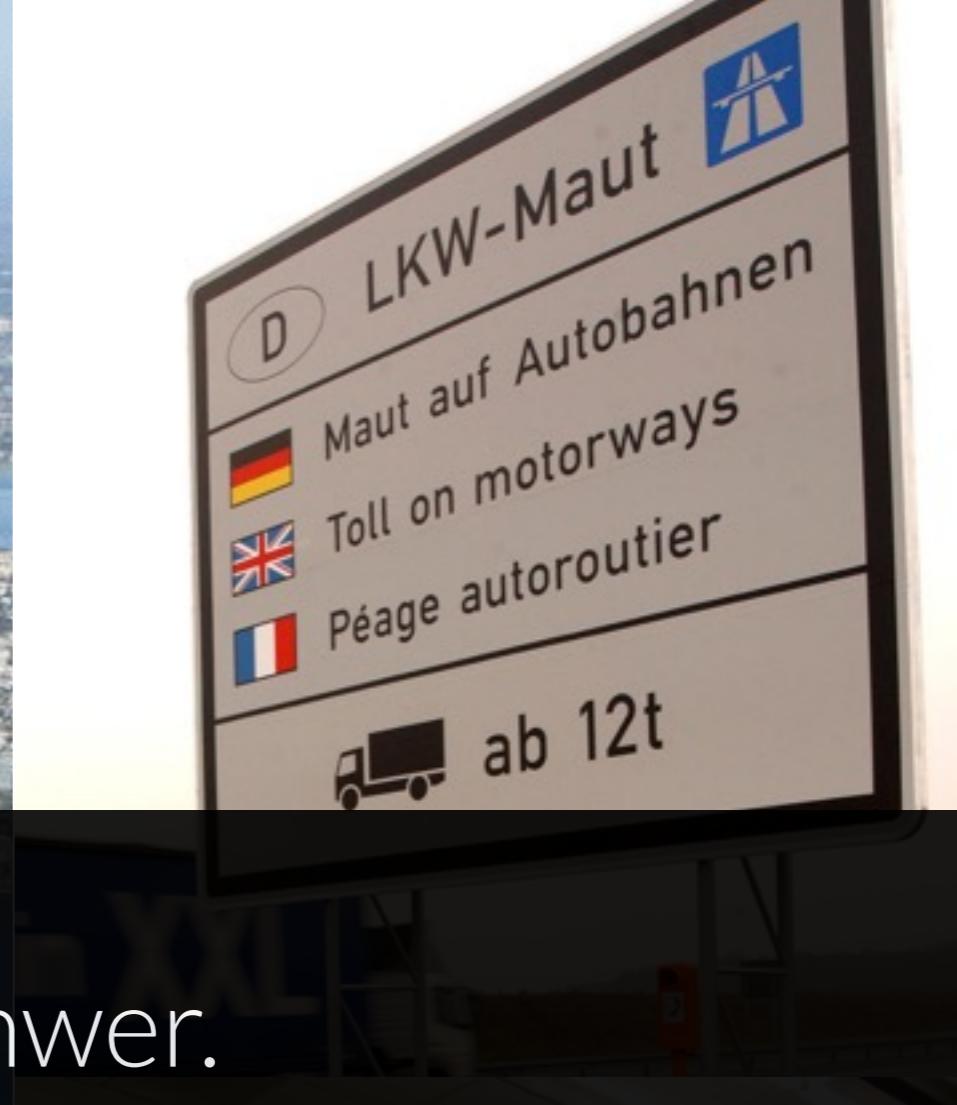
Name		Änderungsdatum	Größe	Art
▼ 00 - Projektauftrag inkl. Spezifikationen		04.11.2015, 19:26	--	Ordner
□ Projektauftrag - V1.3.pages		06.11.2014, 22:28	72 KB	Pages
▼ 05 - Planungen		Heute, 00:22	--	Ordner
☒ PSP-Musterlösung.png		Heute, 00:18	1 MB	PNG-Bild
► 10 - Kommunikation inkl. Kontaktdaten		04.11.2015, 19:18	--	Ordner
► 20 - Protokolle inkl. (Zwischen-)Abnahmen		04.11.2015, 19:21	--	Ordner
► 30 - Reporting		04.11.2015, 19:18	--	Ordner
► 40 - Rechnungen		04.11.2015, 19:21	--	Ordner
► 50 - Präsentationen		04.11.2015, 19:21	--	Ordner
► 60 - Lieferungen von Kunden		04.11.2015, 19:22	--	Ordner
► 70 - Lieferungen an Kunden		04.11.2015, 19:22	--	Ordner
► 99 - Archive		04.11.2015, 19:18	--	Ordner

# Summary

---

- **Andere Darstellungen eines PSP sind denkbar**
  - Wenn es nur um einen PSP geht, sind die Formen „Organigramm“ und Tabelle am besten geeignet
  - Allerdings lässt sich über „Bilder“ (Organigramm, Mindmap) leichter diskutieren, als über ein „Spreadsheet“
  - Visualisierung ist bei fast jeder Planungsarbeit besonders in Teams essentiell
  - PSPs können auch an Metaplanwänden entwickelt werden
- **Zerlegung**
  - Erfordert Domain-Wissen und Übung für notwendigen Detail-Grad
- **Selbsttest**
  - **Traue ich mir zu ein Projekt nach dieser Zerlegung weiter zu beplanen und anschließend durchzuführen?**

Aufwandsschätzung  
oder  
„The Black Arts of Project Management“



Sicher schätzen ist schwer.



# Aufwand vs. Dauer

---

- Definition Aufwand

[Aufwand/Effort ist] Die zum Abschließen eines Terminplanvorgangs oder einer Komponente eines Projektstrukturplans erforderliche Anzahl von Arbeitseinheiten. **[PMBOK]**

- Definition Dauer

[Dauer ist die] Zeitspanne vom Anfang bis zum Ende eines Vorgangs. **[DIN 69900-1]**

# Aufwand vs. Dauer

---

- **Aufwand ist ein Maß für eine zu erbringenden Leistung, z.B.**
  - 40 Personenstunden für Feature X
  - 10 Personentage für Story Y
  - 5 Personenjahre für Projekt Z
- **Dauer ist ein Maß für den Zeitraum innerhalb dessen eine Leistung erbracht wird, z.B.**
  - Feature X umzusetzen hat 2 Wochen gedauert
  - Story Y umzusetzen hat 6 Monate gedauert
  - Projekt Z umzusetzen hat 6 Monate gedauert

# Beispiel: Dauer vs. Aufwand

---

## Sei

Teamgröße 5  
Sprintdauer 3 Wochen  
Wir führen 5 Sprints durch  
Das Team arbeitet Vollzeit in allen Sprints

## Frage

Wir wollen eine Schätzung für den Aufwand eines Projekts ermitteln!  
Welche Dauer liegt vor?  
Welcher Aufwand liegt vor?

## Antwort

Dauer == 105 d  
Aufwand == 375 PT

## Warum?

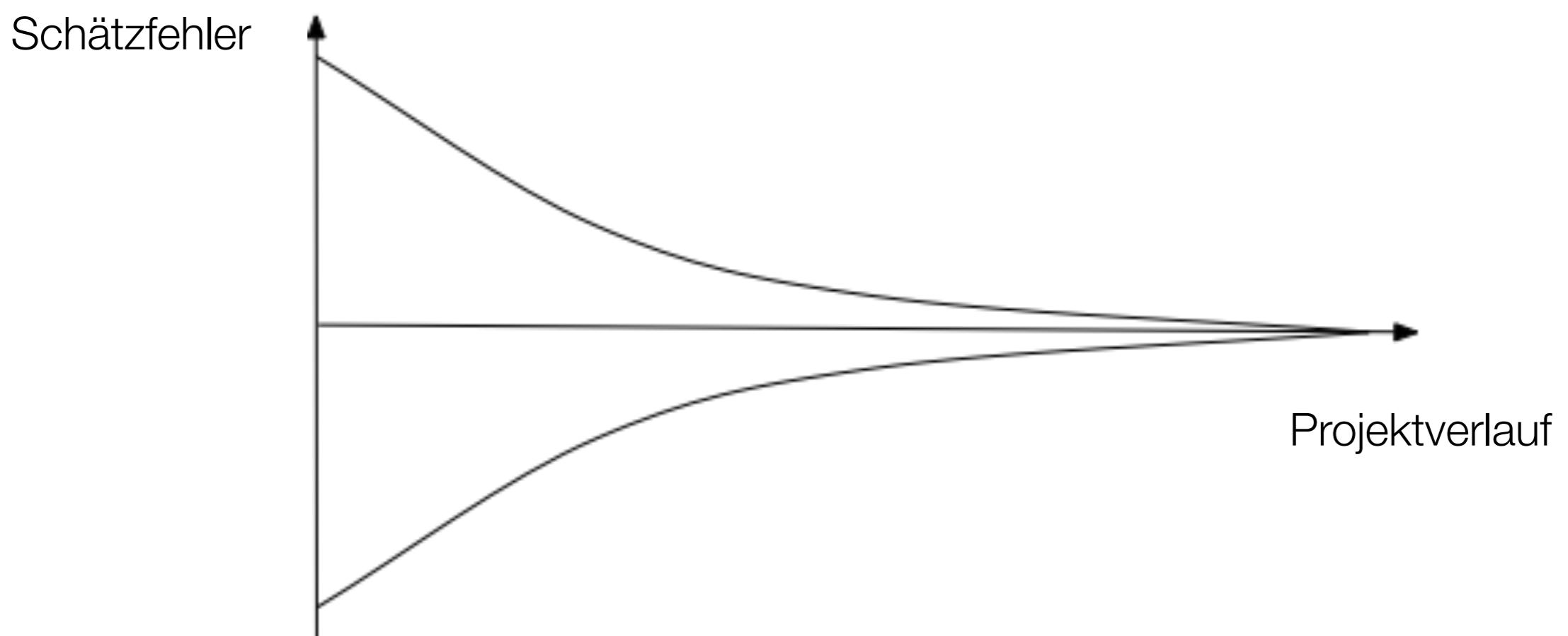
Dauer = Anzahl Sprints \* Wochen pro Sprint \* Tage pro Woche  
Aufwand = Anzahl Sprints \* Wochen pro Sprint \* Arbeitstage pro Woche \* Teamgröße

# Was bedeutet Aufwandsschätzung?

---

- Definition Aufwandsschätzung

Eine Aufwandsschätzung ist die **Prognose** des zur Erreichung eines (Teil-) Projektziels erforderlichen Aufwands. [PM-Glossar]



Simplified „Cone of Uncertainty“ [Boehm 81] / [McConnell]

- Schätzen bedeutet,
  - eine Größe **näherungsweise** zu bestimmen.
  - dass wir in letzter Konsequenz auch „daneben“ liegen können.
  - dass wir in unseren Schätzung nach „oben“ und „unten“ abweichen können.
- Aufwandsschätzungen
  - Sind **Annahmen** über die Zukunft - mit Wissen von heute...
  - Werden im Projektverlauf exakter, bleiben aber während des Projekts ungenau
  - Sind erst nach Projektende exakt, z.B. bei einer Nachkalkulation...

# Warum überhaupt Aufwände schätzen?

---

- Gewinnen einer Aussage, d.h. abgeben einer Prognose über die Zukunft, über
  - Dauer
    - Umfang der Anforderungen & Ressourcen-Einsatz => Mindestdauer
  - Umsetzbare Anforderungen
    - Ressourcen-Einsatz & Zeitrahmen => Maximal umsetzbare Anforderungen
  - Kosten „klassisch Phasenorientiert“
    - Umzusetzende Anforderungen & Ressourcen-Einsatz => Kosten
  - Kosten „Time Boxing“, z.B. Sprints bei Scrum (agile)
    - Dauer & Ressourcen-Einsatz => Kosten

# Wann schätze ich Aufwände?

---

- Vor Projektbeginn, um Machbarkeit zu beurteilen, z.B. Chancen & Risiken
- Während des Projekts
  - Um Projektfortschritt einzuschätzen, z.B. verbleibender Umfang...
  - Um Planänderungen abzuschätzen, z.B. Machbarkeit, Konsequenzen...
  - Um Arbeit zu organisieren, z.B. für Priorisierungen...
- Nach dem Projekt, als Nachkalkulation, um aus dem Projektverlauf zu lernen

# Wer schätzt?

---

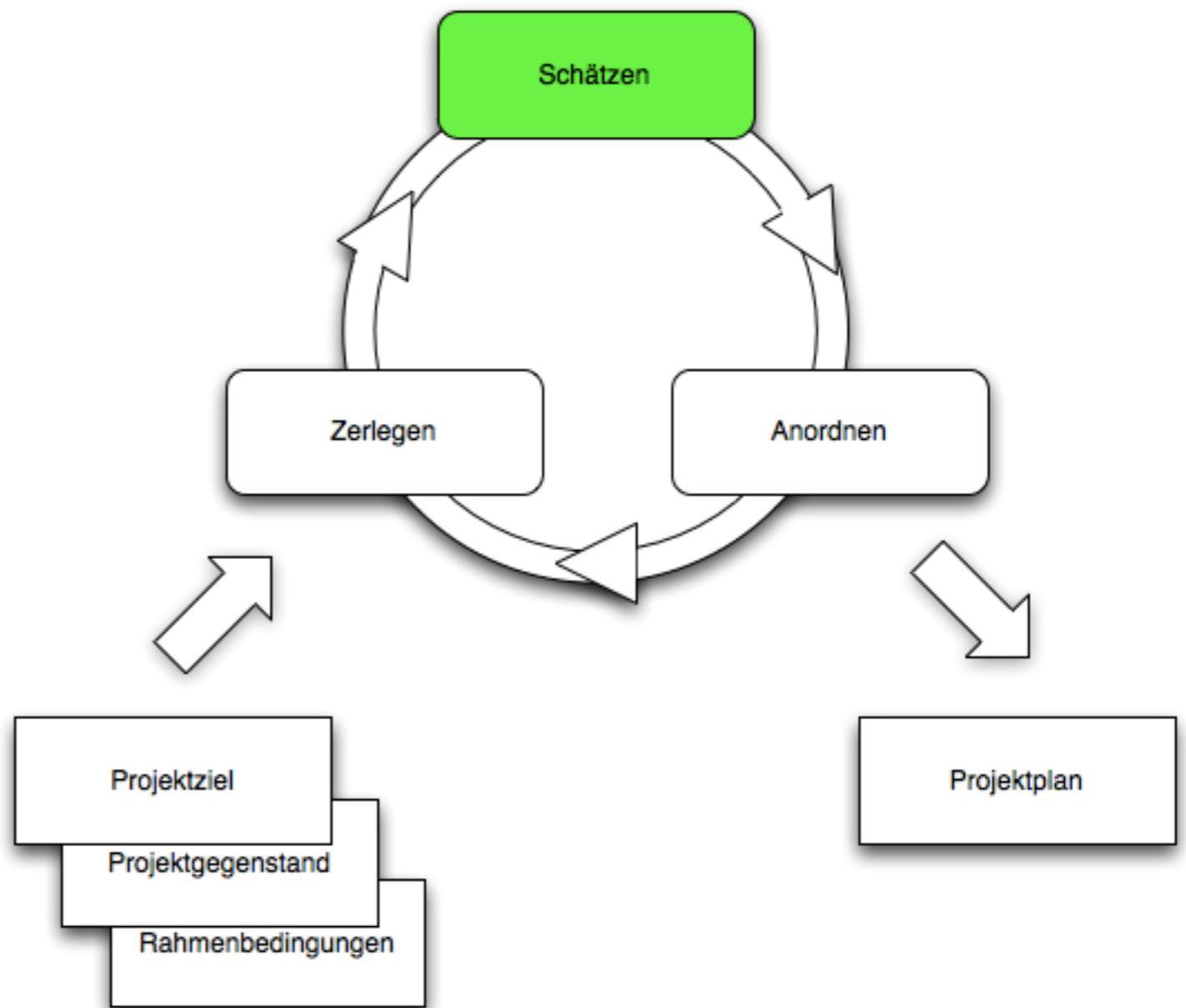
- Diejenigen, die es in erster Linie betrifft, z.B. Entwicklungs-Team
  - Commitment für Umsetzung
  - Erhöhung der Qualität
- **Diejenigen, die Betroffene zu Beteiligten machen!**
  - Um überhaupt schätzen zu können
  - Erhöhung der Qualität
- **Domänen-Expertise gehört stets ins Schätzteam!**

# Schätzen und Messen - Zwei Seiten einer Medaille!

---

- Schätzen und messen, um zu steuern (Monitoring & Controlling)
  - Stand eines Projekts, Projektfortschritt, z.B. realisierte Tasks, Stories...
  - Kosten, Kostenentwicklung
- Schätzen und messen, um zu lernen
  - Kalibrierung des Schätzvorgehens

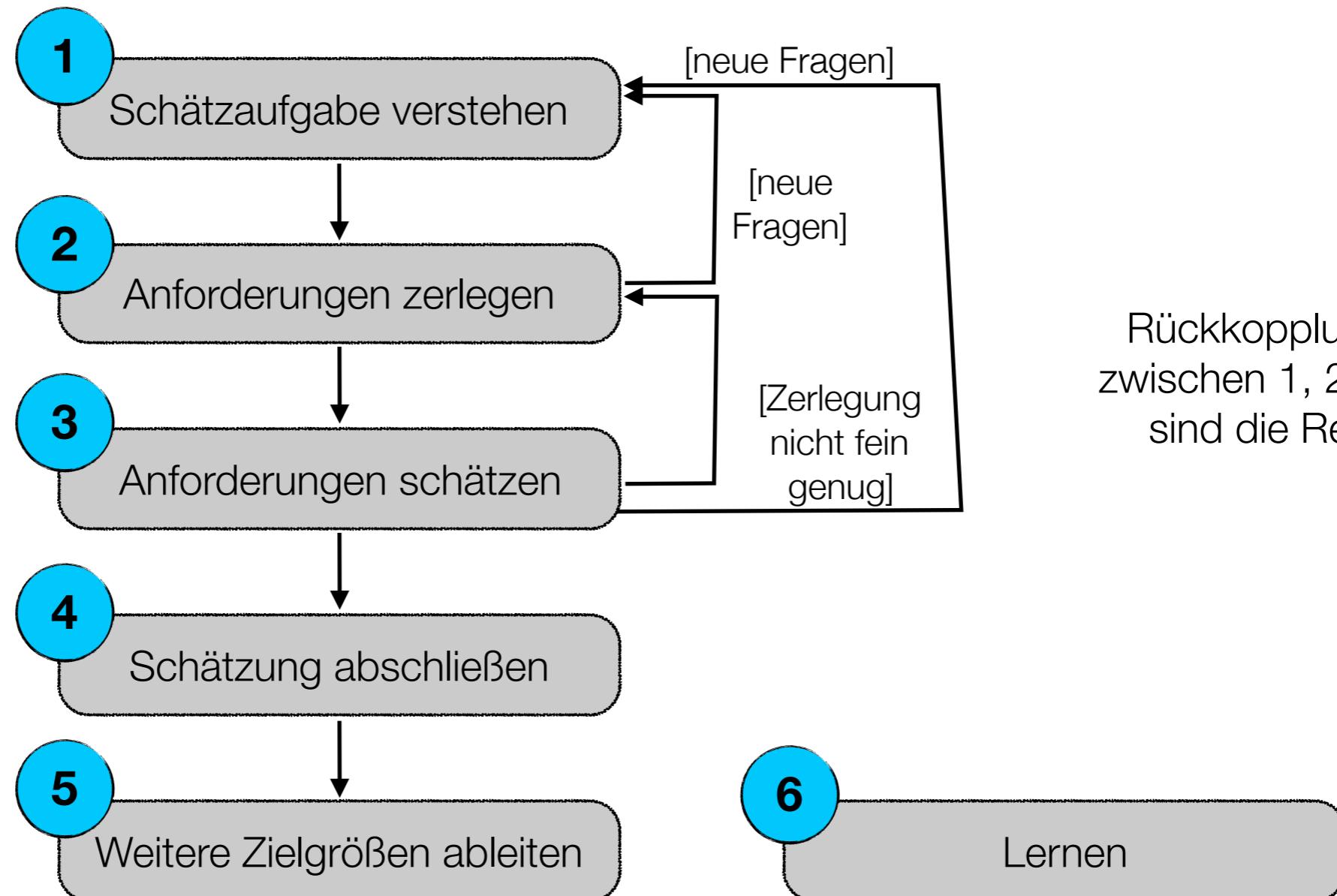
# Schätzen



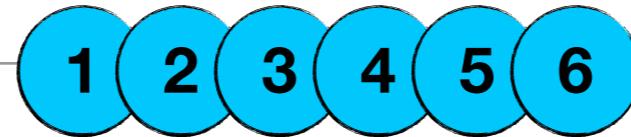


# Wie komme ich zu einer Aufwandsschätzung?

# Eigentlich ist es ganz einfach...



# Zwei begleitende Beispiele



- **Weiterentwicklung der UIs des firmeneigenen Content Management Systems (CMS)**
  - Bereits vorhandene User Stories zur Erweiterung des UIs sollen durch interne Entwicklungsprojekte implementiert werden.
- **Ein Festpreisprojekt**
  - Eine Website soll „schlüsselfertig“ auf Basis des eigenen CMS für einen externen Kunden entwickelt und an Betrieb übergeben werden.

# 1

# Schätzaufgabe verstehen

---

- Was ist das Projektziel?
- Was und wofür schätze ich eigentlich, kann ich das überhaupt schätzen?
- Wie genau soll die Schätzung sein?
  - Genauigkeit, z.B. Schätzung in Personenjahre, -Monate oder -Stunden  
Projektauftrag und PSP sollten uns die meisten Antworten liefern!
  - Zerlegungstiefe, z.B. Schätzung auf Gesamtsystem- oder auf Umsetzungsebene
- **Was sind die funktionalen und nicht funktionale Anforderungen?**
- Was sind die Rahmenbedingungen des Projekts?
- **Wie viel Geld und Zeit kann ich in die Aufwandsschätzung investieren?**
- Die Schätzaufgabe verstehen, bedeutet die wesentlichen funktionalen, nicht funktionalen, organisatorischen und soziologischen Aspekte des Projekts zu kennen.

# Begleitende Beispiele - Schätzaufgabe verstehen

## • Weiterentwicklung des UIs des CMS

- Schätzen zur Priorisierung der User Storys & Entwicklung einer Roadmap
- Exakte Schätzung nicht notwendig, relatives abstraktes Schätzmaß „Mäuse, Schafe, Elefanten“
- Anforderung auf hohem Level gut verstanden, User Storys liegen vor
- Rahmenbedingungen: Agiles Vorgehen, „Time boxing“, Team bekannt

## • Ein Festpreisprojekt

- Detaillierte Aufwandsschätzung für Festpreisangebot in realen Personentagen für Kosten und notwendige Team-Größe.
- Anforderungskatalog liegt vor. Es existiert eine Anzahl vergleichbarer Projekte.
- Rahmenbedingungen: Budget und Zieltermin sind bekannt. Vorgehen klassisch, iterativ inkrementell

## 2

# Anforderungen zerlegen

---

- Der PSP
  - Ist der Ausgangspunkt der Schätzung
  - Arbeitspakete erfassen funktionale und nicht funktionale Anforderungen.
- Initiale Anforderungen häufig zu grob zerlegt
  - Zerlegen in feinere funktionale und nicht funktionale Anforderungen
  - **Führt in der Regel zurück zu „Schätzaufgabe verstehen“!**
  - Kann dazu führen, den PSP weiter zu verfeinern!
- Zerlegen bis ausreichende Detailtiefe erreicht ist, ab der ich schätzen kann

## Begleitende Beispiele - Anforderungen zerlegen

- **Weiterentwicklung des UIs des CMS**

- Keine weitere Zerlegung nötig
  - Ersteller der User Storys schätzten selber
  - Für eine High Level Schätzung reichten die Informationen

- **Ein Festpreisprojekt**

- Komponenten-orientierte Zerlegung der Anforderung bis auf Arbeitspaket-Ebene
- Ergänzung nicht funktionaler Anforderungen aus Erfahrungsschatz
- Mehrmaliges Rückkoppeln mit Kunden, um Anforderungen zu verstehen

## 3

# Anforderungen schätzen

- Schätzen bedeutet, eine Analogie zu etwas Bekanntem bilden, d.h. raten.
- Pro Anforderung Realisierungsaufwand bestimmen, d.h.
  - Schritte überlegen, die ich zur Realisierung einer Anforderung benötige
    - dazu Anforderungen auf eigene Erfahrungen oder Wissen abbilden  
Dies ist das eigentliche Schätzen!
    - Realisierungsaufwand in abstrakten (relativen) oder konkreten Maß ausdrücken
  - Dieser Schritt führt häufig zu Schritt 1 und/oder Schritt 2 zurück.
- Das Ergebnis sind die geschätzten Anforderungen
- Analogiebildung (raten) ist Element eines jeden Schätzvorgehens!

## 3

## Schätzmaße

- Abstrakte (relative) Schätzmaße setzen Aufwand in Relation: A ist größer als B ist größer als C
  - Beispiele: Story Points, Gummibärchen, „Mäuse, Schafe, Elefanten“
  - Vorteil: Teams haben i.d.R. gute Vorstellung, über Größenordnungen von Anforderungen
  - Nachteil: Wir können daraus keine direkte Aussage über Dauer und Kosten ableiten  
*Work expands so as to fill the time available for its completion.* [Parkinson]
- Konkrete Schätzmaße
  - Beispiele: Personenjahre, -Monate, -Tag an Aufwand oder Jahre, Monate, Tage
  - Vorteil: Aufwand muss nicht interpretiert werden, um Aussage über Dauer zu gewinnen
  - Nachteile: Unterstellt gleiche Umsetzungsgeschwindigkeit für alle und definiert Minimalaufwand

# Unterstützende Techniken

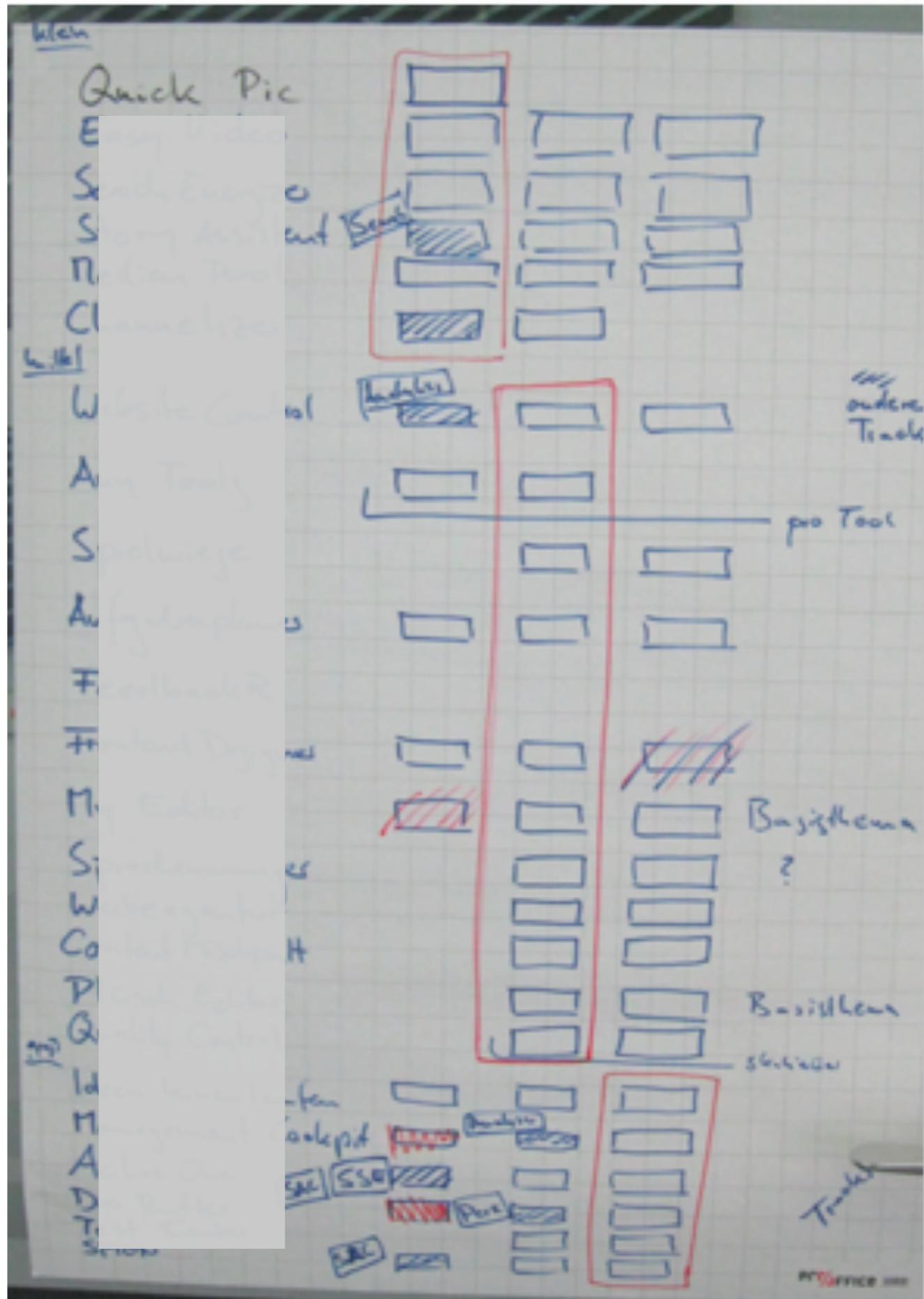
---

- Beispiele
  - 3-Point-Estimation („Best, worst & realistic Case“)
  - Wideband Delphi-Methode (Moderierte Schätzklausur)
  - Expert Opinion
- Es gibt viele Methoden, meistens werden Kombinationen davon genutzt
- Wenn möglich stets mehr als eine Person schätzen lassen!

# Begleitendes

## 3 Beispiel UI

- Relatives Schätzmaß (Maus, Schaf, Elefant)
- Schätzvorgehen
  - 8 Personen Schätz-Team
  - Expert-Opinion
  - Wideband Delphi Variante



# Begleitendes Bsp. Festpreis

Schätzreihe: Ideale Stunden

Arbeitspaket	Kommentar	Milestone	Schätzer 1			Schätzer 2			Schätzer 3			Schätzer 4		
			best	expected	worst									
<b>Standard API Adapter für Editor</b>														
Infrastruktur	Infrastruktur so ausbauen, daß es möglich ist, die Testabdeckung aus IDEA heraus zu bestimmen, Testabdeckung in IDEA feststellen, so ermittelte Testabdeckung auf mehr als 90% erhöhen		1,00	25,50	2,50	403,00	552,00	795,00	405,00	603,00	800,00	403,00	552,00	795,00
API implementieren	50 structs, 67 interfaces, 207 methods - nur direkt aufs API übertragbare Konstrukte, clientseitige Alt-Implementierung ersetzen		475,02	656,39	874,00	1250,00	1656,00	2050,00	1200,00	1720,00	2500,00	403,00	552,00	795,00
Listener	nutzt nur der Editor, kein Äquivalent im Standard API		712,45	923,85	1131,64	16,00	40,00	80,00	16,00	72,00	100,00	1250,00	1656,00	2050,00
Query	nutzt nur der Editor, kein Äquivalent im Standard API		0,00	16,00	19,15	16,00	40,00	100,00	16,00	72,00	120,00	16,00	40,00	80,00
Converter	nutzt nur der Editor, kein Äquivalent im Standard API		0,00	20,00	17,00	16,00	40,00	64,00	16,00	80,00	90,00	16,00	40,00	100,00
Sorting	nutzt nur der Editor, kein Äquivalent im Standard API, daher lokales Sortieren		0,00	12,00	0,00	16,00	40,00	64,00	16,00	64,00	64,00	16,00	40,00	64,00
Dokumentation	Anleitung zur Migration für Kunden, Releasenotes, Deployment-Diagramm		61,22	75,50	66,97	120,00	160,00	180,00	124,00	180,00	180,00	16,00	40,00	64,00
Tests	Funktionalität, Performance, Stabilität, Speicherbedarf		108,73	122,57	127,02	295,00	364,00	400,00	320,00	380,00	400,00	120,00	160,00	180,00
<b>Pusher auf Standard API umstellen</b>														
Modularisieren	aus Server herauslösen		61,78	53,22	50,23	8,00	24,00	32,00	8,00	32,00	64,00	115,00	120,00	132,00
Code Style	Java 5, Check Style conform - 398 Issues, Abhängigkeiten reduzieren und auf aktuellen Stand bringen		57,77	75,74	91,59	115,00	160,00	180,00	100,00	150,00	200,00	8,00	24,00	32,00
Umstellen	größere Teile sind bereits umgestellt, Standard API muß erweitert werden		44,79	63,12	62,70	35,00	40,00	64,00	40,00	64,00	80,00	115,00	160,00	180,00
Dokumentation	API Dokumentation		5,20	12,00	16,49	24,00	40,00	40,00	32,00	64,00	80,00	35,00	40,00	64,00
Tests	Funktionalität, Performance, Stabilität, Speicherbedarf		7,66	30,00	28,30	40,00	80,00	88,00	32,00	100,00	90,00	24,00	40,00	40,00
<b>Tracker auf Standard API umstellen</b>														
Modularisieren	aus Server herauslösen		24,25	24,00	19,04	8,00	16,00	32,00	8,00	32,00	64,00	50,00	64,00	72,00
Code Style	Java 5, Check Style conform - 262 Issues, Abhängigkeiten reduzieren und auf aktuellen Stand bringen		16,49	36,95	54,49	40,00	80,00	120,00	32,00	80,00	132,00	8,00	16,00	32,00
Umstellen	CORBA Client durch Unified API ersetzen		4,00	0,00	10,00	40,00	80,00	120,00	32,00	80,00	140,00	40,00	80,00	120,00
Dokumentation	Anleitung zur Migration für Kunden, Releasenotes, Unified API Dokumentation		12,00	25,61	25,61	24,00	32,00	72,00	16,00	40,00	80,00	40,00	80,00	120,00
Tests	Funktionalität, Performance, Stabilität, Speicherbedarf		10,07	21,04	4,00	32,00	64,00	64,00	8,00	72,00	72,00	24,00	32,00	72,00
<b>Central Server in die Cloud</b>														
Designentscheidung	Datenbankschema, Events, Caching, LDAP		9,45	19,19	18,48	24,00	40,00	64,00	12,00	55,00	64,00	4,00	16,00	32,00
Erstellen	JDBC User und Event Connector		41,19	49,89	73,20	100,00	120,00	180,00	90,00	132,00	200,00	24,00	40,00	64,00
Refactoring	Direkte JDBC Nutzer		18,76	20,62	91,47	64,00	80,00	100,00	72,00	90,00	320,00	100,00	120,00	180,00
Dokumentation	Anleitung zur Migration für Kunden, Releasenotes, Admin/Server/Entwickler-Handbücher, Admin/Entwickler-Schulungen, Sales-Folien		32,33	32,98	39,26	8,00	16,00	32,00	8,00	32,00	32,00	64,00	80,00	100,00
Tests	Funktionalität, Performance, Stabilität, Speicherbedarf		26,53	30,20	27,71	32,00	64,00	80,00	64,00	72,00	80,00	8,00	16,00	32,00
<b>Summe</b>						1573,00	2156,00	2838,00	1346,00	2495,00	3436,00	1379,00	1856,00	2472,00

## 4

# Schätzung abschließen: Summieren & skalieren

- Summieren
  - Ggf. Summe über die Umsetzungsschritte => Aufwand eines Arbeitspakets
  - Summe aller Aufwandsschätzungen => Idealer Gesamtaufwand
- Skalieren (nach „De Facto Industrie-Standard“)
  - Idealer Gesamtaufwand \* Korrekturfaktoren => Realer Gesamtaufwand
  - Korrekturfaktoren sind organisationsabhängig, z.B.
    - Risiko-Zuschläge, z.B. aufgrund von Team-Reife oder Rahmenbedingungen...
  - Alternativ füttern einer Schätzmethode (z.B. Function-Point, CoCoMo II) mit idealem Gesamtaufwand, drehen an div. Stellschrauben und Generierung eines Ergebnisses...

# Ausflug: Function Point Methode

4

- Anforderungen in relativen Maß schätzen

- Nicht funktionalen Anforderungen als Korrekturfaktoren

Ohne sorgfältige Empirie ist die Function Point Methode komplizierteres raten!  
 • Ergebnis: Für Anfänger nicht geeignet. Nicht zeitgemäß (- nicht loszuwerden).

- Kritik

- Auf Unternehmen zu kalibrieren
- Überholte (tech.) Grundannahmen
- Wasserfall geprägt

OUTPUTS :	<input type="text" value="20"/> * 4 = 80	<input type="text" value="10"/> * 5 = 50	<input type="text" value="3"/> * 7 = 21	151
INQUIRIES :	<input type="text" value="5"/> * 4 = 20	<input type="text" value="10"/> * 5 = 50	<input type="text" value="20"/> * 7 = 140	210
INPUTS :	<input type="text" value="20"/> * 3 = 60	<input type="text" value="10"/> * 4 = 40	<input type="text" value="5"/> * 6 = 30	130
FILES :	<input type="text" value="10"/> * 7 = 70	<input type="text" value="5"/> * 10 = 50	<input type="text" value="0"/> * 15 = 0	120
INTERFACES :	<input type="text" value="2"/> * 5 = 10	<input type="text" value="5"/> * 7 = 35	<input type="text" value="2"/> * 10 = 20	65
TOTAL UNADJUSTED FUNCTION POINTS :				676

## PROCESSING COMPLEXITY FACTORS

Factor	Value	Factor	Value
--------	-------	--------	-------

Data communications	<input type="text" value="0"/>	On-line update	<input type="text" value="0"/>
---------------------	--------------------------------	----------------	--------------------------------

Distributed data processing	<input type="text" value="0"/>	Complex processing	<input type="text" value="0"/>
-----------------------------	--------------------------------	--------------------	--------------------------------

Non-functional objectives	<input type="text" value="0"/>	Code reusability	<input type="text" value="0"/>
---------------------------	--------------------------------	------------------	--------------------------------

Tight configuration	<input type="text" value="0"/>	Conversion/installation ease	<input type="text" value="0"/>
---------------------	--------------------------------	------------------------------	--------------------------------

High transaction rate	<input type="text" value="0"/>	Operational ease	<input type="text" value="2"/>
-----------------------	--------------------------------	------------------	--------------------------------

On-line inquiry/data entry	<input type="text" value="0"/>	Multiple site installation	<input type="text" value="0"/>
----------------------------	--------------------------------	----------------------------	--------------------------------

End user efficiency	<input type="text" value="2"/>	Facilitate change	<input type="text" value="0"/>
---------------------	--------------------------------	-------------------	--------------------------------

TOTAL DEGREE OF INFLUENCE : 4

## FINAL CALCULATIONS

1. Adjustment Factor:  $0.65 + (0.01 * 4) = 0.69$

2. Final Function Point Count:  $676 * 0.69 = 466$  Function Points

Number of Person Months :

# Ausflug: Prinzip CoCoMo II

4

- COnstructive COst MOdel II von [Boehm]

- Berechnet Umsetzungsaufwand und Dauer aus bestimmten Eingangsgrößen nach empirisch belegten Teilmodellen

CoCoMo II benötigt sehr viel Empirie sonst endet auch dies im Raten,  
passt nicht zu agilem Vorgehen und ist deutlich zu komplex für Anfänger!

- Eingangsgrößen

- Systemgröße i.d.R. in FP oder LoC

- Software Cost und Scale Driver

- Ergebnis ist idealer Aufwand und Dauer

[CoCoMo II Webtool]

The screenshot shows the 'COCOMO II with Heuristic Risk Assessment' web tool. At the top, it displays the model as 'Post-architecture' and calibration as 'COCOMOII.2000'. Below this is a section titled 'Size' with a table for inputting project details:

	SLOC	% Design Modified	% Code Modified	% Integration Required	Assessment and Assimilation (0% - 8%)	Software Understanding (0% - 50%)	Unfamiliarity (0-1)
New	75000						
Reused	25000	50	100	100	5		
Modified	50000	100	100	100	3	50	0

Below the table, it says 'Rate each cost driver below from Very Low (VL) to Extra High (EH). For HELP on each cost driver, select its name.'

Cost drivers and their ratings (radio buttons):

- Precededness: VL, L, N, H, VH, EH
- Development Flexibility: VL, L, N, H, VH, EH
- Architecture/Risk Resolution: VL, L, N, H, VH, EH
- Team Cohesion: VL, L, N, H, VH, EH
- Product Attributes:
  - Required Reliability: VL, L, N, H, VH
  - Database Size: VL, L, N, H, VH
  - Product Complexity: VL, L, N, H, VH, EH
- Required Reuse Documentation: VL, L, N, H, VH
- Platform Attributes:
  - Execution Time Constraint: N, H, VH, EH
  - Main Storage Constraint: N, H, VH, EH
  - Platform Volatility: L, N, H, VH
- Personnel Attributes:
  - Analyst Capability: VL, L, N, H, VH
  - Programmer Capability: VL, L, N, H, VH
  - Personnel Continuity: VL, L, N, H, VH
  - Applications Experience: VL, L, N, H, VH
  - Platform Experience: VL, L, N, H, VH
  - Language and Toolset Experience: VL, L, N, H, VH
- Project Attributes:
  - Use of Software Tools: VL, L, N, H, VH
  - Multisite Development: VL, L, N, H, VH, EH
  - Required Development Schedule: VL, L, N, H, VH

Buttons at the bottom: Submit, Zurücksetzen, Suchen: Q.deat, Abwärts, Aufwärts, Hervorheben, Groß-/Kleinschreibung.

# Begleitende Beispiele - Schätzung abschließen

- **Weiterentwicklung des UIs des firmeneigenen CMS**

- Summieren: 35 Mäuse, 30 Schafe, 15 Elefanten
- Skalieren: Maus = 3 Story-Punkte; Schaf = 5 SP; Elefant = 10 SP
- Realer Gesamtaufwand: ca. 400 SP (=> abstraktes Maß!)

- **Ein Festpreisprojekt**

- Summieren - Idealer Gesamtaufwand: 150 Tage
- Skalieren 1 - Pauschaler Korrekturfaktor + 40%: 210 Tage
- Skalieren 2 - Aktive Entwicklungszeit (+ 35%): 285 Tage realer Gesamtaufwand
- Politischer „Deckel“ um Budget-Grenze von 250 Tagen zu halten

## 5

# Weitere Zielgrößen ableiten (Beispiele)

---

- Realer Gesamtaufwand \* Aktive Entwicklungszeit \* Team-Faktor => Mindestdauer
  - Aktive Entwicklungszeit: Exklusiv für das Projekt verfügbare Zeit
  - Team-Faktor: Verfügbarkeit, Qualifikation, Erfahrung...
  - Achtung: Reale Dauer hängt von weiteren Faktoren ab,
    - z.B. Anordnung der Arbeitspakete, Abhängigkeiten, Feiertagen...
- Kosten
- Umrechnungen, z.B. realen Gesamtaufwand aus abstraktem in konkretes Maß

## 5

# Weitere Zielgrößen ableiten (begleitende Beispiele)

# Bis hier

- **Weiterentwicklung des UIs des firmeneigenen CMS**

- Weitere Zielgröße: Ungefährre Mindestdauer des Projekts
- Team-Geschwindigkeit: 20 SP / Iteration (3 Wochen)
- Theoretische Gesamtdauer des Projekts: ca. 2,5 Jahre

- **Grenzen der Optimierung:** Auch wenn sich neun Frauen dazu bereit fänden,  
**Ein Festpreisprojekt** werden sie kein Baby in nur einem Monat zur Welt bringen können!

- Weitere Zielgrößen: Rechnerische Team-Größe, Personalkosten

- Aufwand und Zieltermin => Rechnerische Team-Größe
- Realer Gesamtaufwand, Team und Tagessätze => Personalkosten

# Lernschleifen

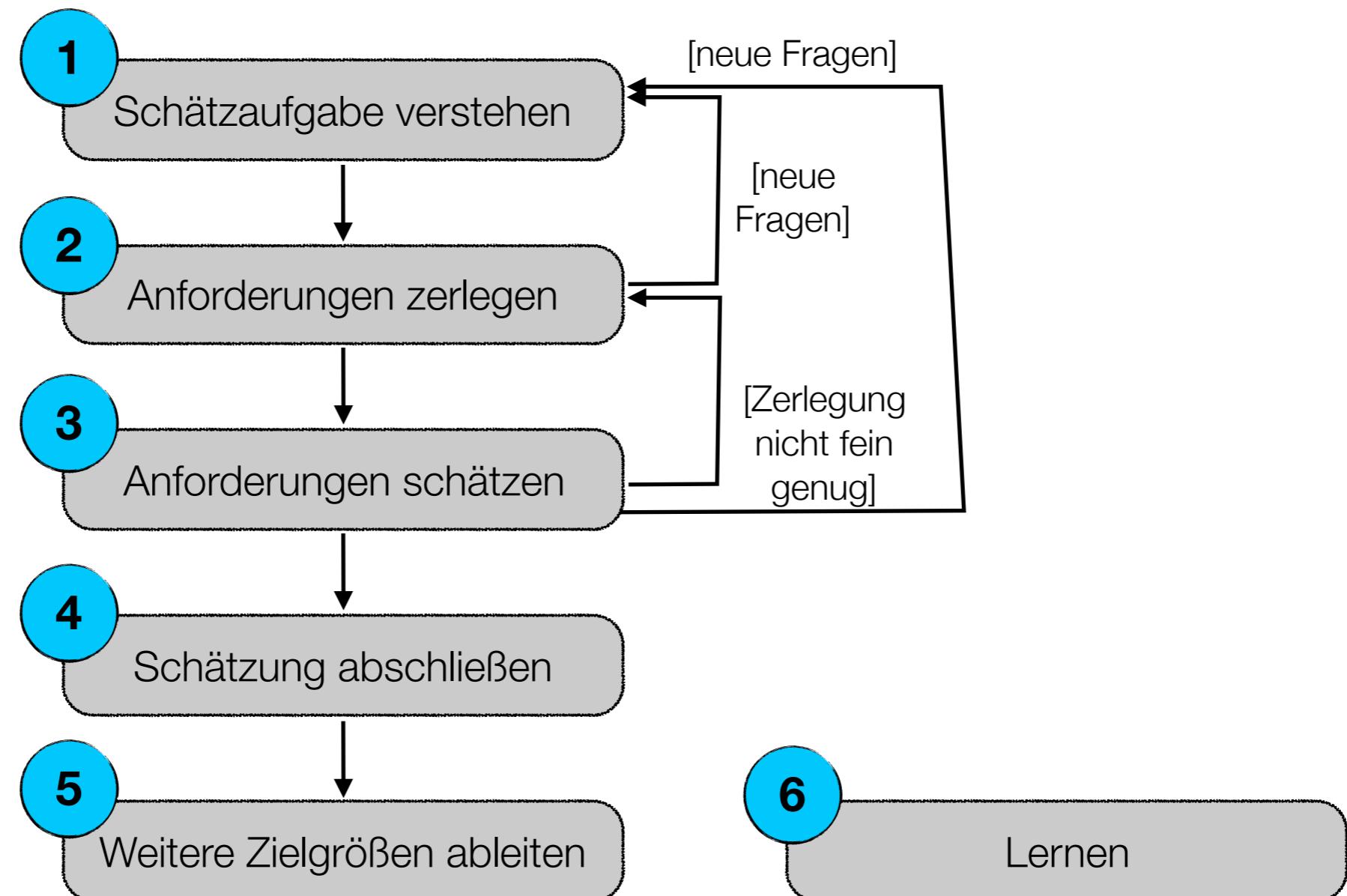
---

- Lernziele
  - Zutreffendere Analogiebildung
  - Prozess und Korrekturfaktoren auf Organisation kalibrieren
- Ansatz: Measure, inspect, adapt(, and document)!
- Einige Beispiele
  - Während des Projekts: Project Controlling
  - Zum Sprint-Ende (Iterationsende): Reviews und Retrospektiven
  - Nach Projektende: Post Mortems

# Begleitende Beispiele - Lernschleife

- **Weiterentwicklung des UIs des firmeneigenen CMS**
  - Iterations-Reviews, sog. Burndown-Charts: Messen Team-Geschwindigkeit
  - Kalibrierung des abstrakten Schätzmaßes: Maus, Schaf, Elefant
- **Ein Festpreisprojekt**
  - Wöchentliches Projekt-Controlling:
    - Ist-Soll-Abgleich und Effort to Completion
    - Aktive Entwicklungszeit
  - Projektnachkalkulation: Adjustierung des pauschalen Projektaufschlags

# Zur Erinnerung...

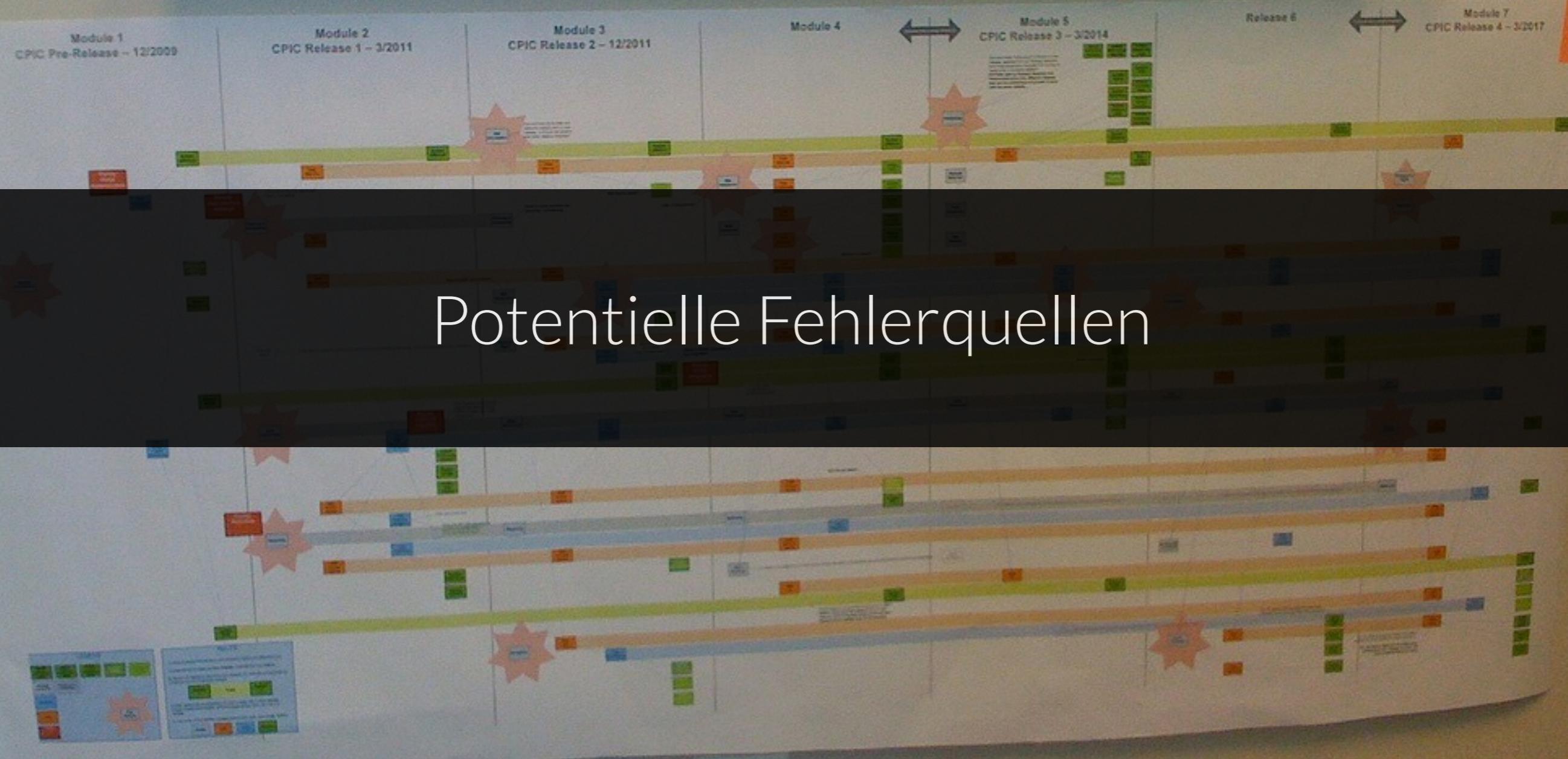


# Agiles vs. Klassisches / Phasenorientiertes PM

---

- Lediglich Zeitpunkt, Schätzmaße und Zuständigkeiten unterscheiden sich, die Tätigkeit, bzw. das Vorgehen beim Schätzen an sich bleibt gleich!

# Potentielle Fehlerquellen



# Fehlerquellen beim Schätzen, da gibt's so einige...

---

- Schätzmethode oder Tool-Unterstützung nutzt nur bei vorhandener Erfahrung
- Bildung nicht zulässiger Analogien, z.B. fehlendes Domänenwissen
- Zu feine Zerlegung der Anforderungen, z.B. „Lost in Recursion“
- Zu grobe Korrekturfaktoren, z.B. „Keep your ass covered“
- Das Lernen auslassen, d.h. der Schätzprozess bleibt immer ad hoc, „blind“
- ...und viele, viele mehr

# Fehlerquelle

## Pseudoschätzverfahr

- Schätzungen werden gerne „politisch“ oder „kommerziell“ genutzt und angepasst.
- „Design to cost“
- „Price to win“



# Zusammenfassung

---

- Es gibt keine Silberkugel - weder Tools noch eigene Methoden
  - Ein reproduzierbares Vorgehen ist Basis für Optimierungen
  - Grundprinzip der Aufwandsschätzung über Methoden & Paradigmen stets gleich
    - Schätzaufgabe verstehen und in Anforderungen zerlegen  
Black Arts: Aufwandsschätzen ist der Blick in die Kristallkugel...
    - Anforderungen schätzen, den großen Abweichungen nachgehen und klären!
  - Schätzung abschließen und weitere Zielgrößen ableiten
  - Messen
- 
- Erfahrung kommt durch: Übung, Ausprobieren und Wiederholung

# Ausblick & Fragen

---

- Heute: Planung
  - Schätzen
- Nächstes Mal
  - Anordnen
  - Ableitungen



THANK YOU!

# Links & Literature

---

- [Albrecht] A. J. Albrecht, "Measuring Application Development Productivity," Proceedings of the Joint SHARE, GUIDE, and IBM Application Development Symposium, Monterey, California, October 14–17, IBM Corporation (1979), pp. 83–92.
- [Boehm] Barry Boehm et.al., „Software Cost Estimation with CoCoMo II“, Prentice Hall, 2000
- [Boehm 81] Barry Boehm, „Software Engineering Economics“, Prentice Hall, 1981
- [McConnell] Steve McConnell, „Taming Wild Software Schedules“, Microsoft Press, 1996
- [CoCoMo II] sunset.usc.edu/csse/research/COCOMOII/cocomo\_main.html
- [CoCoMo II Webtool] sunset.usc.edu/research/COCOMOII/expert\_cocomo/expert\_cocomo2000.html
- [Death March] Ed Yourdan, „Death March (2nd Edition)“, Prentice Hall, 2003
- [FPOT] ivs.cs.uni-magdeburg.de/sw-eng/us/java/fp/
- {Parkinson} Cyril Northcote Parkinson, „Parkinson's Law or The Pursuit of Progress“, John Murray 1958
- [PMBoK] „A Guide to the Project Management Body of Knowledge: PMBoK Guide“, Fourth Edition, PMI, 2008
- [PM-Glossar] www.projektmagazin.de/glossar/
- [Scrum] Ken Schwaber, „Agile Project Management with Scrum“, Microsoft Press, 2004
- [XP] Kent Beck, Cynthia Andres, „eXtreme Programming Explained: Embrace Change“, 2nd Edition, Addison-Wesley Longmann, 2004

# Bildnachweis

---

- „A380 over Sidney Harvour“ by Tilli Dog Faustografix, Flickr



- „Denver Internation Airport. Denver, CO“ by gamp, Flickr



- Tool Collect, from Toll Collect press material

- „Peligro indefinido - undefined danger“ by Antonio Martinez, Flickr



- „Window Cleaners“ by Mallix, Flickr



- Screenshot 3 Time Point Estimation, by Jörg Pechau

- „Now that's a paln by James Melzer, Flickr



- Screenshot Flipchart, by Jörg Pechau

- „?“ by florianmarquardt, Flickr

- Screenshot Function Point Analysis Webtool by Jörg Pechau

- „Red Phone box and Pillar box SE18 35“ by kenjonbro



- Screenshot CoCoMo II Webtool by Jörg Pechau



- Alle nicht markierten/genannten Grafiken von Jörg Pechau