



HOCHSCHULE RUHR WEST
UNIVERSITY OF APPLIED SCIENCES

INSTITUT INFORMATIK

IT Projektmanagement: Projektinitiierung

Prof. Dr. Jan M. Pawlowski

Licensing: Creative Commons



You are free:

to Share — to copy, distribute and transmit the work



to Remix — to adapt the work

Under the following conditions:



Attribution. You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).



Noncommercial. You may not use this work for commercial purposes.



Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Was bisher geschah ...

...erzählen Sie mal ...

Heute:

- **Projektentstehung**
 - Phasen
 - Ergebnisse

- **Projektskizze**
- **Business Case**

• **Lastenheft**

• **Aufwandsschätzung**

CALENDAR

SUN	WED	THU	FRI	SAT		
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Projektphasen (Prozesse und Sub-Prozesse)

- **Entstehung:** Anforderungsanalyse, Lösungsauswahl, Projektklassifizierung, Projektbeantragung
- **Definition:** Projektbeauftragung, Erstellung Gesamtprojektplan, Festlegung Projektorganisationsform, Kick-off-Veranstaltung, Projektstartsitzung
- **Planung:** Planungsarten, Planungsinstrumente, Planungszuständigkeit, Planungszeitpunkt, Planungsentscheide
- **Vorgehen:** inkrementelle, konzeptionelle, empirische und evaluative Vorgehensmodelle insbesondere für Multiprojekte
- **Kontrolle:** Kontrollzeitpunkt, Kontrollsichten, Kontrollverfahren, Kontrollprozess, Kontrollberichte
- **Abschluss:** Projektabnahme, Projektabschlussbeurteilung, Projektabschlussbericht, Erfahrungssicherung, Einführungsnachbearbeitung, Projektauflösung

(Wieczorrek & Mertens, 2011)

Projektentstehung

- Phase der Entwicklung eines Projektes
- Projekt ist noch nicht genehmigt und muss daher vorbereitet werden
- „Vorlauf“

• Inhalte

- **Feststellung und Analyse der Ziele**, des Bedarfs und der groben Anforderungen
- **Analyse der Risiken** und Erfolgsaussichten (Chancen)
- **Abschätzung** der Durchführbarkeit
- **Kosten-Nutzen-Analyse**: Schätzung des Aufwands/der Kosten in Relation zum Nutzen
- **Ermittlung** des Nutzens und Erstellung des Business Case

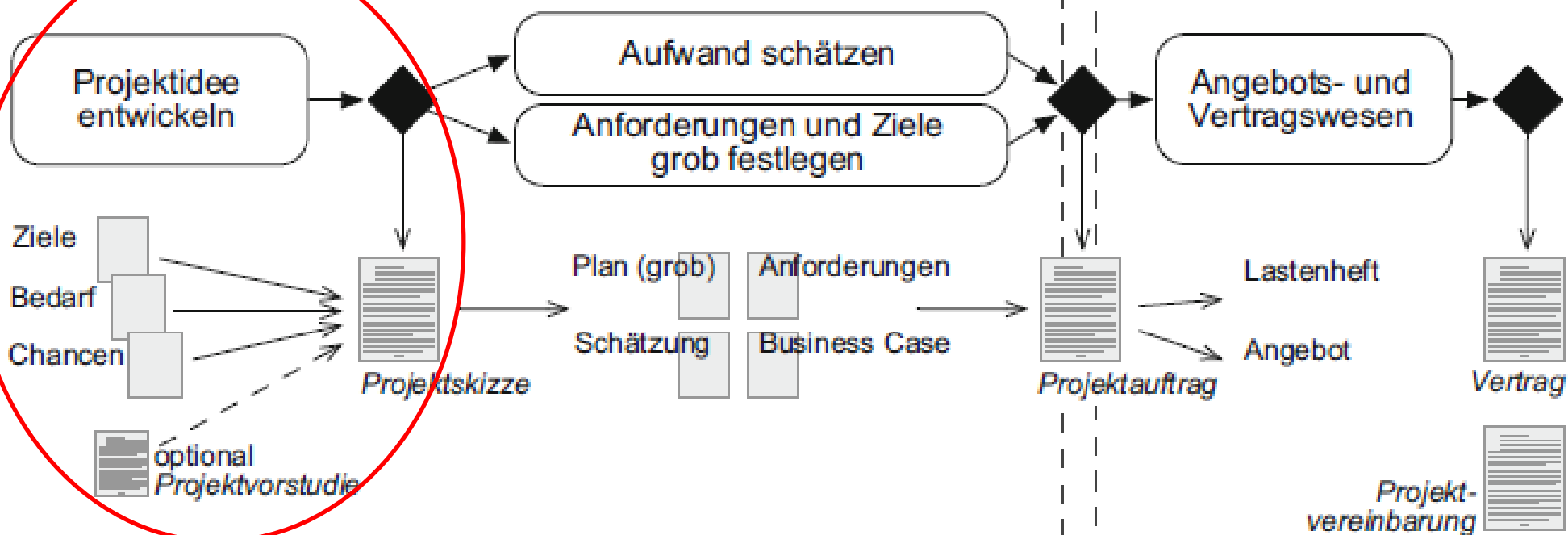
Projektentstehung

Projektorganisation und Management

Projektentstehung

Ideenfindung und Projektdefinition

Projektvereinbarung



(Broy & Kuhmann, 2013)

1. Schritt: Projektidee

- Grundsätzliche Beschreibung des Projektes
 - Überblick über die Projektidee
 - **Situationsbeschreibung:** Welche Problemstellung liegt vor?
 - **Begründungen *intern*:** Schwachstellen / Probleme in Unternehmensabläufen, **Verbesserungsmöglichkeiten**
 - **Begründungen *extern*:** Innovation, Marktpotenziale
- Ziele
 - Was sind die Ziele des Projektes
 - Wie stehen die Ziele in Beziehung zur Organisationsstrategie
- Bedarf
 - Beschreibung der **Zielgruppe**
 - Beschreibung des **Bedarfes** der Stakeholder (z.B. Kunden, Mitarbeiter, Partner)
- Chancen
 - Wie sind die **Marktpotenziale**
 - **Konkurrenten** oder **Partner?**
- Projektvorstudie
 - *Nur bei umfangreichen Projekten*
 - **Marktanalyse**, Kosten-Nutzen-Rechnung
 - Ähnlich der Phase der Projektdefinition...

2. Schritt: Business Case

- **Definition Business Case (PRINCE2)**
 - „Ein Business Case enthält die optimale Zusammenstellung fundierter Informationen für die Beurteilung, ob ein Projekt wünschenswert, lohnend und realisierbar ist (und bleibt) und letztlich eine sinnvolle Investition darstellt.“
- **Technische Machbarkeit**
 - Technische Umsetzungsfähigkeit
 - Ressourcen zur Umsetzung
- **Wirtschaftliche Machbarkeit**
 - Kosten-/Nutzenverhältnis / Return on Investment
 - Finanzierbarkeit
 - Kundennutzen und Akzeptanz im Markt
- **Organisatorische Machbarkeit**
 - Zeitliche Verfügbarkeit der benötigten Ressourcen
 - Erfüllung der Rahmenbedingungen

Beschreibung des Business Case (PRINCE2)

- **Zusammenfassung**
- **Gründe:** Warum ist das Projekt notwendig?
- **Optionen** (alternative Projekte): Ist dieses Projekt lohnender/besser als andere Optionen?
- Erwarteter **Nutzen:** Was ist der Nutzen dieses Projekts?
- Erwartete negative **Nebeneffekte:** Welche unerwünschten Konsequenzen sind abzusehen?
- **Zeitraumen:** Wie lange wird das Projekt dauern?
- **Kosten:** Welche Kosten wird das Projekt verursachen?
- **Investitionsrechnung:** Wann und wie wird sich die Investition auszahlen?
- **Hauptrisiken:** Was kann schief gehen?

Diskussion #1

- Wie würden Sie Ihr Projekt zusammenfassen?
- Was sind Alternativen zu Ihrer App?

- Was sind die Inhalte einer **Projektskizze**, um eine Projektidee zu beschreiben?
- Was sind die Inhalte eines **Business Cases**?

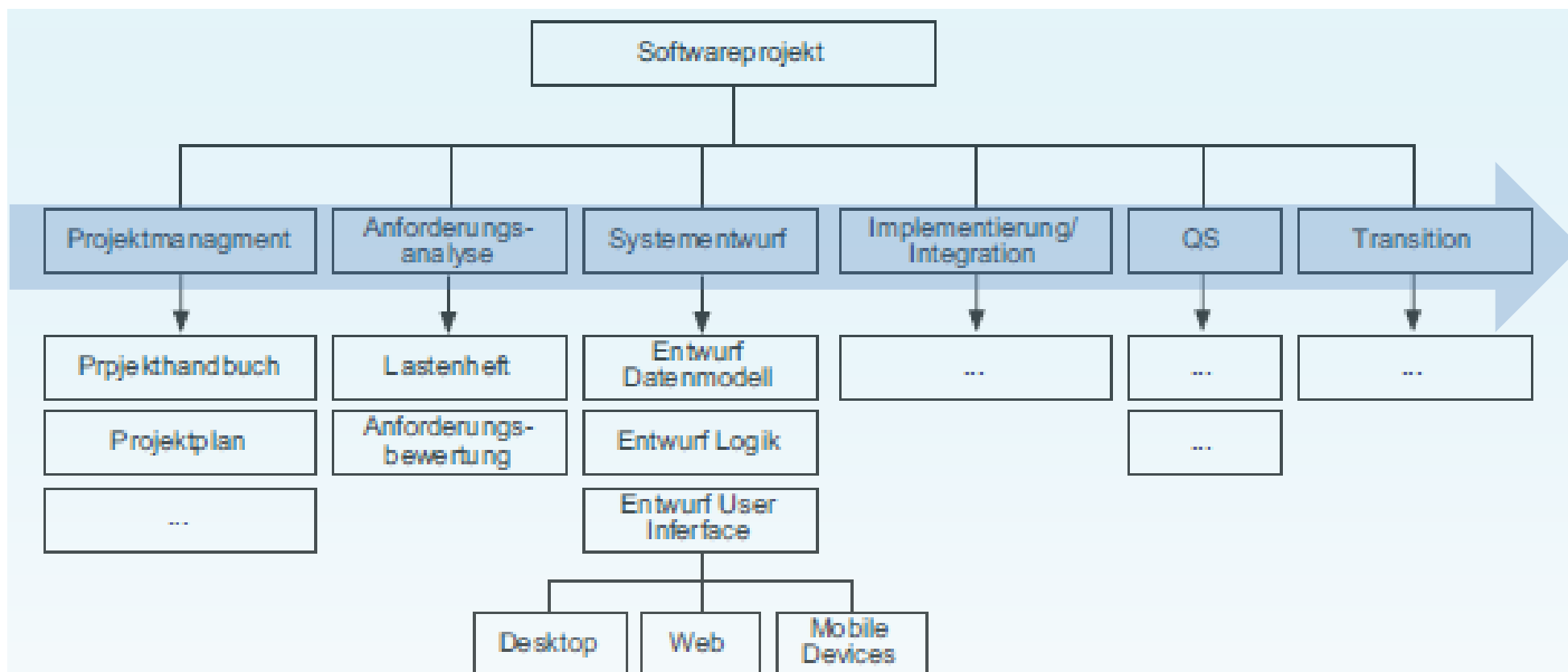
LEITFRAGEN #1

Ich bin bis hier gut mitgekommen ...



Strukturierung von Projekten

- **Projektstrukturplan:** Strukturierung der Aufgaben und Rollen
- **Projektablaufplan:** Zeitliche und sachlogische Abhängigkeiten der Aktivitäten



(Broy & Kuhrmann, 2013)

Strukturierung von Projekten: Vorgangslisten

- Erstellung einer Vorgangsliste
 - Aktivität
 - Abhängigkeit
 - Aufwand in Personentagen
- Beispielhafte Vorgänge in den Projektphasen
 - **Initiierung:** Zusammenstellung von Teams (Erfassung der Kompetenzanforderungen, Suche von Mitarbeitern, Teambuilding, Teammeetings), Grobplanung, Business Case, Risikoidentifikation,
 - **Planung:** Netzplan, Qualitätsmanagement, Finanzplan
 - **Durchführung:** Initiierung, Anforderungsermittlung, Design, Prototyping, Coding, Testing, Einführung, Abnahme (mehrere Zyklen nach dem Agile Model), Controlling, Monitoring, Risikoüberwachung, Berichtswesen
 - **Projektabschluss:** Abnahme, Projektabschlussbericht, Übernahme in Regelbetrieb, Wartung, Verbesserungsmaßnahmen

(Broy & Kuhrmann, 2013)

Strukturierung von Projekten: Vorgangslisten

Nr.	Vorgang	Abhängig von	Aufwand
1	Projektskizze anfertigen	-	3 PT
2	Lastenheft anfertigen	1	2 PT
3	Projektauftrag genehmigen	2	4 PT
4	Entwicklung Komponente A	3	5 PT
5	Entwicklung Komponente B	3	6PT
6	Prototyp Version 0.1	4, 5	3 PT

Diskussion #2

- Welche Aktivitäten können Sie aus dem Modell SCRUM oder V Modell XT ableiten?

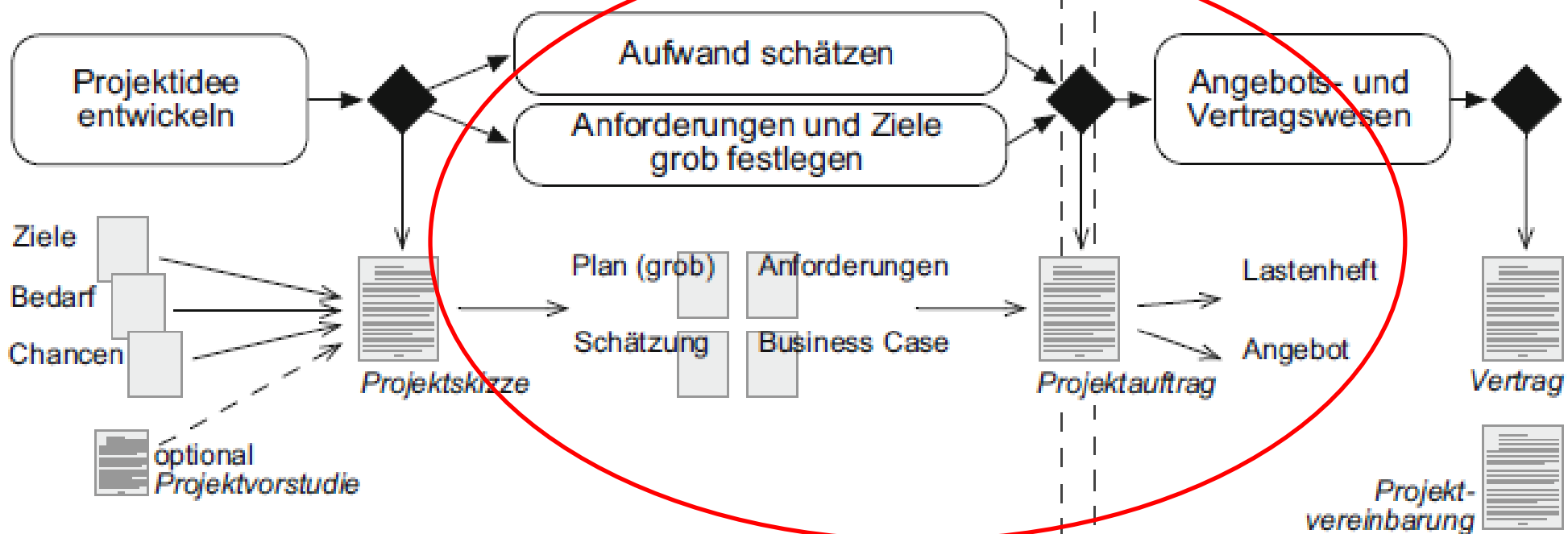
Projektentstehung

Projektorganisation und Management

Projektentstehung

Ideenfindung und Projektdefinition

Projektvereinbarung



(Broy & Kuhmann, 2013)

4. Schritt: Lastenheft

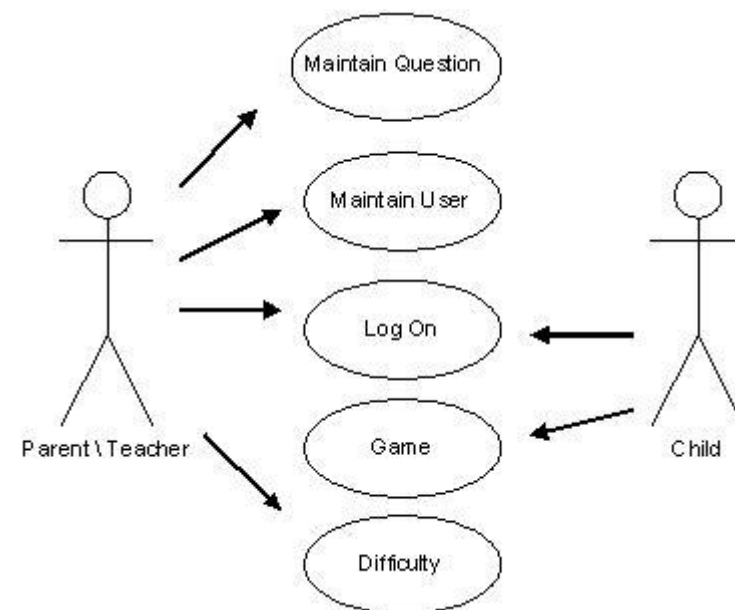
- Definition: „vom Auftraggeber festgelegte Gesamtheit der Forderungen an die Lieferungen und Leistungen eines Auftragnehmers innerhalb eines Auftrages“ (DIN 69901)
- Charakteristika
 - Beschreibt die nachprüfbaren Systemanforderungen aus Auftraggebersicht
 - Quantifizierbare und prüfbare Anforderungen aus Anwendersicht
 - Dient als Ausschreibungs-, Angebots- und Vertragsgrundlage
 - Grundlage für Pflichtenheft und erstes Angebot
- Prinzipiell sollte ein Dokument vorliegen, das Anforderungen umfasst und von Auftraggeber und –nehmer im Konsens abgestimmt wird
- PRINCE2 umfasst kein klassisches Lastenheft sondern beschreibt Anforderungen und verschiedene Planungselemente

Lastenheft: Exemplarische Inhalte

- Einleitung (Motivation, Hintergrund, Historie, etc.)
- Produktkontext (Zweck des Produkts, Stakeholders)
 - Größtenteils aus Projektskizze / Business Case abzuleiten
- Relevante Standards, Vorschriften, Geheimhaltungsaspekte, Referenzdokumente
- Systemkurzbeschreibung (mit Einbettung in seine unmittelbare Umgebung)
- Funktionale Anforderungen
 - Ablauf- und Verhaltensbeschreibung
 - Datenbeschreibung
 - Benutzeroberfläche
 - Rechte
 - Mehrsprachigkeit
- Nicht-funktionale Anforderungen
 - Qualität
 - Technische Anforderungen
- Projektorganisation
 - Verantwortlichkeiten
 - Kosten
 - Phasen und Leilensteine
- Abnahmekriterien

Funktionale Anforderungen

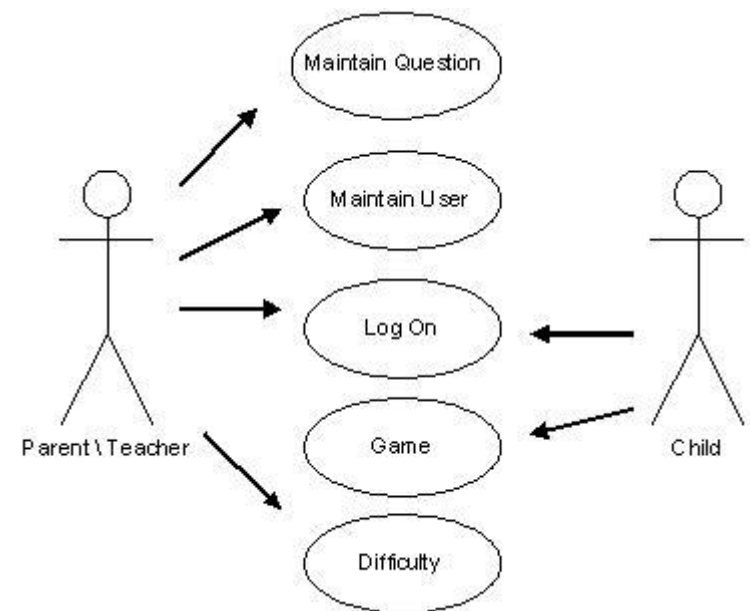
- **Was soll das System leisten? Welche Dienste soll es anbieten?**
- Eingaben und deren Einschränkungen (Daten, Ereignisse, Stimuli, ...)
- Bereitgestellte Dienste, „Funktionen“, die das System ausführen können soll
 - Transformation von Daten („funktionales Verhalten“)
 - Verhaltensweisen abhängig von Ereignissen/Stimuli („reaktives Verhalten“)
 - Ausgabe (Daten, Fehlermeldungen, Reaktionen, ...)
 - Relevante Systemzustände („Betriebsmodi“)
 - Zeitliches Verhalten des Systems
- Nutzung von Anwendungsfällen (Use Cases)



(Quelle: commons.wikimedia.org)

Funktionale Anforderungen: Anwendungsfälle

- Beschreibung Anwendungsfall
- Name
- Kurzbeschreibung
- Akteure
- Auslöser
- Eingehende Daten
- Vorbedingungen
- Nachbedingungen, Ergebnis
- Essentielle Schritte
- Offene Punkte
- Sonstiges, Anmerkungen



(Quelle: commons.wikimedia.org)

Funktionale Anforderungen: Beispiele

- Aktionen, die vom System ausgeführt werden sollen
- Bsp.: Das System muss Vorlesungen in die DB aufnehmen können.
- Systeminteraktionen, die dem Nutzer ermöglicht werden
- Bsp.: Das System muss es dem Administrator beim Eintragen einer Vorlesung ermöglichen, Raum, Zeit und Titel einzugeben.
- Allgemeine funktionale Vereinbarungen u. Einschränkungen
- Bsp.: Der Client ist für den Kommunikationsaufbau zuständig.

Funktionale Anforderungen: Anwendungsfälle

Beschreibung Anwendungsfall				
Name	Seminar registrieren			
Kurzbeschreibung	Erfassen eines neuen Seminars und registrieren in der Seminarverwaltung			
Akteure	Dozent, Sachbearbeiter			
Auslöser	Ein neues Seminar soll abgehalten werden			
Eingehende Daten	Seminarnamen, Beschreibung, Inhalt, Umfang, Kosten			
Vorbedingungen	Account existiert			
Nachbedingungen, Ergebnis(se)	Das neue Seminar wurde erfasst und abgespeichert			
Essentielle Schritte	<ol style="list-style-type: none"> 1.) <<include>>: Berechtigungen prüfen 2.) Seminarerfassung 3.) Seminarerfassung prüfen 4.) Seminarerfassung in Datenbank eintragen 			
Offene Punkte				
Änderungshistorie	Wann	Wer	Neuer Status	Was
Sonstiges, Anmerkungen				

(Quelle Hafenrichter, 2012)

Nicht-funktionale Anforderungen

- Wie soll das System/individuelle Funktionen arbeiten?
- Qualität
 - Leistungsfähigkeit / Performanz
 - Benutzerfreundlichkeit: Gestaltung der Benutzeroberfläche
 - Zuverlässigkeit: Verhalten bei x Nutzern / Anfragen
 - Sicherheit: Zugang, Datenschutz, Privatsphäre
 - Standards: Einhaltung von Branchen-Standards
- Entwicklungsanforderungen
 - Konformitäten
 - Änderbarkeit
 - Übertragbarkeit
 - Installation
 - Wartbarkeit
 - ...
- Technische Anforderungen
 - Entwicklungs- und Laufzeitumgebung
 - Betriebssystem
 - Architektur
 - Datenbanken
 - Web vs App
 - ...

Nicht-funktionale Anforderungen: Beispiele

- Technische Anforderungen:
 - Das System muss mit Java entwickelt werden und muss in der Sun Java VM 1.5 laufen.
- Ergonomische Anforderungen
 - Das System muss die gespeicherten Objekte formatiert ausgeben können (Formatvorgabe).
 - Die Benutzerführung erfolgt in deutsch.
- Anforderungen an die Dienstqualität
 - Das System muss jede Anfrage des Benutzers innerhalb von 30 Sekunden ausführen (auf System XY).
 - Der Speicherbedarf darf 512MB nicht übersteigen.
- Zuverlässigkeit
 - Die Verfügbarkeit des Systems muss bei 99.999 % liegen.
- Anforderungen an den Entwicklungsprozess
 - Der Entwickler muss mit dem Kunden monatliche Reviews der zu erstellenden Dokumente durchführen.
- Rechtlich-vertragliche Anforderungen
 - Der Kunde leistet für jeden abgenommenen Meilenstein ein Drittel der vertraglich vereinbarten Summe für die Entwicklung des Systems.
 - Die deutsche Datenschutzrichtlinie muss erfüllt sein.

Nicht-funktionale Anforderungen: Dokumentation

Identifikation	Eindeutige Identifikation einer Anforderung (z.B. ID0001)
Name	
Kategorie	z.B. Qualität, Technische Anforderung, ...
Beschreibung	Beschreibt in komprimierter Form die Anforderung. Achten Sie darauf, dass die Anforderung messbar formuliert ist.
Autor	
Priorität	Verbindlich – Anforderung muss umgesetzt werden Gewünscht – Anforderung ist wichtig, kann aber evtl. verschoben werden Optional – Anforderung kann auch in späterem Release umgesetzt werden
Testkriterium	Messbare Testanweisung welche die Überprüfung der Anforderung erlaubt

Anforderungen: Methoden

- Interviews
 - Focus Group
 - Fragebögen
 - Dokumentenanalyse
 - Beobachtungen
 - Prototyping
-
- Weitere Diskussion in den nächsten Vorlesungen....

5. Schritt: Projektauftrag

- Definition
 - Der Projektauftrag stellt eine Fixierung dafür dar, dass das Projekt durchgeführt wird, enthält Ziele, Mittelfreigaben und Auflagen, und schafft die Grundlage für die Nutzung von Ressourcen aber auch für eine Limitierung und Überwachung der Kosten. Der Projektauftrag wird für das Gesamtprojekt, aber auch für einzelne Teilprojekte (Ausbaustufen, Lose) schriftlich erteilt
- Aspekte
 - Kurze Beschreibung des durchzuführenden Projekts
 - Einsetzbare und freigegebene Betriebsmittel (Budgetvorgaben): Personal, Rechner, weitere Kosten
 - Zeitlicher Umsetzungsplan, einschließlich Fertigstellungstermine, evtl. weitere Rahmenbedingungen, wie etwa Vorgaben zum projektspezifischen Vorgehen
 - Weitere Vorgaben: Vorgaben zu Aufgaben-, Konfigurations- oder Risikomanagement, zum Problem und Änderungsmanagement und zum Berichtswesen
 - Verantwortliche / Zuständigkeiten und Genehmigungen

- Was sind die Inhalte eines Lastenheftes? Wer erstellt den Großteil des Lastenheftes?
- Nennen Sie Kategorien funktionaler Anforderungen; geben Sie jeweils Beispiele an.
- Nennen Sie Kategorien nicht-funktionaler Anforderungen; geben Sie jeweils Beispiele an.
- Was sind die Inhalte des Projektauftrages?

LEITFRAGEN #2

Aufwandsschätzung

- Wichtige Aktivität auf dem Weg zur **Projektgenehmigung**
- **Beschreibung** von Projektaktivitäten
- **Vorgangslisten / Projektstrukturplan**
- **Abschätzung**

Aufwandsschätzung: Arbeitspakete

- Arbeitspakete als kleinste Einheit des Projektstrukturplans
- Bestehen aus mehreren, zusammengehörigen Vorgängen (Vorgangsliste)
- Umfang
 - Tätigkeiten einer Einzelperson
 - Fachlich zusammengehörige Arbeiten, die von einer Gruppe/Person durchgeführt werden sollen
 - Entwicklungsauftrag an eine Fremdfirma
- Kriterien für Arbeitspakete
 - Konkretes Ergebnis
 - Kalkulierbarkeit (Aufwand, Termin, Kosten)
 - Keine Überschneidung mit anderen Paketen
 - Klare Schnittstellen
 - Eindeutige Verantwortlichkeit

Aufwandsschätzung: Arbeitspakete

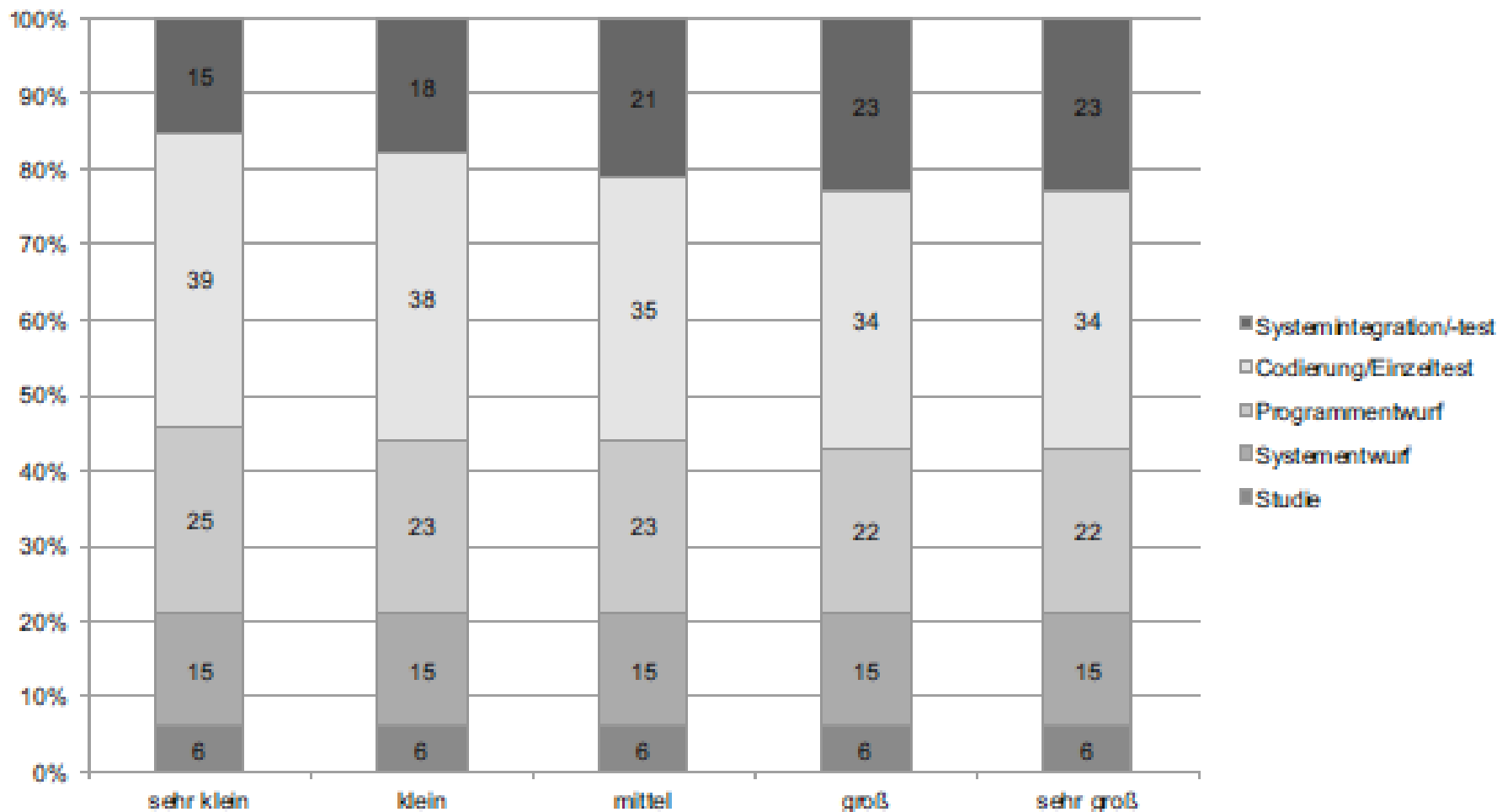
Vorgehen

- Informationen werden durch Erfahrungen der Vergangenheit bewertet und in Arbeitsaufwände umgerechnet
- Gute Wissensbasis erforderlich
 - Genauer Umfang der Arbeitspakete
 - Genaue Ergebnisse der Arbeitspakete
 - Erforderliche Schritte zur Durchführung
- Besprechung der Arbeitspakete mit den Verantwortlichen
- Ermittlung der geschätzten Arbeit in Personentagen (PT) oder in Personenstunden (Ph)
- Mitarbeiterkosten = Personentage x Tagessatz
- Mitarbeiterkosten = Personenstunden x Stundensatz
- Dauer (in Tagen) = Aufwand (in h) / (Einsatzeinheiten x h pro Tag)

(Quelle Leitner, 2009)

Aufwandsschätzung: Arten von Projekten / Phasen

- Aufwandsverteilung nach Projektphase
- Heute je nach Ansatz oft stark variierend



(Quelle Boehm, 1981)

Aufwandsschätzung: Stücklisten

Vorgehen

- Bewertung einzelner Komponenten und / oder Funktionen

Nr.	Bezeichnung	Komplexität	Aufwand (in PM)
1.	Komponente A	einfach	15
2.	Komponente B	einfach	5
3.	Funktion X	mittel	10
4.	Anwenderdokumentation	einfach	25
...
25.	Komponente Y	schwer	25
	Summe:		237

(Quelle Broy, 2013)

Aufwandsschätzung: Aufschläge (Erfahrungswerte)

Vorgehen

- Aufschläge aufgrund ausgewählter Aspekte

Posten	Aufschlag
Projektleitung	
< 5 Mitarbeiter	10 %
6–15 Mitarbeiter	15 %
> 15 Mitarbeiter	20–25 %
Chefarchitekt	10–20 % (zum Teil mehr)
Infrastruktur, Systemverwaltung, Werkzeuge	5 %
Einarbeitungszeit	ca. 4 Wochen pro Mitarbeiter (enthält ggf. Qualifikationsmaßnahmen)
Leerzeiten (Reisen etc.)	

(Quelle Broy, 2013)

Aufwandsschätzung: Aufschläge (Erfahrungswerte)

Vorgehen

- Aufschläge aufgrund ausgewählter Aspekte (→ vgl. S. 54)

Einflussfaktor	Aufwandsverhältnis
schwierige/einfache Benutzerschnittstelle	4 : 2
mit/ohne Benutzermitwirkung	2 : 2
mit/ohne Leistungsforderungen	2 : 2
mit/ohne Entwicklungsmethode	1 : 2
höhere/niedrigere Qualifikation	1 : 3
mit/ohne parallele Geräteentwicklung	2 : 1
mit/ohne CASE-Tools	1 : 7

(Quelle Broy, 2013)

3 Zeiten Methode

- optimistische Schätzung (best case) bc
- pessimistische Schätzung (worst case) wc
- wahrscheinliche Schätzung (most likely case) lc

AP	Best Case	Worst Case	Likely Case	A	s
1	5	15	10	10	1,67
2	8	22	16	15,67	2,33
3	2	9	6	5,83	1,17

$$A_i = \frac{bc_i + 4 \cdot lc_i + wc_i}{6}$$

(Quelle Broy, 2013)

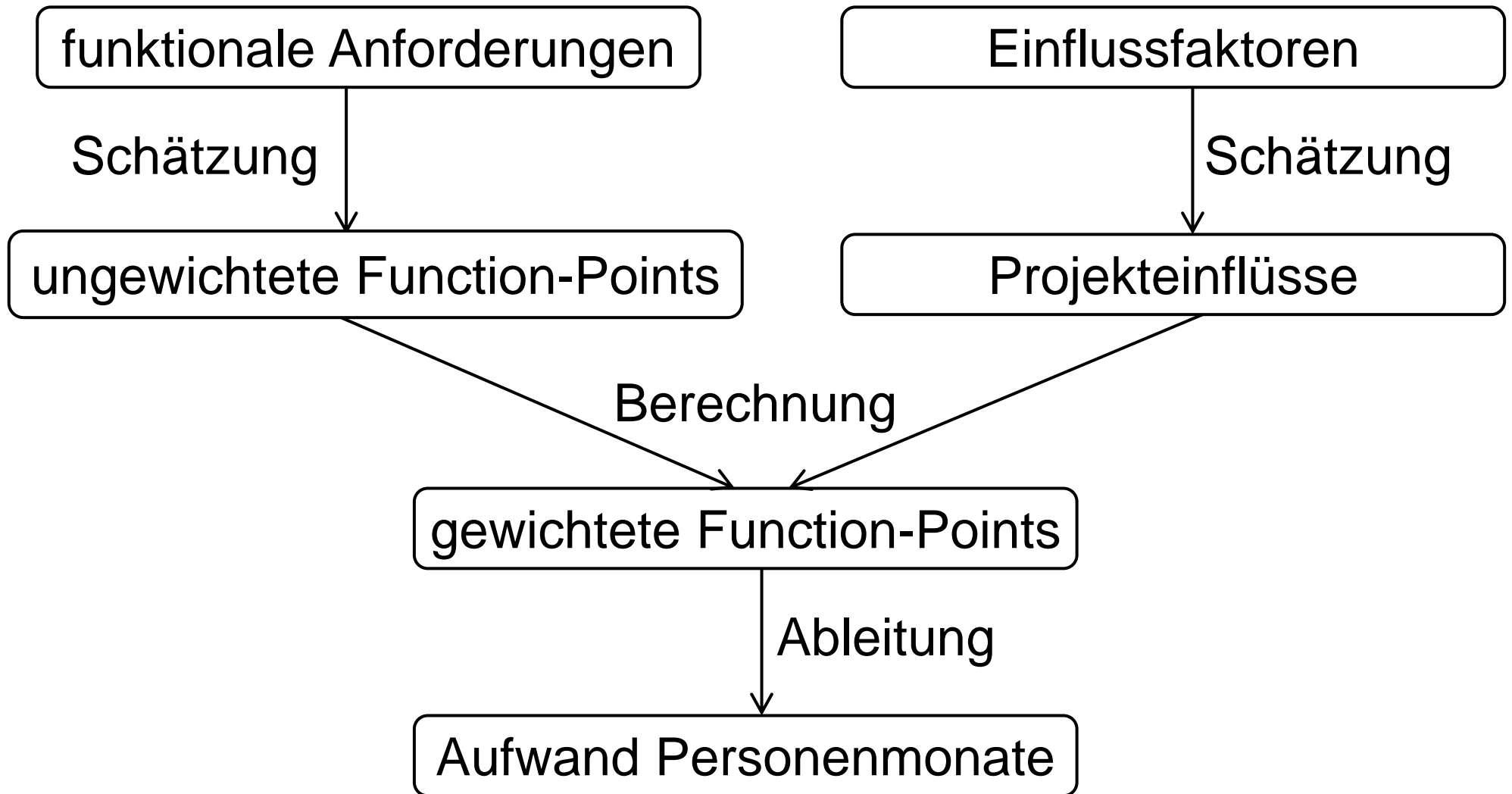
Aufwandsschätzung: Arbeitspakete

Schätzmethoden

- Delphi-Methode
 - Moderator & mehrere Experten
 - Verfahren
 - Zu jedem AP gibt jeder Experte einen Schätzwert ab
 - Mittelwert, falls alle Schätzwerte in einer bestimmten Bandbreite (Mittelwert + 20%)
 - Sonst: Argumente austauschen und neue Schätzung
 - Zuschläge zu den Einzelschätzungen:
 - Plus 10-15 % = Aufwand für Projektmanagement
- 3-Experten-Konzept (oder Mini-Experten-Team)
 - 3 optimistische, 3 realistische und 3 pessimistische Schätzungen
 - Abklärungen durchführen, bis die Ergebnisse in jeder Kategorie übereinstimmen
 - Daraus Schätzwert gemeinsam ableiten

(Quelle Leitner, 2009)

Zusammenfassung der Function Point Analyse



(Quelle Kleuker, 2013)

Heuristische Schätzverfahren

- Auf Basis von Erfahrungen
- Modifikation von Komponenten-Schätzungen aufgrund von Kostentreibern
 - Ausbildungsstand
 - Komplexität
 - ...
- Methoden
 - Function Point Analysis
 - CoCoMo 2
 - ...

Function-Point-Methode (Balzert 1998)

Jede Anforderung (bzw. ermittelte Funktionalität) in eine der fünf Kategorien einordnen

- **Internal Logical Files (ILF, Datenbestand):** Datenbestände, die innerhalb des zu entwickelnden Systems bearbeitet werden, z. B. die selbstentwickelten Klassen
- **External Interface Files (EIF, Referenzdaten):** Datenbestände die von außerhalb des zu entwickelnden Systems kommen und bearbeitet werden
- **External Inputs (EI, Eingabe):** Eingaben von außerhalb in das zu entwickelnde System mit denen der Datenbestand verändert wird, z. B. über Eingabemasken, Dateien in bestimmten Eingabeformaten, Eingaben von externen Systemen
- **External Outputs (EO, Ausgabe):** Ausgaben an außerhalb des zu entwickelnden System liegende Komponente, z. B. Ausgabemasken, Dateien in bestimmten Ausgabeformaten; Ausgaben werden aus vorhandenen Daten berechnet
- **External Inquiries (EQ, Abfrage):** Abfragen von Informationen des Datenbestands von außerhalb des Systems, z. B. Abfragemasken, reines Anzeigen der verwalteten Daten

Function-Point-Methode

Kategorie	Anzahl	Klassifizierung	Gewichtung	Zeilensumme
Eingabedaten	6	einfach	* 3	18
	4	mittel	* 4	16
		komplex	* 6	
Abfragen	4	einfach	* 3	12
		mittel	* 4	
		komplex	* 6	
Ausgaben	6	einfach	* 4	24
	9	mittel	* 5	45
		komplex	* 7	
Datenbestände	7	einfach	* 7	49
		mittel	* 10	
		komplex	* 15	
Referenzdaten		einfach	* 5	
		mittel	* 7	
		komplex	* 10	
Summe ungewichteten Function-Points				164

(Quelle Kleuker, 2013)

Function-Point-Methode

Summe unbewertete Function-Points		164
	Verflechtung mit anderen Anwendungssystemen (0-5)	1
	Dezentrale Daten, dezentrale Verarbeitung (0-5)	2
	Transaktionsrate (0-5)	2
	Verarbeitungslogik	
	Rechenoperationen (0-10)	1
	Kontrollverfahren (0-5)	3
	Ausnahmeregelungen (0-10)	0
	Logik (0-5)	2
	Wiederverwendbarkeit (0-5)	3
	Datenbestandskonvertierungen (0-5)	2
	Anpassbarkeit (0-5)	1
Summe der Einflussfaktoren (EF)		17
Faktor Einflussbewertung (FE) =		$EF / 100 + 0,7$
Gewichtete Function-Points = ungewichtete Functionpoints *FE		143

(Quelle Kleuker, 2013)

Aufwand nach Jones (1996)

- A. Durchlaufzeit [in Monaten] = $FP^{0.4}$
- B. Anzahl Mitarbeiter = $FP / 150$
- C. Aufwand = Durchlaufzeit x Anzahl Mitarbeiter = $FP^{0.4} \times FP / 150$
- Realität
 - Berechnung der Personen-Monate aus Erfahrungen vorangegangener Projekte
 - Aufwand, der von einer Person in einem Monat unter Idealbedingungen erledigt werden kann
 - Zwischen 400 und 1600 Stunden (50-200 PT) pro 100 FP – dies entspricht 2,5-10 PM pro 100 FP (bzw 40 – 10 FP pro PM)

Beispiel: Web Content Management System (Zimmermann, 2005)

- Beispiel: Web Content Management an einer Hochschule:
<http://swt2.informatik.uni-halle.de/downloads//2005ws/swt/Folien/kostenschaetzung-4.pdf>

Eingaben:

- Daten- oder Steuerinformation
- Betritt von aussen die Anwendung oder ändert internen Datenblock
- ☞ Auch Eingaben durch andere Anwendungen

Ausgaben:

- Daten- oder Steuerinformation
- Verlässt Anwendung nach aussen
- ☞ Auch Ausgaben zu anderen Anwendungen

Abfragen:

- Suche nach Informationen im Datenbestand, keine Berechnungen
- Keine Änderungen interner Datenblöcke
- ☞ Komplexe Abfragen z.B. durch SQL-Anfragen werden nicht gezählt

Datenbestände:

- Vollständig im Innern der Anwendung, permanent
- Wird durch Anwendung gepflegt
- Auch logische Datenblöcke, organisatorische Klassifikation
- ☞ Z.B. jeder unterschiedliche Zugriffsweg zählt als Datenblock

Referenzdatenbestände:

- Datenbestände, die von Anwendung nicht gepflegt werden
- ☞ Werden nur gelesen, aber nicht geschrieben

Beispiel: Web Content Management System (Zimmermann, 2005)

Beispiel: Web-Content-Management System der MLU

/LF2_10/	Einfügen von Layoutelementen:	1 Eingabe
/LF2_20/	Bearbeiten von Layoutelementen:	1 Eingabe
/LF2_30/	Löschen von Layoutelementen:	1 Eingabe
/LF2_40/	Erstellen einer lokalen Kopie externer Seiten:	1 Eingabe
/LF2_50/	Einfügen von Formularen:	1 Eingabe
/LF2_60/	Bearbeiten von Formularen:	1 Eingabe
/LF2_70/	Löschen von Formularen:	1 Eingabe
/LF2_80/	Anfordern von Übersetzertätigkeit:	2 Eingaben
/LF2_90/	Bestätigen von Änderungen:	1 Eingabe
/LF3_10/	Einfügen von Rechten zur Pflege für Mitarbeiter:	1 Eingabe
/LF3_20/	Bearbeiten von Rechten zur Pflege für Mitarbeiter:	1 Eingabe
/LF3_30/	Löschen von Rechten zur Pflege für Mitarbeiter:	1 Eingabe
/LF3_40/	Uni-internes Bearbeiten von Zugriffsrechten:	1 Eingabe
/LF4_10/	Bearbeiten des Bereichsdesigns:	3 Eingaben
/LF5_10/	Erstellen von Bereichen:	1 Eingabe
/LF5_20/	Bearbeiten von Bereichen:	1 Eingabe
/LF5_30/	Löschen von Bereichen:	1 Eingabe

Beispiel: Web-Content-Management System der MLU

/LF0_10/	Korrespondenz mit Systemen der Universität:	1 Eingabe, 1 Ausgabe
/LF0_20/	Kontrolle der Konsistenz der Seiten:	1 Abfrage, 1 Eingabe
/LF0_30/	Erstellen von Übersichten:	1 Abfrage, 1 Ausgabe
/LF0_40/	Kontrolle der Aktualität der Inhalte:	1 Abfrage, 1 Ausgabe
/LF0_50/	Speicherung von persönlichen Daten:	1 Eingabe
/LF0_60/	Speicherung von Rechten:	1 Eingabe, 1 Abfrage, 1 Ausgabe
/LF0_70/	Speicherung von Seiteninhalten:	1 Eingabe
/LF0_80/	Speicherung von Layoutelementen:	1 Eingabe
/LF0_90/	Speicherung von Bereichszugehörigkeiten:	1 Eingabe
/LF0_100/	Speicherung von Metadaten:	1 Eingabe
/LF0_110/	Speicherung von Dokumenten im Fremdformat:	1 Eingabe
/LF0_120/	Speicherung von Bildern in diversen Formaten:	1 Eingabe
/LF0_130/	Speicherung von Cascading Style Sheets:	1 Eingabe
/LF1_10/	Abrufen der Seiteninhalte:	1 Abfrage, 1 Ausgabe
/LF1_20/	Abrufen der Metadaten:	1 Abfrage, 1 Ausgabe
/LF1_30/	Abrufen der Informationen zur Aktualität:	1 Ausgabe
/LF1_40/	Informationen an Seitenverantwortliche schicken:	1 Abfrage
/LF1_50/	Druckversion abrufen:	1 Ausgabe
/LF1_60/	Übersichten abrufen:	1 Eingabe, 1 Ausgabe
/LF1_70/	Suchabfrage erstellen:	1 Eingabe

Beispiel: Web Content Management System (Zimmermann, 2005)

Beispiel: Web-Content-Management System der MLU

/LD1_10/	persönliche Daten (Datenschutz):	1 Datenblock
/LD1_20/	Rechte:	1 Datenblock
/LD2_10/	Seiteninhalte:	1 Datenblock
/LD2_20/	Layoutelemente:	1 Datenblock
/LD2_30/	Bereichszugehörigkeit der verschiedenen:	1 Datenblock
/LD2_40/	Metadaten der verschiedenen Seiten:	1 Datenblock
/LD3_10/	Dokumente in Fremdformaten:	1 Datenblock
/LD3_20/	Bilder in verschiedenen Formaten:	1 Datenblock
/LD3_30/	Cascading Style Sheets für verschiedene Bereiche:	1 Datenblock

Beispiel: Web Content Management System (Zimmermann, 2005)

Eingaben:

Kriterium	einfach	mittel	komplex
# Datenelemente	1–5	6–10	> 10
Eingabepfung	formal	auch logisch	auch DB-Zugriff
Bedienerführung	gering	mittel	hoch

Abfragen:

Kriterium	einfach	mittel	komplex
# unterschiedliche Schlüssel	1	2	> 2
Bedienerführung	gering	mittel	hoch

Referenzdateien:

Kriterium	einfach	mittel	komplex
<i>Read-Only-Dateien</i>			
# unterschiedlicher Datenelemente	1–5	6–10	> 10
# Schlüssel/Satzarten	1	2	> 2
<i>Tabellen</i>			
# unterschiedlicher Datenelemente	1–5	6–10	> 10
Dimension	1	2	3

Beispiel: Web Content Management System (Zimmermann, 2005)

Ausgaben:

Kriterium	einfach	mittel	komplex
# Spalten	1-6	7-15	> 15
# unterschiedlicher Datenelemente	1-5	6-10	> 10
Gruppenwechsel	1	2-3	> 3
Datenelemente Druckaufbereitung	keine	einige	viele

Gruppenwechsel: logische Abfolge der Ausgabe von Datenelementen

Datenbestände:

Kriterium	einfach	mittel	komplex
# Schlüssel/Satzarten	1	2	> 2
# unterschiedlicher Datenelemente	1-20	21-40	> 40
Datenbestand vorhanden	ja	-	nein
Veränderung impl. Datenstruktur	nein	ja	-

Beispiel: Web Content Management System (Zimmermann, 2005)

Eingabe	# DE	EP	BF	Bewertung	
/LF0_10/	Korrespondenz mit anderen Systemen	> 10	DB-Zugriff	gering	mittel
/LF0_20/	Kontrolle der Konsistenz der Seiten	> 10	DB-Zugriff	gering	mittel
/LF0_30/	Erstellen von Übersichten	> 10	DB-Zugriff	gering	mittel
/LF0_50/	Speicherung persönlicher Daten	> 10	DB-Zugriff	mittel	komplex
/LF0_60/	Speicherung von Rechten	> 1	DB-Zugriff	mittel	mittel
/LF0_70/	Speicherung Seiteninhalte	> 10	DB-Zugriff	hoch	komplex
/LF0_80/	Speicherung Layoutelemente	> 10	DB-Zugriff	hoch	komplex
/LF0_90/	Speicherung Bereichszugehörigkeit	> 1	DB-Zugriff	gering	mittel
/LF0_100/	Speicherung Metadaten	1 - 5	DB-Zugriff	gering	mittel
/LF0_110/	Speicherung Fremddokumente	1 - 5	DB-Zugriff	gering	mittel
/LF0_120/	Speicherung Bilder	6 - 10	DB-Zugriff	gering	mittel
/LF0_130/	Speicherung CSS	> 10	DB-Zugriff	gering	mittel
/LF1_60/	Übersichten abrufen	1 - 5	formal	gering	einfach
/LF1_70/	Suchabfrage stellen	1 - 5	formal	gering	einfach
/LF2_10/	Einfügen Layoutelement	> 1	DB-Zugriff	mittel	mittel
/LF2_20/	Bearbeiten Layoutelement	> 1	DB-Zugriff	mittel	mittel
/LF2_30/	Löschen Layoutelement	1	DB-Zugriff	gering	mittel
/LF2_40/	Erstellen lokaler Kopie	1	DB-Zugriff	gering	mittel
/LF2_50/	Einfügen Formular	1 - 5	DB-Zugriff	hoch	mittel
/LF2_60/	Bearbeiten Formular	1 - 5	DB-Zugriff	hoch	mittel
/LF2_70/	Löschen Formular	1	DB-Zugriff	gering	mittel
/LF2_80/	Anfordern Übersetzen	1 - 5	DB-Zugriff	gering	mittel
/LF2_90/	Bestätigen Änderung	1	DB-Zugriff	gering	mittel
/LF3_10/	Einfügen Rechte	1 - 5	DB-Zugriff	gering	mittel
/LF3_20/	Bearbeiten Rechte	1 - 5	DB-Zugriff	gering	mittel
/LF3_30/	Löschen Rechte	1	DB-Zugriff	gering	mittel
/LF3_40/	Bearbeiten uniinterne Rechte	1	DB-Zugriff	gering	mittel
/LF4_10/	Bearbeiten Bereichsdesign	> 10	DB-Zugriff	hoch	komplex
/LF5_10/	Erstellen Bereich	6 - 10	DB-Zugriff	mittel	mittel
/LF5_20/	Bearbeiten Bereich	6 - 10	DB-Zugriff	mittel	mittel

Abfrage	AS	BF	Bewertung
/LF0_20/	Kontrolle der Konsistenz der Seiten	> 2 gering	mittel
/LF0_30/	Erstellen von Übersichten	1 gering	einfach
/LF0_40/	Kontrolle Aktualität der Inhalte	2 gering	einfach
/LF0_60/	Speicherung Rechte	1 gering	einfach
/LF1_10/	Abrufen Seiteninhalte	> 2 gering	mittel
/LF1_20/	Abrufen Metadaten	1 gering	einfach
/LF1_40/	Infos an Seitenverantwortliche	1 gering	einfach

AF: # unterschiedlicher Schlüssel, BF: Bedienungsführung

Ausgabe	AS	AD	GW	DD	Bewertung	
/LF0_10/	Korrespondenz mit anderen Systemen	> 15	> 10	1	keine	mittel
/LF0_40/	Kontrolle Aktualität der Inhalte	1 - 6	> 1 - 5	1	keine	einfach
/LF0_60/	Speicherung Rechte	1 - 6	> 1 - 5	1	keine	einfach
/LF1_10/	Abrufen Seiteninhalte	> 15	> 10	2 - 3	viele	komplex
/LF1_20/	Abrufen Metadaten	1 - 6	> 10	1	einige	einfach
/LF1_30/	Abrufen Aktualität	> 15	> 10	1	einige	mittel
/LF1_50/	Abrufen Druckversion	> 15	> 10	2 - 3	viele	komplex
/LF1_60/	Abrufen Übersicht	1 - 6	> 10	1	einige	einfach

AS: # Spalten, AD: # unterschiedlicher Datenelemente, GW: # Gruppenwechsel,
DD: Datenelemente Druckaufbereitung

Beispiel: Web Content Management System (Zimmermann, 2005)

Beispiel: Web-Content-Management System der MLU

Produkt	AS	AD	DB	DS	Bewertung
/LD1_10/ persönliche Daten	1	1-20	ja	ja	einfach
/LD1_20/ Rechte	1	1-20	nein	nein	einfach
/LD2_10/ Seiteninhalte	>2	21-40	-	-	komplex
/LD2_20/ Layoutelemente	>2	>40	nein	-	komplex
/LD2_30/ Bereichszugehörigkeit von Seiten	1	1-20	nein	nein	einfach
/LD2_40/ Metadaten von Seiten	1	1-20	nein	nein	einfach
/LD3_10/ Fremdformatdokumente	1	1-20	nein	nein	einfach
/LD3_20/ Bilder	1	1-20	nein	nein	einfach
/LD3_30/ CSS	1	1-20	nein	ja	mittel

AS: # Schlüssel/Satzarten, AD: # unterschiedlicher Datenelemente,
DB: Datenbestand vorhanden, DS: Veränderung der implementierten Datenstruktur

Kategorie	Anzahl	Klassifik.	Gewichtung	Zeilensumme
Eingaben	2	einfach	×3	= 6
	25	mittel	×4	= 100
	4	komplex	×6	= 24
Abfragen	5	einfach	×3	= 15
	2	mittel	×4	= 8
	0	komplex	×6	= 0
Ausgaben	4	einfach	×4	= 16
	2	mittel	×5	= 10
	2	komplex	×7	= 14
Datenblöcke	6	einfach	×7	= 42
	1	mittel	×10	= 10
	2	komplex	×15	= 30
Summe				= 275

Beispiel: Web Content Management System (Zimmermann, 2005)

- 1 Verflechtung mit anderen Anwendungssystemen (0-5)
- 2 Dezentrale Verwaltung/Verarbeitung der Daten (0-5)
- 3 Hohe Transaktionsrate (0-5)
- 4 Verarbeitungslogik
 - a Komplexe Rechenoperationen, verbunden mit Simulationen/Hochrechnungen (0-10)
⇒ Mittelung
 - b Umfangreiche Kontrollverfahren zur ordnungsgemäßen Verarbeitung der Daten (0-5)
 - c Anzahl von Ausnahmeregelungen (0-10)
 - d Komplexe Logik (0-5)
- 5 Anteil von Wiederverwendung:

< 10%	10-20%	20-30%	30-40 %	40-50%	> 60%
0	1	2	3	4	5
- 6 Datenbestandskonvertierungen (0-5)
- 7 Anpassbarkeit (0-5)

1 Verflechtung mit anderen Systemen	3
2 dezentrale Daten/Verarbeitung	3
3 Transaktionsrate	1
4 Verarbeitungslogik	
a Rechenoperationen	1
b Kontrollverfahren	3
c Ausnahmeregelungen	1
d Logik	2
5 Wiederverwendbarkeit	1
6 Datenbestandskonvertierungen	1
7 Anpassbarkeit	1
Summe	17

$$FP_{\text{bew}} = FP \cdot \left(\frac{\text{Einflussfaktor}}{100} + 0.7 \right) \quad \rightarrow \text{vgl. S. 46}$$

Im Beispiel:

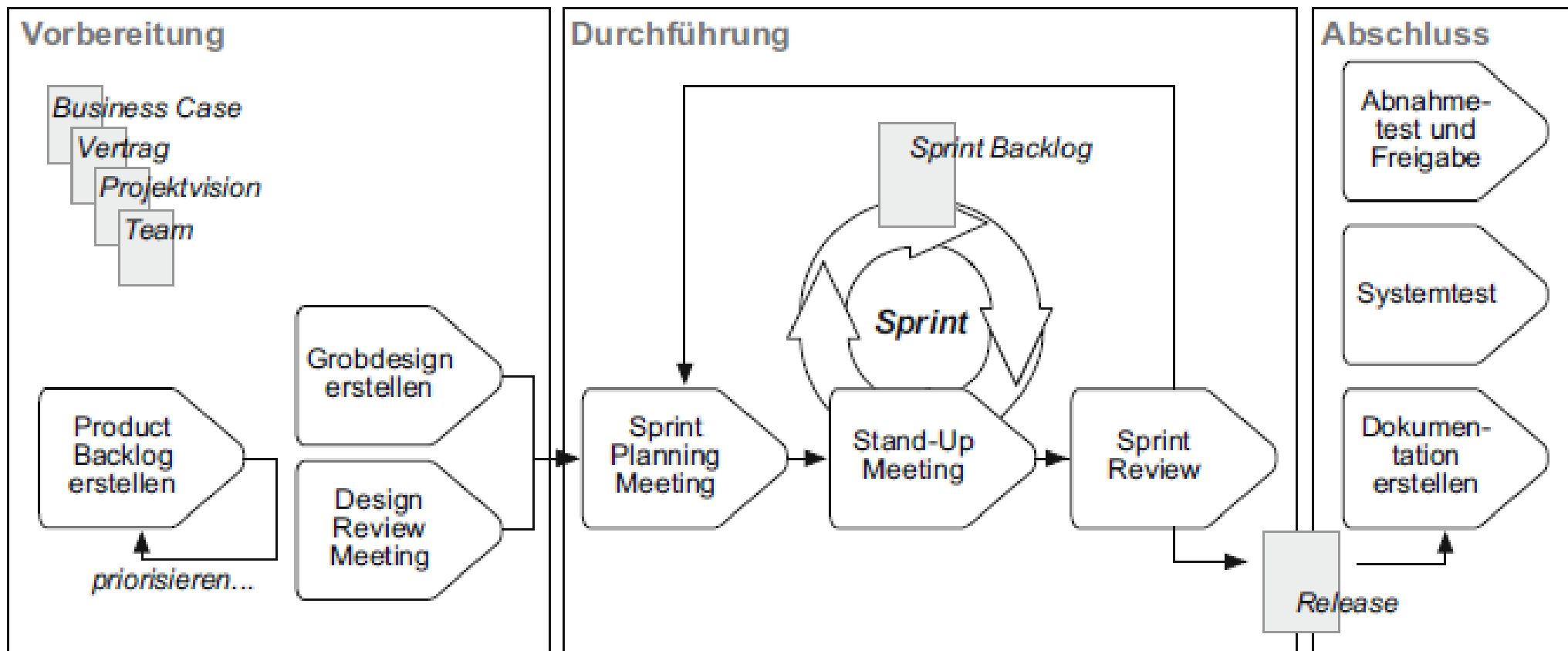
$$FP_{\text{bew}} = 275 \cdot 0.87 = 239.25$$

Fazit zur Function-Point Analyse

- Wissenschaftlich fundiert
- Schätzdetails nur von speziellen Schätzexperten beurteilbar
- firmenspezifische und methodenspezifische Optimierung schwierig
- Wenn FPA eingesetzt werden soll, muss sie (evtl. rückwirkend) für verschiedene Projekte evaluiert werden
- sinnvoll, wenn eigene Schätzabteilung zur Betreuung aller Schätzungen aufgebaut wird
- wesentlich höherer Aufwand als später vorgestellte vereinfachte Analogieschätzung, Schätzer fühlen sich nicht für Schätzergebnis verantwortlich

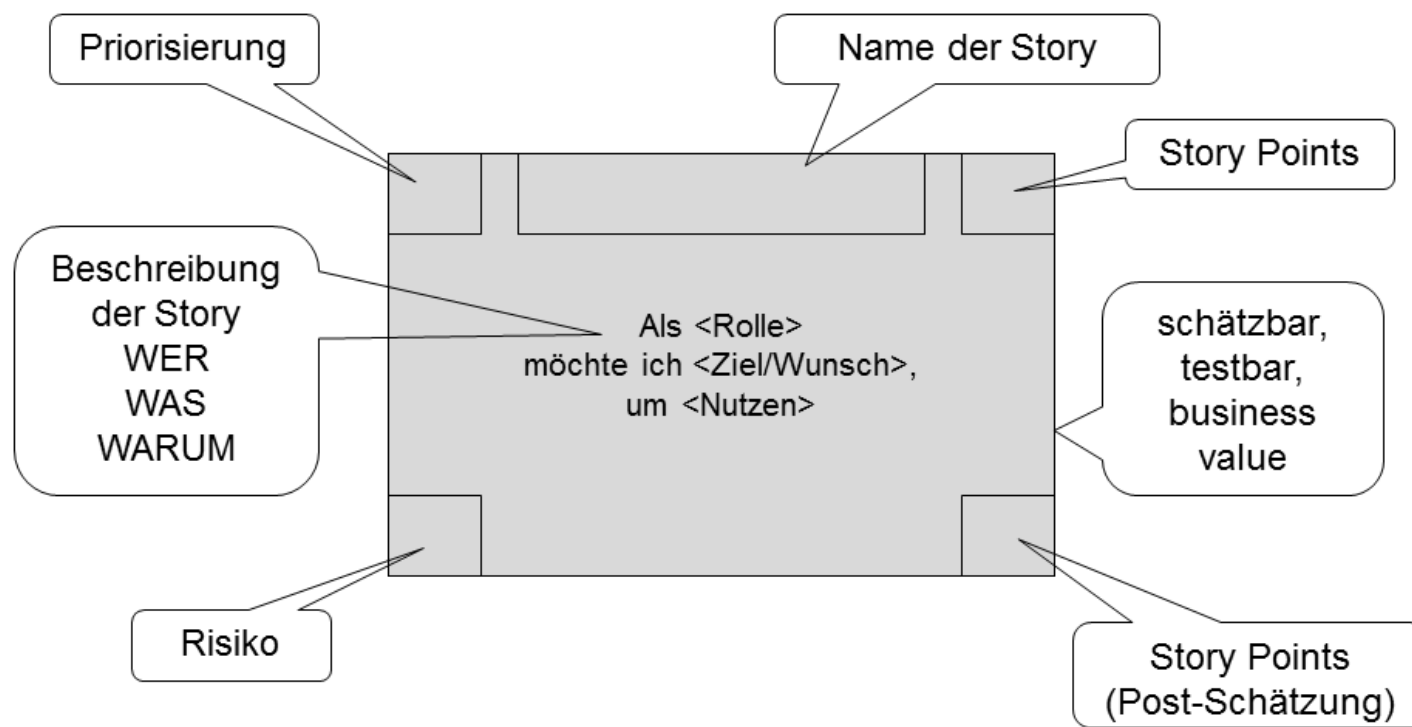
SCRUM: Schätzverfahren

- Problemstellung: flexible Entwicklungsplanung, trotzdem Notwendigkeit guter Schätzungen
- Startpunkt: Product Backlog (Anforderungen), Sprint Backlog (Aufgaben für eine Entwicklungsperiode) und User Stories



User Stories

- Was soll ein Stakeholder machen können?
- Verfeinerung in Aufgabenbeschreibung (z.B. Use Case)
- Aufwand: Story Points



Quelle https://commons.wikimedia.org/wiki/File:Story-Cards_Vorderseite.gif

Planning Poker

- Abschätzung der Komplexität einer Aufgabe
- Äquivalent zu Delphi-Methode (Moderator, Expertenschätzung, Konsens)
- Klärung bei zu großen Paketen oder Unklarheiten („100“, „?“)
- Ergebnis: Einigung auf story points pro user story
- Finaler Schritt: Zuweisung von Aufwänden – auf Basis der Teamhistorie (z.B. 20 story points pro sprint)
- Entwicklungsgeschwindigkeit $v = s/t$ (story points pro sprint)



Zusammenfassung: Abschätzen der Projektkosten

- „Personalkosten“

- Detaillierung und Zuverlässigkeit abhängig von Schätzverfahren bzw. Aufwandsschätzmethode

$$\text{Kosten} = \sum \text{Zeitaufwand}_i \cdot \text{Kostensatz}_i$$

- Grundsätzliche Möglichkeiten

- Mischkostensatz (Erfahrungswert aus der Vergangenheit)

- Kosten für bestimmte Stellenprofile gemäß geplanten Stellenprofil

- Erst wenn Personen und die Zuordnung zu Tasks feststeht können Projektkosten genau geplant werden

- tatsächliche Kosten ansetzen

- u.U. Datenschutzprobleme vs. Lohnkosten

Zusammenfassung Abschätzen der Projektkosten

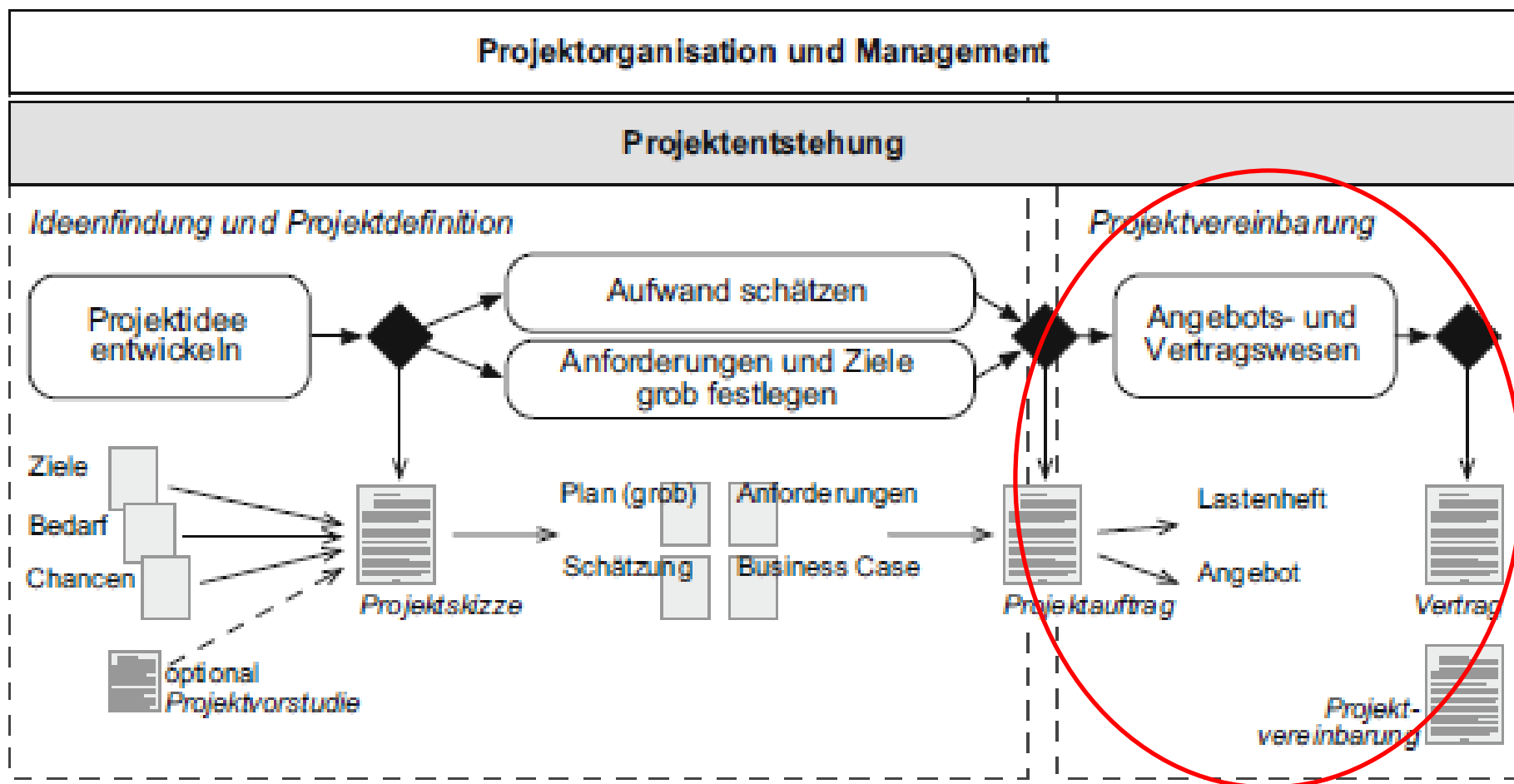
- *Abschätzen der Projektkosten: „Sonstige Aufwendungen“*
 - Reisekosten
 - Consultants
 - Sonstige Lieferungen u. Leistungen
 - Benötigte Hard- u. Software
 - Sonstiges Equipment
 - Training
 - Bürokosten
 - Verbrauchsgüter
 - Telekommunikationskosten
 - Administration
 - Incentives
 - Abschlussfeier
 - ...

Fazit

- Schätzverfahren basieren auf Werten, die aus der Praxis ermittelt wurden
- Viele Diskussionen um Anpassung der Parameter
- Erfahrene Projektmanager sind notwendig, um gute Schätzungen vorzulegen
- Gerade zu Projektbeginn ist die Abschätzung häufig schwierig, eine durchgehende Neuabschätzung ist sinnvoll

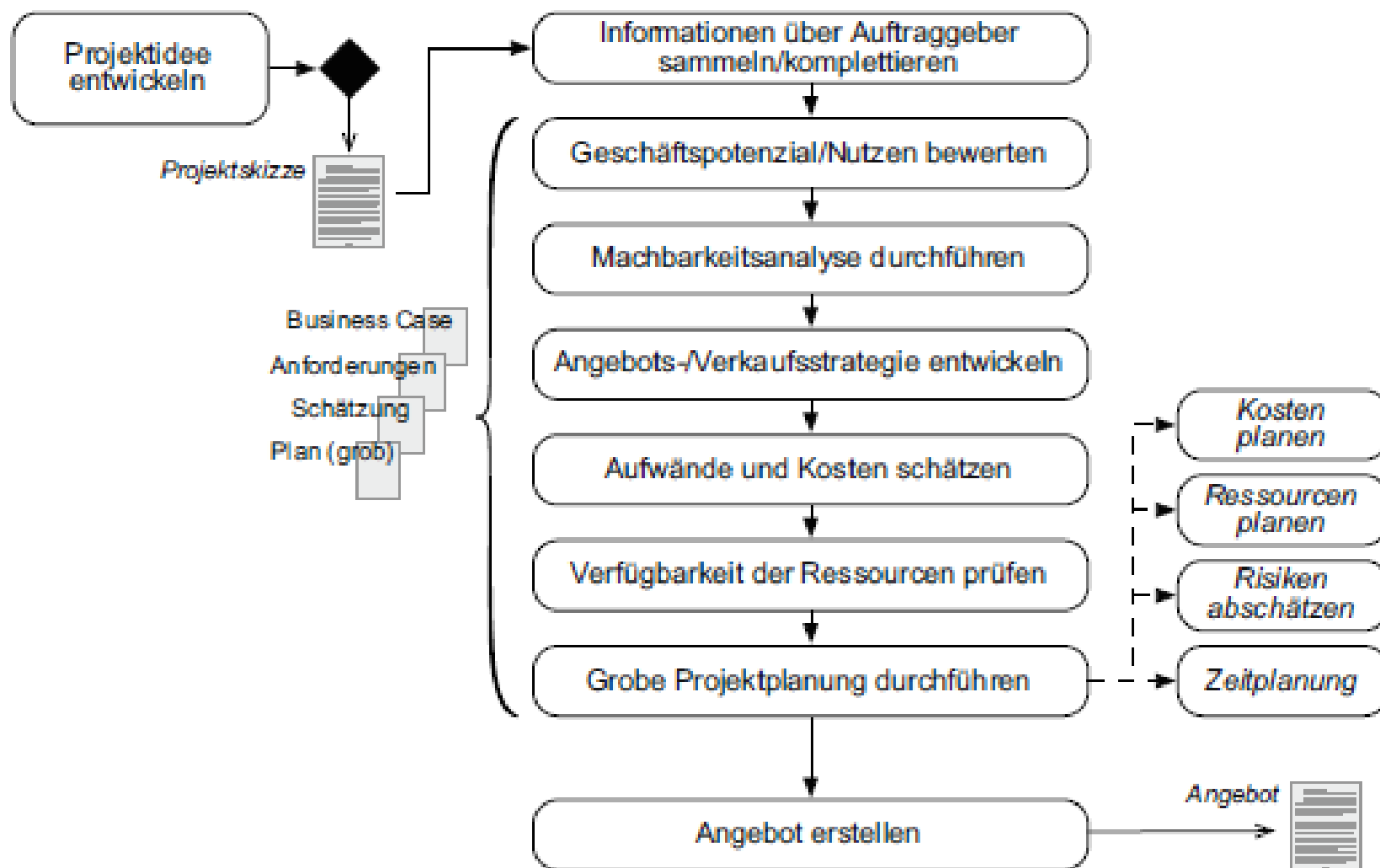
Angebotserstellung

- Formalisierung vor der Beauftragung



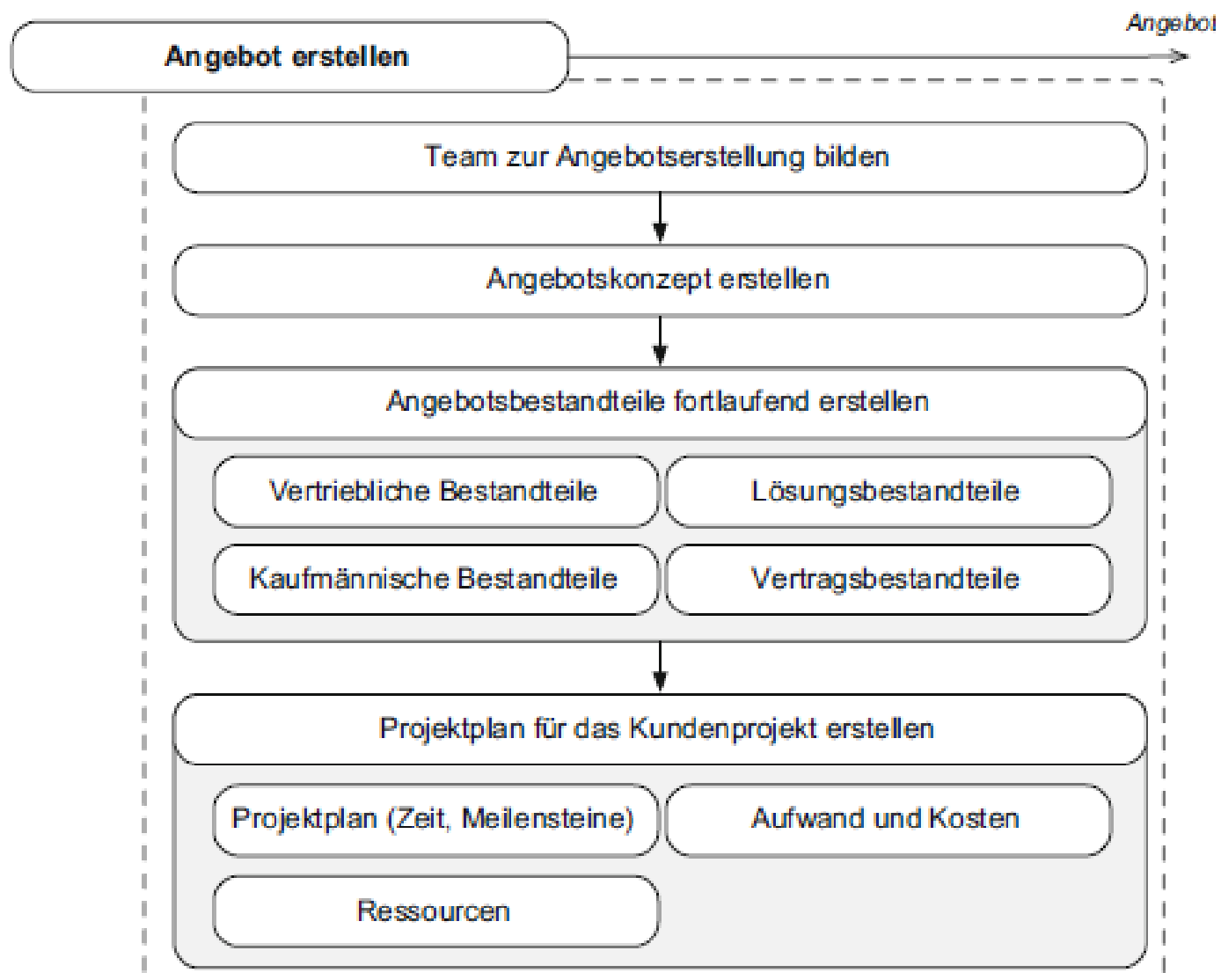
(Quelle Broy, 2013)

Angebotserstellung: Schritte



(Quelle Broy, 2013)

Angebotserstellung: Bestandteile



(Quelle Broy, 2013)

Rechtliche Aspekte

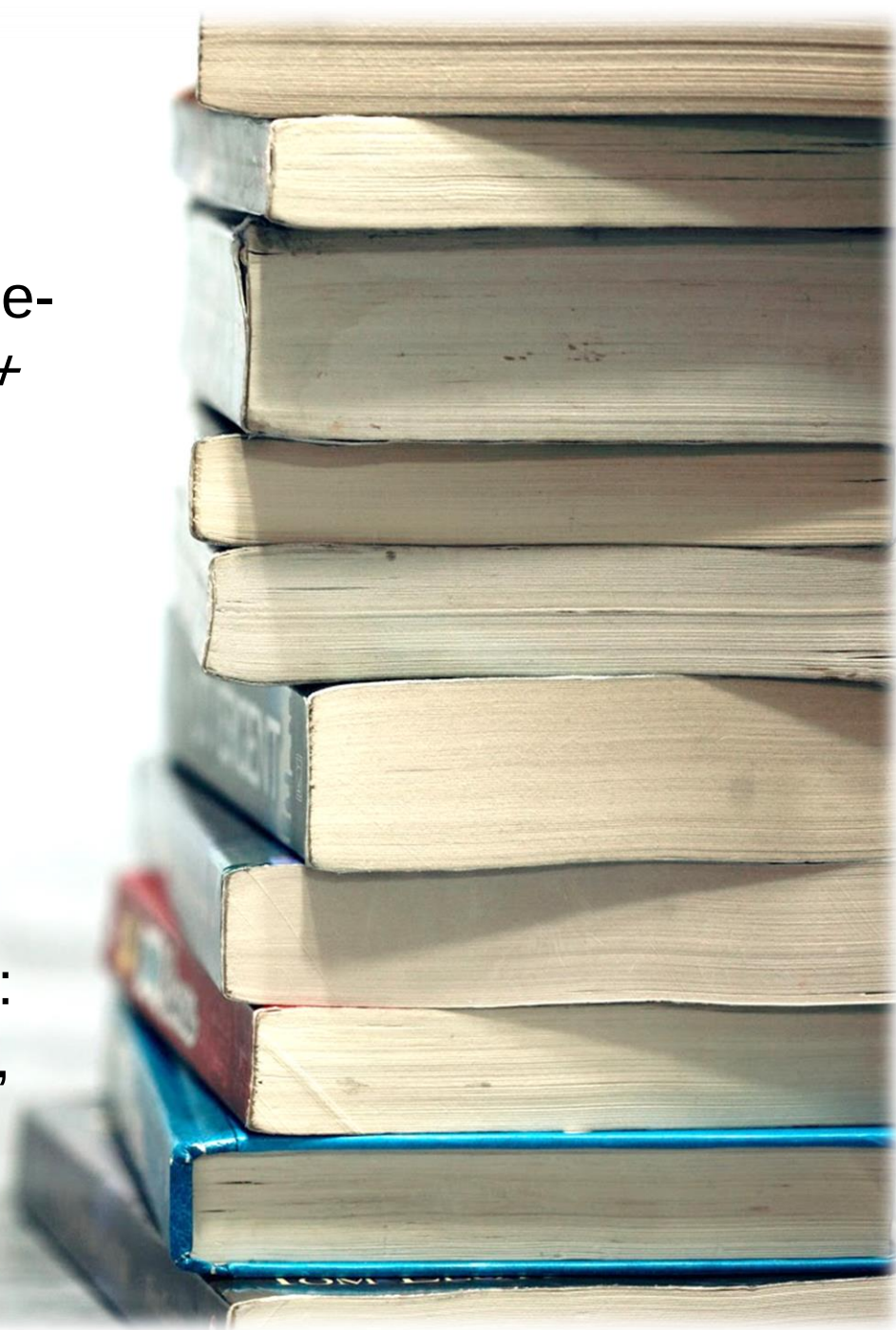
- Leistungen und Verantwortlichkeiten mit Regelungen zu:
 - Funktionsumfang
 - Terminen (auch Modalitäten bei Lieferverzug)
 - Qualitätsanforderungen
 - Preise und Zahlungsvereinbarungen
 - Abnahme
- Gewährleistungen und Haftungsvereinbarungen
- Bereitstellungen und Beistellungen
- Nutzungsrechte
 - Urheberrechts und Urheberschutzes
 - Übertragungsrecht an Dritte (Recht auf Weitervertrieb)
- Einhaltung gesetzlicher Vorschriften

Ich bin bis hier gut mitgekommen ...



Literatur

- Kleuker, S. Grundkurs Software-Engineering mit UML. *Vieweg+Teubner Verlag, Wiesbaden*. 2013.
- Broy, M., Kuhrmann, M.: Projektorganisation und Management im Software Engineering, Springer Vieweg, 2013. (als Ebook verfügbar)
- Wieczorrek, H.W., Mertens, P.: Management von IT Projekten, Springer, 2011. (als Ebook verfügbar)



Ergänzungsfolien

Von CoCoMo81 zu CoCoMo II

Kostenschätzverfahren von Barry W. Boehm (ab 1981)

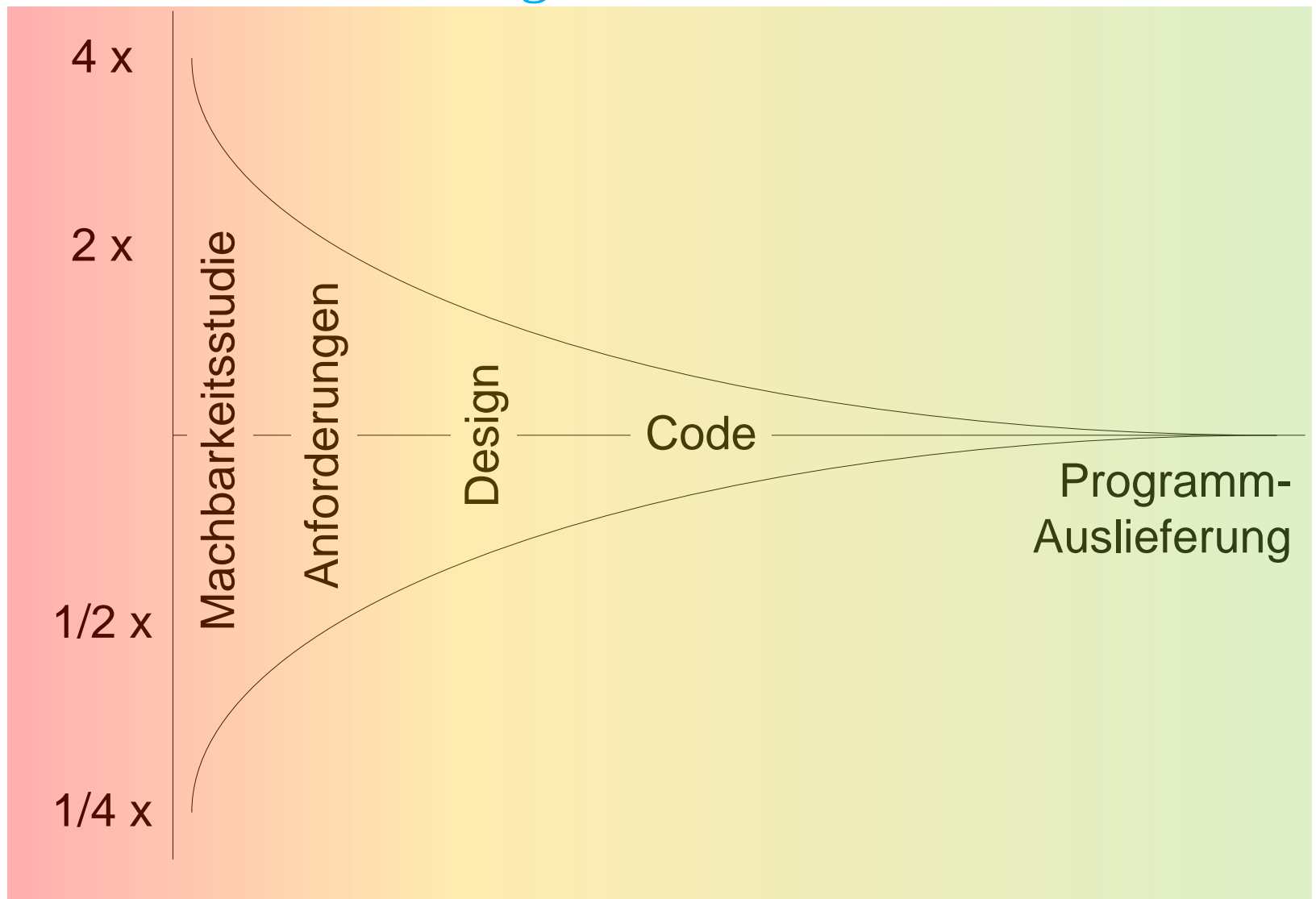
- Die Art der SW-Entwicklung hat sich in den Jahren wesentlich verändert (Methodik)
- Die Programmiersprachen haben sich verändert (Aufkommen von OO-Sprachen)
- Man kann weitere Einflussfaktoren auf den Aufwand feststellen
- Es ist interessant, eine Schätzung zu verschiedenen Zeitpunkten des Projektfortschritts durchzuführen
- Mittlerweile viele Varianten für verschiedene Vorgehensmodelle und Projektarten

-> CoCoMo (Constructive Cost Model)

- Prototypentwicklung (Object Points)
- Grobe Abschätzung (Function Points)
- Entwicklung bei Wiederverwertung
- Detaillierte Abschätzung nach Systemdesign (LOC)

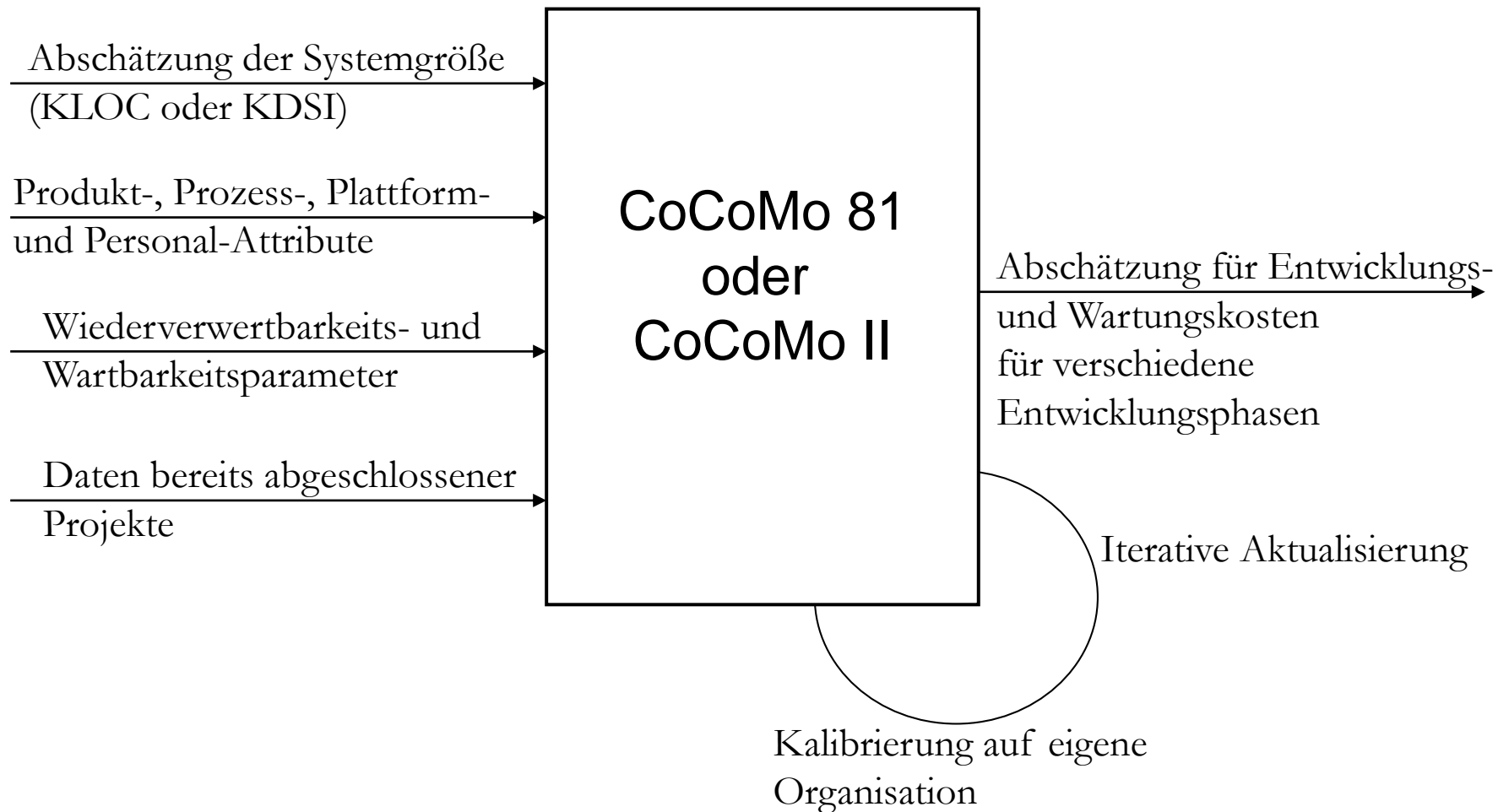
http://sunset.usc.edu/csse/research/COCOMOII/cocomo_main.html

Unsicherheit der Abschätzung



Nach [Sommerville 2005]

CoCoMo II - Einführung



Anmerkung: Die folgenden Folien basieren auf (Linke, 2010)!

CoCoMo II - Merkmale

Merkmale:

CoCoMo ist ein Modell zur **Kosten- & Aufwandsschätzung** von Software, die mit einer **Basisformel** agiert, welche mit **Parametern aus vergangenen und aktuellen** Projekten und deren Charakteristiken festgelegt wird.

Die Parameter werden anhand von **Kosten- und Skalenfaktoren** bestimmt:

„Bei der Schätzung der Projektkosten gilt es die wichtigsten Kostenfaktoren (cost drivers) besonders zu betrachten. Zu diesen zählen die Variablen, die, die Anzahl der, für das Projekt aufzuwendenden Sätze an Geld, beeinflussen.“

Zentraler Kostenfaktor ist die **Größe des Softwareproduktes** (Size-Driven Model)

CoCoMo II – Submodelle (1)

CoCoMo II besteht aus drei Submodellen, die jeweils verschiedene Faktoren für verschiedene Phasen der Abschätzung bieten:

1 - Application Composition Model

- Früheste Phase der Abschätzung : Prototypisierung

2 - Early Design Model

- Abschätzung zur Analyse und Prototyping-Phasen
- Vor der Systemarchitektur
- Wenig von der Architektur bekannt, daher sind eingehende Schätzungen grob
- 7 Kostenfaktoren werden berücksichtigt

3 - Post-Architecture Model

- Abschätzung nach Festlegung der Softwarearchitektur, vor und beim Kodieren
- 17 Kostenfaktoren werden berücksichtigt

CoCoMo II – Wie funktioniert CoCoMo II ?

1. Function Point Analyse (oder äquivalente Systemgrößen Abschätzung)
-> ergibt Größe (LOC – Lines Of Code)
2. Ermitteln der Skalenfaktoren
-> Berechnung des nominalen Aufwandes in Personen Monaten (PM)
3. Ermitteln der Aufwands Faktoren (Cost Drivers)
-> Berechnen des adjustierten Aufwandes in PM

CoCoMo II – Function Point Analyse (3)

Ermitteln der Function Points

$$FP = RFP \times (0,65 + (0,01 \times SI))$$

$$FP = 34 \times (0,65 + 0,01 \times 45) = 37,4 \text{ FP}$$

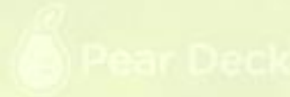
Ermitteln der Lines Of Code für C++ (LOC)

$$37,4 \text{ FP} \times 91 \text{ LOC} / \text{FP} = 3403,4 \text{ LOC}$$

Vgl. <http://www.qsm.com/resources/function-point-languages-table>

CoCoMo II – Vorteile

- Einfaches Verfahren
- CoCoMo ist ein Industriestandard
- CoCoMo ist ein offener Standard
- Sehr fundierte Informationen sind einfach verfügbar
- Wissen auswärtiger Experten ist verfügbar
- Rückwärtskompatibilität
- Verschiedene Erweiterungen für fast jeden Zweck
- Frei wählbarer Detailgrad
- Softwaretool Support
- Bessere Abschätzungen, je länger Projektdatenbanken gepflegt werden



CoCoMo II – Nachteile

- CoCoMo II basiert noch größtenteils auf dem Wasserfall-/Spiralmodell
- Viele der Erweiterungen sind experimentell und noch nicht ausreichend Kalibriert
- Zeitabschätzung ist für kleine Projekte ungeeignet
- Erfahrung ist nötig für das wählen der richtigen Faktorbewertungen
- Bewertungen bedürfen einer tiefgehenden Kenntnis des Projektumfeldes und der Rahmenbedingungen



CoCoMo II – Software

Es gibt professionelle Software, welche die Analyse erleichtert

CoCoMo II <http://www.spc.ca>
CoStar <http://www.softstarsystems.com>
CostXpert <http://www.costxpert.com>
Effort Estimator <http://qb0x.net/projects/EffortEstimator/>

CoCoMo Estimate Software für PDAs

<http://my.raex.com/FC/B1/phess/coco/>

Online Formulare (Java oder Javascript)

<http://ch.twi.tudelft.nl/~arthur/cocomo/>

<http://www.engin.umd.umich.edu/CIS/tinytools/cocomo.html>

<http://tiefighter.et.tudelft.nl/~arthur/cocomo/>

<http://www.siegfried-seibert.de/cocomo.htm>

CoCoMo II – Ausblick

- Die Kalibrierung von CoCoMo soll verbessert werden
- Jährliche Kalibrierung soll erstellt und publiziert werden



HOCHSCHULE RUHR WEST
UNIVERSITY OF APPLIED SCIENCES

INSTITUT INFORMATIK

Kontakt

Hochschule Ruhr West University of Applied Sciences

Martin Idzik

Martin.idzik@hs-ruhrwest.de

Prof. Dr. Jan M. Pawlowski

jan.pawlowski@hs-ruhrwest.de

<http://en.hochschule-ruhr-west.de/research/computer-science/institute-members/professor-jan-pawlowski.html>