

IT-Projektmanagement Schätzung

Kaiserslautern, WS 2008/2009

Dr. Gerhard Pews

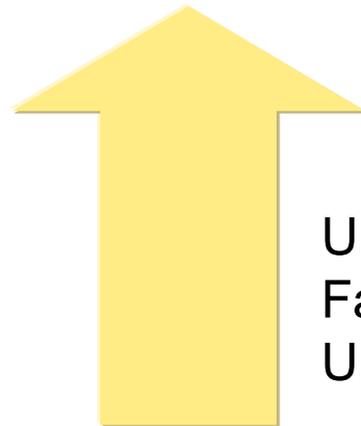
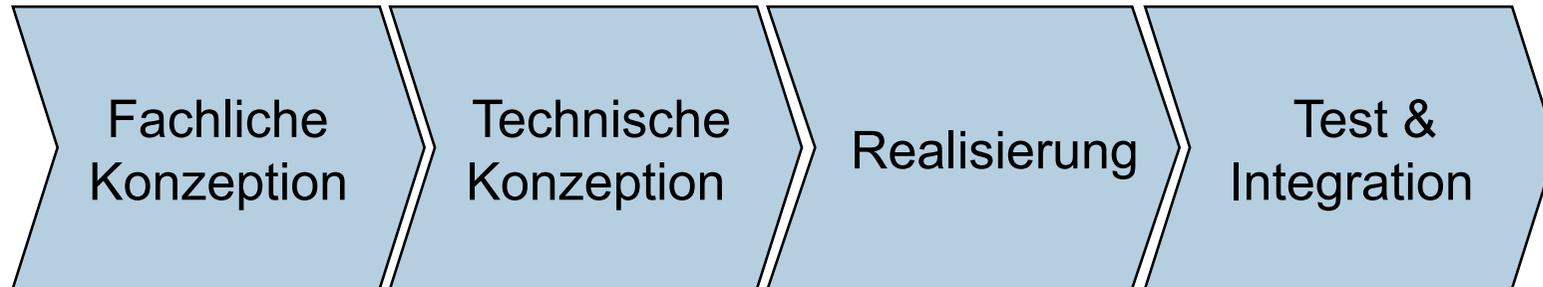
AGENDA

- **Allgemeine Grundlagen zur Schätzung**
- Function Point Verfahren
- Expertenschätzung, Delphi-Verfahren
- CoCoMo Verfahren

Voraussagen sind schwierig, vor allem, wenn sie die Zukunft betreffen.

Mark Twain

Wann ist eine Schätzung machbar? Und wann wird sie gemacht?



Umfang der Funktionalität bekannt
Fachliche Funktionen, Masken
Umsetzung bekannt

Aufwandsschätzung

- Schätzungen sind „unfair“
 - Passieren zu einem sehr frühen Zeitpunkt
 - Man weiß wenig über die Aufgabe
 - Auf die Zahlen wird man später festgenagelt

Herangehensweisen

- Von den Arbeitspaketen zur Aufwandszahl
 - Schätzung der einzelnen Arbeitspakete
 - Summe ergibt Gesamtaufwand
- Schätzung als Grundlage für Aufwand
- Vom fixierten Aufwand zu den Arbeitspaketen
 - Frage: Was darf es denn insgesamt kosten?
 - Projekt so aussteuern, dass es im Kostenrahmen bleibt
 - Aufgaben werden nur so „gut“ erledigt, wie Geld da ist
- Schätzung zur Prüfung der Machbarkeit

Auswirkungen einer „falschen“ Schätzung

- Zu hohe Schätzung
 - Ist kaum festzustellen. Jedes Projekt schöpft den Kostenrahmen voll aus.
 - Gefahr: Business Case rechnet sich nicht, bzw. Projekt wird an Mitbewerber verloren.
- Zu geringe Schätzung
 - Geld reicht nicht, Projekt kann im Budget nicht durchgeführt werden
- Das Projektmanagement hat extrem großen Einfluss auf die verbrauchten Aufwände eines Projekts. Im Nachhinein ist schwer festzustellen, ob der geschätzte Kostenrahmen realistisch war.

Motivation für eine Schätzung

- Wenn eine Schätzung so fehlerbehaftet und schwierig ist, warum schätzt man dann überhaupt?
 - Auch eine „falsche“ Schätzung ist besser als ein kompletter Blindflug
 - Auch Schätzen ist ein Prozess: sobald erste Erfahrungen und ermittelte Aufwände im Projekt vorliegen, wird nachgeschätzt und die Schätzung korrigiert
 - Die Schätzung wird im Laufe des Projekts immer genauer.

Einflussfaktoren auf den Aufwand (wichtigste Faktoren)

Umzusetzende Fachlichkeit (funktional)

- Masken
- Druckstücke
- Batches
- Berechnungen
- zu berücksichtigende
Fehlersituationen
- Migrationen aus Altsystemen
- Abhängigkeit von andern Systemen

PM-Faktoren

- Team
 - Mitarbeiterqualifikation
 - Erfahrung
 - Eingespieltes Team
- Projektorganisation
- Projektvorgehen, Methodik
- Unterstützung durch Tools

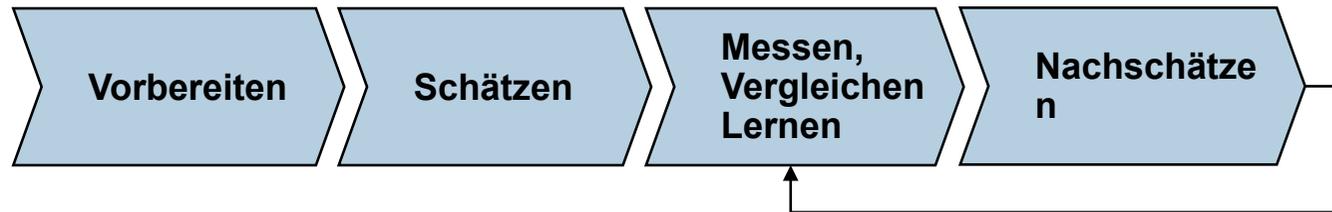
Technologische Umsetzung, nicht-funktionale Anforderungen

- Performance, Antwortzeitverhalten
- Mengengerüste
- Architektur
- Systemplattform, Basis-Technologien

Sonstige Faktoren

- Auftraggeber
- Aufwände steigen mit Größe
der Aufgabe

Generelles Vorgehen bei der Schätzung



- Eine gute Vorbereitung ist elementar für die Schätzung
 - Kompletieren und Strukturieren der Schätzgrundlage (osintots – oh shit, I never thought of this)
 - Sammeln aller Faktoren
- Nachschätzen und aus Projekterfahrungen lernen ist ein stetiger Kreislauf während des Projekts

AGENDA

- Allgemeine Grundlagen zur Schätzung
- **Function Point Verfahren**
- Expertenschätzung, Delphi-Verfahren
- CoCoMo Verfahren

Vorgehen bei der Function Point Schätzung



- Anmerkungen:
 - Function Point Schätzungen werden in verschiedenen Firmen unterschiedlich gelebt.
 - Alle arbeiten nach dem gleichen Prinzip, Unterschiede gibt es in:
 - Kriterien, nach denen die Komplexität der Fachlichkeit gemessen wird
 - Betrachtete sonstige Einflussfaktoren
 - Unternehmensspezifische Gewichtung
- Im Folgenden: ursprüngliches Verfahren

Vorgehen bei der Function Point Schätzung



- Unterteilung der Fachlichkeit:
 - Eingabedaten (Bildschirm, Batch, etc.)
 - Ausgabedaten (Bildschirm, Druck, Interface, etc.)
 - Anfragen (Suchanfragen)
 - Eigene Datenbestände (lesen & schreiben)
 - Extern referenzierte Datenbestände (nur lesen)
- Zu jedem Punkt Bewertung
 - geringe Komplexität
 - mittlere Komplexität
 - hohe Komplexität
- Ableiten der FP aus Tabelle, z. B.:
 - Eingabedaten
 - Gering = 3 FP
 - Mittel = 4 FP
 - Hoch = 6 FP
 - Eigene Datenbestände
 - Gering = 7 FP
 - Mittel = 10 FP
 - Hoch = 15 FP

Vorgehen bei der Function Point Schätzung

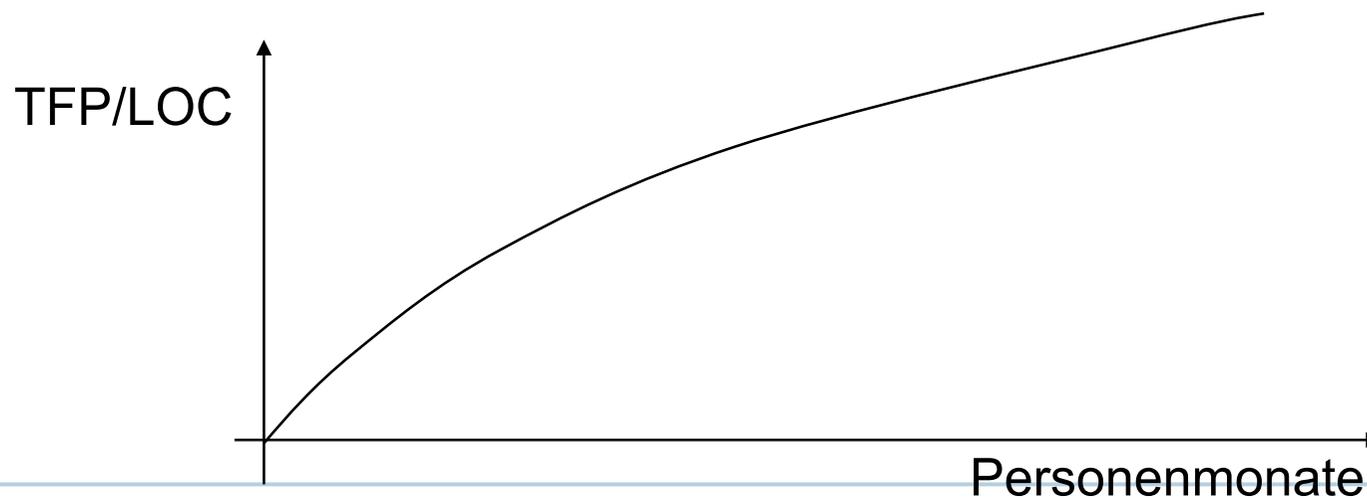


- Sonstige Einflussfaktoren
 - Verflechtung mit anderen Systemen
 - Dezentrale Verarbeitung und Datenhaltung
 - Transaktionsrate und Antwortzeitverhalten
 - Verarbeitungskomplexität (Rechenoperationen, Ausnahmebehandlungen, Logik, ...=)
 - Wiederverwendbarkeit
 - Migrationen
 - Benutzerfreundlichkeit
- Auch daraus wird wieder ein numerischer Faktor abgeleitet.

Vorgehen bei der Function Point Schätzung



- Gesamtkomplexität in „Total Function Points“ (TFP)
- Errechnung des Aufwands z. T. mit Zwischenschritt über die zu erstellenden Codezeilen (Lines of Code – LOC) für eine jeweilige Programmiersprache.



Bewertung Function Point

- Systematische Herangehensweise
- In Function Points wird die fachliche Komplexität der Aufgabe gemessen, nicht die Komplexität der technischen Lösung.
- Lebt von den jeweiligen Erfahrungswerten des Unternehmens und der Personen.
- Wird in unterschiedlichen Unternehmen auch unterschiedlich gehandhabt.
- Nach einiger Zeit der Anwendung erzielt man reproduzierbare Ergebnisse in der Function Point Messung.

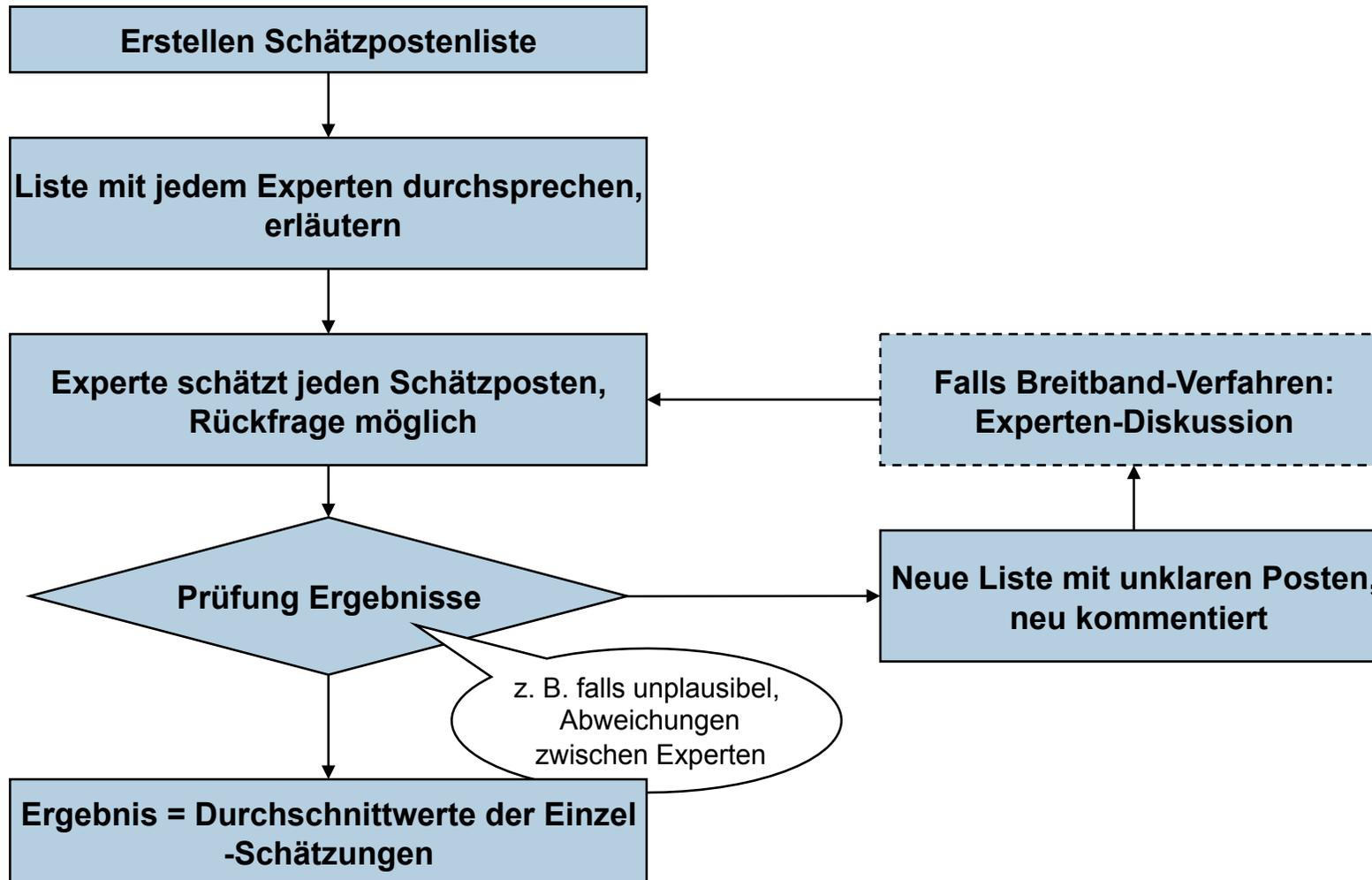
AGENDA

- Allgemeine Grundlagen zur Schätzung
- Function Point Verfahren
- **Expertenschätzung, Delphi-Verfahren**
- CoCoMo Verfahren

Allgemeines zur Expertenschätzung

- Aufwände und Umsetzung werden explizit geschätzt, nicht über Lines of Code oder Faktoren.
- Bei einer Expertenschätzung sind in der Regel mindestens zwei Experten beteiligt, um Schätzergebnisse vergleichen zu können.
- Generelle Unterscheidung in der Vorgehensweise:
 - Standard-Delphi-Verfahren: Experten schätzen komplett unabhängig
 - Breitband-Delphi-Verfahren: Experten diskutieren Zwischenergebnisse.

Vorgehen bei Expertenschätzung



Beispiel für Vorgehen bei einer Expertenschätzung: Bilden der Schätzpostenliste

- Benötigt: vollständige funktionale Beschreibung des Systems
- Beispiele für Schätzposten
 - Dialog
 - Druckstück
 - Funktionen, Services
 - Entitäten & Persistenz der Entitäten
 - Querschnittliche Funktionen (Drucken, Fehlerbehandlung, etc.)
- Granularität eines Schätzpostens (Daumenregeln)
 - nicht kleiner als 1 Tag, sonst addieren sich Nichtigkeiten zu großen Aufwänden
 - nicht größer als 20 Tage, sonst wird die Schätzung zu ungenau
 - guter Bereich: 5 – 10 Tage

Beispiel für Vorgehen bei einer Expertenschätzung: Schätzeinheit „BT“

- BT = Bearbeitungstag, auch PT = Personentag oder MT = Manntag
 - Die Arbeit, die eine Person an einem Tag erledigen kann
 - Die Arbeit ist brutto gerechnet, d. h. die wirkliche Zeit, die man benötigt, d. h. Entwicklung inklusive:
 - Entwicklertest
 - Code-Dokumentation
 - Nacharbeiten
 - Reisekostenabrechnung, Stundenkontierung in SAP, ...
 - Kaffee trinken, Zigarrettenpause
 - Teilnahme an Meetings
 - Anderen Kollegen helfen
 - Rechner ist abgestürzt, Netzwerk ist weg,...
 - ...

Beispiel für Vorgehen bei einer Expertenschätzung: Hochrechnung der Schätzergebnisse

- Geschätzt wird die reine Brutto-Realisierung
 - Alle fachlichen Fragen sind geklärt
 - Algorithmen sind klar
 - Technologie ist klar
 - Mitarbeiter ist geschult
 - „Normale“ Projektmitarbeiter, keine Technologie-Experten (wie diejenigen, die schätzen)
- Aus den Aufwänden für die Realisierung wird auf die Gesamtaufwände für das ganze Projekt hochgerechnet.
- Detaillierte Schätzung ist erst nach fachlicher Konzeption möglich, deshalb kennt man die Aufwände der fachlichen Konzeption schon bzw. kann sie abschätzen. Diese Aufwände kann man mit den geschätzten Realisierungsaufwänden vergleichen.

Beispiel für Vorgehen bei einer Expertenschätzung: Hochrechnung und Verifikation der Schätzung

- Nach abgeschlossenem Fachkonzept
- Orientierung an Erfahrungswerten

(hier: Capgemini sd&m, ähnlich bei anderen Firmen)



Beispiel für Vorgehen bei einer Expertenschätzung: Große Mengen von Schätzposten

- Problem: Was tun, wenn die Schätzpostenliste sehr groß ist, z. B. wenn das System mehrere hundert Dialoge umfasst? Der Aufwand zur Schätzung wird dann sehr groß?
- Lösung 1: Bildung von Klassen, z. B.:
 - Einfache Dialoge, Mittelschwere Dialoge, Schwierige Dialoge
 - Extrem schwierige Dialoge trotzdem individuell schätzen
 - Schätzen von einem Repräsentanten jeder Klasse
- Lösung 2: Schätzen von Stützpunkten, z. B.:
 - Fünf Dialoge wählen, die das ganze Spektrum der Komplexität abdecken.
 - Andere Dialoge werden nicht geschätzt, sondern mit den Stützpunkten verglichen. Aufwandszahlen der Stützpunkte werden übernommen und ggf. leicht angepasst.

Beispiel für Vorgehen bei einer Expertenschätzung: Erfahrungswerte Zuschläge

Zuschlag	Erfahrungswerte
Technik	
- Software-Entwicklungsumgebung	Aufbau geplant, Pflege MA über Zeit
- Technische Infrastruktur	5-10%, oder MA über Zeit
- Konfigurationsmanagement	Aufbau geplant, Pflege MA über Zeit
Datenadministration (optional)	5-10%
Abnahmesupport	MA über Zeit
Chefdesign (CD)	5-10%, oder MA über Zeit
Qualitätssicherung (QS)	Aktivitäten oder 10-20%
Einarbeitung	2-4 Wochen bei Projekteinstieg
Team-Meetings	MA über Zeit
PM/PL	10-25%, 1 PL pro ca. 7 MA über Zeit
Puffer bzw. Risikozuschlag	10-40%
Gewährleistung	3-12%
Sonstiges	Projektspezifisch
Reisezeit	MA über Zeit
Reisespesen	ableitbar aus Reisezeit
Zugekaufte Leistungen	tatsächliche Kosten

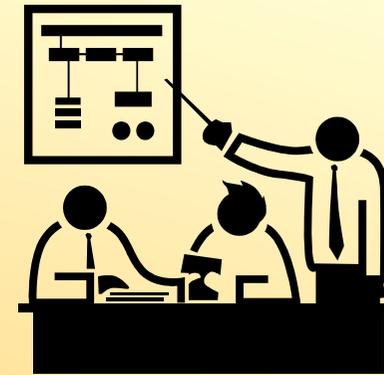
→ Bei Großprojekten kann die Realisierung nach Berechnung aller Zuschläge nur noch 16% des Gesamtaufwands betragen!

Beispiel für Vorgehen bei einer Expertenschätzung: Min – Max Schätzung

- Weitere Verfeinerung der Schätzung durch Min-Max-Schätzung möglich.
- Experte schätzt minimalen und maximalen Aufwand für die Aufgabe
 - Wichtig: kein zu großes Delta zwischen Min und Max, Erfahrungswert ca.: 20%
 - Die Planung wird am Minimalwert ausgerichtet, aber so hinterlegt, dass die maximalen Aufwände in der Projektlaufzeit möglich zu erbringen sind.
- Weitere Möglichkeit
 - Min, Max und Normalwert schätzen, dann rechnen mit $(\text{Min} + \text{Max} + 4 \cdot \text{Norm}) / 6$

Beispiel

Beispiel
Expertenschätzung eines Web-Systems

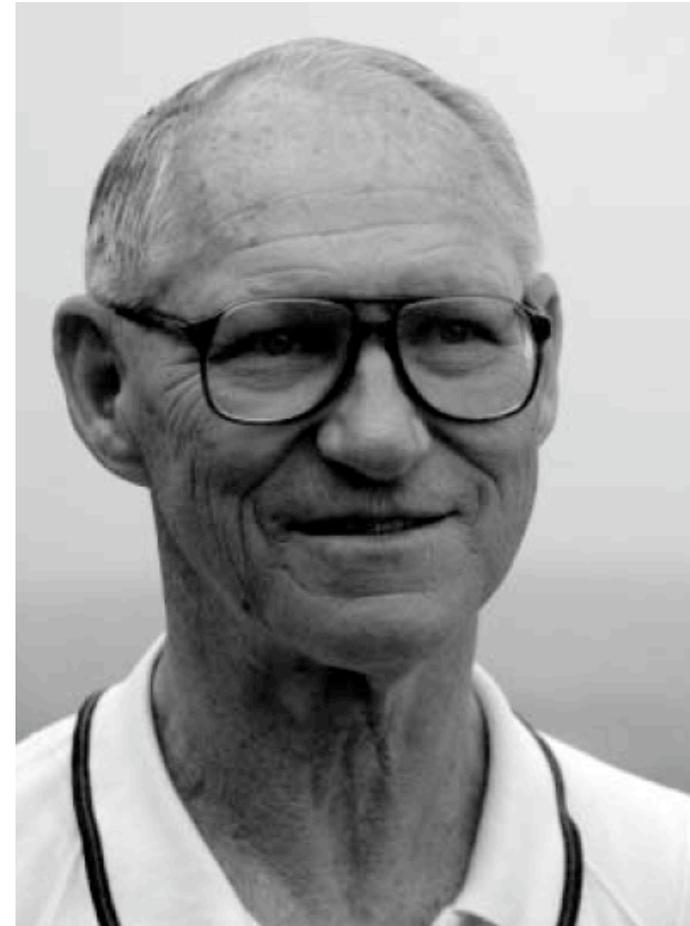


AGENDA

- Allgemeine Grundlagen zur Schätzung
- Function Point Verfahren
- Expertenschätzung, Delphi-Verfahren
- **CoCoMo Verfahren**

CoCoMo, CoCoMo II - Übersicht

- Entwickelt durch B. Boehm
- Idee:
 - Schätzung der Projektgröße in LOC (Lines of Code) bzw. KDSI (Kilo Delivered Software Instructions), d. h. ohne Kommentare.
 - Nach Verrechnung mit weiteren Kennzahlen wird der Gesamtaufwand E berechnet (MM_{DEV} Entwicklungsaufwand in PM) und die Projektlaufzeit (TDEV)
- Formel: $MM_{DEV} = a * KDSI^b * m(X)$



Differenzierung in Projektklasse

Projekt-Klassen

- Organic Mode
 - einfache SW-Projekte
 - eingespieltes Team, bekannte Umgebung, wenig Neuland
 - Größe <50 KDSI
 - Faktor b = 1,05
- Semi-detached Mode
 - mittelschwere Projekte
 - Größe <300 KDSI
 - Faktor b = 1,12
- Embedded Mode
 - schwierige Projekte
 - starker Kosten- Termindruck, viel Neuland
 - Größe: beliebig
 - Faktor b: 1,20
- a hängt von Projektklasse und Modellvariante (nächste Folie) ab und wird aus einer Tabelle entnommen

$$MM_{DEV} = a * KDSI^b * m(X)$$

Modellvarianten in CoCoMo

3 Modell-Varianten

- Basismodell
 - früh, zu Beginn eines Softwareprojekts
 - ganzheitliche Betrachtung
 - Ausgangspunkt für weitere Schätzungen
- Zwischenmodell
 - Berücksichtigung von 15 Einflussparametern
 - keine Differenzierung zwischen Phasen
- Detailmodell
 - Berücksichtigung von 15 Parametern
 - Abweichungen der Aufwände aus den einzelnen Phasen berücksichtigt

$$MM_{DEV} = a * KDSI^b * m(X)$$

Einflussfaktoren, Kostentreiber (1/2)

- Produkt
 - **RELY**: geforderte Zuverlässigkeit der Software
 - **DATA**: Größe der Datenbasis
 - **CPLX**: Komplexität des Produktes
- Computer
 - **TIME**: benötigte Rechenzeit
 - **STOR**: Nutzung des verfügbaren Speicherplatzes
 - **VIRT**: Änderungshäufigkeit der Systembasis
 - **TURN**: Bearbeitungszyklus

$$MM_{DEV} = a * KDSI^b * m(X)$$

$$m(x) = m(x1) * m(x2) * \dots * m(x15)$$

Einflussfaktoren, Kostentreiber (2/2)

- Projekt
 - **MODP**: Verwendung moderner Entwicklungsmethoden
 - **TOOL**: Verwendung von Tools
 - **SCED**: Anforderungen an die Entwicklungszeit
- Personal
 - **ACAP**: Analysefähigkeit der Projektmitarbeiter
 - **AEXP**: Erfahrung der Mitarbeiter im Arbeitsgebiet
 - **PCAP**: Programmierfähigkeit der Mitarbeiter
 - **VEXP**: Erfahrung der Mitarbeiter in der Systemumgebung
 - **LEXP**: Erfahrung der Mitarbeiter in der Programmiersprache

$$MM_{DEV} = a * KDSI^b * m(X)$$

$$m(x) = m(x1) * m(x2) * \dots * m(x15)$$

Berechnung des Gesamtaufwands und der Projektlaufzeit

TABLE V
COCOMO NOMINAL EFFORT AND SCHEDULE EQUATIONS

DEVELOPMENT MODE	NOMINAL EFFORT	SCHEDULE
Organic	$(MM)_{NOM} = 3.2(KDSI)^{1.05}$	$TDEV = 2.5(MM_{DEV})^{0.38}$
Semidetached	$(MM)_{NOM} = 3.0(KDSI)^{1.12}$	$TDEV = 2.5(MM_{DEV})^{0.35}$
Embedded	$(MM)_{NOM} = 2.8(KDSI)^{1.20}$	$TDEV = 2.5(MM_{DEV})^{0.32}$

(KDSI = thousands of delivered source instructions)

Und zum Schluss...

Keine Angst vor großen Zahlen.

- Software ist teuer.
- Eine ehrliche Schätzung ist die Grundlage für den Projekterfolg.

Zusammen. Für nachhaltigen Erfolg.

ZUSAMMEN. FÜR NACHHALTIGEN ERFOLG.

