Entwicklungsmethoden

Prof. Dr. Josef M. Joller
jjoller@hsr.ch

PLANEN UND SCHÄTZEN

Übersicht

Planen der Projekte

Abschätzung der Kosten / des Aufwandes

Planen und Schätzen

Vor der Entwicklung neuer SW oder einem IT Projekt muss das Projekt im Detail geplant werden.

Die Planung wird laufend ergänzt und verbessert

- die erste Grobplanung reicht nicht
- eine detaillertere Planung ist erst nach zusätzlichem Wissen möglich

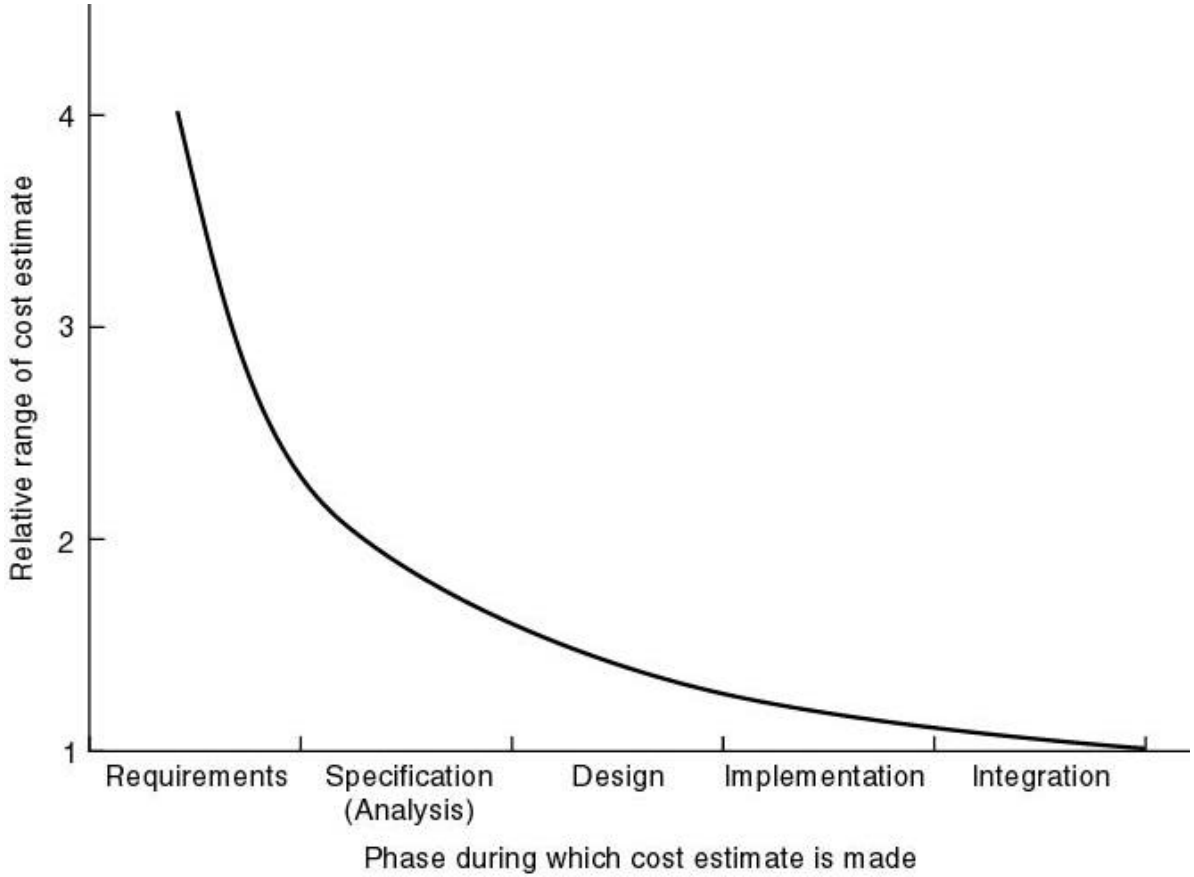
Aufwandschätzungen sind wichtig - aber wie?

Kostenschätzungen sind wichtig - aber wie?

- Kosten setzen sich aus unterschiedlichen Kostenarten zusammen

Aber SW / IT Projekte sind in der Regel zu komplex!

Planung des SW / IT Prozesses



Exakte Planung ist erst recht spät möglich

Beispiel

- Kostenschätzung: $1 Mio in der Anforderungsphase
  - echte Kosten werden im Bereich ($0.25M, $4M) liegen
- Kostenschätzung: $1 Mio in der Spezifikationsphase
  - echte Kosten werden im Bereich ($0.5M, $2M) liegen
- Kostenschätzung: $1 Mio am Ende der Spezifikationsphase
  - echte Kosten werden im Bereich ($0.67M, $1.5M) liegen

Das Modell ist alt (1976)

- die Aufwandschätzungsmethoden sind besser geworden
- aber die Kurve gilt immer noch (einfach nicht mehr so extrem)

Sackman (1968) zeigte, dass Programmierer von 1 bis 28 fache Produktivität zeigen können (Superprogrammer)

Die Studie berücksichtigte

- Produktgrösse
- Produkt-Ausführungszeit
- Entwicklungszeit
- Prgrammierzeit
- Testzeit

Zu beachten:

- was passiert, wenn ein Top Entwickler ausfällt?

Metriken betreffend Projektgrösse

Typische Metriken

- Lines of Code (LOC)
- Wissenschaftliche Ansätze (funktionieren in der Regel nicht)
- Function Points
- COCOMO und Varianten davon
- und viele weitere Techniken

Lines of Code

Lines of code (LOC) oder

Tausend angelieferte Programmzeilen (KDSI)

- allerdings ist der Programmcode nur ein kleiner Teil des gesamten SW Aufwandes
- je nach Programmiersprache resultieren unterschiedlich viele Programmzeilen
- LOC ist beim Einsatz spezieller Programmiersprachen (wie LISP) kaum ein sinnvolles Mass
- wie sollen die Programmzeilen gezählt werden?
  - Ausführbare Programmzeilen?
  - Datendefinitionen?
  - Kommentare?
  - Betriebsystemanweisungen?
  - Geänderte und gelöschte Zeilen?

Lines of Code

LOC kennt man erst am Ende des Projekts

Das Schätzen der LOC ist sehr wackelig

- am Anfang des Projekts muss die LOC des Endprodukts abgeschätzt werden
- LOC wird als Eingabe für viele Werkzeuge für die Aufwandschätzung eingesetzt

FFP Metrik

Gutes Modell bei mittel grossen Projekten

Parameter der FFP Methode / Metrik

- Files, Flows, Prozesse

Anzahl Files (Fi), Flows (Fl), Prozesse (Pr)

- Grösse (Size :S), Kosten (cost :C) werden abgeschätzt:
  - $S = Fi + Fl + Pr$
  - $C = b \times S$

Die Konstante b muss der Organisation angepasst werden

Dieses Modell funktioniert, aber es gibt keine veröffentlichten Daten für moderne Systeme mit Datenbanken, ...

Function Points

Basis: Inputs (Inp), Outputs (Out), Abfragen (inquiries :Inq), Stammdaten (master files ;Maf), Interfaces (Inf)

Die Anzahl Funktionspunkte berechnet sich als

- $FP = 4 \times Inp + 5 \times Out + 4 \times Inq + 10 \times Maf + 7 \times Inf$

Allerdings ist diese Formel zu einfach.

Function Points

1. **Klassifizieren Sie jede Komponente (Inp, Out, Inq, Maf, Inf) gemäss folgendem Schema als “simple”, “average” oder “complex”.**
   - Je nach Komplexität ergeben sich eine bestimmte Anzahl Punkte
   - die Summe bezeichnet man als UFP (unadjusted function points)

	Level of Complexity		
Component	**Simple**	**Average**	**Complex**
Input item	3	4	6
Output item	4	5	7
Inquiry	3	4	6
Master file	7	10	15
Interface	5	7	10

Function Points

2. **Dann berechnet man die Technische Komplexität (TCF)**
   - ein Wert 0 (“not present”) bis 5 (“strong influence throughout”) wird für jeden der14 Faktoren festgelegt
   - alle 14 Zahlen werden zusammengezählt $\Rightarrow$ degree of influence (DI)

     $$TCF = 0.65 + 0.01 \times DI$$
   - der Technical Complexity Factor (TCF) liegt zwischen 0.65 und 1.35

3. **Die Anzahl der Function Points (FP) ergibt sich zu**

   $$FP = UFP \times TCF$$

1. Data communication
2. Distributed data processing
3. Performance criteria
4. Heavily utilized hardware
5. High transaction rates
6. Online data entry
7. End-user efficiency
8. Online updating
9. Complex computations
10. Reusability
11. Ease of installation
12. Ease of operation
13. Portability
14. Maintainability

Analyse der Function Points Methode

Function Points sind in der Regel eine bessere Schätzmethode als KDSI (Kilo delivered source instructions)

Einige publizierten Ergebnisse:

“Errors in excess of 800% counting KDSI, but only 200% in counting function points” (Jones, 1987)

Aber auch bei dieser Methode sind einige Parameter schlecht mess/abschätzbar.

Techniken für die Kostenschätzung

Expertenmeinungen sind das Beste!

Der Experte vergleicht das zu erstellende mit einem erstellten Produkt

- allerdings kann er sich auch täuschen
- der Experte hat vielleicht falsche Projektdaten
- menschliche Experten sind voreingenommen
- mehrere Experten liefern in der Regel gute Resultate

Bottom-up Approach

- zerlegen Sie das Projekt in kleinere
- kleinere Projekte lassen sich besser abschätzen