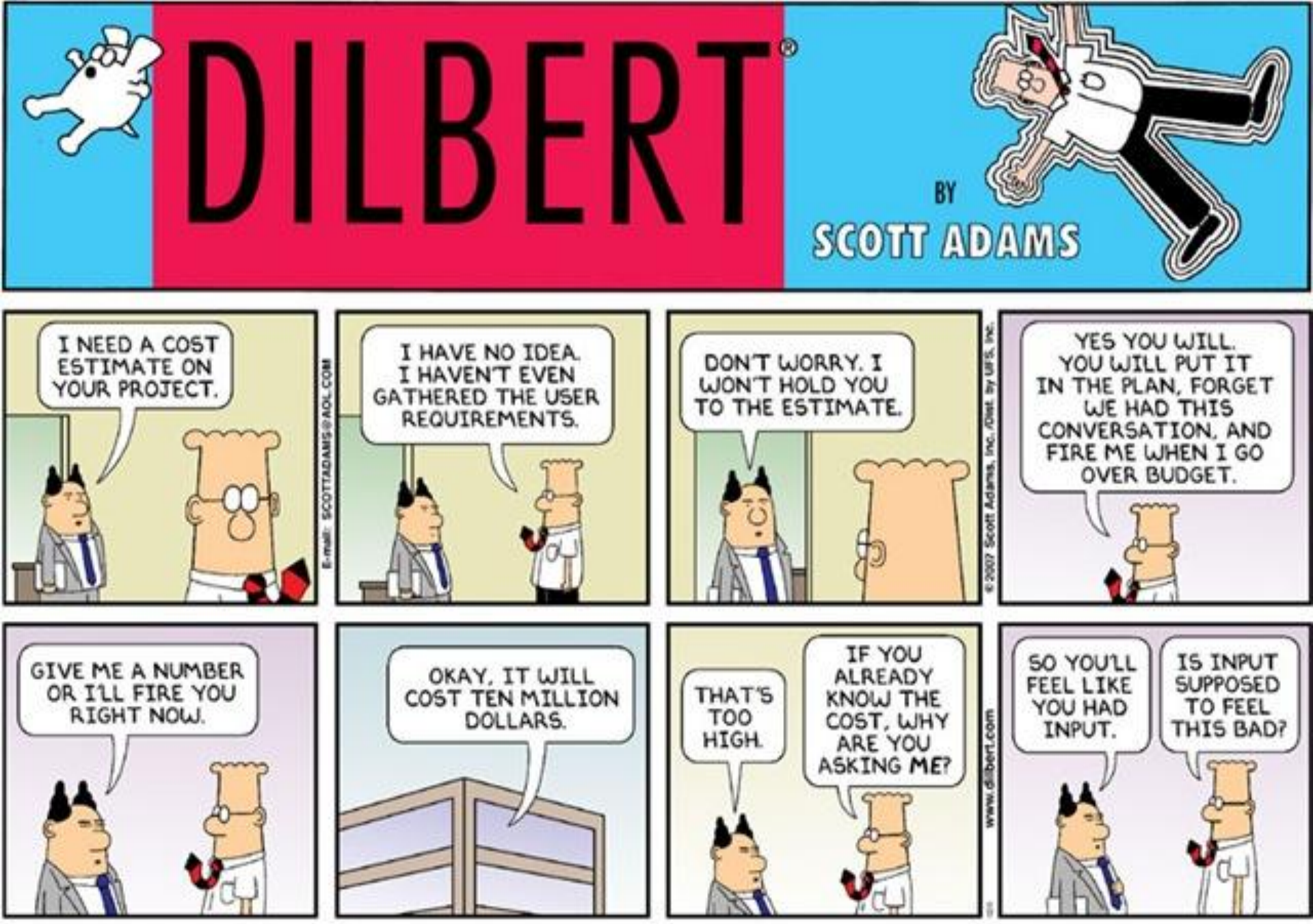
Project Plan

Cost Estimation

I Dewa Md. Adi Baskara Joni S.Kom., M.Kom



Why ?

Hubungan antara konsep umum dengan teknik analisis ekonomi dalam Rekayasa Perangkat Lunak

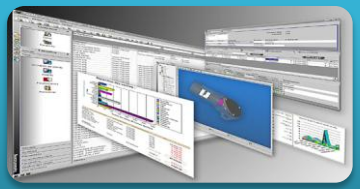
Teknik yang menyediakan bagian penting dari dasar untuk memanajemen Perangkat Lunak

COST

Prengkat keras (fasilitas, peralatan, dll)

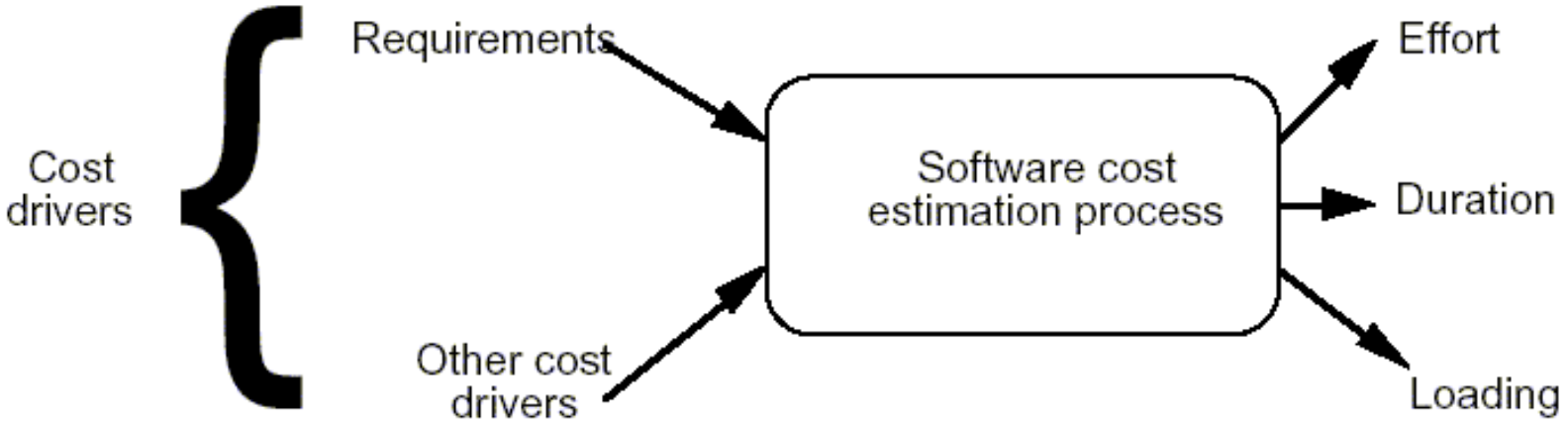
Pelatihan (metode, peralatan, dll)

Perjalanan



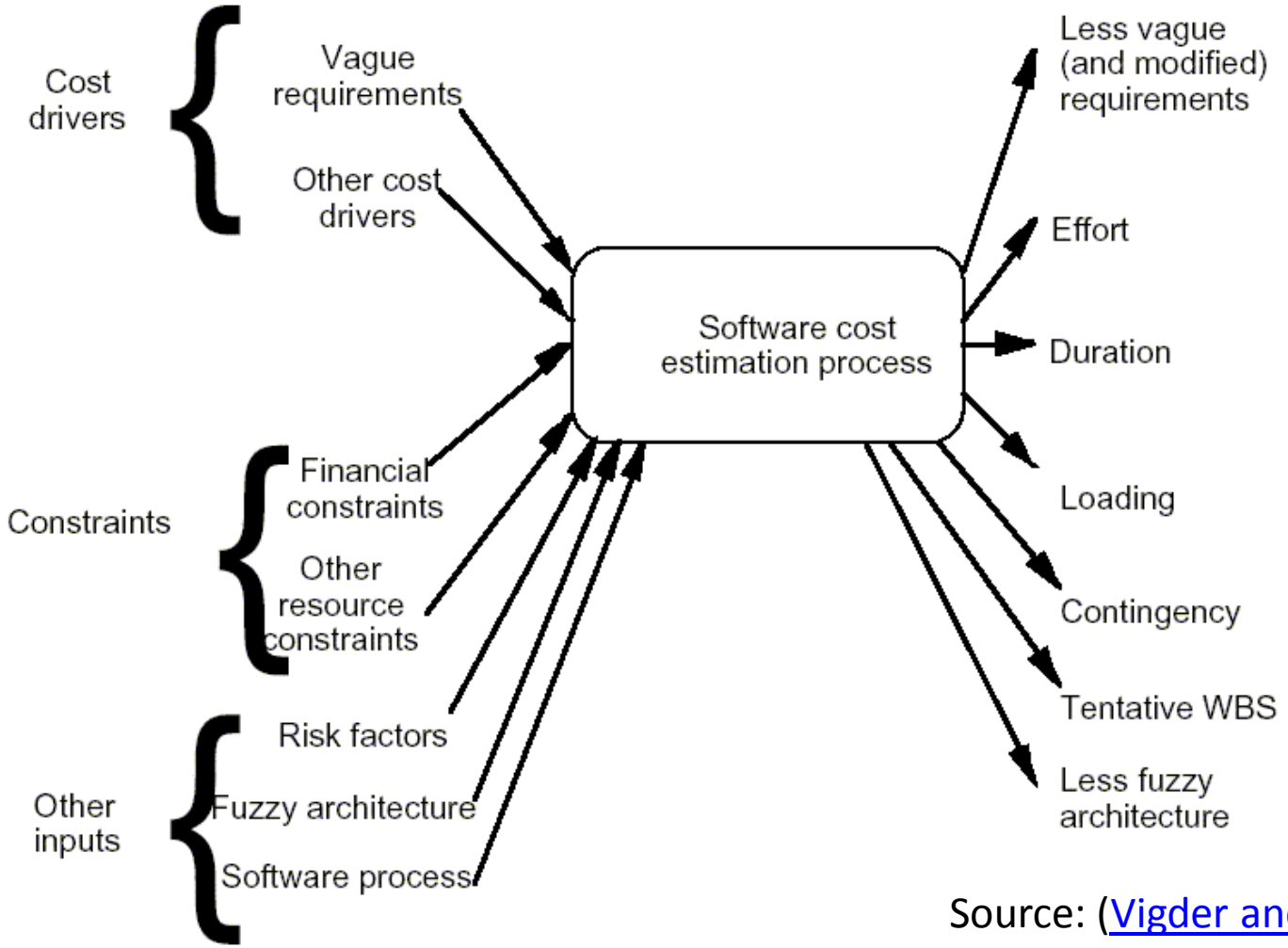
Pengembangan perangkat lunak - (s/w *tools*, *code generators*, dll)

Software Estimation Process



Source: (Vigder and Kark, 1994)

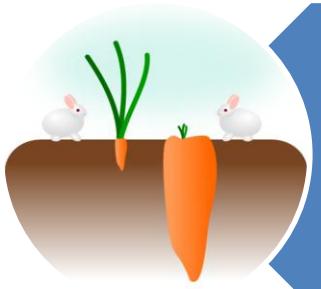
Actual View



Source: (Vigder and Kark, 1994)

Poor Cost Estimation

Overruns → Cancel projects



Underestimates → Penambahan waktu (tanpa biaya tambahan)

Genting bagi PM → Perencanaan sumber daya

Metode-Metode

- Algorithmic (Parametric) model
- Expert Judgment (Expertise Based)
- Top - Down
- Bottom - Up
- Estimation by Analogy
- Price to Win Estimation

Algorithmic model

Perhitungan matematis untuk estimasi biaya pengembangan *software*

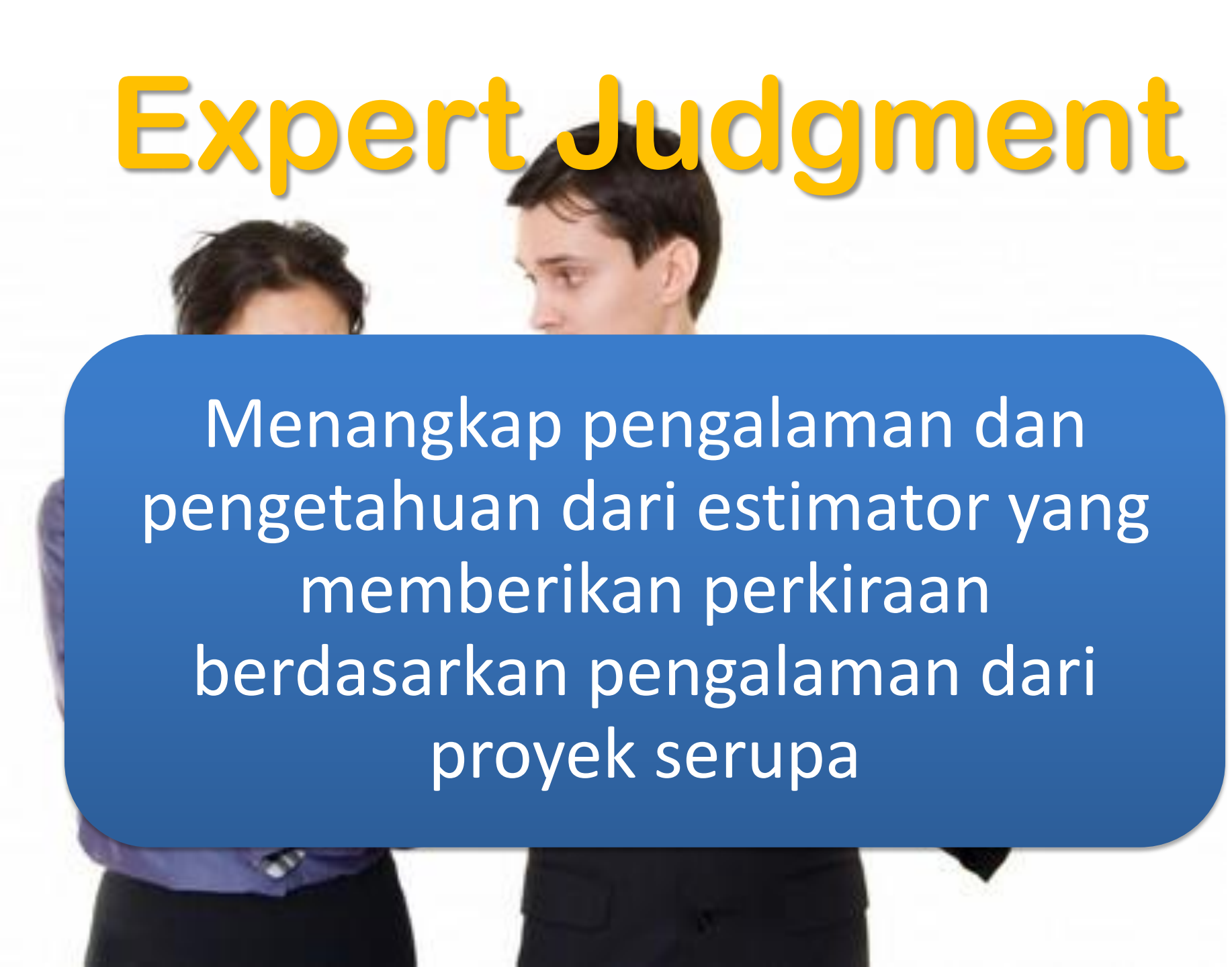
Kelebihan

- Menghasilkan estimasi yg berulang
- Mudah untuk mengubah input data
- Mudah untuk mengubah *formula*
- Untuk yang berpengalaman

Kekurangan

- Tidak dapat menangani kondisi *exceptional*
- Beberapa faktor tidak dapat dikuantifikasi
- *Sometimes algorithms may be proprietary*

Expert Judgment



Menangkap pengalaman dan pengetahuan dari estimator yang memberikan perkiraan berdasarkan pengalaman dari proyek serupa

Expertise

Experts Membuat Prediksi

- Optimistic → x
- Realistic → y
- Pessimistic → z

$$E = (x + 4y + z) / 6$$

Expert Judgment

Kelebihan

- Berguna utk yg tdk dapat dikuantifikasi (*Empirical* data)
- Bisa memprediksi dari pengalaman proyek yg lama dgn *requirements* proyek yg akan datang
- Bisa memprediksi dampak teknologi, aplikasi

Kekurangan

- Estimasi adalah pendapat dari ahli
- Susah untuk mendokumentasikan faktor-faktor yg digunakan ahli

Top - Down

Dari *global view* produk yang kemudian dibagi menjadi beberapa komponen

Kelebihan

- Membutuhkan sedikit detil
- Lebih cepat dan mudah diimplementasikan
- Berfokus pada aktifitas di tingkat sistem

Kekurangan

- Cenderung mengabaikan komponen tingkat rendah
- Tidak terdapat dasar terperinci

Bottom - Up

Mengumpulkan semua komponen biaya pengembangan yang kemudian digabung untuk memperoleh estimasi biaya akhir proyek

Kelebihan

- Lebih stabil
- Lebih detil
- Setiap tim ikut memberikan estimasi

Kekurangan

- Mengabaikan biaya ditingkat sistem
- Lebih memakan waktu

Estimasi Analogi

Memanfaatkan data aktual dari proyek yg sebelumnya dan dibandingkan dgn proyek yang diusulkan dalam domain aplikasi yg sama utk estimasi biaya

Kelebihan	• Berdasarkan data aktual proyek
Kekurangan	• Mustahil jika proyek di masa lalu

Price to Win Estimation

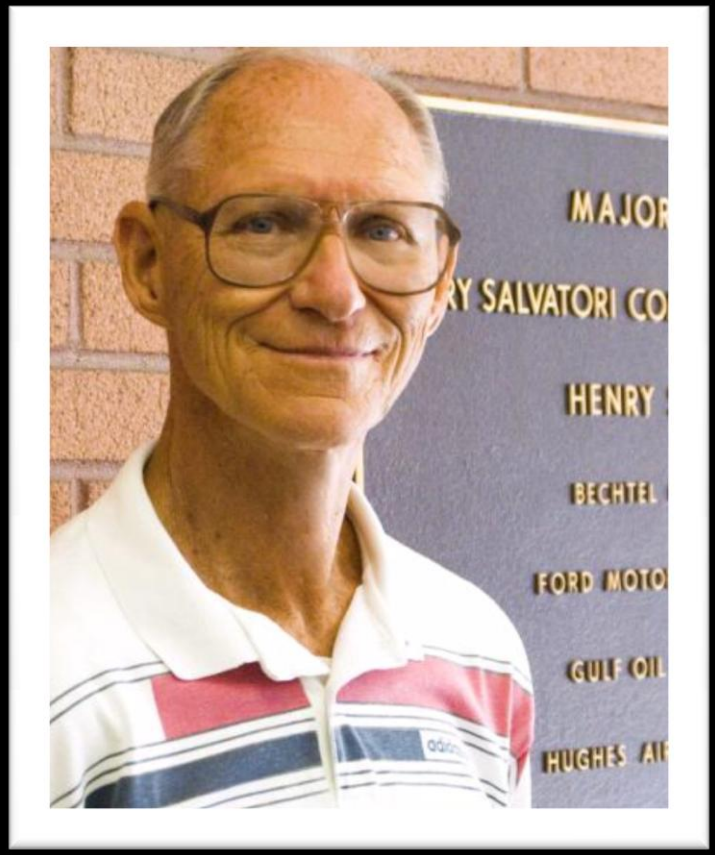
Estimasinya adalah harga yg diperlukan utk memenangkan kontrak atau proyek

Kelebihan	• Menghasilkan kontrak
Kekurangan	• Waktu dan dana abis sebelum pekerjaan selesai

COnstructive **CO**st **MO**del

COCOMO

- Diciptakan Boehm ‘80s
- Perkiraan ***effort*** dan ***schedule*** untuk pengembangan p/l
- Berdasarkan input yg berkaitan dgn ukuran p/l dan jumlah komponen biaya yg mempengaruhi produktivitas



Barry W. Boehm (born 1935)

Berdasarkan SLOC

SLOC = “Source Lines Of Code”

Hanya jumlah baris kode yang menjadi bagian dari produk yg disertakan, tidak termasuk p/l pendukung

Hanya baris kode yg dibuat *team member*, tidak termasuk kode yg dihasilkan dari *generate* aplikasi

Berdasarkan SLOC

Satu SLOC adalah satu baris kode secara logis (satu if-then-else adalah satu baris kode)

Deklarasi dihitung satu SLOC

Comments tidak dihitung sebagai SLOC

COCOMO Models

Basic Model

- Diterapkan diawal pengembangan proyek. Perkiraan awal yg akan disempurnakan dgn model lain

Intermediate Model

- Akan digunakan setelah memiliki *requirements* yg lebih rinci

Detailed Model

- Setelah desain sistem selesai, model ini digunakan utk memperbaiki estimasi biaya

COCOMO Modes

Organic Mode

- Dibangun dengan lingkungan yg familiar
- Mirip dgn proyek yg dikembangkan sebelumnya
- Membutuhkan sedikit inovasi

Semidetached Mode

- Menengah: diantara *organic* dan *embedded*

Embedded Mode

- Ketat, kendala tidak fleksibel
- Produk yg membutuhkan inovasi tinggi

Equations

- Equation 1 $E = a(KDSI)^b * EAF$
- Equation 2 $D = c(E)d$
- Equation 3 $N = E/D$

- Dimana:
  - E adalah *effort* dalam person-months
  - EAF adalah *effort adjustment factor*
  - D adalah *schedule time*
  - KDSI adalah jumlah baris kode (dalam ribuan)
  - N jumlah personil yang dibutuhkan
  - a, b, c, dan d semua konstanta berdasarkan mode

Equations Used

Mode	*Effort*	*Schedule*
Organic	$E=2.4*(KDSI)^{1.05}$	$TDEV=2.5*(E)^{0.38}$
Semidetached	$E=3.0*(KDSI)^{1.12}$	$TDEV=2.5*(E)^{0.35}$
Embedded	$E=3.6*(KDSI)^{1.20}$	$TDEV=2.5*(E)^{0.32}$

Latihan 1

Ceritanya, dalam suatu pengembangan perangkat lunak telah diidentifikasi 50.000 baris kode sebagai suatu sistem yg utuh. Sistem diklasifikasikan dalam *semidetached mode.*

- Tugas:
  - Temukan $\text{EFFORT}_{\text{crude}}$
  - Hitung total durasi yg dibutuhkan utk proyek
  - Hitung total team member yg dibutuhkan

Intermediate COCOMO

- $EFFORT_{improved}$
- Estimasi menggunakan limabelas variabel biaya selain variabel ukuran yg digunakan dalam *basic model*
- Faktor-faktor dibagi dalam 4 kelas:
  - *Product*
  - *Computer*
  - *Personnel*
  - *Project*

FORMULAE

$EFFORT_{improved} = EFFORT_{crude} * m1 * m2 ..... m_N$

Dimana m1, m2 ..... m_N adalah *multipliers* yg merepresentasikan nilai dari faktor produktivitas

Intermediate COCOMO

Cost drivers	Rating					
	Very low	Low	Normal	High	Very high	Extra high
Product attributes						
Reliability required	0.75	0.88	1.00	1.15	1.40	
Database size		0.94	1.00	1.08	1.16	
Product complexity	0.70	0.85	1.00	1.15	1.30	1.65
Computer attributes						
Execution time constraints			1.00	1.11	1.30	1.66
Main storage constraints			1.00	1.06	1.21	1.56
Virtual machine volatility		0.87	1.00	1.15	1.30	
Computer turnaround time		0.87	1.00	1.07	1.15	
Personnel attributes						
Analyst capability	1.46	1.19	1.00	0.86	0.71	
Applications experience	1.29	1.13	1.00	0.91	0.82	
Programmer capability	1.42	1.17	1.00	0.86	0.70	
Virtual machine experience	1.21	1.10	1.00	0.90		
Prog. language experience	1.14	1.07	1.00	0.95		
Project attributes						
Use of modern prog. techniques	1.24	1.10	1.00	0.91	0.82	
Use of software tools	1.24	1.10	1.00	0.91	0.83	
Required development schedule	1.23	1.08	1.00	1.04	1.10	

Latihan 2

Setelah mengetahui $Effort_{crude}$ (152 programmer-months), diketahui:

a) *product complexity*, cukup banyak permintaan

b) *database size*, meningkat drastis

Tugas:

- Temukan $EFFORT_{improved}$
- Hitung total durasi yg dibutuhkan utk proyek
- Hitung total team member yg dibutuhkan


THANK
YOU
GRACIAS
ARIGATO
SHUKURIA
GOZAIMASHITA
EFCHARISTO
JUSPAXAR
DANKSCHEEN
TASHAKKUR ATU
YAQHANYELAY
SUKSAMA
EKHMET
MEHRBANI
GRAZIE
PALDIES
KOMAPSUMNIDA
MAAKE
BOLZÏN
MERCI
BÏYAN
SHUKRIA
TINGKI


Thank You!