

# Odhadování ceny SW

# COCOMO

Zdroje empirických dat:

- větší počet předchozích komplexních projektů
  - aplikace odlišného druhu s odlišnými cíly
  - odlišná vývojová prostředí
- rozhovory s více manažery

Parametry modelu byly nastaveny podle získaných empirických dat.

# COCOMO

Výpočet E(KSLOC) a T(KSLOC):

$$E = a \cdot F \cdot (KSLOC)^b$$

$$T = c \cdot E^d$$

*a, b, c, d*: parametry volené podle úrovně modelu a vývojového módu

# Hodnoty parametrů

## Empirické hodnoty parametrů pro výpočet E(KSLOC) a T(KSLOC)

- tabulky hodnot a,b,c,d pro všechny kombinace úrovně modelu/vývojové módy
- příklady:
  - základní model, bezprostřední mód:  
 $a=3.0, b=1.12, c=2.5, d=0.35$
  - střední model, vázaný mód:  
 $a=2.8.F_c, b=1.2, c=2.5, d=0.32$

# Hodnoty parametrů

Empirické hodnoty parametrů pro výpočet  $E(KSLOC)$  a  $T(KSLOC)$

Intervaly hodnot parametrů:

- $a \in [2.4, 3.6]$  pro základní model
- $a \in [2.8 F_c, 3.2 F_c]$  pro střední a pokročilý model
- $b \in [1.05, 1.20]$
- $c = 2.5$  ve všech případech
- $d \in [0.32, 0.38]$

# Korekční faktor

Atributy, které mají vliv na korekční faktor  $F_c$ :

- atributy SW produktu
- HW atributy
- atributy vývojového týmu
- atributy projektu

# Původní COCOMO

COCOMO lze také použít pro odhad nákladů při modifikaci existujících aplikací

$$ESLOC = ASLOC \cdot (0,4 DM + 0,3 CM + 0,3 IM) / 100$$

ESLOC - ekvivalentní počet SLOC

ASLOC - odhadnutý počet modifikovaných SLOC

DM - procento modifikace v návrhu

CM - procento modifikace v kódu

IM - integrační úsilí (procento původní práce)

# Potřeba změnit COCOMO 81

- nové softwarové procesy
- nové jevy měření velikostí
- nové jevy znovupoužití software
- potřeba rozhodování na základě neúplné informace

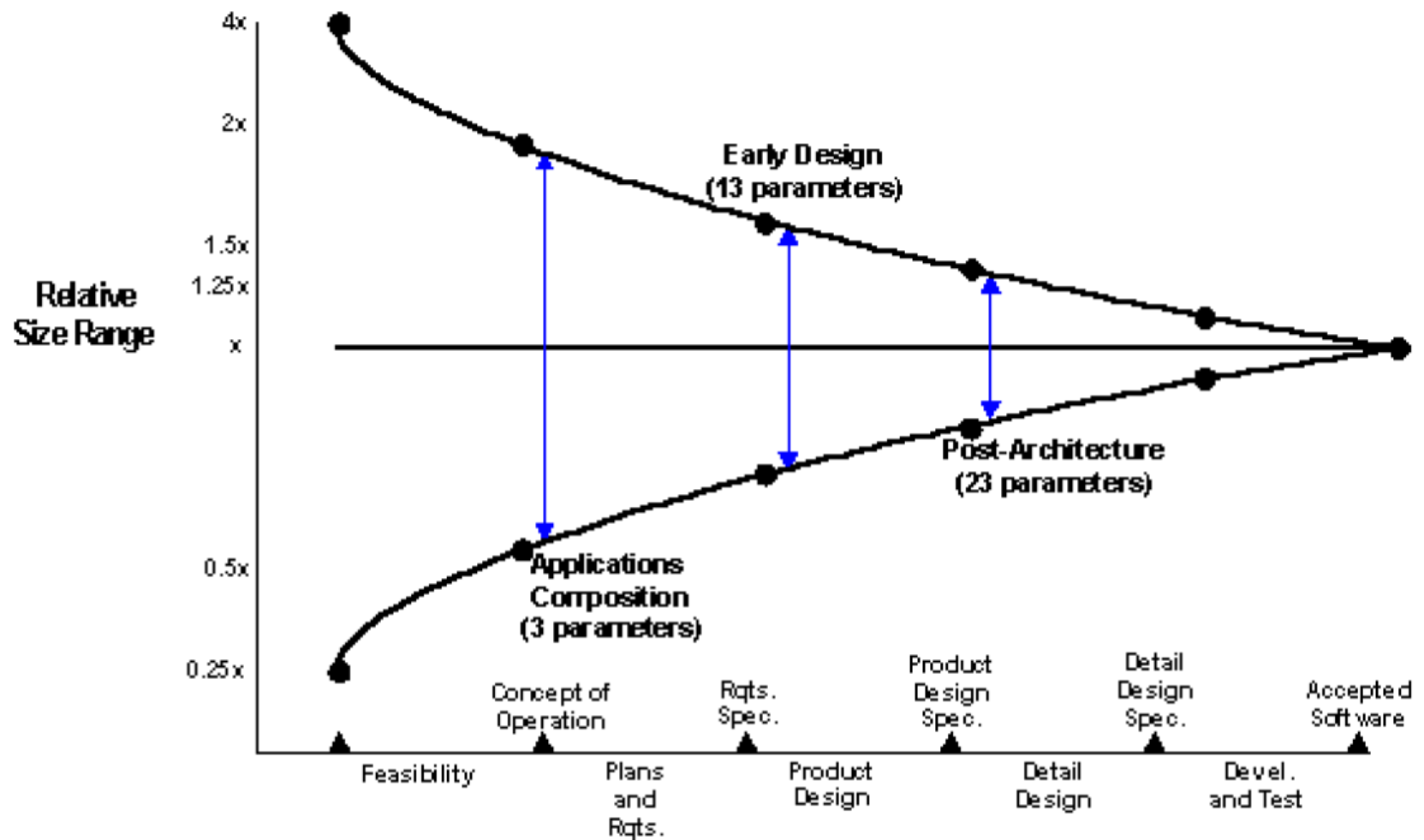


# COCOMO II

## 3 různé modely

- **APM (Application Composition Model)**  
pro projekty s použitím moderních nástrojů a GUI
- **EDM (Early Design Model)**  
pro hrubé odhady v úvodních etapách, kdy se architektura vyvíjí
- **PAM (Post Architecture Model)**  
pro odhady poté, co byla specifikována architektura

# COCOMO II



# COCOMO II

*Úsilí = (multiplikátory okolí)[velikost]<sup>(faktory procesu)</sup>*

*Plán = (multiplikátor)[Úsilí] (faktory procesu)*

# COCOMO II

Odhady práce a velikosti při modifikaci existujících aplikací

$$ESLOC = ASLOC \cdot (AA + SU + 0,4 DM + 0,3 CM + 0,3 IM) / 100$$

- ESLOC, ASLOC, DM, CM, IM - stejné jako dříve
- AA (Assessment and Assimilation) - práce potřebná pro určení, zda a v jakém rozsahu může být existující modul použit beze změn
- SU (pochopení SW) = čitelnost a „uchopení“

# Funkční body

# Funkční body

Výroba software je v podstatě výrobní proces, který vyžaduje lidskou práci.

Takže je to prosté, stačí určit:

- jednotku výroby
- cenu práce za výrobu této jednotky

# Funkční body

Funkční body = normalizovaná metrika softwarového projektu

- Měří aplikační oblast, nezkoumá technickou oblast
- Měří aplikační funkce a data, neměří kód

International Function Point Users Group - [www.ifpug.org](http://www.ifpug.org)

# Typy funkčních bodů

Funkční body vztažené k transakčním funkcím:

- Externí vstupy (EI - External Inputs)
- Externí výstupy (EO - External Outputs)
- Externí dotazy (EQ - External Enquiry)

Funkční body vztažené k datovým funkcím:

- Vnitřní logické soubory (ILF - Internal Logical Files)
- Soubory vnějšího rozhraní (EIF - External Interface Files)



# Výpočet funkčních bodů

Před výpočtem musíme EI, EO, EQ, ILF, EIF roztřídit do skupin podle vah.

Váhy	nízká	průměrná	vysoká	celkem
EI	___ x 3 +	___ x 4 +	___ x 6 =	_____
EO	___ x 4 +	___ x 5 +	___ x 7 =	_____
EQ	___ x 3 +	___ x 4 +	___ x 6 =	_____
ILF	___ x 7 +	___ x 10 +	___ x 15 =	_____
EIF	___ x 5 +	___ x 7 +	___ x 10 =	_____

Neupravené funkční body celkem \_\_\_\_\_

## Funkční body - Matice složitosti

FTRs	1-4 DETs	5-15 DETs	16+DETs
0-1	nízká	nízká	průměrná
2-3	nízká	průměrná	vysoká
4+	průměrná	vysoká	vysoká

- FTR = File Types (User Data Groups) Referenced
- DET = Data Element Type (Attribute)
- RET = Record Element Type (User View)

# Funkční body

14 charakteristik hodnocených podle stupně vlivu na aplikaci

Každý faktor je hodnocený ve stupnici 0 – 5 takto:

- 0 = bez vlivu
- 1 = náhodný
- 2 = mírný
- 3 = průměrný
- 4 = významný
- 5 = podstatný

# Počet funkčních bodů

Počet funkčních bodů

=

[0.65 + (0.01 x součet hodnocení charakteristik  
systému)]

x

[počet nepřizpůsobených funkčních bodů]

# Nové a upravované projekty

Type of Project	Project Function Points	Application Function Points
		Installed Function Pts. (IFP)
Development Project	Project FP = New (Added) FP + Conversion FP	Application FP = New (Added) FP
Enhancement Project	Project FP = Added FP + Changed FP + Deleted FP + Conversion FP	Application FP = Original FP - Deleted FP + Added FP + $\Delta$ Changed FP

1 Funkční bod = X příkazů

- 320 - základní assembler
- 213 - makro assembler
- 128 - C
- 107 - COBOL
- 107 - FORTRAN
- 80 - PL/I
- 71 - Ada 83
- 64 - C++
- 54 - Ada 95
- 32 - Visual BASIC
- 22 - Smalltalk
- 16 - PowerBuilder
- 13 - SQL

## Produktivita (FP a člověkoměsíci)

Nezkušený tým, nestrukturované metody, běžné nástroje, jazyky na nízké úrovni	2.50
Nezkušený tým, nestrukturované metody, nástroje CASE, jazyky na nízké úrovni	3.50
Nezkušený tým, strukturované metody, běžné nástroje, jazyky na nízké úrovni	4.00
Zkušený tým, nestrukturované metody, běžné metody, jazyky na nízké úrovni	4.50
Nezkušený tým, nestrukturované metody, běžné nástroje, jazyky na vysoké úrovni	5.00
Nezkušený tým, strukturované metody, nástroje CASE, jazyky na nízké úrovni	6.00
Nezkušený tým, nestrukturované metody, nástroje CASE, jazyky na vysoké úrovni	7.00
Zkušený tým, nestrukturované metody, nástroje CASE, jazyky na nízké úrovni	8.00
Nezkušený tým, strukturované metody, běžné nástroje, jazyky na vysoké úrovni	8.50
Zkušený tým, strukturované metody, běžné nástroje, jazyky na nízké úrovni	9.00
Zkušený tým, nestrukturované metody, běžné nástroje, jazyky na vysoké úrovni	10.00
Zkušený tým, strukturované metody, nástroje CASE, jazyky na nízké úrovni	12.00
Nezkušený tým, strukturované metody, nástroje CASE, jazyky na vysoké úrovni	14.00
Zkušený tým, nestrukturované metody, nástroje CASE, jazyky na vysoké úrovni	18.00
Zkušený tým, strukturované metody, běžné nástroje, jazyky na vysoké úrovni	25.00
Zkušený tým, strukturované metody, nástroje CASE, jazyky na vysoké úrovni	40.00