

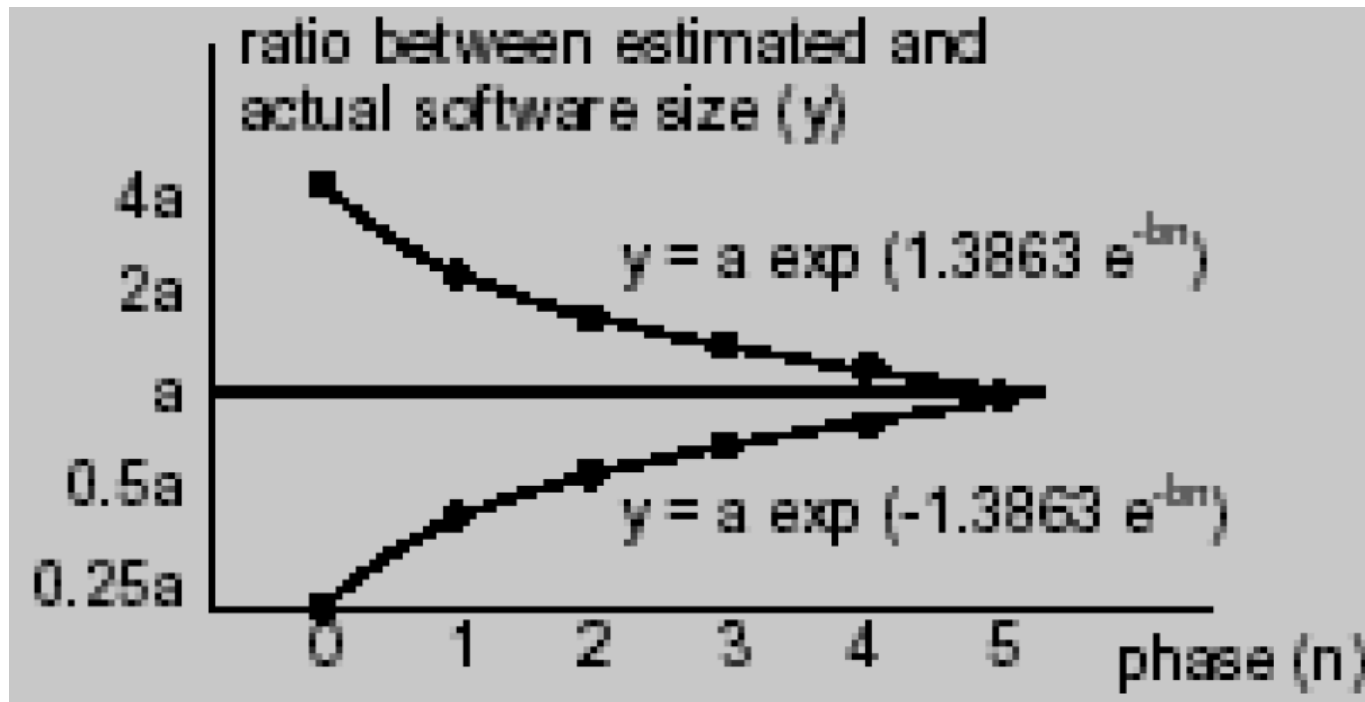
Odhadování ceny SW - původní COCOMO

Idea



- Cena vývoje aplikace přímo závisí na velikosti SW.
- Přesnost odhadu velikosti SW závisí na etapě vývoje.
 - V pozdějších etapách je odhad přesnější.
 - Přesnost odhadu se může lišit až čtyřikrát (4:1) oběma směry.

Idea



COCOMO



- COCOMO – Constructive Cost Model
- B.Boehm (IBM), 1981, 1984, 1995
- #SLOC jako hlavní indikátor velikosti a složitosti SW
- Empirické vztahy pro vyjádření $E=E(KSLOC)$ a $T=T(KSLOC)$
 - E – effort (práce, člověk-měsíc)
 - T – doba vývoje (měsíc)

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.

Původní COCOMO



3 úrovně detailu:

- **Základní model:** hrubý odhad $E(KSLOC)$ a $T(KSLOC)$ založen na odhadu $KSLOC$.
- **Střední model:** vliv jiných faktorů na $E(KSLOC)$ a $T(KSLOC)$.
- **Pokročilý model:** bere v úvahu vlivy vývojové etapy, ve které se projekt nachází.

Původní COCOMO



3 vývojové módy:

- **Organický mód** – jednodušší, dobře řešitelné projekty, zpravidla menšího rozsahu
- **Bezprostřední mód** – středně obtížné projekty
- **Vázaný mód** – rozsáhlé projekty s vysokými nároky na řízení

Organický mód



- malé projekty (SW < 50 KSLOC)
- úplné porozumění požadavkům
- malá omezení, volnost při návrhu rozhraní
- velké zkušenosti při zpracování podobných projektů
- malá závislost na speciálním HW
- minimální potřeba nových algoritmů a architektur
- minimum požadavků na zkrácení termínu dodání

Organický mód - příklady



- vědecké aplikace
- jednoduché obchodní modely a aplikace
- jednoduchá skladová aplikace
- jednoduchý systém pro řízení výroby

Bezprostřední mód



- projekty střední velikosti (SW < 300 KSLOC)
- dobré pochopení požadavků
- zřetelná omezení pro uživatelské rozhraní
- nezanedbatelná zkušenost při práci na podobných projektech
- střední závislost na speciálním HW
- střední potřeba nových algoritmů a architektur
- nezanedbatelný podnět pro ukončení před plánovaným termínem

Bezprostřední mód - příklady



- transakční zpracování
- nový operační systém a překladač
- skladová aplikace střední složitosti
- systém pro řízení výroby střední složitosti

Vázaný mód



- SW všech velikostí
- jen hrubá představa o cílech projektu
- těsná omezení, striktní požadavky na rozhraní
- nezanedbatelná zkušenost při práci na podobných projektech
- vysoká závislost na speciálním HW
- extrémní požadavky na nové algoritmy a architektury
- vysoké podněty pro dokončení před termínem

Vázaný mód - příklady



- složité transakční zpracování
- ambiciózní a složitý operační systém
- RT aplikace
- složité povelové a řídicí systémy



Výpočet E(KSLOC) a T(KSLOC):

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

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

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



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 $E(\text{KSLOC})$ a $T(\text{KSLOC})$

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
- $d \in [0.32, 0.38]$



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

- V základním modelu mají všechny parametry konstantní hodnoty.
- Ve středním a pokročilém modelu ve všech vývojových módech a závisí na F_c , ostatní parametry jsou konstantní. Korekční faktor F_c je součinem hodnot 15 atributů (cost drivers) specifických pro vývojový proces.

Korekční faktor



Atributy, které mají vliv na korekční faktor F_c :

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

Korekční faktor



Atributy, které mají vliv na korekční faktor F_c , mohou nabývat 6 možných hodnot ve stupnici:

- velmi nízký
- nízký
- normální
- velký
- velmi velký
- extrémně velký

Hodnotám odpovídají diskrétní numerické hodnoty.

Atributy SW produktu



- RELY - požadovaná spolehlivost (0.75 - 1.40)
(velmi nízká: 0.75, extrémně velká: 1.40)
- DATA - velikost databáze (0.94 - 1.16)
- CPLX - složitost produktu (0.70 - 1.65)

Atributy SW produktu



Příklady RELY a DATA atributů:

- vysoké finanční riziko \Rightarrow RELY = velký
- $(\text{DB byty})/(\text{program SLOC}) \geq 1000 \Rightarrow$
DATA = velmi velký

Atributy SW produktu



Příklady CPLX hodnot řídicí operace:

Rekurzivní procedury, zpracování přerušení s pevnou prioritou

- výpočty: obtížné, ale strukturované numerické operace, parciální diferenciální rovnice
- I/O operace: detekce přerušení, zpracování a maskování, síťové komunikace
- zpracování dat: složité soubory, vyhledávání

CMPLX = velmi velká

HW atributy



- TIME - omezení času výpočtu (1.00 - 1.66)
- STOR - využití paměti/disku (1.00 - 1.56)
- VIRT - spolehlivost (zranitelnost) virtuálních strojů, tj. HW + DBMS + OS + ... (0.87 - 1.30)
- TURN - doba obrátky (0.87 - 1.15)



Příklady:

- omezení na časy výpočtu větší než 95% \Rightarrow TIME = extrémně velký
- využití paměti/disku $< 50\%$ \Rightarrow STOR = normální
- velké změny ve virtuálních strojích každé 2 týdny, malé změny každé 2 dny \Rightarrow VIRT = velmi velký
- interaktivní práce \Rightarrow TURN = nízký

Atributy vývojového týmu



- ACAP - schopnost analytická (1.46 - 0.71)
- PCAP - schopnost programátorská (1.42 - 0.70)
- AEXP - zkušenost s podobnými aplikacemi (1.29 - 0.82)
- VEXP - zkušenost se specifickým „virtuálním strojem“ (1.21 - 0.90)
- LEXP - zkušenost se specifickým programovacím jazykem (1.14 - 0.95)

Atributy vývojového týmu



Příklady

- 3-letá zkušenost s podobnými aplikacemi => AEXP = normální
- programátorská schopnost v týmové práci = 35 (stupnice 0..100) => PCAP = nízký
- 3-letá zkušenost se specifickým programovacím jazykem => LEXP = velký

Atributy projektu



- MODP - použití moderních programovacích technik (1.24 - 0.82)
- TOOL - použití SW nástrojů (1.24 - 0.83)
- SCED - přesné plánování (1.23 - 1.10)

Atributy projektu



Příklady:

- občasné použití moderních programovacích technik => MODP = normální
- stálé používání SW nástrojů pro programování => TOOL = velký
- použití SW nástrojů také pro návrh, testování a analýzu požadavků => TOOL = velmi velký
- striktní plány = 75% nominálních hodnot => SCED = velmi nízký

Cenové koeficienty



#		Cost Drivers	VL	L	NOM	HGH	VH	EH
1	PROD	Reliability	0,75	0,88	1	1,15	1,40	X
2	PROD	Database size	X	0,94	1	1,08	1,16	X
3	PROD	Product complexity	0,70	0,85	1	1,15	1,30	1,65
4	PFRM	Execution time constraints	X	X	1	1,11	1,30	1,66
5	PFRM	Main storage constraints	X	X	1	1,06	1,21	1,56
6	PFRM	Virtual machine volatility	X	0,87	1	1,15	1,30	X
7	PFRM	Computer turnaround time	X	0,87	1	1,07	1,15	X
8	PERS	Analyst capability	1,46	1,19	1	0,86	0,71	X
9	PERS	Applications experience	1,29	1,13	1	0,91	0,82	X
10	PERS	Programmer capability	1,42	1,17	1	0,86	0,70	X
11	PERS	Virtual machine experience	1,21	1,10	1	0,90	X	X
12	PERS	Programming language exp.	1,14	1,07	1	0,95	X	X
13	TOOL	Use of modern progr. Techn.	1,24	1,10	1	0,91	0,82	X
14	TOOL	Use of software tools	1,24	1,10	1	0,91	0,83	X
15	TOOL	Req. development schedule	1,23	1,08	1	1,04	1,10	X

Kroky při použití COCOMO



Při aplikaci modelu COCOMO se postupně provádějí následující kroky:

- Určení nominálního úsilí E_n
- Určení korekčního faktoru F_c
- Určení aktuálního (zpřesněného) úsilí E
- Určení doby vývoje T a dalších faktorů relevantních pro projekt

Kroky při použití COCOMO



Krok 1 - Určení nominálního úsilí E_n

- definujte (odhadni) úroveň modelu a vývojový mód
- nastavte odpovídající hodnoty a a b podle tabulky
- vypočtete E_n

Kroky při použití COCOMO



Krok 2 - Určení korekčního faktoru F_c

- na základní úrovni není třeba řešit
- určete popisné hodnoty pro každý z 15 atributů
- převed'te na numerické hodnoty podle tabulky

$$F_C = \prod_{i=1}^{15} F_i$$

Kroky při použití COCOMO



Krok 3 - Určení aktuálního (zpřesněného) úsilí E

- na základní úrovni $E = E_n$
- $E [\text{člověk-měsíc}] = F_c \cdot E_n$
- F_c vyjadřuje nárůst pracnosti E_n podle vlivu a významu jednotlivých atributů vývojového procesu

Kroky při použití COCOMO



Krok 4 - Určení doby vývoje T a dalších faktorů relevantních pro projekt

- nastavte odpovídající hodnoty c a d podle tabulky
- $T[\text{měsíc}] = c \cdot E^d$
- COCOMO také umožňuje:
 - výpočet odhadovaných nákladů
 - rozložení práce a ceny v jednotlivých etapách řešení projektu
 - ...

Původní COCOMO



Hodnoty a, b, c, d jsou shodné pro střední a pokročilou úroveň modelu

- pro střední úroveň se aplikuje výpočet na celý projekt
- pro pokročilou úroveň se výpočet aplikuje pro jednotlivé etapy životního cyklu

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)



Existují různé verze COCOMO pro různé účely a prostředí

- COCOMO_85 a COCOMO_87, malé odchylky
- ADA_87 a APM_88, pro projekty v jazyce ADA
- inkrementální COCOMO, upravené pro použití při inkrementálním vývoji SW
- REVIC
 - odlišné hodnoty parametrů a rozložení úsilí v různých etapách projektu
 - statistický odhad SLOC a kvantitativní přístup k nejistotě

Úkoly



- Seznamte se s prostředím systému COCOMO.
- Aplikujte model COCOMO na vámi řešený projekt.