

# Odhady pracnosti a doby řešení

COCOMO a Function Points

# Hofstadterův „zákon“

”V softwaru vše stojí více a trvá déle, a to i tehdy, když provedeme na tuto skutečnost korekci původního odhadu“

# Obtíže odhadů - shrnutí

1. Silná závislost všech parametrů výsledného produktu, jako jsou náklady, kvalita řešení atd., na kvalitě řešitelského týmu.
2. Rychle se měnící podmínky (vlastnosti hardwaru, měnící se způsoby používání počítačů, ..) silně snižují opakovatelnost řešení. Jedná se tedy o stanovení pracnosti úkolu, který je do značné míry unikátní.
3. V programování se dosud neustálily pracovní postupy.
4. Rychlý vývoj - ke změnám podmínek řešení (know-how, hardware) může dojít i během řešení jediného úkolu.

# Dva principy odhadu

1. Přes odhad velikosti – COCOMO
2. Pomocí odhadů složitosti interakce s okolím – Function Points
3. V obou případech opravné (Pišvejcovy) koeficienty
4. V obou případech dvě varianty odhadu použitelné v různých etapách životního cyklu

# Důvody problémů

1. Lidé obecně mají tendenci být příliš optimističtí a očekávají, že se záležitosti budou vyvíjet spíše příznivě. Mezi odhadem doby realizace  $t$  a skutečnou dobou realizace  $a(t)$  platí vztah  $a(t) \cong 4/3t$ .
2. Dosavadní zkušenosti se berou v úvahu jen zčásti. Jestliže se projektuje nějaký systém, odhaduje se jeho rozsah analogií s podobným systémem. Přitom se mlčky soustředujeme na rozsah programů realizujících vlastní, "hlavní", úkoly systému. Tyto části často vyžadují pouze menší část prací na systému. Většinu prací pohltí takové úkoly jako reakce na chybu, přesuny dat, konverze dat, generace návodů (HELP), kontrola vstupních a výstupních dat atd. "Užitečné funkce", zajišťuje někdy jen několik procent kódu

# COCOMO 81

## COCOMO 1

Dvě základní rovnice,  $Del$  jsou efektivní  
tisíce řádků

$$Prac_{\check{c}m} = a * Del^b$$

$$Doba_{m\check{e}s} = 2.5 * Prac^c$$

Koeficient  $a$  se počítá jako součin

$$a = a_0 * \prod_1^{15} F_i$$

# Parametry modelu

Typ projektu	$a_0$	b	c
Organic	3.2	1.05	0.38
Semi-detached	3.0	1.12	0.35
Embedded	2.8	1.20	.0.32

# Typy projektů

Typ projektu	Charakteristiky projektu			
	Size	Innovation	Deadline/ constrains	Devel. Environment
Organic	Small	Little	Not tight	Stable
Semi-detached	Medium	Medium	Medium	Medium
Embedded	Large	Greater	Tight	Complex HW/customer interfaces



# Opravné koeficienty

**ACAP** – míra schopnosti analytiků.

**AEXP** – míra zkušeností programátorů s podobnými aplikacemi.

**PCAP** – míra kvality programátorů.

**VEXP** – míra zkušenosti s počítačem.

**LEXP** – míra zkušenosti s programovacím jazykem.

**MODP** – míra použití moderních metod vývoje softwaru.

**TOOL** – míra použití moderních prostředků vývoje softwaru.

**SCED** – "ostrost" požadavků na dobu realizace.

# Opravné koeficienty COCOMO 1

**RELY** – míra požadavků na spolehlivost.

**DATA** – míra rozsahu datové základny.

**CPLX** – složitost produktu.

**TIME** – míra požadavků na dobu odezvy.

**STOR** – míra využitelnosti paměti.

**VIRT** – míra proměnlivosti OS počítače

**TURN** – míra rychlosti oběhu úlohy počítačem

# Přechod na COCOMO II

- Mění se i exponent

# COCOMO II

- Odhad délky používá metodiku Function Points
- Exponenty odhadovány z atributů projektu
- Odhady pro jednotlivé etapy vývoje
- Jiné koeficienty resp. jejich hodnocení
- Doplněny postupy na znovupoužití
- **Jen slabé prostředky pro iterativní a inkrementální vývoj a obecně pro p2p**

# COCOMO II

$$Prac_{\check{c}m} = a * Del^b$$

$$Doba_{m\check{e}s} = Q * Prac^c$$

kde

$$a = a_0 * \Pi_1^{19}, a_0 = 2.94$$

$$b = B + 0.01 \sum_1^5$$

$$c = D + 0.002 \sum_1^5 F_i = D + 0.2(b - B)$$

$$Q = 3.67, D = 0.28$$

$F_i$  a  $H_i$  se hodnotí známkami 0 až 5

# COCOMO II, Fi

PREC míra podobnosti s předchozími projekty

FLEX neměnnost specifikací a externích rozhraní

RESL míra kritičnosti a

TEAM problémy s týmem (dohodování s uživateli, noví lidé, nevhodné složení)

PMAT kvalita SW procesů podle CMM dotazníků

# COCOMO II, Hi, počáteční odhad

RCPX spolehlivost a složitost systému

RUSE požadovaná znovupoužitelnost

PDIF míra interakce s platformou (využití, novost)

PERS kvalita lidí

PREX zkušenosti lidí

SCED ostrost termínu

FCIL použití a znalost nástrojů, nutnost cestování

# Function points

Základ je hodnocení složitosti  
rozhraní systému a jeho částí



# Principy

- Function points 1
  - Ze složitosti I/O se vypočtou neadjustované FP
  - Takto získaný odhad se znásobí funkcí číselných hodnocení atributů projektu a kalibrační konstantou. Tu nelze v SME vypočítat
  - Tím se získá odhad pracnosti resp. délky
  - Dále se postupuje podobně jako v COCOMO
- FP II Neadjustované body součtem bodů transakcí, používá se více charakteristik projektu

# FP1

$$Del = cF, Prac = d * Del^b$$

$$F = w_1 \cdot (IN) + w_2 \cdot (OUT) + w_3 \cdot (ENQ) + w_4 \cdot (FILE) + w_5 \cdot (FILEE)$$

$$w_1 = 4, w_2 = 5, w_3 = 4, w_4 = 10, w_5 = 7.$$

- IN a OUT je počet logicky nebo formátem odlišných vstupů a výstupů
- ENQ je totéž pro komunikaci s terminálem (jinou aplikací)
- FILE obdoba pro zápisy a čtení do vlastních DB
- FILEE totéž pro společné DB
- Některé metodiky zjemňují odhad zavedením hodnocení složitostí jednotlivých příkazů

# Výpočet $c$

$$c = 2/3 + 0.001 \sum^{14} F_i$$

$F_i$  jsou atributů projektu hodnocené od 0 do 5 podle následujícího hodnocení

0 – daný faktor neexistuje nebo nemá vliv,

1 – nevýznamný vliv,

2 – mírný vliv,

3 – průměrný vliv,

4 – významný vliv,

5 – velmi silný vliv na *celou* architekturu a programování

# Atributy

1. Ovládání a vstup dat přes síť (remote job entry, remote data entry).
2. Distribuované zpracování.
3. Ostré požadavky na výkonnost ovlivňují návrh, realizaci a instalaci.
4. Plné využití dané konfigurace.
5. Množství transakcí, které se provádějí, silně ovlivnilo návrh.
6. On-line vstup dat, např. vstup dat z terminálu.
7. On-line funkce, např. ovládání funkcí z terminálu.

# Atributy

8. Interaktivní přímé změny dat z cizích aplikací
9. Složitost zpracování: mnoho bodů rozhodování a řídicích interakcí, algoritmicky složité úlohy, mnoho zpracování výjimek vedoucích k opakovanému provádění.
10. Obecná použitelnost výsledného produktu.
11. Snadná instalace a přenositelnost.
12. Snadnost práce se systémem za provozu.
13. Použití systému více organizacemi s různým způsobem využití a jinou podnikovou kulturou.
14. Snadnost změn výsledného produktu.

# Funkční body II

- Neadjustované body součtem bodů elementárních akcí (transakcí)
- Více atributů projektů a jiné hodnocení atributů

Transakce	Vstupy	Vnitřní proměnné	Výstupy
Nový zákazník	53	1	3
Dotaz „je na skladě“	2	3	8
Záhlaví obj. – obj. přijata	20	2	1
Záhlaví obj. – obj. zamítnuta	8	2	10
Přidání položky	6	5	1
Odmítnutí položky	6	5	5
Objednávka – souhrn	2	4	4
Celkem	97	22	32

Tab. 16.4: Příklad výpočtu pro transakce prováděné při zpracování objednávek.

# Hodnocení faktorů

## Hodnocení faktorů

- 0 – žádný vliv,
- 1 – průměrný vliv,
- 3 – kritický vliv.
- *FP2* uvažuje celkem 19 faktorů. 14 faktorů je převzato z *FP*, doplněno je následujících pět faktorů:



# Doplněné faktory

- 15. Budování a udržování rozhraní na jiné aplikace.
- 16. Audit a ochrana dat a systému.
- 17. Umožnění přístupu k datům pro cizí systémy přes veřejnou síť.
- 18. Vývoj školicích prostředků pro uživatele.
- 19. Zvýšené nároky na dokumentaci.

# Normy pro funkční body

*ISO/IEC 14143-1 Software Measurement--Functional Size Measurement-Part1: Definition of Concepts*

*ISO/IEC 14143-2 Software Measurement--Functional Size Measurement-Conformity evaluation of software size measurement methods to ISO/IEC 14143-1:1998*

*ISO/IEC 19761 :2011 COSMIC-FFP - A Functional Size Measurement Method*

*ISO/IEC 20926 IFPUG 4.1 Unadjusted functional size measurement method. Counting practices manual*

*ISO/IEC 20968 Mark II Function Point Analysis Counting Manual*

# Hodnocení nástrojů odhadu

- Vhodné spíše pro IS s nepříliš složitými jednotlivými akcemi
- Asi se nehodí na RT systémy.
- Nepřesné, vyžaduje zkušenosti, málo dat , podceňování statistiky
- Funkční body asi lepší

# Kdy nejdříve se dá co použít

Omezená použitelnost pro p2p architektury  
zvláště ty s hrubozrným rozhraním možná  
i pro komponentový přístup v OO

Pochybnosti u agilních metod (spíše nesmělé  
pokusy)

Malý progres metod odhadu v po roce 2005

COCOMO 1, FP 1: počátek specifikací,

# Kdy nejdříve se dá co použít

COCOMO 1, FP 1: záhy po počátku  
specifikací,

COCOMO 1 se dá použít poněkud dříve

Ostatní metody: Skoro na konci specifikací,  
FP1 o něco dříve

# Malá firma

- Pro dobrou volbu hodnot koeficientů a hlavně pro kalibraci málo dat a zkušeností
- Používat intuici s tím že se sledují výrazné změny atributů projektu jako je délka, obvyklá doba řešení, okolnosti silně ovlivňující koeficienty
- Pamatovat, že obchodní okolnosti mívají silnější vliv, než okolnosti technické
- Malá firma mívá výhodu ve větší pružnosti

# Pro malé firmy v podstatě alchymie

- Spousta kalibračních konstant, pro jejichž odhad chybí data
  - Je jich málo
  - Rychle se mění
- „Zimmermanovy opravy“, násobící, Dirac měl aditivní
- Jako kvalitativní varování dobré i pro malé firmy, softwarové zákony
-

Hlavní parametr odhadu	Délka	<i>FP</i>	<i>FP2</i>
Standardizováno	ne	ano, zlepšuje se	ano
Přesnost definice	potenciálně možná <sup>5</sup>	část. subjektivní	téměř ano
Strukturovanost	ne	ne	ano
Snadnost pochopení	ano	jen zdánlivě	ano
Automatizovatelné	ano	obtížně	ano
Veřejné systémy	obojí	veřejné	veřejné
Rozsah použití	velký	velký	přiměřený
Organizace uživatelů	několik	IFPUG <sup>6</sup>	IFPUG
Školení	různé úrovně	ano	ano