

Ocena kosztów oprogramowania

Ocena kosztów opracowania sprowadza się do oceny:

- czasu realizacji,
- rozmiarów zatrudnienia (osobomiesięc).

Jest trudnym i nie do końca rozwiązany problemem.

- Podejścia heurystyczne:
 - ocena przez analogię,
 - ocena przez eksperta.
- Podejście analityczne:
 - budowa modelu złożoności przetwarzania (FPA)
 - obliczenie pracochłonności i czasu realizacji (COCOMO).

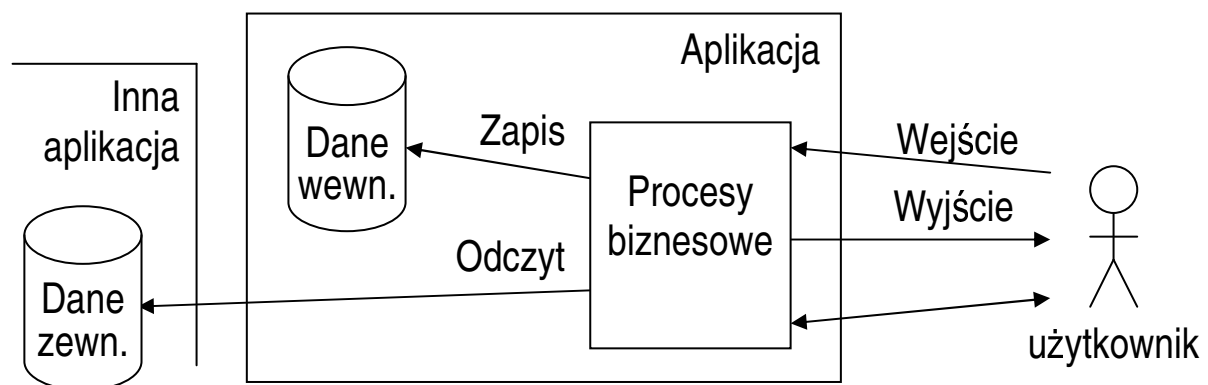
Metoda punktów funkcyjnych

(*Function Point Analysis — FPA*)

- Konsorcjum IFPUG (*International Function Point Users Group*)
- Ocena złożoności struktur danych i przetwarzania
- Wynikiem jest liczba (punktów funkcyjnych)

Możliwe przeliczenie: złożoność → rozmiaru → czas realizacji

Elementy oceny:



- zbiory wewnętrzne (*Internal Logical File — ILF*)
- zbiory zewnętrzne (*External Interface File — EIF*)
- wejścia zewnętrzne (*External Input — EI*)
- wyjścia zewnętrzne (*External Output — EO*)
- zapytania (*External Inquiry — EQ*)

1. Identyfikacja i ocena złożoności zbiorów i transakcji

(w skali: *mała*, *średnia*, *wysoka*)

- Ocena ILF i EIF bierze się pod uwagę:

RET — liczbę różnych formatów rekordów (formularzy),

DET — liczbę pól rekordu,

Ocena ILF, EIF	1...19 DET	20...50 DET	51... DET
1 RET	<i>mała</i>	<i>mała</i>	<i>średnia</i>
2...5 RET	<i>mała</i>	<i>średnia</i>	<i>wysoka</i>
6... RET	<i>średnia</i>	<i>wysoka</i>	<i>wysoka</i>

- Ocena EI i EO bierze pod uwagę:

DET — liczbę pól rekordu,

FTR — liczbę zbiorów ILF i EIF używanych w transakcji

Ocena EI	1...4 DET	5...15 DET	16... DET
0...1 FTR	<i>mała</i>	<i>mała</i>	<i>średnia</i>
2 FTR	<i>mała</i>	<i>średnia</i>	<i>wysoka</i>
3... FTR	<i>średnia</i>	<i>wysoka</i>	<i>wysoka</i>
Ocena EO	1...5 DET	6...19 DET	20... DET
0...1 FTR	<i>mała</i>	<i>mała</i>	<i>średnia</i>
2...3 FTR	<i>mała</i>	<i>średnia</i>	<i>wysoka</i>
4... FTR	<i>średnia</i>	<i>wysoka</i>	<i>wysoka</i>

- Dla EQ oblicza się osobno komplikację wejścia (jak EI), komplikację wyjścia (jak EO) i bierze wartość większą.

2. Obliczenie złożoności aplikacji w punktach funkcyjnych

Złożoność	<i>mała</i>	<i>średnia</i>	<i>wysoka</i>	
zbiory wewnętrzne (ILF)	$x * 7 =$	$x * 10 =$	$x * 15 =$	Σ wiersza
zbiory zewnętrzne (EIF)	$x * 5 =$	$x * 7 =$	$x * 10 =$	Σ wiersza
wejścia zewnętrzne (EI)	$x * 3 =$	$x * 4 =$	$x * 6 =$	Σ wiersza
wyjścia zewnętrzne (EO)	$x * 4 =$	$x * 5 =$	$x * 7 =$	Σ wiersza
zapytania (EQ)	$x * 3 =$	$x * 4 =$	$x * 6 =$	Σ wiersza
nieskorygowane punkty funkcyjne UFP=				Σ

3. Skorygowanie oceny o dodatkowe czynniki

Zdefiniowano 14 czynników wpływających na złożoność przetwarzania, np:

- rozproszenie aplikacji,
- wymagana wydajność,
- złożoność algorytmów przetwarzania,
- zespół projektowy rozproszony w wielu miejscach,
-

Rolę każdego czynnika w projekcie ocenia się w skali (0..5):

żadna, nieznaczna, umiarkowana, średnia, znacząca, duża
i sumuje otrzymując czynnik TDI (*Total Degree of Influence*).

Ostateczny współczynnik korekcyjny:

$$VAF = 0.65 + (0.01 * TDI)$$

pozwala wyrazić złożoność w skorygowanych punktach funkcyjnych:

$$AFP = UFP * VAF$$

4. Oszacowanie rozmiaru programu

AFP jest miarą wysiłku potrzebnego do wytworzenia programu.

- Można na tym poprzestać i używać punktów do porównywania.
- Można obliczyć przewidywaną liczbę linii programu, korzystając ze współczynnika obrazującego wydajność wybranego języka (*Backfire Method*):

$$SLOC = AFP * LM$$

gdzie: *SLOC* — *Source Lines of Code*

LM — *Language Multiplier*

Język	<i>LM</i>
<i>Asembler</i>	320
<i>Makroassembler</i>	213
<i>Pascal</i>	91
<i>Basic (kompilowany)</i>	91
<i>C</i>	128
<i>C++</i>	53
<i>Visual Basic v5</i>	29
<i>SQL</i>	13
<i>AI Shell</i>	49

Model kosztów COCOMO

(Constructive Cost Model)

- Rozwijany przez konsorcjum *COCOMO Research Group*
- Podstawą oceny jest spodziewany rozmiar programu.
- Wynikiem analizy jest liczbowa ocena pracochłonności:

$$Effort = A * (KSLOC)^B * \Pi(F_i)$$

oraz optymalny czas realizacji:

$$Time = C * (Effort)^D$$

gdzie:

A, B, C, D — stałe zależne od rodzaju systemu:

	<i>organic</i>	<i>semidetached</i>	<i>embedded</i>
A	3.2	3.0	2.8
B	1.05	1.12	1.20
C	2.5	2.5	2.5
D	0.38	0.35	0.32

F_i — czynniki wpływające na koszt projektu, np.:

- wymagana niezawodność (0.75 .. 1.40),
- rozmiar bazy danych (0.94 .. 1.18),
- złożoność (0.70 .. 1.65)
- ograniczenia wydajnościowe (1.00 .. 1.66)
- ograniczenia pamięciowe (1.00 .. 1.56)
- biegłość analityków (0.71 .. 1.46)
- kwalifikacje projektantów (0.70 .. 1.42)
- biegłość programistów (0.95 .. 1.14)
- wykorzystanie metod IO (0.82 .. 1.24)
- wykorzystanie narzędzi CASE (0.83 .. 1.24)
-

Model COCOMO II

Zestaw 3 modeli stosowanych do różnych typów i faz projektu:

- kompozycyjny,
(aplikacje generowanych za pomocą *GUI-builder tools*),
- wczesny,
(budowany na początku, przed projektem architektury),
- po-architektoniczny,
(budowany po określeniu architektury projektu),

- Szacowanie rozmiaru programu: $SLOC = \underline{UFP} * LM$

- Podstawowy wzór b.z.:

$$Effort = A * (KSLOC)^B * \Pi(F_i)$$

- Dokładniejsza ocena wykładnika B :

$$B = 1.01 + 0.01 * \Sigma(S_i)$$

gdzie S_i (*scale factors*) oceniane: *very high* (0) ... *very low* (5)
odzwierciedlają różne uwarunkowania projektu:

PREC — powtarzalność,

FLEX — elastyczność wymagań,

RESL — jakość analizy ryzyka,

TEAM — spójność celów,

PMAT — dojrzałość procesu (wg. *Capability Maturity Model*).

$$// B \in [1.01, 1.26]$$

$$// 5 * nominal = 5 * 0.3 \Rightarrow B = 1.16$$

- Model wczesny

$$Effort = A * (KSLOC)^B * \prod_{i=1..7}(F_i)$$

7 mnożników kosztu (ocena w 7 punktowej skali symetrycznej):

RCPX — złożoność i niezawodność systemu,

RUSE — przewidywane ponowne wykorzystanie komponentów,

PDIF — ograniczenia sprzętowe,

PERS — kwalifikacje zespołu projektowego,

PREX — znajomość platformy projektowej,

FCIL — użycie narzędzi wspomagających i rozproszenie zespołu,

SCED — napięty harmonogram.

- Model po-architektoniczny

$$Effort = A * (KSLOC)^B * \prod_{i=1..17}(F_i)$$

17 mnożników kosztu.

- Czas realizacji:

$$Time = [3.67 * (\underline{Effort})^{(0.28+0.2*(B-1.01))}] * SCED$$

gdzie Effort oznacza pracochłonność **bez** uwzględnienia SCED.