

Zarządzanie jakością i metryki oprogramowania

Organizacja i Zarządzanie
Projektem Informatycznym

Jarosław Francik,
kwiecień 2002



Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Zapewnienie jakości produktu informatycznego

- **Pomiar** jako główny element technik zapewniania jakości
- Statystyczna kontrola procesu
(jedna z podstaw sukcesu gospodarczego Japonii)
- **Zapewnienie jakości – w oprogramowaniu:**
zbiorczy termin odnoszący się do zestawu mgliście zdefiniowanych działań, w związku z którymi większość ludzi nie dostrzega żadnej potrzeby dokonywania pomiarów

Norman Fenton

Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Plan

- Zapewnienie jakości produktu informatycznego
- Podstawowe parametry mierzalne
- Metryki oprogramowania
- Rejestrowanie problemów
- Wybrane normy i standardy
- Podsumowanie



Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Podstawowe parametry mierzalne

- Koszt i nakład pracy
- Wydajność ludzi
- Oprogramowanie
- Niezawodność
- Wydajność
- Złożoność
- Dojrzałość procesu wytwarzania



Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Zapewnienie jakości produktu informatycznego

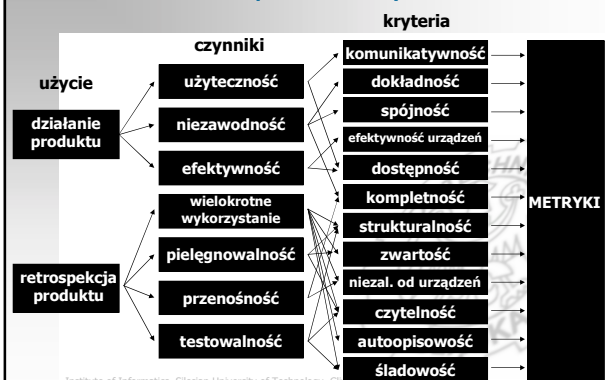
- **Zapewnienie jakości (quality assurance):**

Działanie związane z dostarczeniem wszystkim zainteresowanym dowodów wystarczających do powstania przekonania, że funkcja jakości jest realizowana poprawnie

Zaplanowany i systematyczny ogół wszystkich działań niezbędnych do wytworzenia odpowiednio silnego zaufania, że element lub produkt odpowiada ustalonym względem niego wymaganiom technicznym
norma IEEE 729 (1983)

Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Podstawowe parametry mierzalne



Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Metryki oprogramowania

Plan

- Wstęp: Elementy pomiaru oprogramowania
- Pomiary gęstości defektów
- Pomiary wielkości
- Pomiary złożoności
 - nauka o programach Halsteada
 - liczba cyclomatyczna McCabe
- Szacowanie nakładów
 - model COCOMO
 - modelowanie regresyjne
- Szacowanie funkcjonalności – metoda punktów funkcyjnych
- Dojrzałość procesu wytwarzania - model CMM

Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Metryki oprogramowania

Pomiary gęstości defektów

metryka gęstości defektów

$$M = \frac{\text{liczba defektów odkrytych w C}}{\text{rozmiar C}}$$

Problemy:

- wyraża jakość programu czy może rzetelność testów?
- brak zgody co do definicji defektu
- brak zgody co do sposobu pomiaru rozmiaru kodu

Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Metryki oprogramowania

Elementy pomiaru oprogramowania

- **Pomiar:**

Proces, w którym atrybutom elementów świata rzeczywistego przydzielane są liczby lub symbole w taki sposób, aby scharakteryzować te atrybuty według jasno określonych zasad. Jednostki przydzielone atrybutom nazywamy ich **miarą**.
- **LOC (Line of Code):** jednostka miary długości programu (ale nie złożoności ani nawet rozmiaru)

Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Metryki oprogramowania

Pomiary gęstości defektów

metryka gęstości defektów

$$M = \frac{\text{liczba defektów odkrytych w C}}{\text{rozmiar C}}$$

Mimo to:

- standard *de facto*
- formalne definicje na użytek wewnętrzny
- poszukiwanie możliwości porównania zewnętrznego

Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Metryki oprogramowania

Elementy pomiaru oprogramowania

- **Elementy – przedmioty pomiaru:**
 - procesy
(projektowanie, kodowanie, pierwsze trzy miesiące)
 - produkty
(kod źródłowy, specyfikacja, modyfikacja, plan testów, podręcznik użytkownika, pakiet instalacyjny)
 - zasoby
(osoby, zespoły, oprogramowanie, infrastruktura, biura)
- **Mierzone atrybuty:**
 - wewnętrzne
 - zewnętrzne

Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Metryki oprogramowania

Pomiary wielkości

• Metryki długości kodu źródłowego:

- LOC – Lines of Code
- KLOC – Kilo Lines of Code
- Liczba instrukcji
- Liczba instrukcji wykonywalnych (bez deklaracji)
- DSI – Delivered Source Instruction

• Problemy:

- mierzalne na etapie implementacji
- trudne do wcześniejszego prognozowania
- wielkość \neq długość (co ze złożonością?)

Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Metryki oprogramowania

Pomiary złożoności

Nauka o programach Halsteada *Halstead's software science*

n_1 – liczba różnych operatorów

n_2 – liczba różnych operandów

N_1 – całkowita liczba wystąpień operatorów

N_2 – całkowita liczba wystąpień operandów

Wysilek:

$$E = n_1 N_2 (N_1 + N_2) \log(n_1 + n_2) / 2n_2$$

Czas:

$$T \approx E / 18 [s]$$

oraz wiele innych wskaźników...

(<http://yunus.hun.edu.tr/~sencer/complexity.html>)

Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Metryki oprogramowania

Szacowanie nakładów

Model COCOMO *Constructive Cost Model*

$$\text{nakład pracy} = a * \text{rozmiar}^b \quad [\text{osobomiesiące}]$$

rozmiar = KDSI – kilo DSI (Delivered Source Instructions)

system ograniczony $a = 2,4$ $b = 1,05$

system częściowo wydzielony $a = 3,0$ $b = 1,12$

system wbudowany $a = 3,6$ $b = 1,2$

Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Metryki oprogramowania

Pomiary złożoności

Liczba cyklomatyczna McCabe'a *M McCabe's cyclomatic number*

M = liczba niezależnych dróg w schemacie blokowym

M = liczba decyzji w programie + 1

Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Metryki oprogramowania

Szacowanie nakładów

Model COCOMO *Constructive Cost Model*

$$\text{czas} = a * \text{nakład-pracy}^b \quad [\text{miesiące}]$$

system ograniczony $a = 2,5$ $b = 0,38$

system częściowo wydzielony $a = 2,5$ $b = 0,45$

system wbudowany $a = 2,5$ $b = 0,32$

WADA:

- wymagana znajomość rozmiaru kodu (jeden problem prognostyczny zastępujemy innym)

Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Metryki oprogramowania

Pomiary złożoności

Problemy:

Niektórzy twierdzą, że metryki te nie są lepsze niż LOC!

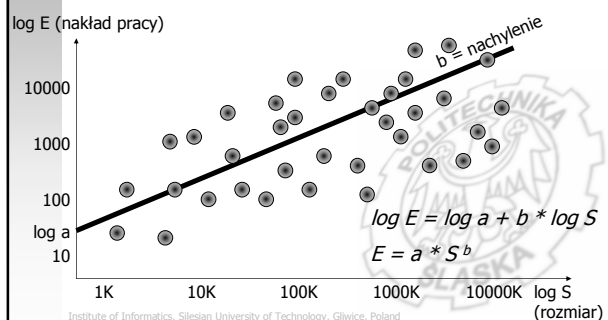
- nie pozwalają przewidzieć nakładu pracy, niezawodności, pielęgnowalności
- mierzalne na etapie implementacji
- wciąż zbyt uproszczone (nie uwzględniają złożoności przepływów danych, zagnieźdzenia itd..)

Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Metryki oprogramowania

Szacowanie nakładów

Model regresyjny



Metryki oprogramowania

Szacowanie funkcjonalności

Metoda **punktów funkcyjnych** Albrechta

$$FP = UFC * TCF$$

UFC – pierwotna liczba punktów funkcyjnych:
ważona liczba wejść, wyjść, logicznych plików głównych
(pamięć danych), plików interfejsowych i usług

TCF – współczynnik złożoności technicznej:
dodatkowy wpływ 14 innych czynników

International Function Point User Group

Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Metryki oprogramowania

Dojrzałość procesów wytwarzania

- Model dojrzałości procesu wytwarzania CMM



Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Metryki oprogramowania

Szacowanie funkcjonalności

Metoda **punktów funkcyjnych** Albrechta

- Metoda precyzyjna
- Prognoza uzyskiwana na podstawie **specyfikacji**
NIE TRZEBA PROGNOZOWAĆ ROZMIARU
- Zastosowania:
 - metryka rozmiaru (zamiast LOC),
 - wydajności programistów,
 - metryki gęstości defektów
- Podważana sensowność użycia **TCF**
- WADA**: metoda skomplikowana i trudna w użyciu

Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Rejestrowanie problemów

- Rodzaje problemów:

- usterka (defekt)** – wada w produkcie powstała na skutek błędu człowieka
- awaria** – odstępstwo systemu od wymaganego zachowania
- incydent** – niepożądane zdarzenie zaobserwowane podczas pracy systemu, mogące być skutkiem awarii (ale niekoniecznie!)

Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Metryki oprogramowania

Dojrzałość procesów wytwarzania

- Cel: Poprawa procesów programowych
(**SPI** – *Software Process Improvement*)
- Model dojrzałości procesu wytwarzania
(**CMM** – *Capability Maturity Model*)
 - Carnegie Mellon Univ.
 - zastosowanie: Dept. Obrony USA

Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Rejestrowanie problemów

- Rejestrowane informacje:

- Lokalizacja
- Czas
- Tryb
- Skutek
- Mechanizm
- Przyczyna
- Natężenie
- Koszt

ortogonalna baza cech

Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Rejestrowanie problemów

- **Raport o incydencie:**

- Lokalizacja – id komputera, nr seryjny, platforma
- Czas – data i godzina wystąpienia incydentu
- Tryb – komunikat o błędzie
- Skutek – np. utrata danych, załamanie systemu
- Mechanizm – np. poprzedzające komendy klawiatury
- Przyczyna – próba diagnozy
- Natężenie – nieznaczny / poważny / krytyczny
- Koszt – koszt naprawy + utraconych danych

Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Podsumowanie

- **Rola metryk w zarządzaniu jakością**

- **Cele:**

- polepszenie oceny nakładów na produkcję oprogramowania
- ocena wydajności programistów
- ocena i kontrola jakości końcowego produktu

Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Rejestrowanie problemów

- **Raport o usterce:**

- Lokalizacja – id modułu lub dokumentu
- Czas – faza w której usterkę wprowadzono
- Tryb – sposób wykrycia usterki
- Skutek – opis awarii będącej skutkiem usterki
- Mechanizm – jak usterka powstała i jak ją wykryto
- Przyczyna – na czym polegał błąd człowieka
- Natężenie – czy może prowadzić do awarii (jakiej)
- Koszt – koszt lokalizacji i skorygowania

Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Wybrane normy i standardy

- **ISO 9001: Standard kontroli jakości**

nie odnosi się do żadnej konkretnej branży

- **ISO 9003**

wersja ISO 9001 dotycząca oprogramowania

- **TickIT**

uregulowanie brytyjskie, oparte na ISO 9003

Institute of Informatics, Silesian University of Technology, Gliwice, Poland