

Inżynieria Oprogramowania

Jakość



Wydział Matematyki, Informatyki i Mechaniki
Uniwersytet Warszawski
www.mimuw.edu.pl/~dabrowski



- Zestrzelenie samolotu Airbus 320, 1988
 - 290 zabitych
- Śmierć pacjentów chorych na raka, 1985-87
- Pentium floating point, 1994
 - Koszt $\sim 475\,000\,000\$$
- Rakieta Ariane 5, 1996
 - Budowa $\sim 7\,000\,000\,000\$$, Straty $\sim 500\,000\,000\$$



Czym jest jakość?

- Podstawowe pytania:
 - Czym jest jakość?
 - Jak można mierzyć jakość?
 - Mierzenie jakości przed dostarczeniem systemu
 - Związek jakości produktu z jakością procesu wytwórczego
 - Jak można porównywać jakość różnych dostawców?
 - Normy i standardy jakości
 - Dojrzałość firm wytwarzających oprogramowanie



- Pomyślmy o przedmiocie codziennego użytku
 - Np. krzesło
- Jak zmierzylibyśmy „jakość” krzesła?
 - Jakość konstrukcji?
 - Np. wytrzymałość
 - Wartości estetyczne?
 - Np. elegancja
 - Zgodność z oczekiwaniami?
 - Np. wygodne
- Wszystkie miary jakości są względne
 - Nie ma skali absolutnej
 - Nawet jeśli uda się stwierdzić:
 - A jest lepsze niż B
 - to trudno będzie stwierdzić:
 - o ile lepsze



- Czy da się łatwo mierzyć jakość w produkcji oprogramowania?
 - Jakość konstrukcji?
 - Oprogramowanie nie powstaje w procesie manufaktury
 - Wartości estetyczne?
 - Zdecydowana większość napisanego oprogramowania jest niewidoczna
 - Wartości estetyczne mają znaczenie w przypadku interfejsu użytkownika, ale to jest element marginalny z punktu widzenia jakości
 - Zgodność z oczekiwaniami?
 - Wymagałoby prawdziwego rozumienia przeznaczenia systemu



Naturalne jest podejście...

- Jakość projektu polega na zapewnieniu, że oprogramowanie jest dopasowane do potrzeb
 - Czy wykonuje to co jest potrzebne?
 - Czy wykonuje to tak jak potrzebują tego użytkownicy?
 - Czy jest wystarczająco niezawodne? Wystarczająco szybkie? Wystarczająco bezpieczne?
 - Czy użytkowników stać będzie na nie? Czy będzie dostępne gdy użytkownicy będą go potrzebować?
 - Czy będzie można je modyfikować zgodnie z potrzebami zmian?
 - ...

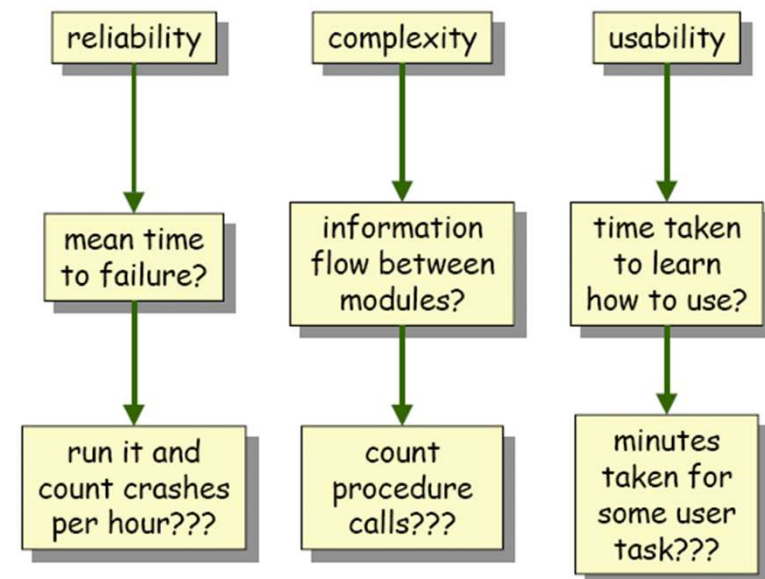


Cztery najważniejsze elementy?

- **Niezawodność**
 - Projektant musi przewidzieć jak system się zachowa:
 - Kompletność
 - Czy robi wszystko, co potrzeba?
 - Czy obsługuje wszystkie dane wejściowe?
 - Spójność
 - Czy zawsze zachowuje się zgodnie z oczekiwaniami?
 - Odporność
 - Czy dobrze zachowuje się w warunkach niestandardowych?
- **Wydajność**
 - Wykorzystanie zasobów takich jak czas procesora, pamięć, przepustowość sieci
 - W większości przypadków to jest mniej istotne niż niezawodność
- **Pielęgnowalność**
 - Jak łatwo można będzie w przyszłości modyfikować oprogramowanie?
 - Działania doskonalące, adaptujące, naprawcze
- **Użyteczność**
 - Jak łatwe jest w użyciu?



- Potrzeba zamienienia nieprecyzyjnych pomysłów w konkretne miary
- Konceptcje jakości
 - (pojęcia abstrakcyjne)
- Pojęcia mierzalne
 - (definicja metryk)
- Pobranie pomiarów
 - (realizacja metryk)





Jakość trzeba przewidywać

- Ale taka definicja oznacza, że jakość oprogramowania nie jest mierzona w odosobnieniu
 - Jest miarą dla relacji pomiędzy oprogramowaniem a dziedziną zastosowań
 - Może nie da się mierzyć jakości przed uruchomieniem oprogramowania w środowisku docelowym...
 - ...a jakość będzie się różnić w zależności od środowiska
- Natomiast:
 - Podczas projektowania musi być możliwe przewidywanie jak dobrze oprogramowanie spełni zakładane cele
 - Należy rozumieć potrzeby (inżynieria wymagań)
 - Należy szukać wyznaczników jakości (które pomogą przewidywać jakość)



Jak uzyskać wysoką jakość?

- Dwa aspekty
 - Kontrola jakości
 - Zapewnianie jakości

- Dodatkowo
 - mała i duża skala (projektu)



- Kontrola jakości
 - Walidacja
 - Weryfikacja

- Techniki kontroli jakości



- Czy system realizuje to co trzeba?
 - "Czy budujemy właściwy system?"
- Trudno to stwierdzić
- Oceny są zazwyczaj subiektywne



- Czy oprogramowanie jest zgodne ze specyfikacją?
 - “Czy prawidłowo budujemy system?”
- Może być obiektywne
- Specyfikacje muszą być wystarczająco precyzyjne



Proces weryfikacji i walidacji

- Weryfikacja i walidacja musi wykonywana na każdym etapie tworzenia oprogramowania

- Główne cele:
 - Wykrycie błędów w systemie
 - Ocena czy system jest możliwy do wykorzystania produkcyjnego



Trzy sposoby



- Testowanie
 - System jest uruchamiany z danymi testowymi i sprawdzane jest jego zachowanie
- Inspekcje
 - Analiza statycznej reprezentacji systemu w celu wykrycia problemów
- Metody formalne
- Testować należy wszystko
 - Także sam proces weryfikacji i walidacji