

# Inżynieria Oprogramowania wprowadzenie

Dr hab. inż. Ilona Bluemke

# Plan wykładu

- Powstanie inżynierii oprogramowania
- Cele i zadania inżynierii oprogramowania
- Czym zajmuje się inżynieria oprogramowania
- Oprogramowanie wysokiej jakości

# Powstanie inżynierii oprogramowania -1

- Koniec lat sześćdziesiątych - tzw. kryzys oprogramowania.
- Wiele realizowanych wówczas projektów kończyło się fiaskiem, a ceny realizowanego wówczas oprogramowania rosły szybko (około 12% na rok) przy zmniejszających się cenach sprzętu.

# Powstanie inżynierii oprogramowania - 2

Przyczyny upadku wielu projektów to:

- duża złożoność systemów,
- nowe dziedziny zastosowań, niepowtarzalność przedsięwzięć,
- niesystematyczny proces budowy oprogramowania,
- trudności w ocenie stopnia zaawansowania prac programistycznych,
- pozorna łatwość wytwarzania i dokonywania poprawek (np. 100 linii w 1 dzień, 1000 linii w 10 dni ?).

# Powstanie inżynierii oprogramowania - 3

- Ulepszenia w procesie produkcji oprogramowania mogą przynieść duże korzyści ekonomiczne.
- Pomysłów na poprawę procesu produkcji szukano w innych naukach inżynieryjnych np. u inżynierów mechaników (ang. mechanical engineering) czy inżynierów budowy dróg i mostów (ang. civil engineering).
- Powstająca dziedzina, poprzez analogie została nazwana ang. **software engineering**, **inżynieria oprogramowania**.

# Cel inżynierii oprogramowania

- **poszukiwanie i wdrażanie metod oraz technik produkcji programów o wysokiej jakości,**
- **produkcja w sposób najbardziej efektywny.**

# Oprogramowanie wysokiej jakości

- działa zgodnie z wymaganiami określonymi przez specyfikację,
- jest tak szybkie, wydajne i funkcjonalne jak oczekuje użytkownik,
- daje się łatwo pielęgnować (korekcja i modyfikacja),
- posiada pełną dokumentację użytkową i projektową, która umożliwia spełnienie poprzednich postulatów.

# Inżynieria oprogramowania

- dotyczy oprogramowania tworzonego przez zespoły,
- jej zasady są wykorzystywane w rozwoju systemu,
- zawiera aspekty techniczne i nie-techniczne,
- występują w niej podejścia formalne i praktyczne.



# Co oferuje inżynieria oprogramowania

- techniki i narzędzia ułatwiające pracę nad złożonymi systemami,
- systematyzację procesu produkcji oprogramowania, tak by ułatwić jego monitorowanie i planowanie,
- metody wspomagające analizę nieznanych problemów i ułatwiające wykorzystywanie wcześniejszych doświadczeń.

# Inżynieria oprogramowania zajmuje się :

- sposobami prowadzenia przedsięwzięć informatycznych,
- technikami szacowania kosztów, harmonogramowania,
- metodami analizy i projektowania systemów,
- technikami zwiększania niezawodności oprogramowania,

# Inżynieria oprogramowania zajmuje się – 2:

- sposobami testowania systemów, szacowania niezawodności,
- sposobami przygotowywania dokumentacji technicznej i użytkowej,
- procedurami kontroli jakości,
- technikami pracy zespołowej.

# Jakość oprogramowania

Ocena jakości oprogramowania jest sprawą subiektywną.

Model Mc Call'a dzieli kryteria oceny jakości na grupy związane:

- ze sposobem działania
- z możliwością zmian i poprawek
- z mobilnością oprogramowania.

# Kryteria związane ze sposobem działania

- przyjazność - dotyczy projektu interfejsu
- bezpieczeństwo - kontrola uprawnień dostępu,
- wydajność,
- poprawność - stopień realizacji wymagań,
- kompletność i logiczność implementacji, zgodność działania ze specyfikacją,
- niezawodność - odporność na błędy.

# możliwość wprowadzenia zmian i poprawek

- pielęgnowalność - stopień przystosowania do poprawienia, modyfikacji, rozszerzania, adaptowania,
- elastyczność - możliwości rozbudowywania oprogramowania o nowe funkcje oraz uniwersalność zaimplementowanych rozwiązań,
- testowalność

# mobilność oprogramowania

- przenośność - zdolność do łatwego uruchamiania na innych systemach,
- uniwersalność - odnosi się do możliwości wykorzystania istniejącego oprogramowania lub jego fragmentów do konstrukcji innych systemów,
- otwartość - stopień przystosowania programu do współpracy lub wymiany informacji z innymi systemami komputerowymi.

# problem

- Problem - osiągnięcia optimum
- **Co optymalizować** powinno być ustalone z klientem - np. lepszy interfejs z użytkownikiem to spadek efektywności.



# Jakość

- produktu
- procesu wytwarzania

**Technologie**

**Jakość  
procesu**

**Jakość  
produktu**

**Ludzie**

**Koszt, czas  
harmonogram**

# niezawodność a efektywność oprogramowania

- Szybszy, tańszy sprzęt, ważniejsza wygoda użytkownika
- Zawodne oprogramowanie będzie unikane
- Są zastosowania gdzie koszt błędu systemu może znacznie przekraczać koszt samego systemu. Koszt ludzkie - nie do zaakceptowania (safety -critical systems)
- System zawodny trudno jest ulepszyć, poprawić.

# niezawodność a efektywność oprogramowania -2

- System niezawodny można stroić, lokalizować przyczyny opóźnień
- Nieefektywność - program wykonuje się dłużej, skutki można przewidzieć
- Zawodność - skutki mogą być trudne do przewidzenia, błędy w projekcie mogą prowadzić do katastrofy
- Zawodne systemy mogą powodować utratę danych

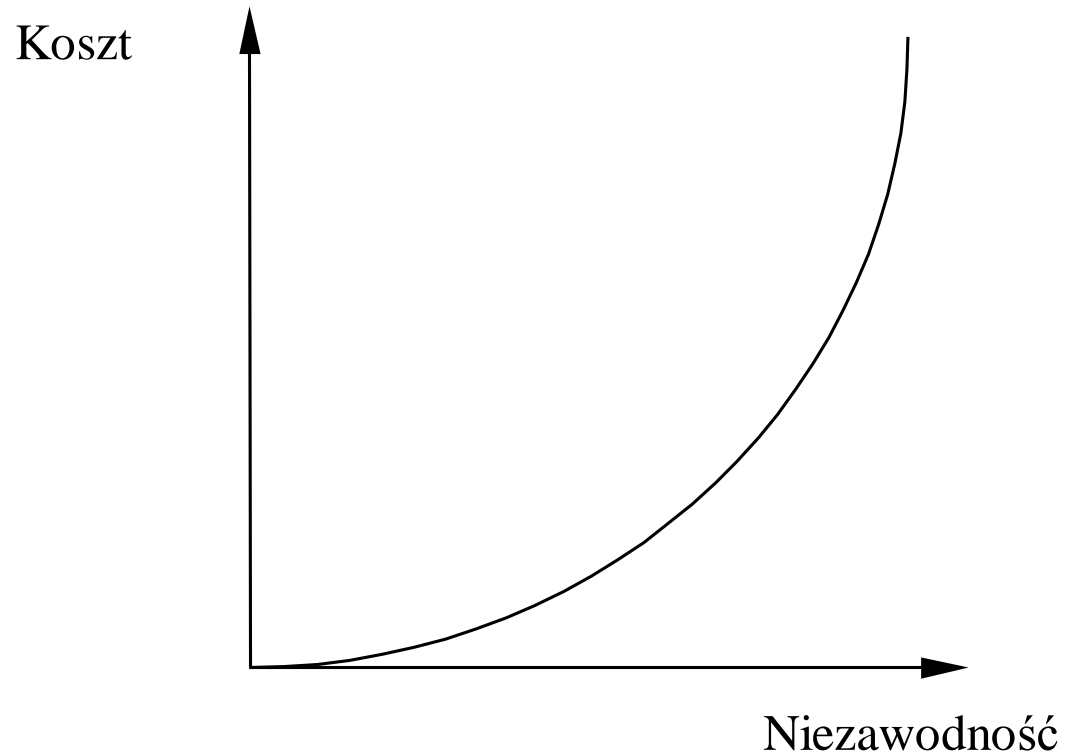
# Niezawodność zależy od:

- poprawności projektu
- poprawności odwzorowania projektu w implementację
- poprawności elementów i ich złożenia

Miary niezawodności oprogramowania - inne niż sprzętu

Kluczem do niezawodności oprogramowania jest specyfikacja

# Koszt a niezawodność



Rys. 1.1 Koszt oprogramowania a niezawodność

# Przykładowe pytania

- Czym zajmuje się inżynieria oprogramowania ?
- Jakie są cechy oprogramowania wysokiej jakości ?
- Jakie można stosować kryteria oceny oprogramowania ?
- Jakie cechy oprogramowania są związane z jego działaniem ?
- Co oferuje inżynieria oprogramowania ?
- Dlaczego ważniejsza jest niezawodność oprogramowania, niż efektywność ?