

# **INŻYNIERIA OPROGRAMOWANIA**

## **wykład 10: ZAPEWNIENIE JAKOŚCI OPROGRAMOWANIA**

**dr inż. Leszek Grocholski**  
Zakład Inżynierii Oprogramowania  
Instytut Informatyki  
Uniwersytet Wrocławski

# POJĘCIE JAKOŚCI

Krótką i jasną definicją:

**Jakość (quality, die Qualität) to stopień realizacji wymagań**

Inne definicje

1. Pewien stopień doskonałości •Platon, IV wiek p.n.e.
2. Zgodność z wymaganiami (conformance to user requirements)
  - P.B. Crosby, Quality Is Free, McGraw-Hill, 1979
3. Osiągnięcie doskonałych poziomów zdolności do użycia (achieving excellent levels of fitness for use)
  - W. Humphrey, Managing the Software Process, Addison-Wesley, 1989
4. Jakość sterowana rynkiem (market-driven quality) •IBM, lata 80-te
5. Stopień, w jakim zbiór inherentnych charakterystyk spełnia wymagania (the degree to which a set of inherent characteristics fulfills requirements)
  - ISO 9001:2000, Quality Management Systems - Requirements, ISO, 2000
  - Przykładowe charakterystyki: funkcjonalność, niezawodność, użyteczność, wydajność, rozszerzalność

# Co to jest

## “zapewnienie jakości oprogramowania” ?

Zapewnienie jakości jest rozumiane jako zespół działań zmierzających do wytworzenia u wszystkich zainteresowanych przekonania, że dostarczony produkt właściwie realizuje swoje funkcje i odpowiada aktualnym wymaganiom i standardom.

Problem jakości, oprócz mierzalnych czynników technicznych, włącza dużą liczbę niemierzalnych obiektywnie czynników psychologicznych.

Podstawą obiektywnych wniosków co do jakości oprogramowania są **pomiary** pewnych parametrów użytkowych (niezawodności, szybkości, itd.) w realnym środowisku, np. przy użyciu metod statystycznych.

**Niestety, obiektywne pomiary cech produktów programistycznych są utrudnione lub niemożliwe.** Jakość gotowych produktów programistycznych jest bardzo trudna do zmierzenia ze względu na ich złożoność (eksplozja danych testowych), wieloaspektowość, identyczność (?) wszystkich kopii produktu, oraz niską przewidywalności wszystkich aspektów ich zastosowań w długim czasie.

# Trudności z oceną jakości oprogramowania

- Oceny jakości najczęściej muszą być znane zanim powstanie gotowy, działający produkt, co wyklucza zastosowanie obiektywnych metod pomiarowych.
- Wiele czynników składających się na jakość produktu jest niemierzalna.
- Produkty programistyczne są złożone i wieloaspektowe, co powoduje trudności w wyodrębnieniu cech mierzalnych, które odzwierciedlałyby istotne aspekty jakości.
- Produkty programistyczne mogą działać w różnych zastosowaniach, o różnej skali. Pomiary jakości mogą okazać się nieadekwatne przy zmianie skali (np. zwiększonej liczbie danych lub użytkowników), w innym środowisku, itp.

Pomiary mogą okazać się bardzo kosztowne, czasochłonne lub niewykonalne (z powodu niemożności stworzenia środowiska pomiarowego przed wdrożeniem);

Nie ma zgody co do tego, w jaki sposób pomierzone cechy danego produktu składają się na syntetyczny wskaźnik jego jakości.

**Stąd oceny jakości produktów programistycznych są skazane na metody spekulacyjne, oparte na uproszczeniach oraz dodatkowych założeniach, algorytmach, wzorach i heurystykach.**

# Kultura jakości

**To LUDZIE tworzą jakość!**

**DOSKONAŁOŚĆ i ORIENTACJA na WYNIK KOŃCOWY kluczem do SUKCESU**

**SUKCES** - Kiedy naprawdę wiemy, że odnieśliśmy sukces ?

**DOSKONAŁOŚĆ** - Co robimy, aby rzeczywiście stawać coraz się lepszymi?

**ORIENTACJA na WYNIK KONCOWY**

- W jaki sposób dbamy o to, aby dostarczać produkt zgodny z wymaganiami?

# Kultura jakości

***Jakość jest jak kultura osobista...***

***Trzeba o nią dbać i ją kształtować – inaczej przepadnie!***

**Ideał jakości może pokazać tylko pewien kierunek;**

**Cele jakości są celami projektu**

W przypadku zarządzania jakością w IT (i nie tylko ) widać wyraźny wpływ 2 podejść:

1. TQM ( total Quality management)
2. ISO serii 9000

# TQM - zarządzanie przez jakość

Koncepcja wymyślona przez Japończyka **Eiji Toyodę** dla potrzeb naprawy japońskiego przemysłu motoryzacyjnego - 1950 r.

**Główna jej myśl mówiła o tym, że w związku z tym, że to klient stanowi o rentowności przedsiębiorstwa, to należy tak sterować wszystkimi fazami procesu produkcyjnego wyrobu, aby klient był zadowolony z jakości tego wyrobu,**

TQM została sformalizowana przez Amerykanów (W.E.Deming, P.Crosby, J.M.Juran, A.V.Feigenbaum), Japończyków (E.Toyoda, M.Imai, K.Ishikawa) i Brytyjczyka J.Oaklanda,

Każdy z powyższych Autorów zdefiniował własne zasady TQM. Wszystkie one obracają się jednak wokół zasady Toyody: **„Jakość jest najważniejszym kryterium oceny przydatności produktów dla klienta, a to właśnie klient umożliwia funkcjonowanie wytwórcy tych produktów”.**

Stąd wniosek, że producent wytwarzający produkty kiepskie powinien wypaść z rynku.

# TQM w wytwarzaniu oprogramowania

Podstawowe działania:

1. Ciągłe badanie czy:

- Budujemy właściwy system ?
- Budujemy system dobrze ?
- Właściwy system budujemy dobrze od samego początku?

2. W razie problemów czy niepewności wprowadzamy działania modyfikujące

- produkt,
- ew. proces.

3. Badamy wyniki działań modyfikacyjnych.



# Cykl Deminga – podstawowy cykl zarządzania realizacją TQM, ISO 900...

**Cykl Deminga** (znany także jako **Koło Deminga**, **Cykl PDCA**) jest koncepcją zarządzania (jakością) opracowaną przez W.E. Deminga ( 14 zasad Deminga).

Koncepcja ta mówi o ciągłym doskonaleniu (pojęcie znane m.in. z TQM) przebiegającym w czterech następujących po sobie etapach: planowanie – wykonanie – sprawdzenie – poprawienie (ang. Plan – Do – Check – Act).

Poszczególne etapy polegają na:

1. **Planowanie** – w tym etapie określony zostają sposób działania, który doprowadzić ma do określonego celu.
2. **Wykonanie** – ten etap polega na wykonaniu wcześniej zaplanowanych działań.
3. **Sprawdzenie** – w tym etapie bada się wyniki wcześniej podjętych działań. Sprawdza się stopień wykonania celów zawartych w planie.
4. **Poprawienie** – na podstawie wniosków wyciągniętych podczas sprawdzania doskonalą się procesy oraz dostarcza pomysły i rozwiązania, które można zawrzeć w kolejnym planie.

źródło: [www.jakosc.biz/cykl-deminga/](http://www.jakosc.biz/cykl-deminga/)

# Jakość w terminologii ISO 9000

**jakość** - ogół cech i właściwości wyrobu lub usługi decydujący o zdolności wyrobu lub usługi do zaspokojenia stwierdzonych lub przewidywanych potrzeb użytkownika wyrobu;

**system jakości** - odpowiednio zbudowana struktura organizacyjna z jednoznacznym podziałem odpowiedzialności, określeniem procedur, procesów i zasobów, umożliwiającym wdrożenie tzw. *zarządzania jakością*;

**zarządzanie jakością** - jest związane z aspektem całości funkcji zarządzania organizacją, który jest decydujący w określaniu i wdrażaniu *polityki jakości*;

**polityka jakości** - ogół zamierzeń i kierunków działań organizacji dotyczących jakości, w sposób formalny wyrażony przez najwyższe kierownictwo organizacji, będącej systemem jakości;

**audyt jakości** - systematyczne i niezależne badanie, mające określić, czy działania dotyczące jakości i ich wyniki odpowiadają zaplanowanym ustaleniom, czy te ustalenia są skutecznie realizowane i czy pozwalają na osiągnięcie odpowiedniego *poziomu jakości*.

# Polityka i system jakości

**Polityka jakości** to ogólne intencje i zamierzenia danej organizacji w odniesieniu do jakości [ISO8402] wyrażana w sposób formalny przez zarząd firmy.

- Musi być zdefiniowana i udokumentowana;
- Muszą być określone cele i zaangażowanie w jakość;
- Musi być zgodna z działaniami przedsiębiorstwa i oczekiwaniami klienta;
- Musi być zakomunikowana i rozumiana na wszystkich szczeblach zarządzania.

**System jakości** to struktura organizacyjna, przydział odpowiedzialności, procedury postępowania, zasoby użyte do implementacji polityki jakości w danej organizacji [ISO8402]

- **pełnomocnik** lub zespół do spraw jakości;
- **księga jakości**: udokumentowane procedury systemu jakości.

# Model jakości ISO 9126

## 1. Funkcjonalność

- odpowiedniość
- dokładność
- współdziałanie
- zgodność
- bezpieczeństwo

## 2. niezawodność

- dojrzałość
- tolerancja błędów
- odtwarzalność

## 3. użyteczność

- zrozumiałość
- łatwość uczenia
- łatwość posługiwania się

## 4. efektywność

- charakterystyka czasowa
- wykorzystanie zasobów

## 5. Pielęgnowalność

- dostępność
- podatność na zmiany
- stabilność
- łatwość walidacji

## 6. Przeność

- dostosowywalność
- instalacyjność
- zgodność
- zamienność

*Atrybut jakościowy to cecha lub charakterystyka mająca wpływ na jakość danego wyrobu lub usługi*

# Zasady zarządzania jakością

Istnieje kilka głównych zasad zarządzania jakością. Oto one:

1. **Ukierunkowanie na klienta** (również klient wewnętrzny)
2. **Przywództwo** (budowa wizji, identyfikacja wartości)
3. **Zaangażowanie ludzi** (satysfakcja, motywacja, szkolenia)
4. **Podejście procesowe** (koncentracja na poszczególnych krokach procesu i relacjach pomiędzy tymi krokami, pomiary)
5. **Podejście systemowe** (całe otoczenie procesu wytwórczego - wewnątrz i na zewnątrz firmy - dostawcy)
6. **Ciągłe doskonalenie** (doskonalenie stanu obecnego, ewolucja a nie rewolucja)
7. **Rzetelna informacja** (zbieranie i zabezpieczanie danych do podejmowania obiektywnych decyzji)
8. **Partnerstwo dla jakości** (bliskie związki producentów z klientami)

# Zapewnienie Jakości Oprogramowania (ZJO) *software quality assurance, SQA*

Zgodnie z normą jest to „**planowany i systematyczny wzorzec wszystkich działań potrzebnych dla dostarczenia adekwatnego potwierdzenia że element lub produkt jest zgodny z ustanowionymi wymaganiami technicznymi**”.

**ZJO oznacza sprawdzanie:**

1. Czy zdefiniowano standardy i procedury ?
2. Czy plany są zdefiniowane zgodnie ze standardami ?
3. Czy procedury są wykonywane zgodnie z planami ?
4. Czy produkty są implementowane zgodnie z planami ?

Kompletne sprawdzenie jest zwykle niemożliwe. Projekty bardziej odpowiedzialne powinny być dokładniej sprawdzane odnośnie jakości.

W ramach ZJO musi być ustalony plan ustalający czynności sprawdzające przeprowadzane w poszczególnych fazach projektu.

Najbardziej istotnym kryterium przy zapewnianiu jakości jest **ryzyko**.

# Ryzyko utraty jakości

Najbardziej istotnym kryterium przy zapewnianiu jakości jest **ryzyko**. Najczęstszymi czynnikami ryzyka utraty jakości są:

- inowacyjność projektu,
- złożoność projektu,
- niedostateczne wykształcenie personelu,
- zbyt małe doświadczenie personelu,
- niesformalizowane (tworzone i zarządzane ad hoc) procedury
- niska dojrzałość organizacyjna wytwórcy.

**Dla zmniejszenia ryzyka personel ZJO powinien być zaangażowany w projekt programistyczny jak najwcześniej.**

Powinien on sprawdzać wymagania użytkownika, plany, procedury i dokumenty na zgodność ze standardami i przyjętymi procedurami postępowania.

Wynika to z faktu, że dodatkowe koszty związane z problemem lub błędem są tym większe, im później zostanie on zidentyfikowany.

# **Zadania zapewniania jakości**

## **Firma**

- ciągła pielęgnacja (doskonalenie) procesu wytwarzania
- definiowanie standardów
- nadzór i zatwierdzanie procesu wytwarzania

## **Przedsiębiorstwo (project)**

- dostosowywanie standardów
- ocena planów wytwarzania i jakościowych
- przeglądy przedsiębiorstwa
- testowanie i udział w inspekcjach
- audyt systemu zarządzania konfiguracją
- udział w Komitecie Sterującym projektem



# Procesy obsługiwane przez personel ZJO

## **Tworzenie standardowych wzorców produktów, działań, procesu**

- tworzenie standardu
- wdrażanie standardu

## **Kontrola jakości**

- ocena produktu
- ocena działania, procesu
- zatwierdzanie jakości

## **Analiza działalności firmy**

- zbieranie danych
- analiza danych

# Personel ZJO powinien ustalić, czy... (1)

- Projekt jest właściwie zorganizowany, z odpowiednim cyklem życiowym;
- Członkowie zespołu projektowego mają zdefiniowane zadania i odpowiedzialności;
- Plany w zakresie dokumentacji są implementowane;
- Dokumentacja zawiera to, co powinna zawierać;
- Przestrzegane są standardy dokumentacji i kodowania;
- Standardy, praktyki i konwencje są przestrzegane;
- Dane pomiarowe są gromadzone i używane do poprawy produktów i procesów;
- Przeglądy i audyty są przeprowadzane i są właściwie kierowane;
- Testy są specyfikowane i rygorystycznie przeprowadzane;

# Personel ZJO powinien ustalić, czy... (2)

- Problemy są rejestrowane i reakcja na problemy jest właściwa;
- Projekty używają właściwych metod, narzędzi i technik;
- Oprogramowanie jest przechowywane w kontrolowanych bibliotekach;
- Oprogramowanie jest przechowywane w chroniony i bezpieczny sposób;
- Oprogramowanie od zewnętrznych dostawców spełnia odpowiednie standardy;
- Rejestrowane są wszelkie aktywności związane z oprogramowaniem;
- Personel jest odpowiednio przeszkolony;
- Zagrożenia projektu są zminimalizowane.

# Zakres działań dla zapewnienia jakości

- Modele i miary służące ocenie kosztu i nakładu pracy
- Modele i miary wydajności ludzi
- Gromadzenie danych
- Modele i miary jakości
- Modele niezawodności
- Ocena i modelowanie wydajności oprogramowania
- Miary struktury i złożoności artefaktów: dokumentów, projektu, kodu
- Ocena dojrzałości technologicznej
- Zarządzanie z wykorzystaniem metryk
- Ocena metod i narzędzi

# Klasyfikacja zadań zapewnienia jakości

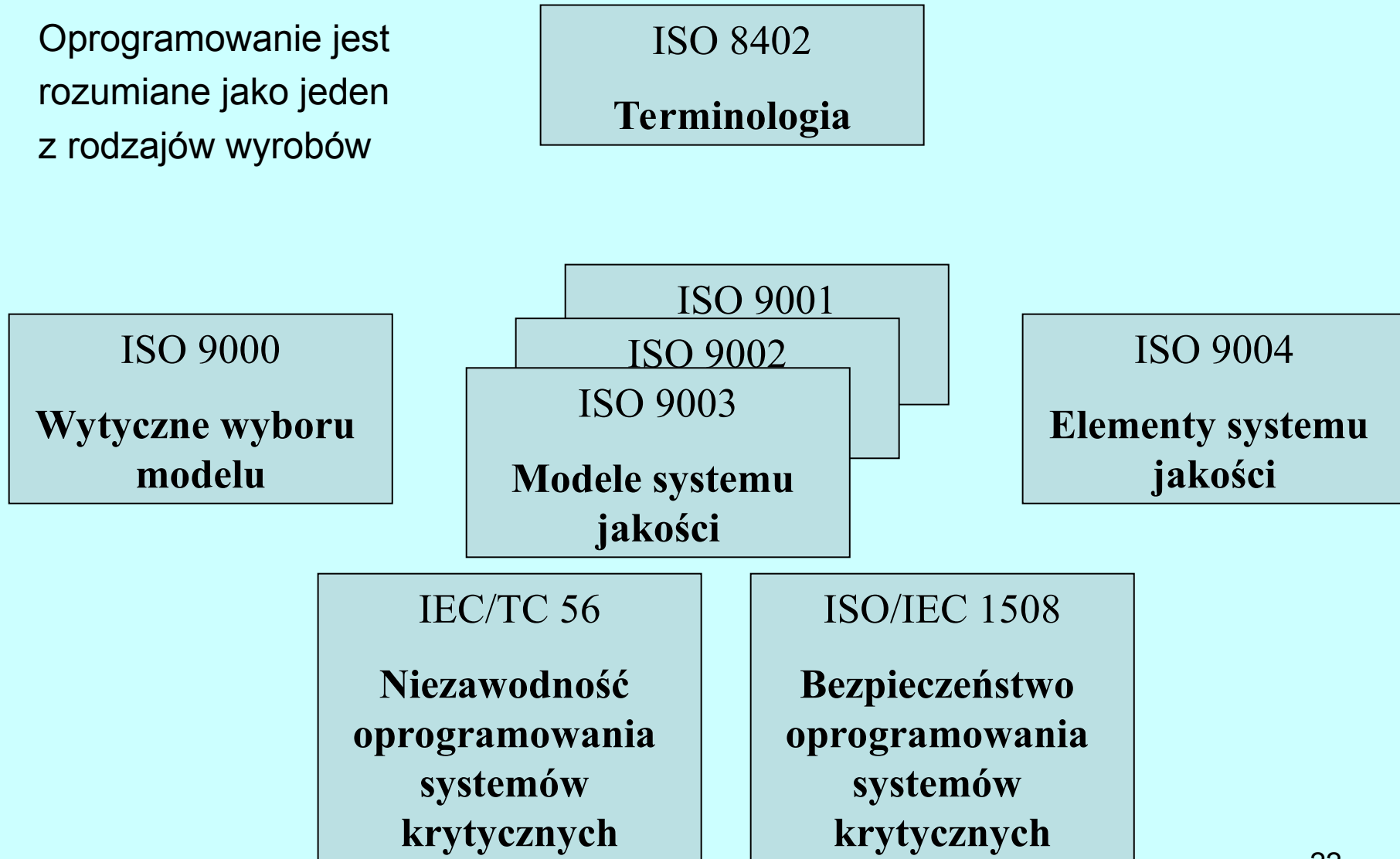
- Certyfikacja systemów przed skierowaniem do produkcji
- Wymuszanie standardów gromadzenia i przetwarzania danych
- Recenzowanie i certyfikacja wytwarzania i dokumentacji
- Opracowanie standardów dotyczących architektury systemu i praktyk programowania
- Recenzowanie projektu systemu pod względem kompletności
- Testowanie nowego lub zmodyfikowanego oprogramowania
- Opracowanie standardów zarządzania
- Szkolenie

## **Uwaga:**

**Pomiary odgrywają istotną rolę, jednakże są one postrzegane jako jedno z wielu specjalistycznych działań, a nie podstawa całego procesu zapewnienia jakości.**

# Normy dotyczące jakości

Oprogramowanie jest rozumiane jako jeden z rodzajów wyrobów



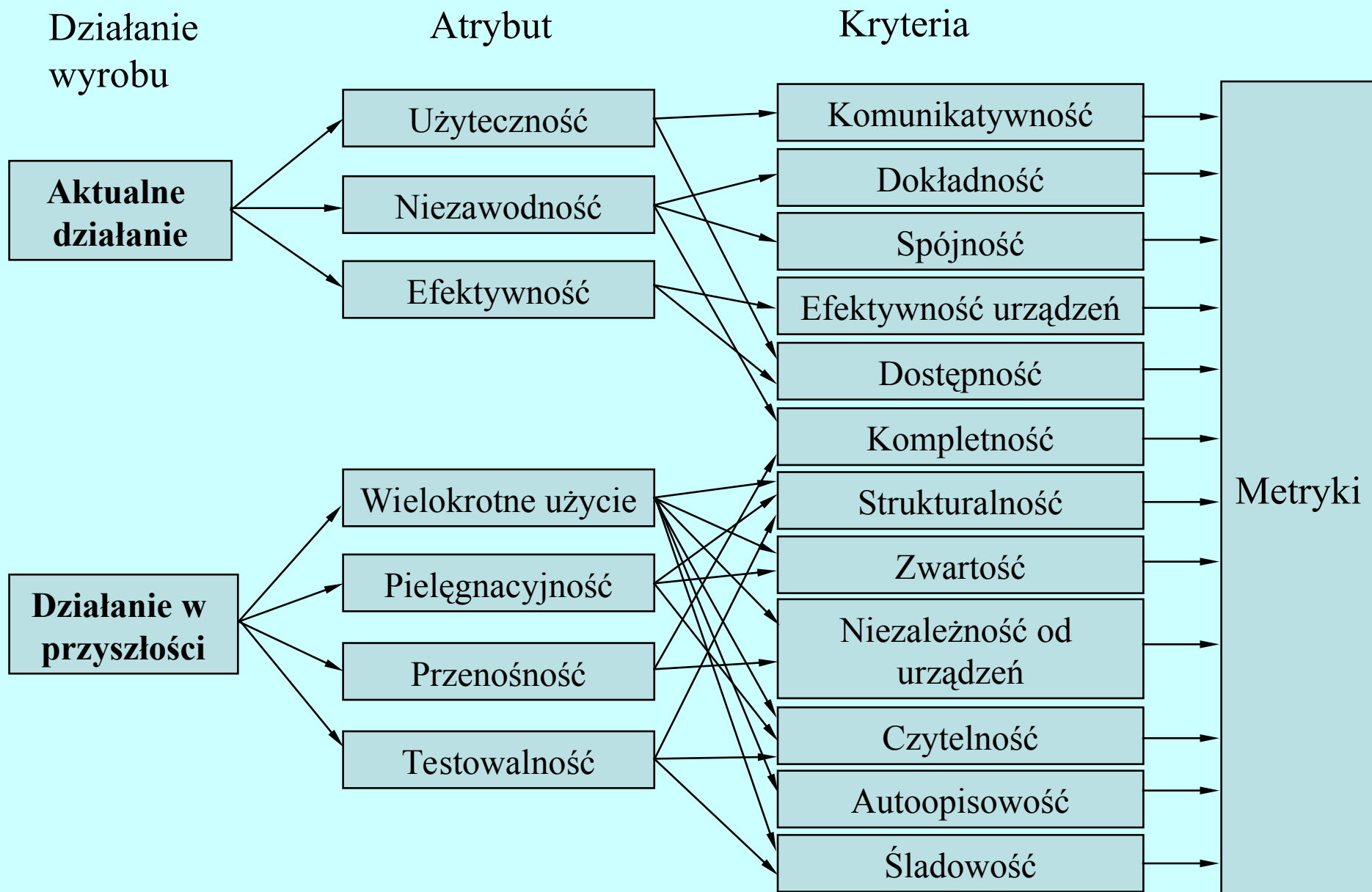
# Norma IEEE-730

Norma IEEE-730 podaje ogólne ramy planu zapewniania jakości. Powinien on obejmować następujące zagadnienia:

- analiza punktów widzenia
- referencje wykonawcy
- zarządzanie przedsięwzięciem informatycznym
- dokumentacja
- standaryzacja działań
- przeglądy i audyty
- zarządzanie konfiguracją oprogramowania
- raport napotykanym trudności i podjętych działań prewencyjnych
- wykorzystywane metody i narzędzia
- kontrola kodu, mediów, dostawców
- zarządzanie hurtowniami danych
- pielęgnacja

Norma IEEE-730 została uzupełniona i uszczegółowiona normą IEEE-983.

# Model jakości oprogramowania





# Dojrzałość procesów wytwórczych

## Niedojrzałość

1. Improwizacja podczas procesu wytwórczego
2. Proces jest wyspecyfikowany, ale specyfikacja nie jest stosowana
3. Doraźne reagowanie w sytuacji kryzysów
4. Harmonogram i budżet są przekraczane
6. Funkcjonalność jest stopniowo okrajana
7. Jakość produktu jest niska
8. Brak obiektywnych kryteriów oceny

## Dojrzałość

1. Zdolność do budowy oprogramowania jest cechą organizacji a nie personelu
2. Proces jest zdefiniowany, znany i wykorzystywany
3. Proces jest obserwowany i ulepszany
4. Prace są planowane i monitorowane
5. Role i odpowiedzialności są zdefiniowane
6. Obiektywna, ilościowa ocena

# CMM ( CMMI) - model dojrzałości procesu wytwórczego

## CMM – Capability Maturity Model

- Wykorzystywany w procedurach klasyfikacji potencjalnych wykonawców oprogramowania dla Departamentu Obrony USA
- Wyróżniono 5 poziomów dojrzałości wytwórców (poczynając od poziomu najniższego):
  - poziom początkowy - 1 (proces chaotyczny)
  - poziom powtarzalny - 2 (proces zindywidualizowany)
  - poziom zdefiniowany - 3 (proces zinstytucjonalizowany)
  - poziom zarządzany - 4 (proces + informacje zwrotne dla sterowania procesem)
  - poziom optymalizujący - 5 (proces + informacje zwrotne wpływające na na ulepszenie procesu )
- Niewiele firm uzyskało poziom 3-ci, umożliwiający uzyskanie prawa dostaw dla Departamentu Obrony USA. Poziom 5 posiadają: Motorola, Intel
- IBM w zakresie oprogramowania promu kosmicznego dla NASA uzyskał poziom 5-ty (najwyższy).

# Główne czynniki poprawy jakości

W zależności od występujących problemów dążymy do

- Poprawa zarządzania projektem
- Wzmocnienie inżynierii wymagań ( zarządzanie wymaganiami )
- Zwiększenie nacisku na kwalifikacje i wyszkolenie ludzi ( rola szkoleń !)

Możliwe do uzyskania wyniki dot. poprawy jakości

- Szybsze wykonywanie pracy (lepsze narzędzia, RAD, CASE) 10%
- Bardziej inteligentne wykonywanie pracy (lepsz organizacja i metody) 20%
- Powtórne wykorzystanie pracy już wykonanej (ponowne użycie) 65 %

# Plan zapewnienia jakości oprogramowania (PZJO)

**Plan zapewnienia jakości oprogramowania (PZJO) powinien być sporządzany i modyfikowany przez cały okres życia oprogramowania. Pierwsze jego wydanie powinno pojawić się na końcu fazy wymagań użytkownika.**

PZJO powinien ustalać i opisywać wszelkie aktywności związane z zapewnieniem jakości dla całego projektu. Odpowiednie sekcje planu jakości powinny dotyczyć wszystkich ustalonych w danym modelu rozwoju oprogramowania faz cyklu życia oprogramowania.

- Podane dalej zalecenia co do PZJO pochodzą z norm ISO serii 25 000
- Zalecany (podany dalej) spis treści może i niekiedy powinien być uzupełniony o punkty specyficzne dla konkretnego projektu.

# Styl, odpowiedzialność, sekcje PZJO

- **Styl.** PZJO powinien być zrozumiały, lakoniczny, jasny spójny i modyfikowalny.
- **Odpowiedzialność.** PZJO powinien być wyprodukowany przez komórkę jakości zespołu podejmujący się produkcji oprogramowania. PZJO powinien być przejrzany i recenzowany przez ciało, któremu podlega dana komórka jakości oprogramowania.
- **Medium.** Zwykle PZJO jest dokumentem papierowym. Może być także rozpowszechniony w formie elektronicznej.
- **Zawartość.** PZJO powinien być podzielony na 4 rozdziały, każdy dla następujących faz rozwoju oprogramowania:
  1. PZJO dla etapu wymagań użytkownika i analizy;
  2. PZJO dla etapu projektu architektury;
  3. PZJO dla etapu projektowania i konstrukcji;
  4. PZJO dla etapu budowy, testowania i instalacji oprogramowania.
- **Ewolucja.** PZJO powinien być tworzony dla następnej fazy po zakończeniu fazy poprzedniej.

# Spis treści PZJO (1)

## Informacje organizacyjne



- a - Streszczenie (maksymalnie 200 słów)
- b - Spis treści
- c - Status dokumentu (autorzy, firmy, daty, podpisy, itd.)
- d - Zmiany w stosunku do wersji poprzedniej

## Zasadnicza zawartość dokumentu



1. Cel
2. Referencje, odsyłacze do innych dokumentów
3. Zarządzanie
4. Dokumentacja
5. Standardy, praktyki konwencje i metryki
  - 5.1. Standardy dokumentacyjne
  - 5.2. Standardy projektowe
  - 5.3. Standardy kodowania
  - 5.4. Standardy komentowania
  - 5.5. Standardy i praktyki testowania
  - 5.6. Wybrane metryki pomocne przy ZJO
  - 5.7. Ustalenia dotyczące sposobu monitorowania zgodności z planem

*..... na następnym slajdzie*

# Spis treści PZJO (2)

Zasadnicza  
zawartość  
dokumentu



..... z poprzedniego slajdu

- 6. Przeglądy i audyty**
- 7. Testowanie**
- 8. Raportowanie problemów i akcje korygujące**
- 9. Narzędzia, techniki i metody**
- 10. Kontrola kodu**
- 11. Kontrola mediów**
- 12. Kontrola dostawców**
- 13. Zbieranie, pielęgnacja i utrzymanie zapisów**
- 14. Szkolenie**
- 15. Zarządzanie ryzykiem**
- 16. Przegląd pozostałej części projektu**
- Dodatek A: Słownik pojęć i akronimów**

Numeracja punktów nie powinna być zmieniana. Jeżeli pewien punkt nie ma treści, powinna tam znajdować się informacja „Nie dotyczy”.

Informacje nie mieszczące się w tym spisie treści powinny być zawarte w dodatkach. Punkty 3-15 powinny określać jak plany techniczne i zarządzania będą sprawdzane.

# Spis treści PZJO - omówienie (1)

- **Cel.** Sekcja ta powinna krótko określać: cel PZJO, rodzaj czytelnika, produkty programistyczne podlegające PZJO, zamierzone użycie oprogramowania, fazę cyklu życiowego, do którego PZJO się odnosi.
- **Zarządzanie.** Sekcja powinna opisywać organizację zarządzania jakością i związane z nią odpowiedzialności i role, bez określania przypisania ludzi do ról i bez określania pracochłonności i harmonogramu. Zalecana jest następująca struktura tego rozdziału:
  - **Organizacja:** identyfikacja ról w organizacji (kierownik projektu, prowadzący zespół, inżynierowie oprogramowania, bibliotekarze oprogramowania, prowadzący weryfikację i walidację, inżynier ZJO), opis związków pomiędzy rolami, opis interfejsu z organizacją użytkownika.
- **Zadania:** Opisuje zadania ZJO, które będą wykonywane w tej fazie.
- **Odpowiedzialności:** Opisuje odpowiedzialność poszczególnych ról za poszczególne zadania oraz ustala kolejność wybranych zadań.



# Spis treści PZJO - omówienie (2)

- **Dokumentacja.** Identyfikuje wszystkie dokumenty, które będą wyprodukowane w tej fazie. Sekcja powinna ustalać jak te dokumenty będą sprawdzane na zgodność ze standardami.
- **Standardy, konwencje metryki.** Opisuje je detalicznie lub zawiera odsyłacze do innych dokumentów.
- **Przeglądy i audyty.** Identyfikuje techniczne przeglądy, przejścia, inspekcje, audyty mające zastosowanie w tej fazie, oraz cel każdego z nich. Opisuje sposoby monitorowania zgodności tych procedur z planem oraz rolę personelu ZJO w tych procedurach.
- **Testy.** Opisuje w jaki sposób czynności weryfikacji i walidacji oprogramowania będą monitorowane i w jaki sposób będą sprawdzane testy akceptacyjne .
- **Raportowanie problemów i akcje korygujące.** Identyfikuje procedury zgłaszania problemów oraz podejmowania akcji mających na celu usunięcie problemów. Może opisywać metryki stosowane do procedur zgłaszania problemów mające wpływ na jakość oprogramowania.

# Spis treści PZJO - omówienie (3)

- **Kontrola kodu.** Procedury stosowane do pielęgnacji, przechowywania, zabezpieczania i dokumentowania kodu oprogramowania.
- **Kontrola mediów.** j.w., ale dotyczy mediów, na których oprogramowanie i dokumentacja będą przechowywane.
- **Kontrola dostawców.** Procedury stosowane do kontroli zewnętrznych organizacji lub osób, które rozwijają lub dostarczają oprogramowanie niezbędne dla projektu. Procedury powinny określać standardy, które mają być stosowane przez dostawców, oraz powinny określać sposoby kontroli przestrzegania tych standardów.
- **Zbieranie, pielęgnacja i utrzymanie zapisów.** Identyfikuje procedury stosowane do przechowywania informacji zebranych ze wszelkich aktywności, takich jak spotkania, przeglądy, przejścia, audyty, sporządzanie notatek, korespondencji. Powinny określać gdzie te informacje/dokumenty są przechowywane i jak długo, oraz określać sposób dostępu do tych informacji.

# INŻYNIERIA OPROGRAMOWANIA

**Dziękuję za uwagę**