

Inżynieria Oprogramowania

Specyfikacja



Inżynieria Oprogramowania

W poprzednim odcinku ...

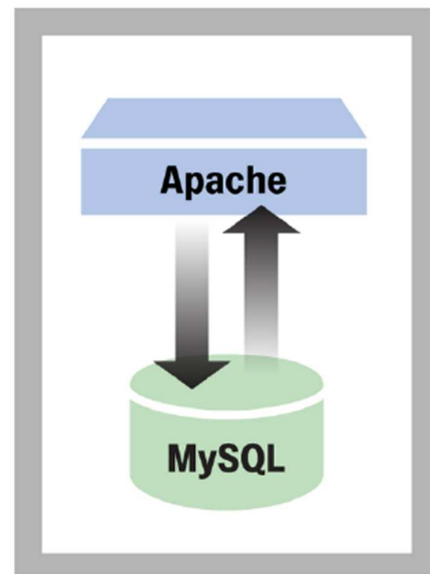


ISO/IEC 42010

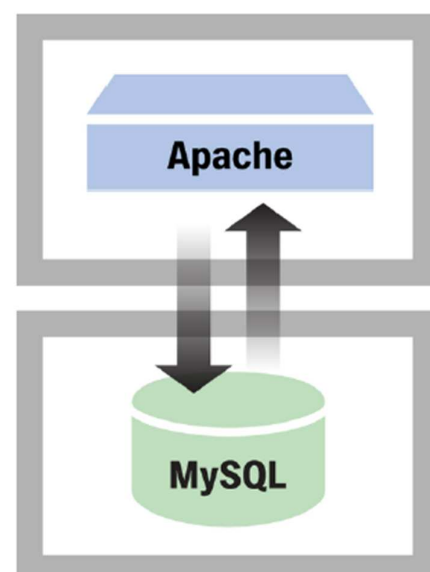


Architecture: the fundamental concepts or properties of a system in its environment embodied in its elements, their relationships, and in the principles of its design and evolution

- Dwie warstwy:
 - Klient
 - przeglądarka
 - Serwer
 - LAMP = Linux + Apache + MySQL + PHP

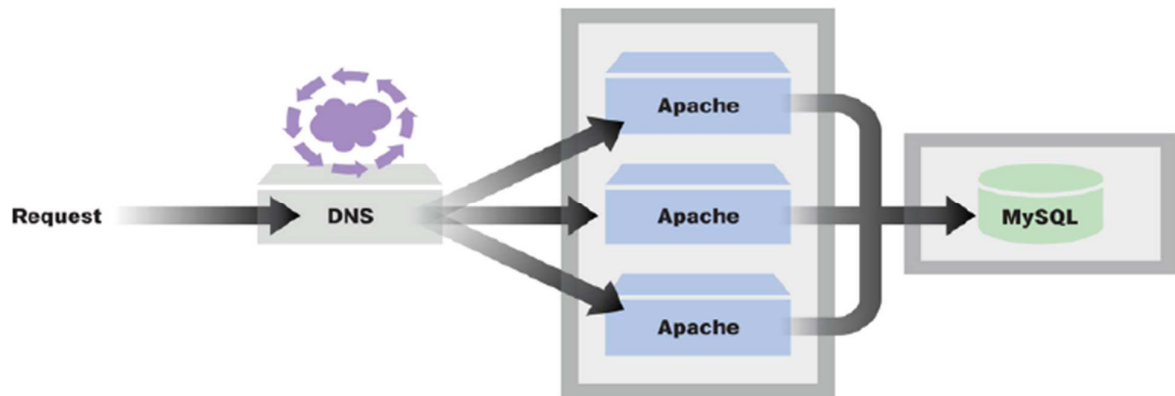


- Trzy warstwy
 - Klient
 - przeglądarka
 - Serwer aplikacji
 - Linux + Apache
 - Baza danych
 - Linux + MySQL



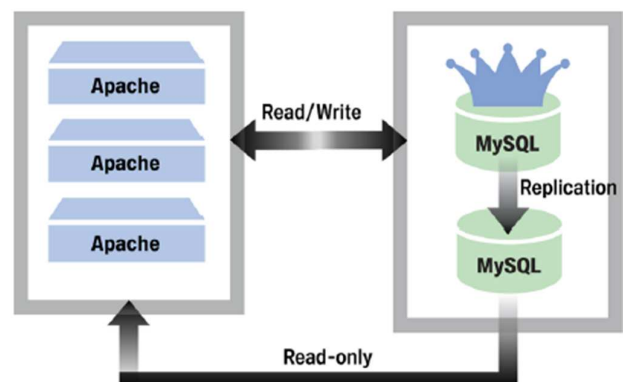
■ Trzy warstwy z częściowym HA

- Klient
 - przeglądarka
- Klaster serwerów aplikacji
 - wiele serwerów Apache
- Baza danych
 - jeden serwer MySQL



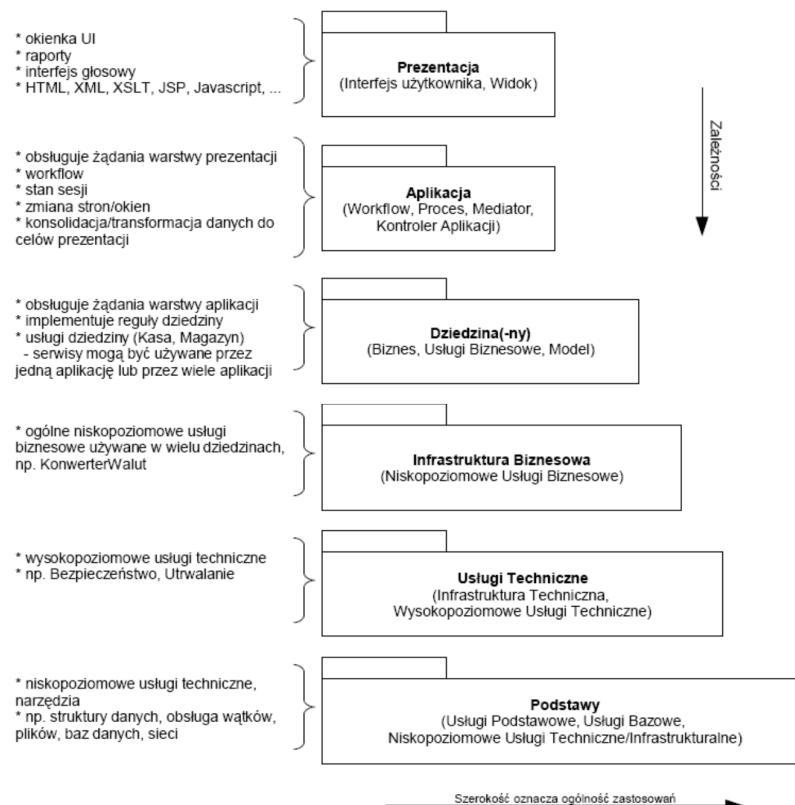
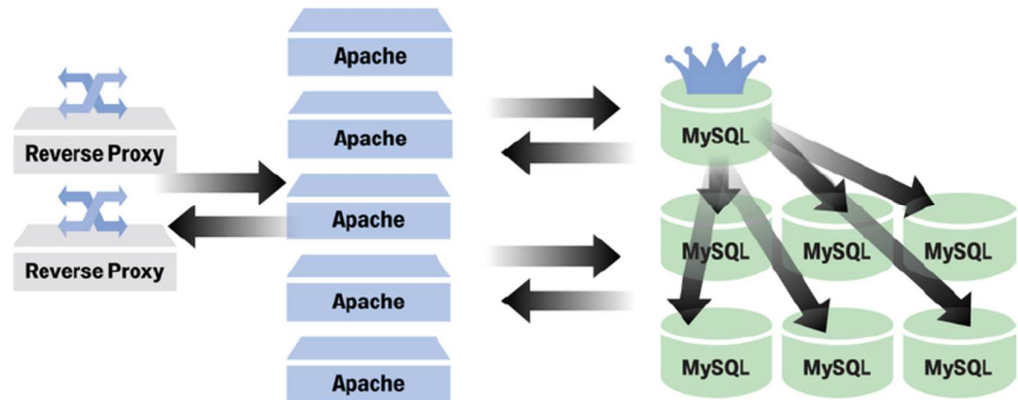
■ Trzy warstwy z nieco większym HA

- Klient
 - przeglądarka
- Klaster serwerów aplikacji
 - wiele serwerów Apache
- Klaster baz danych
 - jeden serwer MySQL do zapisu
 - wiele serwerów MySQL do odczytu

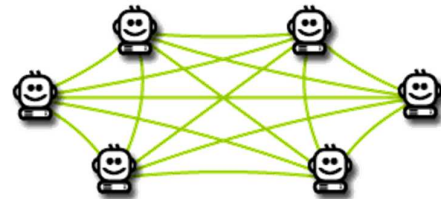


■ Trzy warstwy z pełnym HA

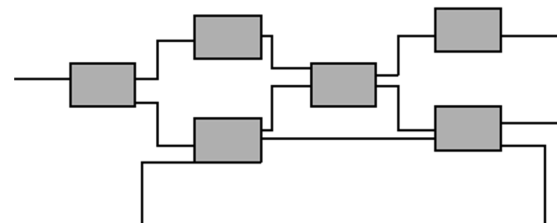
- Klient
 - przeglądarka
- Klaster serwerów aplikacji
 - wiele serwerów Apache
- Klaster baz danych
 - wiele serwerów MySQL (do zapisu / odczytu)

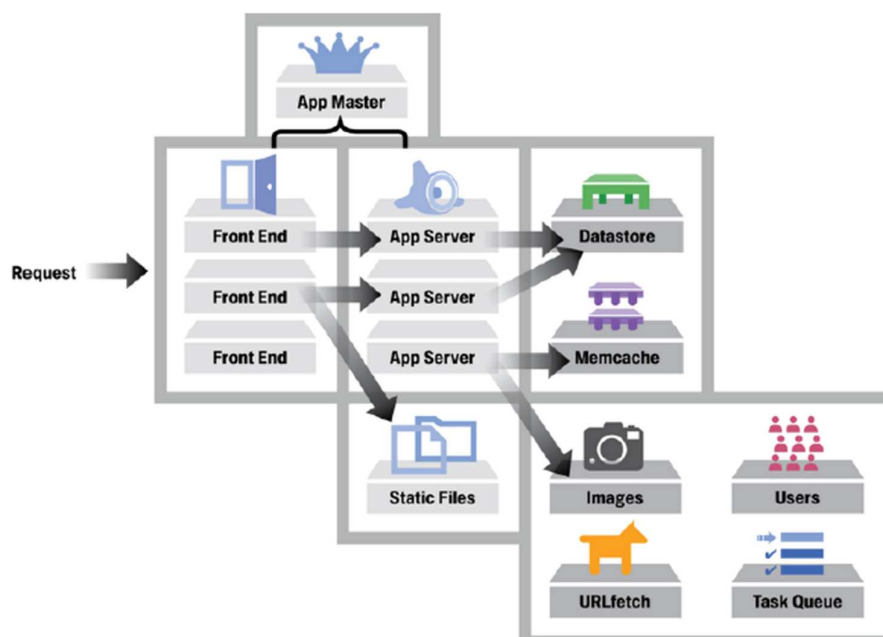
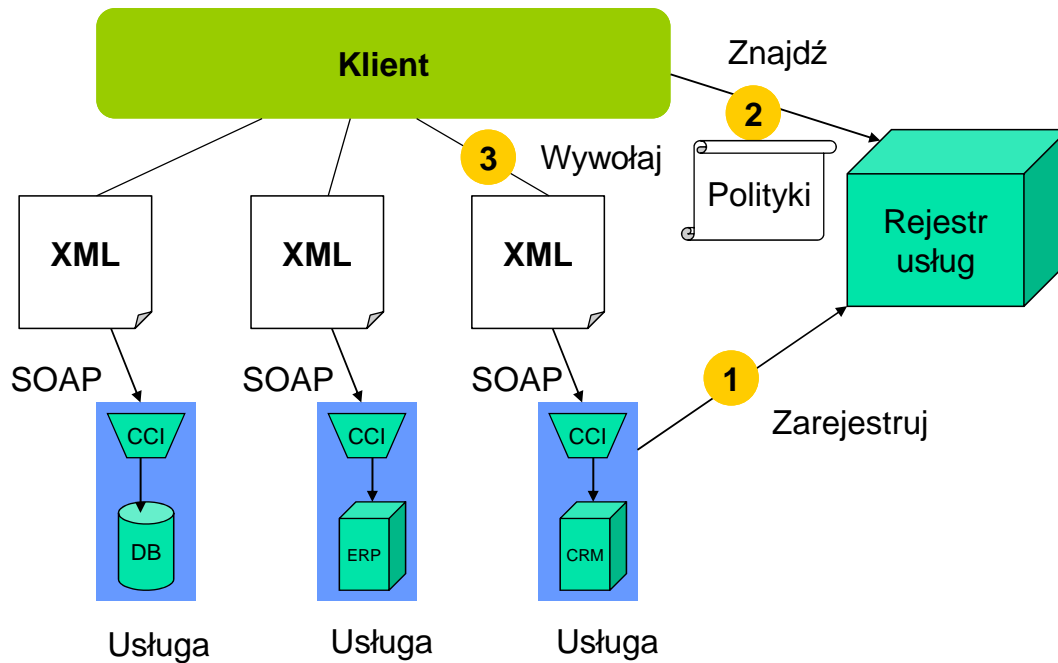


- Peer-to-peer
 - każdy jest klientem i serwerem



- Pipe-and-filter
 - Przetwarzanie potokowe







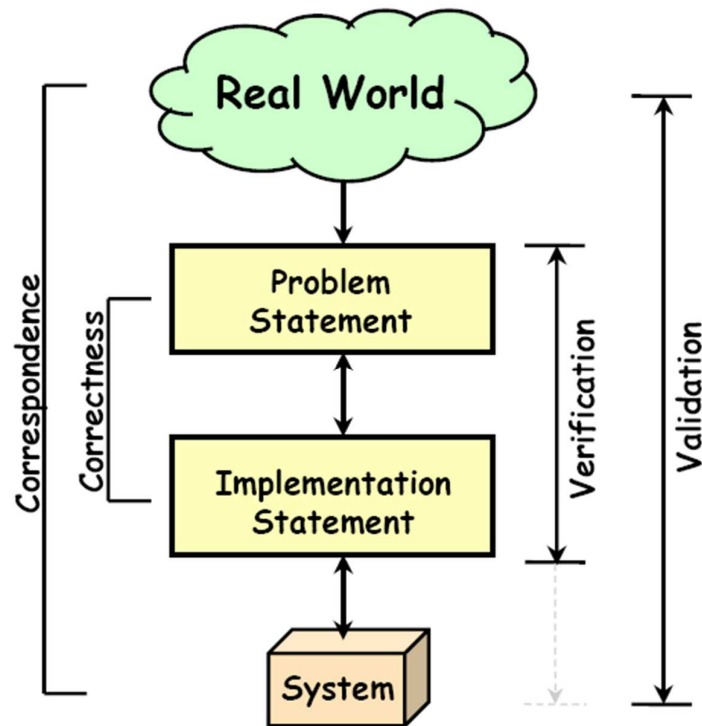
Google App Engine



ORACLE®

IBM

<http://dilbert.com/strips/comic/2011-01-07/>



	<i>Agreement between</i>	
	<i>Producer</i>	<i>Consumer</i>
<i>Requirements Specification</i>	Development Contractor	Purchaser
<i>Design Specification</i>	Implementor	System architect
<i>Module specification</i>	Programmer writing the module	Programmer using the module

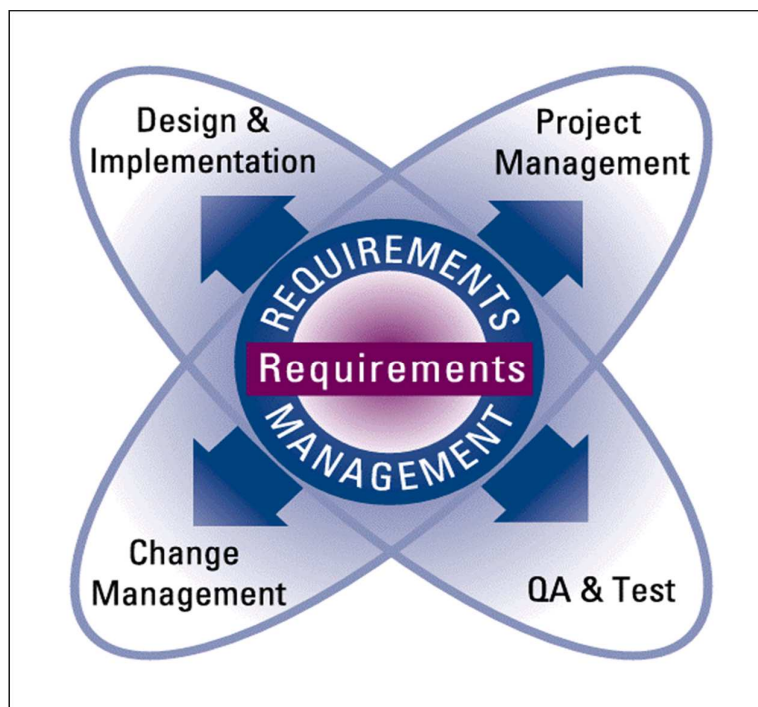
- Specyfikacja jest umową
 - pomiędzy **dostawcą** (np. usługi) ...
 - ... a **zamawiającym** (np. usługi)
- Oprogramowanie **musi** być wyspecyfikowane gdy
 - istnieje niebezpieczeństwo **niezrozumienia** (lub zapomnienia) o potrzebach klienta
 - reprezentowane są potrzeby **więcej niż jednej** osoby
 - **więcej niż jedna** osoba wytwarza oprogramowanie

Czy specyfikacja oznacza dużo formalizmu?

- Manifesto for Agile Software Development
 - Individuals and interactions over processes and tools
 - Working software over comprehensive documentation
 - Customer collaboration over contract negotiation
 - Responding to change over following a plan
- Principles behind the Agile Manifesto
 - Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
 - Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
 - Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
 - Business people and developers must work together daily throughout the project.
 - Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
 - The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
 - Working software is the primary measure of progress.
 - Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
 - Continuous attention to technical excellence and good design enhances agility.
 - Simplicity – the art of maximizing the amount of work not done – is essential.
 - The best architectures, requirements, and designs emerge from self-organizing teams.
 - At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

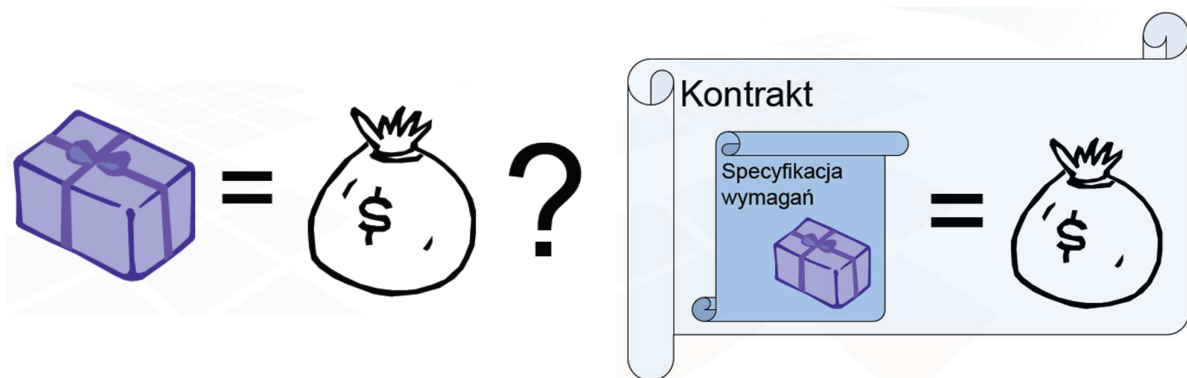
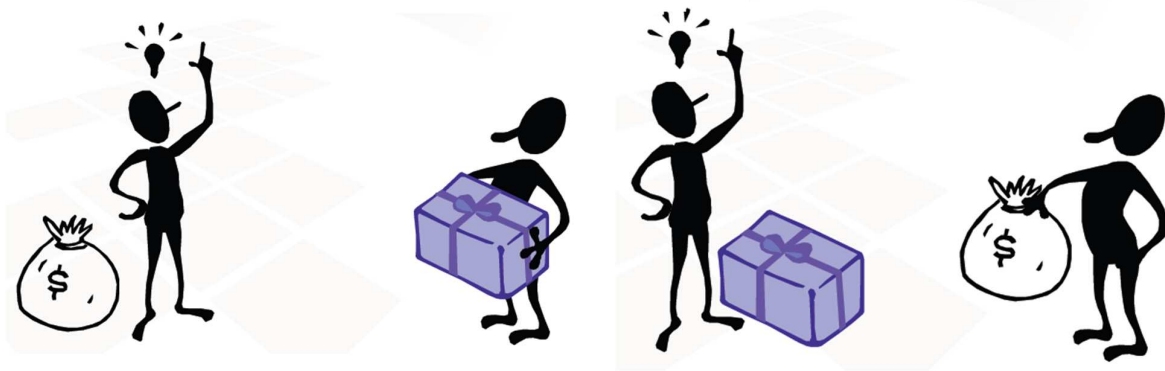
17

Specyfikowanie wymagań



18

Jeden z powodów zajmowania się specyfikacją wymagań



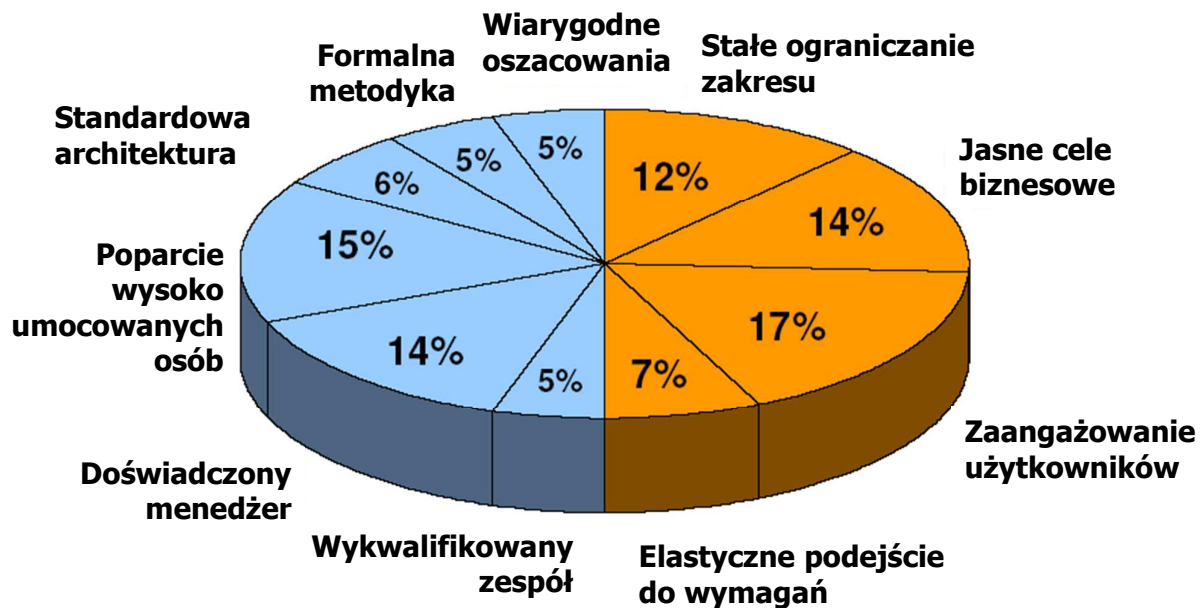
19

Kolejny powód zajmowania się specyfikacją wymagań

- Co jest ważniejsze dla sukcesu projektu? (uporządkuj)
 - A. Doświadczony menedżer
 - B. Elastyczne podejście do wymagań
 - C. Formalna metodyka
 - D. Jasne cele biznesowe
 - E. Poparcie (dla projektu) wysoko umocowanych osób
 - F. Stałe ograniczanie zakresu
 - G. Standardowa architektura
 - H. Wiarygodne oszacowania
 - I. Wykwalifikowany zespół
 - J. Zaangażowanie użytkowników

20

Kolejny powód zajmowania się specyfikacją wymagań



Przyczyny udanych projektów informatycznych

21

Specyfikowanie wymagań

- Jest
 - procesem komunikacji
- pomiędzy
 - klientami / zamawiającymi / użytkownikami
- a
 - twórcami
- systemów informatycznych

- Jest prawdopodobnie
 - największym wyzwaniem
- dla procesu wytwarzania
 - dużych i złożonych
- systemów informatycznych

22

- Wymagania określają

- CO

- zamiast

- jak

- Wymagania określają

- zadania systemu

- a nie

- sposób ich implementacji

- Specyfikacja to

- zbiór wymagań

- IEEE definiuje wymaganie jako:

- 1) *A condition or capability needed by a user to solve a problem or achieve an objective*
- 2) *A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document*
- 3) *A documented representation of a condition or capability as in 1) or 2)*

- Telefon komórkowy

- duży wyświetlacz
 - wygodne klawisze
 - aparat o wysokiej rozdzielczości
 - pojemna bateria
 - ...

- Wskazane wymagania są nieprecyzyjne

- duży, wygodne, pojemne, ...

- Przyjęto wiele założeń niejawnych

- klawiatura **musi** być po tej samej stronie co ekran

Czy mogę wydrukować zdjęcie w formacie A3?

Czy ten wyświetlacz jest duży?

Czy klawiatura nie mogłaby być po drugiej stronie?



Zatem jak przeprowadzać specyfikacje?

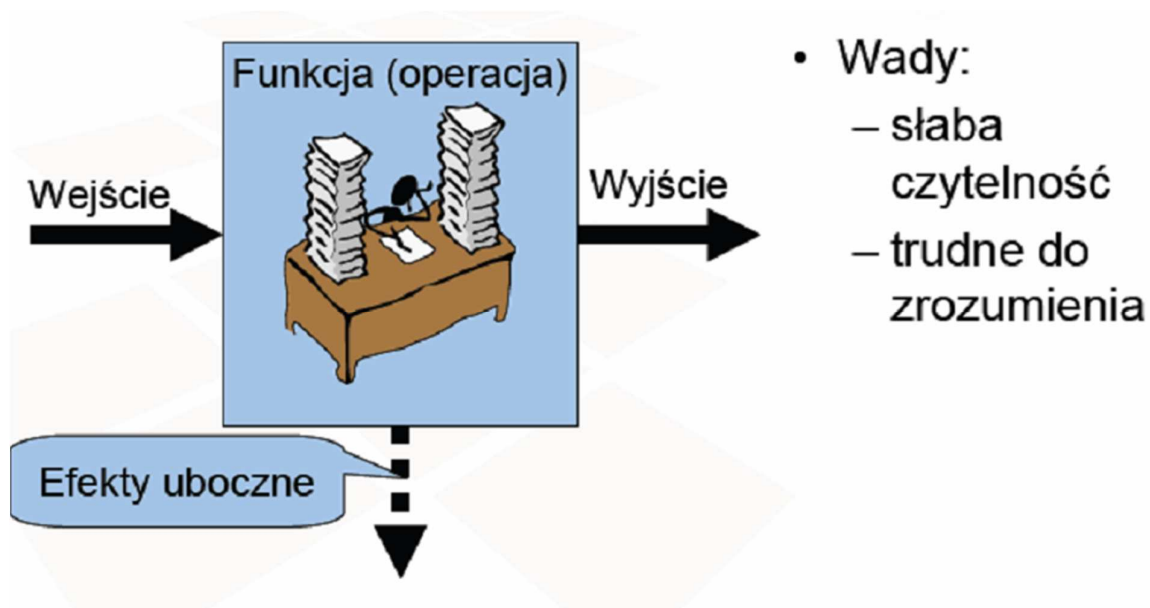
- ~~Zatem:~~
 - ~~spisanie wymagań~~
- ~~=~~
 - ~~dostarczenie systemu zgodnie z potrzebami~~
- ~~?~~
- Niestety nie:
 - Słowa:
 - wieloznaczność
 - Wiedza:
 - świadoma
 - nieświadoma
 - ...
- Specyfikowanie wymagań
 - to
- sztuka
- Nie ma
 - uniwersalnego sposobu
- ale istnieją
 - dobre praktyki

25

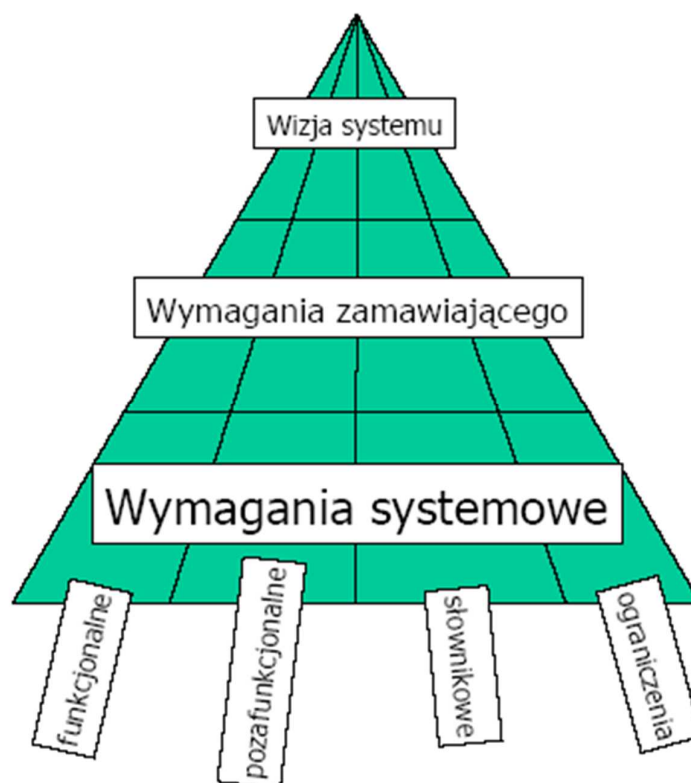
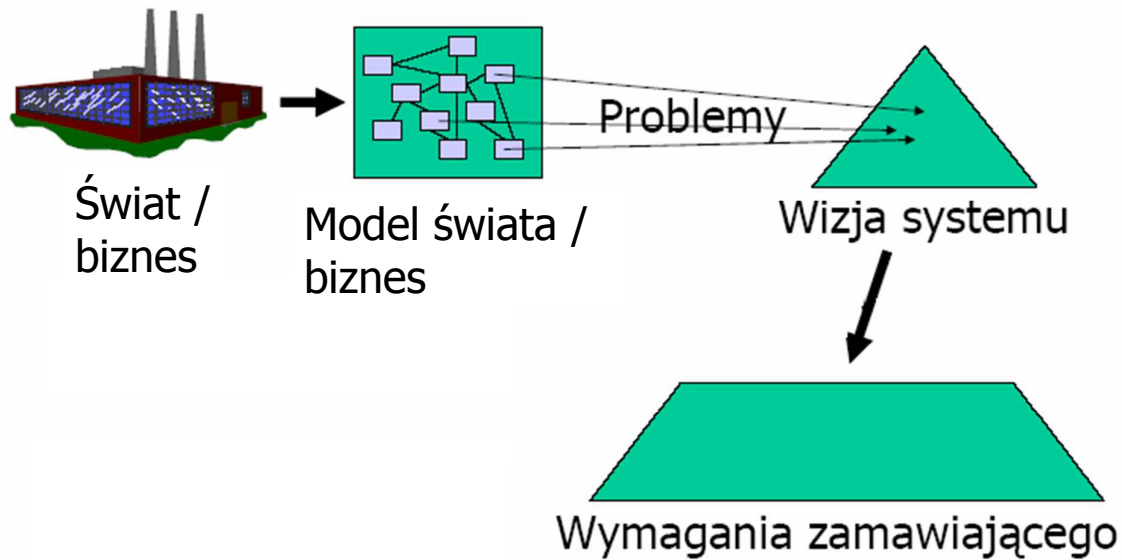
Można specyfikować systemy tak...

- Zalety:
 - łatwość spisywania
 - Wady:
 - słaba czytelność
 - trudne sprawdzanie kompletności, spójności
- System powinien umożliwić wystawianie faktur
 - System powinien generować zestawienie miesięczne faktur
 - Faktura powinna zawierać co najmniej jedną pozycję
 - ...
 - (tak przez kolejnych 200 stron)

26



- Ważne etapy procesu:
 - **Wydobywanie** wymagań w oparciu o model biznesu
 - **Analiza i negocjacja** wymagań
 - **Zatwierdzanie** wymagań
 - Zarządzanie **zmianami** wymagań



Wizja



Procesy biznesowe

- czynności poza systemem
- automatyzacja

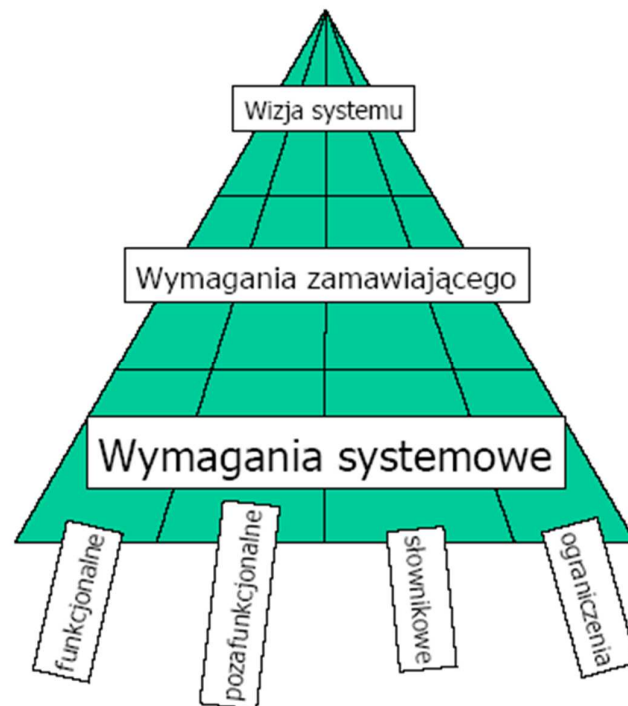


Przypadki użycia

- aktorzy
- scenariusz główny
- scenariusze alternatywne
- model zależności



Specyfikacja pozafunkcjonalna



■ Składniki wizji systemu

- Postawienie problemu
- Motto dla systemu
- Osoby zainteresowane, kluczowi użytkownicy
- Główne cechy oprogramowania
- Główne ograniczenia środowiska

FURPS

F unctionality

U sability

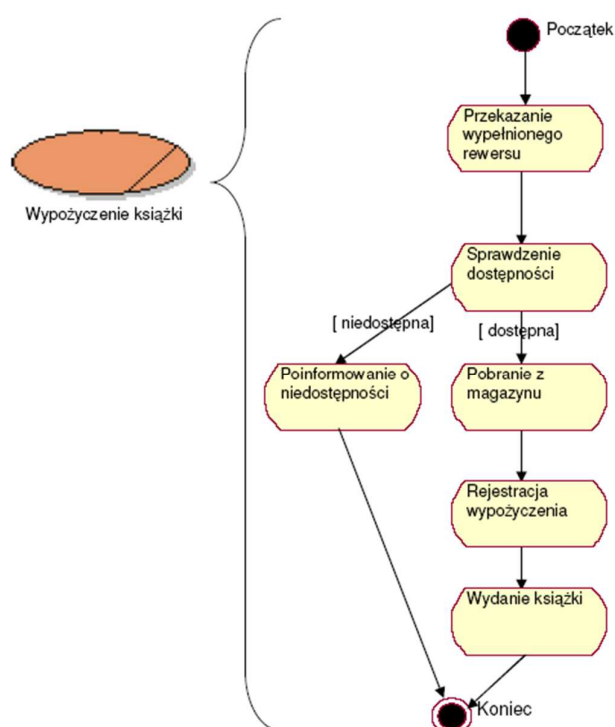
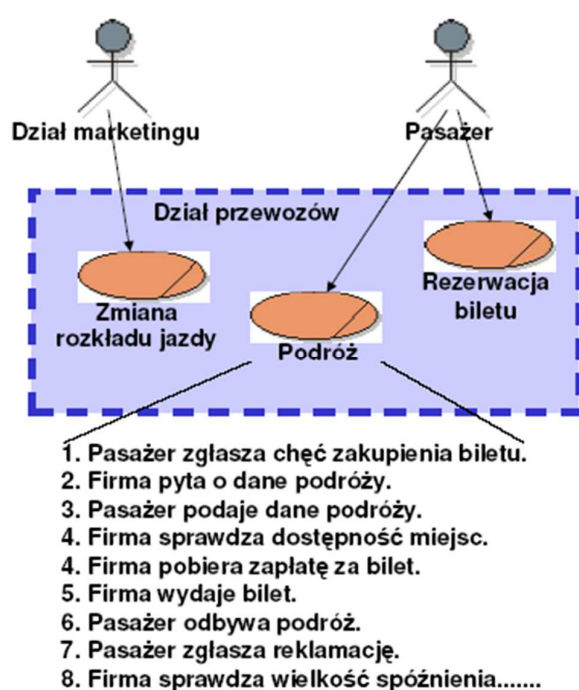
R eliability

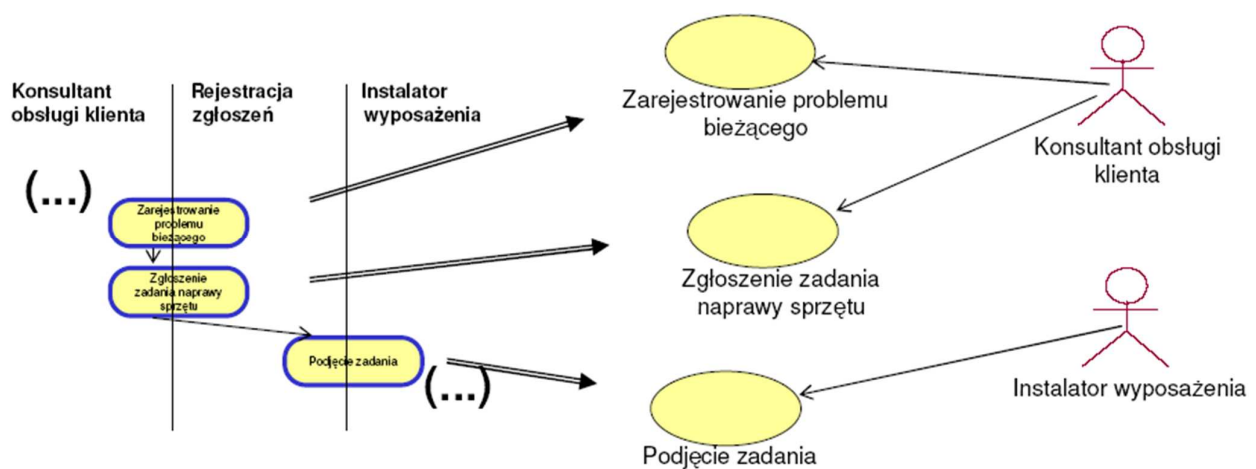
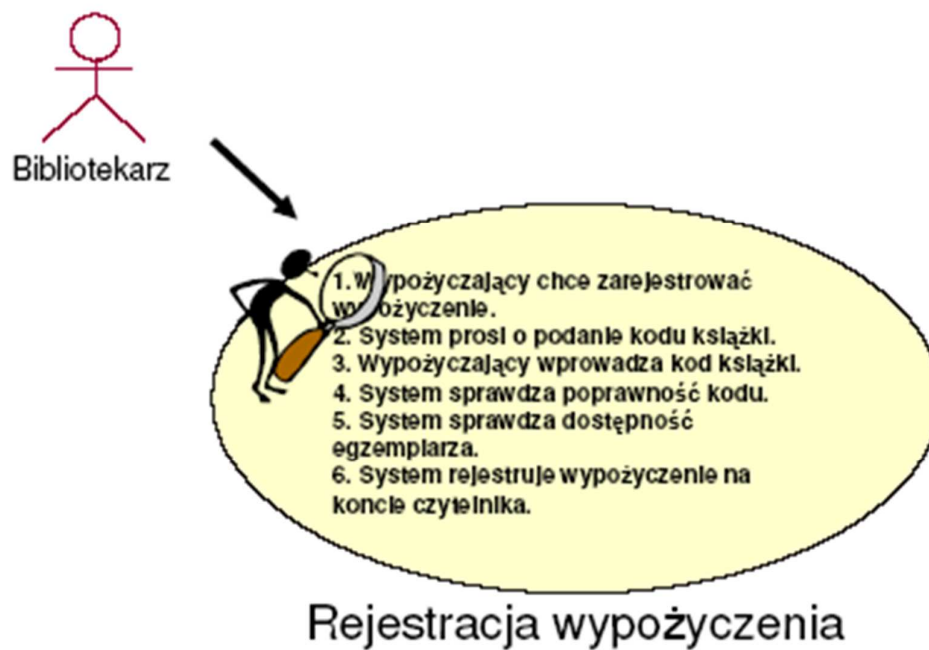
P erformance

S ecurity

■ Wymagania funkcjonalne

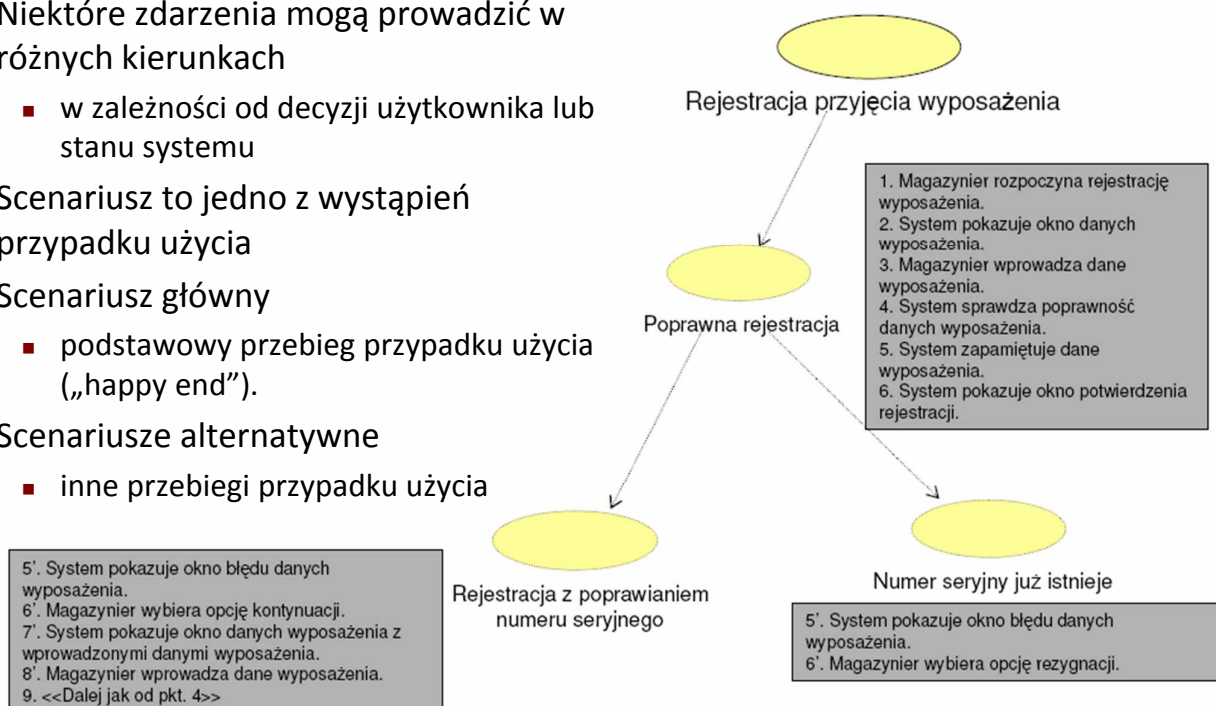
■ Wymagania pozafunkcjonalne





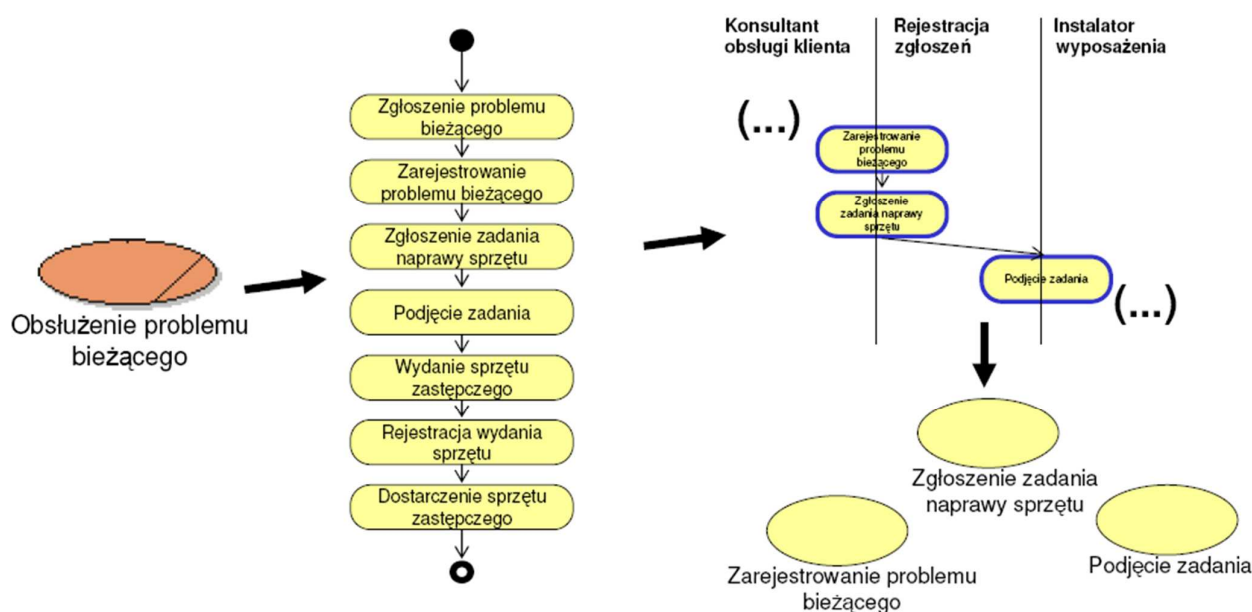
Przypadek użycia – scenariusze

- Niektóre zdarzenia mogą prowadzić w różnych kierunkach
 - w zależności od decyzji użytkownika lub stanu systemu
- Scenariusz to jedno z wystąpień przypadku użycia
- Scenariusz główny
 - podstawowy przebieg przypadku użycia („happy end”).
- Scenariusze alternatywne
 - inne przebiegi przypadku użycia



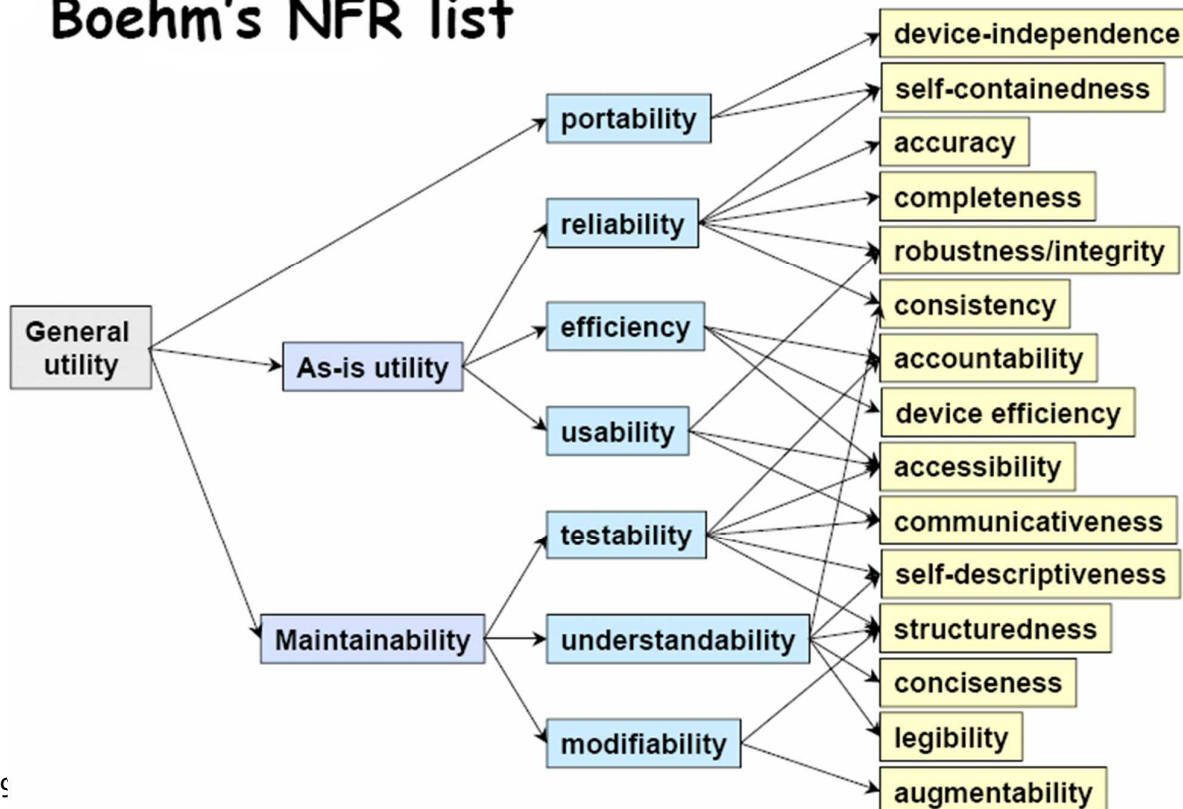
37

Specyfikacja funkcjonalna – podsumowanie



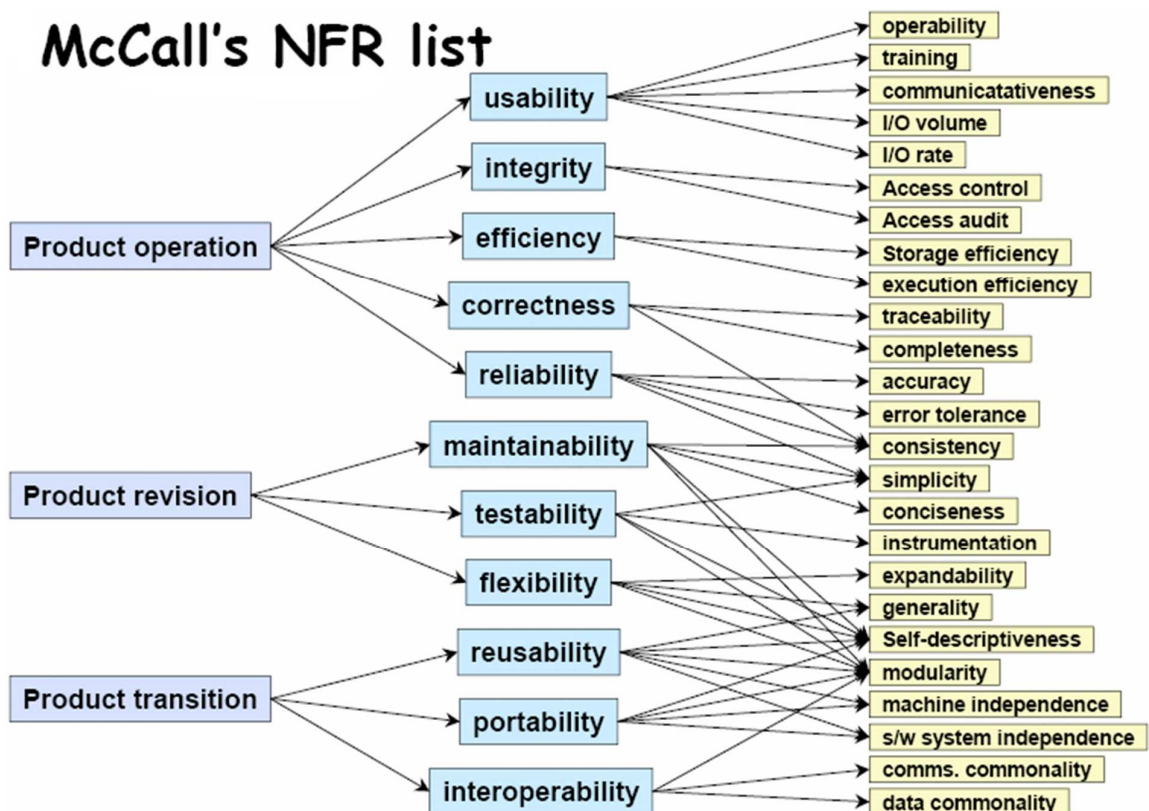
38

Boehm's NFR list



35

McCall's NFR list



40

- Fraza czasownikowa w nazwie
- Aktor, scenariusz i rozszerzenia
 - Nadrzędny cel: czytelność!
 - Scenariusz główny – najbardziej prawdopodobna ścieżka
 - Alternatywne scenariusze – kiedy coś pójdzie nie tak
- Obojętność technologiczna
 - technologia jest zmienna
 - niepotrzebne ograniczenia
 - szczegóły GUI zaciemniają obraz
 - klient nie rozumie terminów technicznych
- Historia możliwa do opowiedzenia na różnych poziomach szczegółów
 - Hierarchiczne scenariusze
 - Można rozwijać lub zwijać w celu pokazania lub ukrycia szczegółów

- Poprawność:
 - Czy rzeczywiście tak ma działać system?
 - Wyraża faktyczne wymagania
- Jednoznaczność:
 - Czy wymagania mają tylko jedną interpretację?
 - Każde stwierdzenie może być odczytane w dokładnie jeden sposób
 - Pojęcia mylące są definiowane w słowniku
 - Jest przejrzysta (zrozumiała dla nie-informatyków)
- Kompletność
 - Czy zamieszczono wszystkie wymagania funkcjonalne i pozafunkcjonalne?
 - Określa wszystkie rzeczy, jakie system musi robić
 - ...oraz wszystkie rzeczy, których nie może robić
 - Odpowiada na wszystkie typy danych wejściowych
 - Brak elementów wymagających uzupełnienia (kompletność strukturalna)
- Konieczność
 - Czy wszystkie wymagania są istotne dla zamawiającego?
 - Nie zawiera niczego, co nie jest rzeczywiście potrzebne

- Spójność
 - Czy wymagania zamawiającego są zgodne z wizją i niesprzeczne ze sobą?
 - Nie zaprzecza samej sobie (np. jest spełnialna)
 - Jednolicie korzysta z pojęć
- Możliwość porządkowania wymagań
 - Czy każde wymaganie ma atrybuty umożliwiające określenie wagi i znaczenia?
- Weryfikowalność:
 - Czy każde wymaganie ma przyporządkowane kryterium jakości, które można przetestować?
 - Istnieje procedura pozwalająca na przetestowanie spełnienia każdego z wymagań
 - Każde wymaganie jest opisane deklaratywnie
- Modyfikowalność
 - Czy można w łatwy sposób wprowadzać zmiany do specyfikacji wymagań?
 - Dobrze zorganizowana
 - Minimalna redundancja
 - Możliwość śledzenia
- Utrzymywanie śladu
 - Czy wymagania posiadają identyfikatory i wynikają bezpośrednio z wymagań wyższego poziomu?

