


**Politechnika
Warszawska**

**Specyfikowanie wymagań
jakościowych**

dr hab. Michał Śmiałek, prof. uczelni

 **Wydział
Elektryczny**
POLITECHNIKA WARSZAWSKA

1DI1615: Inżynieria wymagań oprogramowania



1

Treść wykładu

- Sposób opisu wymagań jakościowych
- Rodzaje wymagań jakościowych
- Metryki wymagań jakościowych
- Ograniczenia techniczne i środowiskowe
- Wpływ wymagań jakościowych

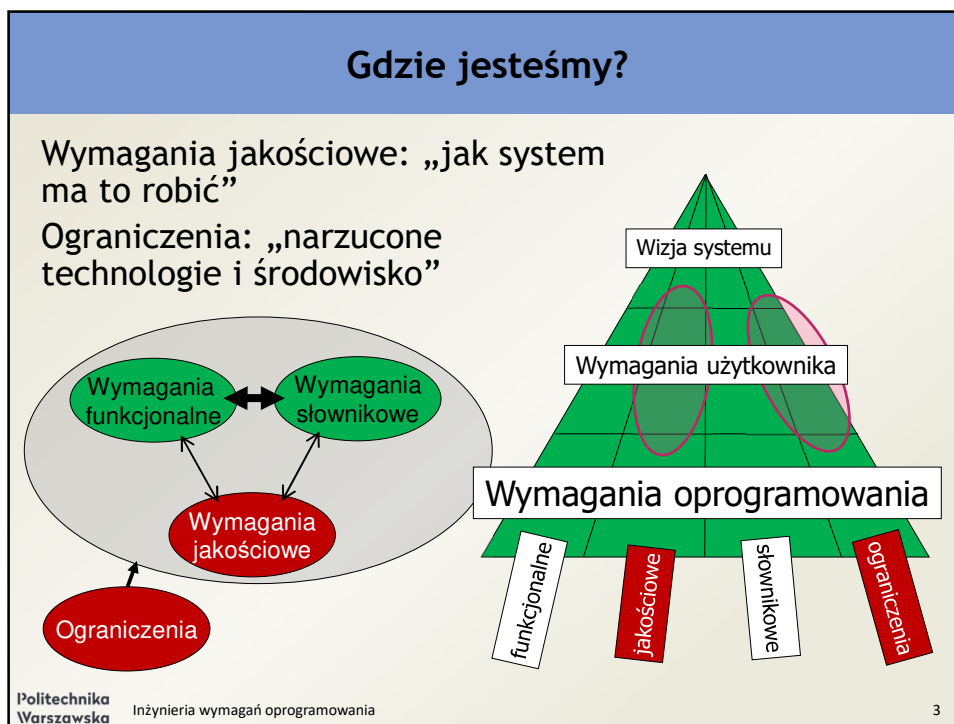






**Politechnika
Warszawska** Inżynieria wymagań oprogramowania

2



3



4

Formułowanie wymagań jakościowych

- Proponowany sposób zapisu wymagań jakościowych
 - stosowanie prostych zdań zawierających orzeczenia modalne oparte o czasowniki takie jak „musieć”, „być”, „zawierać” oraz dopełnienia precyzyjnie zdefiniowane w słowniku
- Można stosować na wszystkich poziomach piramidy wymagań (wizja, wymagania użytkownika, wymagania oprogramowania)
- Przykłady:
 - System będzie przechowywał dane klientów w formie zaszyfrowanej
 - System musi pokazywać wartości wagowe z dokładnością do 1 mg
 - System powinien wyświetlać mapę terenu przeciętnie w 2 sekundy
- Przykłady (strona bierna):
 - Minimalny czas dostępności systemu w ciągu doby będzie wynosił 23 godziny
 - Formularze będą umożliwiały szybkie wprowadzanie danych adresowych
 - Użytkownicy będą w stanie szybko nauczyć się sprawnego wprowadzania transakcji

Kompletny opis wymagania jakościowego

- Nazwa wymagania w postaci jednego zdania,
- Standardowe atrybuty wymagania,
- Rodzaj wymagania zgodnie z wybraną metodą klasyfikacji wymagań,
- Opis - dokładniejsze informacje o właściwości obejmowanej danym wymaganiem jakościowym,
- Metryka wymagania:
 - przedstawienie sposobu pomiaru,
 - możliwe wyniki pomiaru
 - akceptowalne wartości pomiaru.

Fiszka wymagania jakościowego

- Opis wymagania w narzędziu CASE
- Opis wymagania w dokumencie

| | | | |
|--|---|-------------------------------------|------------------------------|
| ID: identyfikator | Nazwa: Zdanie typu „System powinien ...” | | |
| Wersja: numer | Waga klienta: wartość wagi | Waga wykonawcy: wartość wagi | |
| STATUS | Trudność: liczba | Wydanie: numer | Odpowiedzialny: osoba |
| Rodzaj: identyfikacja rodzaju wymagania zgodnie z wybraną klasyfikacją | | | |
| Opis: akapit opisujący dokładniej właściwości jakościowe systemu | | | |
| Sposób pomiaru: opis zawierający krótkie przedstawienie procedury pomiarowej | | | |
| Możliwy wynik pomiaru: definicja jednostek i możliwego zakresu uzyskanych wyników | | | |
| Oczekiwane wartości: zakres wartości pomiaru, który oznacza akceptację wymagania | | | |

Przykłady opisu wymagań jakościowych (1)

| | | | |
|---|---|--------------------------------|------------------------------|
| ID: JU012 | Nazwa: System powinien umożliwiać szybkie dodawanie samochodów | | |
| Wersja: 1.0 | Waga klienta: ważne | Waga wykonawcy: średnie | |
| STATUS | Trudność: 2 | Wydanie: 0.21 | Odpowiedzialny: Magda |
| Rodzaj: użyteczność – operacyjność | | | |
| Opis: Średni czas dodania do systemu nowego samochodu nie powinien przekroczyć 2 minut, a maksymalnie może być dwa razy wyższy. Dotyczy przypadku użycia PU076 . | | | |
| Sposób pomiaru: Mierzmy czas, jaki zajmie magazynierowi dodanie nowego samochodu, czyli przejście przez główny scenariusz PU076. Pomiaru dokonujemy na próbie 10 osób po 30 minutach przeszkolenia. Każda osoba dodaje po 10 samochodów. | | | |
| Możliwy wynik pomiaru: Czas mierzymy z dokładnością do 5 sekund w zakresie od 5 sekund do 10 minut (powyżej 10 minut uznaje się, że magazynier nie był w stanie dodać samochodu). | | | |
| Oczekiwane wartości: Średni czas dla całej grupy wynosi mniej niż 2 minuty. Najgorszy czas wynosi poniżej 4 minut. | | | |

Przykłady opisu wymagań jakościowych (2)

| | | | |
|--|---|--------------------------------|-------------------------------|
| ID: JE021 | Nazwa: System musi szybko wyświetlać ekran podsumowania czynności magazynowych | | |
| Wersja: 1.1 | Waga klienta: kluczowe | Waga wykonawcy: średnie | |
| STATUS | Trudność: 7 | Wydanie: 0.21 | Odpowiedzialny: Stefan |
| Rodzaj: efektywność wydajnościowa – czas | | | |
| Opis: Czas, w którym system wyświetla ekran podsumowania czynności magazynowych musi wynosić średnio poniżej 2 sek. Magazynier nie powinien czekać dłużej niż 5 sekund przy najgorszym obciążeniu systemu. Dotyczy przypadku użycia PU081, krok 5 . | | | |
| Sposób pomiaru: Mierzmy czas od momentu wybrania opcji wyświetlania ekranu podsumowania (krok 4) do momentu pełnego wyświetlenia ekranu (krok 5). Pomiaru dokonujemy przy różnych poziomach obciążenia systemu zgodnie z zasadą OB011. Wyniki mierzymy dla 100 prób na każdy poziom obciążenia. | | | |
| Możliwy wynik pomiaru: Czas mierzymy z dokładnością do 0,1 sekund w zakresie od 0,1 sekund do 1 minuty (powyżej 1 minuty uznaje się, że system nie wyświetlił prawidłowo ekranu). | | | |
| Oczekiwane wartości: Średni czas dla wszystkich poziomów obciążenia wynosi mniej niż 2 sekundy. Najgorszy czas dla poziomu obciążenia 5 wynosi mniej niż 5 sekund. | | | |

Przykłady opisu wymagań jakościowych (3)

| | | | |
|---|--|-------------------------------|-------------------------------|
| ID: JK023 | Nazwa: System musi umożliwiać współpracę z systemem CEPIK | | |
| Wersja: 1.0 | Waga klienta: kluczowe | Waga wykonawcy: trudne | |
| STATUS | Trudność: 21 | Wydanie: 0.21 | Odpowiedzialny: Stefan |
| Rodzaj: kompatybilność – interoperacyjność | | | |
| Opis: System będzie umożliwiał wymianę danych o sprowadzanych i sprzedawanych samochodach z systemem CEPIK. Wymiana odbywać się będzie poprzez API dostarczane przez system CEPIK. Dotyczy wszystkich przypadków użycia w relacji z aktorem System CEPIK . | | | |
| Sposób pomiaru: W trakcie testów akceptacyjnych (TA056 do TA071) sprawdzamy działanie systemu pod kątem współpracy z API systemu CEPIK. Sprawdzamy prawidłowość przesyłu danych o pojazdach oraz rejestracjach pojazdów. Przesył danych sprawdzamy dla różnych prędkości łącza oraz dla różnych poziomów obciążenia. | | | |
| Możliwy wynik pomiaru: Liczba prawidłowo przesłanych rekordów w stosunku do wszystkich prób, w rozbiu na prędkości łącza oraz poziomy obciążenia (0-100%). | | | |
| Oczekiwane wartości: Dla prędkości łącza powyżej 1Mb/s oraz poziomu obciążenia poniżej 3, poziom przesyłu wynosi 100%. Dla niższych prędkości oraz od 3 poziomu w górę, poziom przesyłu wynosi co najmniej 98%. W przypadku przesyłów nieprawidłowych, błąd jest zarejestrowany we wszystkich przypadkach. | | | |



11

Standardy klasyfikacji wymagań jakościowych

- **FURPS:** Functionality, Usability, Reliability, Performance, Supportability
- **ISO/IEC 9126-1:2001** Software engineering – Product quality
- **ISO/IEC 25010:2011** Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models

12

Rodzaje wymagań jakościowych (ISO 9126)

Standard ISO/IEC 9126-1:2001 wyróżnia 6 rodzajów wymagań jakościowych

- **Funkcjonalność** (ang. Functionality)
– UWAGA: nie mylić z wymaganiami funkcjonalnymi!
- **Niezawodność** (ang. Reliability)
- **Użyteczność** (ang. Usability)
- **Wydajność** (ang. Efficiency)
- **Łatwość utrzymania** (ang. Maintainability)
- **Przeność** (ang. Portability)

Rodzaje wymagań jakościowych (ISO 25010)

Nowszy standard ISO/IEC 25010:2011 wyróżnia 8 rodzajów wymagań jakościowych

- **Przydatność funkcjonalna** (ang. Functional suitability)
- **Efektywność wydajnościowa** (ang. Performance efficiency)
- **Kompatybilność** (ang. Compatibility)
- **Użyteczność** (ang. Usability)
- **Niezawodność** (ang. Reliability)
- **Bezpieczeństwo** (ang. Security)
- **Łatwość utrzymania** (ang. Maintainability)
- **Przeność** (ang. Portability)

Rodzaj wymagań: Przydatność funkcjonalna

Czy system, używany w określonych warunkach, zapewnia funkcje, które spełniają wyrażone lub domniemane potrzeby.

- **Kompletność funkcjonalna** (ang. functional completeness) – stopień, w jakim zbiór funkcji pokrywa wszystkie wyspecyfikowane zadania oraz cele użytkowników
- **Poprawność funkcjonalna** (ang. functional correctness) – stopień, w jakim system zapewnia prawidłowe rezultaty z wymaganym poziomem dokładności
- **Adekwatność funkcjonalna** (ang. functional appropriateness) – stopień, w jakim funkcje ułatwiają wykonanie zadań i osiągnięcie celów

Przykłady: Przydatność funkcjonalna

- **Kompletność funkcjonalna**
 - System będzie walidowany poprzez pokrycie funkcjonalności testami w stopniu zgodnym z metodyką XYZ
 - System będzie zapewniał wystarczającą operacyjność przy zrealizowaniu wszystkich funkcji o najwyższym priorytecie i XY% funkcji o średnim priorytecie
- **Poprawność funkcjonalna**
 - System musi zapewniać najwyższą poprawność reakcji na zdarzenia krytyczne poprzez zwielokrotnienie odpowiednich podsystemów decyzyjnych
 - System musi zapewniać dokładność pomiarów czasu na poziomie 0,001 sek
- **Adekwatność funkcjonalna**
 - System musi zapewniać funkcjonalność wystarczającą do prawidłowej obsługi wydawania towarów z magazynu

Rodzaj wymagań: Efektywność wydajnościowa

Zbiór atrybutów opisujących powiązania między poziomem wydajności oprogramowania, a wykorzystywanymi zasobami (konfiguracji sprzętu i oprogramowania oraz materiałów eksploatacyjnych) przy spełnieniu określonych warunków

- **Czas** (ang. time behaviour) – stopień, w jakim czasy odpowiedzi (przetwarzania danych) i wskaźniki przepustowości systemu podczas wykonywania jego funkcji spełniają wymagania
- **Zużycie zasobów** (ang. resource utilization) – stopień, w jakim ilości różnych typów zasobów (technologicznych) używanych przez system podczas wykonywania jego funkcji, spełniają wymagania
- **Pojemność** (ang. capacity) – stopień, w jakim maksymalne limity parametrów systemu (liczba przechowywanych elementów, liczba jednoczesnych użytkowników, rozmiar przestrzeni danych itp.) spełniają wymagania

Przykłady: Efektywność wydajnościowa

- **Czas**
 - System powinien minimalizować czas otwierania bramek wejściowo/wyjściowych aby utrzymać płynny ruch osób
 - System musi zapewnić przepustowość na poziomie 500 transferów pełnowymiarowych zdjęć medycznych na minutę
- **Zużycie zasobów**
 - System będzie obciążał zasoby obliczeniowe maszyn wykonawczych w maksymalnie 50%
- **Pojemność**
 - System będzie zapewniał przechowywanie co najmniej 5 mln rekordów transakcji finansowych z możliwością rozszerzenia do 10 mln po 5 latach

Rodzaj wymagań: Kompatybilność

Zdolność do wymiany informacji z innymi produktami, systemami lub komponentami i/lub wykonywania zadanych funkcji współdzieląc ten sam sprzęt lub środowisko software'owe

- **Współistnienie** (ang. co-existence) - stopień, w jakim system może efektywnie wykonywać swoje funkcje podczas dzielenia wspólnego środowiska i zasobów z innymi systemami, bez negatywnego wpływu na działanie
- **Interoperacyjność** (ang. interoperability) - stopień, w jakim dwa lub więcej systemów mogą wymieniać i używać wymieniane informacje

Przykłady: Kompatybilność

- **Współistnienie**
 - System musi zachowywać zadane poziomy wydajności w sytuacji, gdy w tym samym środowisku wykonawczym jest uruchomiony w trybie ciągłym system analizatora ruchu
- **Interoperacyjność**
 - System powinien mieć możliwość odczytywania danych o stanie magazynu z systemem magazynowym
 - System powinien udostępniać interfejs programisty, umożliwiający odczyt oraz zmianę parametrów analizatora ruchu przez zewnętrzne aplikacje mobilne

Rodzaj wymagań: Użyteczność

- **Rozpoznawalność odpowiedności** (ang. appropriateness recognizability) - możliwość rozpoznania przez użytkowników czy produkt jest odpowiedni dla ich potrzeb; np. rozpoznanie na podstawie pierwszego wrażenia z użycia, dostarczonej dokumentacji, demonstracji, strony internetowej
- **Łatwość nauczania** (ang. learnability) - miara wysiłku, jaki musi włożyć użytkownik do nauczania się nieznanych lub nowych funkcji aplikacji w celu osiągnięcia niezbędnej efektywności działania i/lub satysfakcji
- **Operacyjność** (ang. operability) - nakład pracy użytkownika niezbędny do obsługi i kontroli funkcji realizowanych przez oprogramowanie
- **Zabezpieczenie przed błędami użytkownika** (ang. user error protection)
- **Estetyka interfejsu użytkownika** (ang. user interface aesthetics)
- **Przystępność** (ang. accessibility) - stopień dostosowania systemu do potrzeb użytkowników ze specjalnymi potrzebami (np. osoby niepełnosprawne)

Przykłady: Użyteczność

- **Rozpoznawalność odpowiedności**
 - System będzie zapewniał możliwość zautomatyzowanego przejścia przez proces konfiguracji funkcji analizy w celu szybkiego rozpoczęcia działania podsystemu analizatora
- **Łatwość nauczania**
 - System musi zapewniać możliwość osiągnięcia odpowiedniej wydajności rejestracji zdarzeń magazynowych po trzygodzinnym przeszkoleniu
- **Operacyjność**
 - System musi zapewniać dostęp do funkcji analizatora skupień bezpośrednio z panelu głównego aplikacji użytkownika
 - System musi umożliwiać rejestrację nowego towaru przez wyszkolonego pracownika w czasie krótszym o 20% niż dla starego systemu
- **Zabezpieczenie przed błędami użytkownika**
 - System musi zapobiegać wprowadzeniu przez użytkownika parametrów analizatora skupień spoza prawidłowego zakresu
- **Estetyka interfejsu użytkownika**
 - System musi zapewniać zgodność kolorystyki interfejsu użytkownika z księgą identyfikacji wizualnej firmy
- **Przystępność**
 - System musi umożliwiać dostosowanie rozmiaru czcionek do użytkowników o stopniu ubytku wzroku XYZ

Rodzaj wymagań: Niezawodność

Zdolność do zapewnienia określonego poziomu wydajności, przy spełnieniu określonych warunków oraz w zadanym przedziale czasu

- **Dojrzałość** (ang. maturity) - poziom zapewnienie niezawodnego działania (brak defektów) systemu w normalnych warunkach
- **Dostępność** (ang. availability) - poziom dostępności i operacyjności działania systemu, kiedy jego działanie jest wymagane
- **Odporność na błędy** (ang. fault tolerance) - zdolność do stabilnego funkcjonowania nawet w przypadku wystąpienia błędu lub niestabilnego działania sprzętu lub oprogramowania systemowego
- **Odtwarzalność** (ang. recoverability) - zdolność oprogramowania do przywrócenia stabilnego i wydajnego działania po awarii oraz przywrócenia stanu danych sprzed awarii

Przykłady: Niezawodność

- **Dojrzałość**
 - System powinien zapewniać wystąpienie co najwyżej jednego błędu klasy 1 w miesiącu
- **Dostępność**
 - System będzie dostępny w trybie 24/7 z możliwością okresowych przerw w działaniu - co najwyżej 4 godziny w miesiącu w godzinach nocnych
- **Odporność na błędy**
 - System będzie zapewniał prawidłowe działanie nawet w przypadku unieruchomienia połowy maszyn wykonawczych
- **Odtwarzalność**
 - System musi zapewniać średni czas między awariami na poziomie N23 (1 awaria klasy 1 a rok)
 - System musi zapewniać odtworzenie danych po awarii klasy 2 w czasie 1 godziny z okresem utraty danych mniejszym niż 2 minuty

Rodzaj wymagań: Bezpieczeństwo

Stopień zabezpieczenia danych przez system, który umożliwia do nich dostęp odpowiedni do rodzajów uprawnień użytkowników lub innych systemów

- **Poufność** (ang. confidentiality) - zapewnienie dostępu do danych tylko dla jednostek do tego autoryzowanych
- **Integralność** (ang. integrity) - uniemożliwienie nieautoryzowanej możliwości zmiany programów i danych
- **Niezaprzeczalność** (ang. non-repudiation) - zapewnienie możliwości dowiedzenia wykonania akcji, tak, aby nie można im było zaprzeczyć w przeszłości
- **Odpowiedzialność** (ang. accountability) - zapewnienie identyfikacji jednostki odpowiedzialnej za akcje
- **Autentyczność** (ang. authenticity) - zapewnienie identyfikacji tożsamości jednostki (osoby)

Przykłady: Bezpieczeństwo

- **Poufność**
 - System musi zabezpieczać dostęp do parametrów analizatora kwantowego tylko dla osób o poziomie autoryzacji 10 przy poziomie szyfrowania Alfa
- **Integralność**
 - System powinien uniemożliwiać zmianę kodu procedury sterującej strumieniem kwantowym przy pomocy algorytmów 5 generacji
- **Niezaprzeczalność**
 - System powinien rejestrować wszystkie akcje wykonywane przez analizator kwantowy wraz z informacją o operatorze
- **Odpowiedzialność**
 - System musi dodawać identyfikator użytkownika do każdej transakcji sprzedażowej
- **Autentyczność** (ang. authenticity) -
 - System powinien umożliwiać podpisywanie każdego dokumentu przy pomocy podpisu elektronicznego zgodnego ze standardem Beta Plus

Rodzaj wymagań: Łatwość utrzymania

Parametry opisujące nakład pracy niezbędny do wprowadzenia zmian przez osoby utrzymujące system w eksploatacji

- **Modularność** (ang. modularity) - poziom podziału na rozdzielne komponenty i ich wzajemnego wpływu
- **Reużywalność** (ang. reusability) - możliwość wykorzystania zasobu w więcej niż jednym systemie
- **Łatwość analizy** (ang. analysability) - nakład pracy konieczny do identyfikacji niedoskonałości, przyczyn defektów lub wskazania tych części oprogramowania, które powinny zostać poddane modyfikacji
- **Łatwość zmiany** (ang. modifiability) - nakład pracy konieczny do implementacji zmiany, usunięcia zidentyfikowanego błędu lub dostosowania aplikacji do nowego środowiska
- **Testowalność** (ang. testability) - nakład pracy potrzebny do weryfikacji poprawności działania aplikacji po wprowadzeniu do niej modyfikacji

Przykłady: Łatwość utrzymania

- **Modularność**
 - System będzie posiadał niezależny komponent analizatora plazmy, udostępniający funkcje analizy innym komponentom
- **Reużywalność**
 - Komponent analizatora plazmy będzie możliwy do zastosowania bez zmian w całej serii systemów Alfa Plus
- **Łatwość analizy**
 - Kod system będzie zawierał dokumentację zgodną ze standardem kodowania XYZ (UWAGA: standard XYZ zapewnia szybką identyfikację błędów)
- **Łatwość zmiany**
 - System będzie zaprojektowany zgodnie z metodyką ABC (UWAGA: metodyka ABC stosuje modelowanie wizualne, odpowiednie standardy komponentyzacji, standardy kodowania itd., co ułatwia dokonywanie zmian)
- **Testowalność**
 - System będzie implementował interfejs użytkownika umożliwiający zastosowanie narzędzia do automatyzacji testów XYZ

Rodzaj wymagań: Przenośność

Zbiór atrybutów opisujących zdolność oprogramowania do przenoszenia między różnymi środowiskami sprzętowymi, software'owymi lub operacyjnymi

- **Adaptowalność** (ang. adaptability) - możliwość dostosowania oprogramowania do różnych, wyspecyfikowanych środowisk, bez konieczności realizacji dodatkowych czynności lub środków poza tymi, które są obecne w samym oprogramowaniu
- **Łatwość instalowania** (ang. installability) - nakład pracy konieczny do instalacji oprogramowania w danym środowisku
- **Łatwość zamiany** (ang. replaceability) - zdolność oprogramowania, do zastąpienia innej, konkretnej aplikacji w jej środowisku

Przykłady: Przenośność

- **Adaptowalność**
 - System będzie automatycznie dostosowywał układ menu głównego do rozmiarów ekranu
- **Łatwość instalowania**
 - System będzie instalowany automatycznie za pośrednictwem systemu App Shop
- **Łatwość zamiany**
 - System będzie realizował wszystkie funkcje systemu w wersji 3.4



31

Jak sprawdzać spełnienie wymagań jakościowych?

- W metodykach iteracyjnych konieczna jest ciągła kontrola poziomu spełnienia wymagań jakościowych
- Spełnienie wymagań jakościowych jest często kluczowe dla właściwego zaspokojenia potrzeb i oczekiwań zamawiających oraz użytkowników
- Mierzenie spełnienia wymagań jakościowych wymaga podania metryki



?

POMIAR

Łatwość utrzymania
Przenośność Wydajność
Użyteczność Bezpieczeństwo
Nieawodność

32

Metryki dla wymagań jakościowych

- Co to jest pomiar?
 - proces, w którym atrybutom obiektów świata rzeczywistego są przypisywane odpowiednie symbole
 - proces przeprowadzony zgodnie z jednoznacznie zdefiniowanymi zasadami
- Pomiar wymagania jakościowego wymaga metryki
- Cechy metryki wymagania jakościowego:
 - sposób pomiaru (w tym: wymagana liczba wykonywanych pomiarów)
 - zbiór możliwych wartości pomiaru (wraz z jednostką dla wartości liczbowych)
 - oczekiwana (akceptowalna) wartość (w tym: charakterystyka błędów)

Procedury dla metryk

- Pomiary parametrów systemu (eksperyment)
 - Czas, przepustowość, zajętość pamięci itp.; pomiary pod symulowanym obciążeniem
- Audyty
 - audyt kodu/projektu, audyt bezpieczeństwa itp.; oczekiwany rezultat - pozytywny wynik, certyfikat, itp.
- Ankietyzacja
 - satysfakcja użytkowników, zrozumienie działania systemu itp.; oczekiwany rezultat - odpowiedni poziom satysfakcji itp.
- Sprawdziany
 - obserwacja użytkowników, testy sprawności używania systemu; oczekiwany rezultat - uzyskanie odpowiedniej sprawności używania systemu
- Symulacja zdarzeń
 - symulowane wywołanie błędów, wstrzykiwanie błędów do kodu; oczekiwany rezultat - uzyskanie odpowiedniej szybkości naprawy błędów

Przykład: audyt user experience (UX)

- **Zgodność ze standardami i regułami komunikacji człowiek-komputer** - spełnienie założeń interfejsów dla danej platformy, systemu, lub środowiska.
- **Intuicyjność** - odpowiednia ilość informacji prezentowanych jednocześnie użytkownikowi, właściwe rozplanowanie konkretnych widoków (wyglądu ekranów).
- **Spójność** - jednolity wygląd i zachowanie różnych części interfejsu, przemyślane przepływy między ekranami aplikacji, spójna (także z np. systemem operacyjnym, pod którego kontrolą pracuje aplikacja) terminologia, skróty klawiszowe, układ menu.
- **Elastyczność** - możliwość dostosowania aplikacji do sposobu pracy różnych osób
- **Wygoda** - łatwość dostępu do pewnych funkcji programu (powinna być tym wyższa, im częściej używana funkcja)
- **Poprawność** - zgodność oczekiwanego zachowania się programu pod wpływem inicjowania danych operacji w jego interfejsie.

Przykład: ankiety użyteczności

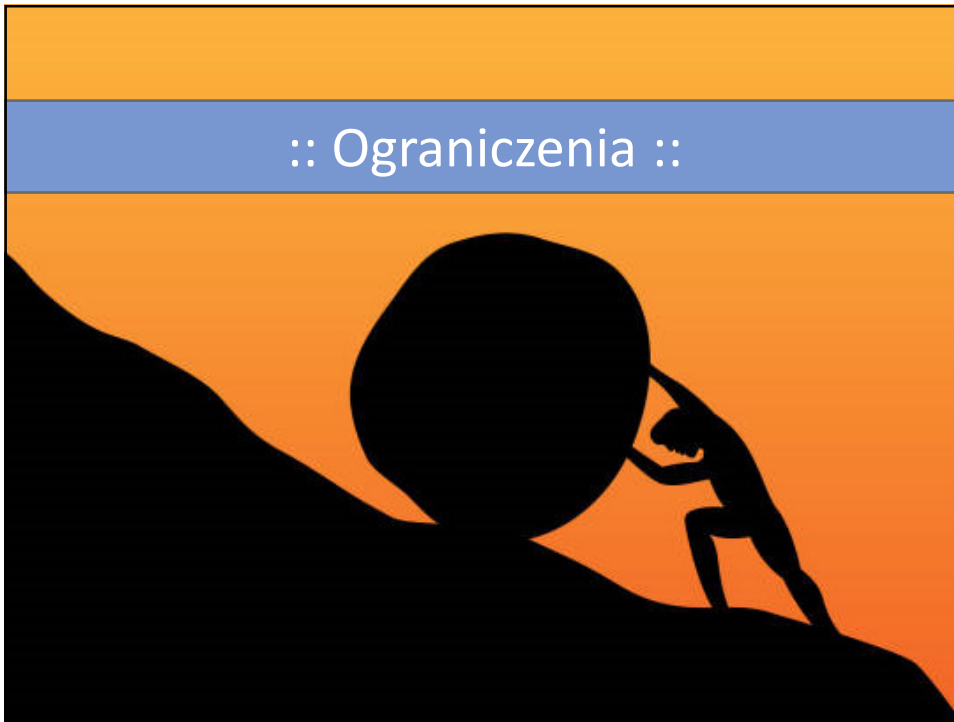
- **Miara wysiłku użytkownika** (ang. Customer Effort Score, CES) - pomiar poziomu wysiłku, jaki się pojawia w czasie interakcji użytkownika z systemem poprzez jego interfejs
- **Miara satysfakcji klienta** (ang. Customer Satisfaction Score, CSAT) - pomiar poziomu satysfakcji z użytkowania systemu poprzez odpowiedzi na pytania.
- **Miara chęci promowania** (ang. Net Promoter Score, NPS) - pomiar poziomu satysfakcji poprzez wskazanie jak bardzo użytkownik jest skłonny produkt polecić osobie trzeciej.
- Skale: 1-10, ikony poziomu zadowolenia, skala trudne - łatwe, skala bardzo zadowolony - bardzo niezadowolony

Przykład: sprawdziany użyteczności

- **Współczynnik powodzenia wykonania zadań** (ang. Task Success Rate) - określa w sposób procentowy poprawność wykonania zadań (np. wypełnienie formularza) przez użytkowników. Badanie problemów z użytecznością.
- **Czas przeznaczony na wykonanie zadania** (ang. Time spent on task) - określa czas, który użytkownicy przeciętnie muszą poświęcić na wykonanie określonych zadań. Łatwość interakcji z systemem.
- **Miara błędów użytkownika** (ang. User Errors) - pomiar liczby błędów różnego rodzaju, które popełnia użytkownik podczas uruchamiania danej funkcjonalności systemu. Identyfikacja miejsc problemowych.

37

:: Ograniczenia ::



38

Ograniczenia vs. wymagania jakościowe

- Niektórzy traktują ograniczenia jako rodzaj wymagań jakościowych, ale:
- Jak weryfikujemy spełnienie ograniczeń?
- Brak procedur weryfikacyjnych!
- Ograniczenia są narzuconymi cechami warsztatu deweloperskiego lub otoczenia systemu.
- Niezmienniki, rządzące procesem wytwarzania danego systemu oprogramowania.

Ograniczenia techniczne

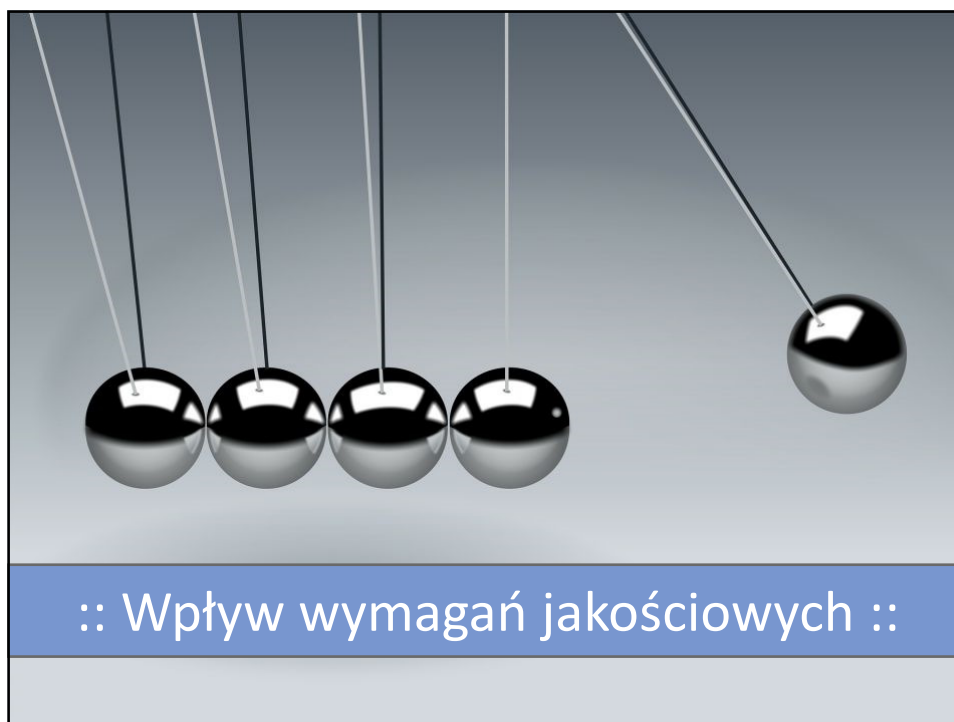
- Dotyczą technologii, jakie zespół deweloperski musi zastosować podczas budowy systemu
- Ograniczają stopień swobody w wyborze architektury, sprzętu czy narzędzi programistycznych.
- Często narzucane przez klienta z powodu posiadanych przez niego zasobów technicznych lub po prostu - aktualnie obowiązującej mody na dane rozwiązania techniczne.

Rodzaje ograniczeń technicznych

- **Ograniczenia dotyczące języków programowania.**
 - „Frontend systemu będzie napisany w języku TypeScript w technologii React z wykorzystaniem programowania obiektowego”
 - „Backend systemu będzie wykonany w języku Java i przy użyciu technologii Jakarta EE 10”
- **Ograniczenia dotyczące środowiska deweloperskiego**
 - „Zespół będzie stosował DevOps z wykorzystaniem GitLab-a”
 - „Testy będą automatyzowane przy pomocy narzędzia Selenium”
 - „Zespół będzie stosował środowisko programistyczne (IDE) JetBrains”
- **Ograniczenia dotyczące stosowanego sprzętu.**
 - „System będzie wykorzystywał istniejącą sieć rozległą TransNet”
 - „Backend systemu będzie zainstalowany na węźle B21 Centrum Obliczeniowego K-Banku, gdzie będzie mógł wykorzystywać 40% mocy obliczeniowej”
 - „Terminale będą wykorzystywać obecnie zainstalowane tablety T21A”
- **Ograniczenia dotyczące stosowanych komponentów programowych**
 - „System musi przechowywać dane w centralnej bazie OctaBase, w osobnej przestrzeni danych”
 - „System musi wykorzystywać posiadane licencje na SZBD - K-SQL v. 10.1”
 - „System musi stosować moduł kolejowania zadań obliczeniowych w oparciu o kolejkę BunnyMQ”

Ograniczenia środowiskowe

- **Ograniczenia dotyczące parametrów fizycznych środowiska**
 - „Terminale dla nurków będą podlegać ciśnieniu 5 atm”
 - „Stacje robocze będą pracować w temperaturze od 20 do 35 oC”
 - „Wydzielone komory operatorskie H20 będą miały maksymalny poziom zapylenia 100 µg/m³”
- **Ograniczenia dotyczące zachowań i możliwości użytkowników.**
 - „Użytkownicy systemu będą poruszać się na wózkach”
 - „Kierowcy nie będą mieli możliwości dotarcia do panelu pomocniczego”
 - „Klienci będą mieli tylko od 5 do 10 minut na podjęcie decyzji”
 - „Uczestnikami zabaw będą dzieci w wieku 3-5 lat”
 - „Operatorzy będą musieli używać rękawic roboczych RR12”.
- **Ograniczenia dotyczące infrastruktury nieinformatycznej**
 - „Centrum obliczeniowe będzie zasilane z podstacji 10kV”
 - „Dostępne systemy UPS mogą podtrzymywać zasilanie przez maksymalnie 5 godzin”
 - „Do serwerowni jest dostęp jedynie przez wejście N2 o szer. 50 cm”



43



44

Wymagania jakościowe a wym. funkcjonalne (1)

- Operacjonalizacja wymagań bezpieczeństwa (niezaprzeczalność)

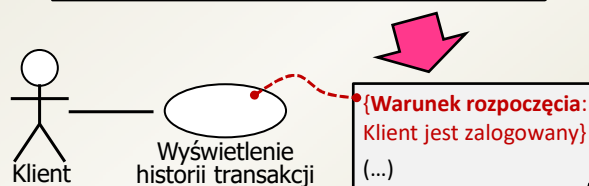
JB112: Każda operacja gotówkowa na koncie musi być potwierdzona wydrukiem papierowym.
(bezpieczeństwo – niezaprzeczalność)



Wymagania jakościowe a wym. funkcjonalne (2)

- Operacjonalizacja wymagań bezpieczeństwa (poufność)

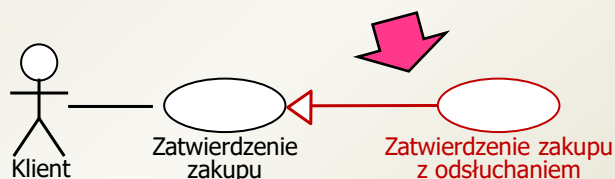
JB124: Dostęp do danych o koncie jest możliwy tylko dla zautoryzowanego właściciela konta.
(bezpieczeństwo – poufność)



Wymagania jakościowe a wym. funkcjonalne (3)

- Operacjonalizacja wymagań użyteczności (przystępność)

JU037: Dla wszystkich transakcji powinna istnieć możliwość odsłuchania parametrów przed zatwierdzeniem.
(użyteczność – przystępność)



47

Wymagania jakościowe a wym. funkcjonalne (4)

- Operacjonalizacja wymagań użyteczności (operacyjność)

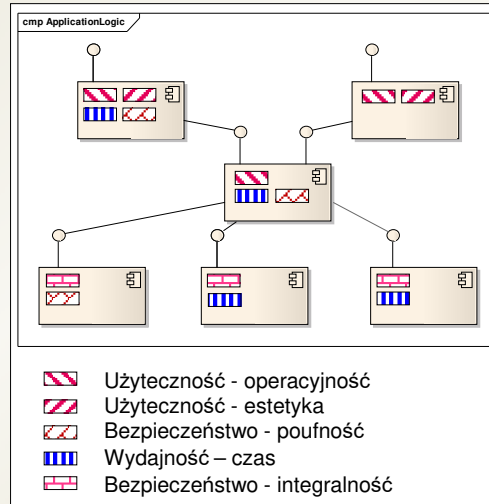
JU044: Aby dotrzeć do głównych formularzy z menu głównego powinny wystarczyć maksymalnie dwa kliknięcia.
(użyteczność – operacyjność)



48

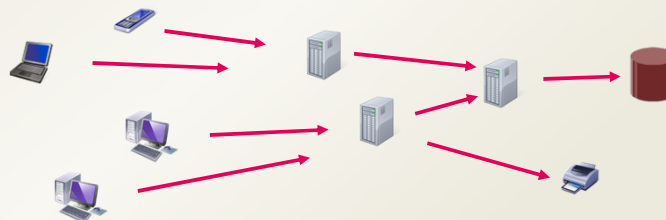
Wymagania jakościowe a zagadnienia przekrojowe

- Większość wymagań jakościowych „przecina” realizacje wymagań funkcjonalnych
- Jedno wymaganie jakościowe realizowane w wielu komponentach systemu



Wymagania jakościowe a koszt systemu (1)

- Wybór odpowiednich technologii pozwalających na spełnienie określonych wymagań jakościowych - np. wybranie komercyjnej bazy danych w celu zapewnienia odpowiedniej wydajności lub skalowalności rozwiązania
- Wybór odpowiedniej infrastruktury sprzętowej na której system zostanie wdrożony - np. w celu zapewnienia odpowiedniej niezawodności, wydajności, bezpieczeństwa.



Wymagania jakościowe na koszt systemu (2)

- Trudność/koszt weryfikacji wymagań jakościowych
- Dostarczenie systemu spełniającego zdefiniowane wymagania jakościowe, wymaga zainwestowania dodatkowych zasobów na zdefiniowanie obiektywnych metryk dla poszczególnych wymagań oraz przetestowanie, czy system spełnia te metryki.

