

Interakcja człowiek-komputer

W4

Jakość interfejsu i jej ocena

Dr inż. Marek Miłoś, prof. uczelni



**Fundusze
Europejskie**
Wiedza Edukacja Rozwój



**Rzeczpospolita
Polska**

Unia Europejska
Europejski Fundusz Społeczny



Plan wykładu

- Pojęcie jakości interfejsu oprogramowania
- Kryteria jakości i rekomendacje
- Użyteczność i dostępność
- Ocena jakości interfejsów oprogramowania – metody
- Testowanie jakości interfejsów – ocena z udziałem użytkowników
- Metody oceny bez udziału użytkownika
- Narzędzia wspomagające ocenę jakości interfejsu oprogramowania

Pojęcie jakości interfejsu oprogramowania

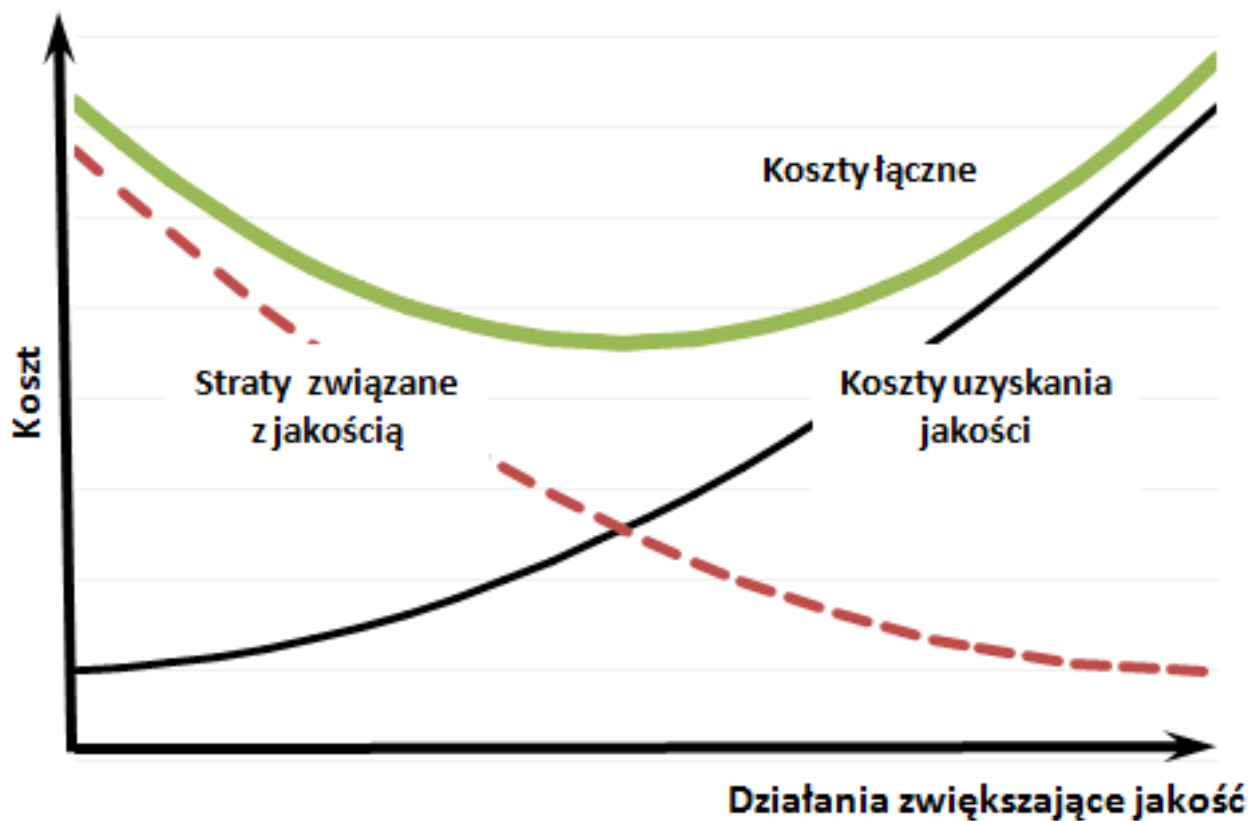
Jakość oprogramowania

- Pojęcie rozmyte
- „zgodności z wymaganiami użytkownika”
- „zaspokojenia potrzeb klienta”
- „zgodności ze standardami”
- „ogół cech i właściwości wyrobu lub usługi, które decydują o zdolności wyrobu lub usługi do zaspokojenia stwierdzonych lub przewidywanych potrzeb” (norma ISO 8402)

Ekonomiczny wymiar jakości (1)

- Dodatkowe koszty wynikające ze złej jakości:
 - poprawiania błędów
 - wydłużonego czasu realizacji zadań (zmniejszenie produktywności pracowników)
 - koszty błędów
- Im system większy i częściej stosowany tym większe koszty złej jakości (choć nie zawsze: zła jakość oprogramowania kontroli reaktora jądrowego)
- Teoretyczny pomiar:
potencjalny koszt = prawdopodobieństwo x koszt naprawy

Ekonomiczny wymiar jakości (2)



Kryteria jakości i rekomendacje

Zintegrowany
Program
Rozwoju
Politechniki
Lubelskiej -
część druga

Metryka jakości interfejsu

- Ocena jakościowa i ilościowa
- Mapowanie ocen jakościowych na ilościowe:
 - binarne (0/1)
 - skale ilościowe: zwykle pięciostopniowe lub procentowe (0%-100%)
- Skale % - wynik analiz statystycznych wielu rezultatów (wypowiedzi, ankiet itp.)

Metryki oceny jakości interfejsu

Klasyfikacja

- Metryki wydajnościowe (ang. *Performance Metrics*)
- Metryki bazujące na problemach (ang. *Issue-based Metrics*)
- Metryki bazujące na ocenach użytkowników (ang. *Self-reported Metrics*)
- Metryki behawioralne i fizjologiczne (ang. *Behavioural and Physiological Metrics*)

Metryki oceny jakości interfejsu

Metryki wydajnościowe

- Sukces realizacji zadania (ang. *Task Success*) – mierzy, jak skutecznie użytkownicy są w stanie ukończyć dany zestaw zadań
- Czas realizacji zadania (ang. *Time on Task*) – mierzy, ile czasu jest potrzebne do wykonania zadania
- Stopa błędów (ang. *Errors*) – określa liczbę błędów popełnionych w trakcie realizacji zadania
- Przyswajalność (ang. *Learnability*) – mierzy zmiany w produktywności użytkowników w trakcie pracy (zaznajamiania się) z oprogramowaniem

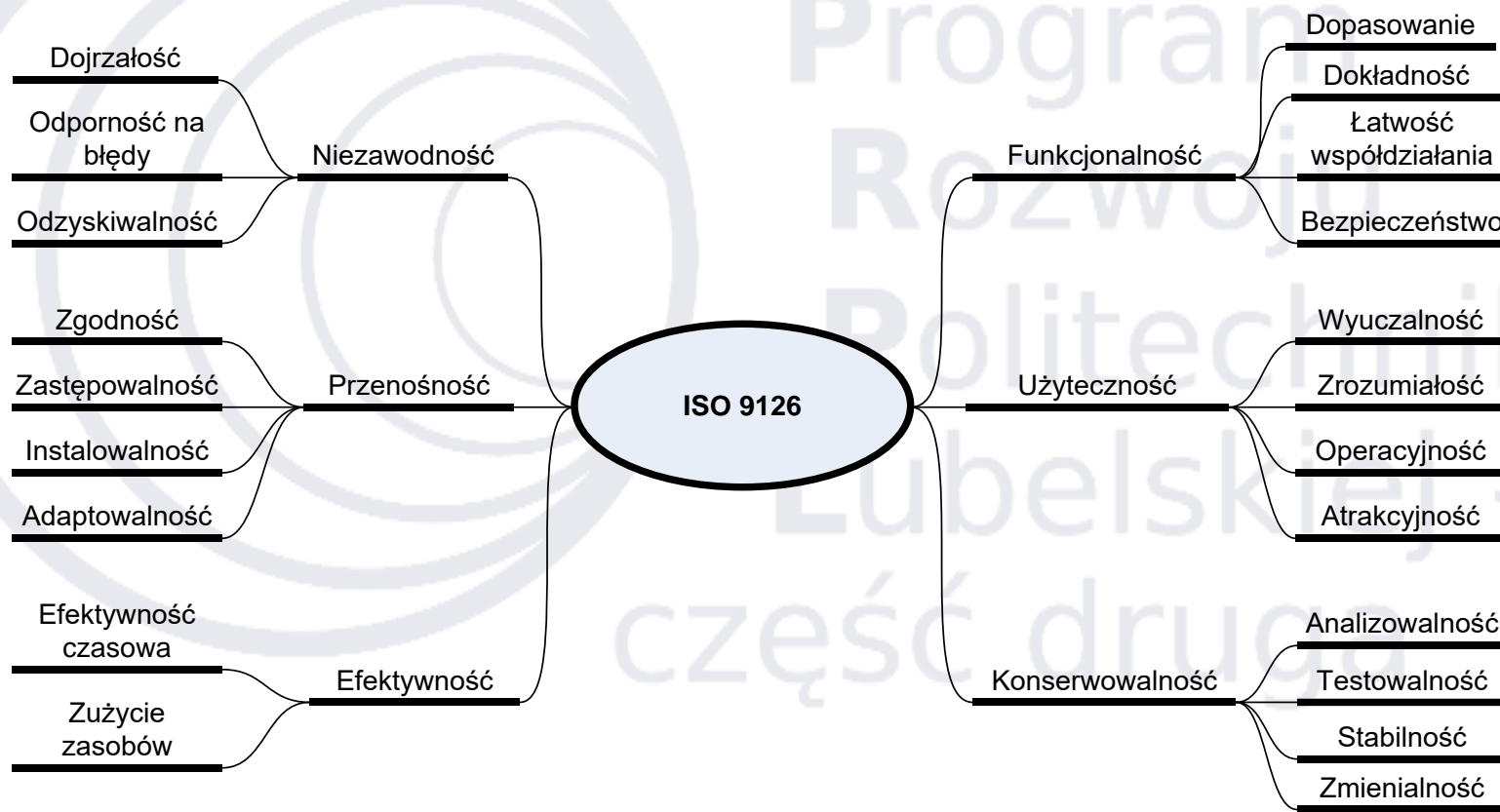
Metryki oceny jakości interfejsu bazujące na problemach

- Częstość występowania problemów (ang. *Frequency of Unique Issues*) – unikalnych problemów
- Częstość występowania problemów na jednego użytkownika (ang. *Frequency of Unique Issues per Participant*) – unikalnych problemów
- Procent uczestników doświadczających błędu (ang. *Frequency of Participants*)
- Kategoryzacja błędów (ang. *Issues by Category/Task*) – pomiar liczby błędów w zagregowanych grupach/kategoriach

Metryki oceny jakości interfejsu bazujące na **ocenach** użytkownika

- **Ocena po zadaniu** (ang. *Post-task Ratings*) – ocena realizowana po wykonaniu określonego zadania przez użytkownika z wykorzystaniem określonej skali. Równocześnie mogą być oceniane różne parametry jakościowe oprogramowania
- **Ocena po sesji badawczej** (ang. *Post-session Ratings*)
- Ocena specyficznego aspektu oprogramowania (ang. *Assessing Specific Attribute*) – ocena przez użytkownika specyficznej cechy/elementu oprogramowania (np. dostosowanie do osób niesprawnych ruchowo lub niedowidzących)

Cechy jakości (wg. ISO 9126:2001)

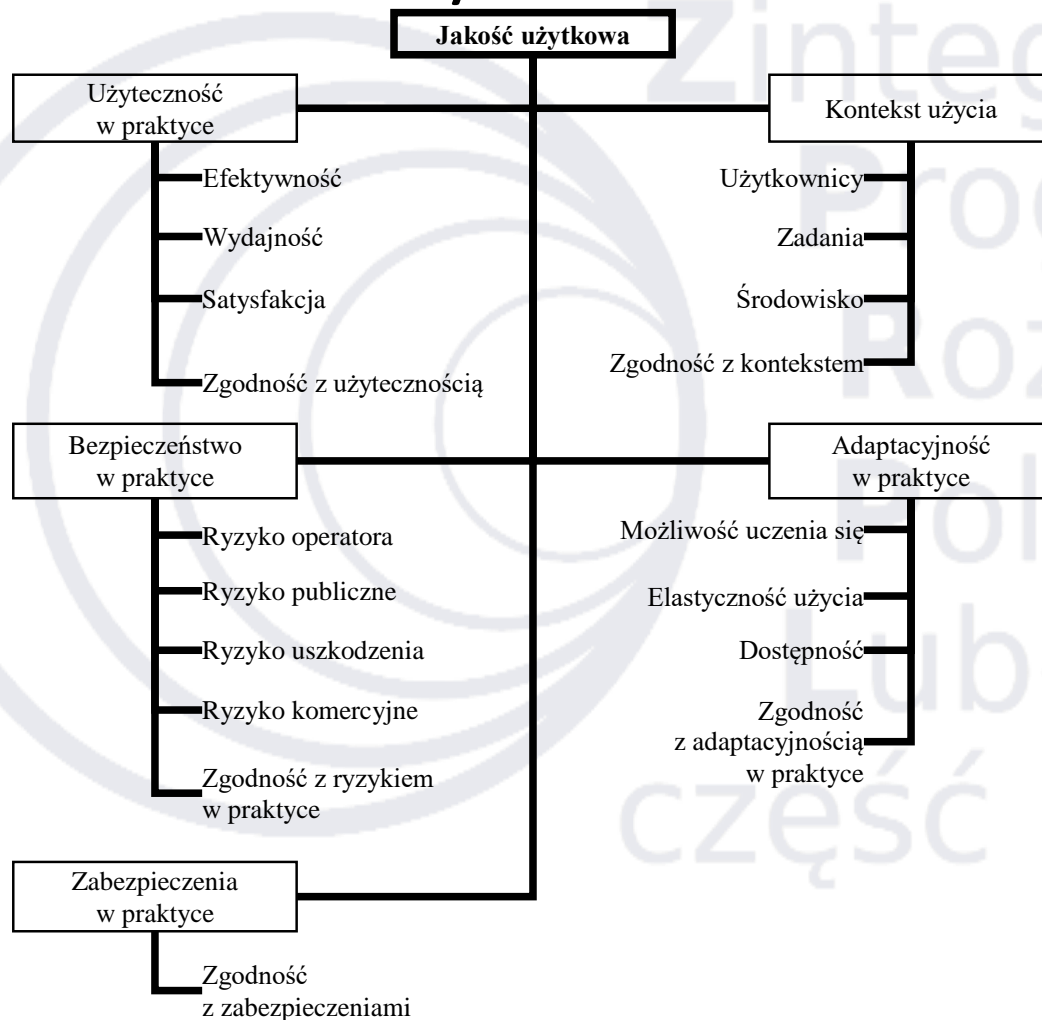


Jakość użytkowa (wg. ISO 9126:2001)



- Efektywność – zdolność oprogramowania do osiągania zadanych celów z zadaną dokładnością
- Produktywność – efektywna (potrzebna) wydajność
- Bezpieczeństwo – zdolność do długotrwałej, poprawnej pracy oprogramowania
- Satysfakcja – zadowolenie klienta w trakcie użytkowania oprogramowania (a także po)

Jakość użytkowa (wg. ISO/IEC 25010:2011)



Cechy jakości (1)

(wg. ISO/IEC 25010:2011)

- Funkcjonalność:
 - przydatność
 - dokładność
 - zgodność z funkcjonalnością
- Bezpieczeństwo:
 - odporność dostępu
 - odporność kopii
 - szyfrowalność
 - odporność na manipulacje
 - uodpornienie
 - zgodność

Cechy jakości (2)

(wg. ISO/IEC 25010:2011)

- Interoperacyjność:
 - zgodność z OSI
 - kompatybilność programu
 - kompatybilność danych
 - możliwość śledzenia
 - zgodność
- niezawodność:
 - dojrzałość
 - tolerancja błędów
 - możliwość odzysku
 - zgodność niezawodności

Cechy jakości (3)

(wg. ISO/IEC 25010:2011)

- **Użyteczność:**
 - odpowiedniość
 - możliwość nauczania się
 - operacyjność
 - pomocniczość
 - zadowolenie
 - zgodność użyteczności
- **Efektywność:**
 - zachowanie czasowe
 - zużycie zasobów
 - zgodność efektywności

Cechy jakości (4)

(wg. ISO/IEC 25010:2011)

- **Możliwość utrzymania:**
 - możliwość analizy
 - możliwość zmian
 - stabilność
 - możliwość testowania
 - zgodność utrzymania
- **Przenośność:**
 - adaptacyjność
 - współegzystencja
 - zamiennność
 - zgodność przenośności

Niestandardowe cechy jakości

- Agnostycyzm danych
- Intuicyjność interfejsu
- Zachęcanie
- Minimalizm
- Charyzma interfejsu:
 - unikalność
 - profesjonalizm
 - zaciekawienie
 - nowość
 - innowacyjność
 - pierwsze wrażenie
 - historia

Zintegrowany
Program
Rozwoju
Politechniki
Lubelskiej -
część druga

Użyteczność i dostępność

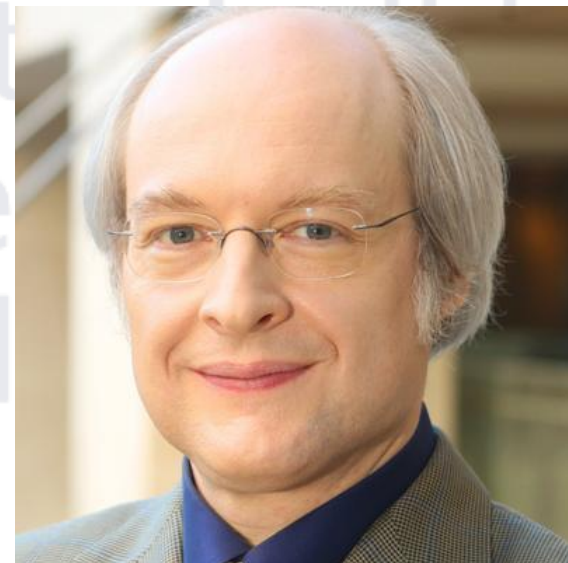
Zintegrowany
Program
Rozwoju
Politechniki
Lubelskiej -
część druga

Użyteczność (ang. *Usability*)

- Łatwość nauki (ang. *Learnability*)
- Efektywność (ang. *Efficiency*)
- Zapamiętywalność (ang. *Memorability*)
- Stopa błędów (ang. *Errors*)
- Satysfakcja (ang. *Satisfaction*)

Jakob Nielsen – „the usability Pope”

Źródło: <https://www.nngroup.com/people/jakob-nielsen/>



„Typowe” problemy interfejsu

- Trudność uczenia się
- Przeciążenie ekranu
- Nadmierna innowacyjność
- Brak spójności i przewidywalności interfejsu
- Nieprawidłowy kształt lub rozmiar obiektów
- Nieprawidłowy dobór kolorów
- Ograniczenie możliwości korzystania z klawiatury
- KONTRAST

Rezultaty złej użyteczności

- Zniechęcenie (w skrajnych przypadkach zaprzestanie użytkowania)
- Brak spójności (zaskakiwanie użytkownika)
- Ukrycie funkcjonalności
- Brak poczucia kontroli nad oprogramowaniem (element satysfakcji)
- Brak możliwości dopasowania domyślnych ustawień do swoich wymagań
- Brak dostępności

Dostępność (ang. *Accessibility*)

- Ograniczenia użytkowania – czynniki:

- człowiek
- sprzęt i oprogramowanie
- otoczenie

- Niepełnosprawność człowieka:


- wzrokowa
- słuchowa
- motoryczna
- poznawcza (kognitywistyczna)

ograniczająca możliwość korzystania z oprogramowania

Dostępność – biznes i prawo

- Biznes – zwiększenie liczby użytkowników oprogramowania
- Przepisy prawne wynikają z zagwarantowania równych praw ludzi i niedyskryminację ich (zawarte w Konstytucji RP)
- Inne przepisy:
 - gwarancja dostępu do informacji publicznej
 - odwoływanie się w umowach do zawartości stron internetowych
 - zasada równych szans

Ocena jakości interfejsów oprogramowania – metody



Zintegrowany
Program
Rozwoju
Politechniki
Lubelskiej -
część druga

Klasyfikacja metod oceny jakości interfejsu

- **Automatyczne** (ang. *Automated*), tj. takie, w których procedurę oceny wykonują narzędzia komputerowe w całości lub w znacznej części. Możliwe do stworzenia jeśli są wzorce-standardy poddające się algorytmizacji
- **Manualne** (ang. *Manual*), czyli wykonywane ręcznie przez człowieka; mogą być wspomagane komputerowo, ale główna ich część, związana z oceną (pozyskaniem wiedzy), wykonywana jest przez człowieka – osobę oceniającą

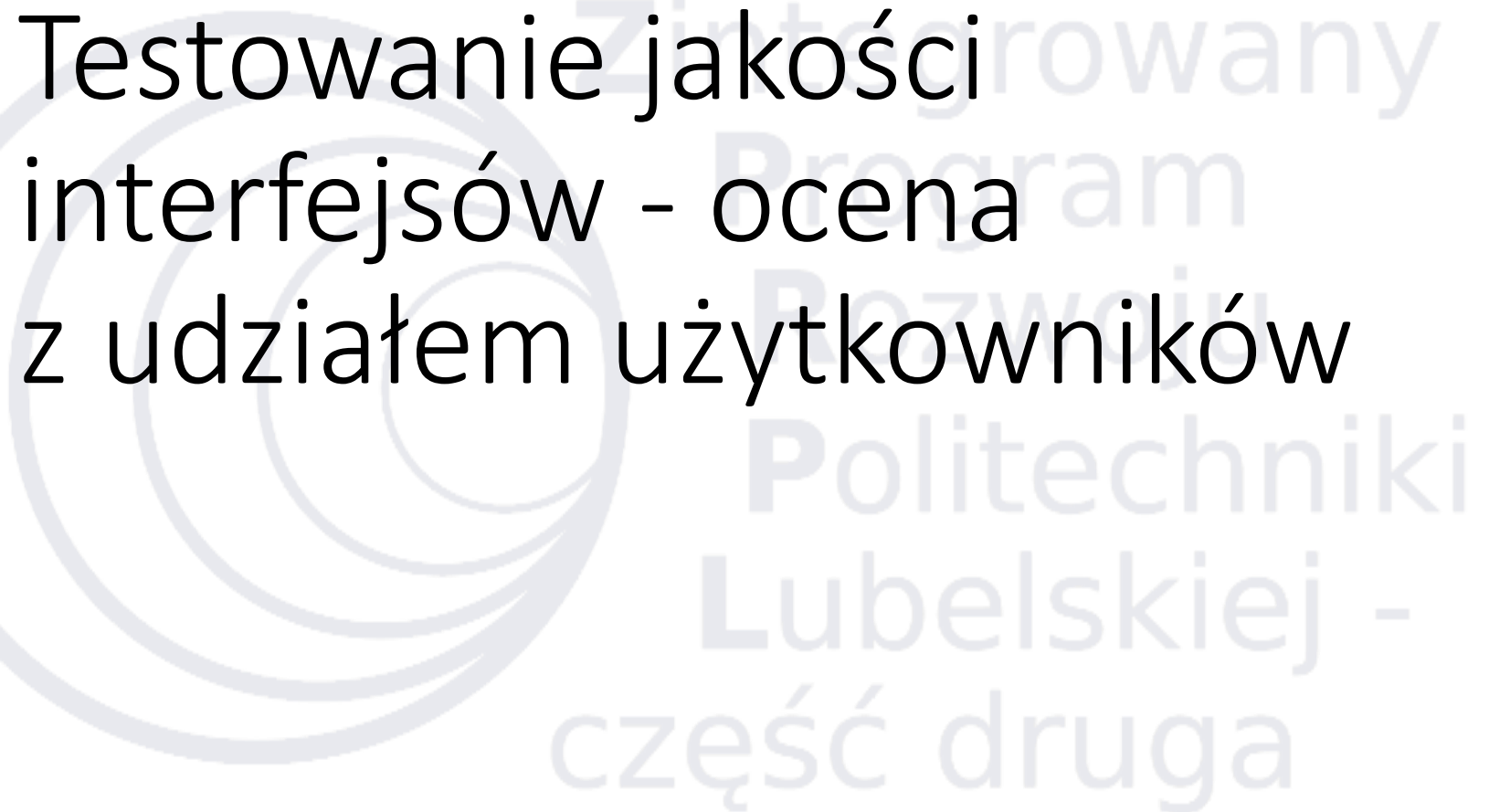
Klasyfikacja **manualnych** metod oceny jakości interfejsu

- **Z udziałem** użytkownika: ocena jakości interfejsu jest realizowana przy aktywnym udziale grupy użytkowników – uczestników oceny
- **Bez udziału** użytkownika: ocena jest realizowana i pochodzi od ekspertów w sprawach interfejsu oprogramowania

Typy metod oceny jakości

- **Testowanie** (ang. *Testing*)
- **Inspekcja** (ang. *Inspection*)
- **Ankieta** (ang. *Survey*)
- **Wywiad** (ang. *Interview*)
- **Werbalizacja** (np. „myślenie na głos”) (ang. *Verbalisation*)
- Modelowanie analityczne (ang. *Analytical Modeling*)
- Symulacja (ang. *Simulation*)

Testowanie jakości interfejsów - ocena z udziałem użytkowników



Techniki oceny jakości interfejsów z udziałem użytkowników (1)

- Testowanie interfejsu przez użytkowników lub przyszłych użytkowników – realizacja scenariuszy (eksperyment aktywny)
- Metoda A/B testowania – w sytuacji porównywania ocen
- Okulografia (ang. *Eye tracking*)
- Śledzenie działań użytkownika w interfejsie (ang. *Click tracking*)

Techniki oceny jakości interfejsów z udziałem użytkowników (2)

- Zbieranie opinii użytkowników oprogramowania (ang. *User Feedback*) – ankietowanie, wywiady
- Myślenie na głos (ang. *Thinking Aloud*) w trakcie realizacji zadań z analizą zapisów
- Obserwacja działań użytkownika – technika bierna
- Analiza dzienników/zapisów – technika bierna

Testowanie interfejsu – definicja

- Odpowiednio dobrana grupa użytkowników pracuje z oprogramowaniem wykonując określone scenariusze lub też bez tych scenariuszy
- W trakcie lub po eksperymencie zbierane są jego rezultaty
- Zwykle stosuje się techniki mieszane: obserwacja w trakcie działań, rejestracja działań i ich parametrów oraz ankietowanie po eksperymencie

Testowanie interfejsu – rezultaty – metryki szczegółowo (1)

- Czas ukończenia zadań
- Procent pomyślnie ukończonych zadań
- Procent zadań poprawnie ukończonych w przyjętym limicie czasowym
- Czas poświęcony na obsługę błędów
- Procentowa liczba błędów w ogólnej liczbie wykonanych przez użytkownika operacji
- Liczba używanych funkcji i poleceń
- Częstotliwość używania pomocy
- Czas spędzony na przeglądaniu dostępnej dokumentacji w celu wyszukania informacji dotyczącej sposobu realizacji różnych operacji lub obsługi pojawiających się błędów

Testowanie interfejsu – rezultaty – metryki szczegółowo (2)

- Liczba powtórzeń lub nieudanych prób użycia funkcji
- Liczba wprowadzeń w błąd użytkownika
- Liczba poprawnie i błędnie wywoływanych funkcji przez użytkownika w określonym czasie
- Liczba komend dostępnych, ale nie używanych nigdy przez użytkowników do realizacji zadań
- Liczba sytuacji, w których użytkownicy musieli poszukiwać alternatywnych sposobów rozwiązania danego problemu
- Liczba sytuacji rozproszenia uwagi użytkownika
- Liczba przypadków utraty kontroli nad systemem
- Liczba zgłoszeń frustracji użytkownika

Testowanie interfejsu

Zalety i wady

- Najbardziej efektywna metoda
- Zalety:
 - dużo informacji
 - możliwe niespodziewane działania użytkowników
 - sugestie użytkowników odnośnie nowych funkcji
 - precyzyjne informacje z obserwacji/pomiarów
- Wady:
 - niezręczna i nienaturalna sytuacja dla użytkowników
 - brak kontroli nad przebiegiem obserwacji
 - trudna do zorganizowania
 - kosztowne
 - ludzie się uczą ;-)

Powtarzalność eksperymentów

- Eksperymenty (badawcze) powinny być powtarzalne
- Ludzie się uczą → brak możliwości powtórzenia eksperymentu w identycznych warunkach z tymi samymi ludźmi
- Wyjście (statystycznie te same warunki):
 - statystycznie te same grupy badawcze (dla konkretnej osoby)
 - każda grupa wykonuje eksperyment
 - rezultaty eksperymentu uśredniane są w grupie
 - uśrednione rezultaty mogą być porównywane
- Metoda A/B

Okulografia

- Okulografia (ang. *Eye Tracking*) – badanie aktywności gałek ocznych człowieka i umożliwiające śledzenie co i jak postrzega człowiek
- Dostarcza informacji o aktywności wzroku, a NIE o zrozumieniu postrzeganej informacji. Zwykle jest łączona z innymi: obserwacja, myślenie na głos
- Pozwala zbadać postrzeganie obiektu (na czym człowiek skupia wzrok a jakie obszary całkowicie pomija)

Okulografia – oko

- Organ wzroku
- Soczewka (ok. 2 dioptry), źrenica (przesłona ograniczająca ilość światła) i siatkówka
- Siatkówka:
 - receptor światła
 - 125 mln. komórek rejestrujących obraz: pręcików (ruch, 120 mln.) i czopków (kolor, 5 mln.)
 - rejestracja z częstotliwością $< 50\text{Hz}$
 - plamka ślepa (wyjście połączeń nerwowych)
- Obraz w oku odwrócony (działanie soczewki) – mózg go zmienia (przetwarza)

Okulografia – użycie wzroku

- Obserwowanie ruchomych obiektów – po kolei: fiksacje (ang. *Fixations*) – zatrzymanie wzroku i sakady (ang. *Saccades*) – przeskoki do innego miejsca fiksacji
- W przypadku ruchu ponad 30 %s konieczne są dodatkowe sakady
- Rejestracja punktów fiksacji w czasie

Źródło: <https://www.usability.gov/how-to-and-tools/methods/eye-tracking.html>



Okulografia – urządzenia

- Metody: inwazyjne i bez- (oświetlenie oka promieniami podczerwonymi i rejestracja linii patrzenia)
- Eye tracker: stacjonarny i mobilny

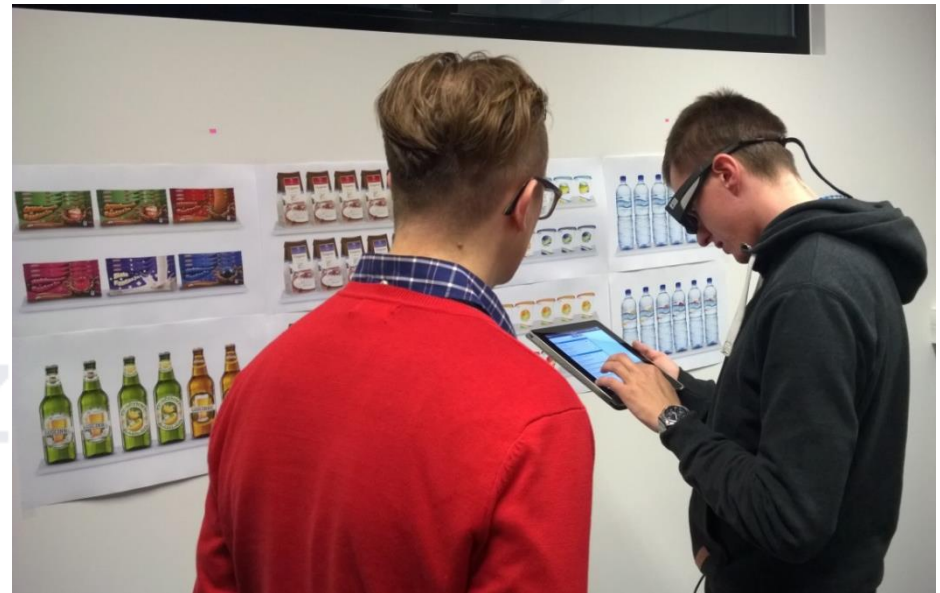


Źródło: <https://www.tobii.com/>



Okulografia – badania mieszane

- Badania:
 - interfejsu
 - otoczenia pracy użytkownika
- Interakcje z oprogramowaniem oraz/lub otoczeniem



Okulografia – prezentacja wyników badań

- Mapy fiksacji (tj. koncentracji wzroku) – sekwencja punktów fiksacji z jej czasami
- Mapy cieplne dla wielu badań
- Filmy z punktem skupienia wzroku (dla badań dynamicznych, np. działań użytkowników realizujących scenariusze wykorzystania oprogramowania)
- Podział obrazu na strefy (segmentacja) i wyznaczenie częstotliwości/długości fiksacji wzroku w strefach zainteresowania

Okulografia – mapa fiksacji

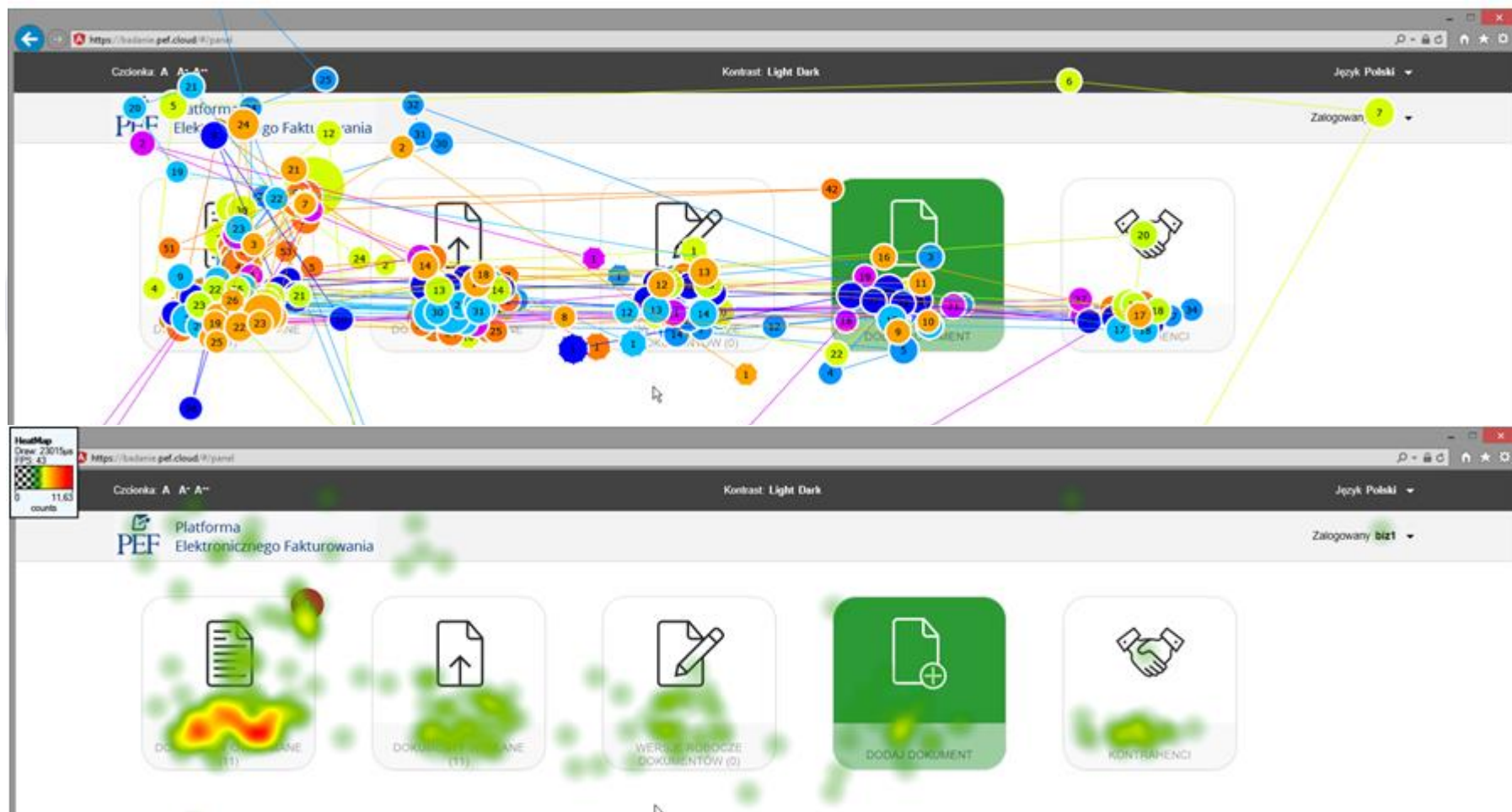


Okulografia – mapa cieplna



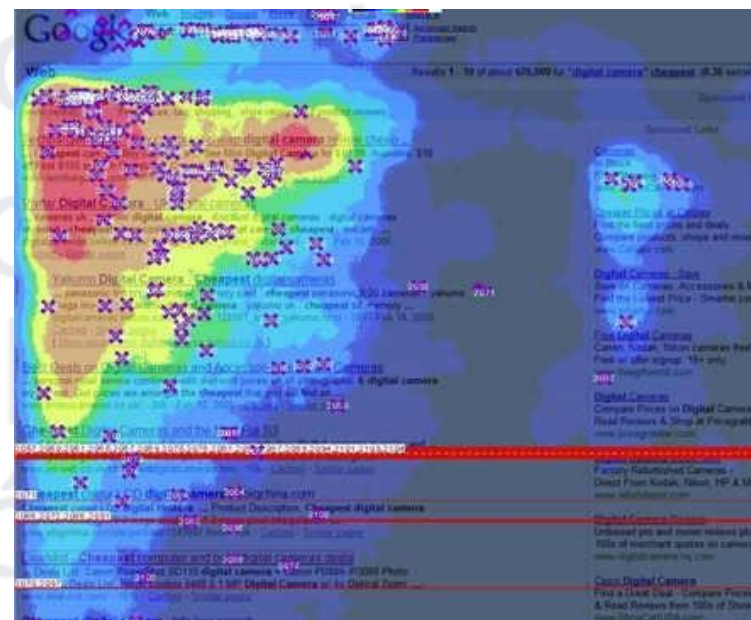
Źródło: <https://www.usability.gov/how-to-and-tools/methods/eye-tracking.html>

Okulografia – mapa fiksacji vs. cieplna



Okulografia – strony internetowe

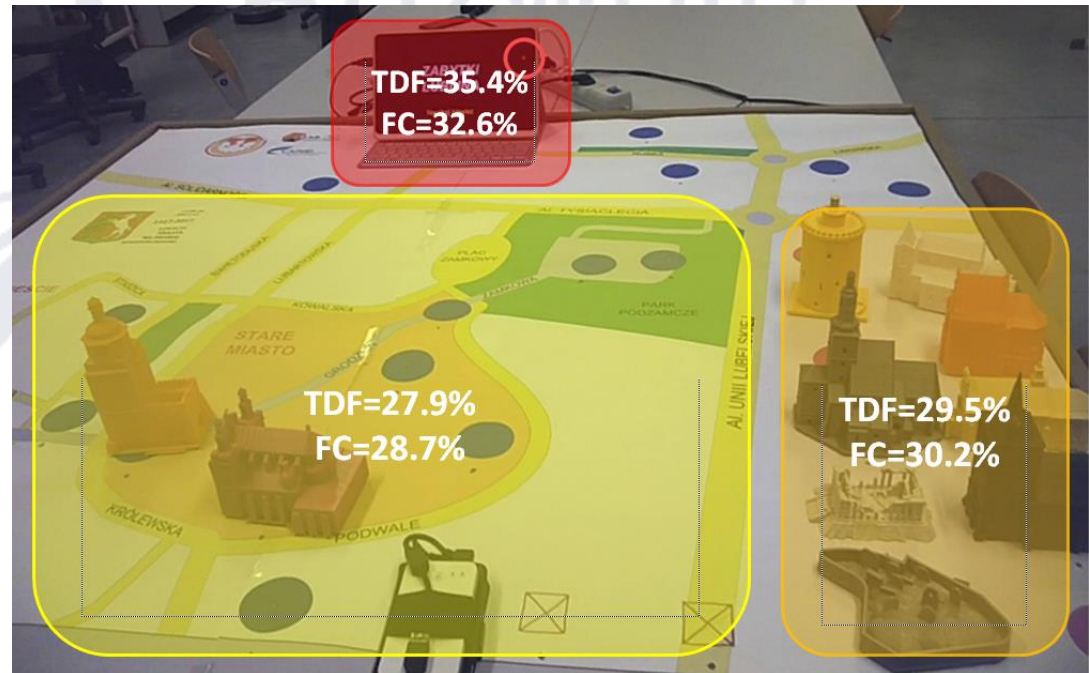
- F-shape wzorzec przeglądania stron www:
 - odczyt poziomy
 - ponowny odczyt poziomy
 - odczyt pionowy
- Bywają inne (E i L kształtne) wzorce – rzadko
- Rezultaty:
 - odczyt strony nie przypomina odczytu książki
 - 2 pierwsze wiersze najważniejsze



Źródło: <https://poradnikprzedsiębiorcy.pl/-eyetracking-czyli-na-co-zwracamy-uwage>

Okulografia – segmentacja

- Obszary zainteresowania (AOI – *Area of Interest*)
- Liczba fiksacji (FC – *Fixation Count*)
- Łączny czas fiksacji (TDF – *Total Duration Fixation*)



Zbieranie opinii użytkowników

- Dotyczy:
 - wersji eksploatowanych
 - wersji testowych (beta testy)
- Metody:
 - automatyczne
 - wywiad z użytkownikami
- Wywiad z użytkownikami:
 - ankieta papierowa (ang. *Paper-based Survey*)
 - ankieta elektroniczna
 - e-maile
 - fora dyskusyjne
 - wywiady zogniskowane (grupowe i jednostkowe)

Zbieranie opinii – zalety i wady

- Zalety:
 - szybkość działania
 - mało kłopotliwe
 - niskie koszty
 - proces ciągły
 - wychodzi od użytkowników
- Wady:
 - niereprezentatywna próbka
 - mało konkretne odpowiedzi
 - czasami mylne sugestie
 - subiektywizm
 - sprzeczności wypowiedzi



Metodyka SUS

Zintegrowany
Program
Rozwoju
Politechniki
Lubelskiej -
część druga

SUS – istota

- *System Usability Scale* - subiektywny poziom satysfakcji (użytkowników oprogramowania)
- Technika „quick and dirty” oceny użyteczności przez użytkowników, którzy:
 - realizują scenariusz wykorzystania systemu
 - wypełniają ankietę ewaluacyjną
- Skala Likerta: (od 1 do 5 - od „zdecydowanie się nie zgadzam” do „zdecydowanie się zgadzam”)

SUS – pytania/stwierdzenia ankiety

- Q1** Myślę, że chciałbym często używać tego systemu
- Q2** Uważam, że system jest niepotrzebnie zbyt skomplikowany
- Q3** Uważam, że system jest łatwy w użyciu
- Q4** Myślę, że będę potrzebował pomocy specjalisty, aby móc w pełni używać tego systemu
- Q5** Uważam, że funkcje systemu są dobrze zintegrowane
- Q6** Sądzę, że w systemie jest zbyt dużo niespójności
- Q7** Oceniam, że większość osób bardzo szybko nauczy się używać tego systemu
- Q8** Uważam, że system jest bardzo niewygodny w użyciu
- Q9** Czułem się pewnie korzystając z systemu
- Q10** Musiałbym sporo nauczyć się, zanim mógłbym zacząć swobodnie pracować z tym systemem

SUS – obliczenia

$$SUS = \left(\sum_{i=1,3,5,7,9} (S_i - 1) + \sum_{i=2,4,6,8,10} (5 - S_i) \right) * 2,5$$

gdzie : i – numer pytania; S_i – ocena i -tego pytania

- Wartość SUS: 0-100
- Średnia wartość SUS dla 500 różnych systemów wyniosła **68** → punkt odniesienia w ocenie
- SUS w rzeczywistości bada dwa czynniki:
użyteczność (8 pozycji) i możliwość nauczania
(2 pozycje – konkretnie pozycje numer: 4 i 10)

Organizacja eksperymentu z użytkownikiem

Etapy eksperymentu

1. Przygotowanie eksperymentu (tj. plan, scenariusze, sprzęt, oprogramowanie, miejsce itd.)
2. Dobór i pozyskanie grupy badawczej
3. Przeprowadzenie eksperymentu z grupą badawczą
4. Analiza i uogólnienie wyników eksperymentu
5. Wyciągnięcie wniosków i wypracowanie rekomendacji

Dobór grupy badawczej

- Grupa badawcza – użytkownicy
- Użytkownicy vs. potencjalni użytkownicy → Persony
- Czynniki wpływające na dobór grupy badawczej:
 - cel badań
 - doświadczenie użytkowników z interfejsem i/lub poprzednimi wersjami oprogramowania
 - zaburzenie widzenia
 - wiek i poziom wykształcenia
 - pochodzenie społeczne i kulturowe
 - wymagana liczebność grupy

Dobór grupy badawczej – liczebność

- Zasady:
 - 80% problemów wykrywa 5 osób (wg. badań Nielsena)
 - 20 osób wykrywa praktycznie wszystkie
 - „optymalnie”: 5-8 osób
- Różne osoby – różne grupy badawcze (każda po 5-8 osób)
- Badania niekontrolowane (np. ankietowe) wymagają większej liczby użytkowników

Dobór grupy badawczej – stres i przeciwdziałanie

- Uczestnicy mogą być poddani sytuacji stresowej w trakcie badań – należy jej unikać (chyba że jest celem badań)
- Stres wywołuje:
 - traumę
 - odczucie bycia sprawdzanym czy testowanym
 - odczucie bycia porównywanym z innymi
 - współzawodnictwo z innymi
 - odczucie bycia obserwowanym i ocenianym przez mądrzejszych badaczy
- Należy niwelować (zapewnić: komfort, prywatność, dyskrecję, szkolenie, tłumaczenie)

Scenariusz

- Zestaw zadań
- Każde zadanie ma:
 - cel: Co będzie badane w zadaniu?
 - pomiar ilościowy: metryki
 - rys historyczny (opis sytuacji)
 - lista kroków-zadań: do wykonania przez uczestnika
 - pomiar jakościowy: uczucia, trudności, komentarze
- Sprawdzony i realizowalny

Scenariusz – szczegóły

- Musi być napisany (na papierze lub na ekranie)
- Bądź zwięzły i precyzyjny
- Rys historyczny powinien być realistyczny
- Dopasowany do doświadczeń uczestników
- Należy unikać używania tego samego słownictwa co w interfejsie
- Zadania powinny być w kolejności, jeśli są od siebie zależne
- Jeśli kolejność sekwencyjna nie jest konieczna, zróżnicuj zadania pod kątem ważności
- Nie może być informacji o sposobie wykonywania zadań

Metody oceny bez udziału użytkownika

Techniki oceny jakości interfejsów bez udziału użytkowników

- Analiza ekspercka (ang. *Expert Review*)
- Uproszczona wędrówka poznawcza (ang. *Cognitive Walkthrough*)
- Rozwinięta wędrówka poznawcza (ang. *Pluralistic Walkthrough*)
- Ocena heurystyczna (ang. *Heuristic Evaluation*)
- Inspekcja standardów (ang. *Guidelines Inspection*)
- Rekomendacje: W3C i WCAG

Analiza ekspercka

- Opublikowano wiele wyników badań dotyczących projektowania interfejsu użytkownika
- Idea analizy: poszukiwaniu w literaturze analogicznych aspektów projektu
- Oszczędza prowadzenie własnych eksperymentów
- Wyniki są przenoszone wiarygodnie tylko wtedy, gdy kontekst (użytkownicy, założenia) jest bardzo podobny

Uproszczona wędrówka poznawcza

- Cel:
ocena procesu uczenia się oraz poznanie
płynności procesu realizacji zadań z użyciem
oprogramowania
- Metoda:
ekspert realizuje scenariusz wykorzystania
oprogramowania
- Pytania badawcze
- Ocena pytań w skali Likerta: 1...5

Uproszczona wędrówka poznawcza – zakres oceny

- Jak interakcja wpływa na użytkownika?
 - Jakich procesów poznawczych będzie potrzebowała?
 - Jakie problemy może wiązać się z nauką/wykonaniem tego kroku?
 - Czy system pomaga użytkownikowi przejść od celów do intencji i działań?
-
- Ocenia ekspert

Rozwinięta wędrówka poznawcza

- Rozszerzenie uproszczonej wędrówki poznawczej
- Zespół ekspertów rozszerza się o:
 - użytkowników
 - programistów
 - innych członków zespołu projektowego
- Pozostałe działania jak w uproszczonej wędrówce poznawczej

Ocena heurystyczna

- Eksperti oceniają oprogramowanie wykorzystując standardowy zestaw zasad dobrego interfejsu (tzw. heurystyki)
- Heurystyka – zbiór optymalnych (quasi-optymalnych) zasad wynikających z:
 - doświadczenia
 - zdrowego rozsądku
 - dobrych praktyk
 - fizycznych właściwości człowieka
- Heurystyka uwzględnia kontekst użycia oprogramowania

Ocena heurystyczna – etapy

1. Planowanie działań (wybór heurystyki i kontekstu użycia oprogramowania oraz zaznajomienie z nimi ekspertów, a także opracowanie scenariuszy)
2. Realizacja badań (eksperci niezależnie od siebie wykonują zadania i odnotowują wszystkie odchylenia od zasad podanych w heurystyce, ocena istotności problemów)
3. Analiza wyników (scalenie list opracowanych przez ekspertów i oszacowaniu stopy wykrytych problemów)
4. Opracowanie raportu

Ocena heurystyczna – efektywność

- Jeden ekspert wykrywa ok. 35% problemów
- Badanie realizowane na wczesnych etapach projektowania interfejsu pozwala na dostarczenie informacji zwrotnej
- Zmniejszenie liczby problemów przed testowaniem z udziałem użytkowników → zmniejszenie kosztów testowania z udziałem użytkowników

Heurystyki Nielsen-Molicha

- Opracowane na podstawie badań statystycznych, dotyczących prawidłowej interakcji człowiek-maszyna o możliwie najszerszym spektrum zastosowań w projektach informatycznych
- Lista 10 zaleceń, spełnienie których jest oceniane przez ekspertów

Heurystyki Nielsena-Molicha – lista (1)

1. Widoczny status systemu – system powinien zawsze informować użytkownika o swoim stanie z użyciem stosownych i zrozumiałych elementów systemu. Przy czym, ważne jest by oprogramowanie informowało odpowiednio szybko, bez zbędnych opóźnień
2. Zgodność pomiędzy systemem a rzeczywistością – system powinien być zrozumiały dla użytkownika nie tylko poprzez wersję językową, ale także poprzez dobór odpowiedniej terminologii. Wszystkie informacje powinny być podawane w logicznym, naturalnym porządku. Zrozumiałe powinny być również konwencje multimedialne (np. metafory graficzne)

Heurystyki Nielsena-Molicha – lista (2)

3. Użytkownik musi mieć kontrolę i swobodę działań – zarówno w przypadku nawigacji w oprogramowaniu, jak i w przypadku wyboru błędnej opcji, użytkownik musi mieć zapewnioną prostą możliwość powrotu do poprzedniego położenia. Ważne jest by ta „ucieczka” nie wymagała zbyt długiego dialogu, a opcja ją umożliwiająca powinna być jasno oznaczona i łatwo dostępna
4. Zachowanie jednakowych konwencji w obrębie serwisu – wszystkie słowa, symbole i elementy graficzne powinny być stosowane w jednakowy sposób w obrębie całego oprogramowania. Użytkownik nie powinien być zaskakiwany nietypowymi elementami (zarówno graficznymi jak i behawioralnymi) dialogu. Najlepiej jest w tym przypadku używać konwencji platformy na jakiej ma działać oprogramowanie

Heurystyki Nielsena-Molicha – lista (3)

5. Zapobieganie błędom – dialog z użytkownikiem i jego poszczególne elementy powinien być tak zrealizowany by zapobiegać błędom. Twórcy powinni być ukierunkowani na ochronę użytkownika i aplikacji przed możliwością popełnienia błędów, a mniej na ich obsługę
6. Rozpoznawanie, a nie zapamiętywanie – interfejs powinien być tak zaprojektowany by użytkownicy nie musieli pamiętać informacji przechodząc z jednej do drugiej części aplikacji. Wszystkie potrzebne w danej sytuacji informacje (np. wprowadzone dane) i instrukcje powinny być widoczne na ekranie tak, aby nie obciążać pamięci krótkookresowej użytkownika

Heurystyki Nielsena-Molicha – lista (4)

7. Elastyczność i efektywność – doświadczeni użytkownicy powinni mieć możliwość przyspieszonego dostępu do używanych funkcji. Powinni mieć także możliwość wyboru najbardziej im pasującego sposobu wykonywania typowych zadań. Wybór ten powinien być dokonywany z wielu możliwych metod
8. Estetyka i minimalizm interfejsu – interfejs nie powinien zawierać elementów zbędnych w danym momencie, utrudniających zrozumienie i przyswojenie informacji. Jednocześnie interfejs powinien mieścić się w kanonach estetyki obowiązujących powszechnie u użytkownika

Heurystyki Nielsena-Molicha – lista (5)

9. Właściwa obsługa błędów – wszystkie komunikaty powinny być napisane w sposób prosty i zwięzły, aby ułatwić użytkownikowi ich zrozumienie. Powinny one również wskazywać na typ problemu i podawać sposób jego rozwiązania. Nie powinno się posługiwać kodami błędów
10. Pomoc i dokumentacja – oprogramowanie powinno być tak wykonane, by mogło być używane bez sięgania do dodatkowej dokumentacji (interfejs powinien być samo- wyjaśniający). Tym niemniej oprogramowanie powinno zapewnić pomoc i dokumentację w zakresie niezbędnych do zrealizowania zadań przez użytkownika. Dostęp do tych informacji powinien być prosty i intuicyjny oraz niezajmujący więcej czasu niż jest to konieczne

Inspekcja standardów

- Inspekcja z wykorzystaniem list kontrolnych
- Formalna kontrola interfejsu na zgodność z wymaganiami
- Technika: lista kontrolna
- Lista kontrolna:
 - specjalny kwestionariusz → lista kontrolna
 - zawartość: kilka, kilkanaście monotematycznych sekcji, które odpowiadają wymaganiom głównym, po kilka, kilkanaście pytań w każdej
 - odpowiedzi: binarne (Tak lub Nie, Spełnia lub Nie) lub też może oceniać poziom spełnienia wymagań (skala 1-5 lub też 2-5)

Inspekcja standardów – działania

1. Wybór odpowiedniej listy kontrolnej
2. Przegląd wstępny oprogramowania, jego funkcji oraz sposobu obsługi
3. Inspekcja oprogramowania
4. Opracowanie wyników
5. Przygotowanie raportu

Inspekcja standardów – wyniki

- Procentowy poziom spełnienia wymagań głównych
- Uśredniony poziom spełnienia wymagań (w wybranej skali) głównych i całkowity dla całej listy
- Lista niespełnionych wymagań, wraz z oceną ich istotności oraz propozycjami poprawy

Lista LUT (1)

- Heurystyka – lista kontrolna
- Obszary → Podobszary → Pytania
- Ocena pytań:

Ocena	Opis
1	Wystąpiły krytyczne problemy dotyczące użyteczności, uniemożliwiające korzystanie z aplikacji bądź zniechęcające do korzystania z niej
2	Napotkano poważne problemy dotyczące użyteczności mogące uniemożliwić większości użytkowników realizację zadań
3	Wystąpiły drobne problemy związane z użytecznością, które pojedynczo nie stanowią utrudnień dla większości użytkowników, jednak ich nagromadzenie może wpłynąć na jakość pracy użytkownika
4	Zidentyfikowano pojedyncze drobne problemy związane z użytecznością mogące obniżyć jakość pracy z aplikacją (np. słaba czytelność tekstu)
5	Nie stwierdzono problemów związanych z użytecznością ani mających wpływ na jakość pracy użytkownika

Lista LUT (2)

Obszar	Podobszar	Pytanie
Nawigacja i struktura	Łatwość nawigowania	Czy dostęp do wszystkich sekcji aplikacji jest łatwy i intuicyjny?
		Czy dostęp do wszystkich funkcji aplikacji jest łatwy i intuicyjny?
	Hierarchia inform.	Czy hierarchia informacji nie jest zbyt głęboka?
	Struktura informacji	Czy struktura informacji jest przemyślana?
		Czy struktura informacji jest spójna?
		Czy struktura informacji jest zrozumiała dla użytkownika?
	Elementy ekranu	Czy wspierają proces nawigacji?
Komunikaty, <i>feedback</i> , pomoc dla użytkownika	Komunikaty (ogólne)	Czy dostarczają wystarczająco dużo informacji zwrotnych dotyczących stanu operacji wykonywanych przez użytkownika?
	Komun. o błędach	Czy zawierają podpowiedzi dotyczące rozwiązania problemu?
	Informacje zwrotne i pomoc	Czy pojawiają się w miejscach, gdzie mogą być potrzebne?
		Czy treść pomocy jest dostępna dla przeciętnego użytkownika?
		Czy treść pomocy jest zrozumiała dla przeciętnego użytkownika?
		Czy prezentowane podpowiedzi bądź rozwiązania problemów są możliwe do wykonania przez zwykłego użytkownika?

Lista LUT (3)

Interfejs aplikacji	Layout	Czy <i>layout</i> jest czytelny?
		Czy <i>layout</i> jest dostosowany do różnych rozdzielczości?
		Czy <i>layout</i> jest dostosowany do urządzeń mobilnych?
		Czy układ graficzny jest spójny?
		Czy <i>layout</i> wspiera realizację zadań?
	Dobór barw	Czy kontrast pomiędzy tekstem a tłem jest odpowiedni?
		Czy dobór barw umożliwia skorzystanie z aplikacji przez osoby z zaburzeniami widzenia barw?
		Czy dobór barw umożliwia skorzystanie z aplikacji przy wykorzystaniu różnego rodzaju wyświetlaczy?
Treść podstron	Teksty	Czy są zrozumiałe dla użytkownika?
	Nazewnictwo	Czy używane w aplikacji nazewnictwo jest spójne?
		Czy używane w aplikacji nazewnictwo jest zrozumiałe?
	Etykiety	Czy używane w interfejsie etykiety dostarczają wystarczająco dużo informacji?
		Czy elementy interfejsu posiadają niezbędne etykiety?
Wprowadzanie danych	Formularze	Czy posiadają czytelny projekt?
		Czy umożliwiają wprowadzenie niezbędnych danych?
		Czy są dostosowane do urządzeń mobilnych?
	Dane	Czy zwykły użytkownik nie ma trudności z wprowadzeniem danych do formularza?
		Czy formularze posiadają elementy podpowiedzi dotyczące wprowadzanych danych (m.in. formatu, zakresu)?
		Czy formularze posiadają elementy walidujące wprowadzone dane?

Punkty WUP dla listy LUT

- Ang. *Web Usability Points*
- Metryka oceny jakości aplikacji webowej

$$WUP = \frac{1}{n_a} \sum_{i=1}^{n_a} \frac{1}{s_i} \sum_{j=1}^{s_i} \frac{1}{q_{ij}} \sum_k^{q_{ij}} p_{ijk}$$

n_a – liczba obszarów

s_i – liczba podobszarów w obszarze i

q_{ij} – liczba pytań w obszarze i oraz podobszarze j

p_{ijk} – ocena pytania o numerze k w podobszarze j obszaru i

- Zakres wartości oceny: 1...5 (im większa wartość tym lepsza jakość interfejsu)

Rekomendacje W3C

- W3C standardy
- <https://www.w3.org/standards/>
- Grupa standardów Web Design and Applications:
 - HTML & CSS
 - JavaScript Web APIs
 - Graphics
 - Audio and Video
 - Accessibility -> Web Content Accessibility Guidelines (WCAG)
 - Internationalization
 - Math on Web
 -
- Rekomendacje: Easy Checks – A First Review of Web Accessibility (<https://www.w3.org/WAI/test-evaluate/preliminary/#headings>)

Rekomendacje WCAG

- Web Content Accessibility Guidelines (WCAG) 2.1
- <https://www.w3.org/TR/WCAG21/>
- Otwarty standard w obszarach:
 1. Postrzegalność
 2. Funkcjonalność
 3. Zrozumiałość
 4. Kompatybilność
- Poziomy zgodności: A, AA i AAA; ich wypełnienie jest mierzalne
- Polskie tłumaczenie: <https://wcag21.lepszyweb.pl/>

Narzędzia wspomagające ocenę jakości interfejsu oprogramowania

Narzędzia komputerowe

- Walidatory W3C:
 - <https://validator.w3.org/>
 - <http://jigsaw.w3.org/css-validator/>
 - Inne: <https://w3c.github.io/developers/tools/>
- Elektroniczne listy kontrolne
- Walidator rekomendacji WCAG → WAVE

W3C Html checker

Zintegrowany

W3C[®] Markup Validation Service
Check the markup (HTML, XHTML, ...) of Web documents

Validate by URI Validate by File Upload Validate by Direct Input

Validate by URI


Validate a document online:

Address:

▶ More Options


Check

This validator checks the [markup validity](#) of Web documents in HTML, XHTML, SMIL, MathML, etc. If you wish to validate specific content such as [RSS/Atom feeds](#) or [CSS stylesheets](#), [MobileOK content](#), or to [find broken links](#), there are [other validators and tools](#) available. As an alternative you can also try our [non-DTD-based validator](#).


 The W3C validators rely on community support for hosting and development.

[Donate](#) and help us build better tools for a better web.

Home About... News Docs Help & [FAQ](#) Feedback Contribute

W3C[®]  open source

COPYRIGHT © 1994-2013 W3C[®] (MIT, ERCIM, KEIO, BEIHANG), ALL RIGHTS RESERVED. W3C LIABILITY, TRADEMARK, DOCUMENT USE AND SOFTWARE LICENSING RULES APPLY. YOUR INTERACTIONS WITH THIS SITE ARE IN ACCORDANCE WITH OUR PUBLIC AND MEMBER PRIVACY STATEMENTS.



W3C Html checker – rezultaty

eye tracking – Szukaj w Go... x | Eyetracking - gdzie skupia... x | Baza Publikacji Pracownik... x | Tłumacz Google x | Showing results for https:// x | WAVE Report of Politechnik... x | + -

← → ↻ https://validator.w3.org/nu/?doc=https%3A%2F%2Fpollub.pl%2F

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <https://pollub.pl/>

Checker Input

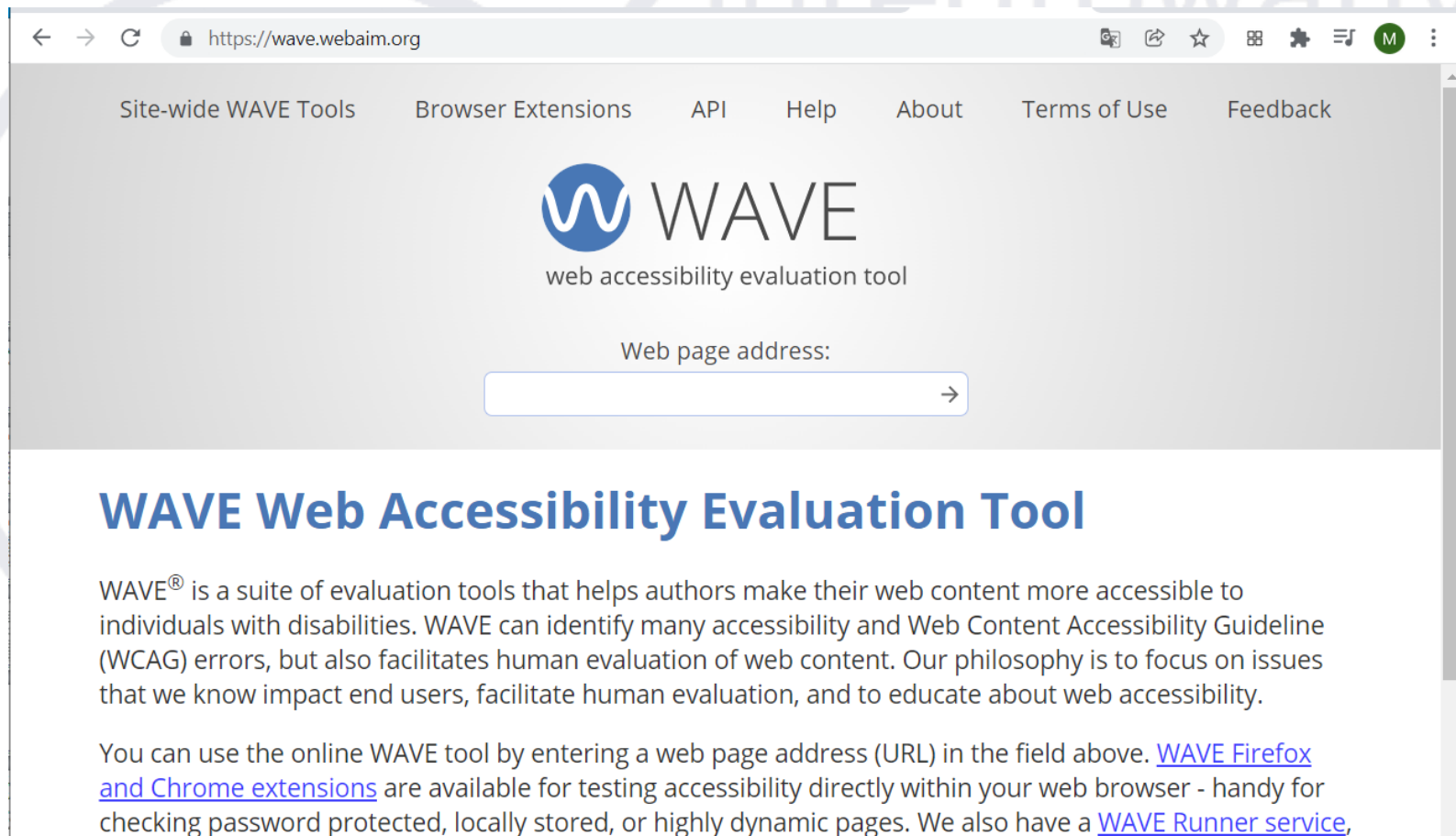
Show ☐ source ☐ outline ☐ image report

Check by

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

- Warning** Section lacks heading. Consider using `h2` - `h6` elements to [add identifying headings to all sections](#).
From line 74, column 5; to line 74, column 85
`/ul> <section class="colored py-0 top-header" style="background-color: rgb(0, 0, 0);"><div c`
- Error** Bad value for attribute `srcset` on element `img`: No width specified for image `/thumb/VPREjGBETDzQRHQ4BH...k1hdQg/grafika-film_1.png`. (Because one or more image candidate strings specify a width (e.g., the string for image `/thumb/VPREjGBETDzQRHQ4BH...k1hdQg/grafika-film_1.png`), [all image candidate strings must specify a width](#).)
From line 249, column 69; to line 250, column 386
``
- Error** Bad value for attribute `srcset` on element `img`: No width specified for image `/thumb/TPREjGBETDzQRHQ4BH...JwTBV2A8V01ISg8/slide.jpg`. (Because one or more image candidate strings specify a width (e.g., the string for image `/thumb/TPREjGBETDzQRHQ4BH...QAWmR0G0BBWFBI/slide1.png`), [all image candidate strings must specify a](#)

WAVE



WAVE Web Accessibility Evaluation Tool

WAVE® is a suite of evaluation tools that helps authors make their web content more accessible to individuals with disabilities. WAVE can identify many accessibility and Web Content Accessibility Guideline (WCAG) errors, but also facilitates human evaluation of web content. Our philosophy is to focus on issues that we know impact end users, facilitate human evaluation, and to educate about web accessibility.

You can use the online WAVE tool by entering a web page address (URL) in the field above. [WAVE Firefox and Chrome extensions](#) are available for testing accessibility directly within your web browser - handy for checking password protected, locally stored, or highly dynamic pages. We also have a [WAVE Runner service](#),

WAVE – rezultaty analizy

eye tracking – Szukaj w Go... Eyetracking - gdzie skupia... Baza Publikacji Pracownik... Tłumacz Google W3 Easy Checks – A First Review WAVE Report of Politechni...
https://wave.webaim.org/report#/www.pollub.pl

The following apply to the entire page:

WAVE powered by WebAIM
web accessibility evaluation tool
Address: www.pollub.pl
Styles: OFF ON

Summary

12 Errors	5 Contrast Errors
17 Alerts	20 Features
19 Structural Elements	49 ARIA

View details

Poczta Biblioteka Intranet Stara Strona Książka adresowa Kontakt

Wyszukaj

POLITECHNIKA LUBELSKA

Godło Polski

UCZELNIA REKRUTACJA

STUDENCI NAUKA

WSPÓLPRACA

h2 Listy do Świętego Mikołaja

Wesołych Świąt

Code



Pytania?

Dziękuję

Materiały zostały opracowane w ramach projektu
„Zintegrowany Program Rozwoju Politechniki Lubelskiej – część druga”,
umowa nr **POWR.03.05.00-00-Z060/18-00**
w ramach Programu Operacyjnego Wiedza Edukacja Rozwój 2014-2020
współfinansowanego ze środków Europejskiego Funduszu Społecznego