

GUI - projektowanie interfejsów

dr Przemysław Juszczuk

Katedra Inżynierii Wiedzy, Uniwersytet Ekonomiczny w Katowicach

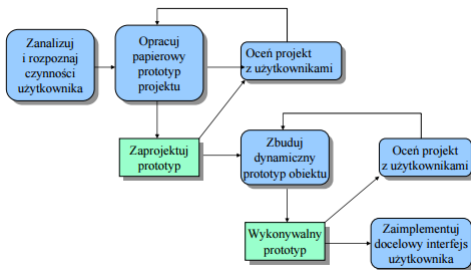
Wykład 2

Interfejs użytkownika

- dobry projekt interfejsu jest warunkiem niezbędnym powodzenia systemu;
- interfejs trudny w użyciu/posiadający liczne błędy w najlepszym wypadku doprowadzi do wielu pomyłek;
- w najgorszym wypadku użytkownicy mogą odmówić jego używania pomimo jego dobrej funkcjonalności;
- użytkownicy mogą mieć problemy z użytkowaniem systemu, jeżeli przestawine w nim informacje są błędne/niespójne.

Właściwości graficznego interfejsu

- okna - umożliwiają wyświetlanie informacji;
- ikony - reprezentują różnego rodzaju informacje;
- menu - obiekty umożliwiające wybranie określonej funkcji systemu;
- wskazywanie - możliwość wyboru konkretnego elementu z menu;
- grafika - elementy graficzne mogą zostać połączone z informacjami tekstowymi.



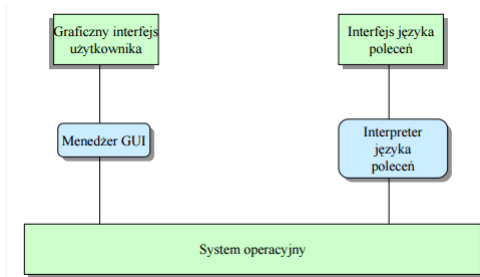
Rysunek: Proces projektowania interfejsu

Ogólna charakterystyka użytkownika

- Dane podstawowe: wiek, poziom wykształcenia;
- Klasyfikacja: przynależność organizacyjna, doświadczenie i umiejętności;
- Preferencje i uprzedzenia użytkownika;
- Cel korzystania z systemu;
- Podział na role w systemie;
- Scenariusze zadań dostępne dla każdej roli w systemie.

Rodzaje interakcji z użytkownikiem

- działanie bezpośrednie - użytkownik porozumiewa się bezpośrednio z obiektami - np. przeciągnięcie pliku do kosza;
- wybór z menu - wybór jednej z opcji dostępnych w obiekcie menu;
- wypełnienie formularza - wypełnienie pól/odpowiedź na pytania prowadzi do pewnej akcji;
- język poleceń - wydawanie specjalnych poleceń (wraz z parametrami) umożliwiających podjęcie akcji systemu;
- język naturalny - wydawanie poleceń w języku naturalnym.



Rysunek: Podstawowy podział interfejsów

Podstawowe zasady projektowania interfejsu

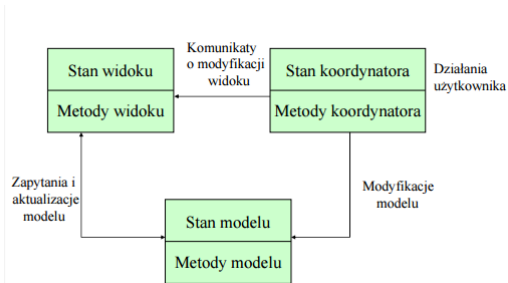
- Wygląd - interfejs powinien być podzielony na obszary przeznaczone do różnych celów;
- Uświadamianie zawartości - interfejs powinien uświadamiać użytkownika, w którym miejscu się znajdują i co oznaczają poszczególne elementy;
- Estetyka - równowaga pomiędzy funkcjonalnością a czytelnością;
- Doświadczenie użytkownika - uwzględnienie możliwości obsługi przez początkującego oraz zaawansowanego użytkownika;
- Spójność - interfejs powinien umożliwiać użytkownikowi przewidywanie skutków podejmowanych działań;
- Minimalizacja wysiłku - ograniczenie liczby kroków wiodących do celu do minimum.

Zasady projektowania interfejsu II


- zasada zbliżenia do użytkownika - użytkownik nie powinien być zmuszony do adaptowania się do systemu;
- potwierdzanie akcji destrukcyjnych;
- zapewnienie możliwości anulowania - czyli przywrócenia stanu systemu sprzed wykonania polecenia;

Czynniki określające funkcjonalność - projektowanie


- czynniki ludzkie - wpływ stresu, zmysły człowieka, proces zapamiętywania i zapominania;
- interakcja HCI - Human Computer Interaction - zagadnienia związane z logiką i funkcjonalnością parametrów, korzystanie z operacji we/wy, ergonomia korzystania z komputera;
- Projektowanie zorientowane na użytkownika - nie opiera się na zasadzie "klient ma zawsze rację".




Rysunek: MVC dla interakcji z użytkownikiem



[sieć](#) | [poczta](#) | [mapa](#) | [english](#) | [pzyceani](#)


UNIWERSYTET ŚLĄSKI
 W KATOWICACH



STUDENT | **KANDYDAT** | **BIZNES**
ABSOLWENT | **DOKTORANT** | **PRACOWNIK**

Uczelnia

.....

Studia

.....

Nauka

.....


Współpraca

.....

Media i kultura

.....





Kontakt



Dzień
Młodego Naukowca
 Uniwersytet Śląski

„Dzień Młodego Naukowca”

● ● ● ● ● ● ● ●

Polecamy	Wiadomości	Kalendarz
 <p>Wreczenie dyplomów najlepszym absolwentom oraz nagrody w plebiscyście „Absolwent z Pasją”</p> <p>W poniedziałek 30 listopada 2015 roku o godz. 16.00 w auli Im. Kazimierza Lepszego w rektoracie Uniwersytetu Śląskiego odbędzie się uroczystość wręczenia dyplomów najlepszym absolwentom roku akademickiego 2014/2015 oraz nagrody w trzeciej edycji plebiscytu „Absolwent z Pasją”.</p>	 <p>Posiedzenie Zarządu European University Association w Genewie</p>  <p>Jubileusz 35-lecia NSZZ „Solidarność” na Uniwersytecie Śląskim</p>  <p>Pracownik SWFIS US Mistrzem Świata Karate Kyokushin</p>	<p>27.11 Posiedzenie Zarządu European University Association w Genewie</p> <p>27.11 Uroczystość rozdania stypendystom dyplomów Prezesa Rady Ministrów na rok szkolny 2015/2016</p> <p>27.11 Profesor prof. Andrzej Kowalczyk uczestnikiem obchodów Dnia Górnika na Politechnice Śląskiej</p>

Rysunek: Przykładowa strona z podziałem na obszar nawigacyjny

Zasady podziału

- każdy obszar interfejsu powinien mieć jasno wytyczone granice;
- każdy obszar powinien mieć określone przeznaczenie;
- każdy obszar powinien zawierać informacje potrzebne do realizacji wybranego zadania;
- obszary informacyjne powinny być uszeregowane w kolejności przetwarzania informacji przez użytkownika;

Uświadamianie zawartości

- wszystkie okna oraz sekcje powinny mieć tytuły jednoznacznie identyfikujące ich zawartość;
- menu powinno pokazywać ścieżkę - wskazywać sposób, w jaki użytkownik dostał się do danego punktu;
- przyciski powinny mieć napis jednoznacznie identyfikujący ich funkcje;
- forma informacji na sąsiadujących obszarach powinna być różna, np. tekst - grafika;
- jeżeli informacje nie są różne - powinny być oddzielone dodatkowym elementem;
- każde pole edycji musi mieć etykietę jednoznacznie identyfikującą to pole;
- pola edycji, których format nie jest oczywisty (np. data) muszą mieć dodatkowe oznaczenie formatu;

Estetyka

- interfejs powinien być funkcjonalny oraz mieć przyjemny wygląd;
- ilość wolnego miejsca pomiędzy elementami powinna być zależna od doświadczenia użytkownika - im mniej doświadczenia, tym więcej wolnego miejsca;
- należy unikać tworzenia dużych raportów i formularzy (kilkadziesiąt pól);
- tekst główny prezentowany czcionką 8-10 pkt;
- należy unikać stosowania dwóch różnych czcionek. Zdecydowanie unikać stosowania czcionek ozdobnych;
- stosowane kolory powinny być stonowane i nie powinny "męczyć" oka;
- kolor nie może być jedynym wyróżnikiem informacji.

Doświadczenie użytkownika a interfejs

- interfejs powinien być na tyle łatwy, aby początkujący użytkownik był w stanie go opanować;
- interfejs powinien ułatwiać i przyspieszać wykonywanie działań zaawansowanych użytkowników;
- menu powinno prezentować wszystkie dostępne funkcje;
- menu na każdym poziomie powinno być ograniczone i nie zawierać nadmiarowych informacji;
- najczęściej stosowane opcje powinny być dostępne bezpośrednio poprzez przyciski narzędziowe;
- przyciski powinny być logicznie pogrupowane;
- w przypadku bardziej złożonych aplikacji wskazane jest umożliwienie użytkownikowi konfiguracji paska menu;
- system podpowiedzi i skrótów klawiszowych dla bardziej złożonych aplikacji;

Spójność

- interfejs powinien być spójny dla zapewnienia przewidywalności podejmowanych działań przez użytkownika;
- wszystkie elementy: raporty i formularze powinny być zaprojektowane w jednolity sposób;
- jeżeli dana aplikacja powiązana jest z innymi narzędziami - to wszystkie powiązane elementy powinny być spójne.

Minimalizacja wysiłku

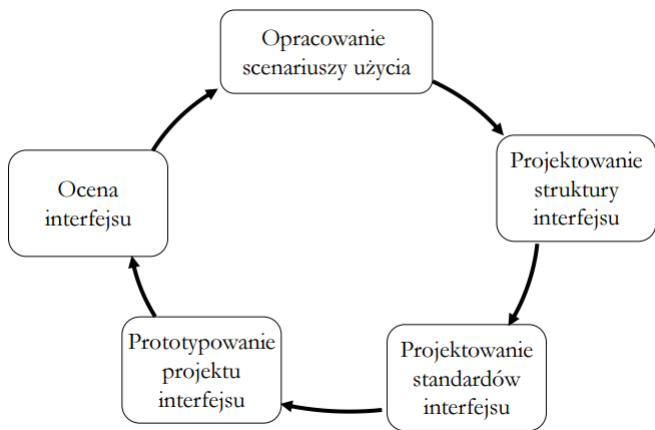
- interfejs powinien być przygotowany tak, aby minimalizować wysiłek użytkownika włożony w wykonanie czynności;
- 3 kliknięcia - czyli zalecana maksymalna liczba kliknięć służąca do wybrania interesującej użytkownika funkcji;
- dla bardziej złożonych aplikacji zaleca się łączenie kilku funkcji w jedną.

Dobór kolorów i schematów kolorów

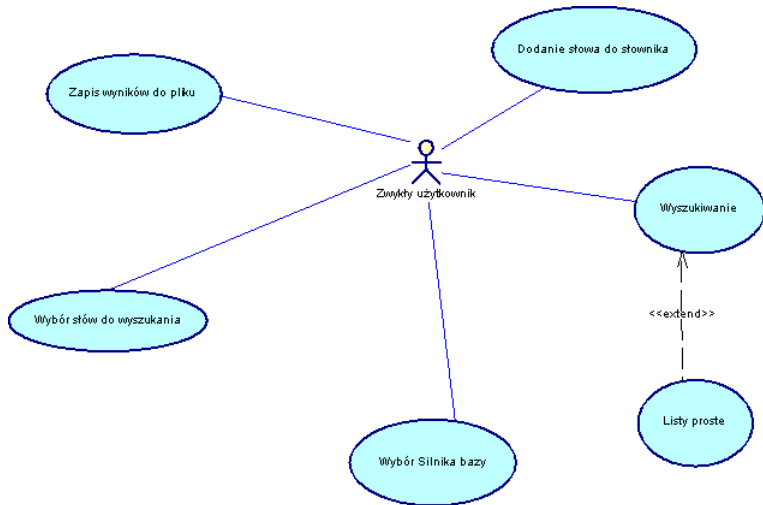
- można stosować intensywne kolory, ale nie intensywne;
- najlepiej, by kolory były do siebie dopasowane (np. niebieski i zielony, czerwony i żółty itp.) lub kontrastowe: biały i czarny;
- kolor tła nie powinien być zbyt intensywny;
- tło musi być spokojne, najlepiej jednolite, ewentualnie ze stopniowymi przejściami koloru;
- stosowanie drobnych elementów i obiektów jako tła nie jest zalecane;
- niebieski jest bardziej "prestizowym" kolorem (podobnie jak większość chłodnych kolorów i fioletowy).

Dobór kolorów i schematów kolorów II

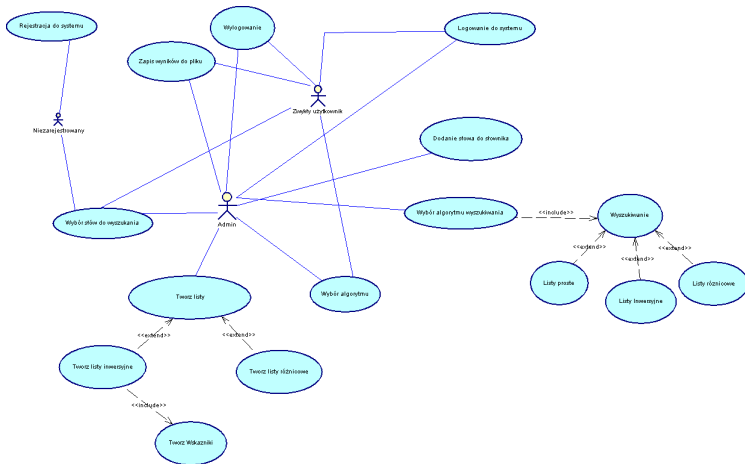
- ograniczenie liczby stosowanych kolorów;
- zmiany kolorów do oznaczenia zmian stanów systemu;
- powiązanie kolorów z realizacją zadań - zadania podobne powinny być oznaczone kolorem podobnym;
- zwracanie uwagi na niektóre związki kolorów np. czerwony z niebieskim - które powodują szybkie zmęczenie oczu.



Rysunek: Projektowanie interfejsu



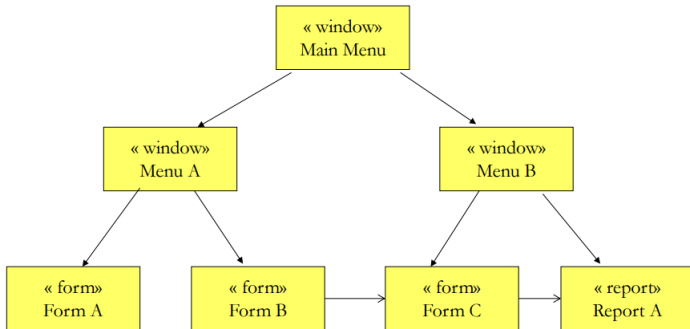
Rysunek: Use case 1



Rysunek: Use case 2

WND - Window Navigation Diagram

- każdy element wykresu reprezentuje pojedynczy komponent - menu lub formatkę;
- przejścia pomiędzy komponentami oznaczone są jako pojedyncza lub podwójna strzałka;
- strzałka pojedyncza oznacza, że informacja zwrotna do komponentu nadrzędnego nie jest potrzebna;
- każdy komponent opisany jest jako stereotyp, np. << *form* >>;
- każdy komponent ma etykietę.



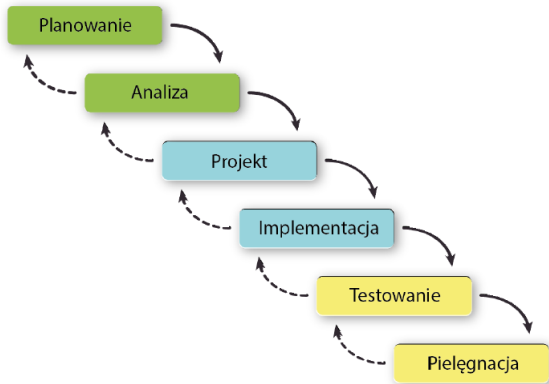
Rysunek: Window Navigation Diagram WND

Jaki model wybrać?

- zależny od dostępnych środków;
- jaki jest sposób organizacji zespołu;
- jakie narzędzia są dostępne;
- na ile można korzystać z gotowych rozwiązań?

Podejście podstawowe

- 1 Zaproponuj koncepcję;
- 2 Wykonaj projekt;
- 3 Uzgodnij z klientem/zespołem.
- 4 Zakończenie / powrót do punktu drugiego.



Rysunek: Model kaskadowy

*Winston Royce, Managing the Development of Large Software Systems, 1970 rok.

Cechy modelu

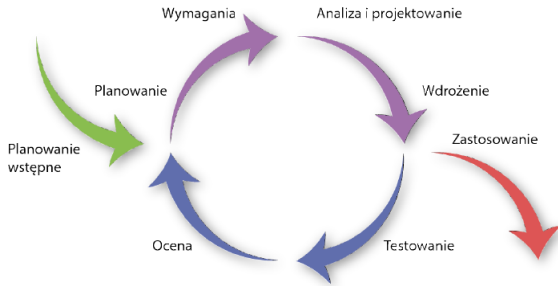
- iteracyjnie następujące po sobie fazy utrudniają elastyczne działanie;
- ewentualne powtórzenie iteracji jest kosztowne;
- problem przy komunikacji z klientem;
- żeby przejść dalej, to musimy zakończyć bieżącą fazę;
- może być skuteczny, jeżeli wymagania są poprawnie zdefiniowane;
- standaryzacja podejścia – dokładnie wiemy, jaki etap jest następny;
- ułatwia organizację;
- połowa zespołu może czekać druga połowa zakończy poprzednią fazę.

Model kaskadowy z nawrotami

- po każdej fazie następuje jej weryfikacja;
- powrót nie jest obligatoryjny – daje możliwość korekty błędów;
- konieczny dodatkowy nakład czasowy związany z weryfikacją;
- łatwiejsze (niż w klasycznym modelu kaskadowym) planowanie.

Model przyrostowy

- Przygotowanie ogólnych wymagań oprogramowania na niskim poziomie szczegółowości oraz wyodrębnienie funkcji systemu;
- Wyselekcjonowanie funkcji przeznaczonych do realizacji w najbliższym przyroście;
- Uszczegółowienie wybranych funkcji - szczegółowe zaprojektowanie;
- Implementacja wybranych funkcji;
- Testowanie wersji wzbogaconej o nowe funkcje dodane w przyroście;
- Dostarczenie wersji demo wzbogaconej o funkcje z aktualnego przyrostu;
- Powtarzanie przyrostów aż do ukończenia projektu.



Rysunek: Model przyrostowy

Model z prototypem

- ogólne określenie wymagań;
- budowa i weryfikacja prototypu;
- dobry moment na ewentualne porzucenie koncepcji / zmianę założeń i ponowne przygotowanie prototypu;
- szczegółowe określenie wymagań (już po pozytywnej weryfikacji prototypu);
- realizacja pełnej wersji;
- prototyp istotnie zwiększa nakład czasowy a także generuje dodatkowe koszty.

Model RAD - Rapid Application Development

- podział projektu na niezależne składowe;
- równoległa realizacja poszczególnych funkcjonalności bazując np. na modelu kaskadowym;
- zespoły pracują niezależnie nad różnymi funkcjami;
- konieczność integracji podzespołów na finiszu projektu.

Model reuse

- możliwość wykorzystania gotowych rozwiązań/komponentów;
- znaczna redukcja kosztów i czasu realizacji projektu;
- narzucenie standardów.

Dziękuję za uwagę