

Inżynieria oprogramowania 2018-19

ćwiczenia/pracowania

II FAZA WYTWARZANIA OPROGRAMOWANIA

- FAZA OPRACOWANIA -

ZADANIA DO WYKONANIA W LISTOPADZIE

I SPECYFIKACJA WYMAGAŃ

1. Dla aplikacji, której wytwarzanie opisuje zajmuje się grupa należy napisać 3 nietrywialne, REPREZENTATYWNE historyjki użytkownika.
Uwaga: historyjki użytkownika dotyczące jak np. logowania, wyszukiwania z listy, itp... wysyłania wiadomości SMS, MMS, wiadomości e-mail NIE SĄ REPREZENTATYWNE

2. Bardziej precyzyjne określenie wymagań

- a. Należy wymienić i krótko opisać wymagania funkcjonalne, rozumiane jako funkcje które aplikacja powinna posiadać. Funkcje te będą miały odpowiedniki w funkcjach menu systemu.

Dla aplikacji stosunkowo prostych należy sporządzić pełną (90% funkcji, przypadków Użycia, skrócie PU) specyfikację wymagań funkcjonalnych.

Dla aplikacji złożonych (obejmujących np. zarządzanie personelem, zarządzanie magazynem, analizy finansowe, itp..). Wymienione powinny być wszystkie główne podsystemy (pakiety PU), a dla 1, 2 podsystemów – takich, które można szczegółowo a nie ogólnikowo opisać należy sporządzić dokładną (90%) listę funkcji oraz krótki opis funkcji.

- b. Dla aplikacji należy wymienić wymagania нефункционалне. Trzeba określić podstawowe wymagania нефункционалне, zaproponować ich jednoznaczne definicje i przedstawić propozycje zweryfikowania tego czy dane wymaganie нефункционалне zostało spełnione.

Wykonanie powyższych zadań powinno być udokumentowane w odpowiednim dokumencie o nazwie Specyfikacja wymagań.

II KONCEPCJA WYKONANIA SYSTEMU

- 1.. Dla 3 historyjek użytkownika opisanych części I, demonstrujących reprezentatywnie działanie aplikacji należy napisać na poziomie logicznym (tym opisujących długości pól, typy danych, długości łańcuchów znaków ukazujących się na ekranie) dokładne (umożliwiające późniejsze testowanie) scenariusze dialogu człowieka z komputerem (tzw scenariusze przypadków użycia).

2. Dla powyższych reprezentatywnych 3 historyjek użytkownika (scenariuszy przypadków użycia) trzeba projektować na poziomie fizycznym dialogu z komputerem ukazujące się ekrany

(screeny, formatki ekranowe, przyciski itd) i jeżeli dany scenariusz przypadku użycia przewiduje drukowanie dokumentów np. biletów, faktur to trzeba te dokumenty też zaprojektować.

3.A. Sporządzenie modelu conceptualnego rzeczywistości, której dotyczy aplikacja. tzn identyfikacja encji i powiązań oraz warunków dotyczących powiązań.

3.B. Wymienienie oraz przedstawienie graficzne podstawowych elementów aplikacji oraz powiązań między nimi:

- sprzęt,
- oprogramowanie systemowe, bazy danych, narzędzia programistyczne, oprogramowanie do automatycznego testowania.
- struktury logicznej oprogramowania obiektowego
(podział kodu na główne komponenty np. klasy i powiązania między nimi)

3.C. Omówienie interfejsów aplikacji z otoczeniem.

4. Sporządzenie schematu bazy danych aplikacji tzn diagramu pokazującego tabele, ich klucze i podstawowe pola oraz powiązania między tabelami i warunki integralności bazy.

5. Przedstawienie głównych zasad kodowania
(oczywiście na podstawie dostępnych np. w sieci materiałów).

6. Bardziej dokładna identyfikacja ryzyka i opracowanie zasad zarządzania ryzykiem.

7. Ocena zgodności wykonanych prac z wizją przedstawioną w tablicy koncepcyjnej i specyfikacji wymagań.

Wykonanie powyższych tzn od 2 do 7 zadań powinno być udokumentowane w odpowiednim dokumencie o nazwie Koncepcja wykonania systemu.

HISTORYJKA UŻYTKOWNIKA

Historyjka użytkownika (ang. user story) – wysokopoziomowe wymaganie użytkownika lub wymaganie biznesowe, często używane w zwinnym wytwarzaniu oprogramowania.

W typowych sytuacjach historyjka użytkownika składa się z jednego lub kilku zdań w codziennym lub biznesowym języku i opisuje funkcjonalność potrzebną użytkownikowi, wymagania нефункционалне oraz kryteria akceptacji fragmentu oprogramowania odpowiedzialnego za realizację danej funkcjonalności.

Historyjki użytkownika nie powinny zawierać szczegółów technicznych, za to powinny być łatwe do napisania i w całości mieścić się na jednej kartce typu post-it.

Zadaniem analityka jest oprócz opisanie (też nagrania, rysunki, filmiki) historyjki użytkownika – znalezienie scenariuszy negatywnych oraz założeń do danej historyjki.

Wg Roya Jeffersa twórcy eXtreme Programming, historyjki użytkownika powinny składać się z 3 komponentów: karty opisującej funkcjonalność, konwersacji (ma być zaproszeniem do rozmowy o funkcjonalności i wymaganiach нефункционалных) oraz confirmacji (inaczej: potwierdzenia czyli testów, które je zweryfikują).

Cechy dobrej historyjki użytkownika opisuje **akronim INVEST (Independent, Negotiable, Valuable, Estimable, Small, Testable)** wymyślone przez Billa Wake'a.

Niezależne – poszczególne historyjki powinny być niezależne od siebie tak, aby można je było łatwiej zaplanować.

Negocjowalne – powinny móc podlegać różnego rodzaju zmianom i tworzyć płaszczyznę porozumienia między wszystkimi interesariuszami.

Wartościowe – powinny nieść jakąś wartość dla kogoś, niekoniecznie dla użytkownika – może być np. wartość dla testera.

Szacowalne – dla każdej historyjki powinna istnieć możliwość określenia z rozsądnym przybliżeniem czasu.

Małe – tak aby można określić potrzebne zadania oraz oszacować pracochłonność.

Testowalne – należy zweryfikować historyjki pod kątem niejasności, sprzeczności, dwuznaczności, niepewności.

Historyjki mogą mieć szablon, np.:

Jako <typ użytkownika>	dla kogo	jest ta historia
chce <jakiś cel>	co	on chce zrobić
żeby <jakiś powód>	dlaczego	chce to zrobić