

# Inżynieria oprogramowania 2017-18

## ćwiczenia/pracowania

### II FAZA WYTWARZANIA OPROGRAMOWANIA - FAZA OPRACOWANIA -

#### ZADANIA DO WYKONANIA W LISTOPADZIE

##### 1. Bardziej precyzyjne określenie wymagań

- a. Wymagania funkcjonalne ( funkcje, przypadki użycia ) – nie tylko nazwa ale również krótki opis

Dla systemów stosunkowo prostych należy sporządzić pełną ( 90% funkcji, przypadków użycia PU ) specyfikację wymagań.

Dla systemów 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. Należy określić podstawowe wymagania niefunkcjonalne, zaproponować ich jednoznaczne definicje i przedstawić propozycje zweryfikowania tego czy dane wymaganie niefunkcjonalne zostało spełnione.

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

- 2. Dla 3 funkcji (przypadków użycia) demonstrujących reprezentatywnie działanie systemu/program należy napisać na poziomie logicznym dokładne ( umożliwiające późniejsze testowanie) scenariusze dialogu człowieka z komputerem ( tzw scenariuszy przypadków użycia) .

Uwaga takie funkcje ( przypadki użycia ) jak np. logowanie, wyszukiwanie z listy, itp... wysyłanie wiadomości SMS, MMS, wiadomości e-mail NIE SĄ REPREZENTATYWNE

- 3. Dla powyższych reprezentatywnych 3 funkcji ( przypadków użycia) trzeba projektować na poziomie fizycznym dialogu z komputerem ukazujące się ekrany ( screeny, formatki ekranowe) i jeżeli w systemie drukowane są dokumenty np. bilety, faktury to trzeba te dokumenty też zaprojektować

- 3. Zaprojektowanie architektury:

Wymienienie oraz przedstawienie graficzne podstawowych elementów oraz powiązań między nimi:

- :
  - sprzęt,
  - oprogramowanie systemowe, bazy danych, narzędzia programistyczne, oprogramowanie do automatycznego testowania.

- struktura logiczna oprogramowania  
( podział kodu na główne komponenty np. klasy )

Omówienie interfejsów.

4. Sporządzenie schematu bazy danych systemu tzn diagramu pokazującego tabele, ich klucze i podstawowe pola oraz powiazania między tabelami.
5. Przedstawienie głównych zasad kodowania  
( oczywiście na podstawie dostępnych np. w sieci materiałów).
6. Bardziej dokładna identyfikacja ryzyk 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.**