Wymagania funkcjonalne, przypadki użycia

Inżynieria oprogramowania II

Problem i cel

- Tworzenie projektów bez konkretnego celu nie jest dobre
- Praktycznie każdy projekt informatyczny powstaje z uwagi na jakiś problem, który wymaga rozwiązania
- Ustalenie konkretnego celu przybliża nas do pomyślnego zakończenia projektu
- Mniejsza szansa, iż projekt zacznie się niebezpiecznie rozszerzać

Przykład

- Problem:

Zarządzanie papierowymi indeksami

- Propozycja rozwiązania:

Stworzenie systemu eProto

(opisane w materiałach)

Przebieg prac nad projektem

1. Określenie celu
2. Analiza i specyfikowanie wymagań
3. Wybranie fragmentu funkcjonalności
4. Implementacja
5. Testowanie
6. Odbiór, refleksja, wdrożenie
7. Powrót do punktu 3. (kolejna iteracja)
8. Utrzymywanie

Role w projekcie (1/5)

Klient

- Zleca i płaci za projekt
- Prezentuje swoje oczekiwania i cele związane z projektem
- Ma decydujący głos w sprawie kolejnych faz powstawania systemu
- Weryfikuje postęp prac i końcowy kształt systemu
- Klienci bywają różni

Role w projekcie (2/5)

Kierownik projektu

- Reprezentuje i zarządza zespołem
- Uzgadnia z klientem szczegóły projektu (m.in. harmonogram prac)
- Przydziela zadania członkom zespołu
- Musi być świadomy potrzeb i możliwości członków zespołu
- Stara się usuwać wszystkie przeszkody stojące na drodze zespołu

Role w projekcie (3/5)

Analityk

- „Tłumaczy” oczekiwania klienta na wymagania zrozumiałe dla zespołu (dyskusja, analiza, specyfikacja)
- Posiada pełną wiedzę na temat kontekstu i środowiska biznesowego systemu
- Wyjaśnia wszelkie wątpliwości i szczególne przypadki
- Kontroluje powstający system pod kątem oczekiwań klienta

Role w projekcie (4/5)

Architekt

- Projektuje system od strony technicznej
- Podejmuje decyzje technologiczne
- Przydziela członkom zespołu konkretne zadania związane z technologią
- Pomaga lub sam rozwiązuje wszelkie problemy techniczne

Role w projekcie (5/5)

Programiści, testerzy i inni

- Siła robocza
- Zwykle nie mają bezpośredniego kontaktu z klientem – „klientem” jest analityk

Zasada dziewięciu kroków

1. Określenie problemu
2. Określenie procesu biznesowego
3. Identyfikacja aktorów (diagram kontekstu)
4. Identyfikacja przypadków użycia (diagram przypadków użycia)
5. Identyfikacja obiektów biznesowych
6. Zbadanie kompletności przypadków użycia (tablica CRUD)
7. Opisanie scenariuszy głównych
8. Identyfikacja zdarzeń
9. Opisanie scenariuszy rozszerzeń

Aktor

- Rola w procesie biznesowym
- W uproszczeniu: grupa użytkowników
- Posiada dostęp do specyficznej funkcjonalności
- Ma mniejsze lub większe uprawnienia
- Istnieją role „specjalne” (Administrator)

Proces biznesowy

- Proces biznesowy jest listą kroków, które muszą być wykonywane, aby osiągnąć zamierzony wcześniej cel biznesowy
- Interakcja aktorów między sobą, przepływ informacji
- Ogólny pogląd na wymagania
- Na tym etapie stopniowo uszczegóławiamy naszą wiedzę o dziedzinie problemu

Obiekt biznesowy

- Obiekt zarządzany przez system, niosący pewne dane
- Na obiekcie najczęściej są zdefiniowane operacje CRUDL (Create, Retrieve, Update, Delete, List)
- Nie należy mylić aktora z obiektem!

Wymaganie funkcjonalne

- Pojedyncza funkcjonalność (funkcja) wymagana od systemu
- Opisanie „co” robi system
- Pochodzą z opowieści użytkownika lub procesów biznesowych
- Wymagania są skojarzone z aktorami
- Specyfikacja wymagań ułatwia implementację, ale też oszacowanie pracochłonności

Nadawanie priorytetów

- H – High, wysoki priorytet (wymagania obowiązkowe)
- M – Medium, średni priorytet (ważne wymagania)
- L – Low, niski priorytet (wymagania opcjonalne)

Przypadek użycia (ang. *use case*)

- Sposób opisu wymagań funkcjonalnych
- Lista kroków, które muszą być wykonane w celu „ukończenia” danego wymagania
- Forma przyjazna – opisuje kolejno co należy zrobić, jest zrozumiała i łatwa w przygotowaniu
- Forma uniwersalna – dla klienta, programisty, testera

Przykład przypadku użycia

UC01: Wyświetlenie ocen

Aktorzy: Student

Scenariusz Główny

1. Student wybiera opcję wyświetlenia semestrów.
2. System wyświetla listę dotychczasowych semestrów.
3. Student wybiera jeden z dostępnych semestrów.
4. System wyświetla listę przedmiotów wraz z ocenami.

Rozszerzenia

3.A Student chce wrócić do menu głównego.

3.A.1 Student wybiera opcję powrotu.

3.A.2 System wyświetla menu główne.

3.A.3 Koniec przypadku użycia.

Ważne informacje

- Przedstawiony przypadek użycia jest napisany na tzw. poziomie użytkownika
- Opisuje interakcję użytkownika z systemem – akcję i reakcję
- System jest domyślnym aktorem, nie trzeba go dodatkowo podawać

Poziomy przypadków użycia

- Biznesowy – najbardziej ogólny, opisuje interakcję różnych organizacji, nie rozgrywa się „natychmiast”
- Użytkownika – najbardziej typowy, interakcja pomiędzy użytkownikiem a systemem, bez szczegółów technicznych
- Podfunkcji – szczegóły techniczne, dotyczy detali interfejsu, dla lepszego zrozumienia wymagania

Dobra praktyka nr 1

- Zawsze wypisz aktora, który wykonuje daną akcję - dla każdego kroku musi być on zdefiniowany.

Dobry przykład:

3. Wykładowca wybiera opcję edycji danego protokołu.

Zły przykład:

3. Dla danego protokołu wybierana jest edycja.

Dobra praktyka nr 2

- Przypadek użycia powinien być odpowiedniej długości. Scenariusze poniżej 3 kroków wydają się bezużyteczne i mogą być „wchłonięte" przez inne, natomiast powyżej 9 kroków są ciężkie w odczytaniu, zrozumieniu i najprawdopodobniej dotyczą zbyt wielu rzeczy.

Dobra praktyka nr 3 (1/2)

- Należy unikać „technikaliów", w tym określeń typowych dla interfejsu użytkownika (jak „klika", „wpisuje“ - zamiast tego piszemy „wybiera", „podaje") . Ważna jest funkcjonalność, a nie sposób jej realizacji - ten aspekt pokrywają prototypy i trochę wymagania pozafunkcjonalne. Jednakże, na poziomie podfunkcji szczegóły techniczne są dopuszczalne.

Dobra praktyka nr 3 (2/2)

Dobry przykład:

3. Student wybiera jeden z dostępnych semestrów.

Zły przykład:

3. Student klika na przycisk związany z danym semestrem.

Dobra praktyka nr 4 (1/2)

- Główny scenariusz to lista typowych, nieprzerwanych kroków, które kończą się optymistycznie. Na wszelkie konstrukcje warunkowe i możliwe anomalie miejsce jest w rozszerzeniach

Dobra praktyka nr 4 (2/2)

Dobry przykład:

7. System wyświetla informację o wprowadzonych zmianach.

…

Rozszerzenia:

7.A System nie może zapisać wprowadzonych zmian.

7.A.1 System wyświetla informację o błędzie.

7.A.2 Powrót do punktu 4.

Zły przykład:

7. Jeśli informacje są poprawne, system wyświetla....

Dobra praktyka nr 5

- Należy stosować proste zdania i nie bać się powtórzeń - przypadek użycia nie jest wierszem z okresu romantyzmu do analizy na języku polskim, lecz ma precyzyjnie „opowiadać" o danym wymaganiu i być bezproblemowo zrozumiany przez czytającego (a przede wszystkim realizującego).

Dobry przykład:

5. Wykładowca zmienia żądane dane.

Zły przykład:

5. Wykładowca zmienia dane, które wymagają edycji.

Dobra praktyka nr 6

- Nazwa przypadku użycia powinna odzwierciedlać cel, który chce osiągnąć aktor. Przykładowo, prawidłowa nazwa wymagania to „usunięcie studenta", a nie „dezaktywacja rekordu studenta", nawet jeżeli technicznie tak to wygląda

Dobry przykład:

UC06: Wstawienie oceny

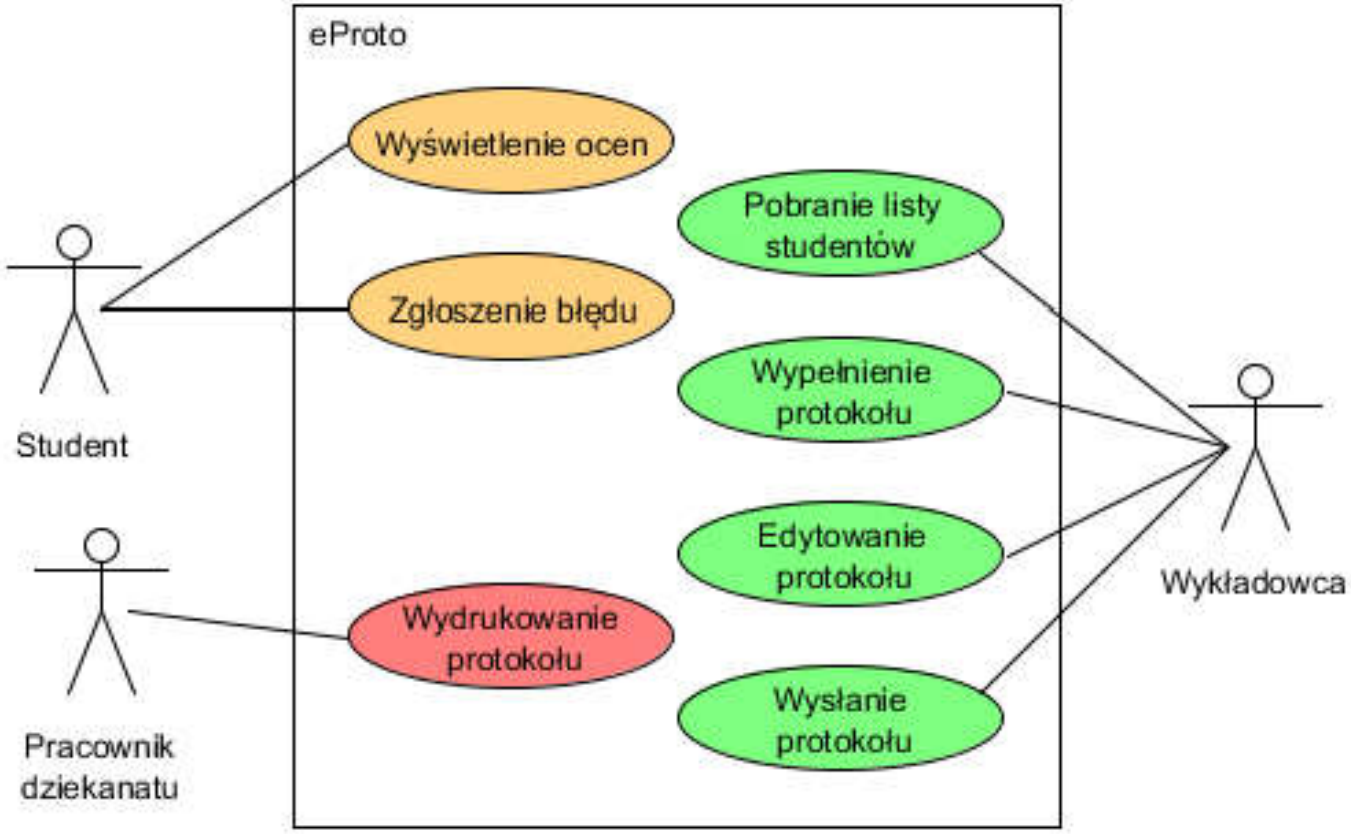
Zły przykład:

UC06: Dodanie obiektu oceny do przedmiotu studenta

Przypadki użycia – dodatki

- Diagramy przypadków użycia
- Zawieranie (*includes*)
- Rozszerzanie (*extends*)
- Specjalizacja

Diagram przypadków użycia



A teraz…

…praca w grupach!