

Modele wytwarzania oprogramowania

Omówienie kamienia milowego 2 – etap 1

Zajęcia 30.04.2025

Wprowadzenie

- Przeglazano wszystkie projekty:
 - Niniejsza prezentacja zawiera wybrane typowe błędy i wskazówki
- Agenda:
 - Wprowadzenie
 - Omówienie wybranych typowych błędów etapu 1
 - Wskazówki do kolejnych prac (bieżąca dokumentacja do dostarczenia w sekcji *Files* – **zakres kolejnego kroku**: uzupełnienie scenariuszy przypadków użycia o diagramy UML poprawiające czytelność bardziej złożonych scenariuszy)

Typowe błędy: brak struktury dokumentu, brak tekstu

- Wymagania:
 - Dokumentacja techniczna ma mieć formę **wyvodu**
 - Struktura dokumentu:
 - Strona tytułowa: tytuł projektu, skład zespołu, dane kontaktowe
 - Spis treści
 - Wprowadzenie: opis i cel projektu
 - Analiza wymagań:
 - Wymagania funkcjonalne: na bazie UML – diagramy przypadków użycia, scenariusz przypadków użycia w formie tabelarycznej (nazwa przypadku, opis przypadku, aktorzy biorący udział, uzasadnienie, warunki początkowe, warunki końcowe, główne ciąg zdarzeń, alternatywne ciągi zdarzeń)
 - Dodatkowy rozdział: wkład w projekt każdego członka zespołu (ocena za projekt jest indywidualizowana i składają się na nią: ocena za projekt jako całość wspólna dla wszystkich członków zespołu oraz ocena za indywidualny wkład w projekt)
 - **Tekst: każdy diagram w tekście powinien być omówiony, tekst ma też pełnić rolę „spoiwa” pomiędzy diagramami: projekt to nie jest zbiór diagramów. Dużo projektów ma za mało tekstu (pojedyncze nie mają go wcale)**
- Na plus: większość projektów co do struktury jest okay

Typowe błędy: brak struktury dokumentu,
brak tekstu c.d.

- Analiza wymagań funkcjonalnych -> diagramy use case są częścią analizy
- Błąd: umieszczanie rozdziału „Diagramy przypadków użycia” na tym samym poziomie co „Analiza wymagań funkcjonalnych”

Typowe błędy: Use Case

- Źle zidentyfikowani aktorzy
- Za dużo związków include/extend
- Warunki początkowe (np. Użytkownik jest zalogowany) modelowane jako związek extend z use case Logowanie.

Typowe błędy: Niekompletne lub scenariusze

- Brak warunków początkowych/końcowych
- Przykładowy krok głównego ciągu zdarzeń: „**Użytkownik uruchamia aplikację i wybiera opcję “Zarejestruj się” -> poza systemem**
- „System wyświetla formularz rejestracyjny, w którym użytkownik wprowadza swoje dane.” lub „Użytkownik wypełnia formularz reklamacyjny” -> precyzja: **jakie dane należy podać?**
- „Użytkownik akceptuje regulamin aplikacji, a system weryfikuje poprawność wprowadzonych danych i zapisuje je w bazie danych.” lub „System sprawdza poprawność tagów i obecność opisu, Jeżeli przesłane dane są niepoprawne, system zwraca informację o niepowodzeniu i uruchamia ciąg zdarzeń «Niepoprawne dane grafiki»,” -> **na czym polega walidacja?**
- „3. Pracownik administracji wprowadza dane o nowym statku i zatwierdza. 4. Statek o podanych parametrach znajduje się już w systemie, wyświetlony zostaje komunikat.” -> **czy to jedyna walidacja?**

Typowe błędy: Niekompletne scenariusze lub brak precyzji

- Przykład: „użytkownik wpisuje dane”
- Problemy:
 - Jakie dane, jak wygląda ich walidacja
 - Co się dzieje gdy wprowadzone dane są niepoprawne
 - Jakie są komunikaty błędów

Typowe błędy: Niekompletne lub scenariusze-przykłady

1. System **prosi** o wprowadzenie nazwy kursu. (w jaki sposób system „prosi”?)
2. Twórca wprowadza nazwę (czy tylko nazwę, czy są jakieś ograniczenia co do nazwy etc.)
3. Twórca kończy tworzenie kursu (na czym to polega?).
4. System zapisuje nowy kurs do bazy.

Use Case: „Użytkownik przegląda ogłoszenia”

Główny ciąg zdarzeń: Użytkownik wchodzi na stronę, wybiera opcję "Przeglądaj ogłoszenia", wybiera kryteria wyszukiwania, takie jak lokalizacja, termin dostawy czy rodzaj ładunku, a następnie przegląda dostępne ogłoszenia.

Typowe błędy: Niekompletne lub scenariusze – brak zrozumienia czym jest scenariusz

- Często powtarza się w scenariuszu to czym jest use case a nie rozpisuje się tego jak dokładnie dana funkcjonalność ma działać
- Kolejny przykład błędnego scenariusza (Wyszukiwanie):
 1. System oferuje wewnętrzną wyszukiwarkę po produktach dostępną u góry strony (to nie jest krok scenariusza - to jest element opisu use case, jego uzasadnienie etc.)
 2. Zainteresowany klient wpisuje wybraną przez siebie frazę (to jest krok scenariusza)
 3. Dzięki temu, że każdy zarejestrowany produkt w naszej bazie ma tagi, system sprawdza dopasowanie wpisanej frazy z tagami produktów i wyświetla w pierwszej kolejności te, które są najbardziej zgodne (częściowo dobre, brak precyzji: jak działa wyszukiwanie po tagach? Czy da się w oparciu o taki opis dokonać implementacji?)

Typowe błędy 2: przykład scenariusza

- Precyzja:
 - Czy warunki wstępne jasne?
 - Czy warunek końcowy jasny?
 - W jaki sposób użytkownik „prosi”?
 - Co to są „te wiadomości”
 - Czy potrafimy dokonać implementacji?
- Czy Aktor dobrze określony?
Może: Użytkownik, warunki początkowe: użytkownik jest zalogowany
ew. „użytkownik zarejestrowany”
- Czy zdefiniowanie aktora w ten sposób: „Użytkownik zalogowany loginem pskrzynski” jest poprawne?

Sekcja	Treść
Nazwa	Wyświetl statystyki
Aktorzy	Użytkownik zalogowany
Warunki wstępne	Uruchomiona aplikacja, dostęp do bazy produktów
Warunki końcowe	Wyświetlenie wykresów naszych statystyk
Rezultat	Wyświetlanie wykresów
Scenariusz główny	1. Użytkownik prosi o wyświetlenie statystyka 2. Program odczytuje statystyki z najczęściej używanych produktów i informacje o kaloriach 3. Przekonwertowanie tych wiadomości w wykresy i rankingi 4. Wyświetlenie
Scenariusz alternatywny	Gdy nasi znajomi też używają aplikacji 1. Porównanie wykresów z naszymi znajomymi
Scenariusz wyjątku	Błąd załadowania, albo konwersji danych 1. Wyświetlenie komunikatu o zrestartowanie aplikacji

Typowe błędy: odizolowane ciągi zdarzeń

- Często pełna ścieżka scenariusza dla sukcesu obejmuje przejście przez wiele rozgałęzień
- Różne ciągi zdarzeń mogą być zatem powiązane: alternatywny scenariusz może pojawić się jako element na ścieżce scenariusza sukcesu

Typowe błędy: brak powiązania dodatkowych diagramów z use case i jego scenariuszem

- Źródło problemu: brak tekstu, brak wywodu, błędna struktura dokumentu lub brak struktury dokumentu
- Wywód: Wymagania funkcjonalne -> use case -> scenariusz use case -> tabela + diagramy wspomagające

Typowe błędy: czytelność diagramów

- Aby diagram był czytelny na jednej stronie można umieścić 7 do 9 use casów – umieszczanie większej liczby powoduje utratę czytelności
- Brak granic systemu: wpływa na czytelność

Typowe błędy: „diagramy dla diagramów”

- Diagramy zastępują tekst i pomagają w lepszym zrozumieniu projektu – nie tworzymy diagramów dla siebie – przykłady:
 - rozdział „Pozostałe diagramy”
 - rozdział „Diagramy przypadków użycia”
- Celem projektu jest stworzenie dokumentacji technicznej systemu umożliwiającej jego dalszą implementację. Wykorzystanie diagramów UML ułatwia realizację tego celu. Celem naszych projektów NIE JEST rysowanie diagramów.
- Spis treści nie powinien przypominać podręcznika do UML (ex. rozdział „Diagramy przypadków użycia”)
- Diagramy pełnią określone funkcje i tak na tym etapie używamy ich:
 - Diagramy Use Case: do opisu funkcjonalności systemu oraz użytkowników systemu (z lotu ptaka)
 - Diagramy aktywności: pomocne w otoczeniu use case do opisu scenariuszy przypadków użycia (uwaga: bardziej złożonych)
 - Diagramy sekwencji: do opisu wymiany komunikatów w czasie, pomocne w m. in. w otoczeniu use casów (ale nie tylko)
 - Diagramy kolaboracji: j.w. ale inaczej rozłożone akcenty
 - Diagramy stanów: pomocne przy opisie działania systemu lub jego elementów jako całości, czasami w otoczeniu specyficznych use casów

Typowe błędy: błędy językowe

- Wstęp: „...aplikacja **zaprojektowana** przez nasz zespół...”
- We wstępie stawiamy problem i opisujemy co chcemy zrobić i dlaczego
- Dalsza dokumentacja to wywód, w którym najpierw dokładnie specyfikujemy ten problem (część „co” z analizą wymagań) a potem go rozwiązujemy (część „jak”).
- Na tym etapie aplikacja nie jest zaprojektowana: problem dopiero ma zostać postawiony
- Inny przykład, wstęp:
 - JEST: „Jest to platforma umożliwiająca...”
 - POWINNO BYĆ: „Celem projektu jest stworzenie platformy umożliwiającej...”

Wskazówki: diagramy stanów VS diagramy aktywności

- Maszyny stanowe, przejścia między stanami
- Czynności, opis przepływu od czynności do czynności
- Diagram aktywności: pomocny w otoczeniu przypadków użycia (opisu scenariusz)
- Diagram stanów: pomocny przy opisie systemu/podsystemu jako całości, na etapie projektowym, rzadziej w otoczeniu przypadku użycia

Wskazówki: wykorzystanie UML

- Na etapie analizy możemy wykorzystać nie tylko diagramy Use Case ale również **inne elementy UML?**
- Czego można użyć w otoczeniu Use Case: diagramy sekwencji, kolaboracji, czynności, stanów...
- Maszyny stanowe, przejścia między stanami -> diagramy stanów
- Czynności, opis przepływu od czynności do czynności -> diagramy aktywności (inaczej czynności): pomocny w otoczeniu przypadków użycia (opis scenariusza)
- Diagram stanów: pomocny przy opisie systemu/podsystemu jako całości, na etapie projektowym, rzadziej w otoczeniu przypadku użycia

Zajęcia 16.05- wskazówki

- Kolejna wersja do dostania do 16.05 (zamknięcie etapu 1)
- Wstępny projekt systemu obejmujący (na 30.05):
 - Architekturę systemu
 - Technologiach realizacji systemu (wymagania niefunkcjonalne)
 - Identyfikację modułów i pakietów
 - Identyfikację podstawowych klas
 - Komunikacja między modułami
 - Identyfikację podstawowych kolaboracji między klasami
- Forma zajęć 16.05
 - Prezentacja w dniu 16.05 na bazie dostarczonych artefaktów (przypomnienie + uzupełnienie dzisiejszej)
 - Ew. Pytania i dyskusja (15-20 min.)
 - Lektura i analiza dokumentacji dostarczonych 16.05 (PS)
 - Konsultacje indywidualne w grupach (**uwaga:** zgłoszenia potrzeby konsultacji do środy 14.05 do godz. 12)