Inżynieria oprogramowania

Definicja IEEE:
inżynieria oprogramowania jest to zastosowanie
- systematycznego,
- zdyscyplinowanego,
- ilościowego
podejścia do
- rozwoju,
- eksploatacji
- utrzymania
oprogramowania.

Inżynieria oprogramowania

Inżynieria oprogramowania jest jednym z czternastu obszarów informatyki wyodrębnionych w dokumencie Computing Curricula 2001 stworzonym wspólnie przez ACM oraz IEEE (inne to np. sztuczna inteligencja, bazy danych, systemy operacyjne, grafika komputerowa).

Obligatoryjne jednostki wiedzy

- Wymagania
- Projektowanie
- Walidacja
- Ewolucja
- Procesy
- Zarządzanie
- Narzędzia
- API

Opcjonalne jednostki wiedzy

- Metody formalne
- Systemy specjalne
- Komponenty
- Niezawodność

Specyfikacja wymagań

- Powinna być to najważniejsza część kontraktu pomiędzy klientem a dostawcą (firmą informatyczną)
- Wymaganie = opis co system powinien robić
- Specyfikacja = zbiór wymagań

Problemy

- W założeniu po spisaniu wymagań klient dostanie to, czego potrzebuje.
- Problemy lingwistyczne: raport miesięczny, SAD (dokument celny)
- Wiedza świadoma i nieświadoma
  (np. klawiatura musi być po tej samej stronie telefonu co ekran)
- Spisywanie wymagań to sztuka

Podział wymagań

- Funkcjonalne (użytkownika):
  - Wprowadzanie nowej faktury
  - Generowanie raportu miesięcznego
- Pozafunkcjonalne (bezpieczeństwo, wydajność, niezawodność)
  - minimum 20 faktur na godzinę
  - 200000h MTBF
  - maksimum 2 godziny potrzebne na przeszkolenie 1 pracownika

Podejścia do tworzenia specyfikacji

- System powinien...
- Funkcje systemu
- Przypadki użycia

System powinien...

- Wystawiać faktury, generować miesięczne zestawienie faktur, faktura powinna ... itp.
- IEEE 830-1998
- Łatwe spisywanie
- Słaba czytelność
- Trudne sprawdzanie spójności, kompletności i poprawności

Funkcje systemu

- Analogicznie do funkcji matematycznych, mamy:
  - Wejście: pozycje faktury
  - Wyjście: wysłanie faktury
  - Efekty uboczne: zapis faktury w rejestrze
- Słaba czytelność (rozbicie na wiele małych funkcji)
- Trudne do zrozumienia

Przypadki użycia

- Łatwość spisywania
- Czytelność
- Łatwość zrozumienia, podział hierarchiczny

Przypadki użycia

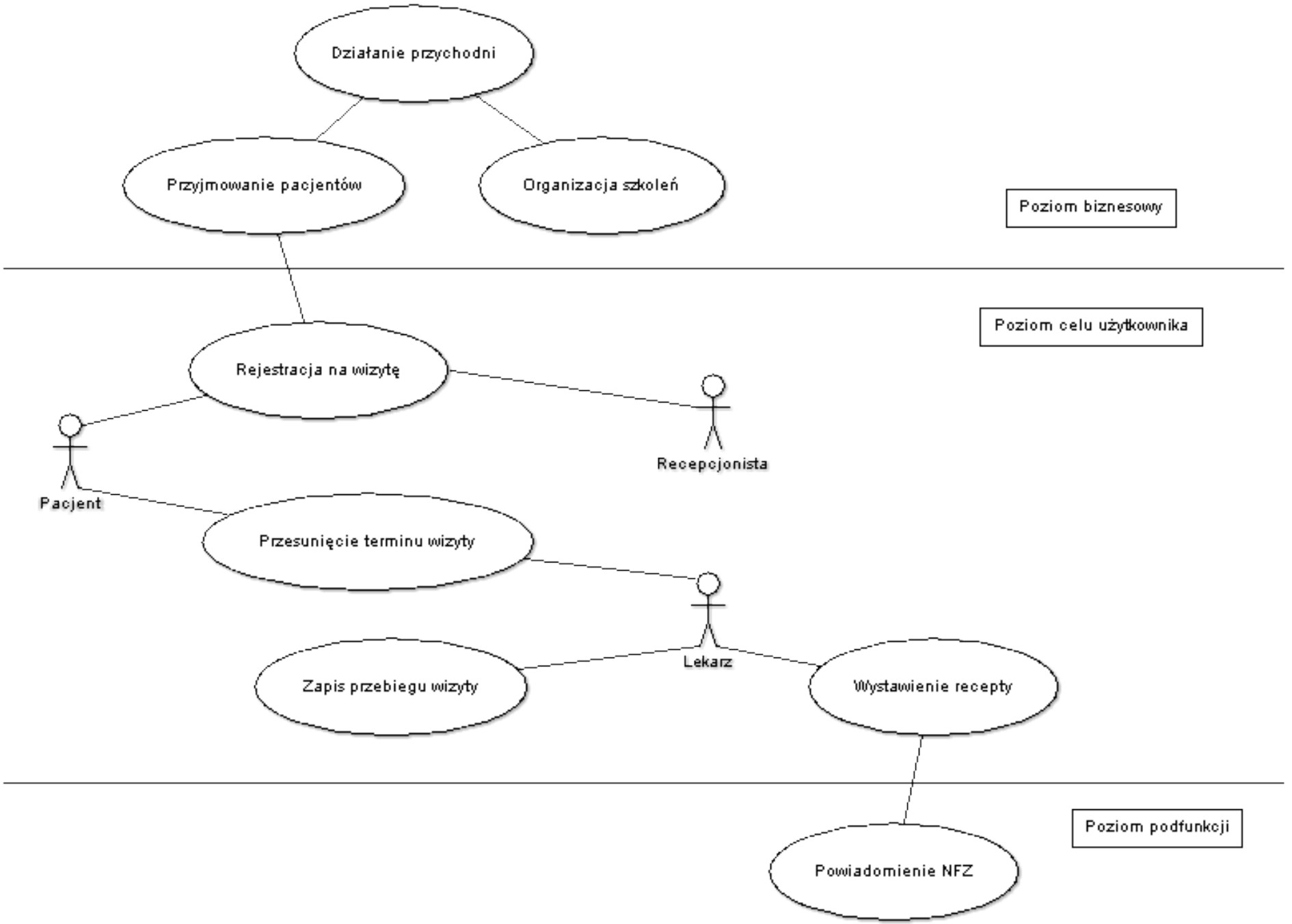
- Nazwa: Wystawianie faktury – fraza czasownikowa
- Identyfikator: UC1
- Główny scenariusz:
  - 1. Sprzedawca pragnie wystawić fakturę.
  - 2. Sprzedawca wpisuje pozycje faktury.
  - 3. System podlicza fakturę, nadaje jej nowy numer i zapisuje w rejestrze
  - 4. Sprzedawca drukuje fakturę.

Przypadki użycia

- **Rozszerzenia:**
  - 3.A. Sprzedawca nie dodał żadnej pozycji
  - 3.A.1. System prosi o ponowne wprowadzenie pozycji (powrót do 2.)
- Atrybuty:
  - **Główny aktor:** Użytkownik
  - **Priorytet**: Wysoki
  - **Źródło**: Jan Kowalski

Diagram przypadków użycia

- Nazwy przypadków użycia
- Powiązania z aktorami
- Dobre jako „**mapa**”


Działanie przychodni
Przyjmowanie pacjentów
Organizacja szkoleń
Poziom biznesowy
Poziom celu użytkownika
Rejestracja na wizytę
Recepcjonista
Pacjent
Przesunięcie terminu wizyty
Lekarz
Zapis przebiegu wizyty
Wystawienie recepty
Poziom podfunkcji
Powiadomienie NFZ

Fraza czasownikowa w nazwie

- Kilka przykładów złych nazw:
  - Nazwy nic nie znaczące: Główny przypadek użycia, Przypadek użycia 2
  - Zbyt ogólna, przez co nie stanowi żadnej wartości dla czytelnika: Zarządzanie

Scenariusz i rozszerzenia

Generalną zasadą jest zwięzłość i przejrzystość, aby czytelnik nie pogubił się.
Dlatego, każdy scenariusz powinien zawierać od 3-9 kroków (gdy jest ich mniej, specyfikacja wymagań jest zbyt fragmentaryczna, natomiast gdy jest ich więcej - czytelnik nie jest w stanie ogarnąć pamięcią całości).
Ze względu na czytelność, również poszczególne kroki scenariusza nie powinny być zbyt skomplikowane. Najlepiej gdy są wyrażone prostym zdaniem zawierającym podmiot (czyli aktora). Rozszerzenia natomiast, to sekwencje kroków wykonywane podczas sytuacji wyjątkowych. Numer rozszerzenia składa się zawsze z numeru kroku, którego rozszerzenie dotyczy, oraz litery – stanowiącej numer rozszerzenia (w ramach tego kroku). Czyli przykładowo możemy się spotkać z takimi rozszerzeniami:
1.A. 2.A. 3.A. 1.B.

Obojętność technologiczna

- Błędem jest, gdy przypadki użycia zawierają szczegóły technologiczne:
  - technologia jest zmienna - może sie okazać, że następny podobny projekt będzie realizowany w nowej technologii, mimo że część wymagań będzie identyczna. Jeżeli przypadki użycia będą zawierać szczegóły technologiczne, to ponowne zastosowanie raz napisanych wymagań będzie dużo trudniejsze
  - szczegóły Graficznego Interfejsu Użytkownika (ang. GUI - Graphical User Interface) zaciemniają jedynie obraz czytelnika - gubi się on w gąszczu szczegółów. Czasem jednak jest potrzeba zobrazowania klientowi takich szczegółów - wtedy warto naszkicować poszczególne ekrany aplikacji i dodać je jako załącznik do specyfikacji wymagań
  - klient nie rozumie terminów technicznych w stylu SOAP, baza danych, HTTP, itp.

Szerokość przed głębokością

Kolejność:
1. Lista aktorów
2. Nazwy przypadków użycia
3. Główny scenariusz
4. Rozszerzenia

Mały zespół autorów

- wielkość zespołu to **najważniejszy czynnik wpływający na jakość**
- **2-3** osoby w zupełności **wystarczają**
- zaangażuj **więcej osób w proces** recenzji
- duże systemy – **kilka małych zespołów** z **jednym architektem** odpowiedzialnym za spójną wizję systemu

Zrównoważony zespół

- grupa podobnych specjalistów skupi się jedynie na **ograniczonych problemach**
- synergia: kompensuj **słabe** strony jednych, **dobrymi** stronami innych
- połącz ludzi **różnej specjalności**
- analitycy i **użytkownicy**

UC1: Faktura

Główny scenariusz:

1. Sprzedawca wpisuje kod dostępu.
2. System weryfikuje użytkownika.
3. Kliknięcie na przycisk wystawiania faktury.
4. System prezentuje formularz.
5. Wpisanie pozycji w dolnym okienku.
6. Wpisanie wartości pozycji, stawki VAT, liczby pozycji i nr. porządkowego.
7. System podlicza fakturę i prezentuje sumę.
8. System nadaje nowy numer i zapisuje w rejestrze faktur.
9. Wydruk faktury.
10. Jeżeli wystawianie faktur zakończyło się, to użytkownik się wylogowuje.

Rozszerzenia:

3.A. Sprzedawca nie dodał żadnej pozycji

3.A.1. System prosi o ponowne wprowadzenie pozycji (powrót do 2.)

Najważniejsze błędy

- Tytuł przypadku użycia nie mówi, czym przypadek użycia będzie się zajmował (nie wiemy czy to jest wystawienie faktury, wysłanie, przefaksowanie, odbiór, czy cokolwiek innego)
- W krokach 3, 5, 6, 7 znajdują się zbędne szczegóły (część to szczegóły techniczne, np. informacje o GUI, wyliczenia pól na ekranie, itp.)
- W krokach 3, 5, 6, 9 - nie ma sprecyzowanego podmiotu, czyli aktora, który wykonuje krok.
- Krok 10 zawiera konstrukcję warunkową - warunki powinny się znaleźć w sekcji rozszerzeń, a nie w głównym scenariuszu.
- Scenariusz posiada 10 kroków, czyli jest za długi dla przeciętnego czytelnika. Liczba kroków powinna oscylować pomiędzy 3-9.

Zbieranie wymagań

- Oceń możliwość zbudowania systemu
- Zapisuj źródła wymagań
- Zdefiniuj środowisko operacyjne

Analiza i negocjacja wymagań

- Określ granice systemu
- Korzystaj z list kontrolnych podczas analizy wymagań
- Określ priorytety wymagań

Walidacja wymagań

- Sprawdź, czy wymagania spełniają standardy
- Zorganizuj formalne inspekcje wymagań
- Zdefiniuj listy kontrolne walidacji
- Wykorzystaj zespoły wielodyscyplinarne do przeglądów wymagań