Analiza i projektowanie systemów informatycznych

Definiowanie i analiza wymagań

Rodzaje wymagań

- wymagania funkcjonalne
  - określenie funkcjonalności systemu
  - określenie sposobu użycia funkcjonalności przez byty zewnętrzne (użytkownicy, inne systemy)
- wymagania niefunkcjonalne
  - ilościowe
  - jakościowe

Wymagania funkcjonalne

- perspektywa funkcjonalna
  - widok od wewnątrz systemu – określa zestaw funkcji, które system może wykonywać
  - lista funkcji
  - grupy funkcji
  - hierarchia funkcji
- perspektywa użycia
  - widok od zewnątrz systemu – określa sposoby użycia funkcjonalności systemu w realnych przypadkach
  - przypadki użycia
  - aktorzy

Przypadek użycia (Use Case)

- Definicja:
  - Przypadek użycia jest to dokument opisowy objaśniający sekwencje zdarzeń występujących w ramach procesu używania systemu przez „aktora”, czyli obiekt zewnętrzny.
- Własności:
  - Przypadki użycia same nie stanowią specyfikacji wymagań, ale są doskonałym narzędziem do ich ilustracji i weryfikacji.
    - korzystają ze zdefiniowanych funkcji systemu,
    - odnoszą się do innych wymagań (funkcjonalnych i niefunkcjonalnych)
  - Przypadek użycia jest zawsze inicjowany przez aktora
    - nie służy do opisu wewnętrznych działań systemu, ani też działań automatycznych

Przypadek użycia (Use Case)

- Własności:
  - Przypadek użycia jest zawsze opisem „dużego”, kompletnego procesu, zwierającego zwykle wiele kroków lub transakcji, produkującego wynik odbierany przez aktora.
    - Opisuje całość interakcji pomiędzy użytkownikiem, a systemem przy realizacji przez użytkownika pełnego, zamkniętego zadania.
    - Nie jest opisem pojedynczego kroku, wywołania funkcji, itp.

Przypadki użycia – forma opisu

- Opis ogólny:
  - nazwa przypadku użycia,
  - aktorzy,
  - typ,
  - opis,
  - referencje (lista funkcji, wymagań, etc.)
- Przebieg zdarzeń:
  - typowy,
  - alternatywne,
  - wyjątki
  - opis tekstowy, tabelaryczny,
  - diagramy aktywności

Przebieg zdarzeń, opis tabelaryczny

1. inicjacja
2. pierwsza akcja aktora
3. odpowiedź systemu
4. kolejna akcja aktora
5. kolejna odpowiedź systemu
6. kolejna akcja aktora
7. kolejna odpowiedź systemu

...

...

Przypadki użycia - podział

- poziom opisu:
  - wysokiego poziomu *(high level)* – tylko opis ogólny
    - tworzone podczas identyfikacji przypadków użycia, na początku zbierania wymagań
  - rozszerzone *(expanded)* – opis ogólny wraz z przebiegiem zdarzeń
    - rozszerzanie opisu podczas dalszego zbierania wymagań szczegółowych i w trakcie analizy
- waga:
  - główne *(primary)* – najważniejsze procesy systemu
    - muszą być zrealizowane
    - wśród nich wyróżniamy przypadki architektonicznie znaczące
  - drugorzędne *(secondary)* – procesy mniej istotne lub rzadko wykonywane
    - muszą być zrealizowane, ale w końcowych iteracjach
  - opcjonalne *(optional)* – procesy poboczne, dodatkowe
    - nie muszą być koniecznie realizowane

Przypadki użycia - podział

- wnikliwość opisu:
  - merytoryczne, istotne *(essential)* – koncentrujące się na istocie problemu, opisane w formie wolnej od szczegółów technicznych i implementacyjnych
    - tworzone przez analityków, aby nie ograniczać projektantów w doborze technologii
  - rzeczywiste, konkretne *(real)* – opisane przy wykorzystaniu terminologii technologicznej i zawierające szczegóły implementacyjne (np. mechanizmy realizacji wejścia/wyjścia)
    - tworzone przez projektantów – dobór odpowiedniej technologii i sposobu realizacji przy uwzględnieniu wszystkich wymagań niefunkcjonalnych

Aktor *(Actor)*

- Definicja:
  - Aktor jest bytem zewnętrznym w stosunku do systemu, wchodzącym w interakcje z systemem: wysyłającym i odbierającym komunikaty, wymieniającym informacje.
- Własności:
  - Aktor reprezentuje rolę grana w danym przypadku użycia:
    - rola człowieka,
    - system komputerowy,
    - urządzenie elektroniczne.
  - Aktor jest klasą, nie instancją
    - Możliwe jest klasyfikowanie aktorów poprzez mechanizm generalizacji-specjalizacji.

Aktorzy – podział

- ranking aktorów:
  - główny *(primary)* – używający zasadniczych funkcji systemu, biorący udział w głównych przypadkach użycia,
  - drugorzędny *(secondary)* – wykorzystujący fukcje pobocze lub administracyjne.
- aktywność:
  - aktywny *(active)* – inicjujący przypadek użycia,
  - pasywny *(passive)* – uczestnik scenariusza odpowiadający na sygnały.

Identyfikacja przypadków użycia

- poprzez aktorów:
  - identyfikacja aktorów,
  - dla każdego aktora – identyfikacja procesów i funkcji, które inicjują lub w których biorą udział (odczyt, tworzenie, aktualizacja, usuwanie danych, informowane o zdarzeniach).
- poprzez zdarzenia:
  - identyfikacja zdarzeń zewnętrznych, na które system musi odpowiadać,
  - powiązanie zdarzeń z aktorami i procesami.
- poprzez funkcje:
  - identyfikacja funkcji, które system ma wykonywać,
  - określenie kontekstów (procesów), w których każda funkcja jest wykonywana.

Związki pomiędzy przypadkami użycia

- uszczegółowienie – generalizacja
  - jeden przypadek użycia stanowi częściowy opis zachowania innego przypadku użycia
  - odpowiednik dziedziczenia w programowaniu
- włączenie <<include>>
  - jeden przypadek użycia (lub wiele) wykorzystuje zawsze i w całości inny przypadek użycia
  - odpowiednik mechanizmu „include” w programowaniu
- rozszerzenie <<extends>>
  - jeden przypadek użycia (lub wiele) wykorzystuje w pewnych sytuacjach inny przypadek użycia
  - odpowiednik wywołania podprogramu w programowaniu
  - jeśli przypadek użycia nigdy nie jest wykorzystywany samodzielnie, nazywany jest abstrakcyjnym

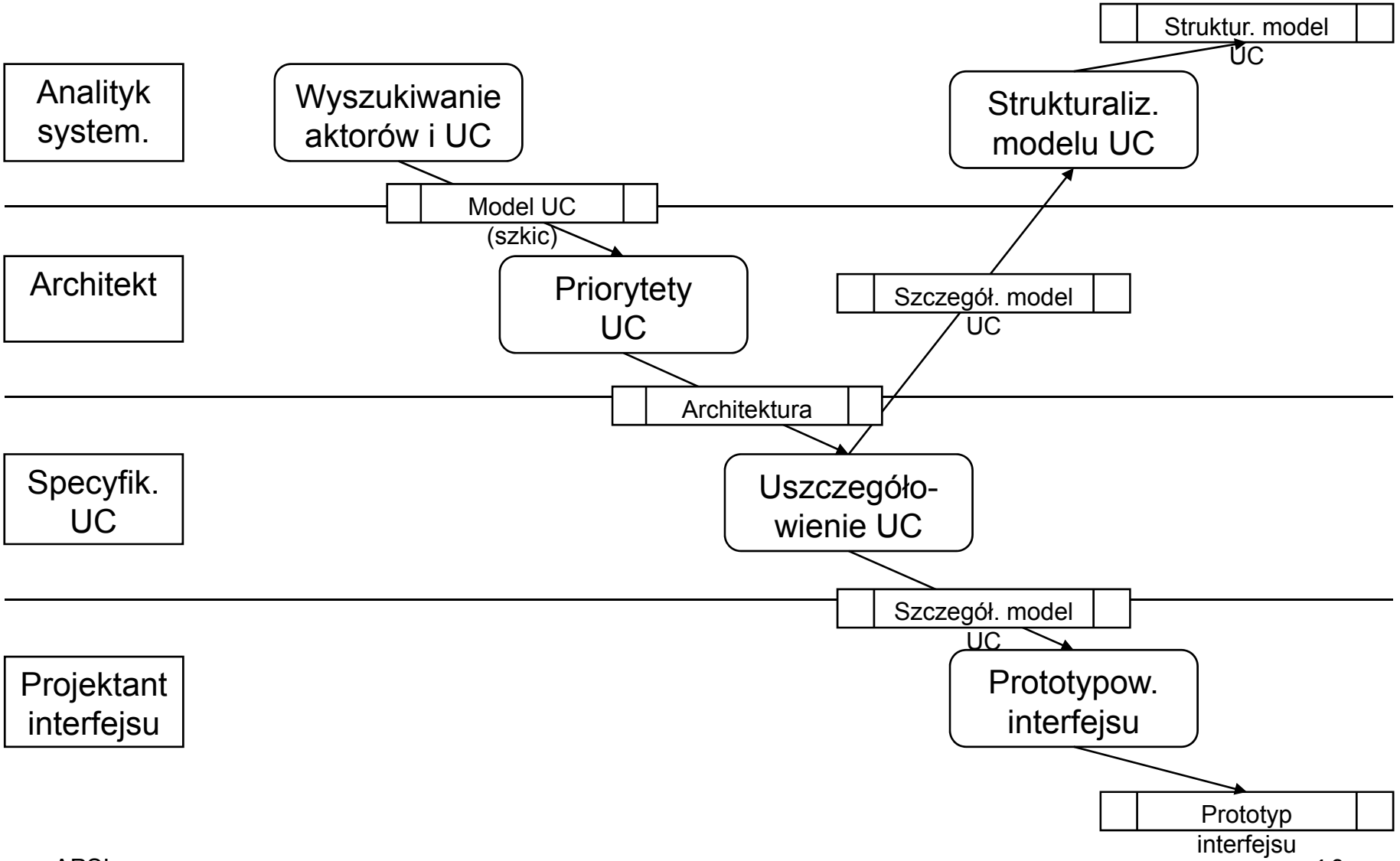
Analiza przypadków użycia

- rozszerzanie i uszczegóławianie opisów
- wyszukiwanie związków pomiędzy przypadkami użycia
  - wyodrębnianie zachowań wspólnych (include, extends)
  - ustalanie szablonów interakcji (generalizacja)
- weryfikacja jakości modelu przypadków użycia:
  - niesprzeczność,
  - spójność,
  - kompletność

Zarys procesu modelowania

- identyfikacja aktorów i przypadków użycia,
- zapis przypadków użycia w formacie wysokiego poziomu,
- stworzenie diagramów, strukturalizacja i powiązanie przypadków użycia,
- określenie wagi i rankingu aktorów i przypadków użycia,
- zapis przypadków istotnych architektonicznie w formie rozszerzonej,
- w razie potrzeby – zapis przypadków typu rzeczywistego *(real)* w formie rozszerzonej,
- analiza – zapis wszystkich przypadków użycia w formie rozszerzonej,
- projektowanie – przechodzenie od przypadków typu istotnego *(essential)* do rzeczywistego.

Diagram procesu



Wymagania niefunkcjonalne

- niezawodność, dostępność (fault-tolerance, accessibility)
  - dostępność
    - 99,72 – 1 dzień/rok
    - 99,93 – 6 h/rok
    - 99,99 – 1 h/rok
  - MTBF (Mean Time Between Failures)
    - maksymalizacja odporności na awarie – redundancja sprzętu, rozwiązania programowe
  - MTTR (Mean Time To Recover)
    - minimalizacja czasu odtwarzania – dobór technologii oprogramowania systemowego, procedury administratorskie

Wymagania niefunkcjonalne

- bezpieczeństwo (security)
  - klasy zakresu zagrożeń:
    - klasa Internal Business – dostęp wyłącznie dla jednostek wewnętrznych organizacji
    - klasa Contract Business – dostęp dla podmiotów zewnętrznych związanych umowami
    - klasa Public Business – dostęp publiczny
  - najistotniejsze zagrożenia:
    - anonimowy dostęp
    - nieautoryzowany dostęp do danych (odczyt, wstawienie, aktualizacja, usunięcie): w spoczynku / przy przesyłaniu
    - nieautoryzowana instalacja / reinstalacja / usunięcie oprogramowania aplikacyjnego / systemowego
    - uszkodzenie / zniszczenie fizyczne sprzętu / nośników danych

Wymagania niefunkcjonalne

- bezpieczeństwo (security)
  - przeciwdziałanie
    - środki techniczne
    - mechanizmy organizacyjne
    - zabezpieczenia fizyczne
  - wykrywanie i działania naprawcze
    - środki techniczne
    - procedury awaryjne

Wymagania niefunkcjonalne

- pojemność (capacity)
  - miary ilościowe: liczba użytkowników, elementów systemu, ilość danych
    - dobór technologii pod względem pojemności
- wydajność, sprawność, efektywność (performance, efficiency, effectiveness)
  - wydajność: bezwzględna miara ilościowa realizacji funkcji systemu
  - sprawność: miara ilościowa realizacji funkcji systemu względem używanych zasobów
  - efektywność: miara stopnia, w jaki system wspomaga użytkowników w wykonywaniu ich pracy
  - dobór technologii pod względem wydajnościowym

Wymagania niefunkcjonalne

- zarządzalność, łatwość utrzymania (manageability, maintainability)
  - łatwość wykonywania procedur administratorskich w ramach rozwiązania, takich jak np. zarządzanie oprogramowaniem, użytkownikami, danymi, monitorowanie, strojenie
    - dobór architektury systemu
    - dobór specjalistycznego oprogramowania
    - wbudowanie mechanizmów w system

Wymagania niefunkcjonalne

- wiarygodność (reliability)
  - miara stopnia, w jakim system wykonuje swoje funkcje w sposób poprawny (stopnia zaufania do systemu)
    - dobór technologii
    - jakość i pełność testów
- trwałość, odporność (robustness)
  - zdolność systemu do działania pomimo występowania zdarzeń niekorzystnych, awarii
    - dobór technologii odpornych na awarie
    - modularna architektura systemu
    - architektura rozproszona

Wymagania niefunkcjonalne

- użyteczność (usability)
  - stopień spełnienia wymagań użytkowników i przynoszenia im korzyści w realizacji ich zadań
- ergonomia (ergonomics)
  - łatwość wykorzystywania funkcji systemu
- zrozumiałość (understandability)
  - zdolność systemu do komunikowania się z użytkownikami w prosty i odpowiedni dla nich sposób
- wielojęzyczność (internationalization)
  - zdolność systemu do komunikowania się w wielu językach

Wymagania niefunkcjonalne

- zgodność (conformability): normy, reguły, infrastruktura
  - zgodność rozwiązania z obowiązującym prawem i innymi regulacjami
- kompatybilność (compatibility)
  - zdolność rozwiązania do współpracy z innymi systemami
- organizacja (organization)
  - wzajemne dostosowanie systemu i organizacji, w której ma funkcjonować
- topologia (topology)
  - sposób konstrukcji rozwiązania pod względem topologii węzłów i sieci w systemie

  - Wszystkie powyższe można rozumieć jako ograniczenia projektu

Wymagania niefunkcjonalne

- modyfikowalność (modifiability)
  - zdolność rozwiązania do wnoszenia modyfikacji do jego funkcjonalności
- indywidualizacja (customizability)
  - zdolność rozwiązania do dostosowywania interfejsu i funkcjonalności systemu do specyficznych wymagań poszczególnych użytkowników
- uniwersalność, elastyczność (versatility)
  - możliwość zastosowania rozwiązań w innych systemach
- przenośność (portability)
  - zdolność rozwiązania do działania na różnych platformach sprzętowo-systemowych

Wymagania niefunkcjonalne

- reużywalność (reusability)
  - możliwość wykorzystania elementów rozwiązania w wielu elementach danego systemu i/lub w innych systemach
- rozszerzalność (extendibility)
  - zdolność systemu do dodawania nowych elementów funkcjonalnych
- skalowalność (scalability)
  - zdolność systemu do zwiększania parametrów ilościowych – wydajności, pojemności
  - wszystkie wymienione wymagają:
    - doboru odpowiedniej technologii
    - opracowania odpowiedniej architektury

Wymagania niefunkcjonalne

- kompletność (completeness)
  - miara stopnia pokrycia przez system funkcjonalności istotnej dla organizacji i użytkowników
    - pełność testów i zapewnienie zgodności z modelem przypadków użycia
- testowalność (testability)
  - zdolność rozwiązania do łatwego, wygodnego i efektywnego testowania
    - wykorzystanie narzędzi do automatyzacji procesu testowania
    - wbudowanie mechanizmów w system

Zasady definiowania wymagań ilościowych

- określenie mierzonych wartości
  - formułowanie założeń w jednostkach biznesowych, a nie fizycznych
    - np. pomiar liczby dokumentów (a nie GB)
  - wyeliminowanie ewentualnych niejednoznaczności w określeniu tego, co ma podlegać pomiarowi
    - np. czas wykonania transakcji na serwerze bazy danych (a nie na stacji klienckiej)
- określenie metody pomiaru
  - zdefiniowanie mechanizmu technologicznego i sposobu (metodologii) wykonywania pomiaru
    - pomiar ręczny / elektroniczny
    - wykorzystanie zewnętrznych narzędzi pomiarowych / wbudowanie mechanizmów w rozwiązanie
    - sposób generowania obciążenia

Zasady definiowania wymagań ilościowych

- określenie środowiska i warunków pomiaru
  - zdefiniowanie sprzętu, oprogramowania systemowego i podstawowego
    - platforma sprzętowo-systemowa, serwery aplikacyjne, bazy danych, technologia sieci, etc.
  - określenie warunków technicznych
    - uruchomione i działające równolegle oprogramowanie, założone obciążenie testowanego systemu i innego oprogramowania
  - określenie warunków organizacyjnych
    - organizacja testów, zespołu testerów
- określenie interpretacji wyników
  - zdefiniowanie sposobu wyliczania rezultatu testu z wyników pomiarów oraz poziomu akceptowalności rezultatu
    - zalecane: N% wyników jest poniżej/powyżej zadanej wartości
    - lub: średnia pomiarów jest poniżej/powyżej zadanej wartości

Opis

Przypadek użycia umożliwia wprowadzenie do systemu danych wniosku o zarejestrowanie nowego gospodarstwa lub aktualizację danych już zarejestrowanego.

Aktorzy

Operator danych wniosków powierzchniowych i gospodarstw (aktywny), zwany dalej operatorem.

Warunki wstępne

- Dokument wniosku został zarejestrowany w przypadku użycia Rejestracja dokumentu wchodzącego.
  Wniosek jest w stanie *Przyjęty* lub *Wprowadzony*.

Warunki końcowe

- Dane wniosku zostały zapisane w systemie.
- Wniosek jest w stanie *Wprowadzony*.
- Operacja wprowadzenia danych wniosku została zapisana w dzienniku systemowym.
- Wniosek jest gotowy do realizacji przypadku użycia Kontrola i akceptacja wniosku o rejestrację danych gospodarstwa.

Przebieg podstawowy

Operator	System
1. Aktor wywołuje funkcję wprowadzenia danych gospodarstwa/gospodarza. 2. Aktor określa parametry dla raportu operacyjnego ***Wybór wniosku o rejestrację danych gospodarstwa***.	3. System generuje i prezentuje raport operacyjny.
4. Aktor dokonuje wyboru wniosku, którego dane chce wprowadzać.	5. System wyświetla wypełniony posiadanymi o gospodarstwie i gospodarzu danymi formularz, umożliwiający wprowadzenie danych z wniosku.
6. Aktor podaje dane z dokumentu wniosku (w zakresie D1).	7. System przeprowadza kontrolę kompletności (R1) i kontrolę administracyjną prostą (R2) wprowadzonych danych. 8. System rejestruje wprowadzenie danych wniosku.

Alternatywne przebiegi zdarzeń

A1. Korekta i uzupełnienie wprowadzonych danych.

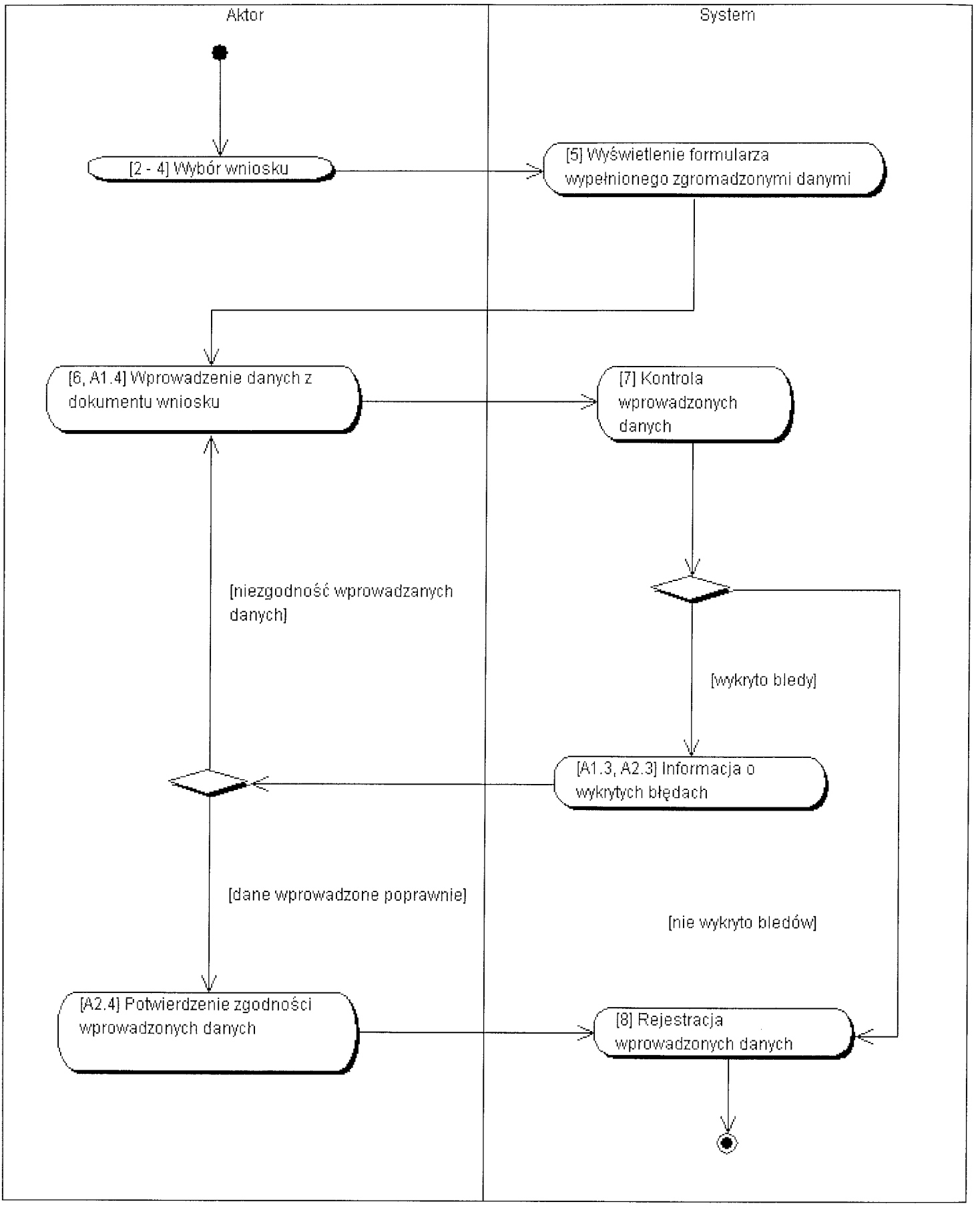
Operator	System
A1.1. Kroki 1 – 7 przebiegu podstawowego.	
	A1.2. System generuje i prezentuje raport o wykrytych błędach (D6).
A1.3. Aktor dokonuje korekty i uzupełnień wprowadzonych danych.	A1.4. System rejestruje dokonane zmiany (D3). A1.5. Powrót do kroku 7 przebiegu podstawowego.

A2. Potwierdzenie zgodności wprowadzonych danych

Operator	System
A2.1. Kroki 1 – 7 przebiegu podstawowego.	
	A2.2. System generuje i prezentuje raport o wykrytych błędach (D6).
A2.3. Aktor zatwierdza wprowadzone, błędne dane jako zgodne z danymi dokumentu wniosku.	A2.4. Powrót do kroku 8 przebiegu podstawowego.

Sytuacje wyjątkowe w obsłudze procesu

W1. Wprowadzany wniosek nie jest w stanie *Przyjęty* lub *Wprowadzony*. Odmowa modyfikacji danych wniosku.


Aktor
System
[2 - 4] Wybór wniosku
[5] Wyświetlenie formularza wypełnionego zgromadzonymi danymi
[6, A1.4] Wprowadzenie danych z dokumentu wniosku
[7] Kontrola wprowadzonych danych
[niezgodność wprowadzanych danych]
[wykryto bledy]
[A1.3, A2.3] Informacja o wykrytych błędach
[dane wprowadzone poprawnie]
[nie wykryto bledów]
[A2.4] Potwierdzenie zgodności wprowadzonych danych
[8] Rejestracja wprowadzonych danych

Zakres przetwarzanych danych

Dane wprowadzane:

D1. Zawartość opisana w Dodatku D: Wniosek o rejestrację gospodarstwa.

D2. Informacja o dokonanej operacji wprowadzenia danych wniosku:

D2.1. typ operacji (wprowadzenie danych wniosku o rejestrację danych gospodarstwa, wprowadzenie danych wniosku o korektę danych gospodarstwa),

D2.2. data operacji,

D2.3. miejsce operacji,

D2.4. przeprowadzający operację,

D2.5. numer kancelaryjny dokumentu źródłowego.

D3. Zmiany dokonane:

D3.1. stara wartość,

D3.2. nowa wartość,

D3.3. operacja (D2).

D4. Raporty operacyjne

D5. Wybór wniosku o rejestrację danych gospodarstwa

Parametry ustalone

D5.1. status wniosku = *Przyjęty,*

D5.2. typ wniosku = rejestracja lub korekta.

D6. Raport o wykrytych błędach

Reguły przetwarzania danych

R1. Kontrola kompletności wprowadzonych danych polega na sprawdzeniu, czy wszystkie wymagane Przepisami Proceduralnymi dane zostały podane.

R2. Kontrola administracyjna prosta wprowadzonych danych polega na:

R2.1. weryfikacji poprawności konstrukcji podanego numeru PESEL,

R2.2. weryfikacji poprawności konstrukcji podanego numeru REGON,

R2.3. weryfikacji poprawności konstrukcji podanego numeru rachunku bankowego.

Wymagania niefunkcjonalne

Nie określono.