

Przykładowy schemat rozwoju modelu projektu podczas trzech sprintów z równoległe rozwijanymi przypadkami użycia.

Zofia Kruczkiewicz

Opis świata rzeczywistego

- Opis zasobów ludzkich: procesy biznesowe bez korzystania z programu (inżynieria odwrotna do wymagań funkcjonalnych)
- Strategia, przepisy, inne ograniczenia: problemy, które należy uwzględnić projektując i wykonując program (inżynieria odwrotna do wymagań niefunkcjonalnych)
- Dane techniczne: dane ilościowe dotyczące danych i procesów, przygotowanie do korzystania z technologii (inżynieria odwrotna do wymagań niefunkcjonalnych)

Wymagania funkcjonalne

1. Dodawanie (należy podać atrybuty, należy podać sposób identyfikacji)
2. Dodawanie (należy podać atrybuty, należy podać sposób identyfikacji)
3. Dodawanie (należy podać atrybuty, należy podać sposób identyfikacji)
4. Dodawanie (należy podać atrybuty, należy podać sposób identyfikacji)
5. Wykonanie... (na podstawie danych identyfikujących oraz danych identyfikujących)
6. Wykonanie (na podstawie danych identyfikujących oraz danych identyfikujących lub/ i)
7. Wykonanie.... (na podstawie danych identyfikujących oraz danych identyfikujących)
8. Wykonanie.... (na podstawie danych identyfikujących... , danych identyfikujących lub/ i poszukiwanych w oraz sprawdzenia dostępności)
9. Analiza1 (dane wejściowe do ustalenia, algorytmy do ustalenia)
10. Analiza2 (dane wejściowe do ustalenia, , algorytmy do ustalenia)

Wymagania нефunkcjonalne

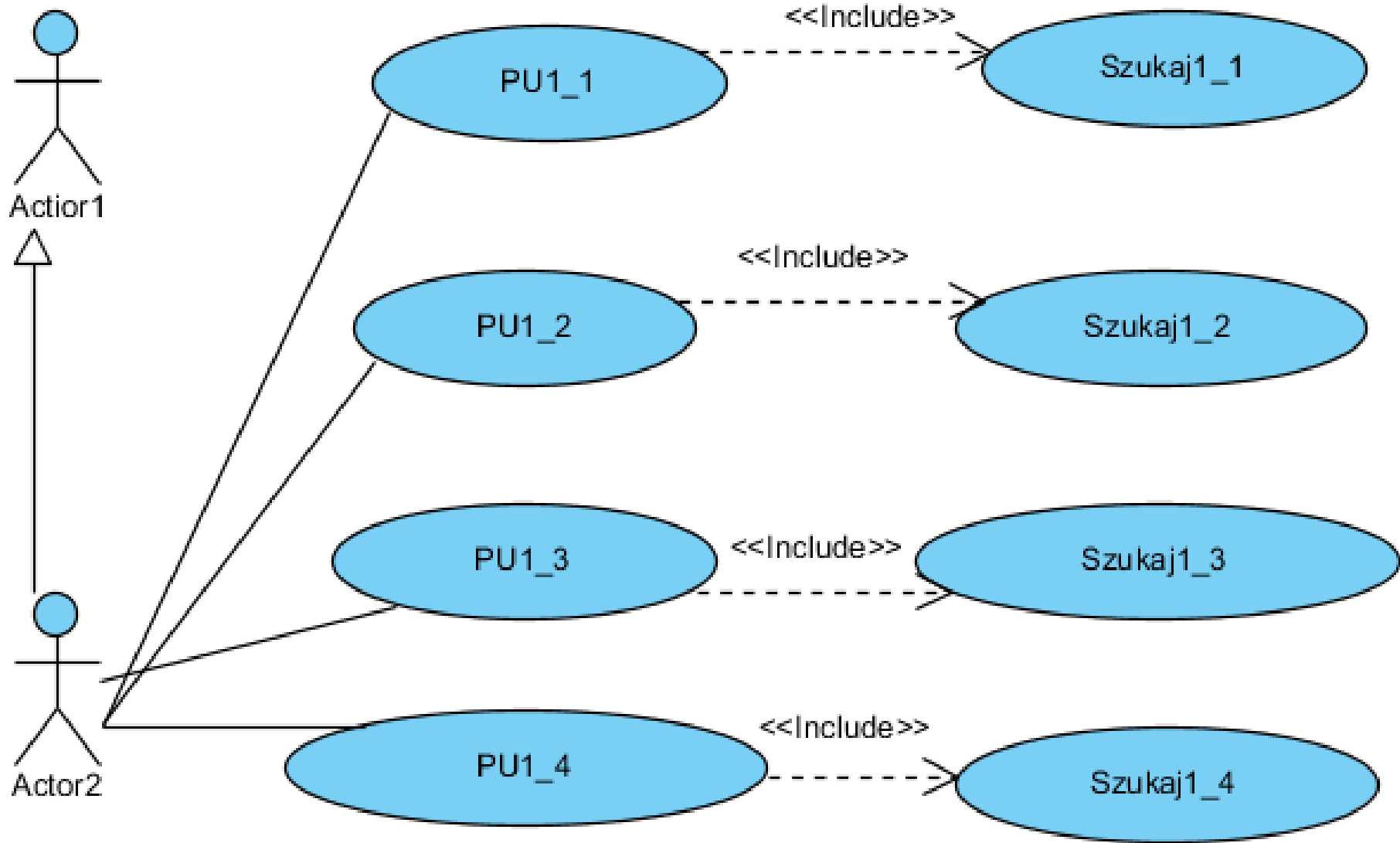
- Liczba poszczególnych danych
- Liczba dopuszczalnych procesów, czas procesów i ich częstotliwość itp.
- Ograniczenia wydajnościowe
- Czy jest wymagany masowy dostęp (Internet)?
- Proponowane technologie

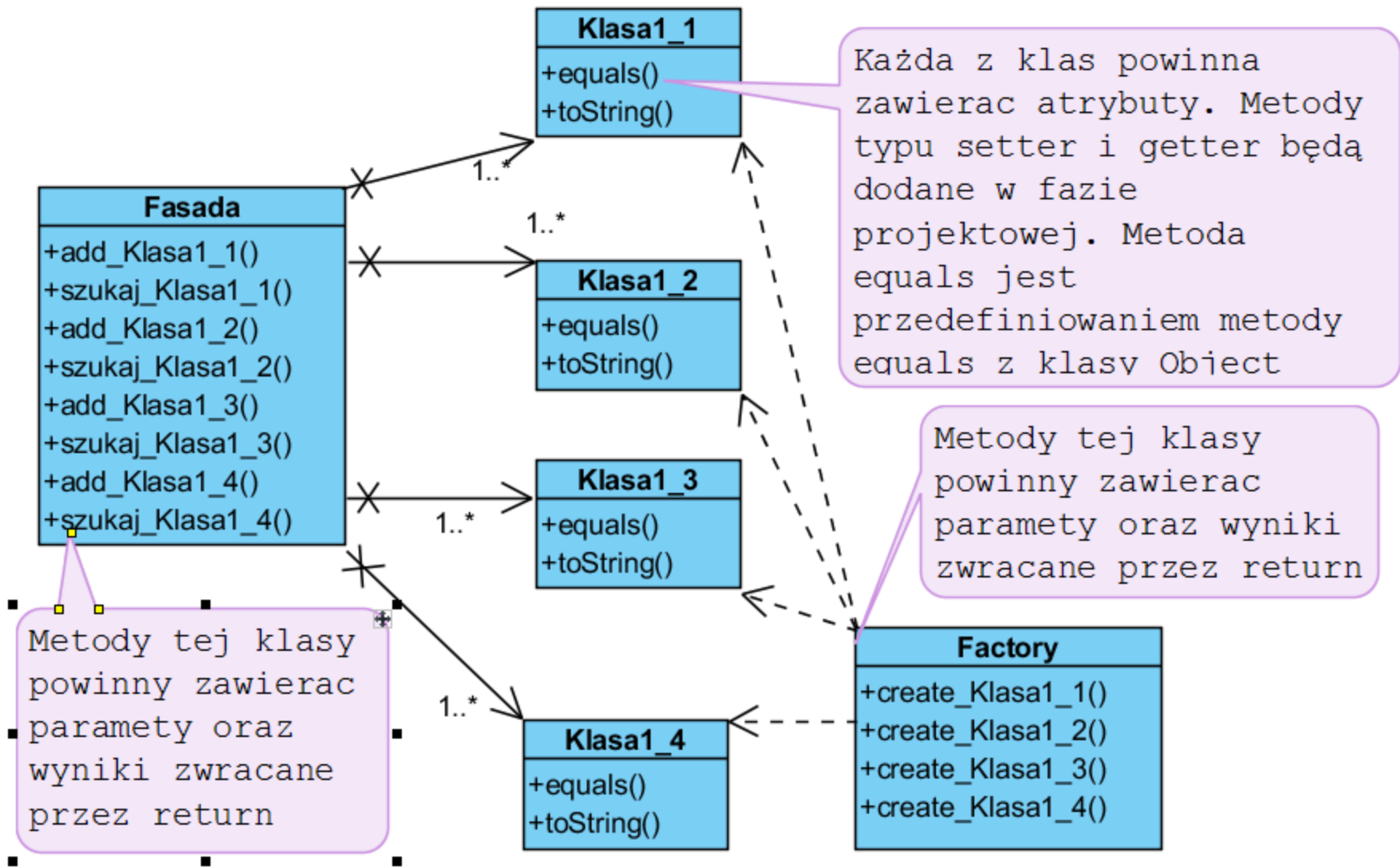
1 Sprint

Wymagania funkcjonalne

1. PU1_1 Dodawanie (należy podać atrybuty, należy podać sposób identyfikacji)
2. PU1_2 Dodawanie (należy podać atrybuty, należy podać sposób identyfikacji)
3. PU1_3 Dodawanie (należy podać atrybuty, należy podać sposób identyfikacji)
4. PU1_4 Dodawanie (należy podać atrybuty, należy podać sposób identyfikacji)

Diagramy przypadków użycia i klas po integracji czterech PU po zakończeniu 1-go Sprintu





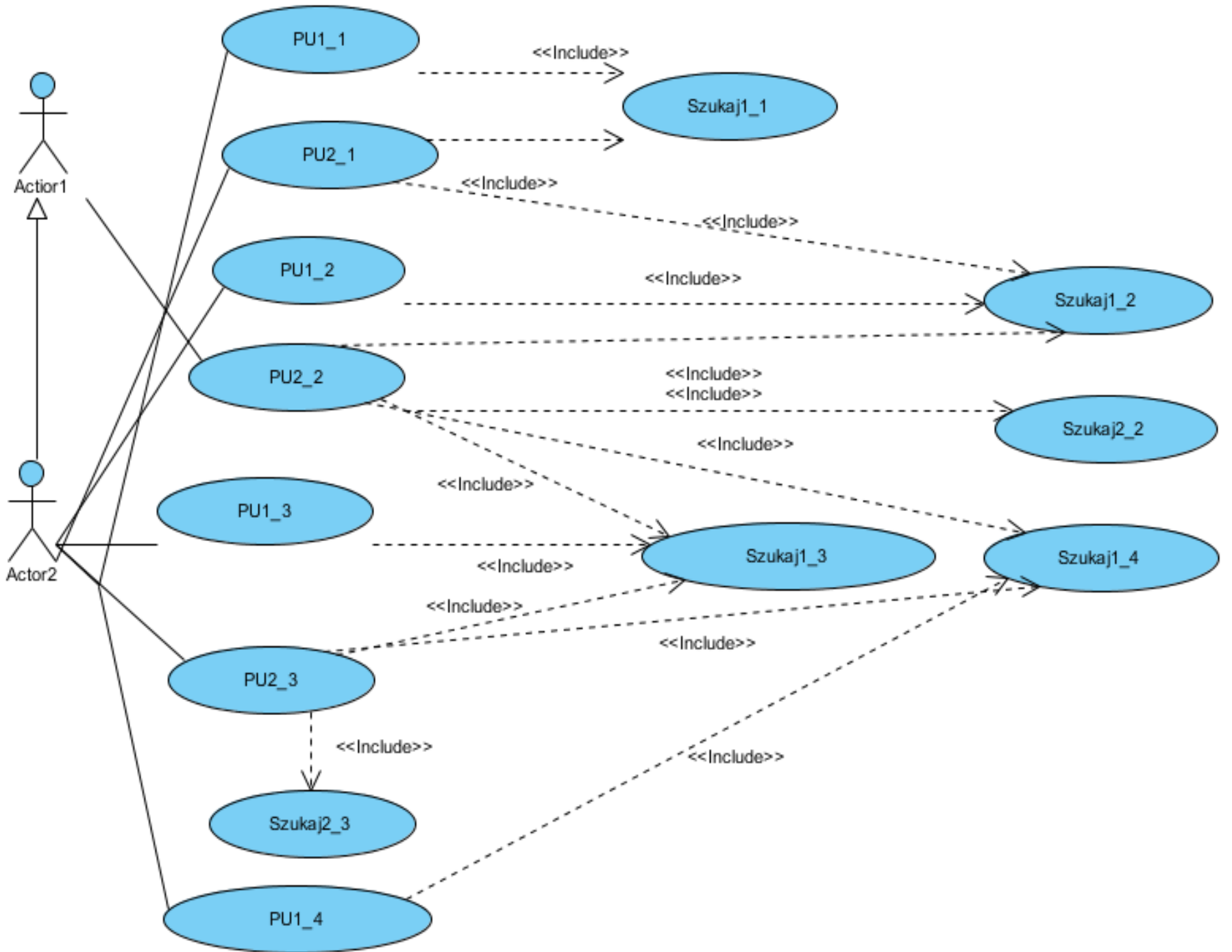
Diagramy sekwencji, aktywności, stanów

2 Sprint

Wymagania funkcjonalne

1. PU2_1 Wykonanie... (na podstawie danych identyfikujących oraz danych identyfikujących)
2. PU2_2 Wykonanie (na podstawie danych identyfikujących oraz danych identyfikujących lub/ i)
3. PU2_3 Wykonanie.... (na podstawie danych identyfikujących oraz danych identyfikujących)

Diagramy przypadków użycia i klas po integracji trzech PU po zakończeniu 2-go Sprintu, realizowanych na diagramach po 1-ym Sprincie



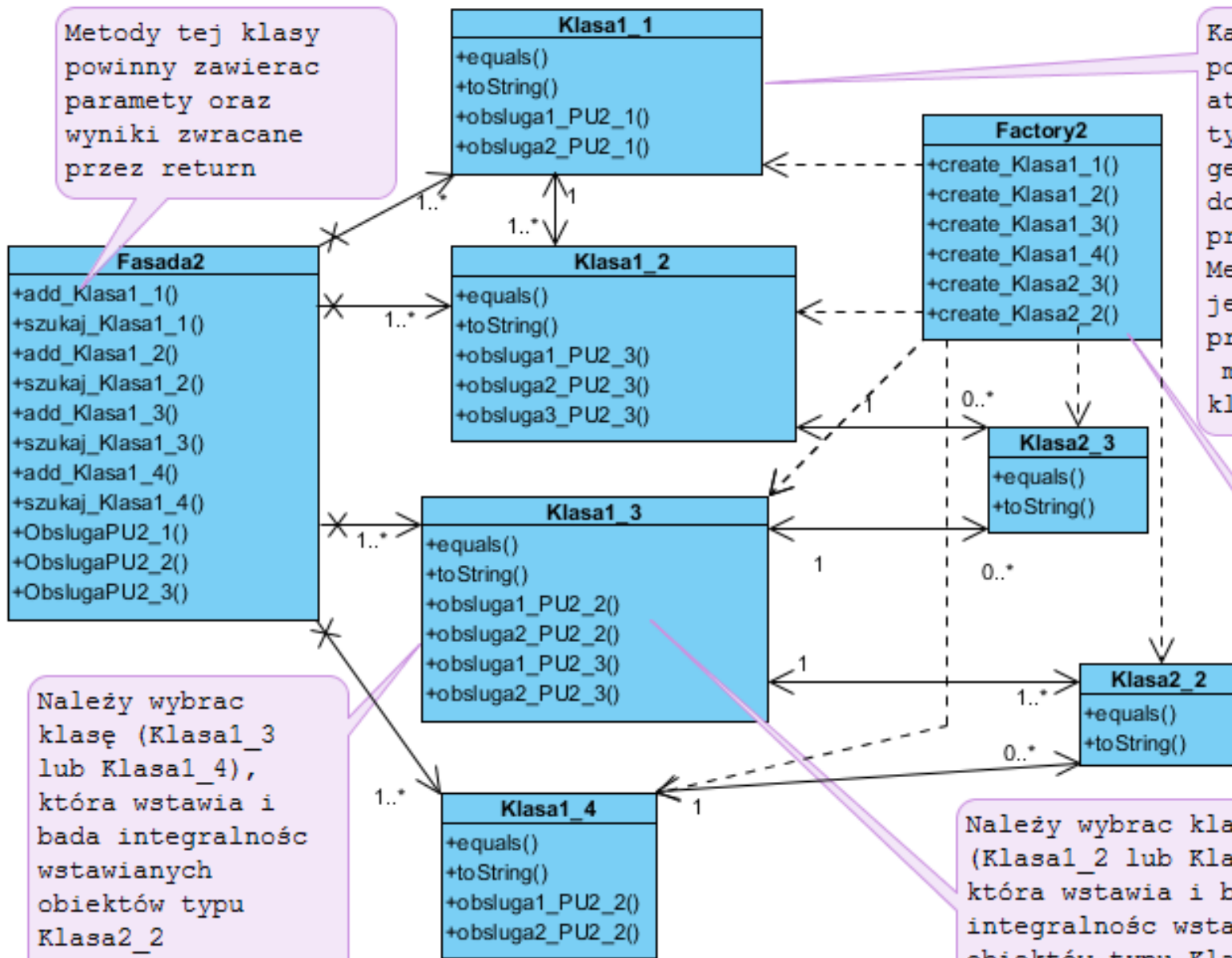
Metody tej klasy powinny zawierać parametry oraz wyniki zwracane przez return

Każda z klas powinna zawierać atrybuty. Metody typu setter i getter będą dodane w fazie projektowej. Metoda equals jest przeddefiniowaniem metody equals z klasy Object

Metody tej klasy powinny zawierać parametry oraz wyniki zwracane przez return

Należy wybrać klasę (Klasa1_3 lub Klasa1_4), która wstawia i bada integralność wstawianych obiektów typu Klasa2_2

Należy wybrać klasę (Klasa1_2 lub Klasa1_3), która wstawia i bada integralność wstawianych obiektów typu Klasa2_3



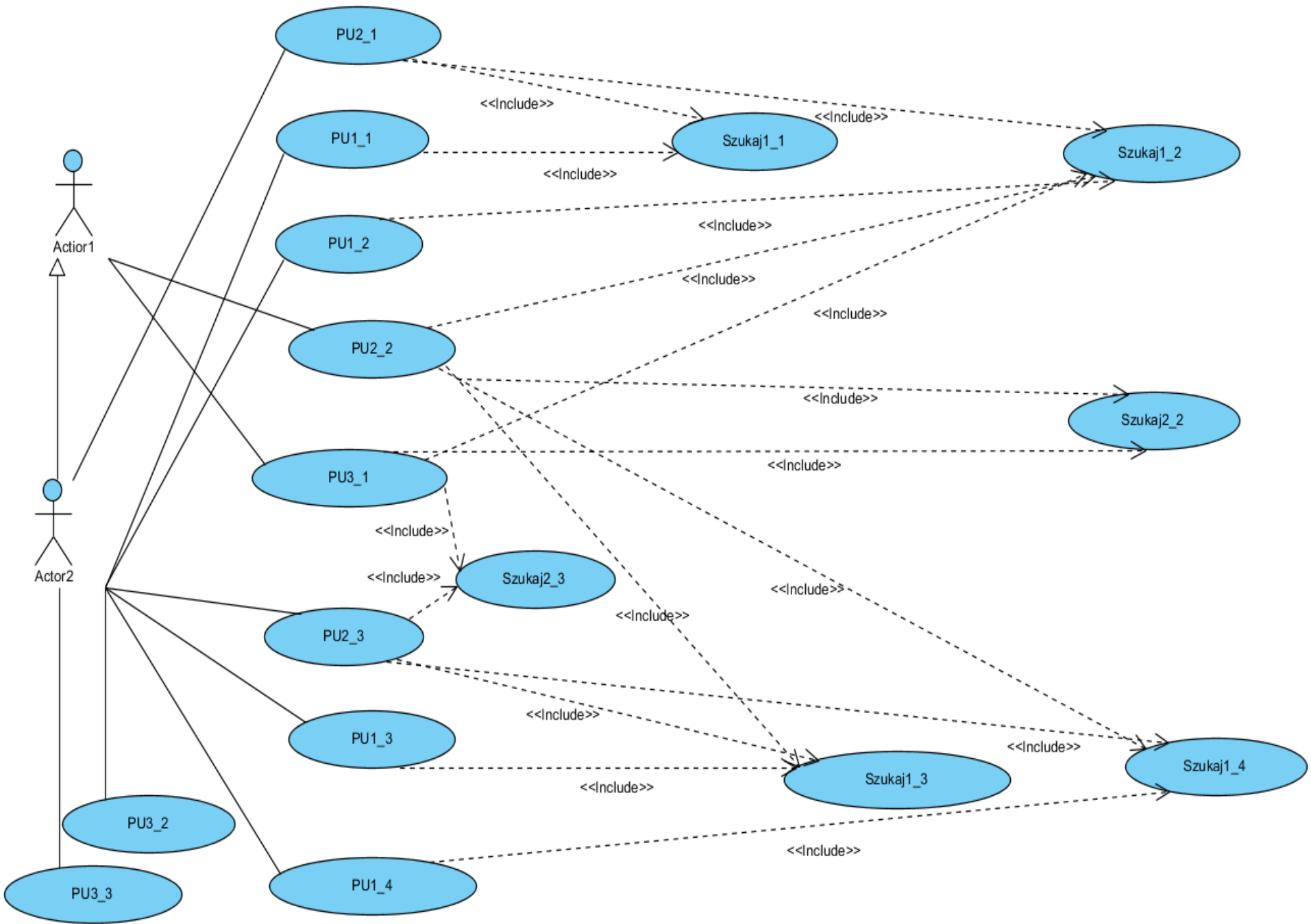
Diagramy sekwencji, aktywności, stanów

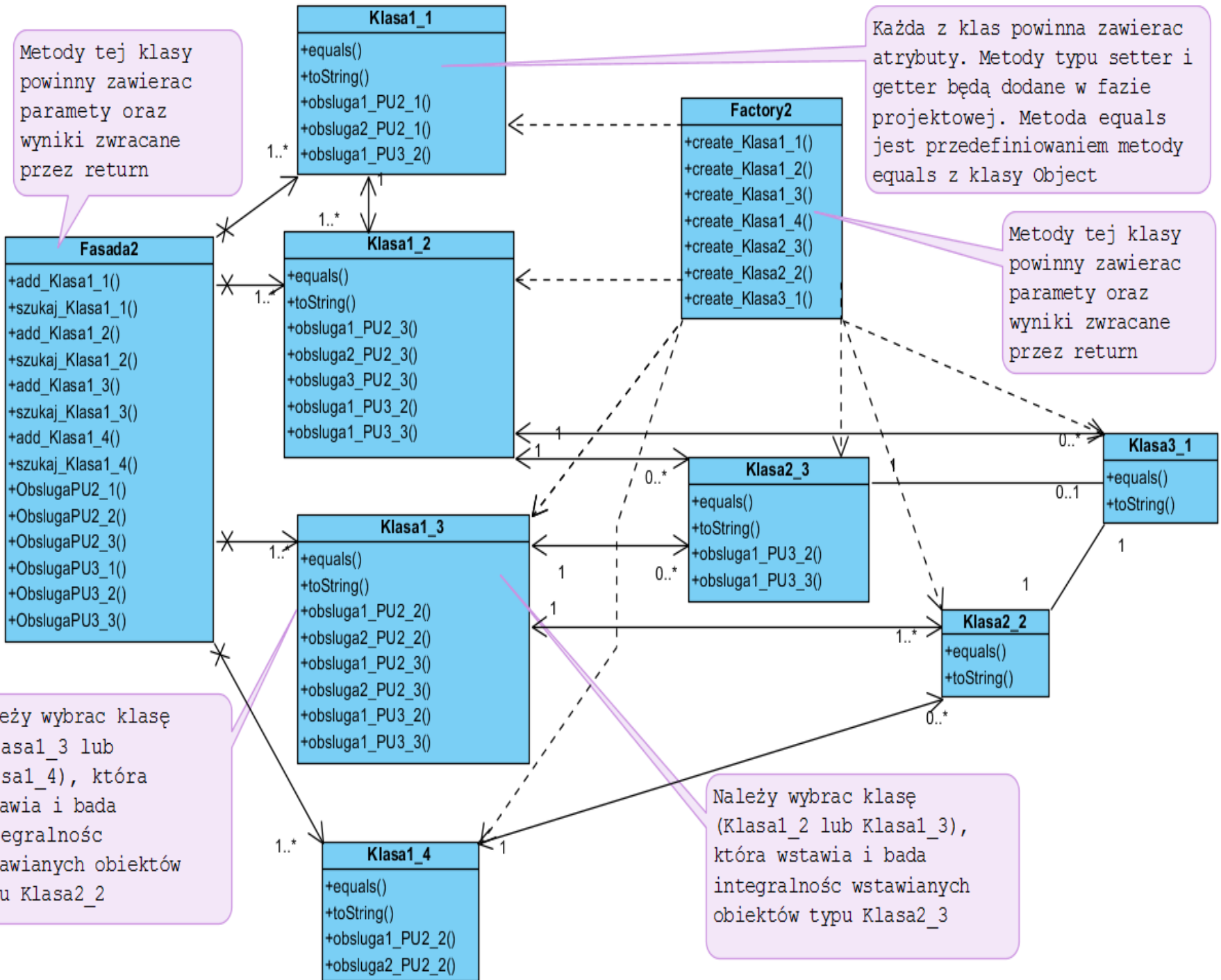
3 Sprint

Wymagania funkcjonalne

1. PU3_1 Wykonanie.... (na podstawie danych identyfikujących... , danych identyfikujących lub/ i poszukiwanych w oraz sprawdzenia dostępności)
2. PU3_2 Analiza1 (dane wejściowe do ustalenia, algorytmy do ustalenia)
3. PU3_3 Analiza2 (dane wejściowe do ustalenia, , algorytmy do ustalenia)

Diagramy przypadków użycia i klas po integracji trzech PU po zakończeniu 3-go Sprintu, realizowanych na diagramach po 2-im Sprincie





Metody tej klasy powinny zawierać parametry oraz wyniki zwracane przez return

Każda z klas powinna zawierać atrybuty. Metody typu setter i getter będą dodane w fazie projektowej. Metoda equals jest przeddefiniowaniem metody equals z klasy Object

Metody tej klasy powinny zawierać parametry oraz wyniki zwracane przez return

Należy wybrać klasę (Klasa1_3 lub Klasa1_4), która wstawia i bada integralność wstawianych obiektów typu Klasa2_2

Należy wybrać klasę (Klasa1_2 lub Klasa1_3), która wstawia i bada integralność wstawianych obiektów typu Klasa2_3

Diagramy sekwencji, aktywności, stanów