Politechnika Warszawska

Wydział Elektroniki i Technik Informacyjnych

System wspierający organizację pracy zespołowej z wykorzystaniem metodyki Scrum.

**Wersja**: 0.1.12  
**Data ostatniej modyfikacji**: 21.04.2013

|  |  |
| --- | --- |
| **Przedmiot** | |
| Analiza i Projektowanie Systemów Informatycznych | |
| **Prowadzący projekt** | |
| mgr inż. Piotr Sałata | |
| **Zespół autorski** | |
| Michał Aniserowicz  Michał Januszewski  Wojciech Marcinkowski  Rafał Podkowiński  Jakub Turek | *m.aniserowicz@stud.elka.pw.edu.pl* *michal.januszewski@gmail.com*  *rpodkowi@mion.elka.pw.edu.pl* *j.turek@stud.elka.pw.edu.pl* |

**Historia zmian**:

|  |  |  |  |
| --- | --- | --- | --- |
| **Data zmiany** | **Autor zmiany** | **Wersja dokumentu** | **Opis zmian** |
| 13.04.2013 | M. Januszewski,  R. Podkowiński | 0.1.1 | Stworzono dokument. Dodano "Wprowadzenie". |
| 15.04.2013 | M. Aniserowicz,  J. Turek | 0.1.2 | Dodano "Słownik pojęć". |
| 16.04.2013 | W. Marcinkowski | 0.1.3 | Dodano "Definicję celów przedsięwzięcia". |
| 16.04.2013 | M. Aniserowicz | 0.1.4 | Dodano "Integrację z systemami zewnętrznymi". |
| 18.04.2013 | M. Januszewski | 0.1.5 | Dodano "Ogólny schemat przypadków użycia". |
| 19.04.2013 | M. Aniserowicz,  J. Turek | 0.1.6 | Dodano przypadki użycia zarządzania: Projektami, Wydaniami, Sprintami, Historyjkami Użytkownika, Zadaniami, Przypadkami Testowymi, Błędami, Przeszkodami, Użytkownikami. |
| 19.04.2013 | M. Januszewski,  W. Marcinkowski | 0.1.7 | Dodano przypadki użycia zarządzania stanem: Historyjek Użytkownika, Zadań, Przypadków Testowych, Błędów. |
| 19.04.2013 | R. Podkowiński | 0.1.8 | Dodano przypadek użycia: "Dodawanie komentarzy". |
| 20.04.2013 | M. Aniserowicz | 0.1.9 | Dodano przypadki użycia zarządzania: Tablicą Zadań, Tablicą Scrum'u. |
| 20.04.2013 | J. Turek | 0.1.10 | Dodano przypadki użycia integracji z: kalendarzem Google, GitHub, dokumentami Google. |
| 21.04.2013 | W. Marcinkowski,  R. Podkowiński | 0.1.11 | Dodano "Hierarchię wymagań funkcjonalnych". |
| 21.04.2013 | M. Aniserowicz,  M. Januszewski,  J.Turek | 0.1.12 | Dodano "Opis wymagań niefunkcjonalnych". |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Spis treści**:

[1. Wprowadzenie 6](#_Toc354336466)

[1.1. Słownik pojęć 7](#_Toc354336467)

[1.2. Definicja celów przedsięwzięcia 9](#_Toc354336468)

[1.3. Integracja z systemami zewnętrznymi 9](#_Toc354336469)

[1.3.1. Kalendarz Google 9](#_Toc354336470)

[1.3.2. GitHub 10](#_Toc354336471)

[1.3.3. Dokumenty Google 10](#_Toc354336472)

[2. Opis wymagań funkcjonalnych 12](#_Toc354336473)

[2.1. Hierarchia wymagań funkcjonalnych 12](#_Toc354336474)

[2.2. Opis przypadków użycia 18](#_Toc354336475)

[2.2.1. Ogólny schemat przypadków użycia 18](#_Toc354336476)

[2.2.2. Zarządzanie Projektami 19](#_Toc354336477)

[2.2.3. Zarządzanie Wydaniami 20](#_Toc354336478)

[2.2.4. Zarządzanie Sprintami 21](#_Toc354336479)

[2.2.5. Zarządzanie Historyjkami Użytkownika 22](#_Toc354336480)

[2.2.6. Zarządzanie Zadaniami 23](#_Toc354336481)

[2.2.7. Zarządzanie Przypadkami Testowymi 24](#_Toc354336482)

[2.2.8. Zarządzanie Błędami 25](#_Toc354336483)

[2.2.9. Zarządzanie Przeszkodami 26](#_Toc354336484)

[2.2.10. Zarządzanie Użytkownikami 27](#_Toc354336485)

[2.2.11. Zarządzanie stanem Historyjek Użytkownika 28](#_Toc354336486)

[2.2.12. Zarządzanie stanem Zadań 29](#_Toc354336487)

[2.2.13. Zarządzanie stanem Przypadków Testowych. 30](#_Toc354336488)

[2.2.14. Zarządzanie stanem Błędów 31](#_Toc354336489)

[2.2.15. Dodawanie Komentarzy 32](#_Toc354336490)

[2.2.16. Zarządzanie Tablicą Zadań 33](#_Toc354336491)

[2.2.16. Zarządzanie Tablicą Scrum'u 34](#_Toc354336492)

[2.2.17. Integracja z kalendarzem Google 35](#_Toc354336493)

[2.2.18. Integracja z GitHub 36](#_Toc354336494)

[2.2.19. Integracja z dokumentami Google 37](#_Toc354336495)

[3. Opis wymagań niefunkcjonalnych 38](#_Toc354336496)

# 1. Wprowadzenie

Scrum jest to iteracyjna i inkrementalna metodyka prowadzenia projektów, zaliczana do metodyk zwinnych, która w szybkim tempie zyskuje popularność na całym świecie, również w Polsce. Metodyka ta zakłada podział pracy zespołu na krótkie, stałe okresy czasu (ang. Sprint), w których do systemu wprowadzane są zmiany dodające wartość funkcjonalną. Pełna definicja metodyki, w sposób formalny, została przedstawiona przez Kena Schwabera w 1986 roku.

Na pojedynczy sprint składają się następujące etapy:

1. Gromadzenie wymagań Właściciela Produktu, wraz z przypisanymi im Priorytetami. Wymagania gromadzone są w postaci Historyjek Użytkownika (ang. User Stories), na podstawie których formułowany jest Rejestr Wymagań (ang. Product Backlog).
2. Planowanie przebiegu Sprintu (ang. Sprint Planning). Spośród Rejestru Wymagań wybierane są Historyjki Użytkownika o najwyższym Priorytecie. Następnie szacowany jest czas realizacji każdego Zadania, ryzyko, które niesie ono ze sobą oraz jego pracochłonność. Do oszacowania tych wielkości wykorzystywane są Punkty Złożoności (ang. Complexity Points). W ten sposób powstaje Rejestr Zaległości (ang. Sprint Backlog).
3. Faza implementacji, w której wykonywane są zaplanowane Zadania. Ważne jest, aby w trakcie implementacji zakres Sprintu nie zmieniał się. Przydział Zadań do poszczególnych członków zespołu nie jest odgórny - poszczególni członkowie zespołu sami wybierają zadania, które zrealizują.

Jednym z podstawowych założeń metody jest organizacja codziennych, 15-minutowych spotkań, podczas których członkowie zespołu przedstawiają:

* zadania zrealizowane poprzedniego dnia,
* trudności, które napotkali w trakcie wykonywania zadań z poprzedniego dnia,
* zadania zaplanowane do wykonania w dniu kolejnym.

## 1.1. Słownik pojęć

* **System** - system informatyczny, którego dotyczy niniejszy dokument.
* **User (Użytkownik)** - użytkownik systemu, członek zespołu realizującego dany Projekt.
* **Role (Rola)** - rola przypisana Użytkownikowi. Może przyjąć następujące wartości:
  + **Product Owner (Właściciel Produktu)** - osoba reprezentująca klienta,
  + **Scrum Master (Mistrz Scrum'u)** - osoba odpowiedzialna za usuwanie wszelkich przeszkód uniemożliwiających zespołowi wykonanie zadania, oraz za poprawną implementację procesu i metod,
  + **Developer (Programista)** - osoba odpowiedzialna za implementację produktu,
  + **QA Tester (Quality Assurance Tester, Tester Zapewnienia Jakości)** - osoba odpowiedzialna za testowanie produktu.
* **Project (Projekt)** - produkt realizowany przez Użytkowników.
* **Release (Wydanie)** - stabilna wersja Projektu, gotowa do wdrożenia w środowisku produkcyjnym.
* **Sprint** - trwający około 5 dni roboczych etap tworzenia danego Release.
* **User Story (Historyjka Użytkownika)** - funkcjonalność systemu implementowana w ramach danego Sprintu.
* **Task (Zadanie)** - zadanie, którego wykonanie przyczyni się do implementacji danej User Story.
* **Bug (Błąd)** - błąd systemu, przypisany danej User Story lub istniejący samodzielnie w danym Sprincie.
* **Test Case (Przypadek Testowy)** - przypadek testowy przypisany dajej User Story.
* **Complexity Points (Punkty Złożoności)** - reprezentuja złożoność (trudność, jak również czas potrzebny na implementację) danej Historyjki Użytkownika, Zadania lub Błędu - każdemu z tych bytów przypisać można odpowiednią liczbę Complexity Points.
* **Business Value (Wartosć Biznesowa)** - wartość biznesowa danej Historyjki Użytkownika lub Błędu. Może przyjmować następujące wartości:
  + **Must Have (Obowiązkowy)**,
  + **Great (Bardzo Duży)**,
  + **Good (Duży)**,
  + **Average (Średni)**,
  + **Nice To Have (Mile Widziany)**.
* **Impediment (Przeszkoda)** - przeszkoda uniemożliwiająca zespołowi wykonywanie Zadań, istniejąca w danym Projekcie niezależnie od innych bytów (Wydań, Sprintów itp.).
* **Comment (Komentarz)** - komentarz dodany przez Użytkownika do Historyjki Użytkownika, Zadania, Błędu lub Przypadku Testowego.
* **Dashboard (Tablica Zadań)** - reprezentuje widok na wszystkie Zadania, Błędy i Przypadki Testowe przypisane do danego Użytkownika.
* **Scrum Board (Tablica Scrum'u)** - reprezentuje widok na pojedyncze Wydanie lub Sprint, zawierający wszystkie Historyjki Użytkownika i Błędy do niego należące, z podziałem na kategorie:
  + **Backlog (Rejestr Zaległości)** - byty odłożone w czasie,
  + **In Progress (W Toku)** - byty w trakcie implementacji,
  + **In Testing (Testowane)** - zaimplementowane byty podlegające testowaniu,
  + **Done (Gotowe)** - byty zaimplementowane i przetestowane.

## 1.2. Definicja celów przedsięwzięcia

Celem przedsięwzięcia jest realizacja systemu wspomagającego pracę w metodyce Scrum. System ten będzie umożliwiał wykonywanie następujących zadań:

* Definiowanie Projektu i jego podział na poszczególne Wydania.
* Definiowanie Wydań i ich podział na Historyjki Użytkownika.
* Podział "Historyjek Użytkownika" na Zadania, Przypadki Testowe i Błędy.
* Przypisywanie Użytkownika do określonego Zadania lub Błędu.
* Szacowanie złożoności czasowej i ryzyka określonej Historyjki Użytkownika, Zadania lub Błędu.
* Monitorowanie postępów w wykonywaniu Historyjek Użytkownika, Zadań i Błędów.
* Dodawanie komentarzy do Historyjek Użytkownika, Zadań i Błędów w postaci tekstu oraz załączników.
* Zarządzanie Historyjkami Użytkownika przy pomocy Scrum Board.

System ma na celu wspieranie, uporządkowanie oraz ułatwianie pracy wszystkim członkom zespołu, a w szczególności.

## 1.3. Integracja z systemami zewnętrznymi

### 1.3.1. Kalendarz Google

Integracja z kalendarzem Google obejmuje synchronizację dat rozpoczęcia i zakończenia:

* Projektów,
* Wydań,
* Sprintów.

W ramach integracji dany Projekt w Systemie może zostać powiązany z wybranym kalendarzem Google. Po rozpoczęciu lub zakończeniu Projektu, Wydania lub Sprintu wydarzenie to odnotowywane jest w kalendarzu. W opisie wydarzenia zawarte jest odniesienie do bytu, którego dane wydarzenie dotyczy. Domyślnie każdy subskrybent kalendarza powiadamiany jest o utworzeniu i zbliżającej się dacie wydarzenia drogą e-mailową.

### 1.3.2. GitHub

Integracja z serwisem GitHub, wspierającym obsługę systemu kontroli wersji Git, obejmuje możliwość automatycznego przypisywania Zmian (ang. Commit) Historyjkom Użytkownika, Zadaniom i Błędom, których ta zmiana dotyczy. Przypisanie to odbywa się poprzez umieszczenie identyfikatora Historyjki Użytkownika, Zadania lub Błędu w opisie zmiany przesłanym do repozytorium.

W ramach integracji w widoku Historyjki Użytkownika, Zadania lub Błędu widoczna jest sekcja Postęp Programisty (ang. Developer Progress), która zawiera informacje o wszystkich Zmianach, m.in.:

* treść opisu Zmiany,
* pliki zmodyfikowane w obrębie danej Zmiany,
* łączna liczba linii kodu dodanych w obrębie danej Zmiany,
* łączna liczba linii kodu zmodyfikowanych w obrębie danej Zmiany,
* łączna liczba linii kodu usuniętych w obrębie danej Zmiany.

### 1.3.3. Dokumenty Google

Integracja z serwisem Google Docs obejmuje generowanie zbiorczych podsumowań:

* Projektu, Wydania lub Sprintu, zawierających liczby Punktów Złożoności przypadających na każdego z członków zespołu.
* Projektu, zawierających liczby Punktów Złożoności przypadających na każdy Sprint.
* Efektywności oszacowań złożoności Sprintów, zawierających zestawienie różnic pomiędzy liczbą Punktów Złożoności szacowanych na dany Sprint, a liczbą Punktów Złożoności wypełnionych w tym Sprincie.

Podsumowania te generowane są w postaci arkuszy kalkulacyjnych (ang. Spreadsheets), opisanych zarówno danymi liczbowymi, jak również wykresami.

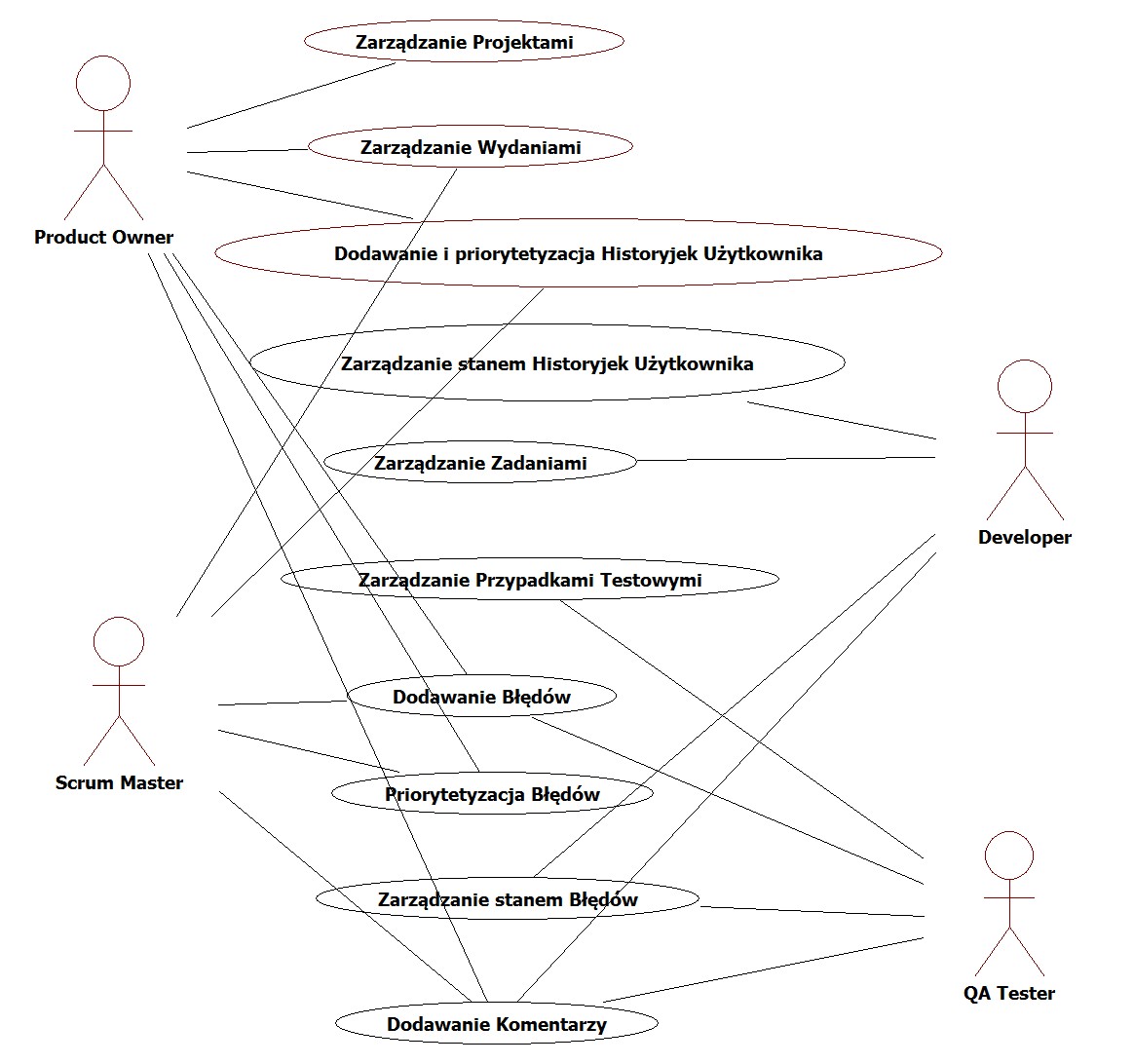
# 2. Opis wymagań funkcjonalnych

## 2.1. Hierarchia wymagań funkcjonalnych

1. Zarządzanie Projektami.
   1. Dodawanie Projektu.
      1. Definiowanie nazwy Projektu.
      2. Definiowanie opisu Projektu.
      3. Definiowanie daty rozpoczęcia Projektu.
      4. Definiowanie daty zakończenia Projektu.
   2. Edytowanie Projektu.
      1. Edytowanie nazwy Projektu.
      2. Edytowanie opisu Projektu.
      3. Edytowanie daty rozpoczęcia Projektu.
      4. Edytowanie daty zakończenia Projektu.
   3. Usuwanie Projektu.
   4. Wyświetlanie wszystkich Projektów.
2. Zarządzanie Wydaniami.
   1. Dodawanie Wydania.
      1. Definiowanie nazwy Wydania.
      2. Przypisanie Wydania do Projektu.
      3. Definiowanie opisu Wydania.
      4. Definiowanie daty rozpoczęcia Wydania.
   2. Edytowanie Wydania.
      1. Edytowanie nazwy Wydania.
      2. Przypisanie Wydania do innego Projektu.
      3. Edytowanie opisu Wydania.
      4. Edytowanie daty rozpoczęcia Wydania.
      5. Edytowanie daty zakończenia Wydania.
   3. Usuwanie Wydania.
   4. Wyświetlanie wszystkich Wydań.
3. Zarządzanie Sprintami.
   1. Dodawanie Sprintu.
      1. Definiowanie nazwy Sprintu.
      2. Przypisanie Sprintu do Wydania.
      3. Definiowanie opisu Sprintu.
      4. Definiowanie daty rozpoczęcia Sprintu.
      5. Definiowanie daty zakończenia Sprintu.
   2. Edytowanie Sprintu.
      1. Edytowanie nazwy Sprintu.
      2. Przypisanie Sprintu do innego Wydania.
      3. Edytowanie opisu Sprintu.
      4. Edytowanie daty rozpoczęcia Sprintu.
      5. Edytowanie daty zakończenia Sprintu.
   3. Usuwanie Sprintu.
   4. Wyświetlanie wszystkich Sprintów.
4. Zarządzanie Historyjkami Użytkownika.
   1. Dodawanie Historyjki Użytkownika.
      1. Definiowanie nazwy Historyjki Użytkownika.
      2. Przypisanie Historyjki Użytkownika do Sprintu.
      3. Definiowanie opisu Historyjki Użytkownika.
   2. Edytowanie Historyjki Użytkownika.
      1. Edytowanie nazwy Historyjki Użytkownika.
      2. Przypisanie Historyjki Użytkownika do innego Sprintu.
      3. Edytowanie opisu Historyjki Użytkownika.
   3. Usuwanie Historyjki Użytkownika.
   4. Wyświetlanie wszystkich Historyjek Użytkownika.
      1. Filtrowanie Historyjek Użytkownika po nazwie.
5. Zarządzanie Zadaniami.
   1. Dodawanie Zadań.
      1. Definiowanie nazwy Zadania.
      2. Przypisanie Zadania do Historyjki Użytkownika.
      3. Definiowanie opisu Zadania.
   2. Edytowanie Zadań.
      1. Edytowanie nazwy Zadania.
      2. Przypisanie Zadania do innej Historyjki Użytkownika.
      3. Edytowanie opisu Zadania.
   3. Usuwanie Zadania.
   4. Wyświetlanie wszystkich Zadań.
      1. Filtrowanie Zadań po nazwie.
6. Zarządzanie Przypadkami Testowymi.
   1. Dodawanie Przypadków Testowych.
      1. Definiowanie nazwy Przypadku Testowego.
      2. Przypisanie Przypadku Testowego do Historyjki Użytkownika.
      3. Definiowanie opisu Przypadku Testowego.
   2. Edytowanie Przypadków Testowych.
      1. Edytowanie nazwy Przypadku Testowego.
      2. Przypisanie Przypadku Testowego do innej Historyjki Użytkownika.
      3. Edytowanie opisu Przypadku Testowego.
   3. Usuwanie Przypadku Testowego.
   4. Wyświetlanie wszystkich Przypadków Testowych.
      1. Filtrowanie Przypadków Testowych po nazwie.
7. Zarządzanie Błędami.
   1. Dodawanie Błędów.
      1. Definiowanie nazwy Błędu.
      2. Przypisanie Błędu do Historyjki Użytkownika.
      3. Definiowanie opisu Błędu.
   2. Edytowanie Błędów.
      1. Edytowanie nazwy Błędu.
      2. Przypisanie Błędu do innej Historyjki Użytkownika.
      3. Edytowanie opisu Błędu.
   3. Usuwanie Błędu.
   4. Wyświetlanie wszystkich Błędów.
      1. Filtrowanie Błędów po nazwie.
8. Zarządzanie Przeszkodami.
   1. Dodawanie Przeszkód.
      1. Definiowanie nazwy Przeszkody.
      2. Przypisanie Przeszkody do Projektu.
      3. Definiowanie opisu Przeszkody.
   2. Edytowanie Przeszkody.
      1. Edytowanie nazwy Przeszkody.
      2. Edytowanie opisu Przeszkody.
   3. Usuwanie Przeszkody.
   4. Wyświetlanie wszystkich Przeszkód.
9. Zarządzanie Użytkownikami.
   1. Dodawanie Użytkownika.
      1. Definiowanie nazwy Użytkownika.
      2. Przypisanie Użytkownika do Projektu.
      3. Definicja Roli Użytkownika w danym Projekcie.
      4. Definiowanie opisu Użytkownika.
   2. Edytowanie Użytkownika.
      1. Edytowanie nazwy Użytkownika.
      2. Przypisanie Użytkownika do innego Projektu.
      3. Edycja Roli Użytkownika w danym Projekcie.
      4. Edycja opisu Użytkownika.
   3. Usuwanie Użytkownika.
   4. Wyświetlanie wszystkich Użytkowników.
10. Zarządzanie stanem Historyjek Użytkownika.
    1. Zmiana Business Value.
    2. Zmiana Statusu z "Backlog" na "Planned".
    3. Zmiana Statusu z "Planned" na "In Progress".
    4. Zmiana Complexity Points.
    5. Zmiana Statusu z "In Progress" na "In Testing".
    6. Zmiana Statusu z "In Testing" na "Done".
11. Zarządzanie stanem Zadań.
    1. Zmiana Complexity Points.
    2. Zmiana Statusu z "Planned" na "In Progress".
    3. Zmiana Statusu z "In Progress" na "Done".
12. Zarządzanie stanem Przypadków Testowych.
    1. Zmiana Statusu z "Planned" na "Passed".
    2. Zmiana Statusu z "Planned" na "Failed".
    3. Zmiana Statusu z "Failed" na "Passed".
13. Zarządzanie stanem Błędów.
    1. Zmiana Business Value.
    2. Zmiana Statusu z "Backlog" na "Planned".
    3. Zmiana Statusu z "Planned" na "In Progress".
    4. Zmiana Complexity Points.
    5. Zmiana Statusu z "In Progress" na "In Testing".
    6. Zmiana Statusu z "In Testing" na "Done".
14. Dodawanie Komentarzy.
    1. Dodawanie Komentarza do Historyjki Użytkownika.
    2. Dodawanie Komentarza do Zadania.
    3. Dodawanie Komentarza do Błędu.
    4. Dodawanie Komentarza do Przypadku Testowego.
15. Zarządzanie Tablicą Zadań.
    1. Wyświetlanie Tablicy Zadań.
    2. Przejście do strony danej Historyjki Użytkownika.
    3. Przejście do strony danego Zadania.
    4. Przejście do strony danego Przypadku Testowego.
    5. Przejście do strony danego Błędu.
16. Zarządzanie Tablicą Scrum'a.
    1. Wyświetlanie Tablicy Scrum'a.
    2. Zarządzanie stanem Historyjki Użytkownika.
    3. Zarządzanie stanem Błędu.
17. Integracja z kalendarzem Google.
    1. Dodawanie wydarzeń w Kalendarzu.
    2. Synchronizacja wydarzeń w Kalendarzu.
    3. Aktualizacja wydarzeń w Kalendarzu.
18. Integracja z GitHub.
    1. Synchronizacja Zmian w repozytorium.
       1. Dodawanie Zmian z repozytorium.
       2. Aktualizacja Zmian z repozytorium.
    2. Przejście do strony danej Historyjki Użytkownika.
       1. Wyświetlenie Zmian danej Historyjki Użytkownika.
    3. Przejście do strony danego Zadania.
       1. Wyświetlenie Zmian danego Zadania.
    4. Przejście do strony danego Historyjki Użytkownika.
       1. Wyświetlenie Zmian danego Historyjki Użytkownika.
19. Integracja z dokumentami Google.
    1. Wygenerowanie raportu Punktów Złożoności przypadających na każdego z członków zespołu.
    2. Wygenerowanie raportu Punktów Złożoności przypadających na każdy Sprint.
    3. Wygenerowanie raportu Efektywności oszacowań złożoności Sprintów.

## 2.2. Opis przypadków użycia

### 2.2.1. Ogólny schemat przypadków użycia



### 2.2.2. Zarządzanie Projektami

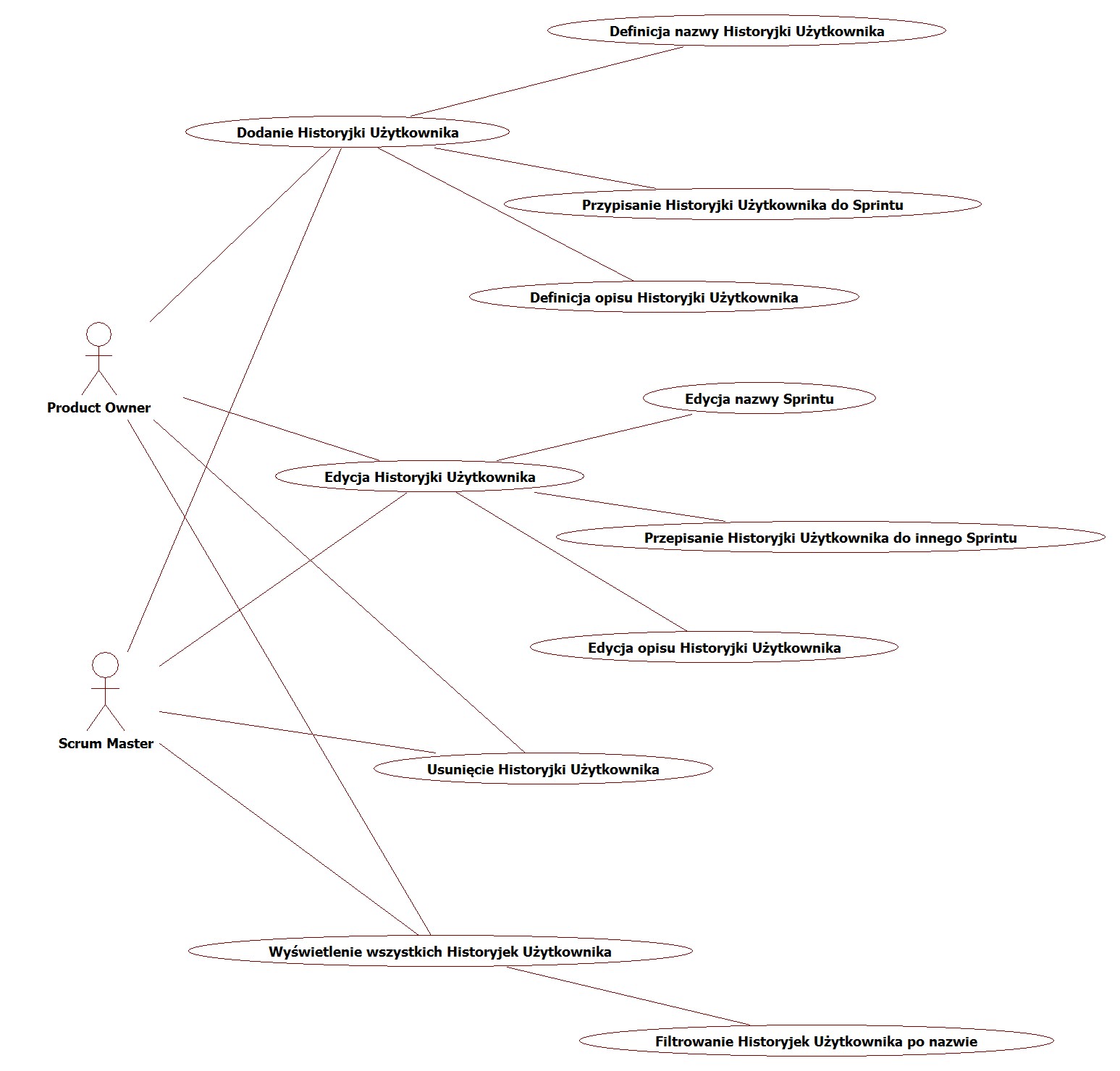
# project_management.jpg

### 2.2.3. Zarządzanie Wydaniami

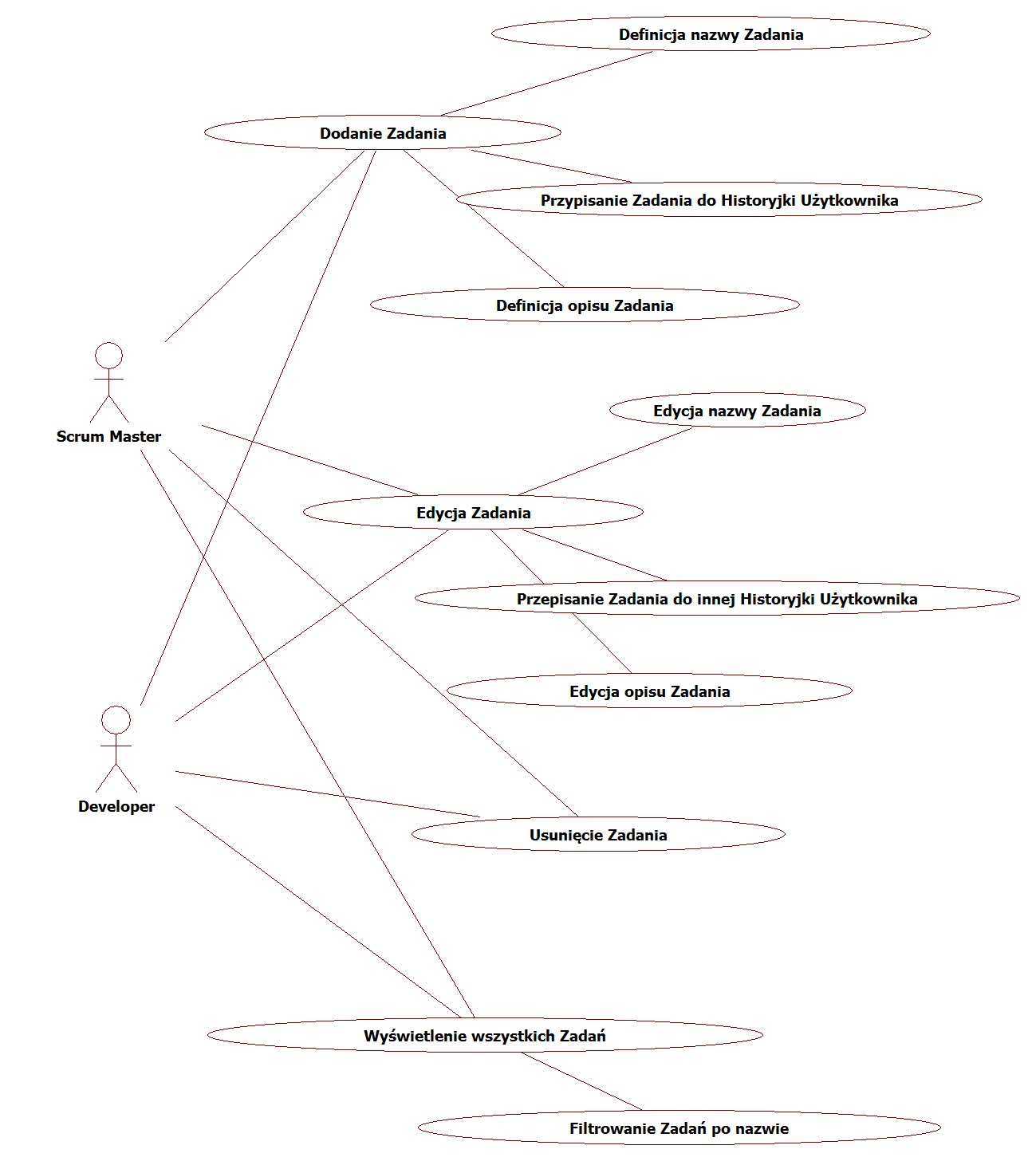
### release_management.jpg2.2.4. Zarządzanie Sprintami

## sprint_management.jpg

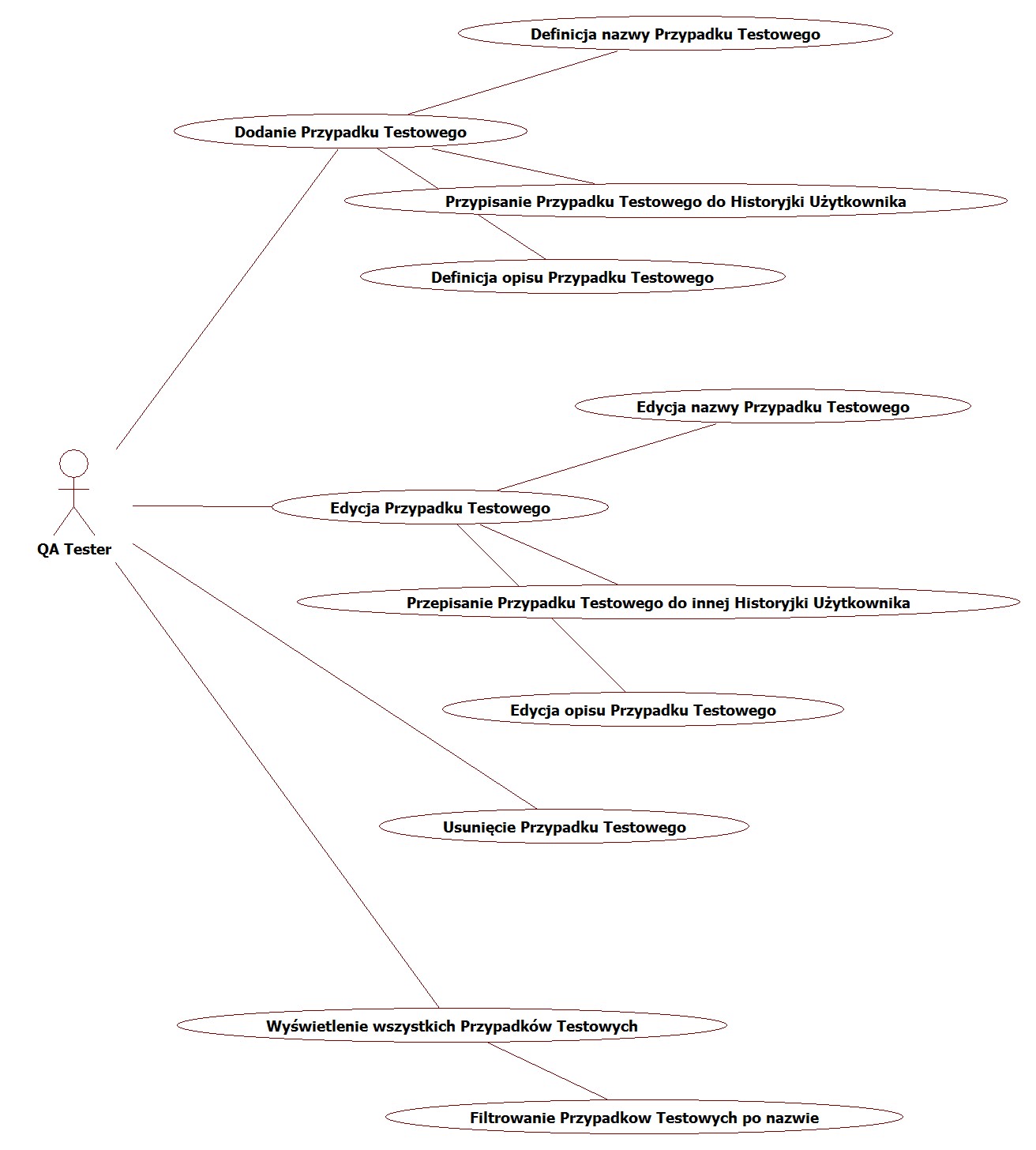
### 2.2.5. Zarządzanie Historyjkami Użytkownika



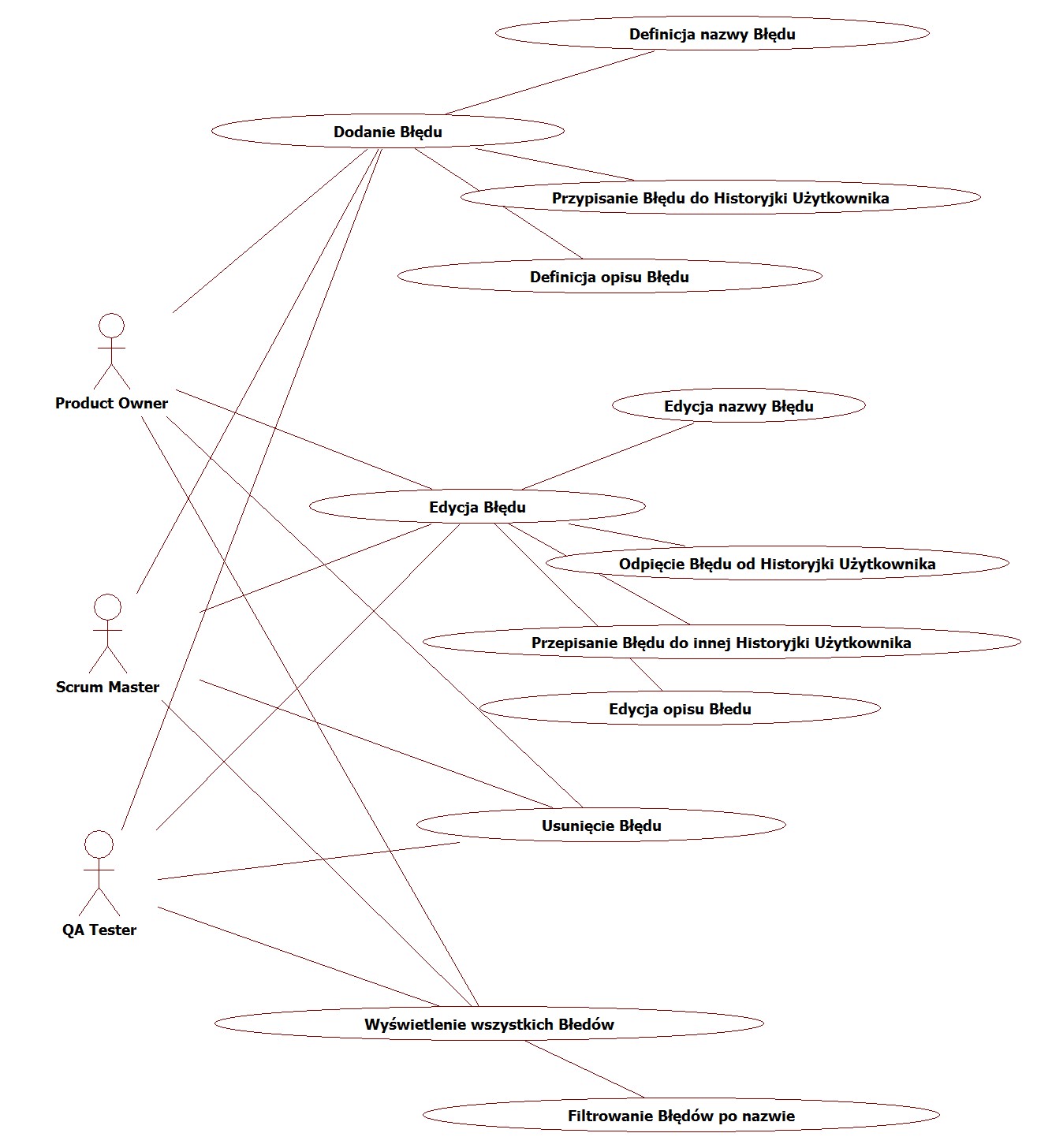
### 2.2.6. Zarządzanie Zadaniami



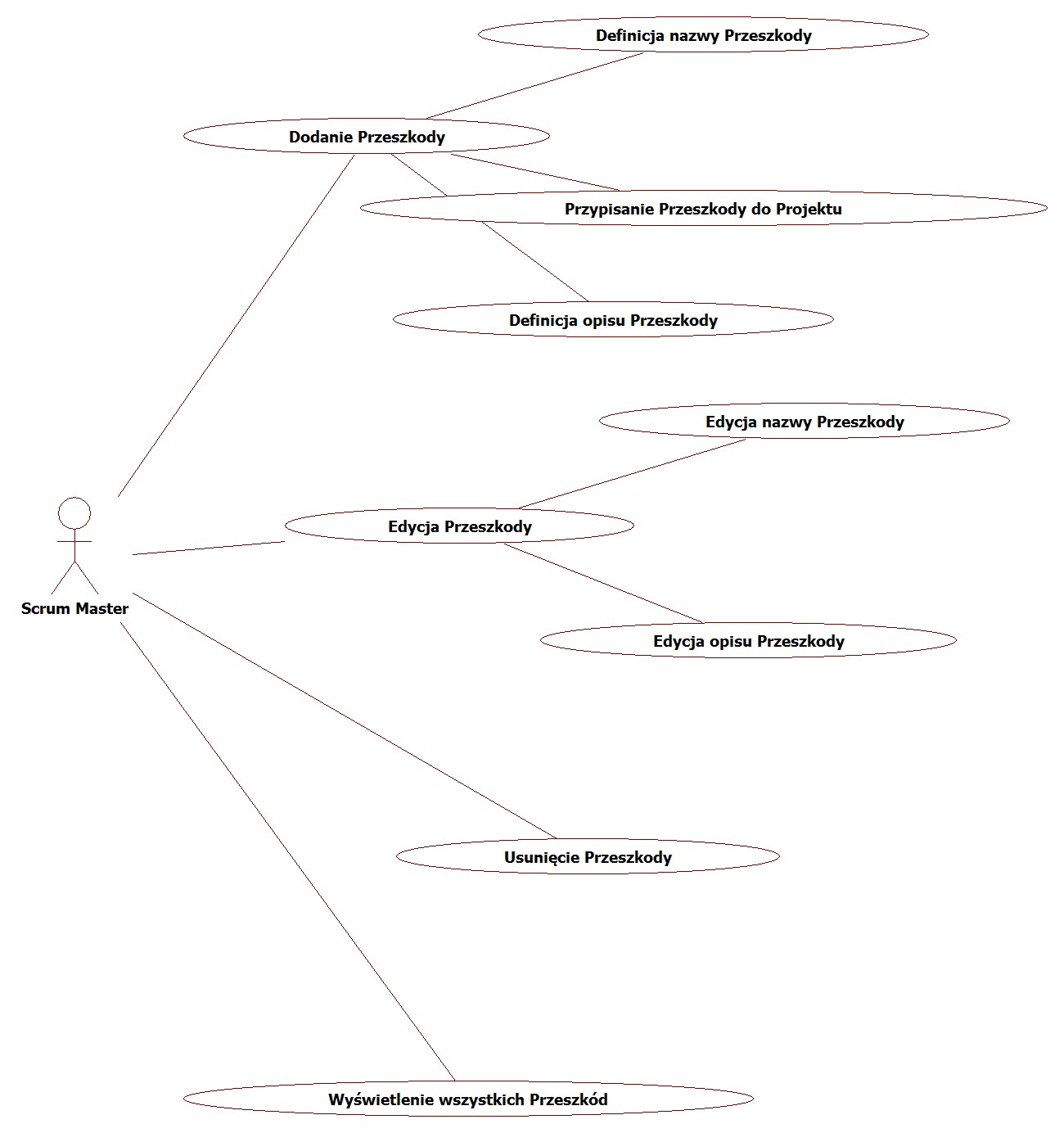
### 2.2.7. Zarządzanie Przypadkami Testowymi



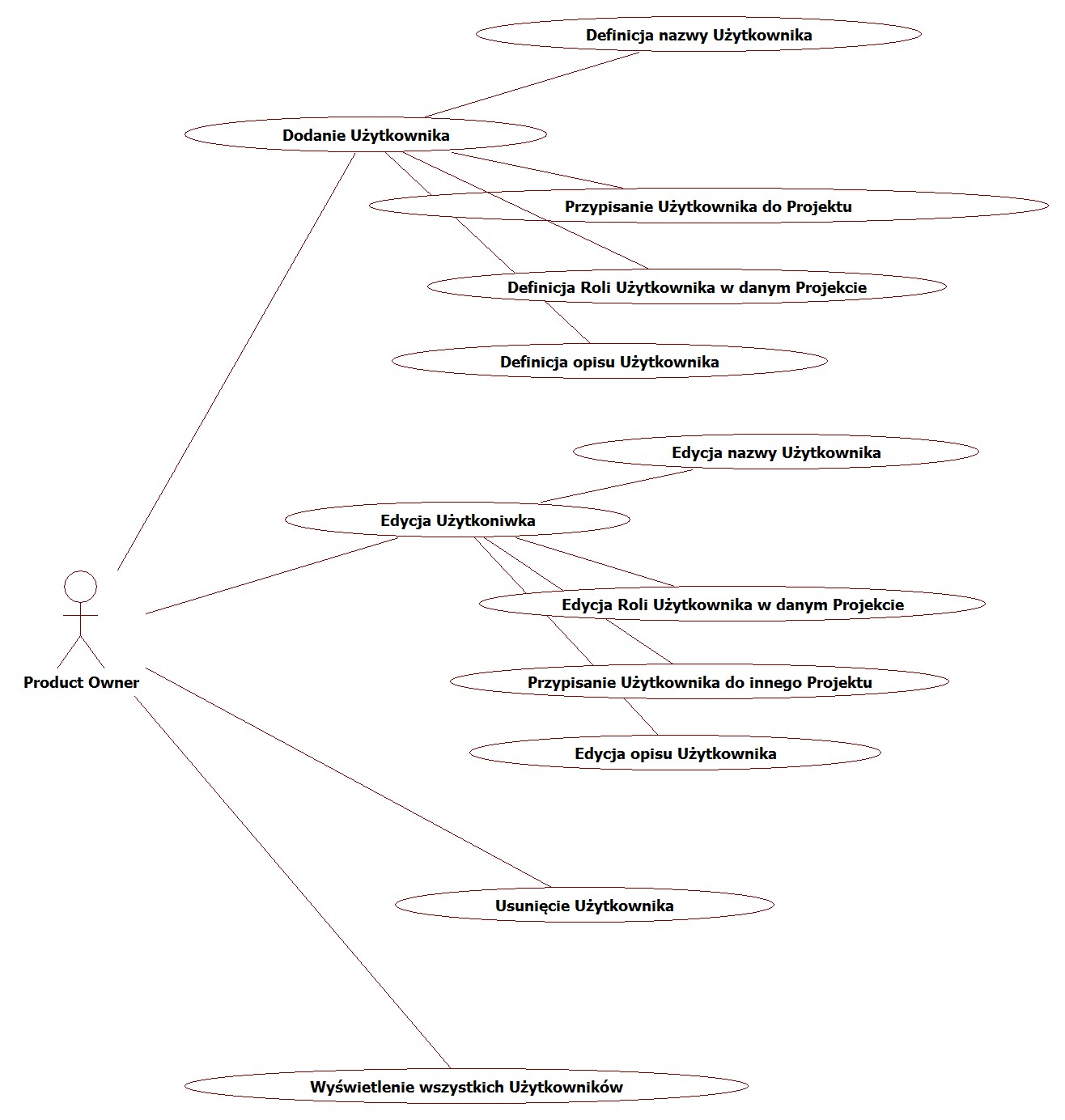
### 2.2.8. Zarządzanie Błędami



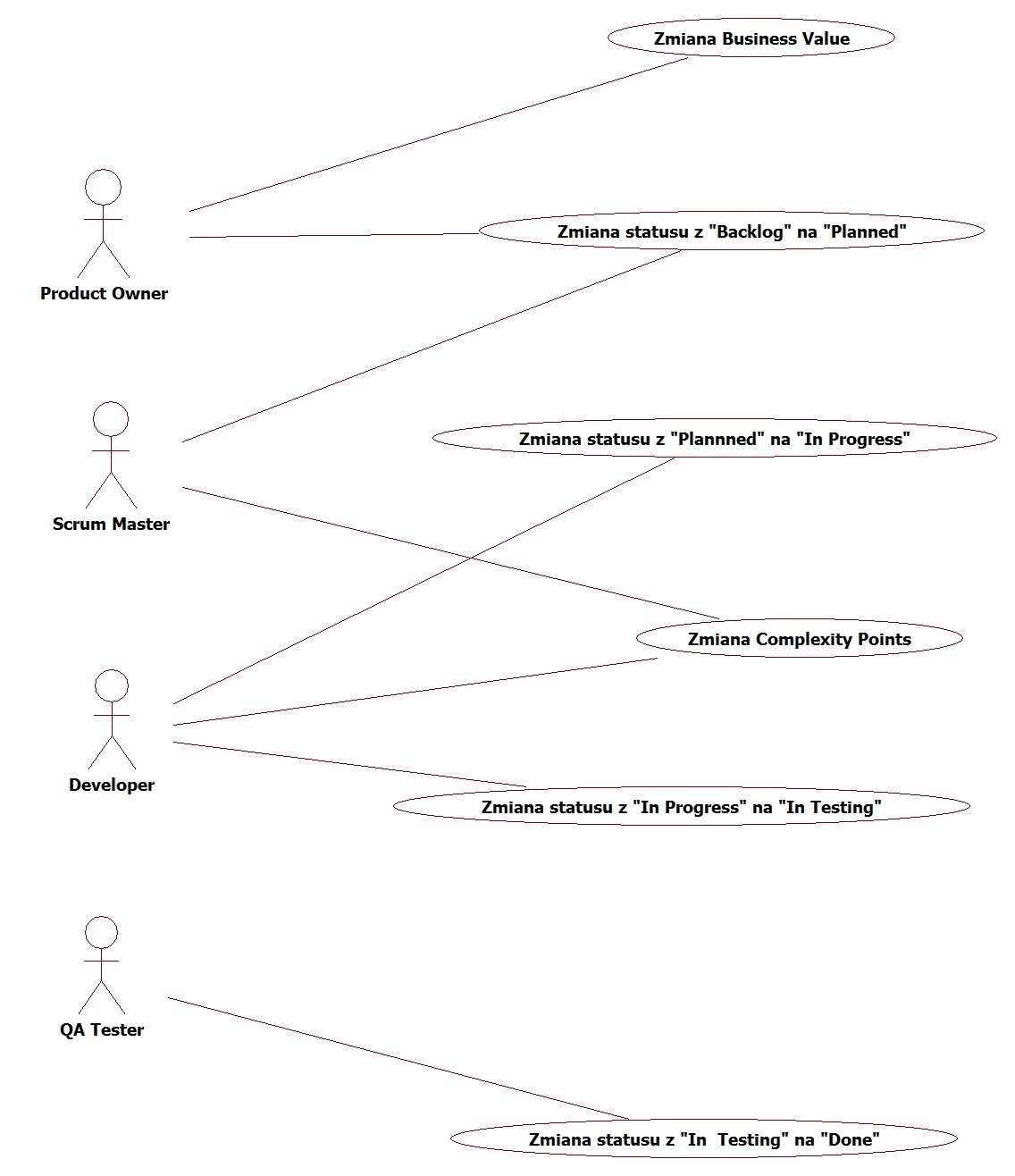
### 2.2.9. Zarządzanie Przeszkodami



### 2.2.10. Zarządzanie Użytkownikami



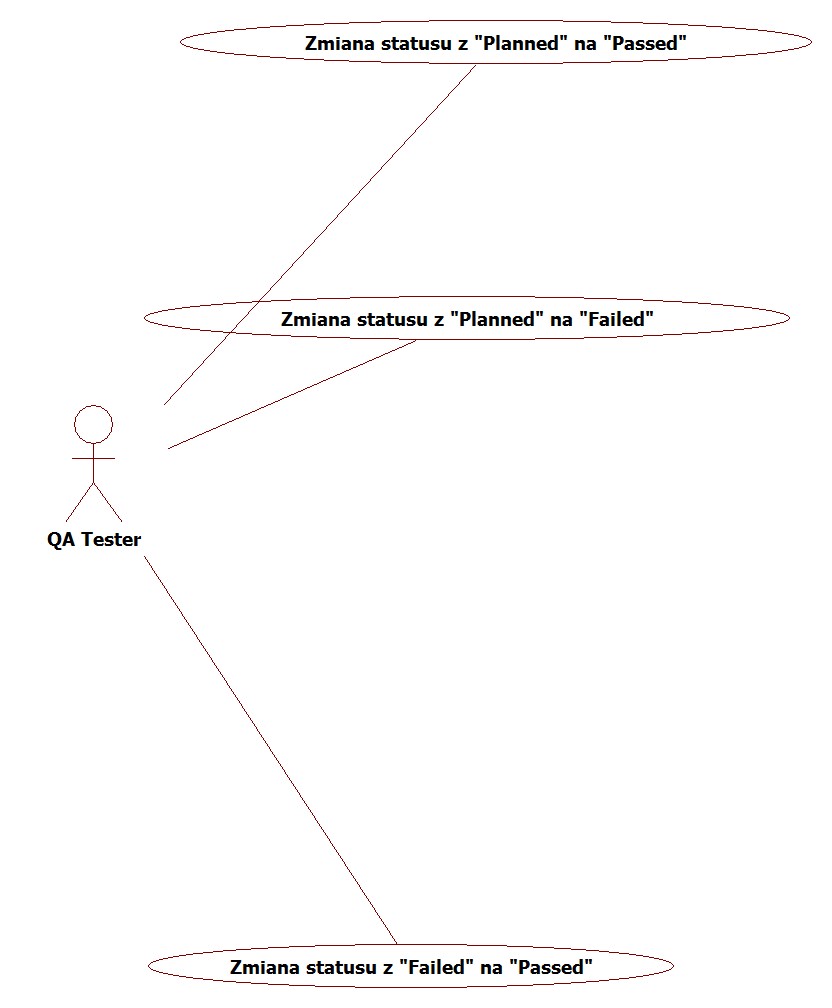
### 2.2.11. Zarządzanie stanem Historyjek Użytkownika



### 2.2.12. Zarządzanie stanem Zadań

## task_state_management.jpg

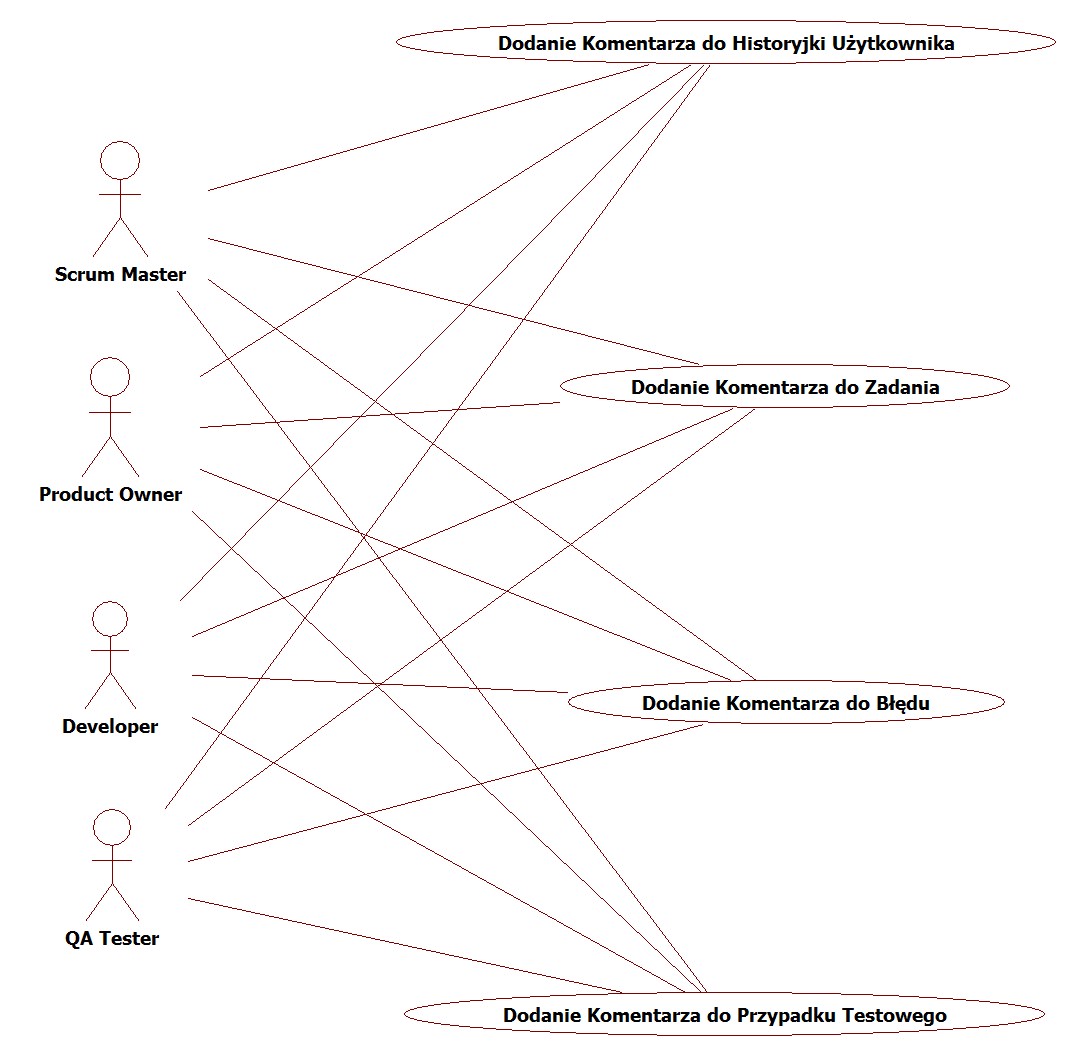
### 2.2.13. Zarządzanie stanem Przypadków Testowych.



### 2.2.14. Zarządzanie stanem Błędów

## bug_state_management.jpg

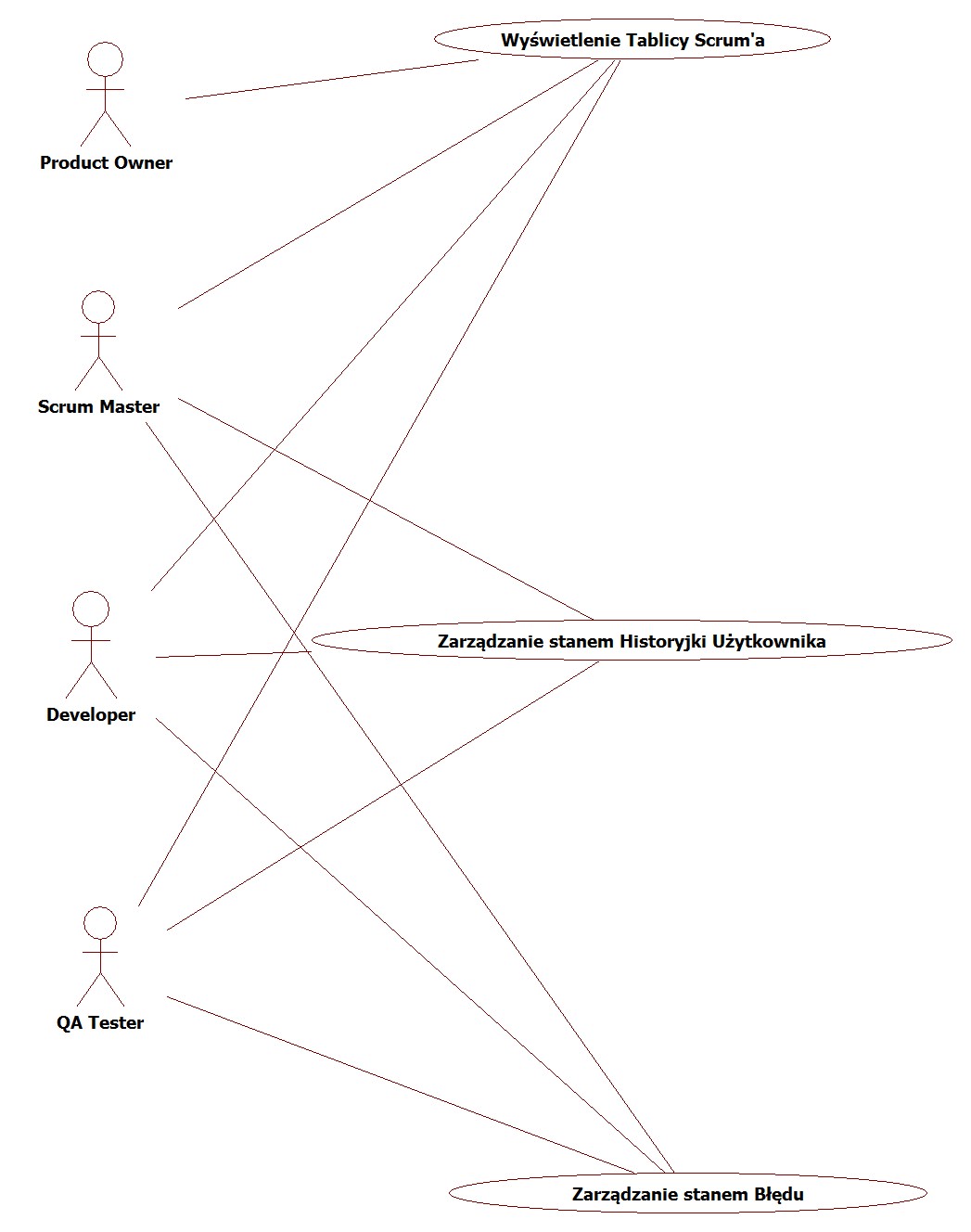
### 2.2.15. Dodawanie Komentarzy



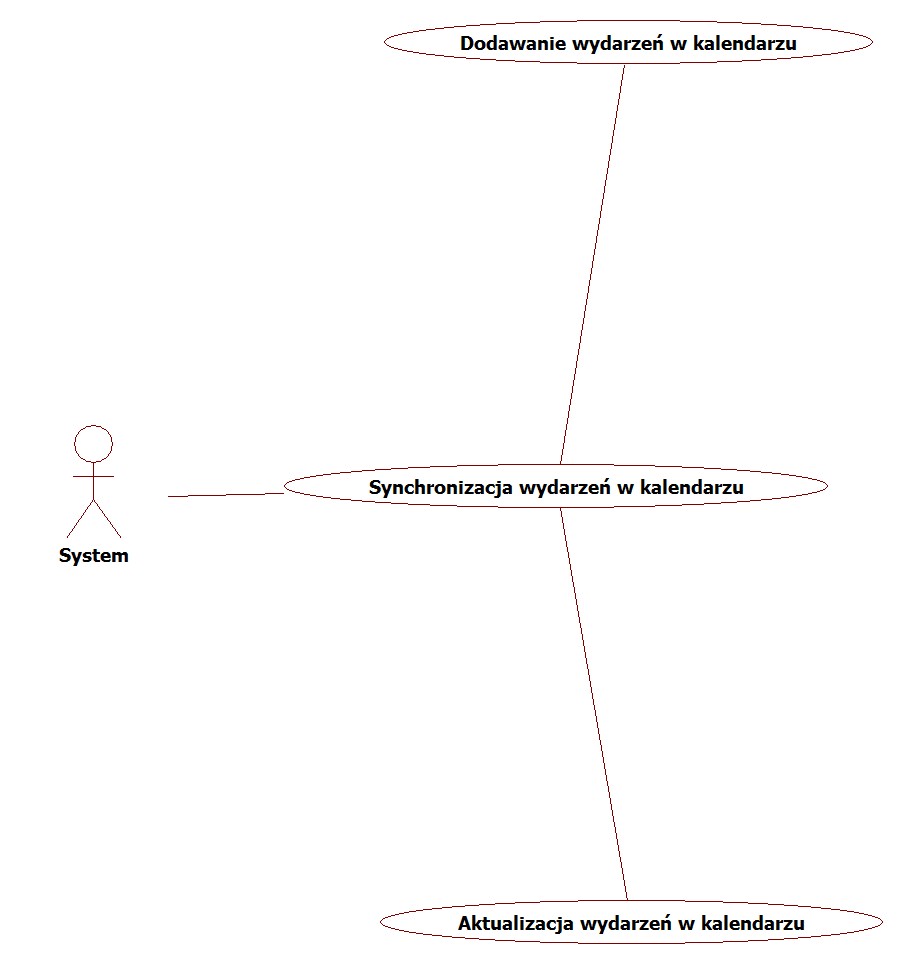
### 2.2.16. Zarządzanie Tablicą Zadań

## dashboard.jpg

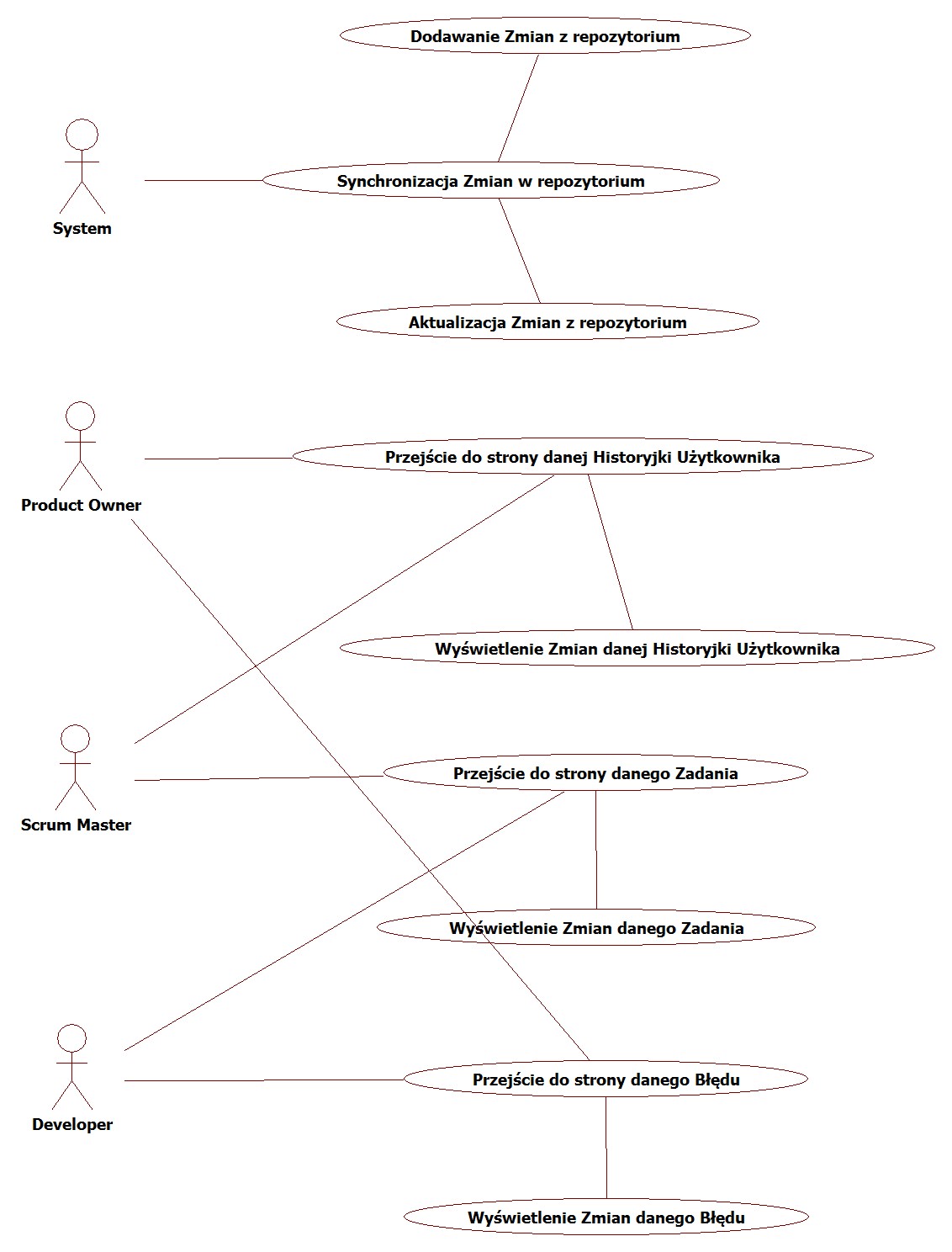
### 2.2.16. Zarządzanie Tablicą Scrum'u

****

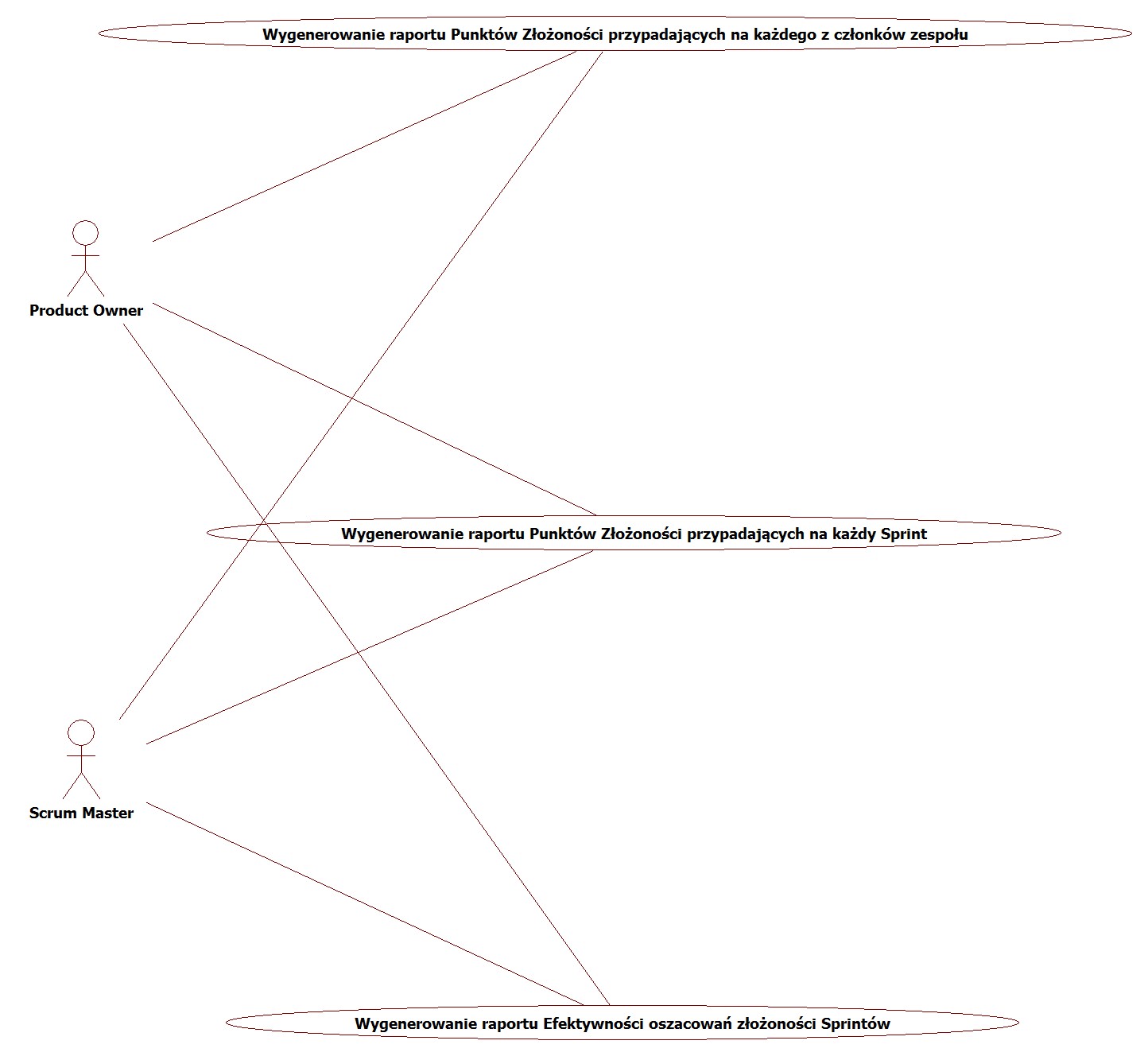
### 2.2.17. Integracja z kalendarzem Google



### 2.2.18. Integracja z GitHub



### 2.2.19. Integracja z dokumentami Google



# 3. Opis wymagań niefunkcjonalnych

1. System jest aplikacją typu Web.
2. Praca z Systemem musi być możliwa w środowiskach Microsoft Windows (od wersji XP wzwyż) oraz Linux, w następujących przeglądarkach Internetowych:
   1. Internet Explorer (wersja 8.0+).
   2. Mozilla Firefox (wersja 18.0+).
   3. Google Chrome (wersja 22.0+).
   4. Opera (wersja 11.0+).
   5. Safari (wersja 5.0+).
3. System powinien być wykonany w technologii Java Enterprise Edition z użyciem:
   1. bibliotek JEE w wersji 6 w specyfikacji JSR-316,
   2. serwera aplikacyjnego Tomcat w wersji 7.0.27,
   3. kontener do osadzania portletów Liferay w wersji 6.1.1 CE GA2,
   4. bibliotek GWT oraz Vaadin 7.0.4 do przygotowania interfejsu graficznego,
   5. bibliotek Google APIs do realizacji integracji z usługami kalendarza oraz dokumentów,
   6. baza danych PostgreSQL w wersji 9.2.4.
4. System ma działać w reżimie 24 godziny przez 7 dni w tygodniu każdego dnia w roku. Jego dostępność w godzinach pracy, czyli 8-18 w dni powszednie, ma wynosić 99,45% (20h/rok), a poza tymi godzinami 98,63% (5 dni/rok). Wartości orientacyjne zostaną zmierzone w 240-godzinnym teście akceptacyjnym z obciążeniem symulowanym 200 użytkowników wykonujących po ok. 1 operację zapisu/odczytu na minutę (200 operacji zapisu/odczytu na minutę) z różną częstotliwością.
5. System powinien realizować politykę bezpieczeństwa wykorzystującą szyfrowane połączenie z witryną przy pomocy protokołu HTTP SSL.
6. Czas odpowiedzi systemu na szczegółowe zapytanie, powinien średnio wynosić 3 sekundy w przypadku pracy zdalnej przez Internet. 98% zapytań powinno spełniać powyższe wymaganie. Mierzone będą wartości realne, od momentu wysłania żądania, do momentu uzyskania odpowiedzi i przeprowadzane w warunkach: 10 Mbps dla łącza internetowego, przy czym zakłada się, że nie ma ograniczeń co do przepustowości łącza na serwerze zdalnym.
7. System powinien być w stanie obsłużyć ok. 5000 Użytkowników jednocześnie.
8. Maksymalny czas przywrócenia Systemu po awarii powinien wynosić 1 godzinę.
9. System powinien utrzymywać kopię roboczą danych w nim zawartych. Awaria Systemu nie może prowadzić do utraty danych.
10. System musi posiadać przejrzysty, przyjazny Użytkownikowi interfejs graficzny oraz wspierać wykonywanie operacji w technice "przeciągnij i upuść" (ang. drag and drop).