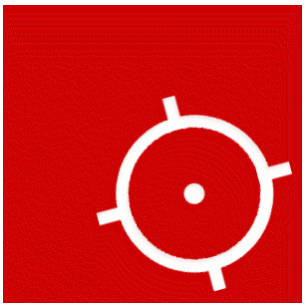

testerzy.pl

# Projektowanie przypadków testowych w praktyce

## Część 1 – PARTYCJE RÓWNOWAŻNE

Poniższe opracowanie skupia się na tworzeniu przypadków testowych oraz daje odpowiedź na pytanie jak tworzyć je w sposób wydajny?

**Partycje równoważne**

**Analiza**

Dane wejściowe do oprogramowania lub systemu są podzielone na grupy, w przypadku wystąpienia, których, można zdefiniować oczekiwane, podobne zachowania systemu. Równoważne partycje (lub klasy) mogą być określone dla poprawnych i niepoprawnych danych (tzn. takich danych, które powinny być odrzucone). Partycje mogą zostać również zidentyfikowane dla danych wyjściowych, wewnętrznych wartości, wartości powiązanych z czasem (np. przed lub po zdarzeniu) i dla parametrów interfejsu (np. podczas testów integracji). Testy są zaprojektowane w celu jak najpełniejszego pokrycia partycji. Technikę tą używa się na wszystkich poziomach testowych. Określa się je jako techniki specyfikacyjne lub czarnoskrzynkowe.

Równoważna partycja jako technika może być użyta dla osiągnięcia pokrycia wejść i wyjść. Może zostać zastosowane dla danych wprowadzanych przez użytkownika, danych wejściowych podawanych poprzez interfejsy do systemu, lub parametry interfejsów podczas testowania integracji.

**Projektowanie**

Przypadek testowy powinien zostać zaprojektowany tak, by sprawdzić partycję. Jeden przypadek może przetestować dowolną ilość partycji. Powinien on uwzględniać:
- dane wejściowe
- sprawdzaną partycję
- oczekiwany wynik.

**Przykład**

Program określa ocenę bazując na liczbie punktów zdobytych w teście. Maksymalna ilość punktów, którą oznaczymy jako *i* wynosi 75.
Gdy

- *I* jest większe równe 70, wtedy ocena wynosi 5
- *I* jest większe, równe 50 oraz jest mniejsze od 70, wtedy ocena wynosi 4
- *I* jest większe, równe 30 oraz jest mniejsze od 50, wtedy ocena wynosi 3
- *I* jest większe, równe 0 oraz jest mniejsze od 30, wtedy ocena wynosi 2

Dla

*I* mniejszego od 0 oraz większego od 75, powinien zostać wyświetlony komunikat błędu.

Obszar poprawnych danych w naszym przypadku wynosi <0;75> (nawias zamknięty oznacza, że zarówno 0 jak i 75 należą do obszaru zainteresowania). Szczegółowe partycje to:

- <70;75>
- <50;70) – nawias otwarty oznacza, że z do tego obszaru nie należy liczba 70
- <30;50)

- <0;30)

Istnieją cztery obszary danych wejściowych wymuszających zachowanie obiektu testowego. Obszar niepoprawnych danych, których pojawienie się na wejściu obiektu skutkować powinno wyświetleniem komunikatu błędu (-∞;0) oraz (75;+ ∞).

Założenia: ocena jest liczbą całkowitą, tak, więc wystąpienie a wyjściu litery, liczby ułamkowej, znaku specjalnego (np. %, $, #) jest oznaką defektu.
Zdefiniowanych pięć możliwych danych wejściowych:
- 5, 4, 3, 2
- komunikat błędu.

Przy poprawnych danych wejściowych inne wartości pojawiające się na wyjściu są oznaką defektów. Pojawienie się na wyjściu wartości takich jak 1, 5+ czy „zero", choć mogą się wydawać poprawne (bazujące na doświadczeniu) muszą zostać zinterpretowane jako błąd.

Po określeniu równoważnych partycji możemy przystąpić do tworzenia przypadków testowych sprawdzającą każdą z nich.

| **Przypadek testowy** | **1** | **2** | **3** | **4** | **5** |
|---|---|---|---|---|---|
| **Dane wejściowe** | 72 | 60 | 33 | 3 | -20 |
| **Testowana partycja** | 75≥/≥70 | 70>/≥50 | 50>/≥30 | 30>/≥0 | /<0 |
| **Dana wyjściowa** | 5 | 4 | 3 | 2 | Komunikat błędnej danej wejściowej |
| **Oczekiwany wynik** | 5 | 4 | 3 | 2 | Komunikat błędnej danej wejściowej |

| **Przypadek testowy** | **6** | **7** | **8** | **9** | **10** | **11** |
|---|---|---|---|---|---|---|
| **Dane wejściowe** | 110 | 56,53 | 80 | -10 | „zero" | W |
| **Testowana partycja** | />100 | 70>/≥50 | />100 | /<0 | - | - |
| **Dana wyjściowa** | Komunikat błędnej danej wejściowej | Komunikat błędnej danej wejściowej | 5+ | 1 | „zero" | Komunikat błędnej danej wejściowej |
| **Oczekiwany wynik** | Komunikat błędnej danej wejściowej | Komunikat błędnej danej wejściowej | Komunikat błędnej danej wejściowej | Komunikat błędnej danej wejściowej | Komunikat błędnej danej wejściowej | Komunikat błędnej danej wejściowej |

Za pomocą zestawu (minimalnej ilości) przypadków testowych można przetestować wszystkie partycje, uzyskując 100% ich pokrycia. W przypadku, gdy pokrycie jest niższe oznacza to, że nie przetestowano wszystkich zidentyfikowanej partycji. Gdy nie uda się zidentyfikować wszystkich partycji, statystyki pokrycia mogą być mylące.
Może się okazać, że wymagania przekazane testerowi nie będą wspominały o dodatkowych granicach narzuconych w fazie programowania. Przykładowo wybierając w C/C++ ocenę jako Int (typ liczbowy, stałoprzecinkowy), którego rozmiar zawsze zależy od kompilatora i platformy. Ten dodatkowy rozmiar nie zostanie, w wielu przypadkach, opisany dodatkowym przypadkiem testowym,

**Opracowanie oparte na:**


- **Standard for Component Testing – wydane przez BCS SIGITS**
- **Techniki projektowania testów – testerzy.pl**
- **A Practitioner's Guide to Software Test Design – autor Copeland, Lee, wydany przez: Artech House Publishers**