

Projektowanie przypadków testowych w praktyce

Część 2 – ANALIZA WARUNKÓW GRANICZNYCH

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

Analiza warunków granicznych

Analiza

W rozdziale poświęconym ekwiwalentnym partycjom niejako podzieliśmy testowany obiekt na kilka obszarów, w zależności od danych wejściowych i/lub wyjściowych. Dane te określa się na podstawie specyficznych zachowań obiektu.

Warunki graniczne sprawdzają natomiast jak zachowuje się obiekt na skraju tychże wydzielonych obszarów. Zarówno po stronie danych poprawnych, jak i niepoprawnych. Gdy dla danego obszaru nie jesteśmy w stanie ustalić innego graniczącego z nim obszaru, analizujemy jego wartości minimalne i/lub maksymalne.

Zachowanie na każdej z krawędzi partycji w wielu przypadkach jest niepoprawne, tak więc granice są obszarami gdzie testowanie może wykryć potencjalne defekty. Wartości minimalna i maksymalna partycji są jego wartościami granicznymi. Wartość graniczna dla poprawnej partycji jest poprawną wartością graniczną i odpowiednio dla niepoprawnej jest niepoprawną wartością graniczną. Testy mogą być zaprojektowane dla pokrycia obu wartości.

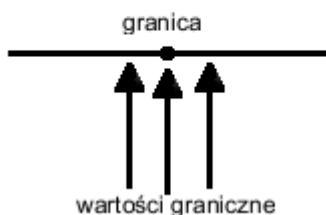
Analiza tego rodzaju może pojawić się na wszystkich poziomach testowych. Jest relatywnie prosta do zastosowania a jej zdolność do znajdowania defektów jest bardzo wysoka. Pomaga w tym szczegółowa specyfikacja.

Technika ta jest często postrzegana jako rozszerzenie równoważnych partycji i może zostać użyta w celu zdefiniowania danych wprowadzanych przez człowieka jak i dla czasów odpowiedzi czy granicznych wartości tabeli. Analiza ta wspiera również dobór danych testowych.

Projektowanie

Przypadek testowy powinien sprawdzać zarówno wartości na granicy jak i w jej najbliższym sąsiedztwie. Każda zidentyfikowana granica wymaga trzech przypadków testowych. Dana wejściowa musi:

- pokrywać się z wartością graniczną
- być mniejszą i najbliższą wartością od wartości granicznej
- być większą i najbliższą wartością od wartości granicznej.



Przypadek testowy powinien składać się z:

- dane wejściowe
- sprawdzana granica partycji
- oczekiwany rezultat.

Przykład

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

Gdy

/ jest większe równe 70, wtedy ocena wynosi 5

/ jest większe, równe 50 oraz jest mniejsze od 70, wtedy ocena wynosi 4

/ jest większe, równe 30 oraz jest mniejsze od 50, wtedy ocena wynosi 3

/ jest większe, równe 0 oraz jest mniejsze od 30, wtedy ocena wynosi 2

Dla

/ 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 ($-\infty;0$) oraz $(75;+\infty)$.

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 partycji możemy określić również wartości graniczne.

Przypadek testowy	1	2	3	4	5	6	7
Dane wejściowe	-1	0	1	29	30	31	49
Testowana granica	0	0	0	30	30	30	50
Oczekiwany rezultat	błąd	2	2	2	3	3	3

8	9	10	11	12	13	14	15
50	51	69	70	71	74	75	76
50	50	70	70	70	75	75	75
4	4	4	5	5	5	5	błąd

Piętnaście opisanych powyżej przypadków testowych pokrywa 100% wartości granicznych. W przypadku, gdy pokrycie jest niższe oznacza to, że nie przetestowano wszystkich zidentyfikowanych granic. Gdy nie uda się zidentyfikować wszystkich granic, 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

testerzy.pl