Magazine

Przypadek testowy. Teoria i praktyka

Autor: Radosław Smilgin

O autorze:

Radosław Smilgin jest trenerem i konsultantem z zakresu testowania oprogramowania. Materiał przedstawiony w tym artykule jest częścią oparty na materiałach szkoleniowych "Dobry Przypadek Testowy".

Basic
Level

1
Magazine Number

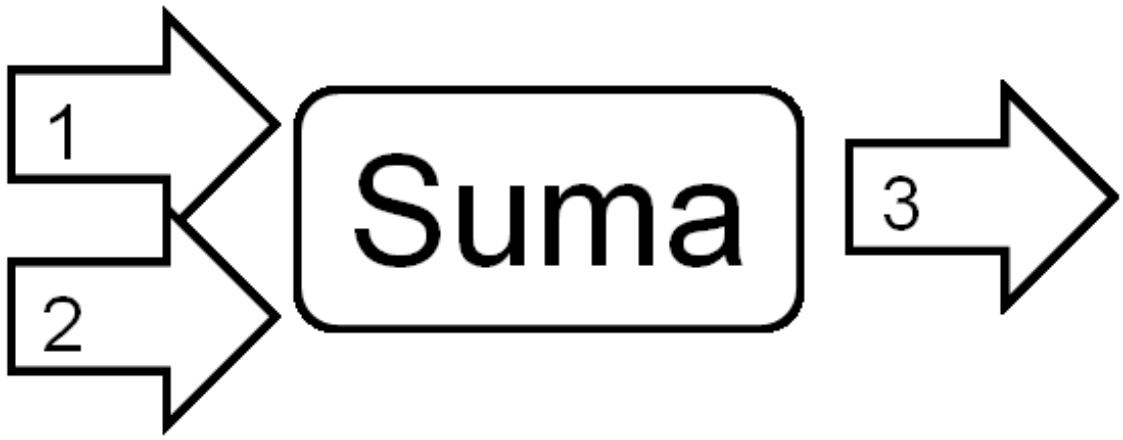
Software testing
Section in the magazine

Wprowadzenie
Managerowie patrzą na testowanie z góry. Widzą procesy, zasoby, możliwe straty i przychody. Ten punkt widzenia na szczęście dotyczy mniejszości. Większość z testerów widzi testowanie jako ciągłą pracę u podstaw. Toczymy bitwę o każdą dodatkową godzinę i każdy poprawiony defekt. Na tym polu bitwy kluczowa jest optymalizacja naszej pracy. Porozmawiajmy więc o przypadku testowym.

W swojej dotychczasowej pracy rozbijałem testowanie na najmniejsze składowe. Poświęciłem dużo energii i uwagi danym testowym. Czas jednak odejść od najniższego poziomu i zainteresować się przypadkiem testowym jako narzędziem wykorzystania danych testowych.

Przypadek testowy niskiego poziomu

Zacznijmy od analizy definicji przypadku testowego na najniższym jego poziomie. "Słownik wyrażeń związanych z testowaniem. Wersja 2.0" definiuje go jako: *"Przypadek testowy z konkretnymi wartościami wejściowymi i wynikami oczekiwanymi. Logiczne operatory z przypadków testowych wysokiego poziomu są zamieniane na konkretne wartości, które odpowiadają celom logicznych operatorów."*. Oznacza to, że przypadek testowy jest złożeniem danych wejściowych, pewnej operacji logicznej i danych wyjściowych będących oczekiwanym rezultatem.



Rysunek 1. Przykładowa wizualizacja przypadku testowego niskiego poziomu

Rysunek 1 prezentuje przykład przypadku testowego gdzie na test składają się dane wejściowe (1 i 2), operacja logiczna (suma) oraz dana wyjściowa (3). Komponent "Suma" może mieć dowolną implementację:

a) może być usługą sieciową (ang. web service), która w oparciu o dwie dane zwraca ich sumę

b) może być desktopowym kalkulatorem dostępnym z poziomu akcesoriów systemu operacyjnego Windows

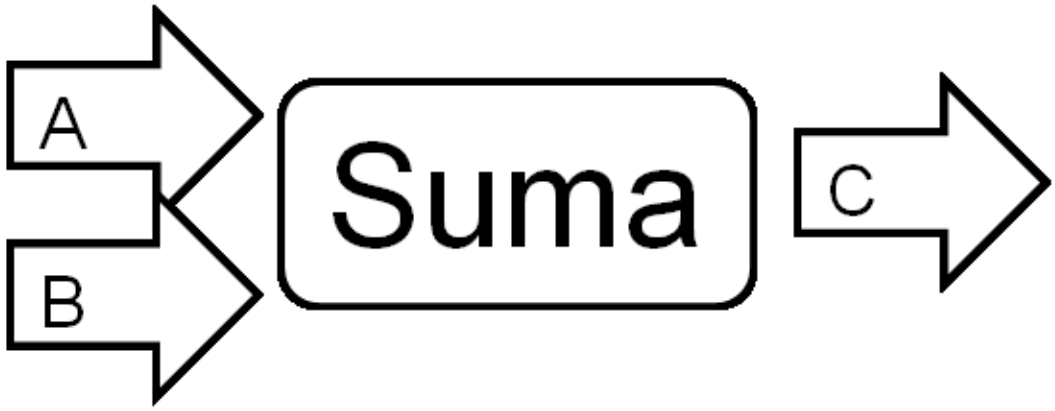
c) może być również systemem używanym w samolotach, który w oparciu o dwie zmienne pomiarowe przesyła do kolejnego komponentu ich sumę w celu wyliczenia np. średniej.

Dla każdego z powyższych rozwiązań przypadek testowy będzie wyglądał dokładnie tak samo, choć jego wykonania będzie miało zupełnie inną postać. Dla a) będzie to przesłanie danych w specjalnie spreparowanym formacie XML i odebranie podobnego XML-a z wartością dodawania, dla b) przez interfejs klawiatury lub GUI przekażemy dwie zmienne i wymusimy wyświetlenie na ekranie ich sumy, a dla c) przygotujemy przypadek jednostkowy, gdzie specjalnie spreparowany sterownik przekaże nasze zmienne do testowego komponentu.

Różne będą również zakresy wykonywanych testów. Systemom a) i b) prawdopodobnie wystarczą pojedyncze przypadki testowe dla danego komponentu. Więcej trzeba będzie przygotować danych dla systemu c), od którego zależeć może bezpieczeństwo lotu. Choć i tu jeden przypadek testowy będzie w zupełności wystarczający, to musi to być inny przypadek od tego opisanego na początku.

Przypadek testowy wysokiego poziomu

Przypadek wysokopoziomowy nie zawiera danych testowych, a jedynie pewną informację o operatorach logicznych. Zgodnie z definicją jest to *"przypadek testowy bez konkretnych wartości danych wejściowych i oczekiwanych rezultatów. "*



Rysunek 2. Przykładowa wizualizacja przypadku testowego wysokiego poziomu, gdzie A, B i C są parametrami pod które podstawia się konkretne dane testowe

Wracając do naszego komponentu sumującego, który zamontowany jest w samolocie, bardziej właściwy będzie przypadek zobrazowany na rysunku 2. Zakładamy, że potrzebujemy większej ilości danych, aby mieć pewność, że nie pojawi się awaria. Dla przypadku, który będzie brzmiał: *do komponentu "Suma" podaj dwie wartości i sprawdź czy otrzymany rezultat jest zgodny wytycznymi dla implementacji*. Wspomniane wytyczne mogą być wynikiem analizy wymagań i przedstawiać sie w formie np. tabeli. Przykładowa tabela 1.

Dana A	Dana B	Dana C	Komentarz
1	2	3	Suma
1,1	0,2	1	Wartości przecinkowe zaokrąglane są zawsze w dół do pełnej wartości całkowitej
-1	-2	0	Wartości ujemne traktowane są jako błędne dane wejściowe, komponent zwraca 0
257	1	0	Wartości większe i równe 257 traktowane są jako błędne dane wejściowe, komponent zwraca 0
...	...	...	...

Tabela 1. Przykładowa dane dla przypadku testowego wysokiego poziomu

Wracamy wiec znów do podstaw, czyli danych testowych, których odpowiedni dobór uzupełni przypadek wysokiego poziomu.

Przypadek w przykładzie

Przeanalizujmy prostą aplikacje z listami rozwijalnymi (rysunek 4) zwracającą ilość dni przy zadanym miesiącu i roku. Oczywiście przypadek jest jeden zmieniają się jedynie dane. Wybieramy te miesiące, które mają 30 i 31 dni oraz luty dla roku przestępnego i nieprzestępnego. Łącznie 4 zestawy danych. Zgodnie z teorią klas równoważności daje nam to 100% pokrycia klas miesięcy i lat.

Rysunek 4. Interfejs z listą wybieralną dla aplikacji zwracającej ilość dni miesiąca

Co jednak jeśli interfejs będzie miał pola tekstowe (rysunek 5).

Rysunek 5. Interfejs z polami tekstowymi dla aplikacji zwracającej ilość dni miesiąca

Przypadek się komplikuje. Jeśli założymy, że nie ma walidacji musimy sobie zadać wiele pytań odnośnie danych akceptowalnych. Nie ma większych wątpliwości odnośnie roku, jeśli uwzględnimy realne granice jego podawania np. z przedziału <-9999; 9999>. Trudność pojawia się z miesiącem. Które dane są rozpoznawalne przez system: "STYCZEŃ", "styczeń", "1", "01", "sty", czy może "I"? A może wszystkie? Pole tekstowe wymagają więc dodatkowej obsługi komunikatami błędów. Oprócz czterech podstawowych danych wejściowych musimy również przygotować dane dla każdego, możliwego komunikatu błędu.

Składowe przypadku testowego

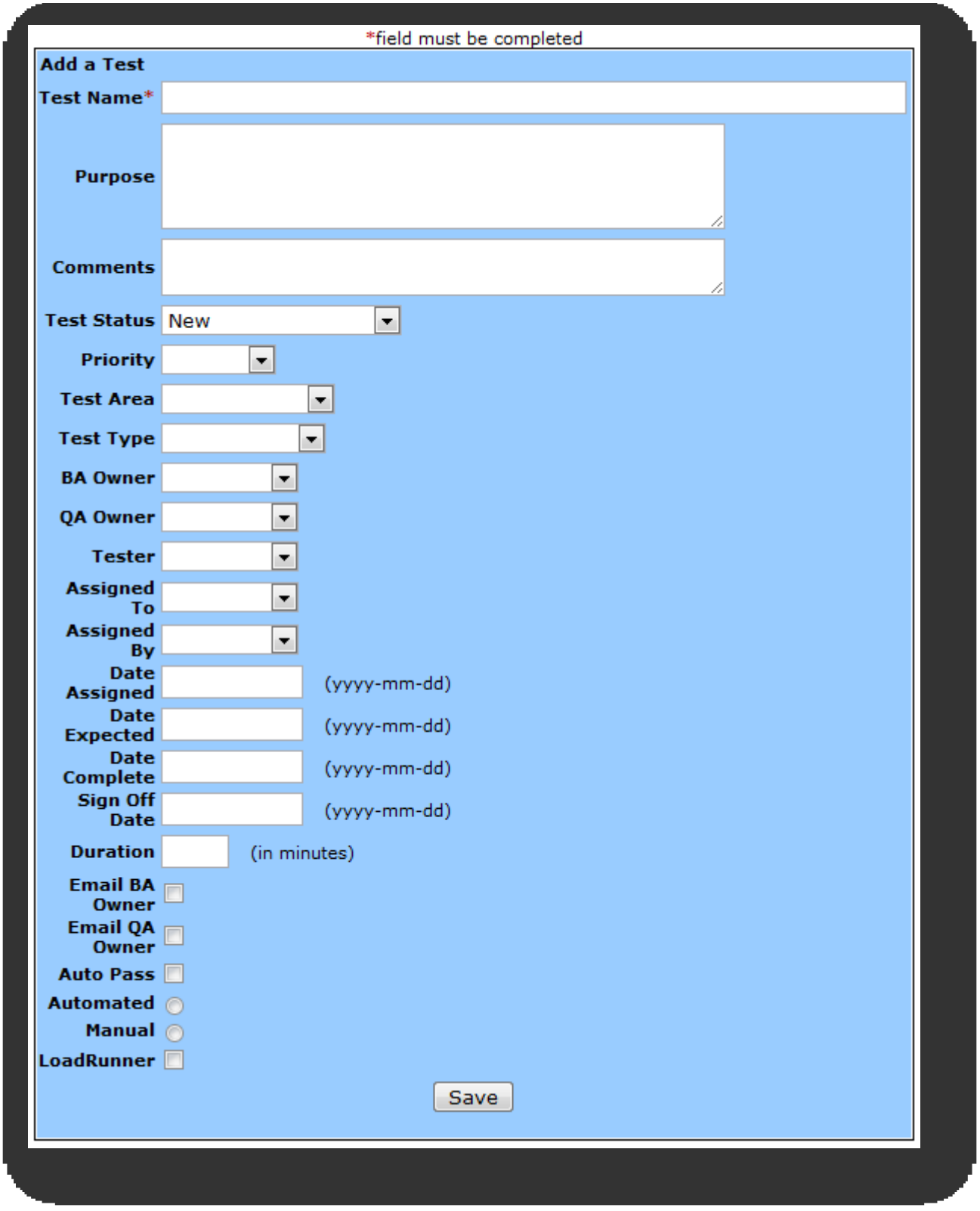
Wiemy, że na przypadek składają się dane wejściowe i wyjściowe, oraz funkcja, która te dane przetwarza. Co jeszcze tworzy pełny opis dobrego przypadku testowego?

Na pewno są to wstępne i końcowe warunki wykonania. Warunki wstępne umożliwiają nam odpowiednie przygotowanie środowiska do wykonania testu. Z kolei końcowe warunki umożliwiają nam przede wszystkim łączenie przypadków w scenariusze, gdzie informacja wyjściowa z jednego przypadku jest informacją wejściową dla kolejnego przypadku.

Od strony organizacyjnej przypadek testowy może być opisany również przez:

a) Unikalny identyfikator dzięki, któremu łatwo przypadek testowy zidentyfikować
b) Tytuł, który w skrócie opisuje co przypadek robi
c) Wersja, która pozwala nam śledzić zmiany i mapować przypadki na konkretne wymagania i implementacje
d) Priorytet, który w optymalnej sytuacji powinien odzwierciedlać ważność danej funkcjonalności dla klienta lub użytkownika
e) Autor, do którego zawsze się można zwrócić z prośbą o uszczegółowienie lub też interpretację
f) Podstawę testów, czyli dzięki czemu wiemy, jaki jest oczekiwany rezultat.

Oczywiście składowe przypadku testowego będą różnić się w zależności od specyfiki organizacji lub też narzędzia, które stosujemy do ich przechowywania.



Rysunek 3. Tworzenie przypadku testowego w narzędzi open source RTH

Dobry przypadek testowy

W podsumowaniu warto sobie powiedzieć jakie cechy powinien spełniać dobry przypadek testowy. Przypadek testowy ma cele pokrewne z samym testowaniem. Pierwszy cel to mierzyć jakość oprogramowania i pomagać stronie biznesowej podejmować decyzje odnośnie procesu wytwarzania. Drugi to wykrywanie błędów. Jeżeli wrócimy do militarnych porównań to dobry przypadek testowy jest bronią w walce o lepsze oprogramowanie, bez względu na to czy znajduje błędy, czy też nie. Cechować się musi skutecznością w znajdywaniu defektów i mieć zdolność mierzenia jakości oprogramowania, nawet gdy błędy nie zostaną wykryte.