

1. Praca z CREAM v.3.0.

Poniżej znajduje się instrukcja użytkownika narzędzia CREAM v.3.0. Proces instalacji i konfiguracji systemu opisany jest w rozdziale 1.1. W kolejnych podrozdziałach przedstawiony został sposób w jaki można wykorzystać funkcjonalności CREAM v.3.0.

1.1. Instalacja narzędzia CREAM i konfiguracja środowiska pracy

CREAM v.3.0. zintegrowany jest z zewnętrznymi programami, dzięki którym możliwe jest wykorzystanie pełnej funkcjonalności narzędzia. Zaleca się zainstalowanie ich przed przystąpieniem do procesu instalacji CREAMa. Należą do nich:

- **.NET Framework 4.0** [1]
- **Visual Studio** w jednej z poniższych wersji:
 - Visual Studio 2010 Premium.
 - Visual Studio 2010 Ultimate.
 - Visual Studio 2008 Team System.
- **NUnit v.2.5.7.**[2]
- **NCover v.3.4.14** [3]
- **Tester** [4]
- **SVN Client** [5]
- **Microsoft Excel 2007** lub wyżej

Pierwsze dwie pozycje z listy są niezbędne, kolejne natomiast opcjonalne – wybór zależy od tego z jakiej funkcjonalności narzędzia CREAM użytkownik będzie chciał korzystać.

Doinstalowanie dodatkowego oprogramowania może uwarunkowane być tym, że użytkownik chce korzystać z innego narzędzie do przeprowadzania testów jednostkowych, niż dostarczane wraz z Visual Studio. W narzędziu CREAM zaimplementowana jest obsługa testów jednostkowych przygotowanych z użyciem MSTest i NUnit.

Dodatkowo, jeżeli użytkownik chce korzystać z innego narzędzia (niż wybudowane w środowisko Visual Studio) do badania pokrycia kodu przez testy, ma do wyboru NCover lub Tester.

Klient SVN niezbędny jest, jeżeli użytkownik będzie chciał przechowywać pliki mutantów na serwerze SVN.

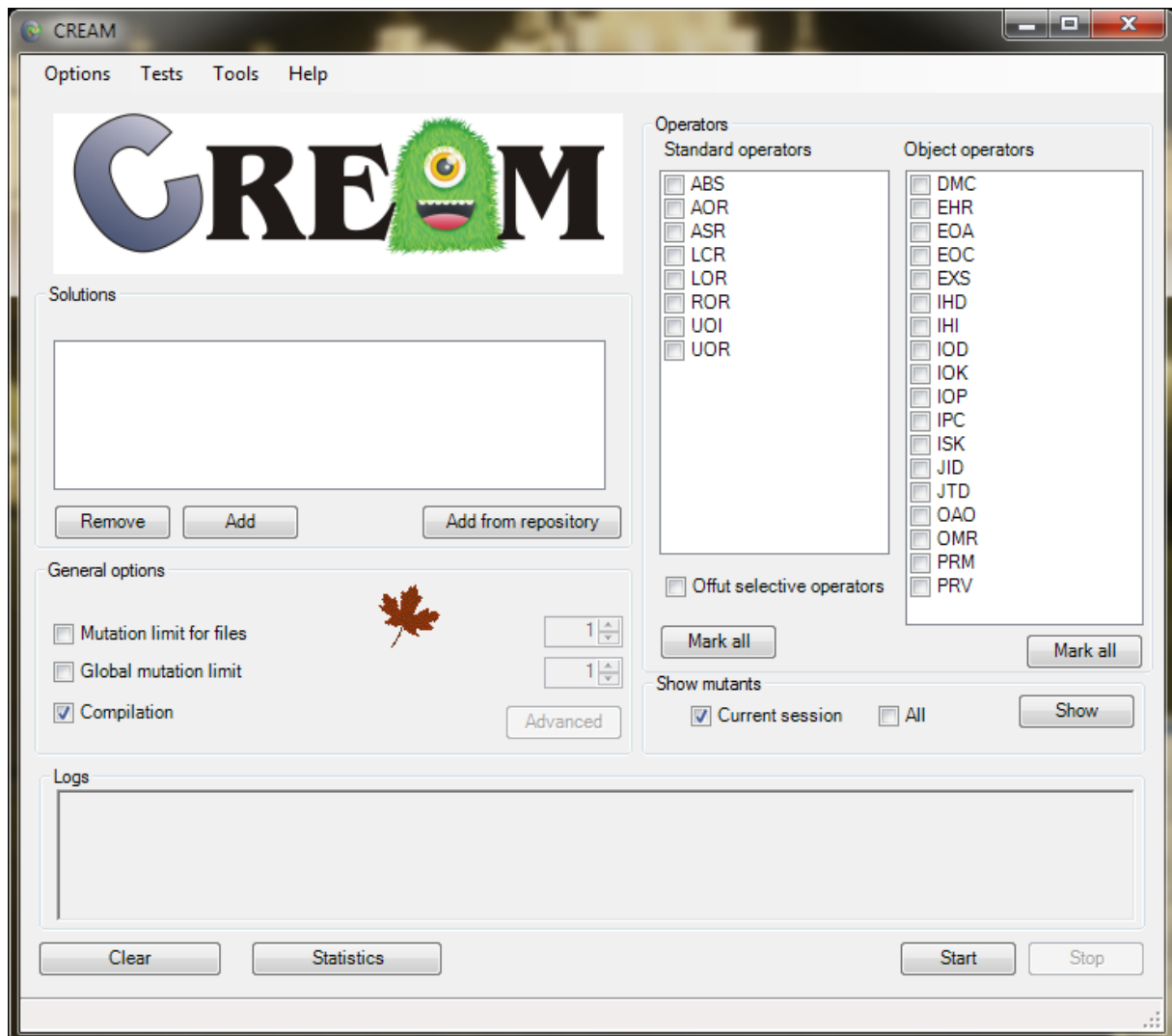
Zaleca się również instalację Microsoft Excel 2003 lub wyżej – biblioteki dostarczane z tym programem wykorzystywane są podczas zapisu raportów do plików z rozszerzeniem xls.

W celu zainstalowania narzędzia CREAM v.3.0. należy zaopatrzyć się w wersję instalacyjną programu. Plik setup.msi można ściągnąć ze strony [6]. Po uruchomieniu instalatora należy postępować zgodnie z komunikatami na ekranie.

Gdy proces instalacji zakończy się sukcesem, użytkownik będzie mógł uruchomić program CREAM v.3.0 poprzez skrót na pulpicie, lub wybór odpowiedniej pozycji z menu Start systemu Windows (*Programy->EiTIPW->CREAM*). Użytkownik w tym momencie może zostać poproszony o zainstalowanie .NET Framework 4.0., gdyż jest on niezbędny do uruchomienia narzędzia.

Wraz z narzędziem dostarczane są przykładowe pliki z zapisanymi mutantami i wynikami przeprowadzonych na nich testów (pliki z rozszerzeniem *.mut*). Dzięki temu użytkownik ma możliwość przetestowania wszystkich dostępnych funkcjonalności z pominięciem długotrwałego procesu generowania i testowania mutantów.

Po uruchomieniu narzędzia CREAM v.3.0. otwiera się okno główne aplikacji (Rys. 1).

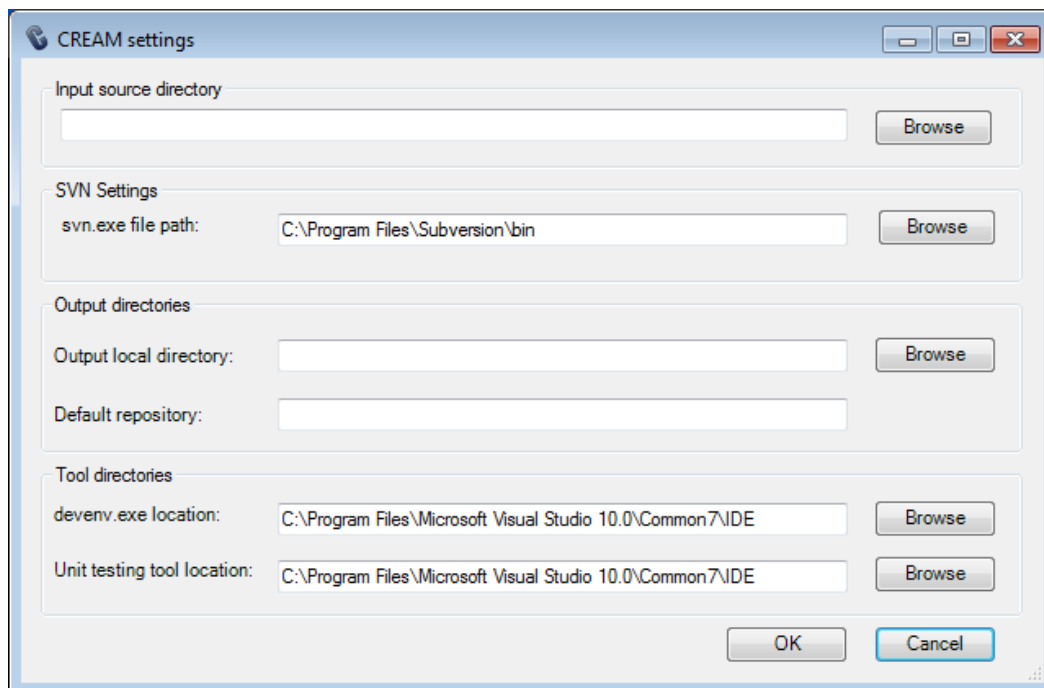


Rys. 1 CREAM v.3.0. - Okno główne aplikacji

Pierwsze kroki pracy z narzędziem powinny być związane z konfiguracją środowiska. W tym celu należy wybrać pozycję *Options->CREAM Settings* z menu na górze okna głównego aplikacji. Ukaże się wówczas okno ustawień narzędzia (Rys. 2). W celu poprawnej konfiguracji należy uzupełnić następujące dane:

- *Input source directory* – domyślny katalog do poszukiwania programów

- *Svn.exe file path* – katalog, w którym znajduje się plik svn.exe (opcjonalne, niezbędne przy korzystaniu z funkcjonalności przechowywania mutantów w repozytorium svn)
- *Output local directory* – katalog roboczy narzędzia CREAM v.3.0. (użytkownik musi mieć prawa zapisu do tego katalogu)
- *Default repository* – domyślny adres repozytorium SVN (opcjonalne, potrzeba jedynie gdy użytkownik będzie chciał zapisywać mutanty na serwerze SVN).
- *devenv.exe localization* – ścieżka do narzędzia devenv.exe (kompilator dostarczany ze środowiskiem Visual Studio).
- *Unit testing tool localization* – lokalizacja narzędzie do przeprowadzania testów jednostkowych. Poprawna jest ścieżka do katalogu, w którym znajduje się plik nunit-console.exe (testy jednostkowe przygotowane z wykorzystaniem NUnit) lub mstest.exe (testy jednostkowe przygotowane za pomocą narzędzia wbudowanego w środowisko Visual Studio)



Rys. 2 CREAM v.3.0. - okno ustawień aplikacji

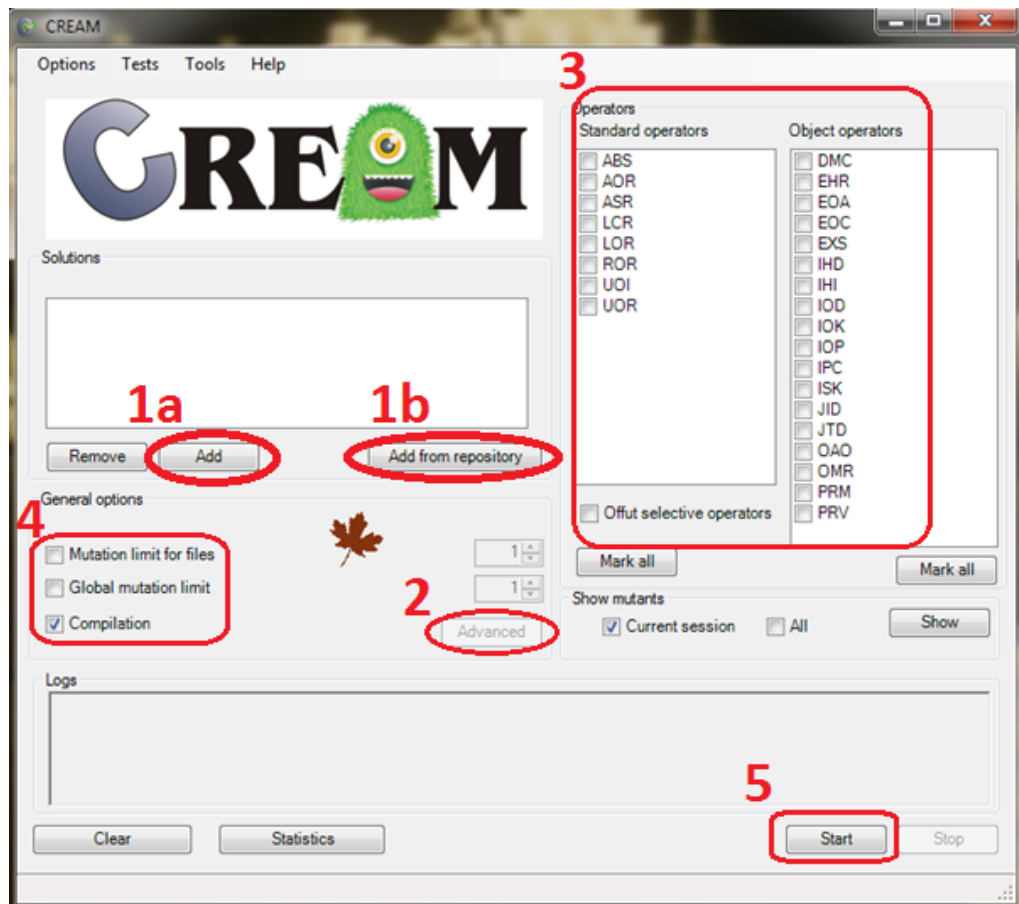
Po uzupełnieniu powyższych informacji narzędzie jest już skonfigurowane i gotowe do pracy. Kolejnym krokiem opisanym w podrozdziałach 1.4-1.9 jest generacja i testowanie mutantów.

1.2. Wczytanie projektu

W celu wczytania projektu należy:

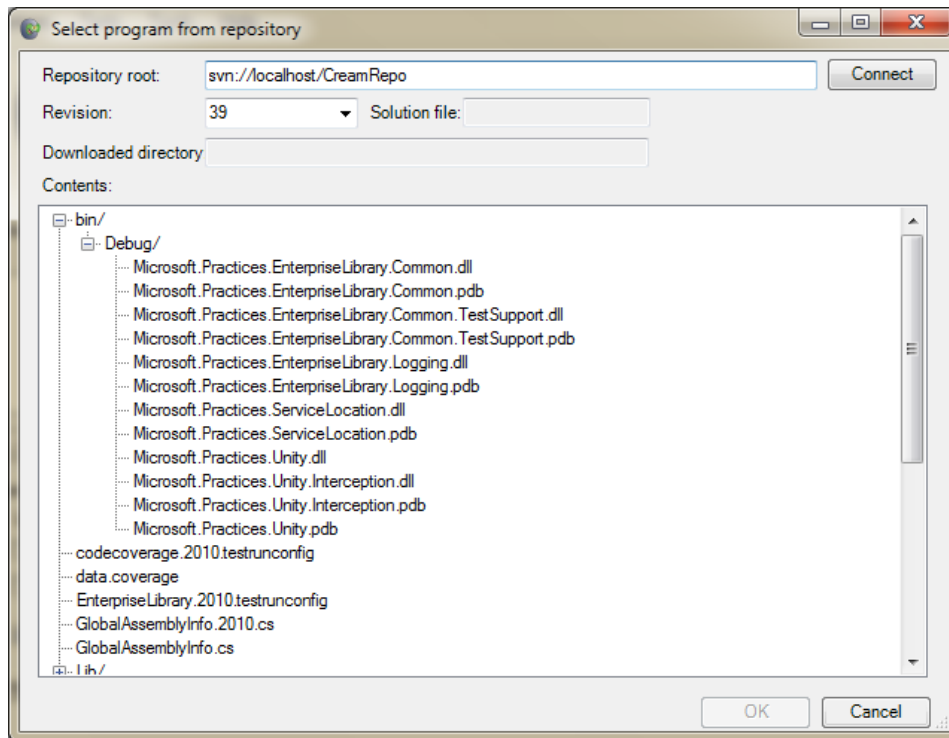
- 1) Uruchomić i skonfigurować narzędzie CREAM v.3.0. zgodnie z opisem w rozdziale 1.1.

- 2) Wczytać rozwiązanie (*ang. solution*) z programem napisanym w języku C# z wykorzystaniem Visual Studio poprzez wskazanie lokalizacji pliku z rozszerzeniem .sln (punkt 1a lub 1b na Rys. 3).



Rys. 3 CREAM v.3.0. okno główne aplikacji - proces generacji mutantów

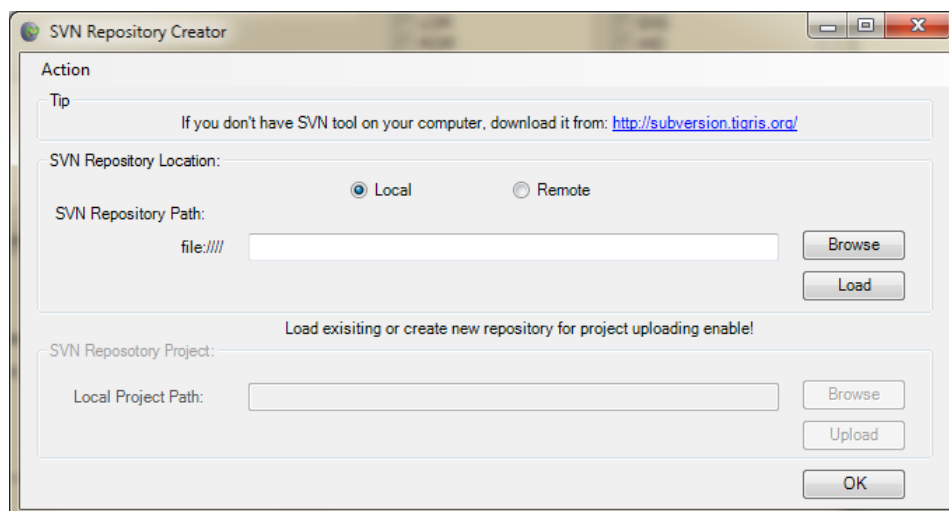
- a) wczytanie rozwiązania zapisanego na dysku lokalnym – w celu wczytania rozwiązania zapisanego na dysku lokalnym użytkownik powinien nacisnąć przycisk *Add* w sekcji *Solutions* okna głównego aplikacji (1a na Rys. 3), a następnie wskazać położenie pliku z rozszerzeniem .sln.
- b) wczytanie rozwiązania zapisanego w repozytorium SVN – w celu wczytania rozwiązania zapisanego na repozytorium SVN użytkownik powinien nacisnąć przycisk *Add from repository* w sekcji *Solutions* okna głównego aplikacji (1b na Rys. 3). Pojawi się wówczas okno (Rys. 4) udostępniające funkcjonalność przeglądania repozytoriów. Aby na wybranym repozytorium wskazać plik rozwiązania należy
- wpisać pełny adres repozytorium w polu *Repository root* i nacisnąć przycisk *Connect*
 - wybrać z listy *Revision* numer rewizji, w której zapisany jest niezmutowany projekt
 - na drzewie w sekcji *Contents* wskazać plik (w ramach danej rewizji) z rozszerzeniem .sln i nacisnąć przycisk *OK*.



Rys. 4 Okno przeglądania repozytorium SVN

1.3. Korzystanie z kreatora repozytoriów SVN

W narzędziu CREAM v.3.0. dostępny jest kreator repozytoriów SVN. Uruchomić go można poprzez wybranie pozycji *SVN Repository Creator* w menu *Tools* okna głównego aplikacji (Rys. 5).



Rys. 5 Okno kreatora repozytoriów SVN

Pierwszym krokiem który użytkownik wykonuje w kreatorze jest wczytanie repozytorium SVN. W tym celu należy w sekcji *SVN Repository Location* wybrać czy jest to

repozytorium lokalne czy zdalne i podać jego adres. Po wprowadzeniu poprawnych danych i naciśnięciu przycisku *Load* repozytorium zostaje wczytane i w przypadku gdy jest ono puste udostępnia się możliwość załadowania do niego nowego projektu, który w dalszych krokach można wykorzystać jako projekt oryginalny dla procesu mutacji.

Gdy w kreatorze użytkownik wybierze repozytorium lokalne i poda adres odwołujący się do pustego katalogu zostanie zapytany czy chce w tym miejscu stworzyć puste repozytorium. Odpowiadając twierdząco na pytanie tworzy się repozytorium, do którego możemy załadować projekt.

1.4. Generowanie mutantów (scenariusz podstawowy)

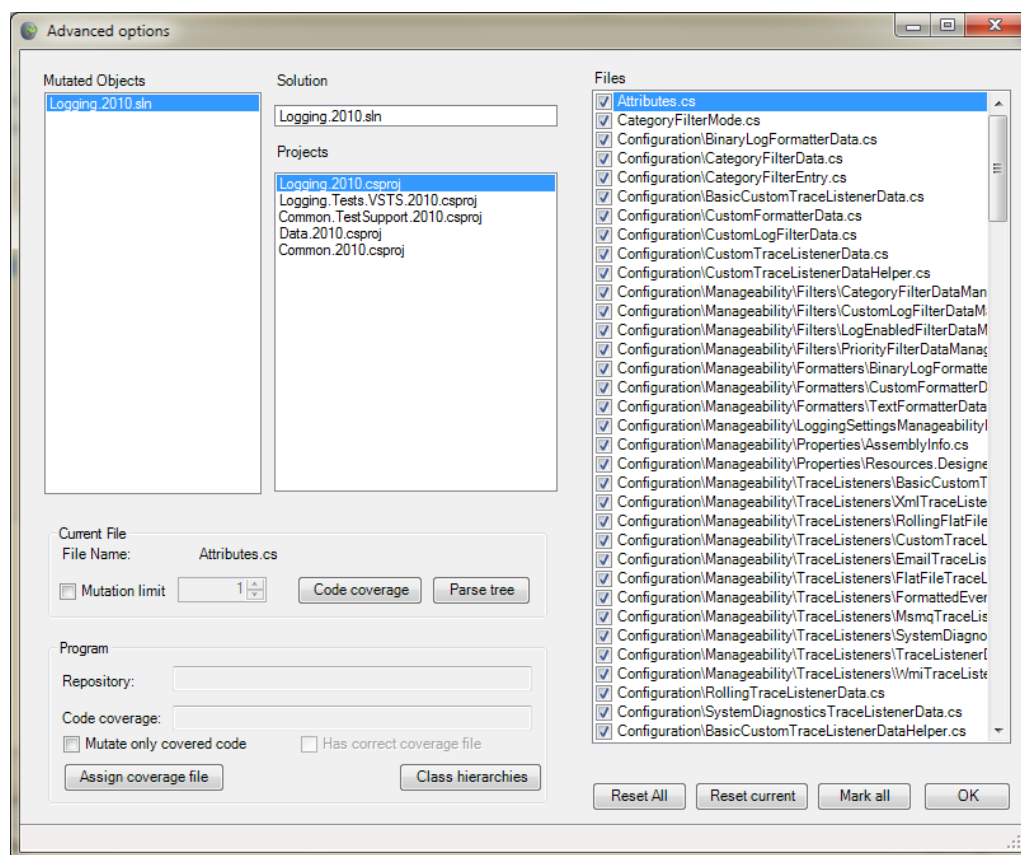
W celu wygenerowanie mutantów należy wykonać następujące czynności:

- 1) Uruchomić, skonfigurować i wczytać projekt zgodnie z opisem w rozdziale 1.2.
- 2) Wybrać operatory, z użyciem których mają zostać stworzone mutanty (punkt 3 na Rys. 3).
- 3). Należy zwrócić uwagę, że pod listą operatorów strukturalnych użytkownik ma możliwość zaznaczyć opcję *Offutt selective operator*. Wybranie jej powoduje wybranie podzbioru selektywnych operatorów strukturalnych (według badań opisanych w [7])
- 3) Ustawić w sekcji *General Options* okna głównego dodatkowe parametry związane z procesem mutacji (punkt 4 na Rys. 3):
 - Zdefiniowanie maksymalnej liczby mutacji, jaka może być wprowadzona z użyciem poszczególnych operatorów do każdego pliku (pole *Mutation limit for files*).
 - Zdefiniowanie maksymalnej liczby mutacji jaka może być wprowadzona przez każdy z operatorów (pole *Global mutation limit*).
 - Określenie czy po wprowadzeniu mutacji ma być uruchamiany proces kompilacji (pole *Compilation*)
- 4) Rozpoczęcie procesu generowania mutantów poprzez naciśnięcie przycisku *Start* (punkt 4 na Rys. 3)

1.5. Generowanie mutantów (scenariusz zaawansowany)

Użytkownik ma możliwość skorzystania z dodatkowych funkcjonalności narzędzia CREAM v.3.0. postępując według poniższego scenariusza:

- 1) Wykonanie kroków 1, 2 i 3 scenariusza podstawowego (rozdział 1.4)
- 2) Po wybraniu opcji *Advance* w sekcji *General options* (punkt 2 na Rys. 3) lub pozycji *Advanced* z menu *Options* okna głównego aplikacji otworzy się okno zaawansowanych opcji mutacji (Rys. 6).



Rys. 6 CREAM v.3.0. - okno zaawansowanych opcji mutacji

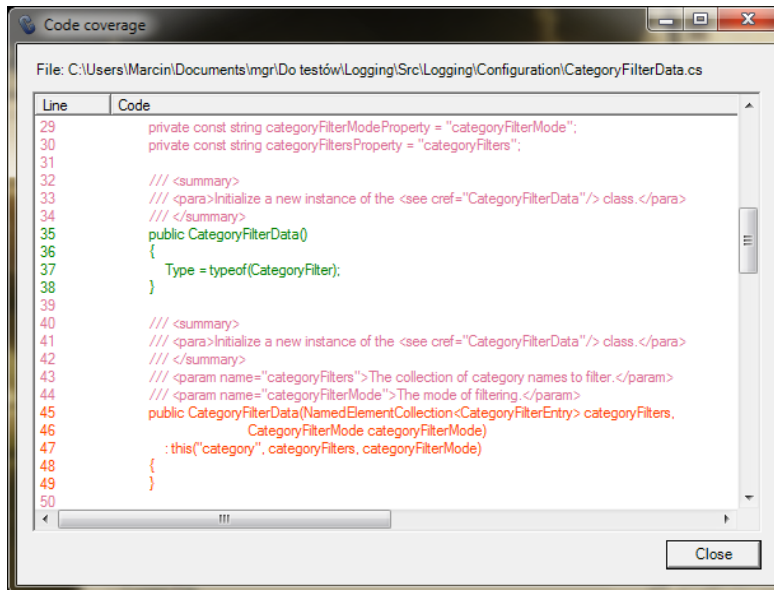
W oknie tym użytkownik widzi strukturę katalogów i plików wczytanego projektu. Poprzez zaznaczenie/odznaczenie wybranego pliku w sekcji *Files* użytkownik może decydować, które z nich będą uczestniczyły w procesie mutacji.

W oknie *Advanced Options* użytkownik może również przypisać plik pokrycia kodu do wczytanego projektu (poprzez naciśnięcie przycisku *Assign coverage file* i wskazanie wygenerowanego wcześniej pliku) i zażądać, aby mutowane były jedynie pokryte fragmenty kodu (poprzez zaznaczenie opcji *Mutate only covered code* w sekcji *Program*). CREAM v.3.0. rozpoznaje pliki z informacjami o pokryciu kodu pochodzącymi z narzędzi: NCover[3] (rozszerzenie .xml), Visual Studio (rozszerzenie .coverage), Tester[4] (rozszerzenie .txt). Po wskazaniu pliku z informacjami o pokryciu użytkownik zostanie zapytany o sposób, w jaki mają być traktowane linie, o których nie ma dostępnych informacji w plikach z pokryciem. Zaleca się wybranie, aby były one traktowane jako niepokryte.

W celu weryfikacji poprawności wczytanych danych użytkownik może wybrać dowolny plik z listy i poprzez naciśnięcie przycisku *Code coverage* (w sekcji *Current File*) wyświetlić jego kod z zaznaczonym pokryciem (Rys. 7). W oknie tym każda linia kodu w zależności od swojego koloru reprezentuje inną informację:

- Czarny – brak pliku z informacją o pokryciu
- Zielony – kod pokryty
- Jasno zielony – brak informacji o danej linii kodu, ale traktowana jako pokryta
- Czerwony – kod niepokryty

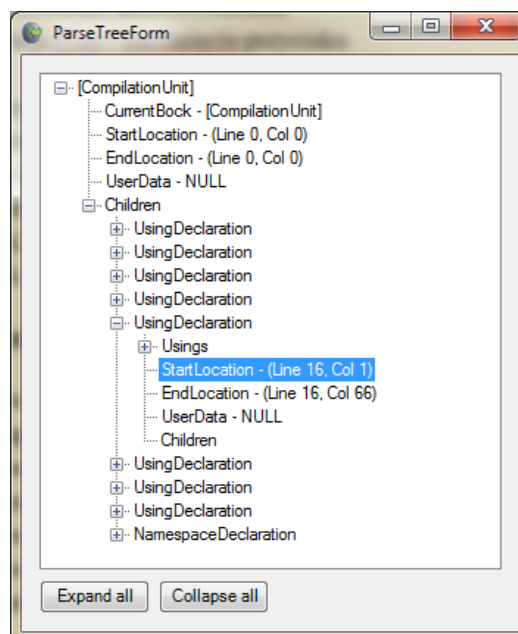
- Jasno czerwony - brak informacji o danej linii kodu, ale traktowana jako niepokryta



Rys. 7 CREAM v.3.0. - Okno z informacją o pokryciu kodu

Okno *Advanced Options* udostępnia funkcjonalność polegającą na ograniczeniu liczby mutacji, jakie maksymalnie będą mogły zostać zaaplikowane do poszczególnych plików w procesie mutacji.

W oknie tym można również dla wybranego pliku wyświetlić drzewo rozbioru składniowego (Rys. 8) poprzez zaznaczeniu pliku w sekcji *Files* i naciśnięciu przycisku *Parse tree* w sekcji *Current File*.

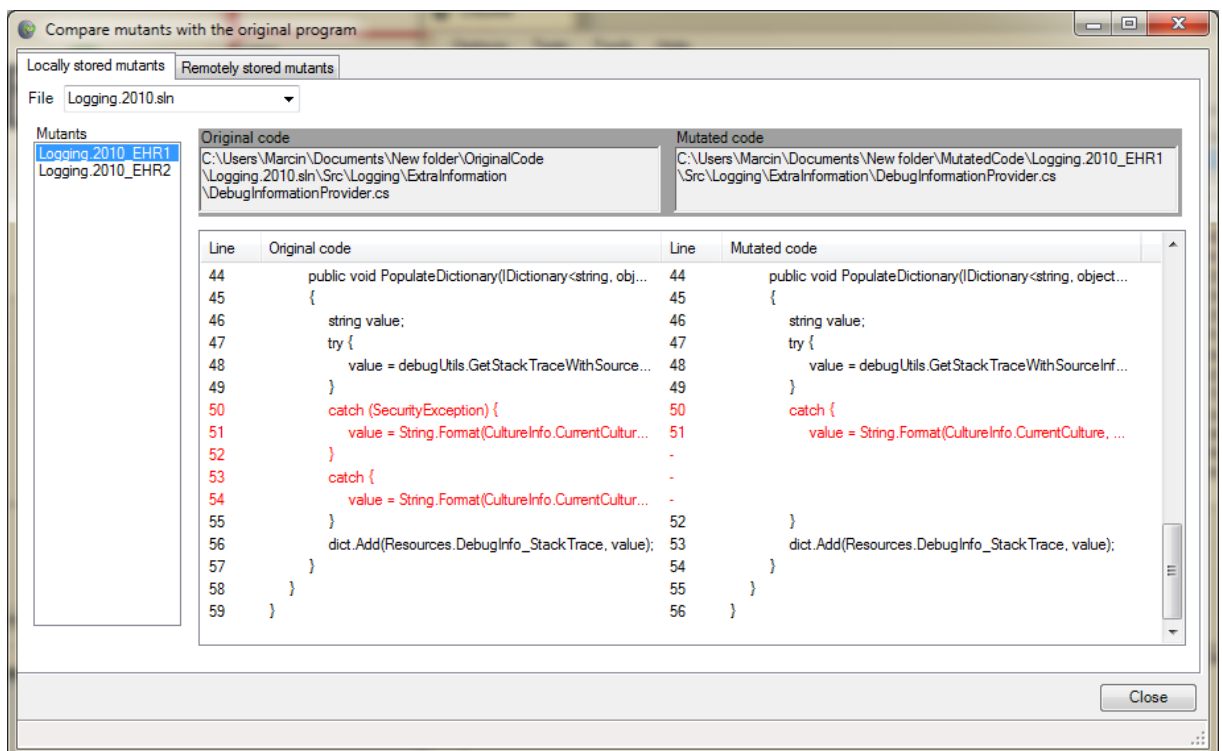


Rys. 8 Okno prezentujące drzewo rozbioru składniowego pliku

1.6. Przeglądnie wygenerowanych mutantów

Aby przejrzeć wygenerowane mutanty należy:

- 1) Wygenerować mutanty zgodnie z opisem w punkcie 1.4 lub 1.5.
- 2) Wybrać przycisk *Show* w sekcji *Show mutants* okna głównego aplikacji. Otworzy się wówczas okno prezentacji mutantów (Rys. 9). W oknie tym użytkownik ma możliwość wyboru z listy po lewej stronie jednego z wygenerowanych wcześniej mutantów. Po dokonaniu tego wyboru pokazuje się kod oryginalny i zmutowany z zaznaczonym miejscem, gdzie dokonana została zmiana. W oknie tym można przeglądać zarówno mutanty zapisane lokalnie, jak też te z repozytorium SVN.



Rys. 9 Okno prezentacji mutantów

1.7. Przeprowadzanie testów na mutantach

CREAM v.3.0. udostępnia (w porównaniu do wersji 2.0.) nowy sposób przeprowadzania testów na mutantach z możliwością zapisania ich wyników do pliku. Podejście takie umożliwiło odseparowanie procesu generowania i testowania mutantów od dalszych analiz (sposób ich przeprowadzania został opisany w następnych podrozdziałach).

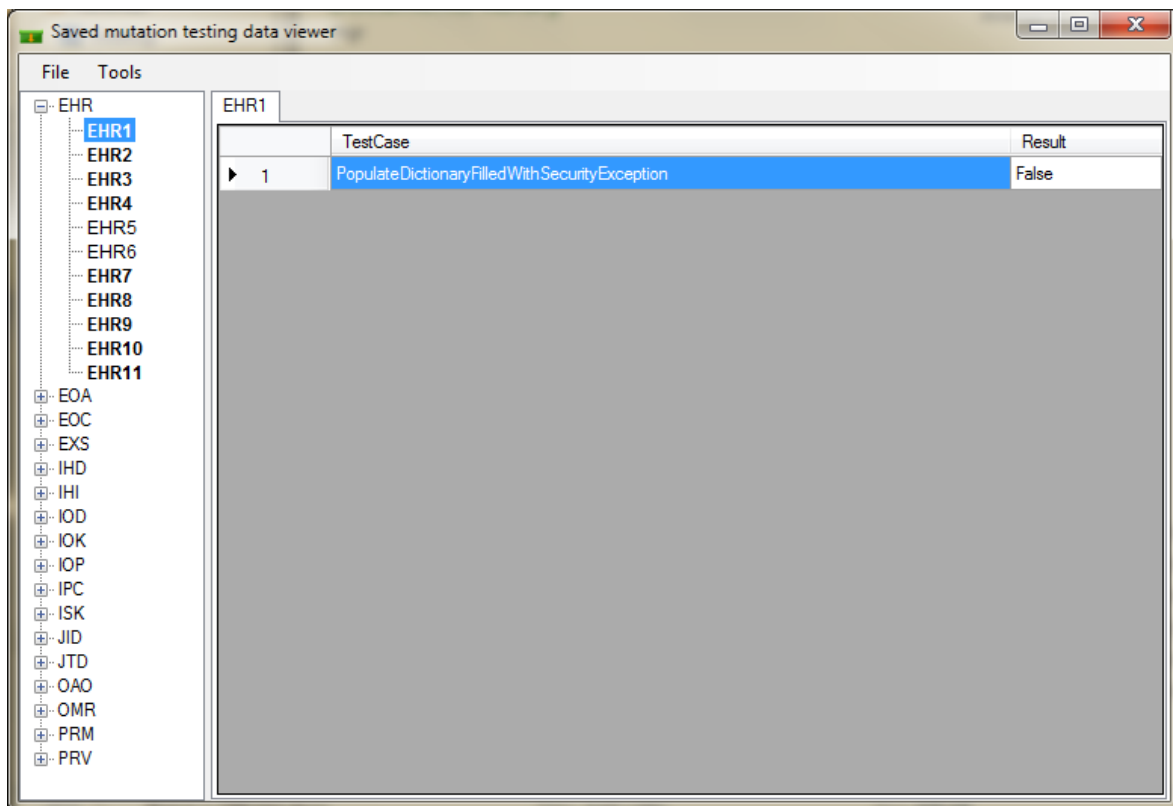
Aby przeprowadzić testy na mutantach i wygenerować plik z wynikami należy:

- 1) Wygenerować mutanty zgodnie z tym, co zostało opisane w rozdziałach 1.4 lub 1.5.
- 2) W oknie głównym aplikacji CREAM v.3.0. wybrać pozycję *Run tests* z menu *Tests*
- 3) W pierwszym oknie kreatora zapoznać się z informacją o celu i sposobie przeprowadzania procesu testowania i nacisnąć przycisk *Next*.
- 4) W drugim kroku wybrać miejsce, gdzie zapisane są mutanty: dysk lokalny lub repozytorium SVN.

- 5) W kolejnym kroku działania kreatora wybrać dane umożliwiające poprawne wczytanie projektu, mutantów i testów. W zależności od tego, czy mutanty i projekt znajdują się na lokalnym dysku, albo w repozytorium SVN, zostaniemy poproszeni o inne informacje. Różnica polega na sposobie wskazania miejsca, gdzie znajduje się oryginalny projekt (analogicznie jak w punkcie 2 rozdziału 1.2). Inne informacje to: względna ścieżka w ramach projektu do plików z testami (z rozszerzeniem .dll) i wybór zewnętrznego narzędzia do przeprowadzania testów jednostkowych (JUnit lub MSTest). Po uzupełnieniu powyższych danych należy nacisnąć *Next*.
- 6) Ostatnim krokiem jest wybranie czy chcemy testować wszystkie mutanty ze wskazanych czy też tylko wybrane (sekcja *Prepare data from* okna kreatora), wygenerowanie wyników (naciśnięcie przycisku *Run*) oraz zapisanie pliku z danymi z rozszerzeniem .mut na lokalnym dysku (przycisk *Save*).
- 7) Wyniki testów zapisane w wygenerowanym pliku możemy podejrzeć z użyciem narzędzia *Data Viewer* (opisanego w rozdziale 1.8).

1.8. Narzędzie *Data Viewer*

Podstawowa funkcjonalność narzędzia *Data Viewer* polega na wyświetlaniu danych zapisanych w plikach z rozszerzeniem .mut. Okno główne aplikacji zostało przedstawione na Rys. 10.



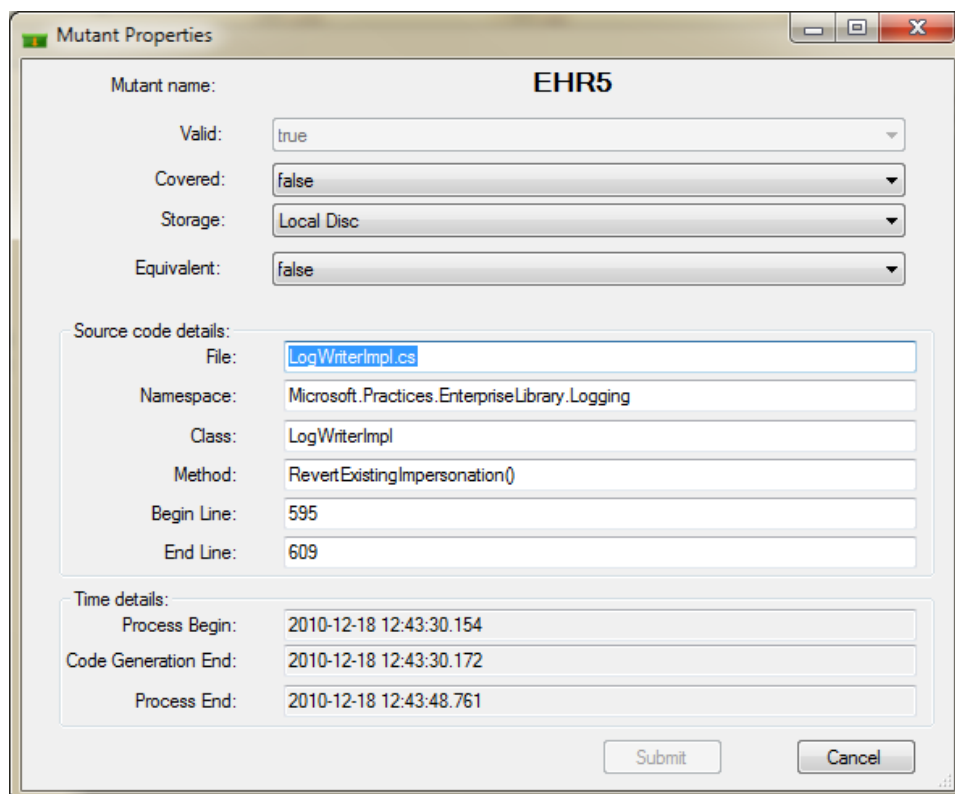
Rys. 10 Okno główne aplikacji *Data Viewer*

Aplikację można uruchomić na jeden z poniższych sposobów:

- Poprzez dwukrotne kliknięcie na plik z rozszerzeniem .mut. Aplikacja *Data Viewer* automatycznie kojarzona jest z plikami podczas procesu instalacji. Po uruchomieniu użytkownik od razu będzie widział dane zapisane w wybranym pliku.
- Wybranie pozycji *Data Viewer* z menu Start systemu Windows (domyślnie katalog EiTIPW). Aby teraz wczytać wybrany plik należy wybrać pozycję *Load* z menu *Options*.
- Poprzez wybranie pozycji *Data viewer* w menu *Tools* okna głównego aplikacji CREAM.

Jeżeli użytkownik uruchomił aplikację w inny sposób niż poprzez dwukrotne kliknięcie na pliku z rozszerzeniem .mut, chcąc korzystać z narzędzia powinien na początku wczytać plik z danymi (wygenerowany przez narzędzie CREAM v.3.0.). Po wczytaniu w oknie głównym aplikacji po lewej stronie wypełni się drzewo z mutantami (z podziałem na operatory). Dokonując wyboru jednego z dostępnych mutantów po prawej stronie ukazuje się zakładka z informacjami o zabijających go testach.

Bardziej szczegółowe informacje o wybranym mutancie można uzyskać klikając na nim prawym przyciskiem myszy i wybierając z menu kontekstowego pozycję *Properties* (Rys. 11). Spośród prezentowanych w oknie informacji część z nich jest wyszarzona (nie możliwa jest ich edycja), część natomiast użytkownik może zmieniać. Należy pamiętać, że po dokonaniu zmian, zapisane zostaną one jedynie po naciśnięciu przycisku *Submit*.



Rys. 11 Okno właściwości mutantu

Dodatkowo narzędzie umożliwia wykonanie kilku operacji na plikach. Funkcjonalności te dostępne są z poziomu menu *Tools* w głównym oknie aplikacji:

- *Merge Files* – po wybraniu tej pozycji z menu otwiera się okno, w którym użytkownik może wybrać plik oryginalny i dodatkowe pliki, które następnie mogą być połączone w jeden plik wyjściowy. Należy zwrócić uwagę, iż mutanty zapisane w plikach dodatkowych dołączane są jedynie w przypadku, gdy nie ma mutantów o takiej samej nazwie w pliku oryginalnym, lub nie został już wcześniej dołączony. Po wykonaniu tej operacji wygenerowany plik należy zapisać (wybranie *File -> Save As* z menu w głównym oknie aplikacji)
- *Statistics* – podsumowanie liczby wczytanych mutantów z podziałem na typ. Dodatkowo prezentowana jest liczba mutantów zabitych i wyliczany jest wynik mutacji MS (*ang. Mutation Score*). Możliwe jest również ustawienie przez użytkownika kryteriów branych pod uwagę przy wyliczaniu statystyk (mutanty żywe/nieżywe, pokryte/niepokryte, równoważne/nierównoważne).
- *Check for uncovered and killed* – wyszukanie mutantów, które posiadają informację o tym, że są niepokryte i istnieją testy, które je zabijają. Informacja ta wykorzystywana jest w celu weryfikacji poprawności wczytanych danych

1.9. Wykorzystanie funkcjonalności badania metod redukcji kosztów

Pliki z wynikami testów (z rozszerzeniem *.mut*) wygenerowane zgodnie z opisem z punktu 1.7 mogą stanowić dane wejściowe dla dalszych analiz badających metody redukcji kosztów, których sposób wykonania został przedstawiony w kolejnych podrozdziałach.

Kolejne kroki analiz wizualizowane są w formie kreatorów. Wszystkie z nich posiadają części wspólne, niezależne od wybranej analizy (opisane w rozdziale 1.9.1), jak też części specyficzne dla danej analizy (opisane w podrozdziałach 1.9.2, 1.9.3, 1.9.4).

1.9.1. Części wspólne wizualizacji

Pierwszym krokiem każdej z wizualizacji jest krótki opis wybranej analizy.

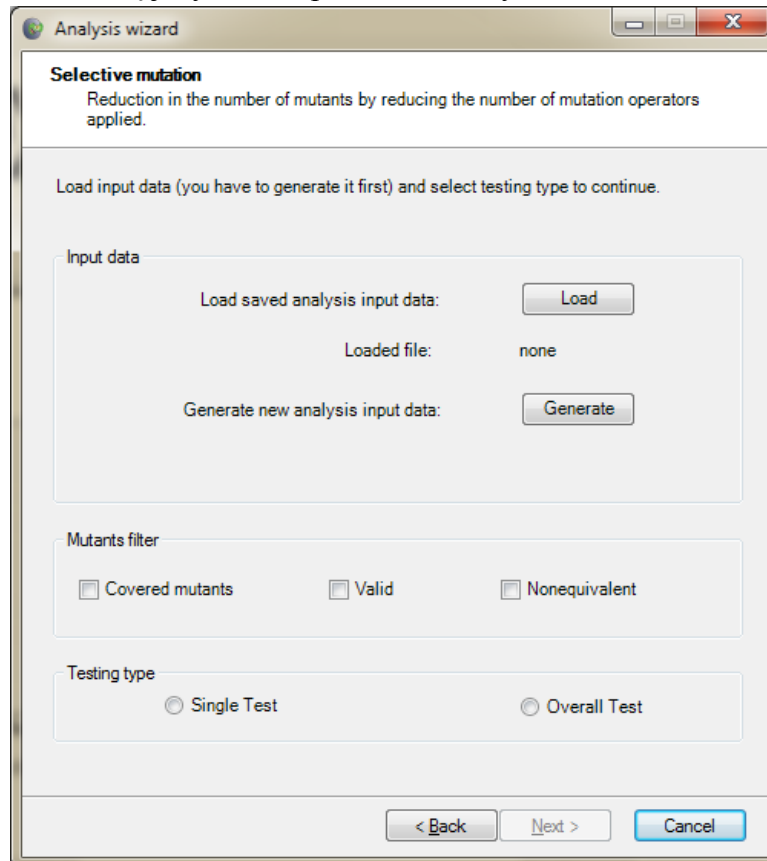
Kolejnym krokiem wspólnym jest wczytanie danych wejściowych (Rys. 12). W oknie reprezentującym ten krok użytkownik ma możliwość załadowania danych z istniejącego pliku (pozycja *Load saved analysis input data* w sekcji *Input data*), lub otwarcie kreatora generowania danych wejściowych (pozycja *Generate new analysis input data* w sekcji *Input data*), który został opisany w punkcie 1.7.

W oknie tym użytkownik ma możliwość wyboru filtru (sekcja *Mutants filter*), według którego dobrane zostaną mutanty do dalszej analizy:

- Pokryte przez testy lub wszystkie niezależnie od pokrycia (pozycja *Covered mutants*)
- Kompilowalne lub wszystkie (pozycja *Valid*)
- Oznaczone jako nierównoważne lub wszystkie (pozycja *Nonequivalent*)

Dodatkowo w tym kroku użytkownik może wybrać rodzaj analizy (sekcja *Testing type*):

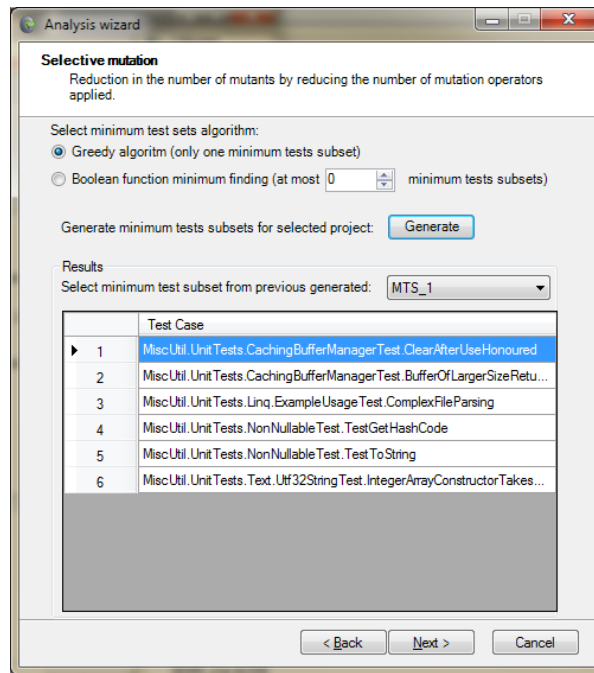
- Pojedynczą (*ang. Single Test*) – przebieg analizy prezentowany jest krok po kroku, użytkownik ma możliwość ręcznego ustawienia parametrów i obserwowania wyników poszczególnych etapów.
- Całościową (*ang. Overall Test*) – przebieg analizy prezentowany jest w formie zbiorczego raportu. Zawiera on dane dla kompletnego zbioru parametrów wejściowych, a jego generowanie przebiega bez interakcji z użytkownikiem. Z punktu widzenia użytkownika generowanie raportu wygląda tak samo, niezależnie od wybranej analizy, różnice widoczne są jedynie w raporcie końcowym.



Rys. 12 Wczytywanie danych wejściowych dla analiz

Po wczytaniu danych następuje etap wyboru podzbioru mutantów. Jest on zależny od wybranej analizy i opisany został w kolejnych rozdziałach.

Gdy mamy już wybrany podzbiór mutantów następuje proces generowania dla niego minimalnych zbiorów przypadków testowych (Rys. 13). Użytkownik ma możliwość wyboru jednego z zaimplementowanych algorytmów, po czym poprzez naciśnięcie przycisku *Run* rozpoczyna proces generacji. Gdy ten się zakończy, jego wyniki zostaną przedstawione w tabeli. W zależności od algorytmu i jego parametrów w procesie tym może być generowany więcej niż jeden zbiór. Użytkownik powinien wówczas wybrać jeden z nich przy użyciu listy rozwijanej znajdującej się powyżej tabeli z wynikami.

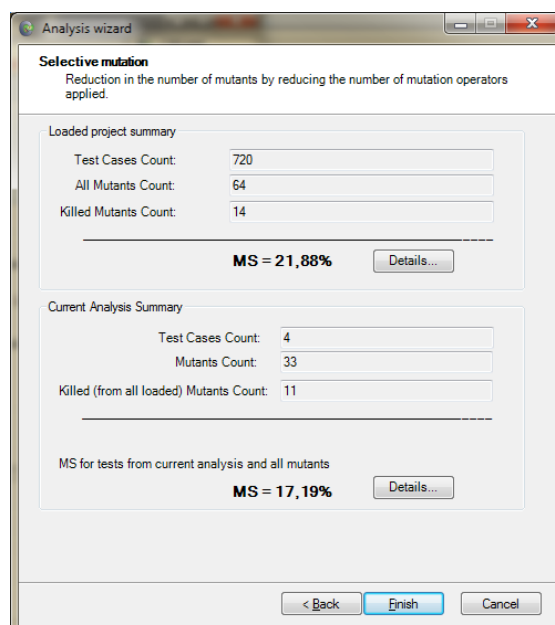


Rys. 13 Generowanie minimalnego zbioru przypadków testowych

Gdy już zostanie wybrany podzbiór przypadków testowych, kolejnym krokiem jest wyliczenie statystyk i odpowiednich wartości MS, które są wynikiem danej analizy (Rys. 14).

Znaczenie wyliczonych wartości MS jest następujące. Pierwsza z nich jest wartością MS liczoną jako procent zabitych mutantów (spośród wszystkich wygenerowanych) z wykorzystaniem wszystkich dostarczonych przypadków testowych. Dla porównania poniżej tej wartości prezentowane jest MS liczone z wykorzystaniem podzbioru przypadków testowych z poprzedniego kroku analizy przypadków testowych i wszystkich wygenerowanych mutantów.

Użytkownik po zapoznaniu się z wynikami może zakończyć działanie kreatora (przycisk *Finish*), lub cofnąć się i dokonać kolejnej analizy z innymi wartościami parametrów (przycisk *Back*).

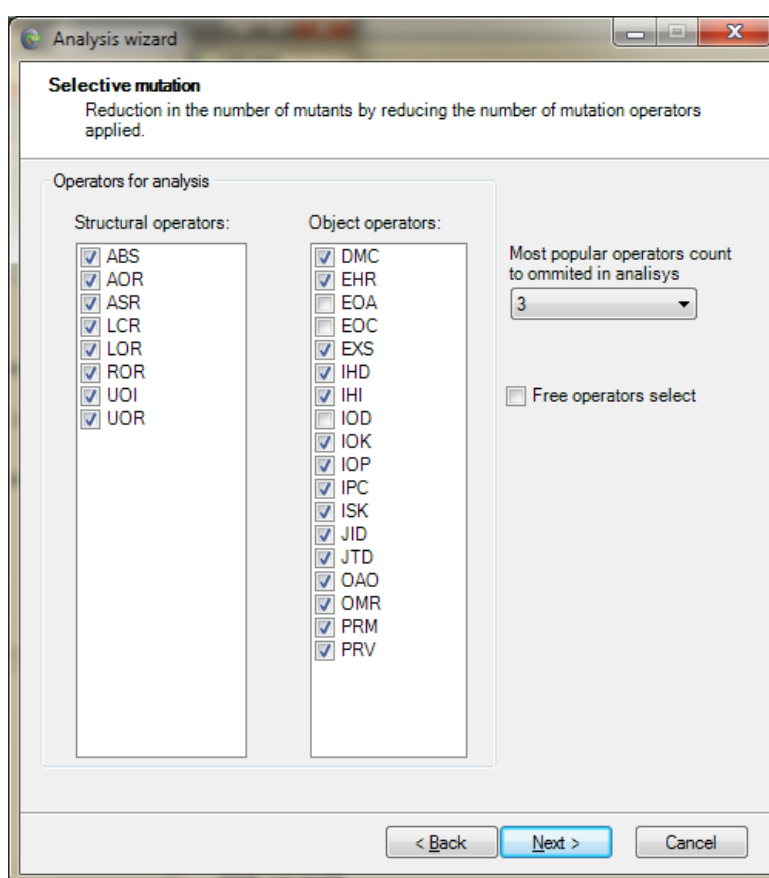


Rys. 14 Okno wyników analizy

1.9.2. Mutowanie selektywne

Podzbiór mutantów dla metody mutowania selektywnego może być wybrany na dwa sposoby:

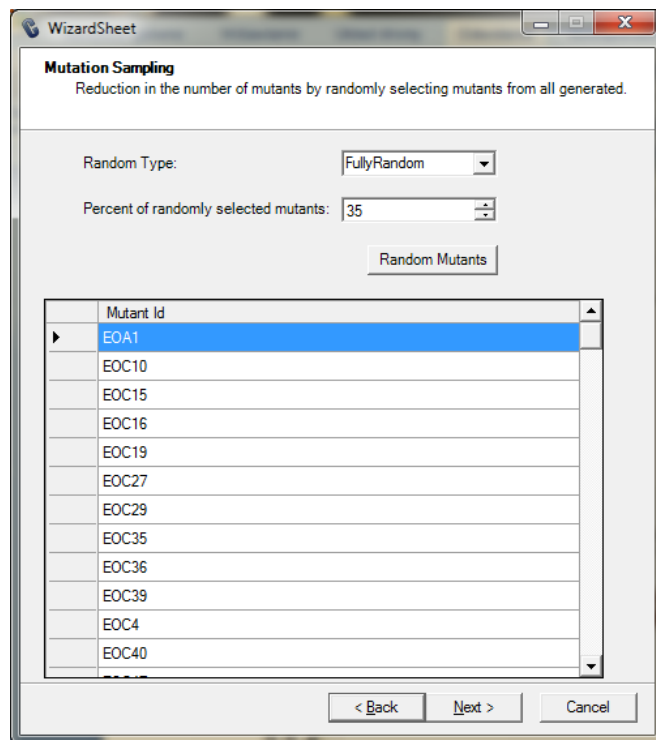
- poprzez zdefiniowanie liczby pomijanych operatorów (Rys. 15). W tym podejściu eliminujemy kolejno operatory według ich popularności (liczby wygenerowanych mutantów). Aby dokonać takiego wyboru należy w oknie kreatora wybrać wartość z listy *Most popular operator count omitted in analysis*.
- poprzez ręczne wybranie operatorów, które mamy brać pod uwagę przy dalszej analizie. Wyboru operatorów z listy dokonujemy po zaznaczeniu opcji *User operators select*.



Rys. 15 Wybór operatorów do pominięcia przy mutacji selektywnej

1.9.3. Próbkowanie mutantów

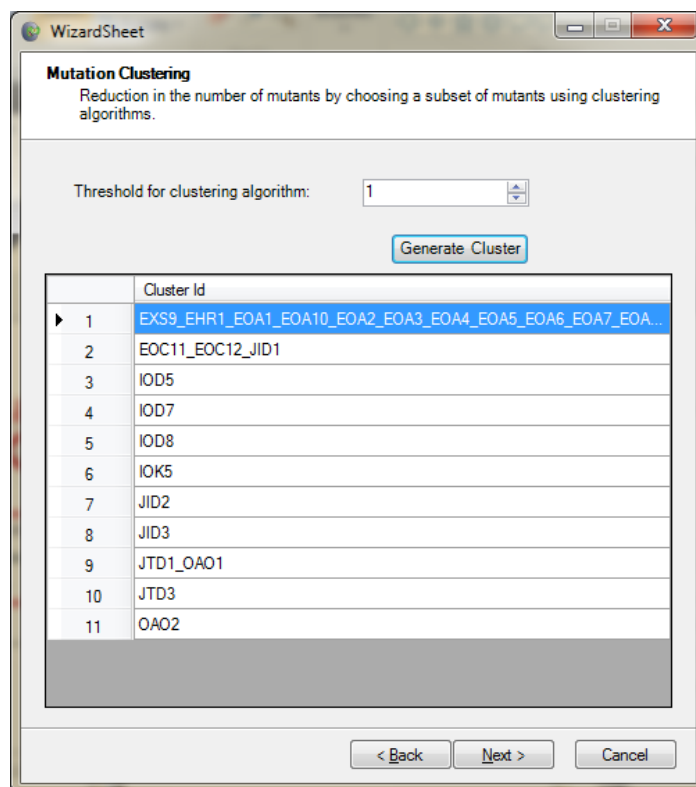
W tym podejściu podzbiór mutantów wybierany jest losowo. Użytkownik jako parametry może podać typ losowości (wybór z listy *Random Type*) i procent mutantów do wylusowania (wybór wartości *Percent od randomly selected mutants*) (Rys. 16)



Rys. 16 Wybór mutantów - próbkowanie mutantów

1.9.4. Grupowanie mutantów

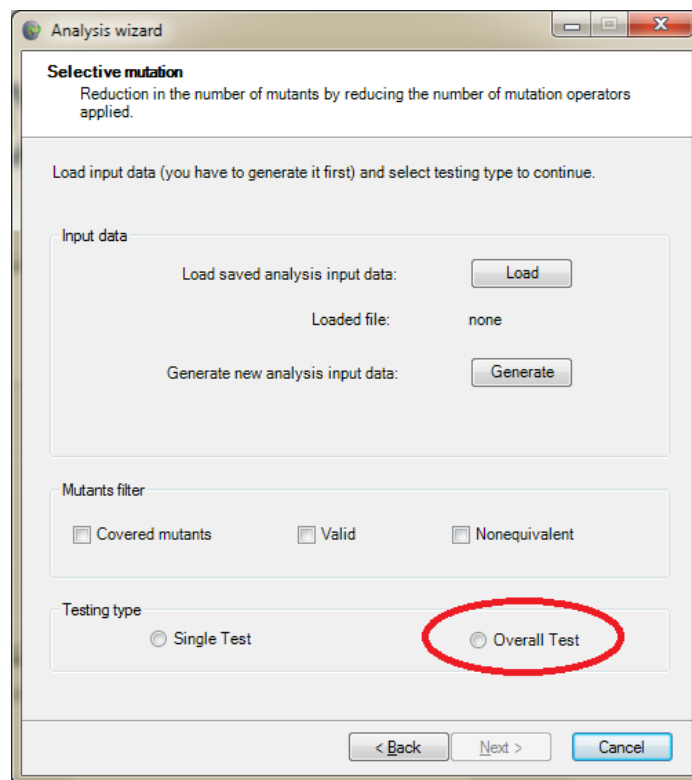
W metodzie grupowania podzbiór mutantów wybierany jest jako przedstawiciele wygenerowanych grup (Rys. 17). Użytkownik jak parametr podaje współczynnik podobieństwa członków w grupie (wartość *Threshold for clustering algorithm*).



Rys. 17 Generowanie podzbioru mutantów w procesie grupowania

1.9.5. Raporty całościowe dla analiz metod redukcji kosztów

Użytkownik podczas pracy z kreatorem do analizy poszczególnych metod redukcji kosztów ma możliwość wyboru testu całościowego (ang. *Overall Test*). Sposób w jaki należy dokonać tego wyboru został przedstawiony na Rys. 18.



Rys. 18 Wybór testu całościowego podczas pracy z kreatorem analizującym metody redukcji kosztów

Rezultatem wyboru tej opcji jest raport wygenerowany w formie pliku .xlsx (domyślnie można otworzyć go przy pomocy narzędzia Microsoft Excel w wersji 2007 lub nowszej). Struktura tak wygenerowanego raportu jest bardzo podobna, niezależnie od wybranej metody redukcji kosztów. Głównymi jej elementami są:

1. Arkusz z ogólnymi informacjami o testowanym projekcie i wartościami parametrów używanymi podczas wyliczeń jakości analizy
2. Arkusze z danymi liczbowymi
3. Arkusze z wykresami wizualizującymi zebrane dane

W kolejnych rozdziałach szczegółowo została opisana zawartość każdego z arkuszy

1.9.5.1. Arkusz z ogólnymi informacjami o testowanym projekcie i wartościami parametrów używanymi podczas wyliczeń jakości analizy

Arkusz ten jest niezależny od metody redukcji kosztów, którą rozpatrujemy. Kolejne, zawarte w nim informacje są następujące (patrz Rys. 19):

- 1) Nazwa projektu
- 2) Data wygenerowania raportu
- 3) Liczba wszystkich dostarczonych przypadków testowych w analizowanym projekcie
- 4) Liczba wszystkich wygenerowanych mutantów dla analizowanego projektu
- 5) Wartość wyniku mutacji liczona dla wszystkich mutantów z wykorzystaniem wszystkich przypadków testowych
- 6) Czas wygenerowania wszystkich mutantów (z kompilacją)
- 7) Czas wygenerowania wszystkich mutantów (bez kompilacji)
- 8) Czas wykonania wszystkich dostarczonych przypadków testowych
- 9) Współczynniki wagowe wykorzystywane przy wyliczeniach jakości analizy

Mutation sampling report	
Project:	1
Date:	2
Test cases count:	3
Mutants count:	4
MS for all test cases:	5
All mutants creation time (with compilation):	6
All mutants creation time (without compilation):	7
All tests execution time	8

weighting factors for analysis quality		
Mutation Score	Test Case Number	Mutant Number
0.6	0.2	0.2

Rys. 19 Arkusz raportu całościowego z informacjami ogólnymi

1.9.5.2. Arkusz z danymi liczbowymi

Arkusz z danymi liczbowymi składa się z dwóch części. Początkowe kolumny są zależne od rozpatrywanej analizy. Następne natomiast są niezależne od wcześniejszych wyborów.

Kolumny z danymi specyficznymi dla mutowania selektywnego przedstawione zostały na Rys. 20. Ich znaczenie jest następujące:

- 1) Liczba pominiętych operatorów mutacji lub nazwa konkretnego operatora (w przypadku gdy jest to tylko jeden)

Mutation Operators Ommited
1

Rys. 20 Kolumny arkusza z danymi specyficzne dla mutacji selektywnych

Kolumny specyficzne dla próbkowania mutantów przedstawione zostały na Rys. 21. Ich znaczenie jest następujące:

- 1) Wartość określająca procent mutantów, które są rozpatrywane w dalszej analizie

	A
1	Selected mutants [%]
2	1

Rys. 21 Kolumny arkusza z danymi specyficzne dla próbkowania mutantów

Kolumny specyficzne dla grupowania mutantów przedstawione zostały na Rys. 22. Ich znaczenie jest następujące:

- 1) Wartość parametru algorytmu grupującego (określa stopień podobieństwa grup)
- 2) Liczba grup powstałych w wyniku działania algorytmu

	A	B
1	Threshold	Cluster Number
2	1	2

Rys. 22 Kolumny arkusza z danymi specyficzne dla grupowania mutantów

Kolumny arkusza z danymi niezależnymi od wybranej analizy zostały przedstawione na Rys. 23. Ich znaczenie jest następujące:

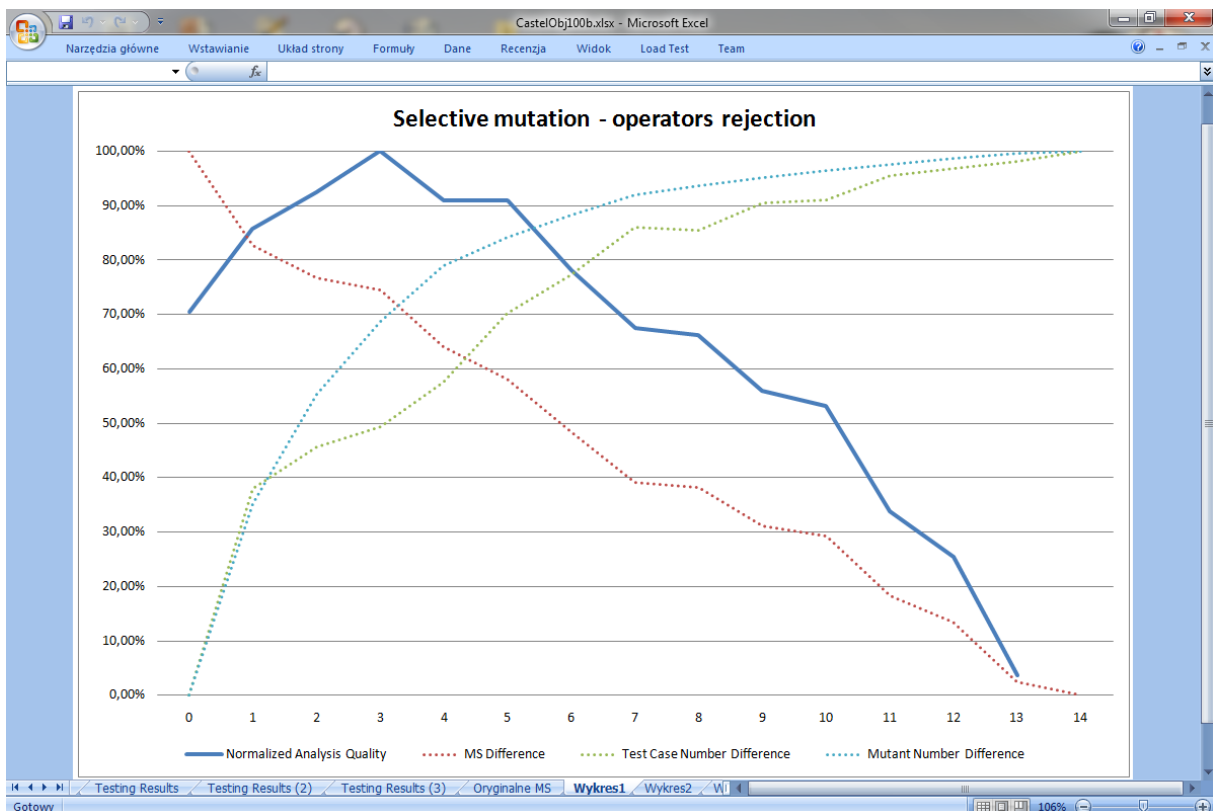
- 1) Liczba zbiorów minimalnych przypadków testowych dla rozpatrywanego podzbioru mutantów.
- 2) Minimalna, średnia i maksymalna wartości wyniku mutacji (ang. *Mutation Score*) wyliczana dla rozpatrywanego podzbioru mutantów z wykorzystaniem każdego spośród zbiorów minimalnych przypadków testowych.
- 3) Minimalna, średnia i maksymalna liczba pozycji w zbiorach minimalnych przypadków testowych.
- 4) Minimalny, średni i maksymalny czas wykonania wszystkich testów rozpatrywany dla każdego ze zbiorów minimalnych przypadków testowych
- 5) Liczba rozpatrywanych mutantów
- 6) Czas utworzenia rozpatrywanych mutantów (bez kompilacji)
- 7) Czas utworzenia rozpatrywanych mutantów (z kompilacją)
- 8) Wartości określające stosunek miar 2, 3, 4, 5, 6, 7 do oryginalnych (liczonych dla wszystkich mutantów i całego zbioru przypadków testowych)
- 9) Znormalizowane wartości wybranych miar z 8
- 10) Wyliczona wartość jakości analizy z uwzględnieniem współczynników wagowych (Rys. 19 p. 9)
- 11) Znormalizowana wartość jakości analizy 10

MTS Number 1	MS 2			Test Case Number 3			Test Case Execution Time 4			Mutant Number 5	Mutants' creation time (without compilation) 6	Mutants' creation time (with compilation) 7	
	MIN	AVG	MAX	MIN	AVG	MAX	MIN	AVG	MAX				
8 Difference									9 Normalized			10 Analysis Quality	11 Normalized Analysis Quality
MS	Test Case Number	Test Case Execution Time	Mutant Number	Mutant Code Creation Time	Mutants Creation Time	MS Difference	Test Case Number Difference	Mutant Number Difference					

Rys. 23 Kolumny arkusza z danymi niezależne od wybranej analizy

1.9.5.3. Arkusz z wykresami wizualizującymi zebrane dane

Do każdego z raportów całościowych dołączone są arkusze z wykresami. Wizualizują one dane zebrane we wcześniejszych arkuszach. Przykład jednego z wykresów zaprezentowano na Rys. 24.



Rys. 24 Przykład wykresu z raportu całościowego

1.10. Praktyczne wskazówki dla użytkowników narzędzia CREAM

- Jeżeli chcesz generować duże zbiory mutantów, wykorzystaj do tego celu repozytorium SVN, a nie dysk lokalny.

- Aby nie generować nadmiarowych mutantów z punktu widzenia dalszych analiz, korzystaj z opcji tworzenia mutantów jedynie z pokrytego kodu.
- Nie generuj mutantów z projektów będących testami jednostkowymi.
- Pamiętaj o wyborze odpowiedniego narzędzia do przeprowadzania testów jednostkowych wykorzystywanego w procesie testowania mutantów.
- Długotrwały proces testowania mutantów możesz rozbić na kilka mniejszych, po zakończeniu których wyniki połączyć można za pomocą narzędzia *Data Viewer*.
- W przypadku gdy użytkownik chce korzystać z plików .coverage w celu zaimportowania informacji o pokryciu, przed wczytaniem ich do narzędzia powinien wygenerować je na własnym komputerze.

[1] .NET Framework v.4.0., <http://www.microsoft.com/downloads/en/details.aspx?FamilyID=9cfb2d51-5ff4-4491-b0e5-b386f32c0992&displaylang=en>, z dnia 2010-11-23

[2] NUnit, <http://www.nunit.org>, z dnia 2010-10-09

[3] NCover, www.ncover.com, z dnia 2010-11-23

[4] K. Sarba - „Automatyzacja procesu testowania programów i analizy statystycznej”, praca magisterska, Politechnika Warszawska, Instytut Informatyki, 2007

[5] SVN, <http://subversion.tigris.org/>, z dnia 2011-01-27

[6] CREAM Website, <http://galera.ii.pw.edu.pl/~adr/CREAM/>, z dnia 2011-09-06

[7] A. J. Offutt, G. Rothermel, and C. Zapf, “An Experimental Evaluation of Selective Mutation,” in *Proceedings of the 15th International Conference on Software Engineering (ICSE '93)*. Baltimore, Maryland: IEEE Computer Society Press, May 1993, pp. 100–107.