

TESSY V4.2 Features

New features in TESSY V4.2 (compared to TESSY V4.1)

Frank Büchner, Oktober 2019 --- 002

Contents

1	Classification Tree Editor (CTEX)	2
1.1	Creating an Initial Classification Tree	2
1.2	Creating Test Cases	3
2	Call Pair Coverage	4
3	New Metrics and Estimation of Testing Effort	5
4	Component Testing for C++	7
5	Tasks	8
6	The Author	8

1 Classification Tree Editor (CTEX)

The Classification Tree Editor (CTEX) can now automatically generate an (initial) classification tree from the interface of the test object and test cases from classes using the “all pairs” method.

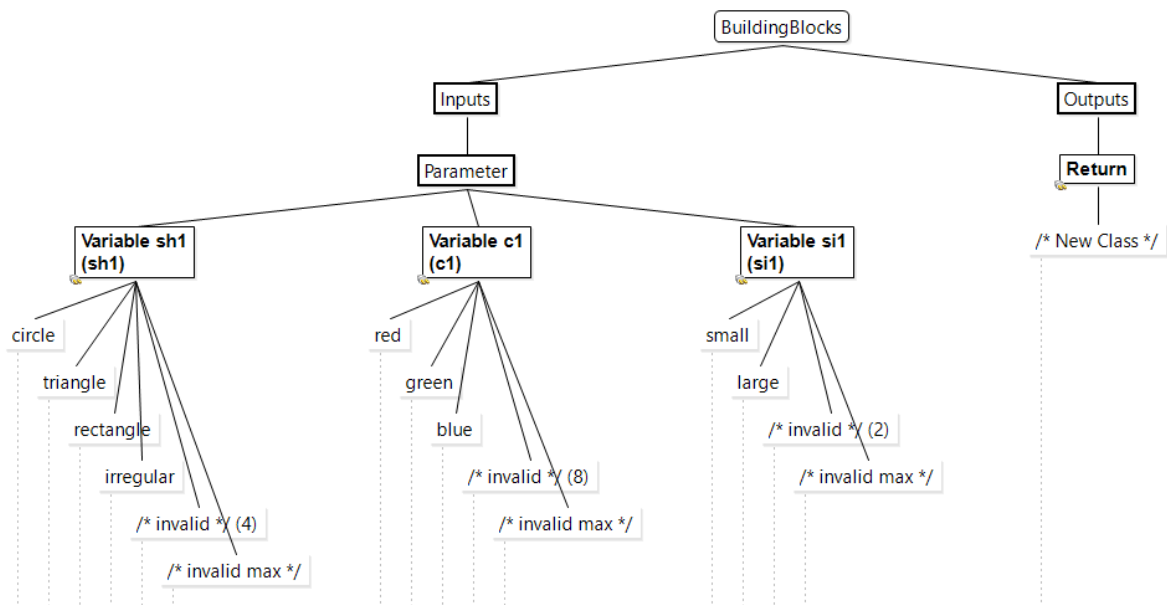
1.1 Creating an Initial Classification Tree

The following source code

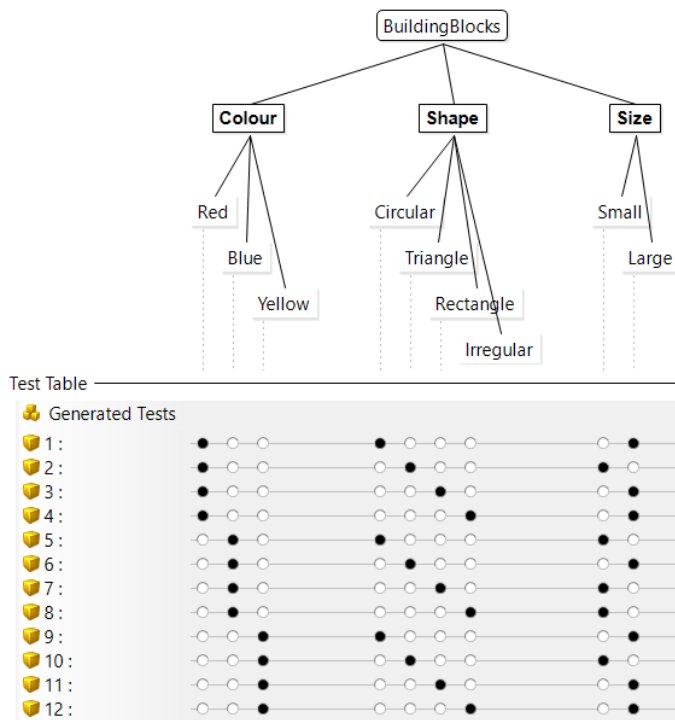
```
typedef enum {circle, triangle, rectangle, irregular} shape;
typedef enum {red = 5, green, blue} colour;
typedef enum {small, large} size;

int BuildingBlocks(shape sh1, colour c1, size si1)
{
    return sh1+c1+si1;
}
```

results in the following tree



1.2 Creating Test Cases



The test case specifications were automatically created from the classification tree using the “all pairs” method.

2 Call Pair Coverage

Call Pair Coverage (aka Call Coverage) is a code coverage measure for integration testing. Its value indicates if a certain function was called from all location in the source code where a call to this function is implemented.

```

int f0(void)
{
    return(divide(7,3));
}

int f1(void)
{
    return(divide(7,0));
}

int divide(int a, int b)
{
    return(a/b);
}

```

Fig. 1: The function divide() can be called from f0() and f1()

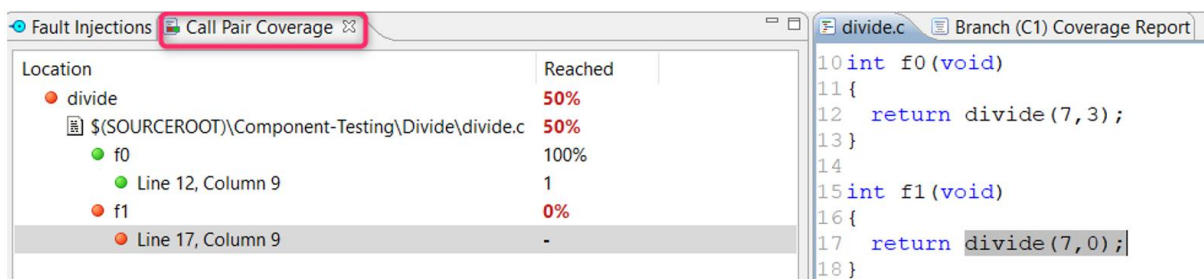


Fig. 2: Call pair coverage for divide() has only reached 50%

In the example above, the function divide() can be called from f0() and f1(). If divide() is called from f1(), a division by zero occurs. The figure above shows the result of call pair coverage measurement. One test case was executed. In this test case, divide() was called from f0(). The call from f1() was not executed (and the division by zero did not take place). To get 100% call pair coverage, the call from f1() needs to be executed also, what would reveal the division by zero.

3 New Metrics and Estimation of Testing Effort

TESSY V4.2 measures additional metrics, namely

- Statement Count – the number of statements in a test object
- Call Count – the number calls to other units in a test object
- Maximum Depth – the nesting level of the test object

The new metrics and existing metrics can be used to calculate the estimated time for testing. The formula for the estimated time is not fixed, but can be specified by the user if the default by TESSY seems not appropriate.

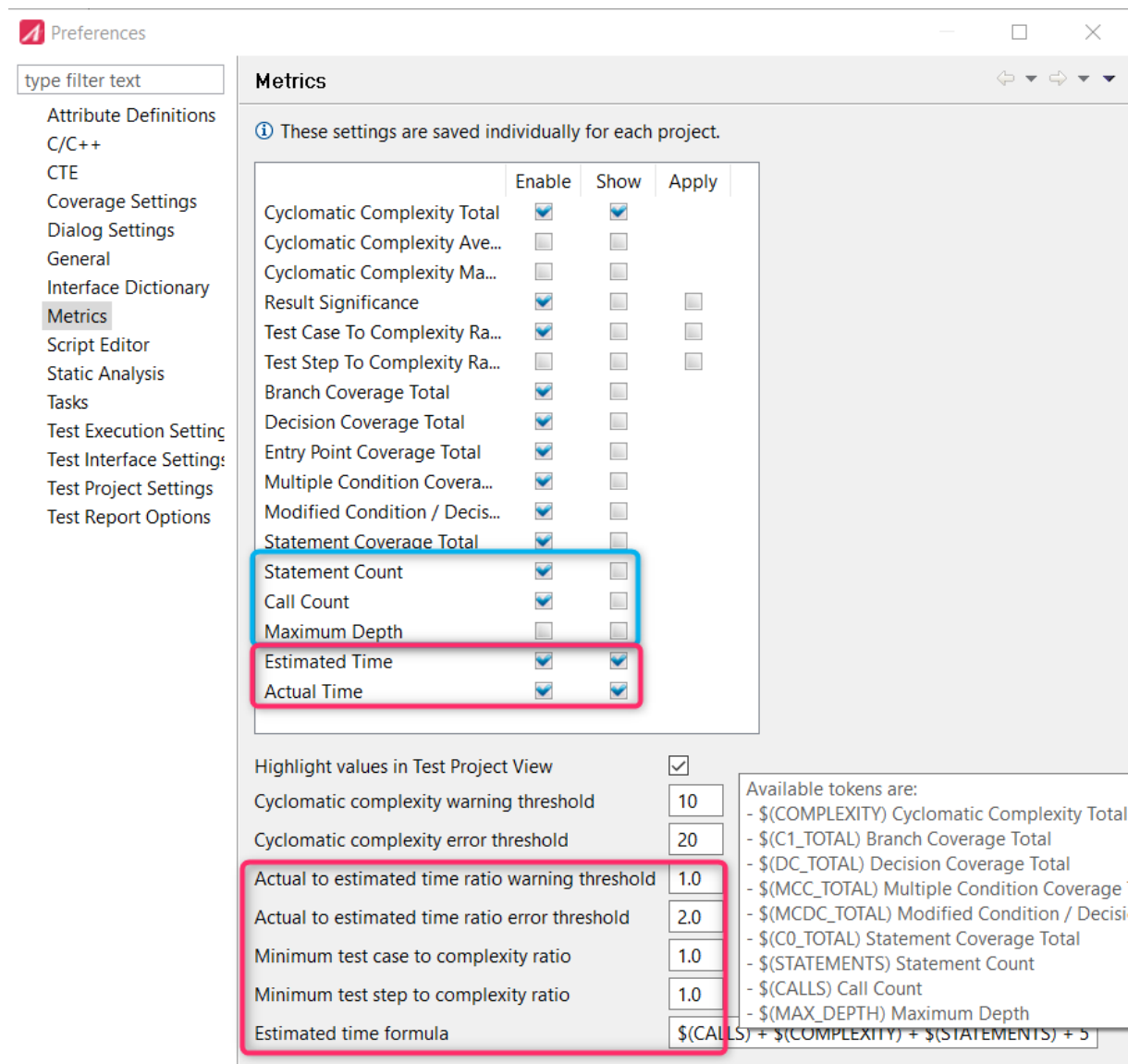


Fig. 3: Three additional metrics (blue) and test effort estimation (red)

It is possible to set thresholds for warning and error.

The estimated time and the time actually needed can be displayed in the project view of TESSY. Of course, you need to measure and set the actual time manually. Depending on the set thresholds, the actual time is displayed in green, yellow, or in red.

	CC	TC/C	RS	ET	AT
Hitex-Examples	608	0.00	-	-	1h10m
BatchRestore	28	0.80	✓	1h41m	1h10m
Categorize	5	0.80	✓	17m	-
categorize	5	0.80	✓	17m	
Triangle	23	0.80	✓	1h24m	1h10m
is_equilateral	4	1.50	✓	17m	
is_isosceles	5	1.20	✓	18m	
is_right	5	1.40	✓	18m	24m
is_scalene	5	0.80	✓	18m	16m
is_triangle	4	1.00	✓	13m	30m

Fig. 4: Estimated and actual time in the project view in TESSY

4 Component Testing for C++

Integration testing / component testing is now possible for software written in C++ also.

```
stack.push(x=1)
stack.push(x=2)
stack.push(x=3)
stack.pop() == 2
stack.pop() == 2
```

Actual Value: 3, Expected Value: 2

Fig. 5: A test case (scenario) testing the test object “stack” written in C++

The scenario in the figure above is a test case for the abstract data type “stack”. First, the values 1, 2, and 3 are pushed on the stack. The first pop operation expects 2 as the value to be taken from the stack. Because prior to the pop operation the value 3 was pushed on the stack, the expectation fails. Therefore, the scenario fails with respect to the first pop operation. However, the second pop operation passes, because after 3 was removed from the stack, 2 is the value at the top of the stack, what is the expected value of the second pop operation.

5 Tasks

You can now create tasks, which are associated to folders. This allows including external tasks like reviews or system tests in TESSY. If a task was executed, you can set their state to passed / failed, depending on the outcome.

You can link requirements to these tasks. The task counts as one test for the linked requirements, i.e. the requirements report will indicate if all task (which are linked to a requirement) are passed or failed.

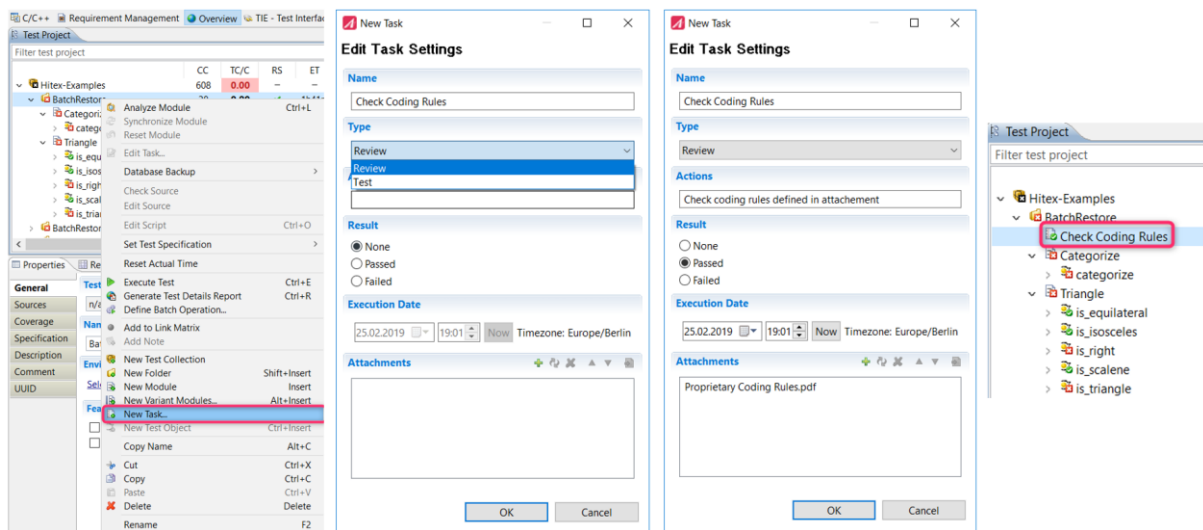


Fig. 6: Task in TESSY

6 The Author

Frank Büchner, Hitex GmbH, frank.buechner@hitex.de



Any comments or questions to this document are welcome.