



Test Coverage

Eric Heim

Bugsquashing Seminar

Code Coverage

Definition:

“Code coverage is a measure used in software testing to describe the degree the source code of a program has been tested.”

Source: [1]

Coverage criterias / metrics

- **Function coverage:**

- Has each function been called?

- **Statement coverage:**

- Has each executable statement been executed?

- **Decision coverage:**

- Has each if else statement of a program been evaluated true and false?

- **Condition coverage:**

- Has each boolean condition been evaluated true and false?

Coverage criterias / metrics

And many more: multiple condition coverage, loop coverage, race coverage, path coverage, etc. ...

Coverage with CMake

- **Available on Linux with gcov**
 - Uses Statement coverage
- **Enabled in Debug builds via compiler flags:**

```
SET(CMAKE_CXX_FLAGS="-g -O0 -fprofile-arcs -ftest-coverage")  
SET(CMAKE_C_FLAGS="-g -O0 -Wall -W -fprofile-arcs -ftest-coverage")  
SET(CMAKE_EXE_LINKER_FLAGS="-fprofile-arcs -ftest-coverage")  
target_link_libraries(MyExe gcov)
```

- **Run Coverage Tests:**
 - In Script with **ctest_coverage()**
 - By dartclient target, e.g. **ExperimentalCoverage**

Coverage in MITK

- Enable Coverage in MITK via Cmake Flag in Dartclient Script:
 - DWITH_COVERAGE:BOOL=TRUE
- Results on the dashboard:

```
229 |  
230 9 void mitk::DataNodeFactory::ReadFileSeriesTypeDCM()  
231 | {  
232 9 const char* previousCLocale = setlocale(LC_NUMERIC, NULL);  
233 9 setlocale(LC_NUMERIC, "C");  
234 | std::locale previousCppLocale( std::cin.getloc() );  
235 9 std::locale l( "C" );  
236 9 std::cin.imbue(l);  
237 |  
238 9 if ( DicomSeriesReader::IsPhilips3DDicom(this->GetFileName()) )  
239 | {  
240 0 MITK_INFO << "it is a Philips3D US Dicom file" << std::endl;  
241 0 this->ResizeOutputs(1);  
242 0 DataNode::Pointer node = this->GetOutput();  
243 | mitk::DicomSeriesReader::StringContainer stringvec;  
244 0 stringvec.push_back(this->GetFileName());  
245 0 if (DicomSeriesReader::LoadDicomSeries(stringvec, *node))  
246 | {  
247 0 node->SetName(this->GetBaseFileName());  
248 | }  
249 0 setlocale(LC_NUMERIC, previousCLocale);  
250 0 std::cin.imbue(previousCppLocale);  
251 9 return;  
252 |  
253 }
```

Conclusion

What coverage can do :

- ✔ Coverage can tell you which part of your code is covered by tests, and which is not

What coverage can't do :

- ✖ It tells nothing about the quality of the tests
- ✖ No guarantee for a bug free program
- ✖ Other techniques like mutation testing are needed to ensure the quality of your tests



Questions?

Sources

- I. The Art of Software Testing, ISBN 0-471-46912-2.
- II. <http://www.bullseye.com/coverage.html>
- III. <http://gcc.gnu.org/onlinedocs/gcc/Gcov.html>

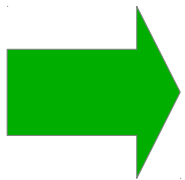
Statement coverage

- The code is only fully covered when the condition is **true**
- If the condition is **false** `foo` isn't executed

Example 1:

```
int p = 0;  
if(condition)  
    p = foo();  
bar(p);
```

What happens when there is a bug when **foo()** is not executed?



The bug can only be found with partial statement coverage !

Decision coverage

- Decision coverage can solve the problem of the previous statement coverage example

Example 2:

```
if(cond1 && (cond2 || func()))  
    foo();  
else  
    bar();
```

- Decision coverage can be fulfilled without executing func()

Consider:

- cond1 = false
- cond1 = true, cond2 = true

 **if-statement is false**

 **if-statement is true**

Source: [I][II]

Condition coverage

- The weakness of decision coverage can be solved by condition coverage
- But condition coverage can't guarantee full decision coverage
- Example 3 can never fulfill decision coverage:
 - **if-statement** is always false

Example 3:

```
if((a && b) ? false : false)
```