Definitions

- "Testing is the process of executing a program with the intent of finding errors" [Myers]
- During **glass box testing**, the execution of code elements is recorded:
  - Statement, branch, condition, loop ...
  - The complete set of program elements is known and finite
- **Glass box testing = white box testing = coverage testing**
- **Test coverage** is the degree to which the complete set of program elements is executed.
- A **test case** consists of execution conditions, input data, and expected results.
- A **test suite** is a set of test cases.
## Glass Box Testing Tools

<table>
<thead>
<tr>
<th>Product</th>
<th>Vendor</th>
<th>Language</th>
<th>SC</th>
<th>BC</th>
<th>Licence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agitar</td>
<td><a href="http://www.agitar.com">www.agitar.com</a></td>
<td>Java</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Bullseye</td>
<td><a href="http://www.bullseye.com">www.bullseye.com</a></td>
<td>C/C++</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>CC Analyzer</td>
<td><a href="http://www.ccaseconsult.de">www.ccaseconsult.de</a></td>
<td>C/C++, COBOL</td>
<td>x</td>
<td></td>
<td>GPL</td>
</tr>
<tr>
<td>Clover</td>
<td><a href="http://www.cenqua.com/coverlipse">www.cenqua.com/coverlipse</a></td>
<td>Java, .net</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Cobertura</td>
<td>cobertura.sourceforge.net</td>
<td>Java</td>
<td>x</td>
<td>x</td>
<td>GPL</td>
</tr>
<tr>
<td>CodeCover</td>
<td><a href="http://www.codecover.org">www.codecover.org</a></td>
<td>Java, COBOL</td>
<td>x</td>
<td>x</td>
<td>EPL</td>
</tr>
<tr>
<td>coverlipse</td>
<td>coverlipse.sourceforge.net</td>
<td>Java</td>
<td>x</td>
<td></td>
<td>GPL</td>
</tr>
<tr>
<td>Dynamic</td>
<td><a href="http://www.dynamic-memory.com">www.dynamic-memory.com</a></td>
<td>C/C++</td>
<td>x</td>
<td></td>
<td>CPL</td>
</tr>
<tr>
<td>EMMA</td>
<td>emma.sourceforge.net</td>
<td>Java</td>
<td>x</td>
<td></td>
<td>CPL</td>
</tr>
<tr>
<td>gcov</td>
<td>gcc.gnu.org/onlinedocs/gcc-3.0/gcc_8.html</td>
<td>C/C++</td>
<td>x</td>
<td></td>
<td>GPL</td>
</tr>
<tr>
<td>Insure++</td>
<td><a href="http://www.parasoft.com">www.parasoft.com</a></td>
<td>C/C++</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J Test</td>
<td><a href="http://www.parasoft.com">www.parasoft.com</a></td>
<td>Java, .net</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J Cover</td>
<td><a href="http://www.mmsindia.com/JCover.html">www.mmsindia.com/JCover.html</a></td>
<td>Java</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Koalog</td>
<td><a href="http://www.koalog.com/php/kover.php">www.koalog.com/php/kover.php</a></td>
<td>Java</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDRA Testbed</td>
<td><a href="http://www.ldra.co.uk/software/testing.asp">www.ldra.co.uk/software/testing.asp</a></td>
<td>C/C++, Ada, COBOL</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Semantic Designs</td>
<td><a href="http://www.semidesigns.com">www.semidesigns.com</a></td>
<td>C/C++, C#, PHP, COBOL</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCAT</td>
<td><a href="http://www.soft.com/Products/index.html">www.soft.com/Products/index.html</a></td>
<td>C/C++, Java, Ada</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Telelogic Logiscope</td>
<td><a href="http://www.telelogic.com/products/logiscope">www.telelogic.com/products/logiscope</a></td>
<td>C/C++, Java, Ada</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Testwell CTC++</td>
<td><a href="http://www.verifysoft.com">www.verifysoft.com</a></td>
<td>C/C++</td>
<td>x</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SC = Statement coverage   BC = Branch coverage

## The Benefits of Glass Box Testing

1. **Testing adequacy metric**
   Coverage is an objective adequacy metric which can be used for example as a test completion criterion.

2. **Test suite extension**
   The glass box test denotes the program elements which were not executed.

3. **Test suite reduction**
   Removing (redundant) test cases from a test suite to reduce regression testing effort without (significantly) decreasing testing effectiveness.

4. **Basis for selective regression testing**
   Instead of „rerun-all“ in regression testing, only those test cases are selected that were „involved“ in the code modification.

5. **Support for program comprehension**
   The glass box test denotes which program code is executed by which test case (traceability).
When test coverage had not previously been measured, testers tended to overestimate coverage of their test cases. The first time testers measured coverage during function test, they found that the coverage was in the range of 50% to 60%. The testers were surprised at the low percentage of coverage they were getting. They expected a much higher percentage of code coverage. Some testers estimated that their coverage was 90% or higher.

Glass Box Testing Process

1. Functional Test (black box test)
2. Correction of detected errors
3. Repeated functional Test, with activated glass box testing tool

4. Analysis of the unexecuted code
   - Determine input data for new test cases
   - Determine expected results
   - Add the new test cases to the test suite

5. Execute the new test cases with activated glass box testing tool
Test Case Selective Glass Box Testing

- Most glass box testing tools on the market accumulate the coverage of the sequentially executed test cases.
  - In this case, there is no test case individual analysis or reporting possible.

- In contrast, test case selective glass box testing handles coverage data for each test case of a test suite.
  - Analysis and reporting for each test case is possible
  - But: the glass box testing tool is not automatically able to separate the coverage information into distinct test cases. Manually notification of the glass box testing tool of begin and end of each test case is necessary.

CodeCover

- CodeCover is a free glass box testing tool developed in 2007 at the university of Stuttgart
- Licence: Eclipse Public Licence (EPL)
- CodeCover measures statement, branch, loop, and MC/DC coverage
- Command line and Eclipse integration
- Analysys ans coverage report per test case
- Open language interface, available languages: Java and COBOL
- www.codecover.org
CodeCover: Identification of test cases

- The tester selects „Start Test Case“ before starting the input of the test input data.
- The tester selects „End Test Case“ after the test case is completely executed, and the test results are verified.
- For Java applications, CodeCover uses the Java JMX technology to notify begin and end of a test case. The JMX server is added to the SUT during instrumentation.

![Diagram showing JMX-Messages for test case begin and end](image)

CodeCover: Coverage Report

- Coverage Report: statement, branch, loop, decision
- Code highlighting
- Executed test cases
**CodeCover: Guidance to find missing Test Cases**

- **Area A**: Executed by the listed test cases
- **Area B**: Unexecuted; **Guidance**: Find a new test case, based on
  - Test cases from area A
  - Predicate of the Condition statement (area B)

**CodeCover: Support for Test Suite Reduction**

- CodeCover visualizes the degree of unique executed program elements pairwise for all test cases.
- Export capability of the table data, especially for huge test suites (e.g., automatically generated test suites or test data from production databases)

| T1 contains 99.9% of all statements of T2 |
| T5 contains 99.5% of all statements of T6 and 97% of T7 |
Conclusion

- Test case selective glass box testing has advantages
  - Support for test suite reduction
  - Guidance to find missing test cases

Thank you for your attention!