

Rainer Schmidberger

Universität Stuttgart, Institut für Softwaretechnologie

Glass Box Testing for Test Suite-Optimization

- ❑ Definitions for testing and glass box testing
- ❑ The benefits of glass box testing
- ❑ The glass box testing process
- ❑ Test case selective glass box testing
- ❑ A new glass box testing tool: CodeCover
- ❑ Conclusion

se

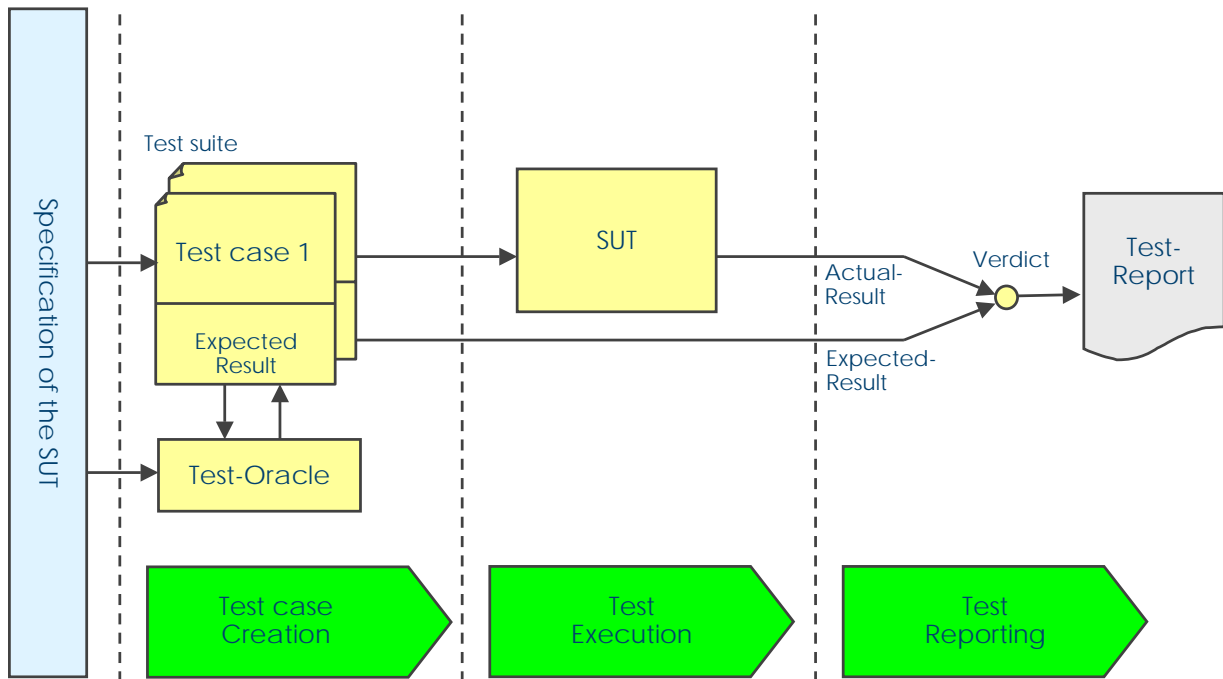
Definitions

- ❑ „**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.

Functional Testing Data Flow

04/03/2008 © 2008, Rainer Schmidberger

SE2008

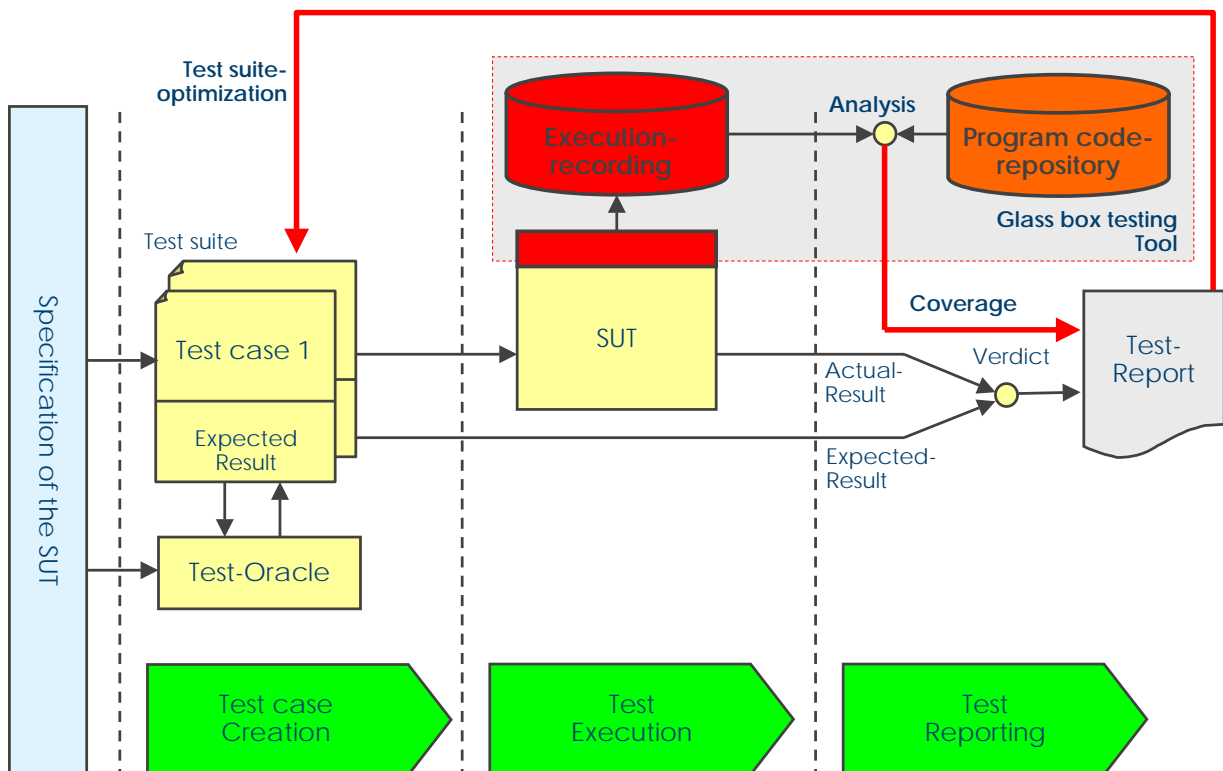


Folie 3 / 15

Glass Box Testing Data Flow

04/03/2008 © 2008, Rainer Schmidberger

SE2008



Folie 4 / 15

Glass Box Testing Tools

Product	Vendor	Language	SC	BC	Licence
Agitar	www.agitar.com	Java	x	x	
Bullseye	www.bullseye.com	C/C++	x	x	
CC Analyser	www.caseconsult.de	C/C++, COBOL, Java, PL/1	x		
Clover	www.cenqua.com/clover	Java, .net	x	x	
Cobertura	cobertura.sourceforge.net	Java	x	x	GPL
CodeCover	www.codecover.org	Java, COBOL	x	x	EPL
coverlipse	coverlipse.sourceforge.net	Java	x		CPL
Dynamic	www.dynamic-memory.com	C/C++	x	x	
EMMA	emma.sourceforge.net	Java	x		CPL
gcov	gcc.gnu.org/onlinedocs/gcc-3.0/gcc_8.html	C/C++	x		GPL
HP Code Coverage Tool Software	h71028.www7.hp.com/ERC/downloads/4AA0-7120ENW.pdf	C/C++, epTAL, ECOBOL	x	x	
Insure++	www.parasoft.com	C/C++	x	x	
JTest	www.parasoft.com	Java, .net	x	x	
JCover	www.mmsindia.com/JCover.html	Java	x	x	
Koalog	www.koalog.com/php/kover.php	Java	x		
LDRA Testbed	www.ldra.co.uk/softwaretesting.asp	C/C++, Ada, COBOL	x	x	
Rational PurifyPlus	www-306.ibm.com/software/awdtools/purifyplus	C/C++, Java, .net	x	x	
Semantic Designs	www.semdesigns.com	C/C++, C#, PHP, COBOL,	x	x	
TCAT	www.soft.com/Products/index.html	C/C++, Java, Ada	x	x	
Telelogic Logiscope	www.telelogic.com/products/logiscope	C/C++, Java, Ada	x	x	
Testwell CTC++	www.verifysoft.com	C/C++	x	x	

SC = Statement coverage BC = Branch coverage

Folie 5 / 15

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).

Folie 6 / 15

Piwowarski, P., Ohba, M., and Caruso, J. „Coverage measurement experience during function test“, *Proceedings of the IEEE 15th international Conference on Software Engineering, 1993*

[...]

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
 6. Continue with 4.

Piwowarski et al.	50% - 60%
Empirical data (J2EE-System „ecadia“)	35%

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.

Folie 9 / 15

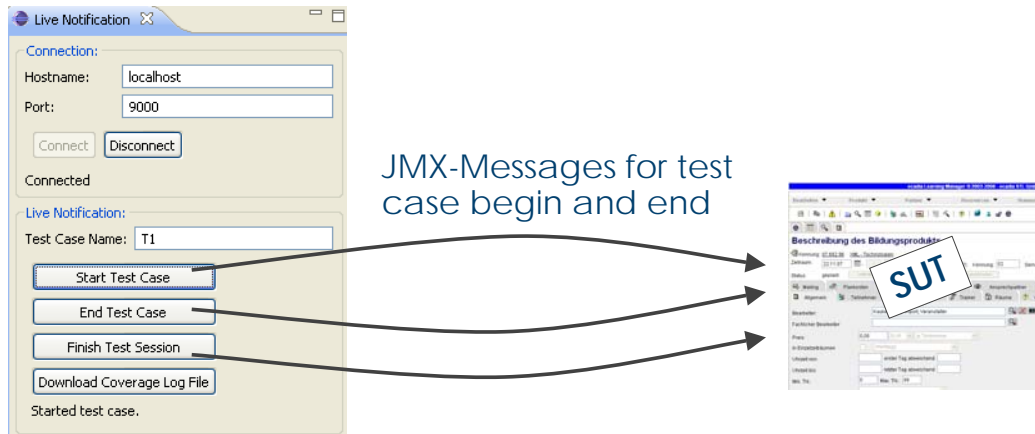
CodeCover

- ❑ 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

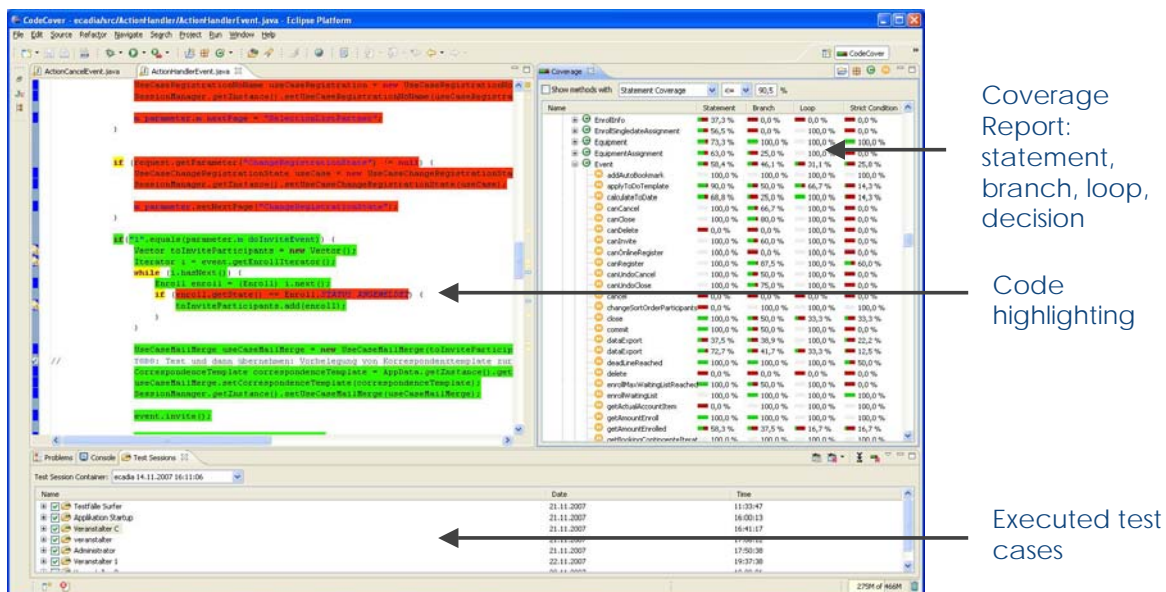
Folie 10 / 15

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.



CodeCover: Coverage Report



Coverage Report: statement, branch, loop, decision

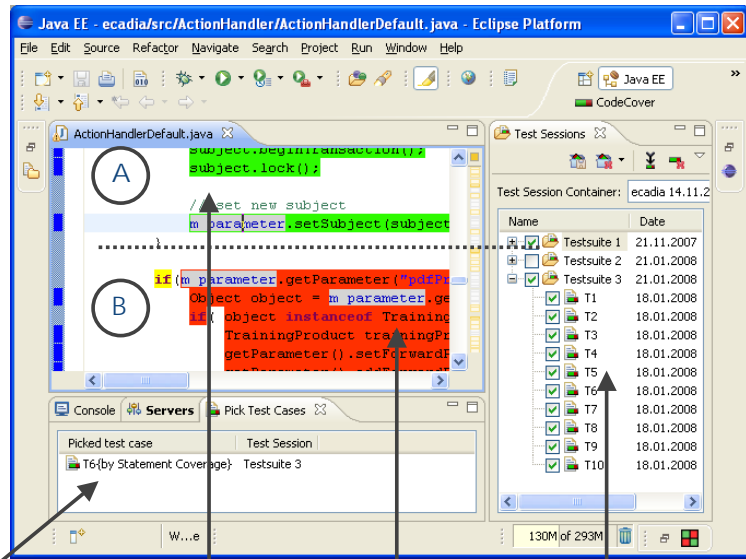
Code highlighting

Executed test cases

CodeCover: Guidance to find missing Test Cases

04/03/2008 © 2008, Rainer Schmidberger
SE2008

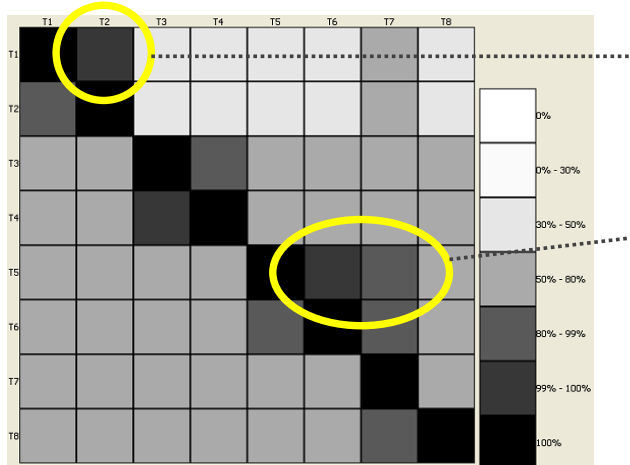
- (A) Area A: Executed by the listed test cases
- (B) Area B: Unexecuted; **Guidance:** Find a new test case, based on
 - Test cases from area A
 - Predicate of the Condition statement (area B)



Test cases that execute the selected statement Selected Statement Unexecuted code All executed test cases

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

04/03/2008 © 2008, Rainer Schmidberger
SE2008

Conclusion

- ❑ Test case selective glass box testing has advantages
 - ⇒ Support for test suite reduction
 - ⇒ Guidance to find missing test cases
- ❑ www.codecover.org and www.sourceforge.net



Thank you for your attention!