Component-Based White-Box Testing

By Prathab K

(Roll No: 06329902)

Under the guidance of

Prof. Deepak B. Phatak



Outline

- Component-based software development
- Software component testing
- White-box testing methods for software components
- Systematic procedure for white-box testing of software components
- Conclusions & Future work



Component-based Software Development



Software Components

• What is software component?

- confirms to a component model
- independently deployable
- composed according to a composition standard (EJB, CORBA, or COM+)





Engineering process for software components

• Component-based software development involves:

- Component qualification
- Component adaptation
- Assembling components into systems
- System evolution





Software Component Testing



Testing software components

Involves

- Testing individual components
- Testing interaction among components

• Necessity

- Inconsistent infrastructure and environment
- Inconsistent interaction model

• Challenges

- Lack of source code availability
- Performance and reliability analysis
- Test Adequacy criteria



Methods to increase component testability

Component testability

- degree of component facility for component testing



[4] Stocks, P. A., and Carrington, D. A, "A Framework for Specification-Based Testing"
[5] Wang, Yingxu, King, Graham, and Wickburg, Hakan, "A Method for Built-in Tests in Component-based Software Maintenance"
[6] Gao, J., et al., "On Building Testable Software Components"



White-box Testing Methods for Software Components



White-box testing

- "Testing that takes into account the internal mechanism of a system or component"
- White-box vs Black-box testing?
 - Black-box testing against the specification
 - White-box testing against the implementation
- Is white-box = unit testing?
 - Not exactly!



Flow graph notation

• Flow graph notation represents the structure of the program



Path testing

- To ensure all independent paths through a code are tested
- Coverage factors:
 - Statement coverage
 - Branch coverage
 - Multiple-condition coverage
 - Path coverage
 - Loop coverage

Independent paths

- McCabe's cyclomatic measurement
 - v(G) = e n + p
 - e Edges
 - n Nodes
 - p no. of connected components



Data flow and Object-oriented testing

• Data flow testing

- Observing the lifecycle of a particular data (variable)

• Object-oriented testing

Test the OO features like inheritance, polymorphism, binding coverage, state-based testing



Issues & Challenges in testing software components

- Components developed in a context and deployed in another context
- More expensive to fix the faults, after component delivery
- Identifying the possible external and internal scenarios for a software component
- Applying strict test adequacy criteria



Systematic procedure for white-box testing of software components



Is white-box testing ignored?

- White-box testing requires high expenses in terms of time and resources
- Test code is greater than the development code
- Developers are confident on their code
- Developers tends to overlook their faults, during unit testing
- Sometimes white-box testing is done for the sake of organizational process
- Criticism causes quarrel among the team members



White-box testing tools

- Wide variety of white-box testing tools are available
- Testing Framework
 - JUnit, NUnit,...
- Test case generation tools
 - JTest, TBrun,...

Test coverage tools

- Clover, Cobertura,...



Does coverage is enough?

- Objects and variables may be in many states
- Presence of conditionals creates multiple paths of execution
- Does not validates the implementation behavior (logic)



Shrinking the development time

- Modern IDEs and Tools shrinks the development time
- Most of the development time wasted in regression errors
- Late detection of regression error costs more
 - more code changes had happened
 - developer forgot the context
 - developer spend more time in building new code on top of faulty code



Continuous testing

- Idea of continuous testing^[9], uses real-time integration with the development environment to asynchronously run tests
- Achieved by constructing safe asynchronous model test execution is done on current code version and feedback is provided to the developer
- Resembles to TOTE (Test-Operate-Test-Exit) model of cognitive behavior
- Continuous testing reduces wasted time by 92-98%, over other approaches



[9] David Saff, Michael D. Ernst, "Reducing wasted development time via continuous testing"



Who watches the watchman?

- Any form of observation is also an interaction, that the act of testing can also affect that which is being tested
- Members of the <u>"context-driven" school of testing</u> believe that there are no "best practices" of testing
- Good software testing is a challenging intellectual process



Re-test model for white-box testing

• Problems

- Identify the component's change and analyze their impacts to its component test suite
- A cost-effective method for test suite updates and maintenance

• Solution: Re-test model

- To identify the relationships between components at unit level
- To identify the reusable test cases and to define re-test criteria
- Facilitate automatic test generation and adaptation to the modified component

• Component Function Access Graph (CFAG) model^[10]

view depicting the dependency and data flow among the functions within a component

• Dynamic-CFAG^[11]

dynamic view of the function invocation by a test case



Mapping of test cases to development code

- Mapping of test cases to development code at statement level helps:
 - to have an effective white-box testing activities
 - to identify the regression test suite (especially for continuous testing)
 - to study the test suite impact analysis
 - prioritizing the test cases



23

Conclusions

- White-box testing contributes the major effort towards the component quality assurance
- IDEs and Tools should assist the developer to form a quality test suite for an effective white-box testing
- To develop a tool which will aid the developer to perform an efficient white-box testing process by collaborating with the existing tools and utilizing the discussed concepts



References

- [1] Heineman, G. T., and W. T. Councill, (eds.), "Component-Based Software Engineering: Putting the Pieces Together", Reading, MA:Addison-Wesley, 2001.
- [2] Jerry Zeyu Gao, H.S. Jacob Taso, Ye Wu "Testing And Quality Assurance For Component Based Software", Artech House Boston, 2003.
- [3] Gao, J., "Challenges and Problems in Testing Software Components", Proc. of ICSE2000's 3rd International Workshop on Component-Based Software Engineering: Reflects and Practice, Limerick, Ireland, June 2000.
- [4] Stocks, P. A., and Carrington, D. A, "A Framework for Specification-Based Testing", IEEE Transaction on Software Engineering, Vol. 22 (11), IEEE Computer Society Press (1996), 777-794.
- [5] Wang, Yingxu, King, Graham, and Wickburg, Hakan, "A Method for Built-in Tests in Component-based Software Maintenance", Proceedings of the 3rd European Conference on Software Maintenance and Reengineering, (1998).
- [6] Gao, J., et al., "On Building Testable Software Components", Proc. of 1st International Conference on Cost-Based Software System, 2002, pp. 108-121.



References

- [7] McCabe, T., "A Complexity Measure", IEEE Transactions on Software Engineering, December 1976.
- [8] <u>http://www.testingfaqs.org/t-unit.html</u> Unit Testing Tools
- [9] David Saff, Michael D. Ernst, "Reducing wasted development time via continuous testing". In 14th International Symposium on Software Reliability Engineering, pages 281 - 292, 2003
- [10] Jerry Gao, Raquel Espinoza, Jingsha He, "Testing Coverage Analysis for Software Component Validation". In Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC'05), 2005
- [11] J. Gao, D. Gopinathan, Quan Mai, Jingsha He, "A Systematic Regression Testing Method and Tool For Software Components". In Proceedings of 30th Annual International on Computer Software and Applications Conference, Volume 1, pages 455 - 466. IEEE Computer Society, September 2006.

