WHITE BOX TEST TOOLS: A COMPARATIVE VIEW

Dilara Ateşoğulları, Alok Mishra

Atilim University
e-mails: dilaraatesogullari@gmail.com, alok.mishra@atilim.edu.tr
Turkey

Abstract: There are multiple software testing techniques in the literature. These are white, black and gray box test techniques. Each of these has different tasks and purposes. Technical selection is decisive for these tasks and objectives. In this paper, information about white box technique and approaches are described. In addition, multiple test instruments that are currently in use are examined and a comparative view is provided. As a result of this comparison, a tabular chart is presented with an information set. This information set is intended to be valuable for software testing professionals and researchers to appreciate many tools at a glance.

Key words: Testing, Tools, White Box Testing, White Box Testing Types, Coverages.

1. INTRODUCTION

Software testing is a process of executing a program with intent of finding errors [45]. Software testing is viewed as significant phase in software engineering life cycle as its objective is to improve the quality of software under development [26]. Software testing can spend about 50%-80% of the total cost of software development [29]. Software testing is a significant activity that should be conducted by developers before the software is ready to launch or implement to ensuring whether software under development meets customer's requirement [26]. Software testing facilitates the quality of the software under development [26]. Software testing process use test techniques such as white boxes and black boxes. These test techniques can be applied by multiple methods. Among these techniques, the white box technique is of greater importance. According to International Software Testing Qualifications Board (ISTQB), which is the testing authority, white-box testing is defined as following [27].

- **white-box testing**: Testing based on an analysis of the internal structure of the component or system [27].
- **white-box test design technique**: Procedure to derive and/or select test cases based on an analysis of the internal structure of a component or system [27].

The Institute of Electrical and Electronics Engineers (IEEE), specifies white box testing as follows.

"Testing that takes into account the internal mechanism of a system or component" IEEE [28].

White box testing is one of the most important and prevalent software techniques and is very effective in validating design, decision, assumptions and finding programming errors and implementation errors in software [22]. There are multiple methods used in white box tests. These methods should be selected and applied according to the situation of the project. In the application of these techniques, people can benefit from white box testing tools. These testing tools provide many different features and attributes. Advantages of white box testing are such as revealing error in hidden code, its side effects are beneficial, and it helps in removing extra lines of code [25].

White box testing is verification and validation technique which software engineers can ensure to examine their code's working [22]. He further concluded that using white box testing a software engineer can design test cases that exercise internal data structure to ensure their validity, independent paths within a module, to execute loops and their boundaries and within their operational bounds, examine logical decisions on both their true and false side.

The objective of the paper is to provide comparative view of important and popular white box testing tools. Testing is a significant phase of software and information system development and constitute major part of any such project's budget and time allocation due to efforts involved. Comprehensive and rigorous testing is pre stage for ensuring quality assurance in software and information system. Therefore, this paper will contribute towards body of knowledge in this direction and invaluable for both academic researchers and software and information technology professionals. This paper is organized as follows. Section 2 presents brief information about different white box testing methods. In section 3 brief information about various white box testing tools is presented. Section 4 discusses main observations along with similarities and differences of these tools valuable for both researchers and software testing practitioners. Finally, section 5 concludes with future research direction.

2. WHITE BOX TESTING TYPES

There are various types of testing methods for white box testing. Some of these methods are briefly explained in this section.

Branch coverage. Branch testing is an approach, which can cover every option with true and false value, and it can control every statement [2]. Branch coverage is formed by conducting each of the different branches at least once in each decision. With this method, each piece of code has been tested at least once. There will be no untouched code.

Statement coverage. Statement coverage could cover each statement to be executed at least once, this concept reflects the idea that error cannot be revealed without applying the wrong expression [1]. In order to provide the statement coverage, the requirement of each line of code that can be played in the program-based tests should be run at least once. This situation tells us all indicates that the lines of code are available during playback. Statement coverage is a weak criterion, because it is insensitive to some control structures [9].

Decision coverage. The scope of the decision is also called the branch scope. In this form of approach, each statement is executed at least once, and each decision receives at least once all possible results [4]. If there are many cases to be checked, each case is included in the table and a test scope is used to ensure that all these cases are tested.

Condition coverage. The scope of the condition covers this separation in more detail; discovery forces not only the possible consequences of a boolean expression that controls a branch, but also combinations of individual conditions in a compound boolean expression [1]. Condition coverage requires that all "conditions" in a decision take all possible values at least once. However, this does not mean that all the results of the "conditions" must be realized at least once [9].

Decision/condition coverage. The requirement here is to approve and verify that whole conditional statements in each branch will be tested. The status of each sub-expression must be tested at least once [3]. Condition/ Decision coverage combines the requirements of "condition coverage" and "decision coverage". This means that all possible combinations of conditions must be provided, and the decision must be made for all possible values [9].

Multiple condition coverage. To have more than one condition coverage, which each statement is executed at least one times, and all possible result-result combinations in each decision occur at least once. Scope of more than one condition always results in the scope of condition, decision and declaration. Multiple condition scopes are the most stringent structural scope test type [4].

Path coverage. Sometimes an error only occurs when a certain sequence of decisions is used, ie a certain path is used throughout the program. Defining a scope criterion based on full paths rather than individual program decisions is simple (but not practical as we will see) [1].

3. WHITE BOX TESTING TOOLS

This section provides brief information about various white box testing tools available and their comparative view on important attributes is provided in table 1.

Veracode. Veracode white box is one of the test tools, but also stands out in terms of different language support and mobility. It has been observed that this tool has a similar structure with other tools in terms of usage [5]. It works with web interfaces especially in security tests. It is a test tool which can be used for security purposes.

EclEmma. EclEmma, which is used to measure code capacity, is an effective test tool in the automation process. This tool, run as a plugin, can improve code quality with various discoveries in code during unit tests. Report support, which is one of the most important features of EclEmma, makes it different from many other known tools in the market, as well as being open source and supporting different operating systems [6, 16, 19].

RCUNIT. RCUnit is mainly used in the C programming language. With this tool, multiple exceptions and terminal signals can be used during testing. In this process, many test groups and single test scenarios are run. RCUNIT is just one of the rare tools that are under the MIT License, unlike many other tools in the market. In addition to this distinctive feature, different language and operating system support is a crucial feature in the open source [7].

Cfix. CFIX is one of many tools available on the test market. This tool is based on Language Support, Different OS Support and Open source (Free Use) [8]. Operating systems that Cfix can support include Windows (32-bit) and NT kernel mode. [15]

Googletest. The Google test was specifically scanned for the C ++ programming language. Unit tests are mostly used. However, it is also used in different scale tests (regression and acceptance). The Googletest tool provides different browser support, unlike many other products available on the market. It also provides Report support, support for different languages, different OS support, and Opensource (Free Use) features [20].

Nunit. NUnit Microsft is a unit testing tool designed with .NET infrastructure. In this context, it provides multiple features. Some of them have a strong data driven test structure, some tests can be run in parallel. In addition, NUnit can provide many of the features provided by many of the tools available on the market. Different language support, Free use a few of them, but it also has an MIT license which is an important authority. In addition to all of these, it is available in Test Driven Development Support [10, 18].

CppUnit. The CPPUnit test tool was created for use in unit tests and in accordance with the C ++ program language. With this infrastructure, C and C ++ programming languages can be tested easily. In addition, test harness results in XML format is another feature of this tool. CPP Unit is another test tool available on the market. This tool provides support for different language options and free use [11].

Junit. Junit test tool was created for use in unit tests and in accordance with the java program language. This test tool usually provides some convenience in unit tests and test automation. JUnit runs as JAR during compile time. Unlike most tests, JUnit provides test driven development support. In addition to this unique feature, it provides different OS support and Free use [12, 17].

JsUnit. JSUnit, which is one of many test tools, has different browser support and test driven support [13]. JSUnit works on client side browsers. Generally used in test automation. In this process, it can also support different operating system.

SeleniumHQ. Selenium is the well-known test tool. It supports few scripting languages like: Java, C#, Python and Ruby. It supports almost all browsers: IE, Firefox, Chrome, Opera, Safari, Android. People need selenium IDE to Record and Playback [30].

Zeta Test. Zeta Test Tool support reports as HTML documents. It provides compatibility with Citrix servers and Microsoft Terminal servers [31]. With Zeta Test, systems that use multiple 3rd parties can be easily integrated. In addition, Excel, PDF, Word and HMTL format reporting feature is also available. It offers two different languages, German and English.

DotCover. DotCover is a very successful tool for statement coverage based reporting. This property. NET framework is often used. This tool also supports many unit test frameworks [32]. DotCover provides multiple test structures. The most important of these is unit tests. Multiple scope analyzes can be performed via DotCover. It also performs continuous coverage analysis.

TOSCA Testsuite. TOSCA Testsuite End to End is a test tool used in test automations. This tool can use multiple testing approaches. For this reason, it is used in API and GUI tests. It is associated with risk-based and model-based tests. TOSCA Testsuite (tm) is agile based test tool. It can be used as a test management tool that helps the organization of test suites as well as automation tools [33].

Protractor. Protractor is an end to end (E2E) test tool for AngularJS application. It utilizes WebDriverJS frames. Native application and browser specific drivers can implement by protractor. Protractor performs tests on the browser [34].

Windmill. Windmill is a software testing framework for Web applications. It provides different programming language support like Python, JavaScript or Ruby. Windmill supports all popular Web browsers and runs on multi-operating systems [35].

RapidRep Test Suite. The RapidRep Test Suite is a tool for automated back-end testing. This tool can be used for just the functionality of programs [36]. The RapidRep Test Suite can use different logics to distinguish similarities and differences in test runs. Automatic reporting is another useful feature.

Testuff. Testuff is an on-demand test management tool for desktop client application. It provides different features Reports, Test case management, and Requirement management [37]. A nice user interface is intact. It has different languages (English, France, Spanish and German) support as well. The main purpose of this tool is to provide SaaS solutions during testing.

Jbehave. JBehave is a tool for Behaviour-Driven Development (BDD). Behaviour-Driven Development is a top stage of test-driven development (TDD) and acceptance-test driven design. JBhave provides report format in JSON and XML support [38].

Cucumber. Cucumber is a BDD test tool and Frame. The cucumber itself is written in Ruby, but not limited to, other programming languages [39]. Originally a Ruby-based tool, it supports different programming languages such as Java and Javascript. It is especially used in acceptance tests.

Visual Assert. Visual Assert is another useful test tool that allows to handle and manage your C/C++ unit tests. The Visual Assert can work without ever leaving the Visual Studio IDE [40]. Visual Assert is especially used for unit tests. This tool is used in Cfix infrastructure.

Rational Functional Tester [RFT]. The Rational Functional Tester tool is used with ease in automation tests. This tool, developed by IBM, imitates user movements and produces a more realistic test scheme. Specifically designed for regression testing. Multiple web browser support is available. Rational Function Tester supports a good amount of applications. This tool supports Java language and working with Eclipse IDE and Visual Basic .NET and working with Visual Studio .NET. IDE [41].

Ncover Collector. Ncover Collector enables the collection of coverage data on Windows Store Apps through the use of pre-instrumentation. It provides to 32 & 64 Bit Support [42]. Ncover Collector is built using the .NET infrastructure. Code Coverage is an effective tool. The structure of this tool makes the quality team easy to use.

Concordion. Concordion is an open source (free-usage) framework for Java. Concordion is the best tool to create and manage living documentation. The program can utilize Specification by Example (SbE) and Behavior Driven Development (BDD) processes [43]. The purpose of using the Concordion test tool is different from other tools. The purposes of this tool's use include improved readability of documents, more "opinionated" (scripting is actively discouraged) and easier to use.

Cleanscape TestWise. Cleanscape TestWise is a software visualization toolkit for managing and testing C or C++ programs on different platforms [44]. CleanScape Test Wise has many different features. For instance scope analysis, multi-platform support, dependency tracking, reporting etc.

Table 1. White Box Testing Tool Features

Tool Name and Features	*Multi Language Support*	*Mobility*	*Different OS Support*	*Test Coverage Report*	*Free Use (Open source)*	*Different Browser Support*	*Parallel and Concurrent Execution*	*Multi Tool Objectives*	*Test Driven Development Support*	*MIT License*	*Report Support*
Veracode	✓	✓									
EclEmma			✓		✓						✓
RCUNIT	✓		✓		✓					✓	
Cfix	✓		✓		✓						
Googletest	✓		✓		✓	✓					✓
Nunit	✓				✓		✓		✓	✓	
CPPUnit	✓				✓						✓
Junit			✓	✓	✓				✓		
JSUnit						✓			✓		
SeleniumHQ	✓		✓		✓	✓		✓			✓
Zeta Test											✓
dotCover	✓										✓
TOSCA Testsuite	✓							✓			✓
Protractor					✓	✓					
Windmill	✓		✓		✓	✓					
RapidRep Test Suite			✓								✓
Testuff			✓					✓			✓
Jbehave	✓		✓		✓						✓
Cucumber	✓		✓		✓						
Visual Assert					✓						
Rational Functional Tester [RFT]	✓		✓			✓		✓			
Ncover Collector											✓
Concordion	✓				✓						
Cleanscape TestWise	✓		✓					✓			✓

4. DISCUSSION

There are many methods used in white box tests. Some of these methods are Statement Coverage, Condition Coverage, Decision Coverage, Branch Coverage and Path Coverage. In order to work with these methods, multiple approaches and planning are required. There are many white box testing tools available to implement these approaches and plans. Although these testing tools offer a common approach at some points, they offer different approaches at some other points.

When some of the existing tools are examined, common and different points emerged. First of all, if you need to address common points Free Usage (Open Source), Multi Language Support and Different Operating System Support. It has multiple special features as well as common points. Some of these features are unique, but some are rare features. Some of these rare features include Mobility, Test coverage report and Parallel and concurrent execution.

In the current available testing tool scenario there are multiple white box testing tools. These gadgets provide different structures and facilities. Within the framework of these structures and facilities, testing professional should decide on the tool to be used in the framework of the test approaches. The most important role in the decision phase is the requirements of the project and the means to be used. Nunit, SeleniumHQ, Junit, Cucumber and Protractor can be used if the project team does not want to pay for the tools to be used. In addition, there are multiple unique features. For example, if the project requirements and dynamics require mobility, the Veracode white box testing tool may be crucial because it is one of the rare test tools that can offer such a feature. Some projects may require multiple browser support, and project workers will need to use a tool with different browser support. Some of these tools include; Googletest, JSUnit, SeleniumHQ and Rational Functional Tester [RFT]. These tools are at the forefront with different browser support compared to others.

Unique features include parallel and concurrent execution and test coverage report. Junit is significant tool for the test coverage report feature because it ranks first among the rare tools that provide this feature. In addition, Nunit will be a wise solution if there is a request such as parallel and concurrent execution for the needs and requirements of the project.

There are multiple options available across the market, but many open source tools are being observed. If the project dynamics has reached the required case and the decision to use an open source tool as a result of this phenomenon in this stage tools such as Googletest, Junit, Jbehave and Cucumber can be preferred. Some projects have some tools approved by authorities such as MIT. In the event that such requirements are required in the project requirements, tools with an MIT license may be selected. These tools are, RCUNIT and Nunit.

Another very common feature is Multi Tool Objectives. In this regard, tool can provide multiple different objective. Considering the feature it provides in this regard, the tools that provide more than one test usage area can be an essential feature for the projects with mixed test facility. In this feature, tools like SeleniumHQ, TOSCA Testsuite, Rational Functional Tester [RFT], Cleanscape TestWise and Testuff may be options to select. The terms of use of these may vary from project to project.

5. CONCLUSION

As a summary there are number of white box testing tools available. When these testing tools are examined in detailed, it is not difficult to find a set of common features provided by many tools. However, among different and unique features are mobility, test coverage report and parallel and concurrent execution. When it comes to all features, the choice of the test tool is a challenging decision. In this decision phase, the dynamics within the scope of the project should be taken into consideration. Observations and examinations should be made within the framework of these dynamics and the most appropriate test tool should be selected. After this selection, a more efficient testing task can be done in the project. After a review is carried out in the light of this information and investigations, we can make some inferences and suggestions about White box test tools. First of all, if you want to use the code writing tool and record and replay, SeleniumHQ is a test tool. In addition to this, although many tools are used for mobile tests, SeleniumHQ is first mentioned with ease of use.

In terms of overall use, SeleniumHQ is a white box test tool that can be recommended due to its many features and browser support. In addition, integration with other test management tools will be an easier process for test management. The ease of processes and the fact that jobs can be planned in more than one part of the project are a building block that must be in the process of software testing.

The limitations of this study are that there is limited research in the literature about the white box testing tools. Therefore, the tools are identified through the websites as well as the literature. This study may be extended to include more white box testing tools as future research direction. Also, more attributes can be included for comparative view. As there is very limited research in the literature about the subject, this paper will contribute to the literature and provide insight to software testing professionals.

REFERENCES

[1] Structural Testing. U. Toronto, 2007/1, pp. 213–246.

[2] Khan, Mohd. Ehmer. Different Approaches To Black Box Testing Technique For Finding Errors. *International Journal of Software Engineering & Applications*, **4** (vol. 2), 2011, pp. 31–40, doi:10.5121/ijsea.2011.2404.

[3] Nidhra, Srinivas, and Jagruthi Dondeti. “Black Box and White Box Testing Techniques - A Literature Review. *International Journal of Embedded Systems and Applications*, **2** (vol. 2), 2012, pp. 29–50., doi:10.5121/ijesa.2012.2204.

[4] Westfall, Linda. *Testing – Systematic Code Coverage Techniques*, Westfall Team (www.westfallteam.com/sites/default/files/papers/Testing-Systematic Code Coverage Teschniques.pdf)

[5] *White Box Testing Tools*. CA Veracode, 1 Aug. 2017 (www.veracode.com/security/white-box-testing-tools)

[6] *Overview. EclEmma - Java Code Coverage for Eclipse*, 28 Mar. 2017 (www.eclemma.org/index.html)

[7] RCUNIT. (sourceforge.net/projects/rcunit/)

[8] Cfix: Unit Testing Framework for C/C, , (sourceforge.net/projects/cfix/)

[9] Erturk, Nezir, et al. *Yazılım Yapısal Kapsama Analizi*. (www.emo.org.tr/ekler/adf71fe88613d84_ek.pdf)

[10] *NUnit.org. (NUnit.org*, nunit.org/)

[11] CppUnit - C Port of Junit (sourceforge.net/projects/cppunit/)

[12] Junit (sourceforge.net/projects/junit/)

[13] *JSUnit.net Java Script Testing Tool*, Soeplus, Revolution Networks, (www.jsunit.net/)

[14] *RCUNIT* (www.qatestingtools.com/testing-tool/RCUNIT)

[15] *Cfix* (www.qatestingtools.com/testing-tool/cfix-testing)

[16] *EclEmma* (www.qatestingtools.com/testing-tool/eclemma)

[17] JUnit. *QATestingTools.com-Junit* (www.qatestingtools.com/testing-tool/junit)

[18] Nunit. *QATestingTools.com-Nunit* (www.qatestingtools.com/testing-tool/nunit)

[19] EMMA. *QATestingTools.com-EMMA* (www.qatestingtools.com/testing-tool/emma)

[20] *Googletest* (www.qatestingtools.com/testing-tool/googletest)

[21] White Box Testing Tools. *ProfessionalQA.com* (www.professionalqa.com/white-box-testing-tools)

[22] Khan, M.E. *Different approaches to white box testing technique for finding errors, International Journal of Software Engineering and its Applications*, **3** (vol. 5), 2011, pp. 1-14.

[23] Software Testing Tools List. *Software Testing Class*, 16 Dec. 2016 (www.softwaretestingclass.com/software-testing-tools-list/)

[24] What Is a White Box Testing? *Software Testing Class*, 1 June 2015 (www.softwaretestingclass.com/white-box-testing/)

[25] Control Flow Testing (http://www.computingstudents.com/notes/software_analysis/control_flow_testing.php)

[26] Barus, A. C., Hutasoit, D. I. P., Siringoringo, J. H. and Siahaan, Y. A. White box testing tool prototype development, *2015 International Conference on Electrical Engineering and Informatics (ICEEI)*, Denpasar, 2015, pp. 417-422.

[27] Foundation Level Syllabus. *ISTQB® International Software Testing Qualifications Board*, 27 Apr. 2018 (www.istqb.org/downloads/syllabi/foundation-level-syllabus.html)

[28] Gavaldo, Eric. "QA/Test Glossary." 12 July 2019, www.xqual.com/documentation/glossary.html.

[29] Collofello, J. S. dan Woodfield, S. N. Evaluating the effectiveness of reliability-assurance techniques. *Journal System and Software.* **3** (vol. 9), 1989, pp.191–195.

[30] Browser Automation. *Introduction - Selenium Documentation* (www.seleniumhq.org/docs/01_introducing_selenium.jsp)

[31] Test Management Software. *Test.com*, Zeta Producer (www.zeta-test.com/index.html)

[32] DotCover: A Code Coverage Tool for .NET by JetBrains. *JetBrains* (www.jetbrains.com/dotcover/)

[33] *Tricentis Tosca Testsuite*, Tricentis GmbH, (documentation.tricentis.com/en/1000/content/resources/overview.htm)

[34] End-to-End Testing for AngularJS. *Protractor* (www.protractortest.org/#/)

[35] CROUIGNEAU, Jean-Baptiste. Windmill.*Software Testing Tools Guide* (www.testingtoolsguide.net/tools/windmill/)

[36] Features & Benefits of the RapidRep Test Suite. *RapidRep*, FINARIS Financial Software Partner GmbH (www.rapidrep.com/en/features-benefits-test-suite)

[37] Main Features. *Testuff*, 19 May 2019 (www.testuff.com/product/main-features/)

[38] What Is JBehave? 28 May 2017 (jbehave.org/)

[39] Cucumber (cucumber.io/)

[40] *Visual Assert* (www.qatestingtools.com/testing-tool/visualassert)

[41] Rational Functional Tester. *IBM Developer: Download : Rational Functional Tester*, 9 Sept. 2014 (www.ibm.com/developerworks/downloads/r/rft/index.html)

[42] Support: NCover Collector. *Ncover,* Gnoso Inc. (www.ncover.com/support/docs/collector/index)

[43] Coding. *Concordion* (concordion.org/coding/java/markdown/)

[44] Welcome to Testwise. *Confluence,* GL Assessment (help.testingforschools.com/)

[45] Myers. G.J. “The Art of Software Testing”. *John Wiley and Sons, second edition, 2004.*

Information about the authors:

Dilara Ateşoğulları – Dilara Ateşoğulları is Senior Software Test Engineer. She has especially worked on test automation. She has been involved in multiple projects on Selenium.

Alok Mishra – Alok Mishra is Professor in Department of Software Engineering. His areas of research interest are Software Engineering, Information system, and Information Technology.

Manuscript received on 15 July 2019