3 СЕКЦІЯ «СУЧАСНІ ІНФОРМАЦІЙНІ СИСТЕМИ І ТЕХНОЛОГІЇ»

UDC 004.415.538

Dorenskyi O. P.[1]

THE METHODOLOGY OF EVALUATING THE TEST CASES QUALITY FOR SIMPLE IT MONOPROJECTS SOFTWARE TESTING

Information technologies (IT) are rapidly developing in Ukraine. Software tools that were developed during the process of IT projects realization are an integral part of information technologies. Analysis of the modern IT market showed that most of IT projects are complex [1]. However, simple IT monoprojects are implemented quite often. In particular, it happens during realization of simple software products, that were developed by one developer or freelance. Also this applies to the educational process that is aimed at training IT professionals, because in the early stages is not possible to implement multiprojects or complex IT projects.

During the realization of the IT project the main focus lands on documenting all processes of the software life cycle. One of them is a testing process [2], which is a mandatory process of any software realization.

Documentation of the testing process may be accomplished through the implementation of test kits. They are an array that consists of formally written set of test cases. Each test case is a set of input data, preconditions and conditions of the implementation, expected results and postconditions. Test cases are developed for a specific test scenarios in order to test performance for a specific requirement.

In [3–7] are presented some methods and recommendations for the structure, design and implementation of test cases, which consist of a set of attributes. Most commonly the following attributes are used: Test Case ID, Test Case Summary, Author, Preliminary Steps, Test Steps (Test Procedure), Expected Result, Test Result (Actual Result), Implementer, and PostConditions. For a simple IT monoproject test case is a set of the following attributes [3, 5]: ID, Action (Steps), Expected Result, and Test Result. However, when creating test cases some problems arise in the context of evaluating their quality. It is primarily constitutes in the synthesis of heuristic evaluation of Action (Test Steps) and Expected Result attributes, which are the essence of the test case. Thus, it is necessary to develop a methodology that will help to evaluate the test

[1] PhD, Lecturer, Department of Programming and Information Security, Kirovograd National Technical University, Kropyvnytsky, Ukraine

case quality, which involves assessing Action and Expected Result attributes. These attributes (fields) of a formally recorded test case describe the performer's (a person who performs testing) sequence of steps. Thus, they are algorithms that do not have a massiveness property. Therefore, test case assessment boils down to evaluating properties of the sequence of steps in the capacity of algorithms.

Test case steps describe the sequence of actions to be implemented in the course of test case realization. The expected result is a description of software tool reaction to every step of the test case. So, following properties should be evaluated: A test case should have such a property as discretisation. It means that its steps should be presented as a series of individual actions, they should represent a consistent process of realization of simple steps, and be performed in time discretely while accurately documenting completion of one activity's performance and beginning of another one. These steps and the expected result of the test case should be deterministic, i.e. have no action for which the essence is not clearly defined or can be perceived ambiguously. Each step must be clear and unambiguous, and should not admit the possibility for arbitrary or subjective interpretation. It is also necessary to assess finitude property. The execution of the test case should end after a finite number of steps, and test case steps should be clear to a performer. That is why it is necessary to evaluate the availability of their definition and recording. Accordingly, its effectiveness is determined by the sufficient simplicity that is needed for precise implementation of each test case step (action) over a finite period of time.

References

1. Nosdrina L. V. Project Management / Nosdrina L. V., Yashchuk V. I., Polotay O. I. – Kyiv: Center of Educational Literature, 2010. – 432 p.

2. Systems and Software Engineering – Software Life Cycle Processes: ISO/IEC 12207:2008 – [Second edition, 2008-02-01]. – ISO/IEC-IEEE, 2008. – 122 p. – ISBN 0-7381-5663-9. – (International Standard).

3. Test Case [Electronic resource]. – Access mode: http://www.tutorialspoint.com /software_testing_dictionary/test_case.htm.

4. Kulikov S. S. Testing Software. Basic Course: A Practical Guide / S. S. Kulikov. – Minsk: Chietyrie Chetverti, 2015. – 294 p.

5. Test Case [Electronic resource]. – Access mode: http://qalight.com.ua/uk/baza-znanij/test-case.html.

6. Williams L. Testing Overview and Black-Box Testing Techniques / Laurie Williams // A (Partial) Introduction to Software Engineering Practices and Methods. – 2011. – P. 34–59.

7. Kaner C. What is a good test case?: slides [Electronic resource] / Cem Kaner // Software Testing Analysis & Review Conference (STAR), East, Orlando, FL, May 12–16, 2003. – Orlando, 2003. – 16 p. – Access mode: http://www.kaner.com/pdfs/GoodTest.pdf.