ACTUAL PROBLEMS AND PROSPECTS OF TESTING SOFTWARE ODULES OF INFORMATION TECHNOLOGIES

Holovko D.
Central Ukrainian National Technical University, Kropyvnytskyi

Software module testing is an essential part of information technology software development. Each module need to pass a series of qualified tests before adding it to the main project. There are two approaches to combining modules during testing: incremental and monolithic. The research will analyze these two methods and consider the prospect of their use in the future.

In the IT field, a software module is understood to mean a part of compiled code that performs some functions and connects to the main code. It should also be noted that the module should only receive data, process it and return the result. In general, the process of testing the module is as follows [1, 5]: 1) a test set is developed, which is a set of control examples; 2) a test driver is implemented, which enables the module to be run with parameters from the test set and to further compare the result of the module with the expected one.

The step-by-step testing method is a method in which the software modules are not tested in isolation from one another but connected in turn to previously tested modules. The monolithic test method is to test each module individually, creating a test driver for each case. Monolithic testing is only profitable in the initial stages of development [2]. Comparing the two methods, it follows: monolithic testing requires more costs, because unlike step-by-step testing, where the previous modules are used, testing for each module requires writing a separate test driver; when step-by-step testing, errors when connecting multiple modules are detected much earlier, because the process of "forming" integral software begins earlier; in monolithic testing the results are limited only by the functionality of the current module, while step by step testing several modules pass at once, thus further testing the functionality of all the connected modules; in monolithic testing there is a possibility of "parallel" testing of several different modules, however the efficiency is shown only at the initial phase of testing.

In general, testing of application modules is one of the ways to save money on product creation. According to the paper [3], the main purpose of testing is far from confirmation of correctness, it is not to show the satisfactory performance of the program, but to clearly determine why the operation of the program is unsatisfactory. Testing costs a lot. For each test program, the more defects found per IT project, the higher the benefits of testing investment. Therefore, the purpose of testing is to detect as many defects as possible with a high level of importance [3].

The following results follow from the results obtained. Step-by-step testing is more sophisticated and convenient when considering the process of creating information technology software. With it, you can gradually "form" a ready-made software tool, which allows you to find conflicts of existing modules in the process of gradually adding them. However, monolithic testing cannot be ruled

out and only step-by-step everywhere. Sometimes, especially in the early stages of development, there is no need to integrate modules. It is at these times that monolithic testing will be a favorite, because, having developed several modules, they can be tested in parallel without any anchoring.

Therefore, both methods discussed in the paper will be relevant and will be difficult to find. As they begin to develop the software, everyone will use monolithic testing to quickly test the module and start developing the next one. After the creation and testing of each module, you can proceed to step-by-step testing, thus checking the compatibility of the modules and further testing them. It should be noted that, according to standard [6], the quality of the system is the degree of satisfaction of the system by the stated needs of different stakeholders, which allows us to evaluate the benefits. These stated and assumed needs are represented in the international standards of the SQuaRE series by means of quality models that represent product quality in the form of a breakdown into characteristic classes, which in some cases are subdivided into sub-characteristics. Therefore, testing will be relevant throughout as long as the software modules and software are generally developed. After all, developers and other contractors of the IT project make mistakes that are much easier and cheaper to detect at the testing stage than after the creation of a complete project.

REFERENCES

1. Slideshows. Modular Programming Course. [Oleksandr P. Dorenskyi] URL: http://moodle.kntu.kr.ua/course/view.php?id=524 (Last accesed: 03/29/2020). [in Ukrainian].
2. Module testing. URL: https://studfile.net/preview/ 273939 / Page: 2 /. (Last accesed: 03/29/2020). [in Ukrainian].
3. Modular testing. URL: https://project.dovidnyk.info/ index.php / home / tehnologiyarazrabotkiprogrammnogoobespecheniya / 26-modulenoetestrovanie. (Last accesed: 03/29/2020). [in Ukrainian].
4. Methods of testing the interaction of modules. URL: https://helpiks.org/7-46213. (Last accesed: 03/29/2020). [in Russian].
5. Dorenskyi O. P. The Methodology of Evaluating the Test Cases Quality for Simple IT Monoprojects Software Testing. Modern Problems and Achievements in the Field of Radio, Telecommunications and Information Technology : Proceedings of the 8th Internat. Scientific and Practical Conf. (September 21-23, 2016). Zaporizhzhia : ZNTU, 2016. Pp. 111–112.
6. DSTU ISO / IEC 25010: 2016 (ISO / IEC 25010: 2011, IDT). Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models [2018-01-01]. Kyiv, 2018. (National Standard of Ukraine).