International Science Group

ISG-KONF.COM

I

INTERNATIONAL SCIENTIFIC AND PRACTICAL CONFERENCE "PROBLEMS OF MODERN SCIENCE AND PRACTICE"

Boston, USA

September 21 – 24, 2021

ISBN 978-1-63972-061-3

DOI 10.46299/ISG.2021.II.I

PROBLEMS OF MODERN SCIENCE AND PRACTICE

Abstracts of I International Scientific and Practical Conference

Boston, USA
September 21 – 24, 2021

Library of Congress Cataloging-in-Publication Data

UDC 01.1
The I International Science Conference «Problems of modern science and practice», September 21 – 24, 2021, Boston, USA. 487 p.

ISBN - 978-1-63972-061-3
DOI - 10.46299/ISG.2021.II.I

EDITORIAL BOARD

TO THE QUESTION OF ANALYSIS OF EXISTING MECHANISMS OF WEB APPLICATION TESTING

Tvoroshenko Iryna,
Ph.D., Associate Professor
Kharkiv National University of Radio Electronics

Maksimenko Heorhii,
Master in informatics
Kharkiv National University of Radio Electronics

Creating any web pages subject area, even by experienced developers, is impossible without errors that can degrade the site as a whole. In terms of time and human resources, the most expensive are the stages of development [1-3], which are associated with finding errors in the finished web products. Although the effort required to make small changes is usually minimal, they may require considerable effort to verify the quality of the modified program. When a certain amount of work is done to test a web page in a short period of time, one of the best solutions is to automate the testing process [4, 5].

Testing as working process contains checking the quality of software in accordance with the documentation of this software. Testing is used in the process of creating not only web pages, but also web applications.

The testing process provides for a set of tasks and actions to check the absence of errors and identify defects in software systems, assess performance, control the availability and completeness of documentation, and assess the quality of design decisions. In addition, testing involves the fact of existence of a test object (web application) and a benchmark against which the object is compared.

A web application is a complex object whose properties change at different stages of development. It is important to agree on software requirements between the customer and the developer before design.

The main feature of web application requirements is the ability to verify it, as the ability to test compliance.

There are two types of requirements:

– Functional (which functions will be performed by the web application);

– Non-functional (for example, time constraints, data access speed, resource requirements, etc.).

Test suite requirements are grouped into test scenarios, the verification of which will give an unambiguous answer about the correctness of certain functions. If all the requirements are not specified in time, the customer may not get what he wanted, and the web application will not be able to properly test due to lack of a standard. The customer and the developer should be able to compare the current operation of the system with its reference, expected behavior.

Discussing the terms of reference, technical design, architecture of the system with the customer also helps to find errors and refine the standard. The main rule for the developer is to work closely with the customer to find common solutions in the process of creating a web application.

Periodic life-cycle testing of a web application will ensure that performance is consistently met. Because the object of testing is complex, it is necessary to apply a systematic approach to planning, organizing and conducting testing. Given the high cost of web applications, it is necessary to evaluate performance indicators at the design stage based on modeling methods and special tests.

The main task of the tester is to identify errors when testing software applications for smooth end-user operation. It is impossible to test the application without using it, but testers specifically create artificial situations that may arise in the future when working with the resource. The analysis of the behavior of the web application is carried out. When a specialist detects a system error, he passes his report to the project manager, who then distributes the work to eliminate errors among other project participants. After fixing the errors, the website is tested again. Testing will be conducted until the website becomes operational.

Website testing, like testing of any component as a whole, is a quality control mechanism that verifies the correspondence between actual and expected application behavior through a final set of tests which are selected in a certain way [6-8].

The testing process involves:

– Test Management;

– Test Design;

– Test Execution;

– Test Analysis.

Software quality is a set of properties of a product (service or program), characterized by the ability to meet the stated or anticipated needs of the customer. The concept of quality has different interpretations depending on the specific software system and its requirements. In addition, quality assessment models have different numbers of levels, completely or partially identical in terms of a set of quality characteristics.

For example, the RUP (Rational Unified Process) testing methodology focuses on quality assessment using the following methods:

– Search and documentation of quality defects;

– Identification of general recommendations for quality;

– Verification of compliance with basic expectations and requirements using specific examples;

– Verification that the product works as designed;

– Verifies that the requirements are met properly.

The International Standard for Software Test Documentation (IEEE Std 829-1983) defines testing as a process of software analysis aimed at identifying the differences between its properties that actually exist and those that are required of it (defect).

The ISO 9126:2011 standard regulates the external and internal quality characteristics that reflect the properties of the software itself, as well as the vision of the customer and the developer.

Quality is not absolute, it is a subjective concept. Therefore, testing, as a process of timely detection of errors and defects, cannot fully ensure the correct operation of the website. The requirements for the concept of quality are described in the standard ISO 9126. The composition and content of the documentation required for testing is given in the standard IEEE 829-2008.

Keep in mind that testers do not change the quality of the software; they do not change the base code. The quality of the software product can be changed only by further actions of the developers.

There are the following types of testing of the web page:

– By the object of testing:
    1) Functional testing;
    2) UI testing;
    3) Load/stress/performance testing;
    4) Security testing;
    5) Usability testing;
    6) Compatibility testing;
– By the degree of isolation of components:
    1) Component testing;
    2) Integration testing;
    3) System or end-to-end testing.

Automated software testing is part of the testing process at the quality control stage in the software development process. It uses software to test and validate the results, and this approach helps to reduce testing time.

There is such a gradation of types of software testing in the process of developing web software:

– Testing parts of software (modules, components);

– Functional testing of subsystems and software in general;

– Load testing to identify the characteristics of the operation of the software when the load changes (the intensity of access to it, filling the database, etc.).

The main objects of test automation are systems that implement the work of the client part. A key feature of testing client-server systems is the ability to verify the correct operation and the required performance of the system based on the work of the client part. Thus, by carefully and thoroughly testing these capabilities, you can get a guarantee of system performance.

Thus, automated testing of software and hardware applications and their compatibility reduces the cost of developing and updating the system through execution speed, reusability, coverage cases, and high accuracy without human error.

An important aspect of the organization of work is the preservation of the created tests. It is recommended to treat the tests in the same way as the source code, and you need to use the version to play the script if necessary (for example, GIT). This information will be necessary at the stage of software maintenance; it will allow you to reuse ready-made tests on several versions of the system [9].

As practice shows, testing of web applications reveals the following problems:

– Input data checks or only partial data checks;

– Incorrect input data processing;

– Buffer overflow;

– Careless operation of the program with files, in the case when the file name is passed to the program from the outside (GET or POST);

– Ignoring the peculiarities of GET and POST requests;

– Incorrect password handling (during data storage, transmission and processing);

– Incorrect access rights;

– Incorrect program rights on the server;

– Ignoring of the peculiarities of the programs for downloading files to the server;

– Incorrect logic of the web program, which with some valid input leads to unpredictable consequences;

– Output of service information in case of program errors or access to a database that is not intended for outsiders;

– Incorrect operation of databases (passwords, number of queries);

– Vulnerabilities of insufficient processing of input data when working with the database (SQL-injection);

– Unoptimized program code, which leads to significant loads on the web server;

– Vulnerability of web users and systems to DoS and DDoS attacks.

Currently, the most common methodologies for penetration testing are:

– The Open Source Security Testing Methodology Manual (OSSTMM);

– The National Institute of Standards and Technology (NIST) Special Publication 800-115;

– OWASP Testing Guide;

– Penetration Testing Execution Standard (PTES);

– Information Systems Security Assessment Framework (ISSAF);

– BSI – Study A Penetration Testing Model.

Let's analyze modern web testing mechanisms:

– Unit Testing. This type of testing performed by the developers before the application is passed to the team of testers for the official execution of test cases. The purpose of unit testing is to highlight each part of the program and show that the individual parts work correctly in terms of requirements and functionality. Thus, as part of this testing, an analysis of the modules integrated into the system is performed. However, unit testing cannot find all program errors. It is not possible to predict all execution alternatives in each software application. This applies to single testing. There is a limit to the number of scripts and test data that a developer can use to validate program code. After applying all possible options, the testing process of this module stops; the code segment is combined with others;

– Integration Testing. Testing the combined parts of a program to determine their balanced functioning is called integration testing. The interaction of the application with external systems is checked. There are two methods of integration testing: bottom-up testing and top-down integration testing. Bottom-up integration testing begins with testing individual modules, and then gradually testing combinations of higher-level hierarchy modules. In top-down integration testing, higher-level modules are tested first, and lower-level modules are tested last. When developing software, testing is usually performed from the bottom up and then from the top down;

– System Testing. After integrating all the components, the application is thoroughly tested for overall quality standards;

– Regression Testing. When some part of the software code is edited, it may affect the operation of other parts of the code in the program. Regression testing is performed to verify that the bug that was fixed did not result in a violation of other functionality or logic .The purpose of this testing method is to ensure that the changes made do not result in other application errors. In regression testing, top-down testing is performed on integrated modules;

– Acceptance Testing. The most important type of testing is performed by the quality assurance team, which checks whether the program meets the stated specifications and requirements of the client. The QA team has a set of pre-written scripts and test cases that will be used to test the program. Acceptance testing is designed not only to indicate simple spelling or cosmetic errors, inconsistencies in the interface, but also to identify errors in the application that can lead to system failures, significant errors in the program. By performing acceptance tests on the system, the test team will determine how the program will work during continuous use;

– Alpha testing. This test is the first stage of testing modules and their integration, the system as a whole; it is conducted by the developers and the quality assurance team. As a result, the application will check: spelling errors, broken links, testing will be performed on machines with the lowest specification to check loading time and possible delay problems;

– Beta testing. This test is performed after a successful alpha test. In beta testing, potential users are testing the application. Beta testing is performed before the release of the developed IT product. Beta tests of software sometimes offer a wide audience on the Internet to perform the test in real life. If the developed software is intended for internal use in a certain company, then for its testing the customer's employees are involved, who must perform the following actions:

1) Install and run the application;

2) Identify possible typographical errors, looping of the flow of applications and even failure of some modules;

3) Via feedback to send test results to the project team, which will solve the identified problems, while improving the quality of the developed application;

– Non-functional testing. This testing involves testing the software application based on requirements that are not inherently functional, but are extremely important, such as performance, security, user interface, etc.;

– Performance testing. This testing is mostly used to identify any bottlenecks or performance issues, rather than to find bugs in the software. There are various reasons that reduce the performance of software.

Thus, based on an analytical review of web application testing mechanisms, it was found that web application testing methods differ from each other depending on their scope [10-12].

For example, compatibility and interface testing is best done manually or using mechanisms to test the system load under different conditions. The following tools can be used to test a web application: Eclipse IDE, Java, Cucumber framework, Git, Selenium library, Junit, Jenkins, etc.

Thus, it is important to automate the testing process, effectively build test scripts, which will reduce time and financial resources [13].

Static testing methods implement the verification of all documentation: terms of reference, specifications, source code of the program in the programming language. All documentation is analyzed for compliance with programming standards. As a result of static verification, compliance with the specified criteria and customer requirements is established.

Dynamic methods are used in the process of direct program execution [14-16]. The correctness of the web application is checked on a set of tests or sets of prepared input data [17-19]. For the analysis of failure causes and their consequences, the most suitable are formal verification methods – event tree analysis, failure tree analysis and analysis of failure modes and consequences.

Thus, the testing process is a very complex and important stage of web application development, which requires detailed study and development.

References:

1. Lyashenko V., Mustafa S.K., Tvoroshenko I., and Ahmad M.A. (2020) Methods of Using Fuzzy Interval Logic During Processing of Space States of Complex Biophysical Objects, *International Journal of Emerging Trends in Engineering Research*, 8(2), pp. 372-377.

2. Asaad Ma. Babker, Abd Elgadir A. Altoum, Irina Tvoroshenko, and Vyacheslav Lyashenko (2019) Information Technologies of the Processing of the Spaces of the States of a Complex Biophysical Object in the Intellectual Medical System HEALTH, *International Journal of Advanced Trends in Computer Science and Engineering*, 8(6), pp. 3221-3227.

3. Gorokhovatskyi V.O., Tvoroshenko I.S., and Vlasenko N.V. (2020) Using fuzzy clustering in structural methods of image classification, *Telecommunications and Radio Engineering*, 79(9), pp. 781-791.

4. Tvoroshenko I.S., and Gorokhovatsky V.O. (2020) Effective tuning of membership function parameters in fuzzy systems based on multi-valued interval logic, *Telecommunications and Radio Engineering*, 79(2), pp. 149-163.

5. Kucherenko Ye.I., Filatov V.A., Tvoroshenko I.S., and Baidan R.N. (2005) Intellectual Technologies in Decision-Making Technological Complexes Based on Fuzzy Interval Logic, *East European Journal of Advanced Technologies*, 2, pp. 92-96.

6. Tvoroshenko I.S., and Gorokhovatsky V.O. (2019) Modification of the branch and bound method to determine the extremes of membership functions in fuzzy intelligent systems, *Telecommunications and Radio Engineering*, 78(20), pp. 1857-1868.

7. Yousef Ibrahim Daradkeh, and Iryna Tvoroshenko (2020) Application of an Improved Formal Model of the Hybrid Development of Ontologies in Complex Information Systems, *Applied Sciences*, 10(19), p. 6777.

8. Daradkeh Y.I., Tvoroshenko I., Gorokhovatskyi V., Latiff L.A., and Ahmad N. (2021) Development of Effective Methods for Structural Image Recognition Using the Principles of Data Granulation and Apparatus of Fuzzy Logic, *IEEE Access*, 9, pp. 13417-13428.

9. Matarneh Rami, Tvoroshenko Irina, and Lyashenko Vyacheslav (2019) Improving Fuzzy Network Models For the Analysis of Dynamic Interacting Processes in the State Space, *International Journal of Recent Technology and Engineering*, 8(4), pp. 1687-1693.

10. Tvoroshenko I.S., and Kramarenko O.O. (2019) Software determination of the optimal route by geoinformation technologies, *Radio Electronics Computer Science Control*, 3, pp. 131-142.

11. Кучеренко Є.І., Творошенко І.С. (2011) Оперативне оцінювання простору станів складних розподілених об'єктів з використанням нечіткої інтервальної логіки. *Штучний інтелект*, 3, С. 382-387.

12. Кучеренко Е.И., Корниловский А.В., Творошенко И.С. (2010) О методах настройки функций принадлежности в нечетких системах. *Системы управления, навигации и связи*, 1(13), С. 94-98.

13. Кучеренко Е.И., Творошенко И.С. (2003) Процессы принятия решений в сложных системах на основе нечетких интервальных представлений. *Вісник Національного технічного університету «ХПІ». Тематичний випуск: Системний аналіз, управління та інформаційні технології.* Х.: НТУ «ХПІ», 1(7), С. 79-86.

14. Gorokhovatskyi V., Rusakova N., and Tvoroshenko I. (2020) The application of image analysis methods and predicate logic in applied problems of magnetic monitoring, *Telecommunications and Radio Engineering*, 79(20), pp. 1801-1811.

15. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., Gadetska S., and Al-Dhaifallah M. (2021) Methods of Classification of Images on the Basis of the Values of Statistical Distributions for the Composition of Structural Description Components, *IEEE Access*, 9, pp. 92964-92973.

16. Гороховатський В.О., Творошенко І.С. (2021) Методи інтелектуального аналізу та оброблення даних: навч. посібник. Харків: ХНУРЕ, 92 с.

17. Творошенко І.С. (2021) Технології прийняття рішень в інформаційних системах: навч. посібник. Харків: ХНУРЕ, 120 с.

18. Flah P. (2015) *Machine learning. The science and art of building algorithms that extract knowledge from data*, Moscow, Russia: DMK Press, 400 p., (in Russian).

19. Nong Ye. (2013) *Data Mining: Theories, Algorithms, and Examples*, Florida, USA: CRC Press, 349 p.