Testing methods library for applications with web-based interfaces

T. Bragina, G. Tabunshchyk, D. Moroka

Abstract — The purpose of this work is to describe the practical implementation of testing methods library for applications with web-based interfaces, which are used for testing at each stage of the software project life cycle, beginning from the prototype, based on an evolutionary prototype creation.

Keywords — testing, web-based interface, metrics, application.

I. INTRODUCTION

The web-project quality is a complex integrated indicator. Quantitative indicators of quality are defined by the introduction of criteria relating to performance, reliability, operation and many other characteristics. However, the most important quality criteria are reliability and compliance with the functional specifications of the project, or, in other words, compliance with specified requirements of the system. One of the effective tools in this case, is the quality control testing, which should be applied at all stages of the project life cycle. There is the problem to make decisions about choosing the necessary tools and metrics for quality control, because of the large selection offered quality control methods, including testing. Therefore, the objectives were as follows:

- To explore the types of web-projects, and identify their characteristics and specific tools for their testing;
- To examine metrics and quality control methods;
- To implement a comprehensive method of testing a web-based interface.

Most attention should be paid to finding bugs early in the project since it is in the early stages the most important decisions are taken. Moreover, with the development of the project at each stage cost of correcting the defect grows. Nevertheless, the initial level of quality control is minimal and it increases with the promotion of the development process and complete testing starts only at later stages. The result is too late error detection, followed by the road processing system or its individual parts.

II. PROBLEM DEFINITION

Web-projects have its specific problems. First, data and requirements for web-projects often change, stored in various structured and unstructured formats in the system directory, creating a problem of tracking their relevance and accuracy. Second, tests for web-projects should be designed so that they are ready for possible changes in the technologies requirements and overall architecture. All these factors necessitate a well-organized quality control process, which laid the analysis stages and requirements specification for web-application and confirmed them at all stages of life cycle.

T. Bragina, Zaporizhzhya National Technical University, Ukraine, (e-mail: bragina.zntu@gmail.com).
G. Tabunshchyk, Zaporizhzhya National Technical University, Ukraine, (e-mail:galina.tabunshchik@gmail.com).
D. Moroka, Zaporizhzhya National Technical University, Ukraine, (e-mail: hotsauce9292@rambler.ru).

A set of test methods depends on the specific structure and purpose of web-projects, but mostly on the result of personal tester experiences. Therefore, the study of types of web-projects is important in order to determine their characteristics and specific tools for their testing [1].

Research scientists in the field of software quality verification touch various aspects of information systems testing and prove the need for a control strategy and implementation methods of testing in the early information systems stages. However, proposed methods do not take into account the internal structure of the web projects and especially the web interface, and there are no formal methods for assessing the functionality of web projects.

On the basis of this, it has been tasked to implement information technology for functional verification web-oriented systems.

III. INFORMATION TECHNOLOGY FOR FUNCTIONAL VERIFICATION WEB-ORIENTED SYSTEMS

Information technology for functional verification web-oriented systems is presented in Fig. 1 and consists of the following methods and models [2-6]:

- Web-oriented systems verification model;
- The regression testing method for web-oriented systems [1];
- Model assessment of the uncertainty level for the process of information system development [2, 3];
- Model and method of functionality assessing of web-oriented systems in conditions of uncertainty;
- Estimation time reserves method for the software web-oriented systems development based on uncertainty [1, 4];
- Evaluation model of software web-oriented systems functionality, taking into account losses [5].

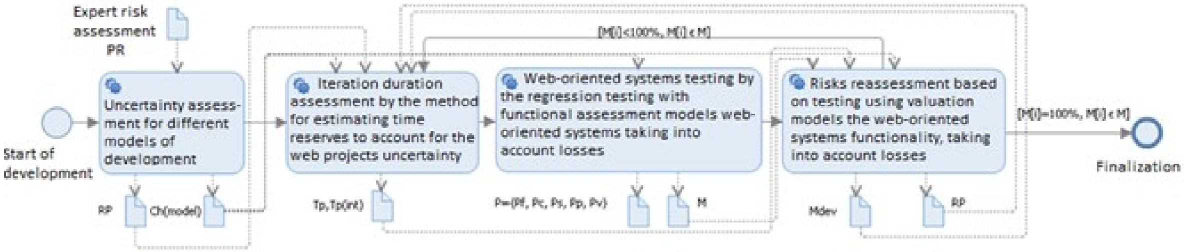

Expert risk assessment PR
Start of development
Uncertainty assess-ment for different models of development
RP
Ch(model)
Iteration duration assessment by the method for estimating time reserves to account for the web projects uncertainty
Tp,Tp(int)
[M[i]<100%, M[i] ∊ M]
Web-oriented systems testing by the regression testing with functional assessment models web-oriented systems taking into account losses
P={Pf, Pc, Ps, Pp, Pv}
M
Risks reassessment based on testing using valuation models the web-oriented systems functionality, taking into account losses
Mdev
RP
[M[i]=100%, M[i] ∊ M]
Finalization

Fig. 1 Information technology for functional verification web-oriented systems

Realized information technology [6] allows estimating the functional suitability of web-oriented systems with the uncertainty of scheduling, and includes the following steps:

Step 1. Uncertainty assessment for different models of development:

- Gathering expert estimates the probability of risks occurrence in the development of web-oriented systems;
- Evaluation of the index reduces the efficiency of information system development by the uncertainty level model assessment during the web-oriented systems development, using fuzzy inference;
- Drawing up the initial schedule using the method of time reserves evaluation during developing web-oriented systems for uncertainty.

Step 2. Iteration duration assessment by the method of estimating time reserves to account for the web-projects uncertainty.

Step 3. Web-oriented systems testing by the regression testing with functional assessment models for web-oriented systems taking into account losses - after the planning stage sprint web-oriented systems implement the tasks assigned to the development of the current sprint, followed by a test of the results. At this stage, automated re-evaluation of input parameters by information are obtained through testing developed in the current sprint artifacts.

Step 4. Risk reassessment based on testing, using evaluation models for the web-oriented systems functionality, taking into account losses.

Step 5. Completion of the development.

Information technology is implemented as a test methods library that consists of the following modules:

1. "Xenus Report Analizer" module (for automation of the regression testing method for web-oriented systems, verification model of web-oriented systems, model and method of functionality assessing of web-oriented systems in conditions of uncertainty);

2. "The risk assessment methods library" module (for the automation for the process uncertainty level assessment of information system development and estimation the time reserves method for the software web-oriented systems development based on uncertainty).

The heart of developed testing methods library for applications with web-based interfaces is Xenus Report Analyzer, because to control the quality of web-project manager it is required to process and monitor a large number of quality parameters of the project, including compliance with the developed web-project prototype approved by the customer. We need to use the necessary means testing at each stage of the project life cycle beginning from creating an evolutionary project prototype [1]. Given the large size and complexity of modern web-projects, there is a need for tools that automate and simplify the control of the developed product. On this basis, it was concluded that there is a necessity for implementation of a module for automation of the regression testing method for web-oriented systems.

IV. DECRIBING XENUS REPORT ANALYZER

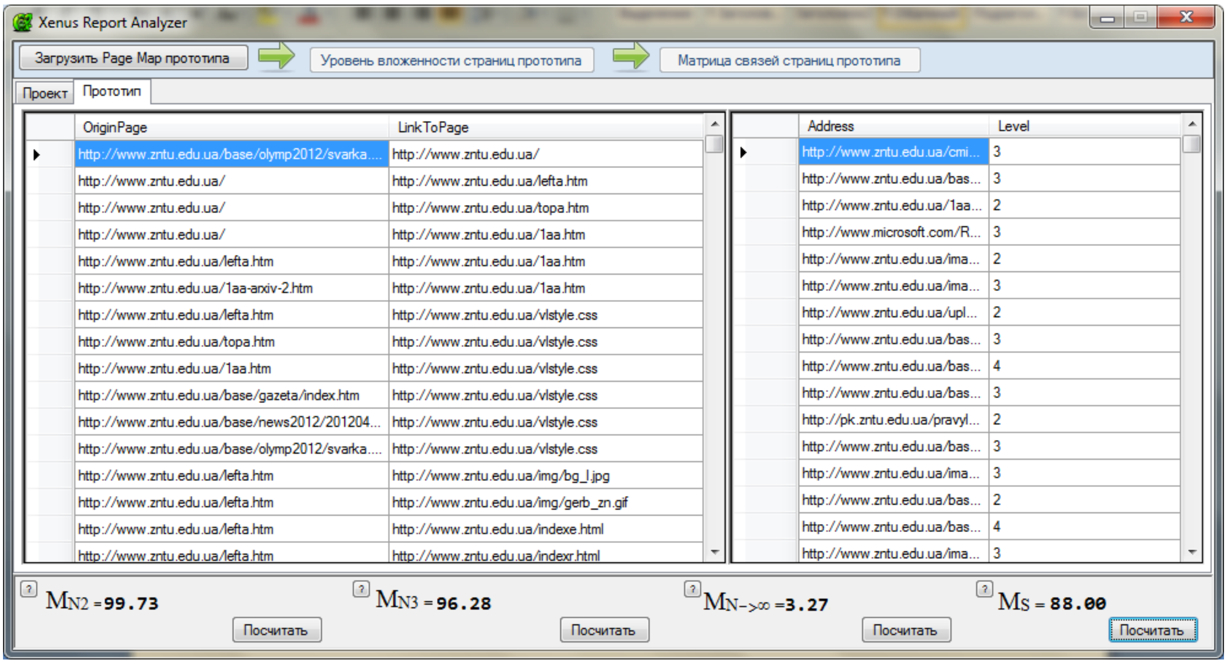
To assess the compliance of prototype, testing was performed by following metrics: compliance with the number of pages per second and third level of nesting prototype version of the project; number of pages being deeper in the third level of nesting; matching project to the structure prototype. With these metrics, the established test for compliance at every stage is artifact with customer.

Visual C# compiler was selected as the environment development method development environment for complex testing of web-projects, because it significantly reduces the amount of code through an ordered hierarchy of program structures and classes [7].

The Xenus Report Analyzer checks the development of the web-sites basing on the reports of the program Xenu, containing information on the list of accessible web-pages of the site and the links between them [1]. Class ExcelWorker was designed to work with these reports, containing methods for converting a file report by Xenu Microsoft Excel, obtaining the name of the report, downloading the report, a list of pages and their nesting level, a list of all the pages of a unique report.

Tested software showed that the greatest amount of time occupied the Page Map project downloading and determination for nesting pages level, because reports have a large number of rows. Matrix operations forming relationships took the least of all time.

The testing Xenus Report Analyzer is shown in Fig. 2. The left side of the interface matrix shows links between pages, and the right one - all pages of the project and their level of nesting. Here are the calculated metrics conformity assessment and call buttons prototype reference values and conversion metrics at the bottom.


Xenus Report Analyzer
Загрузить Page Map прототипа
Уровень вложенности страниц прототипа
Матрица связей страниц прототипа
Проект
Прототип
Посчитать
Посчитать
Посчитать
Посчитать

OriginPage	LinkToPage
http://www.zntu.edu.ua/base/olymp2012/svarka....	http://www.zntu.edu.ua/
http://www.zntu.edu.ua/	http://www.zntu.edu.ua/lefta.htm
http://www.zntu.edu.ua/	http://www.zntu.edu.ua/topa.htm
http://www.zntu.edu.ua/	http://www.zntu.edu.ua/1aa.htm
http://www.zntu.edu.ua/lefta.htm	http://www.zntu.edu.ua/1aa.htm
http://www.zntu.edu.ua/1aa-arxiv-2.htm	http://www.zntu.edu.ua/1aa.htm
http://www.zntu.edu.ua/lefta.htm	http://www.zntu.edu.ua/vlstyle.css
http://www.zntu.edu.ua/topa.htm	http://www.zntu.edu.ua/vlstyle.css
http://www.zntu.edu.ua/1aa.htm	http://www.zntu.edu.ua/vlstyle.css
http://www.zntu.edu.ua/base/gazeta/index.htm	http://www.zntu.edu.ua/vlstyle.css
http://www.zntu.edu.ua/base/news2012/201204...	http://www.zntu.edu.ua/vlstyle.css
http://www.zntu.edu.ua/base/olymp2012/svarka....	http://www.zntu.edu.ua/vlstyle.css
http://www.zntu.edu.ua/lefta.htm	http://www.zntu.edu.ua/img/bg_l.jpg
http://www.zntu.edu.ua/lefta.htm	http://www.zntu.edu.ua/img/gerb_zn.gif
http://www.zntu.edu.ua/lefta.htm	http://www.zntu.edu.ua/indexe.html
http://www.zntu.edu.ua/lefta.htm	http://www.zntu.edu.ua/indexr.html

Address	Level
http://www.zntu.edu.ua/cmi...	3
http://www.zntu.edu.ua/bas...	3
http://www.zntu.edu.ua/1aa...	2
http://www.microsoft.com/R...	3
http://www.zntu.edu.ua/ima...	2
http://www.zntu.edu.ua/ima...	3
http://www.zntu.edu.ua/upl...	2
http://www.zntu.edu.ua/bas...	3
http://www.zntu.edu.ua/bas...	4
http://www.zntu.edu.ua/bas...	3
http://pk.zntu.edu.ua/pravyl...	2
http://www.zntu.edu.ua/bas...	3
http://www.zntu.edu.ua/ima...	3
http://www.zntu.edu.ua/bas...	2
http://www.zntu.edu.ua/bas...	4
http://www.zntu.edu.ua/ima...	3

M_{N2} = 99.73

M_{N3} = 96.28

$M_{N->\infty}$ = 3.27

M_S = 88.00

Fig. 2 Testing Xenus Report Analyzer

Value metrics in Fig. 2 shows that the project is in the final stage of development and meets the requirements for the number of pages of different nesting levels.

V. TESTING XENUS REPORT ANALYZER

To test software four reports were taken. The first two reports contain general information and a map of the project site, and the other two - the prototype. The report contains general information on the project, having 9052 lines, and report on the site map, having 26,949 lines. The report keeps the overall site information prototype contains 429 lines, and report on the site map contains 459 lines.

The tests have shown that the greatest amount of time was occupied by project Page Map downloading, and the determination for the project nesting pages level, because reports have a large number of rows. Matrix operations forming relationships took the least of all time.

TABLE I. TESTING RESULTS

Operation	**Approximate time spent on the operation (s)**
Download Page Map Project	8.1
Determining the level of the project nesting pages	7.4
The formation of matrix connections between project pages	1.8
Downloading the Page Map prototype	1.7
Determining the prototype nesting pages level	2
Forming links between prototype pages matrix	0.5

CONCLUSIONS

The library for applications with web-based interface were developed, implementing the developed methods and algorithms of dynamical verification. This library allows to assess the compliance of web-site prototype at each stage of the project life cycle.

The heart of developed library is Xenus Report Analyzer, allowing to gather information and to increase the efficiency of testing by analyzing the structure of the prototype and developed web-applications.

REFERENCES

[1] T. Bragina, G. Tabunshchik, “The strategy of testing web - projects”, Proceedings of the Donetsk National Technical University. Series "Informatics, Cybernetics and Computer Science" (IKOT 2012). Issue 15 (203), pp. 118-124. (in Russian)

[2] T. Bragina, G. Tabunshchik, “Comparative analysis of iterative software development models”, *Radyoelektronika. Informatics. Management*, Vol. 2, pp. 130 – 139, 2010, (in Russian).

[3] T. Bragina, G. Tabunshchik, “Analysis of approaches to risk management in software projects with an iterative lifecycle”, Radyoelektronika. Informatics. Management, Vol. 2, pp. 120-124. 2011 (in Russian).

[4] T. Bragina, G. Tabunshchik, “Fuzzy analysis of project risk”, *Information Processing Systems*. Vol. 3 (93), pp. 15-21, 2011, (in Russian).

[5] T. Bragina, “Development integration tools”, *Information Processing Systems*, Vol. 8 (106), pp. 127-130, 2012, (in Ukrainian).

[6] T. Bragina, G. Tabunshchik, “Information technology risk-based evaluation of the functionality of web-oriented systems”, *Information Processing Systems*. Vol. 2 (118), pp. 245-252. 2014, (in Russian).

Co-funded by the Tempus Programme of the European Union

This publication is the result of the project implementation:
TEMPUS CERES: Centers of Excellence for young RESearchers.
Reg.no.544137-TEMPUS-1-2013-1-SK-TEMPUS-JPHES