ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2022. No. 1 (19)

UDC 004.415.53 DOI: https://doi.org/10.30837/ITSSI.2022.19.040

A. Popov, M. Momot, A. Yelizieva

CHOOSING THE TEST AUTOMATION SYSTEM ACCORDING TO CUSTOMER REQUIREMENTS

The **subject** of the research are methods and technologies for automating the process of software product testing. The aim of the work is to optimize the time and costs for performing automated testing of software products. The following **tasks** were solved: analysis of existing software testing automation systems; formation of system of selection criteria for testing automation systems; development of formalized model of selection process; development of automation system selection algorithm considering customer's requirements; development of UML diagrams for presentation of functional capabilities of developed application; development of application for informational support of selection process. To solve these **tasks**, we used methods of system analysis, theory of sets and technologies of cross-platform applications development. The following **results** were obtained: The most popular systems of test automation have been analyzed, their scope and main capabilities have been singled out. Selection criteria are singled out, divided into qualitative and quantitative. Formalized model for choosing test automation systems taking into account their characteristics and customer requirements is proposed. Developed UML diagram shows the functionality of the developed subsystem. The proposed algorithm for determining the re-recommended system of test automation allows us to take into account the vectors of criteria for testing systems. On the basis of the formalized model and algorithm we developed a subsystem that allows us to determine the optimal variant of test automation system on the basis of the introduced selection criteria. **Conclusions**: informational support for choosing a test automation system for software products based on the developed algorithm takes into account the customer's requirements and the characteristics of the existing systems, which allows us to select the most preferable option out of the possible systems. The main result of the developed subsystem is a recommendation for a user to use an automated testing system, taking into account customer requirements.

Keywords: testing of the software; automation testing systems; criteria of choice of the systems; the model of multicriteria choice; the algorithm of choice of the system; a diagram of precedents.

Introduction

When planning work on the development of a software system, much attention is paid to one of the stages of the life cycle - testing. Testing is a very important stage because it is at this stage the developed system takes the form of a full-fledged product. At present, systematized testing is actively used in the industrial development of software products. Automation of this process becomes more and more necessary with increasing scale of projects [1]. Automatization systems significantly simplify the process and save both time and money [2, 3].

At present, there is a large number of test automation systems on the market, both traditionally popular and newly emerging. A software company faces a nontrivial task of selecting test automation systems, taking into account existing restrictions/requirements, such as supported testing processes, technologies, standards, integration with development systems, application type, platform type, test execution time, budgets

When implementing the testing phase, it is necessary to meet the resource conditions of the software product development project [4, 5]. It is necessary to take into account the time and cost constraints, which may be the requirements of the customer.

Thus the purpose of this article is to optimize the time and cost of performing automated testing. The article presents analysis and comparison of existing testing automation systems, development of a mathematical model for choosing the automatic testing subsystem and an algorithm of the subsystem performance for choosing an appropriate system for automating the testing process.

Analysis of recent research and publications

The process of automatic testing involves checking the software product for various inaccuracies and errors that were made at the development stage [6]. Analysis of publications on the subject showed that the main attention is paid to testing methods (functional, system, modulus, performance testing, etc.) and capabilities of software tools [6, 7]. Nowadays there are many different automation systems, which are used at testing stage of software products [8, 9]. Table 1 presents a comparative analysis of existing software testing automation systems, reveals their main features and application features [10-17].

Table1. *Analysis of existing test automation systems*

Name	Field of application	Features
1	2	3
Selenium	It is used to automate the testing of web applications, as well as any routine actions performed through the browser. Selenium supports desktop and mobile browsers.	- allows you to develop automation scripts in almost any programming language; - the ability to organize distributed stands that consist of hundreds of machines with different operating systems and browsers; - the ability to run scenarios in the clouds.

The end **Table1.**

1	2	3
TestComplete	Allows you to create tests for Windows applications, web servers and web pages. Used to automate various types of software testing for .NET, Java, Visual C++, Visual Basic, Delphi, C++Builder, web pages and other applications in order to reduce the time required for manual testing.	- the presence of a special mechanism that facilitates the creation of scripts for testing applications; - the necessary actions of the tester are automatically recorded in a file that is available for viewing and editing and repeats all previously performed operations when launched; - recorded tests can be modified later.
Quick Test Professional (QTP)	Faster than its counterparts recognizes graphic controls ("controls")	- offers a wide range of functionality for checking and interacting with the application under test; - uses the VBScript programming language to automate tests; - offers several types of checkpoints that allow users to control different aspects of application testing.
MS Test	Requires installation of Visual Studio, as it is part of the Team Foundation Server product. Plans and test results are saved on the Team Foundation Server. Test scenarios are managed by Microsoft Test Manager (MTM). MTM includes a test plan, test case, and configurations. MTM works only at the application level, so it needs to be installed on the server (if the server is remote, the work is done via VPN). Provides the ability to combine tasks that are assigned to the performer, with the reports (defects) and reports on the time spent on the work.	- research testing; - planning and execution of manual tests; - cross-platform test configurations (different versions of the same test for different platforms/releases); - test passing diagnostics (logs, videos, etc.); - import-export of tests; - cross-project import/export of tests; - recording and playback of manual tests (recorder); - test automation.
Telerik WebUI Test Studio	Used to automate functional testing of web applications (ASP.NET, Silverlight). It is an extension to Microsoft Visual Studio	- Silverlight support. In the initial versions of the program there were enough bugs associated with the support of silverlight elements, now the problem has been eliminated; - excellent handling of AJAX requests of any complexity; - no problems when writing tests with RadControls components (page controls); - has a handy test editor; - integration with other testing tools. All features of Visual Studio are available; - basic functionality of any sensible modern automation tool (Test Recorder, Elements Explorer, DOM Explorer, code generation, support for multiple browsers, etc.) is implemented at a good level.

The effectiveness of the testing phase depends on the correct choice of an automation tool. The choice of an application test automation system depends on the following factors:

- the type of application to be tested (platform orientation);
- the testing objectives (compliance with standards, applicability of the system);
- the requirements of the client (deadline and cost).

Based on this, there is a need to develop a formalized model for selecting a means of performing automated testing.

Materials and methods

To formalize the process of selecting a system for automating the testing process, the following model is proposed. Let's represent in the form of a vector of sets of criteria affecting the choice of a test automation system:

$$K = \{k_i\},\ i = \overline{1,6}\,. \qquad (1)$$

Two types of criteria are used: quantitative and qualitative [18]. Quantitative criteria include:

- K_1 – timing (necessary time to carry out the work);
- K_2 – cost of the work (the cost of their implementation);
- K_3 – number of employed specialists.

Qualitative criteria include:

- K_4 – degree of applicability of the system;
- K_5 – degree of compliance with standards;
- K_6 - platform orientation.

The degree of applicability of systems allows you to determine how effectively a test automation system can be adapted to the real work on a project when going through the testing phase, and which of its benefits can be most useful in doing so [19, 20]. The degree of compliance with standards shows to what extent the various characteristics of activities in the testing phase comply with international standards in this area:

- ISO 90003:2014 (Guidelines for the application of ISO 9001:2008 to computer software);
- ISO 25051:2014 (Requirements for quality of Ready to Use Software Product (RUSP) and instructions for testing);
- IEEE 829 (Software test documentation);
- IEEE 1008 (Standard for Software Unit Testing);
- ISO 8402 (Quality management and quality assurance);
- ISO 9126 (Software engineering – Product quality).

Each of the standards has its own specific specification and reflects a particular aspect of the testing phase activity. The degree of conformance to the standards shows how well the testing phase has been prepared, depending on each specific standard. For example, IEEE Standard 829 describes the procedure for compiling a test plan that will be used for the work, IEEE Standard 1008 describes the procedure for organizing unit testing, etc.

It should be noted that the choice of test automation system is carried out on at least two criteria Ki, which are most important to the customer. Let's represent the existing test automation systems as a set *M* [21, 22]:

$$M = \{ m_j \}, \; j = \overline{1, N} \, . \tag{2}$$

In this case, the values of the vector of criteria for each test automation system are known:

$$\forall m_j \in M : K = < k_i(m_j) > . \tag{3}$$

It should be noted that these criteria are heterogeneous. To bring them into a dimensionless value, it is necessary to normalize them by the formula:

$$k_i^{\text{н}}(m_j) = \frac{k_i(m_j) - k_i^{\text{нх}}(m_j)}{k_i^{\text{нл}}(m_j) - k_i^{\text{нх}}(m_j)}, \tag{4}$$

where $k_i^{\text{нл}}(m_j)$ – the best value of the criterion;

$k_i^{\text{нх}}(m_j)$ – the worst value of the criterion;

$k_i(m_j)$ – the current criterion values.

The value of quantitative criteria is represented by some non-negative number. Qualitative criteria have point estimates, i.e. they are also represented by non-negative numbers. These evaluations can be obtained by expert way. Experts also determine the degree of importance of criteria, i.e. a tuple of dimensionless weight coefficients can be determined:

$$A = < a_i > , \; i = \overline{1, n} \, .$$

The condition must be met that:

$$0 \le a_i \le 1 \, , \; \forall i = \overline{1, n} \, , \; \sum_{i=1}^{n} a_i \, . \tag{5}$$

Thus, we can introduce a utility function:

$$P(x) = F[A, k_i^{H}(m_j)] \, , \tag{6}$$

where $k_i^{H}(m_j)$ – normalized criteria in the interval [0, 1] (4);

A – tuple of dimensionless weight coefficients.

Customer requirements for selecting a test automation system are described by a vector of desired criteria values:

$$R = \{ r_i \}, \; i = \overline{1, 6} \, . \tag{7}$$

In this case, if the customer is not interested in some of the criteria, the corresponding criteria in the vector can take the value -1 (as a negative signal value).

It is also necessary to fulfill the condition of resource feasibility of the testing stage:

$$res_k \le Res_l \, , \tag{8}$$

where Res_l – *l*-th resource (time, financial, etc.) of the software project;

res_k – *k*-th type of resource at the testing stage.

The problem statement is formulated as follows: it is necessary to form the resulting set $M_{rec} \subset M$, which consists of alternatives, corresponding in their values of criteria (3) to the requirements of the customer:

$$M_{rec} = \{ (k_i(m_j) \mid (k_i(m_j) \le r_i) \cup (k_i(m_j) > -1) \} \, . \tag{9}$$

Let's represent the algorithm for choosing the automation system as follows:

Step 1. Initialization: resulting set **M$_{\mathbf{rec}}$** = ∅.

Step 2. For all $m_i \in$**M**, i=1, 2, … $N_{M.}$

2.1. flag = true.

2.2. For all r_j, j = 1, 2, … N.

2.2.1. If r_j == -1, then switch to 2.2.

2.2.2. If the matching condition is not met $r_j \div k(m^i_j)$, then flag = false

2.2.3. If (j==N) and (flag==true), then **M$_{\mathbf{rec}}$** = **M$_{\mathbf{rec}}$** U $m_{i.}$

The designation means the operation of checking the correspondence ($r_j \div z^i_j$) of the value of the *i*-th criterion of the *j*-th test automation system to the value of the same i-th criterion determined by the customer. The final choice of one option of test automation system from the **M$_{\mathbf{rec}}$** set (if it contains more than one element) is recommended to be made based on the customer's preferences or using the criteria convolution according to the formula (6).

Study results

Based on a formal representation of the system selection process for test automation and a generalized selection algorithm, a prototype subsystem was developed to recommend a test automation system to the customer based on various types of actors. Fig. 1 shows the USE CASE diagram, which shows the relationships between the actors and precedents of the subsystem under development.

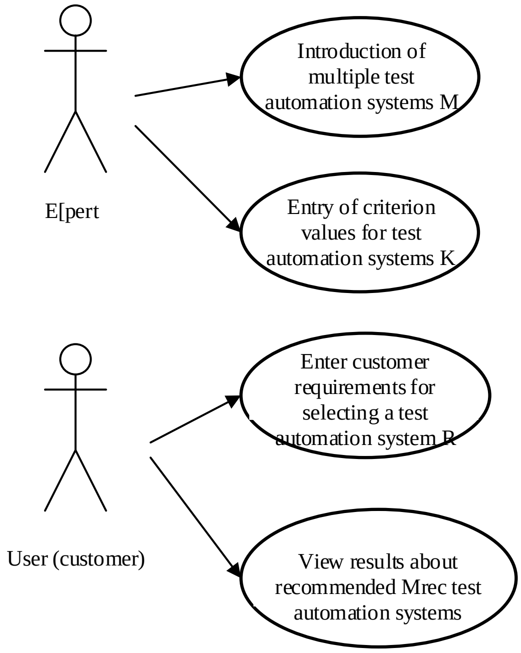
E[pert
Introduction of multiple test automation systems M
Entry of criterion values for test automation systems K
User (customer)
Enter customer requirements for selecting a test automation system R
View results about recommended Mrec test automation systems

Fig. 1. The USE CASE diagram of the subsystem under development

The input data for the subsystem are the following parameters:

- project name;
- version;
- project type;
- platform used;
- start date of the stage;
- end date;
- estimated duration (days);
- anticipated cost;
- number of specialists employed;
- assessment of compliance with standards (1 to 10), such as: ISO 90003:2014, ISO 90003:2014, ISO 25051:2014, IEEE 829, IEEE 1008, ISO 8402, ISO 9126.

A score of 1 to the standard defines the worst value, and 10 defines the best value, i.e. full compliance of the testing system with the specified software standards. It is assumed that the subsystem selects the most preferable system for test automation among the systems presented in table 1.

Table 2 shows data on the correspondence of the set of criterion vector values for test automation systems. A combination of parameter values determines the possibility of recommending a particular test automation system.

Based on the values of the previously mentioned criteria k_i, as well as the set M of the considered test automation systems, an algorithm for determining the recommended test automation system was obtained (fig. 2). The incoming arrow index in the system selection block in the algorithm in fig. 2 was used to refine the set of criterion vector values for the systems - written in brackets in the cells of table 2.

Table 2. *Values of the criteria vectors for automation systems testing*

Values	**Selenium**			**TestComplete**				**QTP**	**MS Test**		**Telerik Test Studio**
Application type											
Windows-Application	-1	-1	-1	-1	-1	1(10)	1(11)	-1	-1	1(7)	-1
Windows, .Net	-1	-1	-1	-1	-1	1(10)	-1	-1	-1	1(7)	-1
Windows, Java	-1	-1	-1	-1	-1	-1	1(11)	-1	-1	-1	-1
Web application	1(3)	1(4)	1(5)	1(8)	1(9)	-1	-1	1(2)	1(6)	-1	1(1)
Web, .Net	1(3)	-1	-1	-1	1(9)	-1	-1	-1	1(6)	-1	1(1)
Web, Java	-1	1(4)	-1	1(8)	-1	-1	-1	-1	-1	-1	-1
Web, other	-1	-1	1(5)	-1	-1	-1	-1	1(2)	-1	-1	-1
Budget											
<=10.000	1(3)	-1	-1	-1	-1	1(10)	-1	-1	-1	-1	-1
>10.000	-1	-1	-1	-1	-1	-1	-1	-1	-1	1(7)	-1
<=20.000	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1(1)
>20.000	-1	-1	-1	-1	1(9)	-1	-1	-1	1(6)	-1	-1
Terms											
> 14 days	-1	-1	-1	1(8)	1(9)	-1	-1	1(2)	1(6)	-1	1(1)
<= 14 days	1(3)	1(4)	1(5)	-1	-1	-1	-1	-1	-1	-1	-1

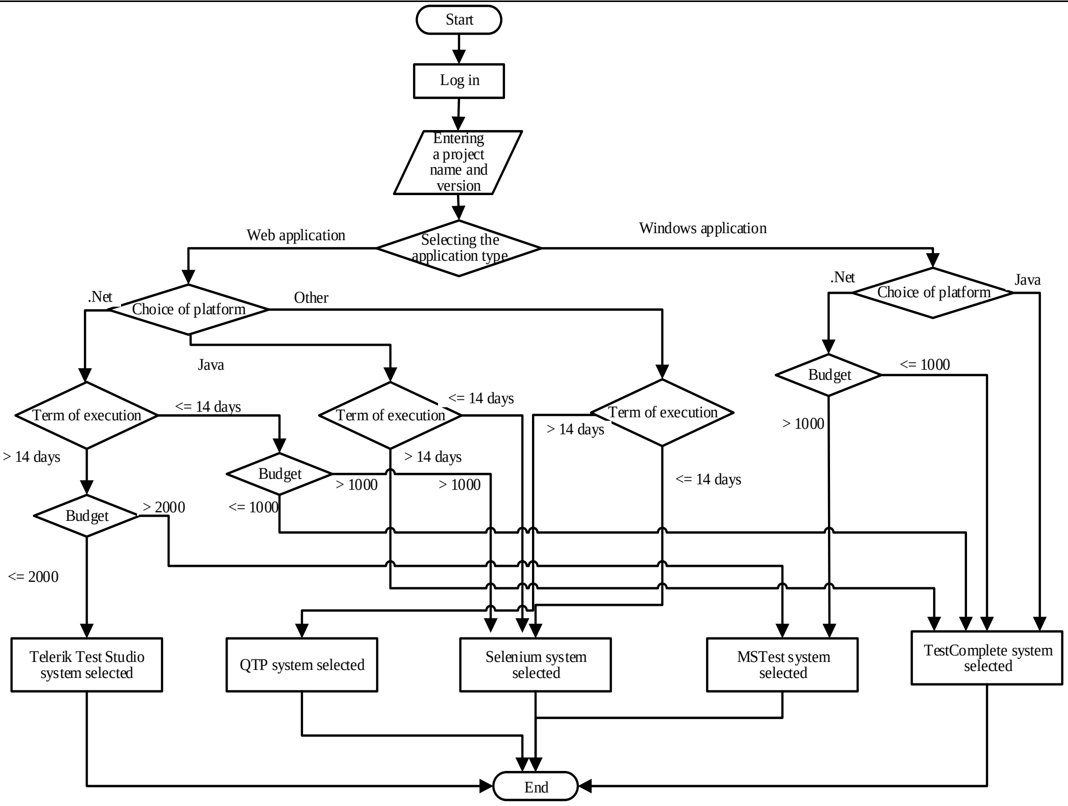

Fig. 2. Algorithm of determination of the recommended test automation system

In the presented scheme, when choosing a web application, the timing and budget are taken into account, because the available systems allow more flexibility in organizing the process of test automation in web development. When choosing a Windows application, the timing and budget (except for the .Net platform) are not taken into account, because the systems supporting these platforms do not allow such flexibility in organizing the automation process.

Generally speaking, the algorithm can be represented by the following steps:

- logging into the system;
- entering the name and version of the project;
- selection of the application type (web application selected);
- entering the start and end date of the stage;
- input of the estimated duration;
- entry of estimated costs.

Consider an example of calculating the choice of the system according to the specified initial data:

Project name: Randomator.

Version: 2.1.2.

Type of the project: Web application.

Platform used: .NET.

Start date of the stage: 25.08.2019.

End date: 12.10.2019.

Estimated duration: 48 days.

Estimated cost: 24000.

Conclusions

A model of multi-criteria choice of test automation system on the basis of the vector of quantitative and qualitative criteria is proposed, which allows taking into account the requirements of the customer and the resources of the project to create a software product with the help of the utility function. The paper presents an algorithm for determining the recommended system of test automation, which is implemented in a prototype software subsystem in which the user receives a specific recommendation for the use of the most effective system of test automation based on the entered data.

Scientific novelty of the article consists in the formalized representation of the process of choosing the system for automating the testing of software products, which allows scientifically justified choice of the system for evaluating the compliance of the finished program with the specified requirements.

Practical significance consists in the possibility of using the developed subsystem for information support of testers of software products. This allows you to choose the best option for automating the testing process, taking into account the requirements of the customer.

References

1. Dustin, E., Garrett, T., Gauf, B. (2009), *Implementing automated software testing: how to save time and lower costs while raising quality*, Addison-Wesle, Boston, 368 p.
2. Berkun, S. (2014), *Art of IT-project management* [*Iskusstvo upravlenija IT-proektami*], SPB, St. Petersburg, 700 p.

3. Novichkov, A., Pankratov, V. (2005), "Automated testing: evaluation of investments return and associated risks" ["Avtomatizirovannoe testirovanie: ocenka vozvrata investicij i soputstvujushhie riski"], *ComputerPress*, No. 11. P. 47 – 49.
4. Golyan, V. V., Kravchenko, O. K. (2019), "Comparison of software life cycle models in order to identify the most effective " ["Porivnyannya modelej zhyttyevyx cykliv programnogo zabezpechennya z metoyu vyyavlennya najefektyvnishogo"], *Infocommunication systems*, No. 2 (157), P. 63 – 70. DOI: http://doi.org/10.30748/soi.2019.157.08
5. Kolesnikova, K. V., Gogunskii, V. D., Negri, A. A., Olekh, H. S. (2016), "Conceptual model of project management" ["Konceptualna model upravlinnya proektamy"], *Electro technical and computers systems*, No. 23 (99), P. 175 – 179. DOI: https://doi.org/10.15276/eltecs.23.99.2016.27
6. Yegorova, O., Bychok, V. (2019), "Software testing tools" ["Programni zasoby dlya testuvannya programnogo zabezpechennya"], *Young scientist*, No. 11 (75), P. 680 – 684. DOI: https://doi.org/10.32839/2304-5809/2019-11-75-144
7. Ivan Romero-Pena, Gerardo Padilla-Zarate, Karen Cortes-Verdin (2021), "Identification of Test Cases Duplication: Systematic Literature Review. Software Engineering Research and Innovation", *9th International Conference*, P. 104 – 111.
8. Troyan, A. M., Modenov, Yu. B. (2017), "The advisability of automated testing to ensure the quality of the software products" ["Docilnist avtomatyzovanogo testuvannya dlya zabezpechennya yakosti programnyx produktiv"], *Problems of automatization and control*, Vol. 1, No. 57 – 58, P. 86 – 89. DOI: https://doi.org/10.18372/2073-4751.1.12798
9. "Top 10 of software test automation tools", available at: https://habr.com/ru/post/481294/ (last accessed: 09.10.2021).
10. "Top 10 tools of test automation 2018", available at: https://habr.com/ru/post/342234/ (last accessed: 09.10.2021).
11. Slinkin, A., Demiyanov, Ye. (2010), *Selenium. Inroduction to the system* [*Selenium. Vvedenie v sistemu*], Simvol-Pljus, Moscow, 656 p.
12. Gundecha, U. (2012), *Selenium Testing Tools Cookbook*, Packt Publishing, Birmingham, 326 p.
13. Alpaev, G. (2013), *TestComplete Cookbook*, Packt Publishing, Birmingham, 282 p.
14. Alpaev, G., *TestComplete Book*, available at: https://alpaev.com/testcomplete/ (last accessed: 27.10.2021).
15. Sankar, V. R. (2012), *QuickTest Professional*, Tata McGraw-Hill, New Delhi, 600 p.
16. "Using MSTest platform in module tests" ["Ispol'zovanie platformy MSTest v modul'nyh testah"], available at: https://docs.microsoft.com/ru-ru/visualstudio/test/using-microsoft-visualstudio-testtools-unittesting-members-in-unit-tests (last accessed: 27.10.2021).
17. "Telerik Test Studio", available at: https://docs.telerik.com/teststudio (last accessed: 29.10.2021).
18. Mukhamediev, R. I., Mustakayev, R., Yakunin, K., Kiseleva, S, Gopejenko, V., "Multi-Criteria Spatial Decision Making Supportsystem for Renewable Energy Development in Kazakhstan," available at: https https://s3-us-west-2.amazonaws.com/ieeeshutpages/xplore/xplore-ie-notice.html (last accessed: 27.12.2021).
19. Hochbaum, D. (1995), "Approximating Covering and Packing Problems", *Journal of the Association for Computing Machinery*, Vol. 32, No. 1, P. 94 – 143.
20. Mukhamediev, R. I., Mustakayev, R., Yakunin, K. O., Kuchin, Y. I., Kiseleva, S. V., Gopejenko, V.I. (2020), "Decision Support System for Optimization of Res Generators Placement Based on Geospatial Data", *News of The National Academy of Sciences of The Republic of Kazakhstan Series of Geology and Technical Sciences*, Vol. 1, No. 439, P. 81 – 89. DOI:https://doi.org/10.32014/2020.2518-170X.10
21. Beskorovainyi, V. (2020), "Combined method of ranking options in project decision support systems", *Innovative Technologies and Scientific Solutions for Industries*, No. 4 (14), P. 13 – 20. DOI: https://doi.org/10.30837/ITSSI.2020.14.013
22. Malyeyeva, O. V., Yelizieva A., V., Kosenko, N. V., Nevlyudova, V. V. (2018), "Information technology of decision making support on production enterprise purchasing management" ["Informacijna texnologiya pidtrymky pryjnyattya rishen z upravlinnya zakupivlyamy vyrobnychogo pidpryyemstva"], *Innovative Technologies and Scientific Solutions for Industries*, No. 3 (5), P. 57 – 66. DOI: https://doi.org/10.30837/2522-9818.2018.5.057

Received 04.02.2022

Відомості про авторів / Сведения об авторах / About the Authors

Попов Андрій Вячеславович – кандидат технічних наук, доцент, Національний аерокосмічний університет ім. М.Є. Жуковського "ХАІ", доцент кафедри комп’ютерних наук та інформаційних технологій, Харків, Україна; e-mail: apv@xai.edu.ua; ORCID: https://orcid.org/0000-0001-8984-731X.

Попов Андрей Вячеславович – кандидат технических наук, доцент, Национальный аэрокосмический университет им. Н.Е. Жуковского "ХАИ", доцент кафедры компьютерных наук и информационных технологий, Харьков, Украина.

Popov Andrei – PhD (Technical Sciences), Docent, National Aerospace University "Kharkiv Aviation Institute"; Docent of the Department of Computer Sciences and Information Technologies, Kharkiv, Ukraine.

Момот Мирослав Олександрович – кандидат технічних наук, доцент, Національний аерокосмічний університет ім. М.Є. Жуковського "ХАІ", доцент кафедри комп’ютерних наук та інформаційних технологій, Харків, Україна; e-mail: m.momot@khai.edu; ORCID: https:// orcid.org/0000-0003-2580-5908.

Момот Мирослав Александрович – кандидат технических наук, доцент, Национальный аэрокосмический университет им. Н.Е. Жуковского "ХАИ", доцент кафедры компьютерных наук и информационных технологий, Харьков, Украина.

Momot Myroslav – PhD (Technical Sciences), Docent, National Aerospace University "Kharkiv Aviation Institute"; Docent of the Department of Computer Sciences and Information Technologies, Kharkiv, Ukraine.

Єлізєва Аліна Володимирівна – кандидат технічних наук, Національний аерокосмічний університет ім. М.Є. Жуковського "ХАІ", доцент кафедри комп’ютерних наук та інформаційних технологій, Харків, Україна; e-mail: a.elizeva@khai.edu; ORCID: https://orcid.org/0000-0002-8228-9383.

Елизева Алина Владимировна – кандидат технических наук, Национальный аэрокосмический университет им. Н.Е. Жуковского "ХАИ", доцент кафедры компьютерных наук и информационных технологий, Харьков, Украина.

Yelizieva Alina – PhD (Technical Sciences), National Aerospace University "Kharkiv Aviation Institute"; Docent of the Department of Computer Sciences and Information Technologies, Kharkiv, Ukraine.

ВЫБОР СИСТЕМЫ АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ С УЧЕТОМ ТРЕБОВАНИЙ ЗАКАЗЧИКА

Предметом исследования являются методы и технологии автоматизации процесса тестирования программных продуктов. **Целью** работы является оптимизация времени и затрат на выполнение автоматического тестирования программного продукта. В работе решены следующие **задачи**: проведение анализа существующих систем автоматизации тестирования программных продуктов; формирование системы критериев выбора систем автоматизации тестирования; разработка формализованной модели процесса выбора; разработка алгоритма выбора системы автоматизации с учетом требований заказчика; разработка UML диаграмм для представления функциональных возможностей разработанного приложения; разработка приложения для информационной поддержки процесса выбора. Для решения указанных задач были использованы **методы** системного анализа, теории множеств, технологии разработки кросс-платформенных приложений. Получены следующие **результаты**. Проанализированы наиболее популярные системы автоматизации тестирования, выделены их область применения и основные возможности. Выделены критерии выбора, которые разделяются на количественные и качественные. Предложена формализованная модель выбора систем автоматизации тестирования с учетом их характеристик и требований заказчика. Разработанная UML диаграмма отражает функциональные возможности разрабатываемой подсистемы. Предложенный алгоритм определения рекомендуемой системы автоматизации тестирования позволяет учитывать векторы критериев для систем тестирования. На основе формализованной модели и алгоритма была разработана подсистема, которая на основе введенных критериев выбора позволяет определить оптимальный вариант системы автоматизации тестирования. **Выводы**: информационная поддержка выбора системы автоматизации тестирования программных продуктов на основе разработанного алгоритма учитывает требования заказчика и характеристики существующих систем, что позволяет выбрать наиболее предпочтительный вариант из возможных систем. Основным результатом работы разработанной подсистемы является получение пользователем рекомендации по использованию системы автоматического тестирования с учетом требований заказчика.

Ключевые слова: тестирование программ; системы автоматизации тестирования; критерии выбора системы; модель многокритериального выбора; алгоритм выбора системы; диаграмма прецедентов.

ВИБІР СИСТЕМИ АВТОМАТИЗАЦІЇ ТЕСТУВАННЯ З УРАХУВАННЯМ ВИМОГ ЗАМОВНИКА

Предметом дослідження є методи та технології автоматизації процесу тестування програмних продуктів. **Метою** роботи є оптимізація часу й витрат на проведення автоматичного тестування програмного продукту. В роботі вирішені наступні **задачі**: проведення аналізу існуючих систем автоматизації тестування програмних продуктів; формування системи критеріїв вибору систем автоматизації тестування; розробка формалізованої моделі процесу вибору; розробка алгоритму вибору системи автоматизації з урахуванням вимог замовника; розробка UML діаграм для подання функціональних можливостей розробленого додатку; розробка додатку для інформаційної підтримки процесу вибору. Для розв’язання вказаних задач були використані **методи** системного аналізу, теорії множин, технології розробки крос-платформних додатків. Отримані наступні **результати**. Проаналізовані найбільш популярні системи автоматизації тестування, виділені їх область застосування й основні можливості. Виділені критерії вибору, які розділяються на кількісні та якісні. Запропонована формалізована модель вибору систем автоматизації тестування з урахуванням їх характеристик і вимог замовника. Розроблена UML діаграма відображає функціональні можливості розроблюваної підсистеми. Запропонований алгоритм визначення рекомендованої системи автоматизації тестування дозволяє враховувати вектори критеріїв для систем тестування. На основі формалізованої моделі та алгоритму була розроблена підсистема, яка на основі введених критеріїв вибору дозволяє визначити оптимальний варіант системи автоматизації тестування. **Висновки**: інформаційна підтримка вибору системи автоматизації тестування програмних продуктів на основі розробленого алгоритму враховує вимоги замовника й характеристики існуючих систем, що дозволяє обирати найбільш переважний варіант серед можливих систем. Основним результатом роботи розробленої підсистеми є отримання користувачем рекомендацій з використання системи автоматичного тестування з урахуванням вимог замовника.

Ключові слова: тестування програм; системи автоматизації тестування; критерії вибору системи; модель багатокритеріального вибору; алгоритм вибору системи; діаграма прецедентів.

Бібліографічні описи / Bibliographic descriptions

Попов А. В., Момот М. О., Єлізєва А. В. Вибір системи автоматизації тестування з урахуванням вимог замовника. *Сучасний стан наукових досліджень та технологій в промисловості*. 2022. № 1 (19). С. 40–46. DOI: https://doi.org/10.30837/ITSSI.2022.19.040

Popov, A., Momot, M., Yelizieva, A. (2022), "Choosing the test automation system according to customer requirements", *Innovative Technologies and Scientific Solutions for Industries*, No. 1 (19), P. 40–46. DOI: https://doi.org/10.30837/ITSSI.2022.19.040