**УДК 004.415.2**
**МРНТИ 68.15.21** DOI 10.55452/1998-6688-2021-18-1-157-161

# INFORMATION SYSTEMS FOR MACHINE INTELLIGENCE TO AUTOMATED SOFTWARE TESTING

**SERALINA N.**

*Kazkh-British Technical University*

***Abstract:*** *The methods of development software develop rapidly. The testing of software has a great role in developing a good product. Many technologies assembled into all aspects of performance, based on software testing. Many advanced automation tools use in a set of test design and validation tests based on the artificial intelligence. The important thing is to focus on changes, to work on basis of collective reasoning of the test command and other commands analogues. The methods of the quality testing are based on the information provided in the modern digital world. The business is relying on new fast processes to provide automatic testing of software. Applying approaches of solutions to financial organization allows increase the transparency of all steps of software development. These steps can help systems show more percentage of the test case rate, can save time and money, but also effectively solves the problem of scaling the process and errors.*

*In this paper, we research information systems for machine intelligence to automated software testing. The aim is divided to tasks: the importance of artificial intelligence, the necessary stage of Software Development - Testing and Quality Controlling System, research of main automation tools.*

*We concluded that use of intellectual intelligence and machine learning: allows automating the repeating process and usage of the database; delivers superb intellectual product; adapts to the progressive algorithm of learning; adds more depth analysis of multiple objects; allows retrieving the maximum amount of data from the databases.*

***Key words:*** *artificial intelligence, machine learning, payment system, web services, software requirements, framework*

# АВТОМАТТАНДЫРЫЛҒАН БАҒДАРЛАМАЛЫҚ ЖАСАҚТАМАНЫ ТЕСТІЛЕУГЕ МАШИНАЛЫҚ ИНТЕЛЛЕКТКЕ АРНАЛҒАН АҚПАРАТТЫҚ ЖҮЙЕЛЕР

***Аңдатпа:*** *Бағдарламалық жасақтама жасау әдістері қарқынды дамып келеді. Бағдарламалық жасақтаманы тестілеу жақсы өнімді жасауда үлкен рөл атқарады. Бағдарламалық жасақтаманы тестілеу негізінде көптеген технологиялар өнімділіктің барлық аспектілеріне жинақталған. Автоматтандырудың біршама жетілдірілген құралдарында жасанды интеллектке негізделген тестілік жиынтықтар мен тексеру тестілері қолданылады. Сынақ тобы мен басқа аналогтық топтардың ұжымдық пайымдаулары негізінде жұмыс жасау, өзгерістерге назар аудару маңызды. Сапаны қамтамасыз ету әдістері қазіргі цифрлық әлемде ұсынылған ақпаратқа негізделген. Бағдарламалық жасақтаманы автоматты түрде тестілеуге тез мүмкіндік беру үшін бизнес жаңа процестерге сүйенеді. Қаржы институтына шешім қабылдау тәсілдерін қолдану бағдарламалық жасақтама жасаудың барлық кезеңдерінің ашықтығын арттыра алады. Жасалған қадамдар жүйелерге сынақ жағдайларының жоғары пайызын көрсетуге көмектеседі, уақыт пен ақшаны үнемдейді, сонымен қатар процестің масштабталуы мен қателіктер мәселелерін тиімді шешеді.*

*Бұл мақалада біз автоматтандырылған бағдарламалық қамтамасыздандыруды тестілеуге арналған машиналық оқыту мен жасанды интеллекті қолданатын ақпараттық жүйелерді зерттейміз. Мақсат міндеттерге бөлінеді: жасанды интеллектің маңыздылығы, бағдарламалық жасақтама жаса-*

*қажетті кезеңі – тестілеу және сапаны бақылау жүйесі, негізгі автоматтандыру құралдарын зерттеу. Шешілі: интеллектуалды интеллект пен машиналық оқытуды қолдану: қайталанатын процесті автоматтандыруға және мәліметтер базасын пайдалануға жағдай жасайды; жоғары интеллектуалды өнімді жеткізеді; прогрессивті оқыту алгоритміне бейімделеді; бірнеше объектілерді тереңірек талдауды қосады; мәліметтер базасынан мәліметтердің максималды көлемін алуға мүмкіндік береді.*

***Түйінді сөздер:*** *жасанды интеллект, машиналық оқыту, төлем жүйесі, веб-қызметтер, бағдарламалық жасақтамаға қойылатын талаптар, құрылым*

## ИНФОРМАЦИОННЫЕ СИСТЕМЫ МАШИННОГО ИНТЕЛЛЕКТА ДЛЯ АВТОМАТИЧЕСКОГО ТЕСТИРОВАНИЯ ПО

***Аннотация:*** *Методы разработки программного обеспечения стремительно развиваются. Тестирование программного обеспечения играет большую роль в разработке хорошего продукта. Многие технологии объединены во все аспекты производительности на основе тестирования программного обеспечения. Передовые инструменты автоматизации используют в наборе тестов и проверочных тестов на основе искусственного интеллекта. Важно сосредоточиться на изменениях, работать на основе коллективных рассуждений тестовой команды и других аналогов команд. Методы проверки качества основаны на информации, представленной в современном цифровом мире. Бизнес полагается на новые быстрые процессы для обеспечения автоматического тестирования программного обеспечения. Применение подходов решений к финансовой организации позволяет повысить прозрачность всех этапов разработки программного обеспечения. Эти шаги могут помочь системам показать больший процент от частоты тестовых случаев, могут сэкономить время и деньги, но также эффективно решают проблему масштабирования процесса и ошибок.*

*В этой статье мы исследуем информационные системы с использованием машинного обучения и искусственного интеллекта для автоматизированного тестирования программного обеспечения. Цель разделена на задачи: важность искусственного интеллекта, необходимый этап разработки программного обеспечения – система тестирования и контроля качества, исследование основных средств автоматизации. Вывод: использование интеллектуального интеллекта и машинного обучения позволяет автоматизировать повторяющийся процесс и использование базы данных; поставляет превосходный интеллектуальный продукт; адаптируется к прогрессивному алгоритму обучения добавляет более глубокий анализ нескольких объектов; позволяет получить максимальный объем данных из баз данных.*

***Ключевые слова:*** *искусственный интеллект, машинное обучение, платежная система, веб-сервисы, программные требования, фреймворк*

### 1 Introduction

The methods of development software develops rapidly. It can be checked and used for software tools and technology development. In the first glance, this is the necessary amount of education, to increase the software product and the quality of the long-lasting product. The testing of software has a great role in developing a good product. To find potential solutions to address all of these challenges, extensive efforts have been devoted to this field from many disciplines, including neuroscience, artificial intelligence, cognitive science, computational theory, statistics, computer science and engineering design and others [1].

The main purpose of work is to research information systems for machine intelligence to automated software testing. The aim is divided to tasks: the importance of artificial intelligence,

the necessary stage of Software Development - Testing and Quality Controlling System, research of main automation tools.

Many technologies are built into all aspects of performance, based on software testing. Many advanced automation tools use in a set of test design and validation tests based on the artificial intelligence (AI). The AI support for the quality test extracted in technical batches, which used to solve multiple problems when testing. The main purpose is to understand the functional application of a common user or tester's demands. The development stage should expect that the developers' team need careful usability and availability. It is important to focus on changes, to work on basis of collective reasoning of the test command and other commands analogues. Testing and Quality Controlling System (QCS) – widely developed and important task of software development stage. The optimal approach is automation of testing, many IT companies' according World Quality Report approximately 60% of commercial projects are used Agile methods workplace, only 16% uses automated QCS process. In process, the important role takes machine learning and AI methods. In the emerging digital economy, the main aim of commerce is to increase the credibility of justice. At Agile Development Agile subsets can be minimized the amount of time spent on product. The automation is based on the practical uniqueness of supporting high-speed processes, which is necessary for competition in modern markets. An important aspect of testing that separates it from other forms of code verification is that a test involves executing the code and examining the results instead of reviewing the code itself. Because of this feature, software testing is vital to finding unexpected errors and ensuring that the system functions correctly [2]. The methods of the quality testing is based on the information provided in the modern digital world. The business is relying on new fast processes to provide automatic testing of software.

According to [3] we build the model of web-based framework for financial organization. We offered logical architecture of payment system.

In developing of software can be used Model-Driven approach. The authors [4] demonstrate the key idea to apply Model-Driven QoS-aware approach to the wireless sensor network that are controlled by the network of "smart" controllers. They propose Model-Driven Architecture, the meta-model and its semantics, as the basis for an adaptation framework.

In developing of software there can be failures. The author [5] proposed a system that performs a dynamic selection of regressors to estimate the number of software failures. The author suggested a method of dynamic selection of regressors using instance grouping. In the study, it is mentioned that this regression area in failure prediction it is little explored, as more emphasis is placed on classification activities.

**2 Main part**

Applying AI and machine learning allows increase the transparency of all steps of software development. In addition, AI can produce tests that are just quick and easy. Controlled AI systems show more than a percentage of the test case rate, can save time and money, but also effectively solves the problem of scaling the process and errors. AI can reflect and predict typical problems, which reflected in the double-sided analysis. In an effort to deal with the increasing complexity of software systems that have to operate in dynamic, unpredictable, and heterogeneous environments, Agent-Oriented Software Development has been put forward [6]. There are several examples of testing tools.

**Selenium**

Programmers and testers in software web application testing processes use selenium. It is famous with open source framework code. The testing tool can be integrated with other tools such as Katalon Studio, Watir, Protractor and Robot Framework. Thar is the main feature in meeting requirements of testers and users. It is important to users to define necessary values, test scripts and nondrug test jobs. The Selenium testing tool has maintenance in basic operating systems: Windows, Mac, Linux. It can support main browsers, such as Chrome, Firefox, Internet

Explorer, and Headless. The used programming tools are Java, Groovy, Python, C #, PHP, Ruby and Perl. Industrial automation systems are relatively static and deterministic with only small changes to them during their operation. This can be used as an advantage when designing the security for the industrial automation systems. Any information known about the operation of the industrial automation systems can be used when designing the security for them [7].

**Katalon Studio**

Programmers, testers to test web, API, mobile and desktop test automation, use Katalon Studio. It deploy on operating systems: Windows, Mac and Linux. The main feature is that tool can be used to beginners of codeless beginners and can infinite testing extension for experts. It can integrates with other testing tools, such as Selenium, Appium, QTest, JIRA, Jenkins, and Git. Katalon. The most important feature is Analytics, which shows the tester or user to access the full prediction of the testing process. There can be used custom metrics, screen displays, and video metrics, charts, and graphs. Testing methods have improved a lot from the early days of programming and more tools are introduced to help to automate the processes. Proper tooling and habits lead to faster execution times, better analytics and more fluent reliable shipping of software. [8]. It reduce potential risks: broken links, navigation, web security and other risks. Katalon Studion is used in many companies over that 160 countries. There are hundreds of GitHub repositories and online courses.

**Unified Functional Testing (UFT)**

UFT / QTP is an automated functional testing tool by Micro Focus. The programmers, testers uses that in alpha and beta tests to identify bugs. It was earlier known as QTP (QuickTest Professional). QTP was originally designed by the company Mercury Interactive which was acquired by Hewlett Packard (HP) in 2006. In 2011 there was implementation of version 11.5, QTP was renamed to UFT. In Sep 2017, was designed, supported and maintained by Micro Focus. The main feature of tool is that it is used for functional, regression and service testing. It helps to improve user actions on a web or client based computer application and test and identify bugs on the same actions for different users, different data set, on various Windows operating systems and/or different browsers[9].

**IBM Rational Functional Tester**

IBM RFT – is a platform for the functional data testing and regression software. It supports the testing spectrum of the application spectrum, defined in different programming languages, including .Net, Java, SAP, Flex and Ajax. RFT supports Visual Basic .Net and Java in script language formats. RFT has the unique function of Storyboard testing. It also combines with automatic testing of the degree to which users are stored and visualized in the video of the subsequent image - screenshots of the application in different stages of greater testing. The latest single-band RFT platform provides enhanced integration with IBM Jazz custom cycle management systems (printers, IBM Rational Team Concert and Rational Quality Manager) [10]. Likewise, each one of them has its own unique tool, which is unique for resolving the post-processing tasks by means of automatic testing software. There can be used tools that allows integrating the necessary tests and integrations, as well as effectively processing the test and printing tasks from the application [11].

**SOAPUI**

Free product for functional testing with a software code derived from SOAP and REST. It was built on the core Java programming language. Mostly is used for direct and functional loading testing based on multiple multifunctional parameters. The graphical interface is user-friendly and has a simple design and use. Some testing function features allow user to protect the website from hackers and viruses [12].

3 Conclusion

According examples of testing tools can be concluded that use of intellectual intelligence and machine learning:

• Allows automating the repeating process and usage of the database;

• Delivers superb intellectual product;

• Adapts to the progressive algorithm of learning;

• Adds more depth analysis of multiple objects;

• Allows retrieving the maximum amount of data from the databases.

In future, in our research work we will develop analyzing and usage of self-adaptive systems for machine intelligence.

## REFERENCES


1. Haibo He. Self-adaptive systems for machine intelligence, John Wiley & Sons, 1-2, 2011.
2. Elona Wallengren Robin S. Sigurdson, Machine Intelligence in Automated Performance Test Analysis, Gothenburg, Sweden, 4 page, 2018.
3. Seralina N. A model for the web-services based framework. ICINCO 2020 Final program and book of abstracts. 17th International Conference on Informatics in Control, Automation and Robotics, Portugal. Online streaming, July 2020 , 26 page (https://www.insticc.org/node/technicalprogram/icinco/2020 )
4. Akzhalova A., Alexeev M., Sarsembayev B. Model-Driven Architecture for the QoS-Based Adaptive Sensor Network System. Springer International Publishing Switzerland, Book Chapter, Business Modeling and Software Design, 43-61 page, 2015.
5. Santosh Singh Rathore and Sandeep Kumar. An approach for the prediction of number of software faults based on the dynamic selection of learning techniques. IEEE Transactions on Reliability, 68(1):216-233, 2019.
6. A. Sturm and O. Shehory. Agent-Oriented Software Engineering: Revisiting the State of the Art. In O. Shehory and A. Sturm, editors, Agent-Oriented Software Engineering, pages 13–26. Springer Berlin Heidelberg, 2014.
7. R. Zurawski, M. Naedele, Integration Technologies for Industrial Automated Systems – IT Security for Automation Systems, CRC Press 2006, pp. 23-1 – 23-15.
8. H. Mohanty, J. Mohanty, and A. Balakrishnan, Trends in Software Testing, ISBN: 9789811014147. DOI: 10.1007/978-981-10-1415-4. [Online]. Available: https://link-springer-com.ezproxy.utu.fi/content/pdf/10.1007%7B%5C%%7D2F978-981-10-1415-4.pdf.
9. Graham Smith. The intelligent solution: automation, the skills shortage and cyber-security. Computer Fraud & Security. Volume 2018, Issue 8, August 2018, Pages 6-9.
10. Maria José Sousa, Álvaro Rocha. Digital learning: Developing skills for digital transformation of organizations. Future Generation Computer Systems. Volume 91, February 2019, Pages 327-334
11. Adriana Giret, Emilia Garcia, Vicente Botti. An engineering framework for Service-Oriented Intelligent Manufacturing Systems. Computers in Industry. Volume 81, September 2016, Pages 116-127.
12. Graham Smith, The intelligent solution: automation, the skills shortage and cyber-security. Computer Fraud & Security, Volume 2018, Issue 8, August 2018, Pages 6-9.