UDC 004.056 **doi: 10.20998/2522-9052.2021.1.12**

Serhii Semenov, Viacheslav Davydov, Daryna Hrebeniuk

National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine

RESEARCH OF THE SOFTWARE SECURITY MODEL AND REQUIREMENTS

Abstract. The **subject** of research in the article is a software security model. The **aim** of the work is to research the quality characteristics of the software and requirements for the software security in order to improve their safety. The article solves the following **tasks**: researching the shortcomings of the existing security model in order to identify its main shortcomings; study of the quality characteristics of software that affect its security in order to identify the possibility of improving the quality of software. The following **results** were obtained: on the basis of the analysis of the existing model of software security, the main features of the attributes of this model were identified, their advantages and disadvantages were given. On the basis of the conducted analytical study, the necessity of improving the existing model of ensuring the security of software has been proved. Existing requirements for software and characteristics that affect its quality are considered. The characteristics of software security are highlighted, the indicators of which should be improved. **Conclusions**: a software security model has been studied. The need to develop this model is shown by introducing the possibility of adapting the existing requirements for the security of software tools throughout the entire life cycle of software development; the study of the quality characteristics of software showed that to ensure its security, it is necessary to improve the following characteristics: integrity, authentication, confidentiality, access control. However, it was shown that an increase in these characteristics can lead to a deterioration in other indicators of software quality: portability, maintainability, performance.

Keywords: software security; software lifecycle; cybercrime.

Introduction

In today's world of ubiquitous computerization, the creation of secure computer systems is getting an increasing role. This is due to the fact that with the development of information technology, the qualitative and quantitative characteristics of malicious influences associated with information technology and computer systems, in particular, cyberattacks, are increasing. Losses from cybercrime reach 100 billion dollars a year worldwide, and millions of dollars a year in Ukraine. Thus, cybercrime was among the five most widespread economic crimes in the world, and in Ukraine, in particular [1, 2]. Comparison of losses from cybercrime on the same resource for previous years [2, 3] showed a tendency for its increase.

Analysis of recent research and publications. An integral part of existing computer systems is software. Studies [4-9] have shown that in terms of security, software is one of the most vulnerable components of computer systems. This is due to a number of factors of an objective and subjective nature (high cost of losses and, as a consequence, increased interest of attackers, shortcomings of modern tools and software development platforms, insufficient competence of user personnel, etc.).

However, in accordance with international and Ukrainian legislation, the requirements for the security of software are among the highest priority in the system of their qualitative assessment. For example, according to the Law of Ukraine "About Copyright" [10], software is intellectual property and requires protection from piracy, plagiarism, counterfeiting of various kinds, illegal distribution and other illegal actions and interventions.

The aim of the article. Thus, the problem of increasing the security of software becomes urgent. The purpose of the work is:

– study of the software security model in order to identify the main shortcomings of the model, as well as identify ways to eliminate them;

– study of the quality characteristics of software that affect its security in order to identify opportunities to improve the quality of software.

Materials and methods

A generalized model of software security assurance is shown in Fig. 1.

The main components of this model include:

– regulatory documents (e.g. ISO standards) regulating the process of ensuring safety. These safety requirements are strict and mandatory. This, in turn, imposes restrictions related to the relevance of these documents. So, many regulatory documents, in particular the Law "On Copyright", have the latest amendments dated 2016.

– Thus, these documents lose their relevance and require correction.

– the software itself;

– methods and means of ensuring safety;

– threats to security services - a potential event, action, process or phenomenon that can harm the software product and the company that produces the software product. They affect the software lifecycle;

– software life cycle - a set of requirements, software development and testing methodologies that define the processes, activities and tasks that are used in the delivery, development, testing, application, maintenance and termination of the use of software products.

As it shown at Fig. 1, one of the important attributes of the presented model is safety requirements. They are regulated by standards and development lifecycle. The development of technologies, development methodologies requires adjustment (adaptation) of the existing requirements for the security of software tools

throughout the life cycle of software development. Unfortunately, the developers of standards and specifications do not pay enough attention to this issue at the moment.

This is largely due to the lack of access to the confidential information required for this by private companies (the number of companies that release software products is too large for all their recommendations to be taken into account; a number of requirements of some companies may contradict the requirements of other companies, which may be due to the difference financial support and target audience; etc.), and possible dynamic changes in the software life cycle.

In Fig. 2 describes the life cycle of the software product. It should be noted that the main results of the software requirements analysis process are:

– determination of requirements for software elements of the system and their interfaces;

– analysis of requirements for software for correctness and testability;

– analysis of the impact of software requirements on the operating environment;

– identifying compatibility and traceability between software requirements and system requirements;

– determination of priorities for the implementation of requirements for software;

– assessing changes in software requirements in terms of cost, work schedules and technical impacts.

Fig. 2 shows that the results of the analysis of software requirements significantly affect decision making, design architecture, software development and testing. At the same time, the requirements at each stage are adapted taking into account the identification of the specifics of the software, development methodologies, development tools (e.g., frameworks, design patterns, development environment) and other factors.

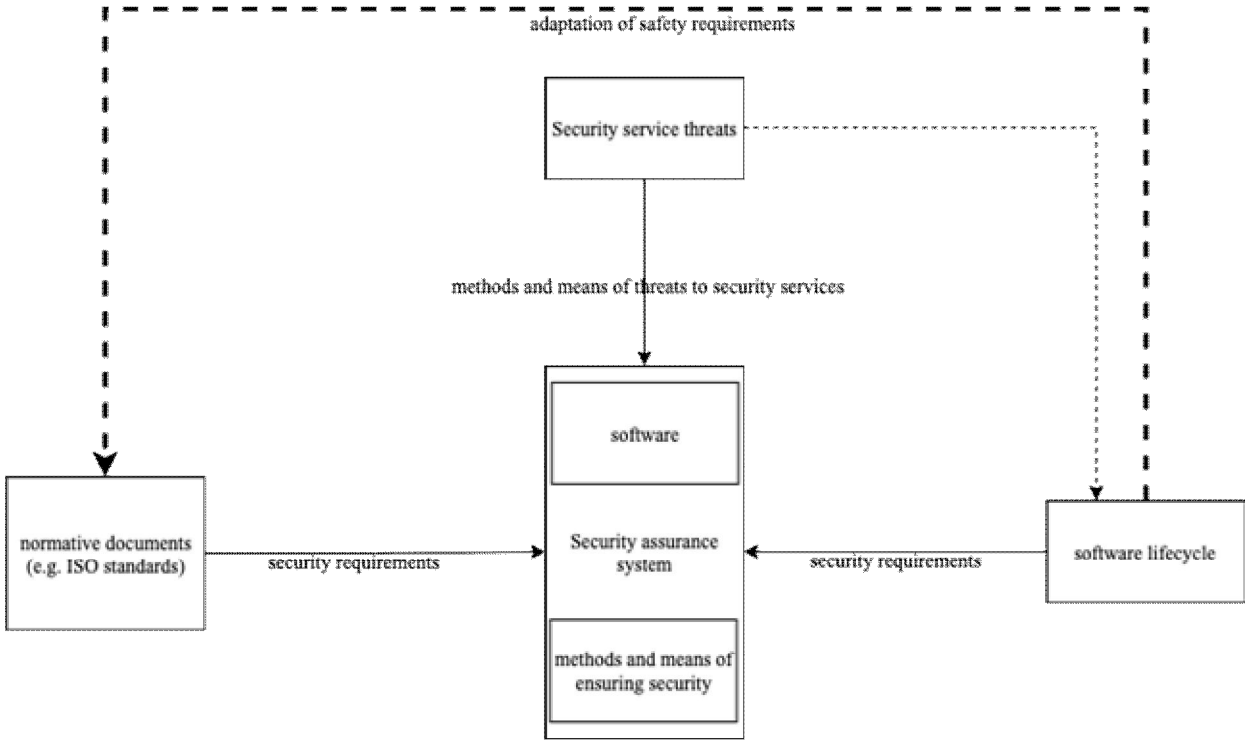

Fig. 1. Generalized software security model

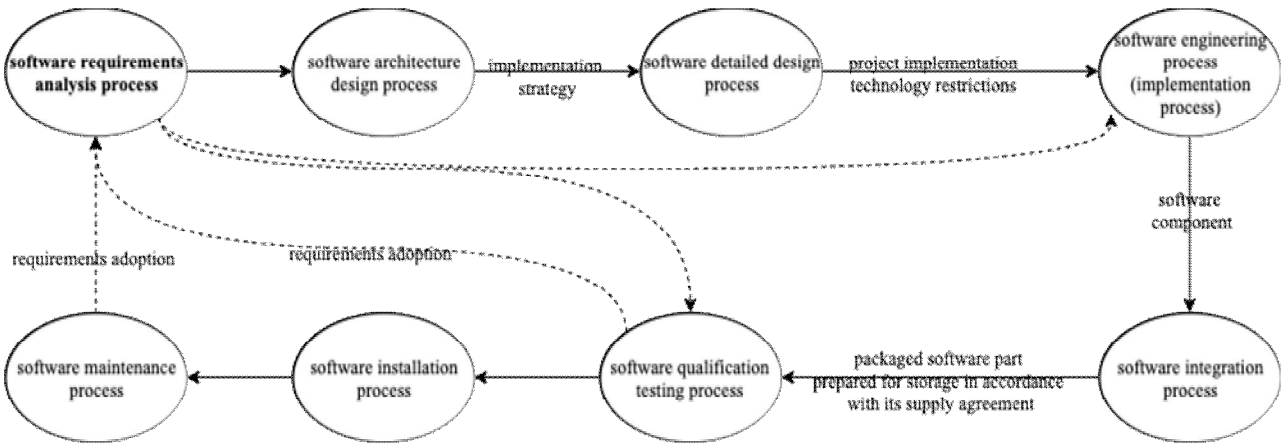

Fig. 2. Schematic description of the software lifecycle

It should also be noted that it is in the testing and maintenance processes of software that requirements adaptation should take place. Analysis of the standards ISO 29148 [11], ISO 9126 [12], ISO/IEC 25010 ([13], Fig. 3) and others [14-16] made it possible to classify the requirements for a software product. Schematically, these requirements can be presented in the form of Fig. 4. So, there are 3 main groups of requirements:

– system requirements determine the external conditions for the implementation of system functions and restrictions on the creation of a product, as well as requirements for the description of software and hardware subsystems. System requirements impose restrictions on the architecture of the system, the means of its visual presentation and functioning;

– functional requirements are a list of functions or services that the system must provide, as well as restrictions on data and behavior of the system during its execution;

– non-functional requirements determine the conditions for performing functions (for example, protecting information in a database, authenticating access to software, etc.) in an environment that are not directly related to functions, but reflect the needs of users to perform them. These requirements characterize the principles of interaction with environments or other systems, and also determine the indicators of uptime, data protection and quality achievement, taking into account the recommendations of the standard used.

At the same time, the studies carried out have shown that the practical implementation of the assigned tasks of increasing security can lead to a deterioration in other indicators of software quality, for example, such as:

– portability. The introduction of additional means of protection may entail the use of architecture-specific structures, the analogues of which may not be found for the rest;

– productivity. Protection mechanisms of a software product, such as obfuscation and encryption, affect performance and require additional processor resources to perform operations;

– maintainability. For example, turning off the debug information necessary to increase software confidentiality leads to a deterioration in analyzability, in particular, when the stack is received, a call in case of errors occurs.

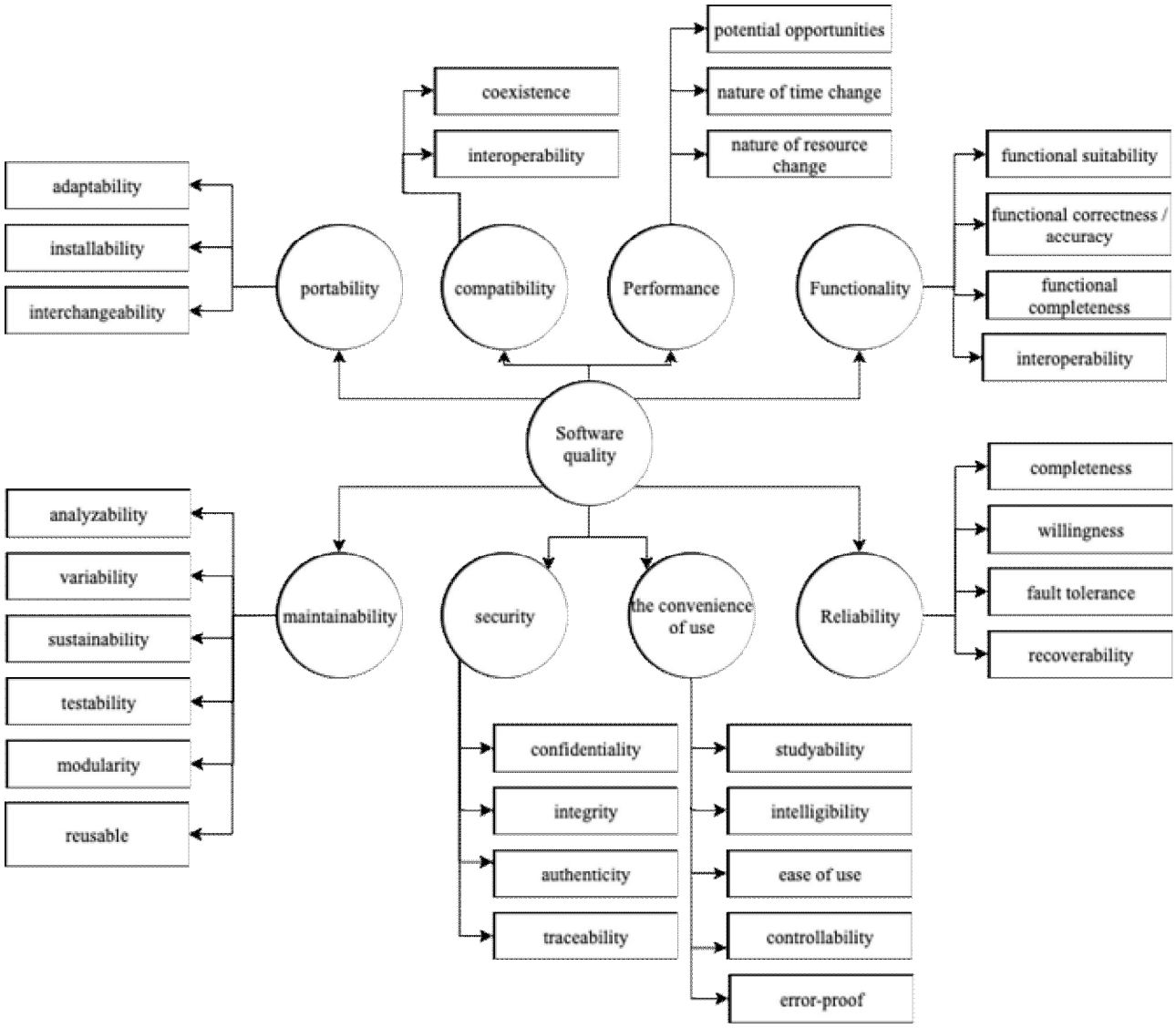

Fig. 3. ISO / IEC 25010 software quality classification

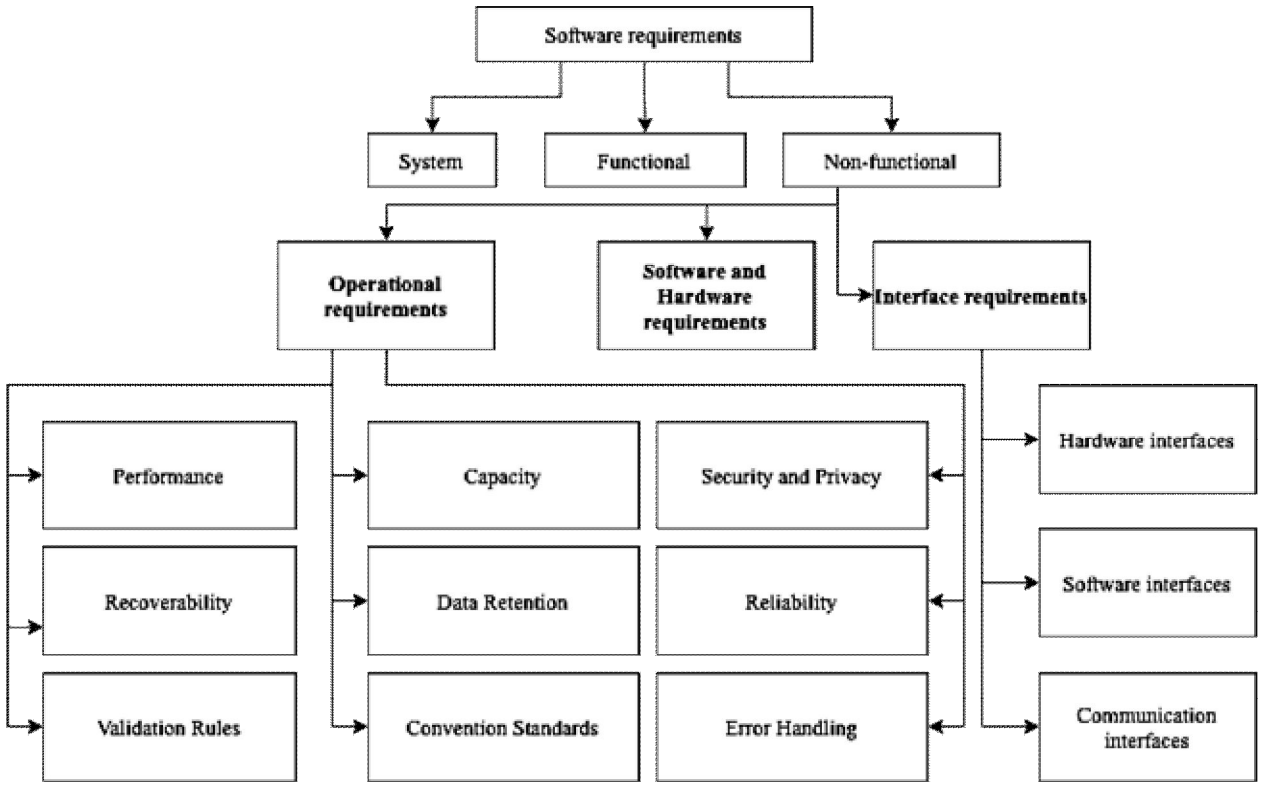

Fig. 4. Software components classification

The ISO 9126 defines the following security metrics covering the characteristics under study:

– ability to track access: X = A / B, where, A is the number of "user accesses to the system and data" registered in the access history database, B is the number of "user accesses to the system and data" performed during the assessment process;

– access control: X = A / B, where, A is the number of detected prohibited operations of various types, B is the number of prohibited operations of various types specified in the specification;

– prevention of data corruption: X = 1 - A / N, where, A is the number of occurrences of the main cases of data corruption, N is the number of test data that tried to cause data corruption;

Also, the standard defines the metrics of software quality characteristics that have a negative impact on security metrics:

– analyzability characteristic: a metric of analyzability (Can the user accurately identify the action that led to the failure? Can the maintainer easily pinpoint the action that led to the failure?): X = A / B, where, A - Number data actually recorded during operation, B - The amount of data that was planned to be recorded and sufficient to observe the state of the software during operation;

– performance characteristic: performance metric (How many tasks can be successfully completed in a given period of time?): X = A / T, where A is the number of completed tasks, T is the observation time.

– mutability characteristic: metric of the complexity of the modification (Can the maintainer easily change the software to solve the problem?): T = Sum (A / B) / N, A - Work time spent on the change, B - The amount of software change, N - Number of changes;

– consistency and maintainability characteristic: a metric of the degree of consistency in maintainability (How well maintainability of products complies with applicable regulations, standards and conventions?): X = 1 - A / B, where, A – number of points of consistency in maintainability that were not met during testing, B - total number of specified maintainability consistency points. Based on the research carried out, a matrix of the influence of models and methods that increase software security on the metrics of other characteristics of software quality has been formed. These matrices are presented in Table 1.

Table 1 – **Influence of basic characteristics of software quality on characteristics of its security**

	Ability to track access	**Controlled access**	**Data corruption prevention**
Analyze ability	+	—	—
Modification difficulty	+	+/-	+
Consistency and maintainability	+	+/-	—
Performance	+	+	+

For example:

– to increase the value of the metric of the ability to track access, mechanisms such as obfuscation, encryption, access control lists (ACL), etc. are used. These mechanisms introduce additional non-functional (redundant) actions, which leads to a decrease in performance, the ability to analyze, and increases the complexity of the modification and decreases the maintainability;

– to increase the value of the access control metric, authorization mechanisms, digital signatures, access control lists (ACL) are used. Additional checks lead to performance degradation. If it is possible to grant access rights to authorized representatives of the company, these mechanisms do not affect other studied metrics;

– to increase the value of the metric for preventing data corruption, digital signature and checksum methods are used, which increases the complexity of the modification, and also affects performance due to the redundancy of the code. Thus, the need to solve the problem of increasing software security while simultaneously meeting the requirements for the main indicators of its quality is actualized.

Conclusions and prospects for further development

1. A software security model has been investigated. The necessarily to develop this model is shown by introducing the possibility of adapting existing requirements for the security of software tools throughout the entire life cycle of software development.

2. Research of the quality characteristics of software showed that to ensure its security, it is necessary to improve the following characteristics: integrity, authentication, confidentiality, access control. However, it was shown that an increase in these characteristics can lead to a deterioration in other indicators of software quality: portability, maintainability, performance.

The findings showed the feasibility of developing research in the direction of improving software quality indicators while minimizing their influence on each other.

REFERENCES

1. Klimchak, M. (2018), "PwC Global Economic Crime and Fraud Survey 2018: Ukrainian findings", *PWC Ukraine*, available at: https://www.pwc.com/ua/uk/survey/2018/pwc-gecs-2018-ukr.pdf.
2. Krul, S. (2008), "Crimes in the Sphere of informative Technologies: the national and international Aspects", *Actual problems of improving of current legislation of Ukraine*, Vol. 20, pp. 200-204, available at: http://nbuv.gov.ua/UJRN/apvchzu_2008_20_32.
3. Krasnyansky, B. (2011), "PwC World Review of Economic Crimes", *PWC Ukraine*, available at: https://www.pwc.com/ua/uk/press-room/assets/gecs_ukraine_ua.pdf.
4. Roger A. Grimes (2017), *Hacking the Hacker: Learn from the Experts Who Take Down Hackers,* Chapter: *Software Vulnerabilities*, DOI: https://doi.org/10.1002/9781119396260.ch6.
5. Robert H. Sloan, Richard Warner (2019), *Why Don't We Defend Better?,* Chapter 6: *Software Vulnerabilities*, available at: https://www.taylorfrancis.com/chapters/software-vulnerabilities-robert-sloan-richard-warner/10.1201/9781351127301-2.
6. M. Aldea, D. Gheorghică and V. Croitoru (2020), "Software Vulnerabilities Integrated Management System", *2020 13th International Conference on Communications (COMM)*, Bucharest, Romania, pp. 97-102, DOI: https://doi.org/10.1109/COMM48946.2020.9141970.
7. Kovalenko, A. and Kuchuk H. (2018), "Methods for synthesis of informational and technical structures of critical application object's control system", Advanced Information Systems, Vol. 2, No. 1, pp. 22–27, DOI: https://doi.org/10.20998/2522-9052.2018.1.04
8. Ruban, I., Kuchuk, H. and Kovalenko A. (2017), "Redistribution of base stations load in mobile communication networks", *Innovative technologies and scientific solutions for industries*, No 1 (1), P. 75–81, doi : https://doi.org/10.30837/2522-9818.2017.1.075.
9. Mozhaev, O., Kuchuk H., Kuchuk, N., Mozhaev, M. and Lohvynenko M. (2017), "Multiservise network security metric", *IEEE Advanced information and communication technologies-2017*, Proc. of the 2th Int. Conf, Lviv, pp. 133-136, DOI: https://doi.org/10.1109/AIACT.2017.8020083
10. *Law of Ukraine "On Copyright and Related Rights"* (1993), available at: https://zakon.rada.gov.ua/laws/show/3792-12?lang=en#Text.
11. (2018), ISO/IEC/IEEE 29148 Systems and software engineering — Life cycle processes — Requirements engineering, available at: https://www.iso.org/ru/standard/72089.html.
12. (2001), ISO/IEC 9126-1 Software engineering — Product quality — Part 1: Quality model, available at: https://www.iso.org/ru/standard/22749.html.
13. (2017), ISO/IEC/IEEE 12207 Systems and software engineering — Software life cycle processes, available at: https://www.iso.org/ru/standard/63712.html.
14. (2015), ISO/IEC 25024 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of data quality, available at: https://www.iso.org/ru/standard/35749.html.
15. (2011), ISO/IEC 25010 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models, available at: https://www.iso.org/ru/standard/35733.html.
16. (2008), ISO/IEC 25012 Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Data quality model, available at: https://www.iso.org/ru/standard/35736.html.

Received (Надійшла) 12.11.2020
Accepted for publication (Прийнята до друку) 20.01.2021

ВІДОМОСТІ ПРО АВТОРІВ/ ABOUT THE AUTHORS

Семенов Сергій Геннадійович – доктор технічних наук, професор, завідувач кафедри "Обчислювальна техніка та програмування", Національний технічний університет "Харківський політехнічний інститут", Харків, Україна;
Serhii Semenov – Doctor of Technical Sciences, Professor, Head of Computer Engineering and Programming Department, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine;
e-mail: s_semenov@ukr.net; ORCID ID: http://orcid.org/0000-0003-4472-9234.

Давидов Вячеслав Вадимович – кандидат технічних наук, доцент кафедри обчислювальної техніки та програмування, Національний технічний університет «Харківський політехнічний інститут», Харків, Україна;
Viacheslav Davydov – Candidate of Technical Sciences, Associate Professor of Computer Engineering and Programming Department, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine;
e-mail: vyacheslav.v.davydov@gmail.com; ORCID ID: https://orcid.org/0000-0002-2976-8422.

Гребенюк Дарина Сергіївна– аспірантка кафедри обчислювальної техніки та програмування, Національний технічний університет «Харківський політехнічний інститут», Харків, Україна;
Daryna Hrebeniuk – graduate student of Computer Engineering and Programming Department, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine;
e-mail: darina.gg1@gmail.com; ORCID ID: https://orcid.org/0000-0001-5331-2444.

Дослідження моделі та вимог до безпеки програмного забезпечення

С. Г. Семенов, В. В. Давидов, Д. С. Гребенюк

Анотація. Предметом дослідження в статті є модель забезпечення безпеки програмного забезпечення. **Метою** роботи є дослідження характеристик якості програмного забезпечення і вимог до безпеки програмних засобів з метою підвищення їх безпеки. У статті вирішуються наступні **завдання**: дослідження недоліків існуючої моделі забезпечення безпеки з метою виявлення основних її недоліків; дослідження характеристик якості програмного забезпечення, що впливають на її захищеність з метою виявлення можливості підвищення якість програмного забезпечення. Отримані наступні **результати**: на основі проведеного аналізу існуючої моделі забезпечення безпеки програмного забезпечення були виявлені основні особливості атрибути даної моделі, наведено їх переваги і недоліки. На основі проведеного аналітичного дослідження доведено необхідність вдосконалення існуючої моделі забезпечення безпеки програмних засобів. Розглянуто існуючі вимоги до програмного забезпечення та характеристики, що впливають на його якість. Виділено характеристики захищеності програмного забезпечення, показники яких повинні бути поліпшені. **Висновки**: досліджена модель забезпечення безпеки програмного забезпечення. Показана необхідність розвитку даної моделі шляхом введення можливості адаптації існуючих вимог до безпеки програмних засобів протягом усього життєвого циклу розробки програмного забезпечення; дослідження характеристик якості програмного забезпечення показало, що для забезпечення її захищеності необхідно підвищити наступні характеристик: цілісність, аутентифікація, конфіденційність, управління доступом. Однак, показано, що підвищення даних показників може спричинити за собою погіршення інших показників якості програмного забезпечення: переносимості, сопровождаемость, продуктивність.

Ключові слова: безпека програмного забезпечення; життєвий цикл програмного продукту; кіберзлочинність.

Исследование модели и требований безопасности программного обеспечения

С. Г. Семёнов, В. В. Давыдов, Д. С. Гребенюк

Аннотация. Предметом исследования в статье является модель обеспечения безопасности программного обеспечения. **Целью** работы является исследование характеристик качества программного обеспечения и требований к безопасности программных средств с целью повышения их безопасности. В статье решаются следующие **задачи**: исследования недостатков существующей модели обеспечения безопасности с целью выявления основных её недостатков; исследование характеристик качества программного обеспечения, влияющих на ее защищенность с целью выявления возможности повышения качество программного обеспечения. Получены следующие **результаты**: на основе проведенного анализа существующей модели обеспечения безопасности программного обеспечения были выявлены основные особенности атрибуты данной модели, приведены их достоинства и недостатки. На основе проведенного аналитического исследования доказана необходимость совершенствования существующей модели обеспечения безопасности программных средств. Рассмотрены существующие требования к программному обеспечению и характеристики, влияющие на его качество. Выделены характеристики защищенности программного обеспечения, показатели которых должны быть улучшены. **Выводы**: исследована модель обеспечения безопасности программного обеспечения. Показана необходимость развития данной модели путем введения возможности адаптации существующих требований к безопасности программных средств на протяжении всего жизненного цикла разработки программного обеспечения; исследование характеристик качества программного обеспечения показало, что для обеспечения её защищенности необходимо повысить следующие характеристик: целостность, аутентификация, конфиденциальность, управление доступом. Однако, показано, что повышение данных характеристик может повлечь за собой ухудшение других показателей качества программного обеспечения: переносимости, сопровождаемость, производительность.

Ключевые слова: безопасность программного обеспечения; жизненный цикл программного продукта; киберпреступность.