

**D. A. GOBOV**, Candidate of Technical Sciences (Ph.D.), National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Senior lecturer of the Department of Computer Science and Software Engineering of the Faculty of Informatics and Computer Science, Kyiv, Ukraine., e-mail: d.gobov@kpi.ua, ORCID: <https://orcid.org/0000-0001-9964-0339>

**O. V. ZUIEVA**, Candidate of Economic Sciences (Ph.D.), Analytics Consultant at CS ltd. Kharkiv, Ukraine, e-mail: olekszueva@gmail.com, ORCID: <http://orcid.org/0000-0001-9661-9657>

## EXAMINING SOFTWARE QUALITY CONCEPT: BUSINESS ANALYSIS PERSPECTIVE

Software quality is a critical aspect of software development that significantly impacts business performance and customer satisfaction. However, defining software quality can be challenging, as different sources provide various definitions and perspectives. The article presents a literature review of software quality, acknowledging an ongoing debate over the years regarding the definition of software quality and the methods used for its assessment. Among all the different ideas about software quality, the article highlights key concepts that are crucial in understanding software quality: meeting requirements, satisfying users, using software features, and spotting defects. The article also checks out international standards like ISO/IEC 25010:2011 and ISO/IEC 5055:2021, introducing terms such as "Quality in use" and "Structural Quality." Unveiling a tripartite perspective elucidated in international standards—internal quality, external quality, and quality in use - the article underscores the intricate interplay between subjectivity and objectivity. The subjective dimension, influenced by user perception and contextual factors, is juxtaposed with more objective criteria such as conformance to requirements and the absence of defects. The standards provide helpful perspectives, but the human side of things, like user feelings and specific contexts, makes finding a universal definition tricky. The pivotal role of business analysis and requirements engineering in ensuring software quality is underscored. Business requirements, stakeholder needs, and the quality of functional and non-functional requirements emerge as integral components. The article argues that software quality is intricately tied to the quality of its requirements, presenting a dual perspective: compliance with quality criteria and alignment with stakeholders' expectations and business goals. Practical software quality assessment is built upon the foundational understanding of contextual nuances, user needs, and operational conditions, all discerned through business analysis.

**Keywords:** software quality, business analysis, requirements, quality in use, internal quality, international standards.

**Introduction.** The question of assuring quality is always essential in the world of technology. Since people started automating their work, ensuring the quality of these automated tools has been a significant concern. It's hard to imagine any organization aiming to produce low-quality products because quality significantly impacts business performance and customer satisfaction [1]. The quality of software solutions can substantially affect organizations' financial performance and projects' overall success. According to the Consortium for Information & Software Quality (CISQ), in 2020, the total Cost of Poor Software Quality in the US reached \$2.08 trillion [2].

In [3], the author analyzed 15468 publications starting from the year 1954 and concluded that the interest in the topic of software quality is growing exponentially. According to the results obtained in [3], most research is dedicated to the software development process, advanced software engineering, software architectures, quality evaluation, software testing, machine learning, and data mining. It is noteworthy that, as indicated by the study's findings [3], there was a noticeable shift in focus during the late 2000s towards topics such as quality assurance in the early stages of software development life cycles and software process improvement. This underscores the increasing significance of these identified subject areas. In the ISO/IEC/IEEE 12207:2017 standard [4], it is mentioned that the initial stages of the software development technical process involve the identification and analysis of requirements. One of the typical stages for a software system is "exploration", which is facilitated by business analysis tools. Consequently, the quality assurance process should commence from this very initial stage. Software quality is a critical characteristic that software engineers

must choose and document at the outset of a project during the requirements definition process [5].

In organizing any process, including the evaluation and assurance of software quality, it is imperative to establish clear boundaries and objectives essential for achieving the desired outcomes. Developing a conceptual framework serves as a vital tool for delineating these boundaries. In essence, to establish effective processes for the assessment and assurance of quality, defining the concept of "Software Quality" is crucial. However, defining software quality can be complex, as different sources provide various definitions and perspectives.

**Literature review and problem statement.** As previously mentioned, a considerable body of literature is dedicated to examining software quality. However, the majority of works primarily center attention on models and methodologies for assessment, relying upon the definitions outlined in international standards. For example, in the [6], the authors explore the significance of software quality in connection with risk evaluation and security considerations. The article offers an examination of ISO standards pertaining to software quality and the advantages they offer to promote the utilization of software quality standards within the industry. In [7], the authors identified three fundamental dimensions for determining software quality: a set of quality factors, user satisfaction, and unexpected software performance or errors. Nonetheless, the primary emphasis of the work remains dedicated to the software quality measurement. The article [8] explores the evolving perspectives on software quality and their influence within the software engineering community. The authors acknowledge an ongoing debate over the years regarding the definition of software quality and the methods used for its

© Gobov D. A., Zuiueva O. V., 2023



**Research Article:** This article was published by the publishing house of NTU "KhPI" in the collection "Bulletin of the National Technical University "KhPI" Series: System analysis, management and information technologies." This article is distributed under a Creative Commons [Creative Commons Attribution \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/). **Conflict of Interest:** The author/s declared no conflict of interest.



assessment. Furthermore, they investigate variations in the emphasis placed on quality attributes within both research and practical application. The authors aim to identify the quality traits that practitioners consider crucial when assessing the quality of code snippets. Furthermore, the authors observe disparities in the evolution of the six quality attributes within software quality models over the past four decades.

The authors' focus on individual components, characteristics, and aspects of assessing software quality underscores the complex nature of studying this concept at a more general level. Karl Wieggers, as referenced in [9, p. 189], acknowledges that no comprehensive yet concise definition of Quality universally applies to software. He presents a segment of the definition of "Quality" from The American Society for Quality (2021a), which describes it as "A subjective term for which each person or sector has its own definition". The whole definition also includes practical aspects: "In technical usage, quality can have two meanings: 1) the characteristics of a product or service that bear on its ability to satisfy stated or implied needs; 2) a

product or service free of deficiencies. According to Joseph Juran, quality means "fitness for use"; according to Philip Crosby, it means "conformance to requirements" [10]. However, in the sixth edition of Juran's Quality Handbook, Joseph Juran and Joseph De Feo redefine it as "fitness for purpose" and highlight, "we must first agree on a meaning of the word so that an organization will know how to manage "it" [11, p.5]. In the [12], the authors assert that the quality is contingent upon the context and the interpretation of quality attributes, as well as the interrelationship among these attributes within a specific context; it is neither substantive nor rational to posit that a single definition can universally fit the diverse needs of all stakeholder groups. Undoubtedly, the notion of quality is broad and subjective. However, this subjective aspect of software quality becomes apparent in practical applications, while at a conceptual level, common attributes can be identified. Therefore, **the purpose of the article** is to identify the essential characteristics inherent in the concept of software quality.

Table 1 – The definitions of the "Software Quality"

Author, source	Definition
Petrasch R. (1999) [13]	the existence of characteristics of a product which can be assigned to requirements. In addition to this, we have to look at the characteristics that are not related to requirements: characteristics, which reduce the software quality (contra-productive) and "neutral" characteristics, which are not relevant for quality.
Tian J. (2005) [14, p.25]	a) may include many different attributes and may be defined and perceived differently based on people's different roles and responsibilities. b) high quality means none or few problems of limited damage to customers.
Nuseibeh B., Easterbrook S.M. (2007) [15]	a) fitness for purpose; b) an attribute of the relationship between software and the purpose for which it is used.
Juran J. M., De Feo J. A. (2010) [11]	a) features of product that meet customer needs and freedom from failures (p. 6) b) degree to which an inherent characteristic fulfills requirements (p. 1056).
Hossain A., Dr. Md. Kashem A., Sultana S. (2013) [16]	High levels of user satisfaction and low defect levels, often associated with low complexity.
Suman, Wadhwa M. (2014) [17]	conformance to explicitly state functional and performance requirements, explicitly documented development standards and implicit characteristics that are expected of all professionally developed software.
Alebeisat F., Alhalhouli Z., Alshabat T., Alrawashdeh T.I. (2018). [7]	Product attribute that meets the stringent performance and functional requirements, specific development criteria, and inherent functions that all professionally designed software must have.
Hussain S., Farid S., Mumtaz I. (2019) [12]	conformance to predefined specifications that meet the customers' needs i.e. perception of a user or customer that up to what extent the software product meets their need and expectations.
ANSI/ASQC A3. (1978) [18]	the totality of features and characteristics of a product or a service that bears on its ability to satisfy the given needs
ANSI/ IEEE Std 729-1983 [19]	a) The totality of features and characteristics of a software product that bear on its ability to satisfy given needs: for example, conform to specifications. b) The degree to which software possesses a desired combination of attributes. c) The degree to which a customer or user perceives that software meets his or her composite expectations. d) The composite characteristics of software that determine the degree to which the software in use will meet the expectations of the customer"
IEEE Std 610.12-1990 [20]	a) The degree to which a system, component, or process meets specified requirements. b) The degree to which a system, component, or process meets customer or user needs or expectations.
ISO / IEC 25010 : 2011 [21]	degree to which a software product satisfies stated and implied needs when used under specified conditions.
ISO 9000:2015 [22]	the degree to which a set of inherent characteristics (existing in object) of an object (e.g., system, product) fulfils requirements (need or expectation that is stated, generally implied or obligatory). The term "quality" can be used with adjectives such as poor, good or excellent.

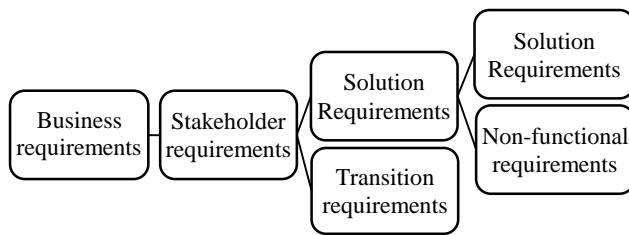


Fig. 1. Requirements classification scheme

**Analysis and results.** Table 1 reviews several definitions of the concept of "Software Quality" as presented in the existing scientific literature and international standards. In addition to the definitions presented in Table 1, ISO/IEC 25010:2011 [21] defines the concept of "Quality in use" as the degree to which specific users can use a product or system to meet their needs to achieve specific goals with effectiveness, efficiency, freedom from risk and satisfaction in specific contexts of use. Furthermore, ISO/IEC 5055:2021 [23] introduces the term "Structural Quality" as the degree to which a set of static attributes of a software product satisfies stated and implied needs for the software product to be used under specified conditions – a component of software quality. The standard also introduces the term "Internal Software Quality," characterizing the degree to which a set of static attributes of a software product satisfies stated and implied needs for the software product to be used under specified conditions.

Analyzing the diverse array of definitions provided, it becomes evident that international standards define Software Quality as a "degree" that can be good, poor, or excellent. In contrast, the authors' definitions revolve around terms such as "conformance," "fitness," "existence of characteristics," and "attributes." It should be noted that words like "conformance" and "fitness" can also be qualified with adjectives like "high" or "low," thus indicating a level or degree.

A comprehensive analysis of these definitions reveals several key concepts associated with the delineation of Software Quality:

1. Conformance to requirements or specifications - Software Quality relates to how well the software meets specific functional and performance criteria, whether explicitly stated or implicitly expected.

2. Customer or user perception of satisfaction needs and expectations - Software Quality depends on how users or customers feel about the software and whether it meets their needs and expectations.

3. Set of features, attributes, and characteristics of a software product - it's about the attributes and characteristics that affect the software's ability to do what it's supposed to do.

4. Existence of defects, problems, and damage - the presence of defects and problems in the software can affect its quality.

5. Specified conditions or context – the context and conditions under which the software is used also play a significant role in determining its quality, influencing the requirements and feelings of the user.

Points 2 and 5 introduce a subjective dimension to the concept of Software Quality. They emphasize the human

element, which acknowledges that what one person or group of users considers "good" or "excellent" may differ from the opinions of others and that quality can be influenced by the unique circumstances in which the software is employed. On the other hand, the remaining points, 1, 3, and 4, strive to objectify the concept by establishing more concrete criteria for evaluation.

It is worth noting that all of the above concepts, in one form or another, refer to certain levels of software requirements, which were the central element of business analysis [24]. The business analysis body of knowledge [25] proposed the following requirements classification scheme that can be applied to software as well (fig. 1):

Considering that business requirements describe the higher-level needs of the organization and measurable representations of goals the business is seeking to achieve [26]. This type of requirement is mentioned in the software quality definition according to [12, 15, 18, 19, 20, 21, 22,]. It allows quality assessment not from a particular user perspective but from the whole enterprise, a business area, or a specific initiative point of view.

At the same time, business requirements can not be achieved if stakeholder requirements are not met. In doing so, business requirements provide the context boundaries for identifying stakeholder requirements in the form of a problem to be solved or an opportunity to be realized. User needs satisfaction as conditions of software quality are mentioned in [11, 12, 16, 19, 20].

Functional requirements, which describe software behavior, and non-functional requirements, which describe conditions under which a solution must remain effective or qualities that a solution must have, are mentioned in the software quality definitions in [7, 10, 12, 13, 18, 19, 20, 22]. Some sources give special attention to non-functional requirements. For example, the quality model proposed in [21] includes eight categories, seven of which relate to non-functional requirements. On this basis, it has even been suggested in [27] that software quality depends only to a relatively small extent on functional conformance. This statement should be recognized as erroneous since functional requirements cover the needs of stakeholders. Therefore, the product quality and non-functional requirements without implementing functional ones do not make sense. It is confirmed in particular by the quality in use model from [21].

Understanding quality as the implementation of a software product in accordance with a requirements specification [7, 12, 19, 20, 22] directly refers to paragraph 3 of the term "requirement" defined in the ISO standard [20]:

1. A condition or capability needed by a user to solve a problem or achieve an objective.

2. A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document.

3. A documented representation of a condition or capability as in 1 or 2.

It allows us to conclude that a software product's quality depends on the requirements' quality, which are usually created before the actual creation of the program

code begins. The quality of requirements can be discussed in two dimensions: compliance with quality criteria and compliance of requirements with stakeholders' expectations and business goals. The first is provided through requirements verification activities that aim to check whether the requirements are sufficiently defined and structured so that the development team can use them in the software solution's design, development, and implementation. The second one is provided through validation activities to ensure requirements support the delivery of the expected benefits to stakeholders. The fact that business, stakeholders, and functional/non-functional requirements are the main deliverables of business analysis tasks allows us to say that the quality of business analysis execution directly affects the quality of the software product. This conclusion is confirmed by the study [28], which shows that problems such as "Customer dissatisfaction/loss of trust", "Low quality of the product/service", and "Gap between end-user real needs and implemented functionality" are caused by problems in business analysis and requirements engineering activities.

Essentially, Software Quality emerges as a multifaceted concept that seamlessly integrates subjective and objective components. While the subjective aspect is linked to user experiences and the dynamic context of use, the objective elements aim to quantify and assess quality based on defined standards, inherent software attributes, and the absence of issues. This multifaceted nature underscores the need for each organization to identify which specific quality aspects are most significant. By doing so, organizations can tailor their quality assessments to align with their unique priorities and requirements, acknowledging that Software Quality is not a one-size-fits-all proposition.

The interplay between subjectivity and objectivity makes Software Quality a comprehensive concept encompassing diverse viewpoints, evaluation approaches, and models. The quality model outlined in international standards offers a tripartite perspective: internal quality, external quality, and quality in use. Internal Quality provides an insider's view of the software, focusing on characteristics typically accessible during development. External Quality takes an outsider's perspective, concentrating on properties related to the software's performance during execution. Quality in use relates to the practical application of the software in its operational environment, serving the needs of specified tasks performed by designated users. It's important to note that these three dimensions are interconnected. Internal quality influences external quality, which, in turn, impacts quality in use [5].

It's crucial to emphasize another significant notion here. The context and conditions under which software operates and the user's needs and expectations are defined through the business analysis process. These aspects are documented in business analysis artifacts and form the foundation for requirement formulation. Consequently, it becomes apparent that software quality cannot be measured unless the requirements are correctly understood [15]. Hence, requirements engineering plays a pivotal role in our capacity to comprehend and evaluate software quality. In essence, understanding the specific circumstances and user

needs forms the bedrock upon which software quality assessment is built.

**Conclusions.** The research aimed to identify the fundamental characteristics inherent in the concept of software quality. Drawing on ISO standards and diverse literature, five key concepts associated with defining Software Quality emerged: conformance to requirements or specifications, user perception of satisfaction needs and expectations, set of features, attributes, and characteristics of a software product, existence of defects, problems, and damage, and specified conditions or context. The synthesis of subjective and objective elements, along with the interconnected dimensions of the quality model, underscores the multifaceted nature of software quality. The analysis highlights that the intricate nature of software quality requires a nuanced approach, and a comprehensive understanding of contextual nuances, user needs, and operational conditions is imperative. The findings emphasize the pivotal role of business analysis and requirements engineering, particularly in the early development stages, shaping the software quality trajectory. Measuring software quality demands a thorough understanding of requirements; lacking a clear grasp of specifications and expectations makes accurately assessing software quality challenging. Essentially, the accuracy and completeness of requirements are fundamental for a practical evaluation of software quality and ensuring a high level of it.

In our [29] publication, we establish the influence of project context on the content of business analysis documents. As a result, our ongoing research will primarily focus on assessing the quality of requirements and business analysis activities as a whole while considering the context and examining how this, in turn, affects the overall quality of software.

#### References

1. Cvjetkovic M., Vasiljević M., Cvjetković M., Josimović M. Impact of quality on improvement of business performance and customer satisfaction. *Journal of Engineering Management and Competitiveness*. 2021. Vol. 11. P. 20–28. DOI: 10.5937/jemc2101020C.
2. Krasner H. *The Cost of Poor Software Quality in the US: A 2020 Report*. Consortium for Information and Software Quality. 2021. URL: <https://www.it-cisq.org/the-cost-of-poor-software-quality-in-the-us-a-2020-report> (date of access: 20.10.2023).
3. Kokol P. Software Quality: How Much Does It Matter? *Electronics*. 2022, no. 11(16), 2485. DOI: 10.3390/electronics11162485.
4. *ISO/IEC/IEEE 12207:2017. Systems and Software Engineering - Software Life Cycle Processes*. ISO/IEC/IEEE International Standard. 2017. 145 p.
5. April A., Laporte C. Y. An Overview of Software Quality Concepts and Management Issues. *Measuring Information Systems Delivery Quality*. 2006. P. 28-54. DOI: 10.4018/978-1-59140-857-4.ch002 (date of access: 25.10.2023).
6. Nyári N., Kerti A. Review of Software Quality Related ISO Standards. *Safety and Security Sciences Review*. 2021. Vol. 3, № 2. P. 61–72.
7. Alebeisat F., Alhalhouli Z., Alshabat T., Alrawashdeh T. I. Review of Literature on Software Quality. *World of Computer Science and Information Technology Journal (WCSIT)*. 2018. Vol. 8, № 5. P. 32–42.
8. Ndukwe I. G., Licorish S. A., Tahir A., MacDonell S. G. How have views on Software Quality differed over time? Research and practice viewpoints. *Journal of Systems and Software*. 2023. Vol. 195. DOI: 10.1016/j.jss.2022.111524.

9. Karl Wiegers. *Software Development Pearls. Lessons from Fifty Years of Software Experience*. Addison-Wesley Professional, 2021. 330 p.
10. *Quality Glossary*. American Society for Quality. URL: <https://asq.org/quality-resources/quality-glossary> (date of access: 25.10.2023).
11. Joseph M. Juran; Joseph A. De Feo. *Juran's Quality Handbook: The Complete Guide to Performance Excellence, Sixth Edition*. McGraw-Hill, 2010. 1113 p.
12. Hussain S., Farid S., Mumtaz I. Is Customer Satisfaction Enough for Software Quality? *International Journal of Computer Science and Software Engineering (IJCSSE)*. 2019. Vol. 8, № 2. P. 40–47.
13. Petrasch R. The definition of software quality: a practical approach. *Proceedings of the 10th International Symposium on Software Reliability Engineering*. 1999. P. 33–34.
14. Tian J. *Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement*. IEEE. 2005, 412 p. URL: <https://onlinelibrary.wiley.com/doi/book/10.1002/0471722324> (date of access: 28.10.2023).
15. Nuseibeh B., Easterbrook S. M. *Fundamentals of Requirements Engineering*. Pearson Education Limited, 2007. 496 p.
16. Hossain A., Dr. Md. Kashem A., Sultana S. Enhancing Software Quality Using Agile Techniques. *IOSR Journal of Computer Engineering*. 2013. Vol. 10, № 2. P. 87–93.
17. Suman, Wadhwa M. A comparative study of software quality models. *International Journal of Computer Science and Information Technologies*. 2014. Vol. 5, № 4. P. 5634–5638.
18. ANSI/ASQC A3. *Quality systems terminology*. American National Standards Institute. Milwaukee, Wisconsin. 1978, 17 p.
19. *IEEE Standard Glossary of Software Engineering Terminology. ANSI/IEEE Std. 729-1983*. 1983, 38 p. DOI: 10.1109/IEEESTD.1983.7435207. URL: <https://ieeexplore.ieee.org/servlet/opac?punumber=7435205> (date of access: 28.10.2023).
20. *IEEE Standard Glossary of Software Engineering Terminology. IEEE Std 610.12-1990*. 1990, 83 p. DOI: 10.1109/IEEESTD.1990.101064. URL: <https://ieeexplore.ieee.org/servlet/opac?punumber=2238> (date of access: 28.10.2023).
21. ISO/IEC 25010:2011. *Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models*. 2011. URL: <https://www.iso.org/standard/35733.html> (date of access: 01.11.2023).
22. ISO 9000:2015. *Quality management systems – Fundamentals and vocabulary*. 2015. URL: <https://www.iso.org/standard/45481.html> (date of access: 01.11.2023).
23. ISO/IEC 5055:2021. *Information technology – Software measurement – Software quality measurement – Automated source code quality measures*. URL: <https://www.iso.org/standard/80623.html> (date of access: 01.11.2023).
24. Gobov D., Yanchuk, V. Network Analysis Application to Analyze the Activities and Artifacts in the Core Business Analysis Cycle. *Proceedings of the 2nd International Informatics and Software Engineering Conference (IISEC)*. 2021. P. 1-6. DOI: 10.1109/IISEC54230.2021.9672373.
25. *A Guide to the Business Analysis Body of Knowledge*. International Institute of Business Analysis, 2015. 512 p.
26. *The PMI guide to business analysis*. Project Management Institute, 2017. 444 p.
27. Norbert, N., András, K. Review of software quality related ISO standards. *Safety and Security Sciences Review*. 2021. Vol. 3, №2. P. 61–72.
28. Gobov D., Titlova O. Towards identifying challenges in business analysis on IT projects—a practical study. *Radioelectronic and Computer Systems*. 2023. Vol. 2. P. 193-206. DOI: 10.32620/reks.2023.2.16
29. Gobov D., Zuieva O. Identifying the dependencies between IT project context and business analysis document content. *Innovative technologies and scientific solutions for industries*. 2023. Vol. 2, №24. P. 39–53. DOI: 10.30837/itssi.2023.24.039.
- Competitiveness. 2021, no. 11. pp. 20–28. DOI: 10.5937/jemc2101020C.
2. Krasner H. *The Cost of Poor Software Quality in the US: A 2020 Report*. Consortium for Information and Software Quality. 2021. Available at: <https://www.it-cisq.org/the-cost-of-poor-software-quality-in-the-us-a-2020-report> (accessed: 20.10.2023).
3. Kokol P. Software Quality: How Much Does It Matter? *Electronics*. 2022, no. 11(16), 2485. DOI: 10.3390/electronics11162485.
4. ISO/IEC/IEEE 12207:2017. *Systems and Software Engineering - Software Life Cycle Processes*. ISO/IEC/IEEE International Standard. 2017, 145 p.
5. April A., Laporte C. Y. An Overview of Software Quality Concepts and Management Issues. *Measuring Information Systems Delivery Quality*. 2006, pp. 28-54. Available at: <https://doi.org/10.4018/978-1-59140-857-4.ch002> (accessed: 25.10.2023).
6. Nyári N., Kerti A. Review of Software Quality Related ISO Standards. *Safety and Security Sciences Review*. 2021, vol. 3, no. 2, pp. 61–72.
7. Alebeisat F., Alhalhouli Z., Alshabat T., Alrawashdeh T. I. Review of Literature on Software Quality. *World of Computer Science and Information Technology Journal (WCSIT)*. 2018, vol. 8, no. 5, pp. 32–42.
8. Ndukwe I. G., Licorish S. A., Tahir A., MacDonell S. G. How have views on Software Quality differed over time? Research and practice viewpoints. *Journal of Systems and Software*. 2023, vol. 195. DOI: 10.1016/j.jss.2022.111524.
9. Karl Wiegers. *Software Development Pearls. Lessons from Fifty Years of Software Experience*. Addison-Wesley Professional, 2021. 330 p.
10. *Quality Glossary*. American Society for Quality. Available at: <https://asq.org/quality-resources/quality-glossary> (accessed: 25.10.2023).
11. Joseph M. Juran; Joseph A. De Feo. *Juran's Quality Handbook: The Complete Guide to Performance Excellence, Sixth Edition*. McGraw-Hill, 2010. 1113 p.
12. Hussain S., Farid S., Mumtaz I. Is Customer Satisfaction Enough for Software Quality? *International Journal of Computer Science and Software Engineering (IJCSSE)*. 2019, vol. 8, no. 2, pp. 40–47.
13. Petrasch R. The definition of software quality: a practical approach. *Proceedings of the 10th International Symposium on Software Reliability Engineering*. 1999, pp. 33–34.
14. Tian J. *Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement*. IEEE. 2005, 412 p. Available at: <https://onlinelibrary.wiley.com/doi/book/10.1002/0471722324> (accessed: 28.10.2023).
15. Nuseibeh B., Easterbrook S. M. *Fundamentals of Requirements Engineering*. Pearson Education Limited, 2007. 496 p.
16. Hossain A., Dr. Md. Kashem A., Sultana S. Enhancing Software Quality Using Agile Techniques. *IOSR Journal of Computer Engineering*. 2013, vol. 10, no. 2, pp. 87–93.
17. Suman, Wadhwa M. A comparative study of software quality models. *International Journal of Computer Science and Information Technologies*. 2014, vol. 5(4), pp. 5634–5638.
18. ANSI/ASQC A3. *Quality systems terminology*. American National Standards Institute. Milwaukee, Wisconsin. 1978, 17 p.
19. *IEEE Standard Glossary of Software Engineering Terminology. ANSI/IEEE Std. 729-1983*. 1983, 38 p. DOI: 10.1109/IEEESTD.1983.7435207. Available at: <https://ieeexplore.ieee.org/servlet/opac?punumber=7435205> (accessed: 28.10.2023).
20. *IEEE Standard Glossary of Software Engineering Terminology. IEEE Std 610.12-1990*. 1990, 83 p. DOI: 10.1109/IEEESTD.1990.101064. Available at: <https://ieeexplore.ieee.org/servlet/opac?punumber=2238> (accessed: 28.10.2023).
21. ISO/IEC 25010:2011. *Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models*. 2011. Available at: <https://www.iso.org/standard/35733.html> (accessed: 01.11.2023).
22. ISO 9000:2015. *Quality management systems – Fundamentals and vocabulary*. 2015. Available at: <https://www.iso.org/standard/45481.html> (accessed: 01.11.2023).
23. ISO/IEC 5055:2021. *Information technology – Software measurement – Software quality measurement – Automated source code quality measures*. Available at: <https://www.iso.org/standard/80623.html> (accessed: 01.11.2023).

#### References (transliterated)

1. Cvjetkovic M., Vasiljević M., Cvjetković M., Josimović M. Impact of quality on improvement of business performance and customer satisfaction. *Journal of Engineering Management and*

24. Gobov D., Yanchuk, V. Network Analysis Application to Analyze the Activities and Artifacts in the Core Business Analysis Cycle. *Proceedings of the 2nd International Informatics and Software Engineering Conference (IISEC)*. 2021, pp. 1-6. DOI: 10.1109/IISEC54230.2021.9672373.
25. A Guide to the Business Analysis Body of Knowledge. International Institute of Business Analysis, 2015. 512 p.
26. *The PMI guide to business analysis*. Project Management Institute, 2017. 444 p.
27. Norbert, N., András, K. Review of software quality related ISO standards. *Safety and Security Sciences Review*. 2021, no. 3(2), pp. 61–72.
28. Gobov D., Titlova O. Towards identifying challenges in business analysis on IT projects—a practical study. *Radioelectronic and Computer Systems*. 2023, no.2, pp. 193-206. DOI: 10.32620/reks.2023.2.16
29. Gobov D., Zuieva O. Identifying the dependencies between IT project context and business analysis document content. *Innovative technologies and scientific solutions for industries*. 2023, no. 2(24), pp. 39–53. DOI: 10.30837/itssi.2023.24.039.

Received 13.11.2023

УДК 004.41

**Д. А. ГОБОВ**, кандидат технічних наук (Ph.D), Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», доцент кафедри інформатики та програмної інженерії факультету інформатики та обчислювальної техніки,

м. Київ, Україна; e-mail: d.gobov@kpi.ua; ORCID: <https://orcid.org/0000-0001-9964-0339>

**О. В. ЗУЄВА**, кандидат економічних наук (Ph.D), аналітик-консультант CS Ltd,

м. Харків, Україна, e-mail: olekszueva@gmail.com, ORCID: <http://orcid.org/0000-0001-9661-9657>

## АНАЛІЗ КОНЦЕПТУ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ: ПЕРСПЕКТИВА БІЗНЕС-АНАЛІЗУ

Якість програмного забезпечення є критичним аспектом розробки програмного забезпечення, який суттєво впливає на продуктивність бізнесу та задоволення клієнтів. Однак визначення якості програмного забезпечення може бути складним завданням, оскільки різні джерела надають різні визначення та погляди. У статті представлено огляд літератури з якості програмного забезпечення, підтверджуючи тривалу дискусію протягом років щодо визначення якості програмного забезпечення та методів його оцінки. Серед різних ідей про якість програмного забезпечення в статті виокремлено ключові концепції, які є важливими для розуміння поняття якості програмного забезпечення: відповідність вимогам, задоволення користувачів, функції програмного забезпечення та виявлення дефектів. В статті розглядаються міжнародні стандарти, такі як ISO/IEC 25010:2011 та ISO/IEC 5055:2021, в яких представлено терміни "Якість під час використання" та "Структурна якість". Розкриваючи трійковий погляд, який пояснюється в міжнародних стандартах - внутрішня якість, зовнішня якість та якість використання – в статті підкреслено тонку взаємодію між суб'єктивністю та об'єктивністю. Суб'єктивний вимір, визначений сприйняттям користувачів та факторами контексту, порівнюється з більш об'єктивними критеріями, такими як відповідність вимогам та відсутність дефектів. Стандарти надають слушний погляд, але людський аспект, такий як почуття користувача та конкретні фактори контексту, роблять надання універсального визначення поняття якості програмного забезпечення складним завданням.

Наголошується на надважливій ролі бізнес-аналізу та інженерії вимог у забезпеченні якості програмного забезпечення. Бізнес-вимоги, потреби зацікавлених сторін та якість функціональних та нефункціональних вимог виокремлюються як невід'ємні компоненти. В статті аргументовано, що якість програмного забезпечення тісно пов'язана з якістю його вимог, представляючи подвійну перспективу: відповідність критеріям якості та відповідність очікуванням зацікавлених сторін і цілям бізнесу. Отже, ефективна оцінка якості програмного забезпечення ґрунтується на фундаментальному розумінні нюансів контексту, потреб користувачів та умов експлуатації, все це визначається в процесі бізнес-аналізу.

**Ключові слова:** якість програмного забезпечення, бізнес-аналіз, вимоги, якість під час використання, внутрішня якість, міжнародні стандарти

*Повні імена авторів / Author's full names*

**Автор 1 / Author 1:** Гобов Денис Андрійович, Gobov Denys Andriyovych

**Автор 2 / Author 2:** Зуєва Олександра Валерівна, Zuieva Oleksandra Valerivna