BULLETIN of Cherkasy State Technological University

Journal homepage: https://bulletin-chstu.com.ua/en

Vol. 31 No. 3. 2026

UDC 004.056.5:004.42
DOI: 10.62660/bcstu/3.2026.105

A qualimetric taxonomy of quality indicators for evaluating protected data processing in socio-economic systems

Yuriy Paslavskyi*
Postgraduate Student
National Forestry University of Ukraine
79057, 103 General Chuprynka Str., Lviv, Ukraine
https://orcid.org/0009-0006-5678-7879
Ivan Hurtovyi
Postgraduate Student
National Aerospace University "Kharkiv Aviation Institute"
61070, 17 Vadym Manko Str., Kharkiv, Ukraine
https://orcid.org/0009-0000-3425-6272
Denis Tsitsiliano
Postgraduate Student
National Aerospace University "Kharkiv Aviation Institute"
61070, 17 Vadym Manko Str., Kharkiv, Ukraine
https://orcid.org/0009-0005-9524-9777
Roman Trishch
Doctor of Technical Sciences, Professor
National Aerospace University "Kharkiv Aviation Institute"
61070, 17 Vadym Manko Str., Kharkiv, Ukraine
https://orcid.org/0000-0002-9503-8428

Abstract. Decisions about protected data-processing software in banks, payment services, public registries, and social-service platforms require indicators whose meaning and role can be traced; a simple list of quality properties does not provide this basis. The study aimed to develop a qualimetric taxonomy and a machine-readable indicator contract that prepare heterogeneous evidence for scenario-specific quality assessment. The study used structured document analysis of software-quality standards, metrological guidance, qualimetric research, and recent protected-processing studies, followed by rule-based consolidation of candidate properties, two illustrative scenario walkthroughs, and fifteen focused software tests. The analysis distinguished five evaluation objects and produced seven indicator groups: functional suitability, resource efficiency, reliability, security and trust, compatibility, maintainability, and socio-organisational impact. Each indicator was represented by an eight-field passport recording its purpose, operational definition, unit, direction of improvement, source and acquisition method, uncertainty, decision role, and missing-evidence rule; the passport was separated from scenario-specific normalisation, thresholds, weights, policy source, and version. Three roles separated mandatory conditions, comparison objectives, and diagnostic information; Know Your Customer and Payment Verification scenarios showed that processing time can be a comparison objective in one service and a mandatory deadline in another. Fifteen verification cases covered schema validity, numerical values, normalisation, constraints, missing evidence, reproducibility, and transfer to the evaluator, while a completed processing-time passport demonstrated

Article's History: Received: 15.05.2026; Revised: 25.08.2026; Accepted: 21.09.2026; Published: 07.10.2026

Suggested Citation:
Paslavskyi, Yu., Hurtovyi, I., Tsitsiliano, D., & Trishch, R. (2026). A qualimetric taxonomy of quality indicators for evaluating protected data processing in socio-economic systems. *Bulletin of Cherkasy State Technological University*, 31(3), 105-112. doi: 10.62660/bcstu/3.2026.105.

*Corresponding author (yuriy.paslavsky@gmail.com)

Copyright © The Author(s). This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/)

how evidence semantics remain traceable before aggregation. The structure can be used by software developers, evaluators, procurement teams, and service owners to prepare traceable inputs for admissibility checks, composite evaluation, and multi-criteria selection without treating missing or incomparable evidence as favourable

Keywords: quality assessment; indicator passport; traceability; decision role; uncertainty; machine-readable data contract

INTRODUCTION

Protected digital services in socio-economic systems are evaluated through several properties at once. An identity-related service must return the correct result, limit disclosure, respond within an acceptable time, and leave evidence for review. A payment service faces similar properties, but their decision roles can differ. Without an explicit definition of the evaluation object, acquisition procedure, unit, and missing-evidence rule, a precise numerical value can still support the wrong conclusion.

J. Ernstberger *et al.* (2024) developed zk-Bench to compare proving frameworks under controlled workloads, whereas M. El-Hajj & B. Oude Roelink (2024) demonstrated that practical efficiency differs across proof families and implementations. These studies provide resource evidence, but they do not determine whether a value is a mandatory condition, a comparison objective, or diagnostic information in a specific service. W. Law (2025) proposed zkSDK to select a zero-knowledge backend from execution traces. J. Liang *et al.* (2025) separately showed that research results remain difficult to transfer into practice when implementations and workloads are reported under different conditions. Together, these studies indicate that automated selection requires not only values but also a record of their meaning, provenance, and decision role.

S. Chaliasos *et al.* (2024) showed that implementation vulnerabilities and security assumptions cannot be inferred from performance alone. D. Bui *et al.* (2025) and Z. Ye *et al.* (2025) demonstrated that circuit transformation and resource-allocation mechanisms change the context in which computational evidence is produced. Their results support keeping security, resource, and execution-context indicators distinct before any aggregate assessment is calculated. O. Cherniak *et al.* (2022) used qualimetric dependencies to evaluate labour-safety indicators, while Y. Rudyk *et al.* (2024) applied information technologies and safety indexes to critical-infrastructure quality assessment. These works confirm that heterogeneous evidence can support socio-economic decisions, but they do not provide a machine-readable contract that separates the meaning of protected-processing evidence from the rules of a particular decision scenario.

An organisational quality perspective extends the assessment beyond an individual software component. In socio-economic information systems, a technical choice affects service continuity, accountability, accessibility, cost, and organisational oversight. The unresolved problem is therefore a traceable bridge from a stakeholder requirement to an operational indicator, supporting evidence, uncertainty statement, and scenario-specific role. The object of the study was the process of evaluating the quality of protected data processing in socio-economic information systems. The subject was the structure of quality-indicator records, their classification, and the rules that connect evidence with admissibility and preference. Accordingly, the study aimed to develop a taxonomy of quality indicators for protected data processing and to verify that its records could be processed consistently in scenario-specific qualimetric assessment. Three tasks were set: to formalise the sources and transformation rules used to construct the taxonomy; to separate the indicator passport from scenario-specific rules and define their decision roles; and to verify the machine-readable contract through predefined software cases and two illustrative scenarios.

MATERIALS AND METHODS

The study combined structured document analysis, rule-based taxonomy construction, scenario walkthroughs, and software verification. The source corpus comprised the standards, metrological guides, qualimetric works, and protected-processing studies cited in the article. Standards and foundational guides were retained for definitions; recent research was used to represent current implementation, security, and resource contexts. The unit of analysis was a statement that connected a quality property with at least one of four elements: an evaluation object, an acquisition procedure, an uncertainty description, or a decision use. Candidate statements were coded by source type and compared by object, meaning, unit, acquisition conditions, and decision role. The structured analysis was therefore used as a separate method: it produced an auditable mapping from source statements to candidate indicators rather than serving only as a narrative literature review.

Candidate properties were transformed into the taxonomy through six rules:

1. A property was retained only when its assessment object and purpose could be identified; a broad label such as performance required an observable quantity, workload, and unit.

2. Every candidate received one primary object: input data, processing activity, software implementation, cryptographic mechanism, or decision output.

3. Synonyms were consolidated only when their objects and acquisition procedures matched;

identical labels were separated when they referred to different objects.

4. Candidates were mapped to one primary quality group according to the question answered by the evidence; cross-cutting relationships remained explanatory notes rather than duplicate indicators.

5. Each indicator received a decision role and a rule for missing or incomparable evidence.

6. The resulting structure was checked in Know Your Customer and Payment Verification scenarios to determine whether one passport could support different roles without transferring a scenario-specific threshold or weight.

The two scenarios were selected because both process protected information but impose different operational priorities. Customer-identity verification can use processing time as a comparison objective after correctness and disclosure requirements have been met. Payment verification can treat the same clearly defined time quantity as a mandatory deadline. This contrast tests the central design requirement: evidence semantics must remain stable when its decision role changes. Software verification was organised as fifteen focused cases, with one case assigned to each high-risk rule or boundary condition in the machine-readable contract. The cases covered five areas: schema and numerical validation; admissibility and missing evidence; normalisation and uncertainty handling; ranking, sensitivity, and reproducibility; and regression of the two stored synthetic scenario reports. The implementation used Python's unit test framework, JSON Schema Draft 2020-12, and the pinned jsonschema 4.26.0 validator. A case passed only when the actual status, rejection message, numerical boundary, or serialised report exactly matched the predefined expectation. Repeated evaluations also had to produce identical output, and the regenerated scenario reports had to match their stored versions. This procedure checked software semantics and traceability; it did not evaluate a real institution or infer domain thresholds and weights.

RESULTS AND DISCUSSION

The structured analysis produced three successive outputs. First, it separated five evaluation objects that had previously been hidden inside broad quality labels: input data, processing activity, software implementation, cryptographic mechanism, and decision output. Second, the six transformation rules consolidated the candidate properties into seven top-level groups. Third, every retained indicator received the eight-field passport presented in Table 1.

Table 1. Core fields of an indicator passport

Passport field	Question answered	Example of the required clarification
Evaluation object and purpose	What is being evaluated, and why does it matter?	Complete request, processing stage, software component, or decision output
Operational definition	What exactly is observed or calculated?	Start and end events, workload, population, or documentary criterion
Unit	How is the result expressed?	Seconds, bytes, proportion, count, or an explicitly defined ordinal scale
Direction of improvement	Does a larger or smaller value represent better quality?	Lower processing time; higher successful-completion rate
Source and acquisition method	Where does the value come from?	Experiment, prediction, document, or structured expert judgement
Uncertainty	How precise and stable is the value?	Standard uncertainty, interval, model assumptions, or stated absence of a probabilistic estimate
Decision role	How is the indicator used in this scenario?	Critical condition, comparison objective, or diagnostic information
Missing-evidence handling	What happens when the value is absent or incomparable?	Stop the relevant comparison; do not replace the value with zero

Source: developed by the authors

The rules resolved three recurrent ambiguity patterns without requiring an aggregate score: the same label applied to different objects, different labels applied to the same object, and an indicator used without a declared decision role. As a result, complete request time remained distinct from the time of one cryptographic stage, while synonymous indicators were combined only when their objects and acquisition procedures matched. Table 1 summarises the resulting passport. Its fields make the meaning, source, and treatment of each value explicit before normalisation or weighting. The analysis also showed that evidence types require different uncertainty statements. Repeated observations may support a standard uncertainty or interval; predictions retain model assumptions and prediction error; documentary evidence identifies its authoritative source; and structured expert judgement records its scale and elicitation procedure. The passport preserves these distinctions instead of forcing all evidence into one generic field. The seven groups obtained from the structured analysis are shown in Table 2. Five groups retain central

software-quality questions, security is extended to security and trust, and socio-organisational impact represents explainability, accessibility, discrimination risk, accountability, and oversight.

Table 2. Top-level indicator groups

Indicator group	What it describes	Typical examples	Possible decision role
Functional suitability	Whether the processing performs the required function correctly and completely	Correctness, rule coverage, successful verification	Critical condition or comparison objective
Resource efficiency	Resources required to deliver the result	Time, memory, throughput, energy	Critical condition, comparison objective, or diagnostic information
Reliability	Ability to deliver stable service and recover from faults	Stability, fault tolerance, recoverability	Critical condition or comparison objective
Security and trust	Protection properties and assumptions on which they depend	Confidentiality, integrity, trust assumptions	Usually a critical condition
Compatibility	Ability to exchange data and operate with surrounding components	Interoperability, data portability	Critical condition or comparison objective
Maintainability	Effort required to analyse, test, modify, and support the component	Testability, analysability, modifiability	Comparison objective or diagnostic information
Socio-organisational impact	Effects on users, staff, governance, and oversight	Explainability, accessibility, discrimination risk, accountability	Critical condition or comparison objective

Source: developed by the authors

The resulting data contract contained two linked layers. The first layer preserved the evidence passport. The second layer stored the scenario identifier and version, normalisation parameters, admissibility threshold, comparison weight, and responsible policy source. This separation retained one compatible evidence record when only a policy threshold changed, while a different workload or observation boundary required a revised passport. Three decision roles were obtained from this separation. A critical condition determined admission; a comparison objective distinguished admissible alternatives; and diagnostic information explained behaviour without affecting order. Missing evidence was handled according to the declared role: an unresolved critical condition prevented admission, an unavailable comparison objective stopped the affected comparison, and optional diagnostic information remained visibly unavailable. The obtained outcomes corresponded to the expected results across the tested cases. A summary of the verification results is presented in Table 3.

Table 3. Software verification results

Verification area	Cases	Conditions covered	Observed result
Schema and numerical validation	5	Weight sum, unknown fields, type-safe equality thresholds, finite numbers, and pinned validator dependency	All five invalid or boundary inputs produced the predefined validation response
Admissibility and missing evidence	3	Non-compensatory mandatory failure, uncertainty-aware constraint, and missing value/provenance	All three cases preserved pass, violation, inconclusive, and missing-evidence semantics
Normalisation and uncertainty	2	Four normalisation forms and seedednon-zero uncertainty	Normalized values remained bounded and repeated uncertainty output was identical
Ranking and reproducibility	4	Weight-induced reversal, tie reporting, identifier-stable sub streams, and byte-stable serialisation	All four cases matched the predefined rankings, ties, and repeatability checks
Scenario regression	1	Regeneration of the stored KYC and Payment Verification reports	Both regenerated reports matched their stored versions; 15 of 15 cases passed in total

Note: KYC – Know Your Customer
Source: developed by the authors

The verification results confirmed the declared software behaviour in all fifteen predefined cases. Invalid weights, unknown keys, incompatible threshold types, and non-finite values were rejected. Mandatory violations could not be compensated by favourable secondary indicators. Inconclusive uncertainty intervals remained distinct from demonstrated violations, ties remained distinct from reversals, and repeated evaluations preserved byte-identical output. The practical effect of the contract was examined in two illustrative protected-processing scenarios. Table 4 reports the scenario-dependent roles rather than performance values of real institutions.

Table 4. Scenario-dependent roles of selected indicators

Indicator or rule	KYC	Payment verification	Practical meaning
Correctness of the result	Critical condition	Critical condition	An incorrect alternative is not admitted to comparison
Processing time	Comparison objective	Critical deadline	The same experimentally obtained quantity may express a preference or a mandatory requirement
Memory use	Comparison objective	Comparison objective	Lower resource use improves the comparison after admission
Auditability	Diagnostic information	Comparison objective	Its influence depends on the purpose of the service
Missing mandatory evidence	Stops admission	Stops admission	Absence is reported and is not replaced by a favourable value

Source: developed by the authors

The scenarios produced the expected role change without changing the meaning of the underlying evidence. Processing time remained a comparison objective for KYC after mandatory requirements had been met, but became a critical deadline for Payment Verification. Correctness and missing mandatory evidence retained their non-compensatory role in both scenarios, while auditability changed from diagnostic information to a comparison objective. Table 5 gives a completed illustrative passport for the KYC processing-time indicator. It describes a reusable evidence record without inventing a measured value or a real institutional threshold.

Table 5. Completed illustrative passport for KYC processing time

Passport field	Completed illustrative record for processing time
Evaluation object and purpose	Complete KYC request; compare the responsiveness of otherwise admissible software alternatives
Operational definition	Elapsed interval from acceptance of a declared request to return of the decision under the same workload and start/end events
Unit	Milliseconds
Direction of improvement	Lower is better
Source and acquisition method	Repeated clock observations in a declared test run; the run and workload identifier accompany the result
Uncertainty	Report the available standard uncertainty or interval; explicitly state when no probabilistic estimate is available
Decision role	Comparison objective in the KYC scenario; the role is stored in the scenario-rules record
Missing-evidence handling	Stop the affected comparison and report insufficient evidence; do not substitute zero

Source: developed by the authors

The obtained results can first be considered in relation to the existing software-quality standards**.** ISO/IEC 25010:2023 (2023) defines a product-quality model, and ISO/IEC 25023:2016 (2016) provides measures for evaluating its characteristics. E.800 (2008) supplies a service-quality vocabulary, while ISO 9000:2026 (2026) connects quality with the fulfilment of requirements. In comparison with these approaches, the proposed taxonomy provides an additional level of specification: these documents define categories and terms but do not prescribe one record that binds an indicator to an evaluation object, evidence source, uncertainty statement, and scenario rule. The result also extends the metrological discipline represented by VIM and GUM (JCGM 100:2008(E), 2008; JCGM 200:2012, 2012). Those guides require clear quantities, procedures, and uncertainty information. The proposed contract carries that discipline into a mixed-evidence software decision and adds the scenario layer needed to distinguish an empirically obtained property from a policy threshold or weight. A. Motalo *et al.* (2017) applied the same discipline to qualimetry, which is particularly relevant when experimental, predicted, documentary, and expert-derived evidence is used in one assessment.

Compared with the general qualimetric framework of OECD/European Union/EC-JRC (2008), the present method acts one stage earlier. The OECD/JRC handbook (OECD/European Union/EC-JRC, 2008) required indicator selection, data treatment, normalisation, weighting, and aggregation to follow an explicit theoretical framework. Their approaches explain normalisation, weighting, and aggregation; the passport checks whether the inputs have compatible meanings before those operations. This reduces the risk that a mathematically correct aggregate is constructed from different workloads, observation boundaries, or evidence types.

Benchmark-oriented tools provide another complementary layer. G. Kadianakis & M. Orrù (2023) estimate cryptographic-operation costs, while A.R. Block *et al.* (2021), J. Bootle *et al.* (2022), and A. Golovnev *et al.* (2023) expose different time-space relationships. B. Chen *et al.* (2023) similarly changed the relationship

between circuit size and prover cost through a linear-time design. T. Liu *et al.* (2024) distributed proof generation, S. Fu & G. Gong (2022) reduced verifier costs, H. Masip-Ardevol *et al.* (2024) extended transparent arguments, and A. Zapico *et al.* (2022) optimised lookup processing. J. Groth (2016) established protocol-level succinctness properties, whereas O. Kuznetsov *et al.* (2026) supplied implementation-oriented performance evidence. These studies address different layers and further demonstrate why the evaluated stage, workload, and resource boundary need to be explicitly identified. B. Steiner *et al.* (2023), although working in a different application domain, similarly demonstrated the importance of an operationally defined resource boundary when assessing memory use. The taxonomy does not replace these values. It records what each value describes and prevents complete service time, one internal stage, and a protocol-level property from being treated as interchangeable.

The five-object structure and seven groups explain why the result is more than a checklist. A checklist can state that time, security, and explainability matter, but it does not show whether time refers to a request or an operation, whether security is a mandatory condition, or whether unavailable evidence stops the decision. The two-layer contract makes these distinctions executable and traceable. The scientific contribution follows from this comparison. Existing quality models define characteristics, metrological guidance defines evidence discipline, benchmark studies supply technical observations, and qualimetric methods support aggregation. The improved applied method links these layers through one eight-field passport, a separate scenario-rules record, and three decision roles before aggregation. The novelty is therefore the operational connection between stakeholder requirement, indicator meaning, evidence provenance, uncertainty statement, and scenario rule, not a new aggregate-score formula. The scope remains formal and software-oriented. The fifteen tests establish internal consistency of the machine-readable contract and the two scenarios demonstrate its use, but they do not rank real institutions or approve domain thresholds and weights. This boundary preserves the practical value of the result without presenting illustrative data as external validation.

CONCLUSIONS

The study distinguished five evaluation objects and consolidated quality properties into seven indicator groups for protected data processing in socio-economic information systems. The resulting eight-field passport specifies the evaluation object and purpose, operational definition, unit, direction of improvement, source and acquisition method, uncertainty, decision role, and missing-evidence handling. A two-layer machine-readable contract separated the evidence passport from scenario-specific normalisation parameters, admissibility thresholds, weights, policy sources, and versions. Three decision roles separated critical conditions, comparison objectives, and diagnostic information, preventing favourable secondary properties from compensating for an unmet mandatory requirement. All fifteen predefined software cases produced the expected validation, admissibility, uncertainty, ranking, and reproducibility outcomes. The KYC and Payment Verification walkthroughs further showed that one processing-time passport could support different scenario roles without transferring a threshold or weight between services.

The practical result is a traceable input structure for admissibility checks, composite qualimetric evaluation, and multi-criteria software selection. It keeps missing and incomparable evidence visible and shows whether a decision changed because the evidence changed or because the scenario policy was revised. Further research will extend the contract to additional protected-processing scenarios and evidence types and will examine automated compatibility checks between indicator passports and versioned scenario policies.

ACKNOWLEDGEMENTS

None.

FUNDING

None.

CONFLICT OF INTEREST

None

REFERENCES

[1] Block, A.R., Holmgren, J., Rosen, A., Rothblum, R.D., & Soni, P. (2021). Time- and space-efficient arguments from groups of unknown order. In T. Malkin & C. Peikert (Eds.), *Advances in cryptology – CRYPTO 2021* (Vol. 12828; pp. 123-152). Cham: Springer. doi: 10.1007/978-3-030-84259-8_5.

[2] Bootle, J., Chiesa, A., Hu, Y., & Orrú, M. (2022). Gemini: Elastic SNARKs for diverse environments. In O. Dunkelman & S. Dziembowski (Eds.), *Advances in cryptology – EUROCRYPT 2022* (Vol. 13276; pp. 427-457). Cham: Springer. doi: 10.1007/978-3-031-07085-3_15.

[3] Bui, D., Chu, H., Couteau, G., Wang, X., Weng, C., Yang, K., & Yu, Y. (2025). An efficient ZK compiler from SIMD circuits to general circuits. *Journal of Cryptology*, 38, article number 10. doi: 10.1007/s00145-024-09531-4.

[4] Chaliasos, S., Ernstberger, J., Theodore, D., Wong, D., Jahanara, M., & Livshits, B. (2024). SoK: What don't we know? Understanding security vulnerabilities in SNARKs. In *Proceeding of 33rd USENIX security symposium* (pp. 3855-3872). Philadelphia: USENIX.

[5] Chen, B., Bünz, B., Boneh, D., & Zhang, Z. (2023). HyperPlonk: Plonk with linear-time prover and high-degree custom gates. In C. Hazay & M. Stam (Eds.), *Advances in cryptology – EUROCRYPT 2023* (Vol. 14005; pp. 499-530). Cham: Springer. doi: 10.1007/978-3-031-30617-4_17.
[6] Cherniak, O., Sorocolat, N., Bahaiev, I., & Fatieieva, L. (2022). Application of functional dependence for multi-criteria assessment of labor safety as an object of qualimetry. *Innovative Technologies and Scientific Solutions for Industries*, 1, 76-84. doi: 10.30837/ITSSI.2022.19.076.
[7] E.800. (2008). *Definitions of terms related to quality of service*. Retrieved from https://www.itu.int/rec/T-REC-E.800-200809-I/en.
[8] El-Hajj, M., & Oude Roelink, B. (2024). Evaluating the efficiency of zk-SNARK, zk-STARK, and Bulletproof in real-world scenarios: A benchmark study. *Information*, 15(8), article number 463. doi: 10.3390/info15080463.
[9] Ernstberger, J., Chaliasos, S., Kadianakis, G., Steinhorst, S., Jovanovic, P., Gervais, A., Livshits, B., & Orrù, M. (2024). zk-Bench: A toolset for comparative evaluation and performance benchmarking of SNARKs. In C. Galdi & D.H. Phan (Eds.), *Security and cryptography for networks* (Vol. 14973; pp. 46-72). Cham: Springer. doi: 10.1007/978-3-031-71070-4_3.
[10] Fu, S., & Gong, G. (2022). Polaris: Transparent succinct zero-knowledge arguments for R1CS with efficient verifier. *Proceedings on Privacy Enhancing Technologies*, 2022(1), 544-564. doi: 10.2478/popets-2022-0027.
[11] Golovnev, A., Lee, J., Setty, S., Thaler, J., & Wahby, R.S. (2023). Brakedown: Linear-time and field-agnostic SNARKs for R1CS. In H. Handschuh & A. Lysyanskaya (Eds.), *Advances in cryptology – CRYPTO 2023* (Vol. 14082; pp. 193-226). Cham: Springer. doi: 10.1007/978-3-031-38545-2_7.
[12] Groth, J. (2016). On the size of pairing-based non-interactive arguments. In M. Fischlin & J.S. Coron (Eds.), *Advances in cryptology – EUROCRYPT 2016* (Vol. 9666; pp. 305-326). Berlin: Springer. doi: 10.1007/978-3-662-49896-5_11.
[13] ISO 9000:2026. (2026). *Quality management systems – fundamentals and vocabulary*. Retrieved from https://www.iso.org/standard/9000.
[14] ISO/IEC 25010:2023. (2023). *Systems and software engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) – product quality model*. Retrieved from https://www.iso.org/standard/78176.html.
[15] ISO/IEC 25023:2016. (2016). *Systems and software engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) – measurement of system and software product quality*. Retrieved from https://www.iso.org/standard/35747.html.
[16] JCGM 100:2008(E). (2008). *Evaluation of measurement data – guide to the expression of uncertainty in measurement*. doi: 10.59161/JCGM100-2008E.
[17] JCGM 200:2012. (2012). *International vocabulary of metrology – basic and general concepts and associated terms (VIM)* (3rd ed.). doi: 10.59161/JCGM200-2012.
[18] Kadianakis, G., & Orrù, M. (2023). zkalc: A cryptographic calculator. *Ethereum Foundation Cryptography Research*. Retrieved from https://crypto.ethereum.org/blog/zkalc.
[19] Kuznetsov, O., Khavikova, Y., Bushkov, V., Shchytov, D., & Mormul, N. (2026). Performance analysis of Groth16 zkSNARK: Systematic benchmarking with Circom-snarkjs. *International Journal of Computing*, 24(4), 645-660. doi: 10.47839/ijc.24.4.4329.
[20] Law, W. (2025). zkSDK: Streamlining zero-knowledge proof development through automated trace-driven ZK-backend selection. *arXiv*. doi: 10.48550/arXiv.2507.05294.
[21] Liang, J., Hu, D., Wu, P., Yang, Y., Shen, Q., & Wu, Z. (2025). SoK: Understanding zk-SNARKs: The gap between research and practice. In *Proceedings of the 34th USENIX security symposium* (pp. 2085-2104). Seattle: USENIX.
[22] Liu, T., Xie, T., Zhang, J., Song, D., & Zhang, Y. (2024). Pianist: Scalable zkRollups via fully distributed zero-knowledge proofs. In *2024 IEEE symposium on security and privacy* (pp. 1777-1793). San Francisco: IEEE. doi: 10.1109/SP54263.2024.00035.
[23] Masip-Ardevol, H., Baylina-Melé, J., Guzmán-Albiol, M., & Muñoz-Tapia, J.L. (2024). eSTARK: Extending STARKs with arguments. *Designs, Codes and Cryptography*, 92, 3677-3721. doi: 10.1007/s10623-024-01457-z.
[24] Motalo, A., Stadnyk, B., & Motalo, V. (2017). Analysis of methods and types of measurements in qualimetry. *Measuring Equipment and Metrology*, 78, 85-92. doi: 10.23939/istcmtm2017.78.085.
[25] OECD/European Union/EC-JRC. (2008). *Handbook on constructing composite indicators: Methodology and user guide*. Paris: OECD Publishing. doi: 10.1787/9789264043466-en.
[26] Rudyk, Y., Mykyichuk, M., & Menshykova, O. (2024). Criteria for the use of information technology in assessing the quality of critical infrastructure by safety indexes. In E. Faure, Y. Tryus, T. Vartiainen, O. Danchenko, M. Bondarenko, C. Bazilo & G. Zaspa (Eds.), *Information technology for education, science, and technics* (Vol. 221; pp. 166-179). Cham: Springer. doi: 10.1007/978-3-031-71801-4_13.
[27] Steiner, B., Elhoushi, M., Kahn, J., & Hegarty, J. (2023). MODeL: Memory optimizations for deep learning. In *ICML '23: Proceedings of the 40th international conference on machine learning* (pp. 32618-32632). Honolulu: JMLR.org.

[28] Ye, Z., Qi, X., Zhang, Z., & Jin, C. (2025). Yoimiya: A scalable framework for optimal resource utilization in ZK-SNARK systems. *arXiv*. doi: 10.48550/arXiv.2502.18288.
[29] Zapico, A., Buterin, V., Khovratovich, D., Maller, M., Nitulescu, A., & Simkin, M. (2022). Caulk: Lookup arguments in sublinear time. In *CCS '22: Proceedings of the 2022 ACM SIGSAC conference on computer and communications security* (pp. 3121-3134). New York: Association for Computing Machinery. doi: 10.1145/3548606.3560646.

Кваліметрична таксономія показників якості захищеного опрацювання даних у соціально-економічних системах

Юрій Паславський
Аспірант
Національний лісотехнічний університет України
79057, вул. Генерала Чупринки, 103, м. Львів, Україна
https://orcid.org/0009-0006-5678-7879
Іван Гуртовий
Аспірант
Національний аерокосмічний університет «Харківський авіаційний інститут»
61070, вул. Вадима Манька, 17, м. Харків, Україна
https://orcid.org/0009-0000-3425-6272
Денис Цициліано
Аспірант
Національний аерокосмічний університет «Харківський авіаційний інститут»
61070, вул. Вадима Манька, 17, м. Харків, Україна
https://orcid.org/0009-0005-9524-9777
Роман Тріщ
Доктор технічних наук, професор
Національний аерокосмічний університет «Харківський авіаційний інститут»
61070, вул. Вадима Манька, 17, м. Харків, Україна
https://orcid.org/0000-0002-9503-8428

Анотація. Рішення щодо програмних засобів захищеного опрацювання даних у банках, платіжних сервісах, державних реєстрах і платформах соціальних послуг потребують показників, значення та роль яких можна простежити; простий перелік властивостей якості не створює такої основи. Метою дослідження було розробити кваліметричну таксономію та машиночитний контракт показників, що готують різнорідні дані для сценарного оцінювання якості. Застосовано структурований аналіз документів зі стандартами якості програмного забезпечення, метрологічними настановами, кваліметричними дослідженнями й сучасними працями із захищеного опрацювання даних, після чого виконано формалізоване об'єднання властивостей, два ілюстративні сценарні проходи та п'ятнадцять цільових програмних тестів. Аналіз розмежував п'ять об'єктів оцінювання та сформував сім груп показників: функціональну придатність, ресурсну ефективність, надійність, безпеку і довіру, сумісність, супроводжуваність та соціально-організаційний вплив. Кожний показник подано восьмипольовим паспортом із метою, операційним визначенням, одиницею, напрямом поліпшення, джерелом і способом отримання, невизначеністю, роллю в рішенні та правилом відсутніх даних; паспорт відокремлено від сценарних параметрів нормування, порогів, ваг, джерела політики та версії. Три ролі розмежували обов'язкові умови, критерії порівняння і діагностичну інформацію; сценарії перевірки клієнта та платежу показали, що час опрацювання може бути критерієм порівняння в одному сервісі та обов'язковим строком в іншому. П'ятнадцять програмних випадків охопили коректність схеми, числових значень, нормування й обмежень, роботу з відсутніми даними, відтворюваність і передавання до засобу оцінювання, а заповнений паспорт часу показав, як зміст доказових даних зберігається до агрегування. Структуру можуть використовувати розробники, оцінювачі, закупівельні підрозділи та власники сервісів для підготовки простежуваних вхідних даних до перевірки допустимості, комплексного оцінювання й багатокритеріального вибору без подання відсутніх або непорівнюваних даних як сприятливих

Ключові слова: оцінювання якості; паспорт показника; простежуваність; роль у рішенні; невизначеність; машиночитний контракт даних