UDC 004.052:004.412 **DOI:** 10.31673/2412-9070.2026.051356

A. SHANTYR[1], PhD, Associate Professor;
ORCID: 0000-0002-0466-3659
O. ZINCHENKO[1], Doctor of Technical Sciences, Professor;
ORCID: 0000-0002-3973-7814
K. STORCHAK[2], Doctor of Technical Science, Professor;
ORCID: 0000-0001-9295-4685
A. BONDARCHUK[3,4], Doctor of Technical Science, Professor;
ORCID: 0000-0001-5124-5102
N. LASHCHEVSKA[1], PhD, Associate Professor,
ORCID: 0000-0003-2148-115X
[1]National University of Life and Environmental Sciences of Ukraine, Kyiv
[2]State University of Information and Communication Technologies, Kyiv
[3]Borys Grinchenko Kyiv Metropolitan University
[4]National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute»

DETERMINATION OF ADMISSIBLE UNCERTAINTY LEVELS FOR QUALITY ASSESSMENTS OF SOFTWARE SYSTEMS

In software quality monitoring, errors in measurement, as well as fluctuations in the execution environment, the software architecture, and the users' behavior, all cause major problems for quantitative assessment of software quality. To address the problems mentioned above, the paper proposes an integrated approach comprising the Bayesian mean correction approach, robust probabilistic models, autoregressive models, and anomaly detection models. The composite approach is adaptive and determines the so-called allowable uncertainty based on historical data and the current system status. The paper presents an example application of the Bayesian mean correction approach and an anomaly-filtering approach, tested on a real data set from an operational enterprise resource planning (ERP) system. By applying approaches for anomaly filtering and Bayesian correction, the relative uncertainty of the considered response-time metric decreased from 84.1% to 4.1% and has remained stable since then, averaging 290–330 ms. In addition, average user satisfaction has increased by 3.8 percentage points, to 87.7% from 83.9%.

Keywords: software systems, quality assessment, uncertainty, monitoring, optimization, modeling, Bayesian correction.

Introduction

There arises an acute problem of determining the allowable level of uncertainty values at analysis of software quality. Various approaches were proposed by the researchers, including statistical approach, fuzzy logic, Bayesian models and approaches concerning machine learning.

Confidence intervals, bootstrap methods, and variance estimation provide statistical uncertainty measures [1] – [3], [5], [6], while Bayesian models incorporate prior knowledge [8]. Their integration with robust modelling, temporal correction, and adaptive uncertainty thresholds remains the unresolved issue addressed in this paper. Additionally, the bootstrap approach facilitates the estimation of variances for performance metrics as input test parameters change [4].

The relevance of fuzzy logic is further highlighted in paper [7], especially in contexts where precise quantitative evaluation is challenging or unfeasible. Finally, paper [9] discusses ensemble machine learning methods, illustrating their capacity to reduce errors and enhance the reliability of quality evaluation results.

Moreover, there are several international standards such as ISO/IEC 25010 and IEEE 982.1 suggesting unified metrics and control thresholds [10], which considerably influence determination of the allowed uncertainty levels [10].

© Shantyr A., Zinchenko O., Storchak K., Bondarchuk A., Lashchevska N., 2026

In the paper [11], the analysis of integrated methods in the context of software development highlights the necessity to integrate different approaches to account for uncertainty. In the paper [12], new opportunities offered by integrated formal methods for the deep analysis of uncertainty in software systems are analyzed.

In the paper [13], it is highlighted that the use of artificial neural networks for quality forecasting prompts the problem of taking into account model errors.

The paper [14] confirms the ability of the Bayesian approach to achieve higher levels of accuracy regarding uncertainty estimate because it takes into account not only the current information but also prior information related to the system under consideration. In the paper [15], the principles of Bayesian network usage in estimating data quality are described, showing that the statistical approach allows predicting the likely boundaries of uncertainty. The paper [16] contains the description of the cross-validation technique used for evaluation of feature importance in view of their correlation. The paper [17] suggests that integration of various approaches and tools in the process of software development may facilitate the further adaptation of such approaches for evaluating acceptable levels of uncertainty.

In particular, the paper [18] confirms that ensemble methods, such as gradient boosting and random forest, reduce uncertainty because they use multiple models in making assessment.

In particular, in the paper [19] the approach to forecasting uncertainty levels based on multilayer models that involve quality metrics and historical data on the behavior of the software system is outlined. Moreover, modern scientific research pays great attention to the definition of uncertainty in software quality assessment [20].

Thus, based on the conducted literature review and the justification of the research relevance, *the object of the study* is the process of assessing the quality of software systems under conditions of uncertainty, considering the influence of dynamic system behaviour, measurement errors, and variability of performance indicators.

Therefore, the literature review reveals several unresolved issues in software quality assessment under uncertainty: the absence of adaptive mechanisms for determining admissible uncertainty thresholds under dynamic conditions, the insufficient robustness of classical methods to small samples and non-Gaussian distributions, and the lack of integrated approaches combining statistical, probabilistic, robust, and time-dependent modeling. These unresolved issues form the basis for the aim and tasks of the present study.

The aim of this research is to determine and substantiate admissible levels of uncertainty in software quality assessment under dynamic operating conditions, thereby improving the reliability and validity of the assessment results. To achieve this aim, the following objectives were formulated:

1. To formalize the indicators and criteria for determining admissible uncertainty levels in software quality assessment under dynamic operating conditions.

2. To substantiate the conceptual principles and structure of a hybrid method for determining admissible uncertainty levels by integrating Bayesian correction, robust probabilistic modeling, autoregressive adjustment, and anomaly filtering.

3. To experimentally evaluate the Bayesian-adjustment and anomaly-filtering components of the proposed method using real operational software-performance indicators and to analyze their influence on key quality characteristics.

4. To compare software quality indicators before and after applying the proposed method in terms of relative uncertainty, response-time stability, and user satisfaction.

5. To formulate recommendations for applying and further improving the proposed method.

Materials and Methods

The selected methods are justified by their complementary ability to address key sources of uncertainty: Bayesian techniques handle small sample bias, robust statistics mitigate outliers, and autoregressive models capture temporal dependencies.

When assessing the quality of software systems, set of metrics $Q = \{q_1, q_2, \ldots, q_n\}$ used, which are estimated to be a certain level of uncertainty. The results of the estimates may contain errors or uncertainties, so it is necessary to establish acceptable levels of this uncertainty in the way that the results

are statistically significant and correspond to the real characteristics of the system. According to [6], mathematically, the level of uncertainty can be expressed through the following parameters:

– expected value of quality metric:

$$E[q_i] = \int_{-\infty}^{\infty} q_i f(q_i)\, dq_i, \tag{1}$$

where $E[q_i]$ – the mathematical expectation of the value of the quality metric, $f(q_i)$ – Probability Density Function (PDF) of values q_i;

– estimation uncertainty variance:

$$\sigma_{q_i}^2 = E\left[\left(q_i - E[q_i]\right)^2\right] = \int_{-\infty}^{\infty} \left(q_i - E[q_i]\right)^2 f(q_i)\, dq_i, \tag{2}$$

where $\sigma_{q_i}^2$ – variance of the estimate q_i, which characterizes the level of its uncertainty.

Lets consider the limits of permissible levels of uncertainty. By defining the permissible level of uncertainty through the relative error of the estimate the threshold levels can be determined:

$$U(q_i) = \frac{\sigma_{q_i}}{E[q_i]}, \tag{3}$$

where $U(q_i)$ – relative uncertainty of the quality metric estimate. It is necessary that $U(q_i)$ did not exceed a certain limit value U_{max}:

$$U(q_i) \le U_{\max}. \tag{4}$$

The confidence interval of the estimate:

$$q_i \in \left[E[q_i] - k\sigma_{q_i}, E[q_i] + k\sigma_{q_i}\right], \tag{5}$$

where k – coefficient that determines the level of confidence (for example, for a 95% confidence level $k \approx 1.96$).

To be considered reliable boundary conditions for acceptable estimates must satisfy condition:

$$\frac{\sigma_{q_i}}{E[q_i]} \le \tau, \tag{6}$$

where τ – the maximum permissible uncertainty, determined from experimental or regulatory requirements. The permissible variance is defined as:

$$\sigma_{q_i} \le \tau E[q_i]. \tag{7}$$

Equations (3)–(7) constitute the formal system of indicators and admissibility criteria used to determine whether a software quality estimate can be regarded as sufficiently reliable.

Consider the problem of choosing an uncertainty threshold value τ. Tolerable level of uncertainty τ is chosen expertly, which can be subjective. If you choose too small τ, then many estimates will be rejected as incorrect. If a value is too large τ, then the system can make correct decisions [6]. For adaptive definition, we can use the expression:

$$\tau = \frac{1}{N} \sum_{i=1}^{N} \frac{\sigma_{q_i}}{E[q_i]}. \tag{8}$$

Or apply optimization methods:

$$\tau^* = \arg\min_{\tau} \sum_{i=1}^{N} L(q_i, \tau), \tag{9}$$

where $L(q_i, \tau)$ – loss function.

The classical average value is calculated according to:

$$\bar{q} = \frac{1}{N}\sum_{i=1}^{N} q_i, \tag{10}$$

where q_i – the value of the quality assessment for i-th measurement, N – number of available measurements. The classical dispersion is calculated according to:

$$\sigma_q^2 = \frac{1}{N}\sum_{i=1}^{N}\left(q_i - \bar{q}\right)^2, \tag{11}$$

where q_i – the value of the quality assessment for i-th measurement, N – number of available measurements.

Bayesian mean correction has the following problem: if N small, then medium $\bar{q}$ has a large variance. To resolve this instead of using the usual mean, we apply a priori bayes correction:

$$\mu_q = \frac{N\bar{q} + \lambda\mu_0}{N + \lambda}, \tag{12}$$

where μ_0 – apriori average based on historical data, λ – weighted confidence coefficient to the prior average ($\lambda \approx 10$ for the little ones N).

As a result, the variability of the average decreases at small N, avoiding displacement. Robust probabilistic modeling has the following problem: if the distribution of scores has heavy tails, classical variance gives false results. Solution: using a generalized t-distribution with a robustness parameter ν:

$$f\left(q_i\right) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)\sigma_q}\left(1 + \frac{\left(q_i - \mu_q\right)^2}{\nu\sigma_q^2}\right)^{-\frac{\nu+1}{2}}, \tag{13}$$

where ν manages outlier resistance ($\nu \to \infty$ gives a normal distribution, $\nu = 1$ gives the Cauchy distribution).

According to [19], autocorrelation correction via ARMA: quality estimates can be time-dependent, where the use of the ARMA(p, q) model will be provided for this:

$$q_i = \sum_{j=1}^{p}\varphi_j q_{i-j} + \sum_{k=1}^{q}\theta_k \in_{i-k} + \in_i, \tag{14}$$

where φ_j – autocorrelation coefficients, θ_k – ma component coefficients, $\in_i$ – white noise.

Next, we will solve the problem associated with the presence of outliers, which can significantly change the mean and variance. The solution to this problem is using the entropy criterion:

$$S\left(q\right) = -\sum_{i=1}^{N} P\left(q_i\right)\log P\left(q_i\right), \tag{15}$$

where q_i – the value of the quality assessment for i-th measurement, N – number of available measurements, $P(q_i)$ is estimated by the nuclear smoothed distribution function.

If $S(q) < S_{limit}$, outliers are present, and they are cut off by the criterion:

$$q_i \notin \left[Q_1 - 2IQR, Q_3 + 2IQR\right], \tag{16}$$

where Q_1 and Q_3 – first and third quartiles, $IQR = Q_3 - Q_1$.

When automatically detecting and excluding anomalous values, the automatic adaptation of the uncertainty threshold has the following problem – the value τ is chosen arbitrarily. Solution:

$$\tau^* = \arg\min_{\tau}\sum_{i=1}^{N}\left(q_i - \mu_q\right)^2 \cdot \mathrm{I}\left(\left|q_i - \mu_q\right| < \tau\right), \tag{17}$$

where q_i – the value of the quality assessment for i-th measurement, N – number of available measurements, $I(\cdot)$ – indicator function.

The study was conducted using an ERP-system for business processes management (SAP ERP) applied by medium-sized companies.

The ERP-system is actively employed by more than 200 users daily and operates in the distributed server architecture with cloud computing of information resources.

Quality indicators include the following parameters:

– average response time of API request (response time), ms, calculated with the help of JMeter (Apache benchmark);

– business process runtime, sec, evaluated with the help of APM tools (New Relic, DynaTrace);

– average CPU load of servers (%), determined by log files and Grafana metrics;

– level of customer satisfaction (%), calculated based on results of user surveys in the service desk system;

– the number of fatal crashes during the observation period (crash count), calculated by analysis of application logs with the help of the ELK stack (Kibana).

The initial set of data for practical research is provided in table 1.

Table 1

Initial data for practical research

Measurement time	Response time (ms)	Process execution time (s)	CPU load (%)	User satisfaction (%)	Crash count
10:00	320	2.5	45	88	0
10:10	330	2.7	47	87	1
10:20	290	2.4	43	89	0
10:30	305	2.6	44	86	0
10:40	315	2.8	48	85	2
10:50	1500 (outlier)	5.2 (anomaly)	90 (outlier)	50 (anomaly)	5 (anomaly)
11:00	295	2.3	42	88	0
11:10	310	2.5	45	87	1
11:20	300	2.4	43	89	0
11:30	295	2.3	41	90	0

Results and Discussion

Formalization of indicators and criteria for admissible uncertainty levels

The expected value and variance provide the initial statistical characterization of each quality metric. Relative uncertainty is then used to normalize the dispersion of the estimate with respect to its expected value. This normalization makes it possible to apply a common assessment procedure to indicators expressed in different units and measured on different numerical scales.

An estimate is classified as admissible when the relative-uncertainty condition defined by equation (4) and the boundary condition defined by equation (6) are simultaneously satisfied. Equation (7) expresses the corresponding maximum permissible standard deviation for the considered quality metric. The confidence interval defined by equation (5) additionally establishes the expected range of metric variation at the selected confidence level.

Under stable operating conditions, a predefined value of U_{max} can be used as the admissibility boundary. Under dynamic conditions, the threshold can be recalculated using the aggregated uncertainty characteristics of the monitored metrics according to equation (8). When the consequences of accepting or rejecting an estimate must be considered explicitly, the threshold can be selected by minimizing the loss function according to equation (9).

If at least one condition is violated, the estimate is classified as requiring correction.

The formalized criteria therefore constitute the input control mechanism of the hybrid method presented in Section 3.2. Bayesian adjustment, robust probabilistic modelling, autoregressive correction, and anomaly filtering are applied when the initial estimate does not satisfy the established uncertainty requirements.

Conceptual principles and structure of the hybrid uncertainty assessment method

The developed hybrid method combines complementary procedures for addressing the main sources of uncertainty in software quality assessment. The structure of the method is summarized in fig. 1.

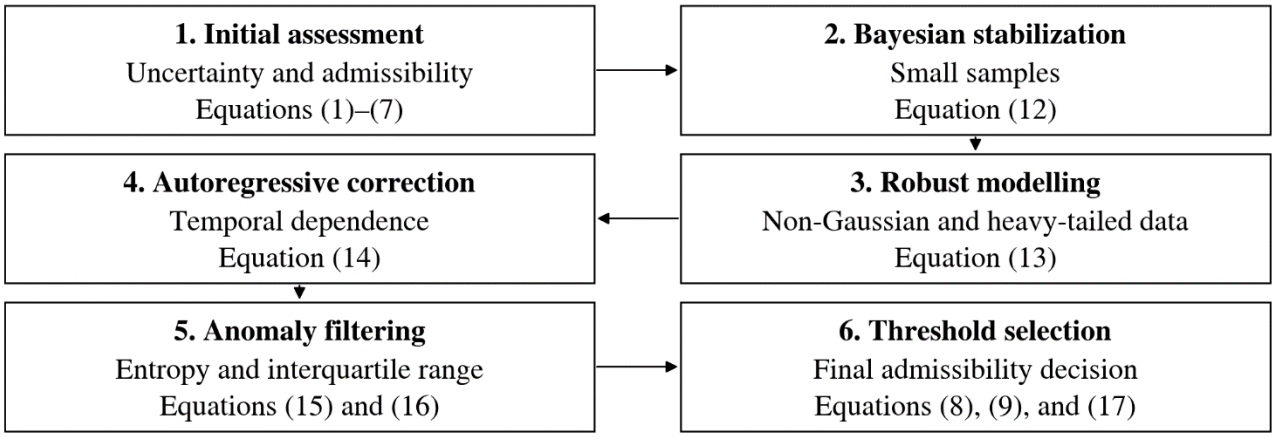


Fig. 1. Structure of the hybrid uncertainty assessment method

The method receives observed quality-metric series, historical estimates, and initial admissibility parameters as inputs. The final outputs are corrected metric estimates, identified anomalous observations, an adapted uncertainty threshold, and a decision on whether the resulting uncertainty level is admissible.

The experimental evaluation presented in Section 3.3 focuses on the Bayesian-adjustment and anomaly-filtering components of the proposed method.

Experimental evaluation of selected components of the proposed method

Application of the proposed method: Bayesian mean adjustment:

$$\mu_{RT} = \frac{N\,\bar{RT} + \lambda\mu_0}{N + \lambda}, \tag{18}$$

where N=9 is the number of measurements remaining after anomaly filtering, $\bar{RT}$ =306.7 ms is the mean response time of the filtered sample, μ_0=310 ms is the historical mean, and λ=5 is the weight assigned to the historical estimate. According to equation (18), the Bayesian-adjusted response-time estimate is μ_{RT}=307.9 ms.

Using linear quartile interpolation, Q_1=296.25 ms, Q_3=318.75 ms, and IQR=22.50 ms. Therefore, the anomaly-filtering interval defined by equation (16) is:

$$[Q_1 - 2IQR, Q_3 + 2IQR] = [251.25, 363.75] \text{ ms}.$$

The observation of 1500 ms falls outside this interval and was excluded from subsequent calculations.

The presented numerical example directly demonstrates the Bayesian-adjustment and IQR-based anomaly-filtering components of the proposed framework. The robust-distribution, ARMA, and entropy-based components were not independently evaluated using this limited dataset and require additional validation with longer time series and larger samples.

Comparative evaluation of the proposed method

The comparative results before and after anomaly filtering and Bayesian adjustment are presented in table 2.

Table 2

Comparison of software quality indicators before and after data processing

Indicator	Initial data	After anomaly filtering and Bayesian adjustment	Change
Relative uncertainty of response time	84.1%	4.1%	−80.0 percentage points
Response-time range	290–1500 ms	290–330 ms	Range width reduced from 1210 to 40 ms
Mean user satisfaction	83.9%	87.7%	+3.8 percentage points

After the anomalous observation of 1500 ms was excluded, the response-time range decreased to 290–330 ms. The relative uncertainty of the response-time estimate decreased from 84.1% to 4.1%, while mean user satisfaction increased from 83.9% to 87.7%.

Discussion

Each quality metric requires initial admissibility limits before assessment, defined by the criteria in equations (4), (6), and (7). The Bayesian prior should use stable-operation data; including an outlier such as the 1500 ms response at 10:50 would inflate the historical mean. The distribution and temporal dependence of each metric should also be examined before selecting the appropriate processing components. Robust probabilistic modelling is suitable for non-Gaussian or heavy-tailed data, whereas autoregressive correction should be applied when statistically meaningful temporal dependence is detected.

Anomaly-detection rules and their parameters should be established in advance and applied consistently. After each processing cycle, relative uncertainty should be recalculated and compared with the admissibility criterion. Identified anomalies and the reasons for their exclusion should be documented, while the uncertainty threshold should be recalibrated when systematic changes in operating conditions are observed.

The results should nevertheless be interpreted considering several limitations. The experimental evaluation was performed using one ERP system, a limited number of observations, and a short monitoring interval. In addition, only the Bayesian-adjustment and IQR-based anomaly-filtering components were evaluated directly. The robust probabilistic, ARMA, and entropy-based components require separate validation using larger datasets and longer time series.

Application of the method to software systems with other architectures, workloads, and quality indicators therefore requires independent validation and adjustment of the prior parameters and admissibility thresholds. Validation on other ERP platforms and on larger datasets with longer observation periods remains future work.

Conclusion

We formalised a unified set of quality indicators with admissibility criteria and combined Bayesian stabilisation, robust probabilistic modelling, autoregressive correction, and anomaly filtering with adaptive threshold selection into a sequential hybrid method. On operational ERP data, IQR-based anomaly filtering followed by Bayesian adjustment produced a response-time estimate of 307.9 ms from the remaining nine observations.

Comparative evaluation demonstrated a substantial stabilisation of the assessed indicators after anomaly filtering and Bayesian adjustment. Relative uncertainty of the response-time metric decreased from 84.1% to 4.1%, while the response-time range narrowed from 290–1500 ms to 290–330 ms. The estimated mean user-satisfaction indicator increased from 83.9% to 87.7%, corresponding to 3.8 percentage points.

Authors' contributions

Anton SHANTYR – Conceptualization, Methodology, Investigation, Writing - original draft; Olha ZINCHENKO – Conceptualization, Methodology, Validation, Writing - review & editing, Supervision; Kamila STORCHAK – Investigation, Validation, Writing - review & editing; Andrii BONDARCHUK – Formal analysis, Validation, Writing - review & editing; Nataliia LASHCHEVSKA – Investigation, Writing - review & editing.

Acknowledgements, funding sources

The study was performed without financial support.

Declaration on artificial intelligence

The authors confirm that they did not use artificial intelligence technologies in creating content the submitted work. Artificial intelligence tools were used for linguistic and stylistic editing of the text and did not affect the scientific content, results, or conclusions of the study.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

Data availability

Data will be made available on reasonable request.

References

1. Herber, P., & Wijs, A. (Eds.). (2024). Integrated formal methods: 18th International Conference, IFM 2023, Leiden, The Netherlands, November 13–15, 2023, Proceedings. Lecture Notes in Computer Science, 14300. Springer Cham. https://doi.org/10.1007/978-3-031-47705-8

2. Ali, A., & Saleem, N. N. (2022). Classification of software systems attributes based on quality factors using linguistic knowledge and machine learning: A review. Journal of Education and Science, 31(3), 66–90. https://doi.org/10.33899/edusj.2022.134024.1245

3. Uddin, S., Ong, S., Lu, H., & Matous, P. (2024). Integrating machine learning and network analytics to model project cost, time and quality performance. Production Planning & Control, 35(12), 1475–1489. https://doi.org/10.1080/09537287.2023.2196256

4. Kosmatov, N., & Kovács, L. (Eds.). (2025). Integrated formal methods: 19th International Conference, IFM 2024, Manchester, UK, November 13–15, 2024, Proceedings. Lecture Notes in Computer Science, 15234. Springer Cham. https://doi.org/10.1007/978-3-031-76554-4

5. Dheeraj, K. N., G., R. J., & A., L. (2021). Crop quality prediction using ML and neural networks. International Journal on Cybernetics & Informatics, 10(2), 7–11. https://doi.org/10.5121/ijci.2021.100202

6. Zhang, J., Liu, Z., Hu, X., Liu, P., Hu, Z., & Kuang, L. (2024). FATE: A flexible FPGA-based automatic test equipment for digital ICs. Electronics, 13(9), Article 1667. https://doi.org/10.3390/electronics13091667

7. Hahn, S.-J., & Lee, B.-H. (2023). Quality evaluation of small-scale software implementation results. Journal of the Korean Institute of Information Technology, 21(1), 1–10. https://doi.org/10.14801/jkiit.2023.21.1.1

8. Hundal, S., Eskola, A., & Troudi, M. (2022). Do board of directors' characteristics and executive remuneration impact financial reporting quality? A quantitative analysis of the Nordic manufacturing sector. Corporate Ownership and Control, 20(1), 59–67. https://doi.org/10.22495/cocv20i1art5

9. Zhang, X., Li, Z., Zou, Z., Gao, X., Xiong, Y., Jin, D., Li, J., & Liu, H. (2023). Informative data selection with uncertainty for multimodal object detection. IEEE Transactions on Neural Networks and Learning Systems, 1–13. https://doi.org/10.1109/TNNLS.2023.3270159

10. Zhao, J., Wang, Y., Mancenido, M. V., Chiou, E. K., & Maciejewski, R. (2023). Evaluating the impact of uncertainty visualization on model reliance. IEEE Transactions on Visualization and Computer Graphics, 1–15. https://doi.org/10.1109/TVCG.2023.3251950

11. Fumagalli, F., Muschalik, M., Hüllermeier, E., & Hammer, B. (2023). Incremental permutation feature importance (IPFI): Towards online explanations on data streams. Machine Learning. https://doi.org/10.1007/s10994-023-06385-y

12. ter Beek, M. H., Chapman, R., Cleaveland, R., Garavel, H., Gu, R., ter Horst, I., Keiren, J. J. A., Lecomte, T., Leuschel, M., Rozier, K. Y., Sampaio, A., Seceleanu, C., Thomas, M., Willemse, T. A. C., & Zhang, L. (2025). Formal methods in industry. Formal Aspects of Computing, 37(1), Article 7, 1–38. https://doi.org/10.1145/3689374

13. Vagaská, A., Gombár, M., & Panda, A. (2023). Optimization methods in general. In Optimization methods in mathematical modeling of technological processes (pp. 29–52). Springer Cham. https://doi.org/10.1007/978-3-031-35339-0_3

14. Jin, W., & Wang, Z. (2022). Research and application of quantitative evaluation method based on product quality characteristics. BCP Business & Management, 17, 84–90. https://doi.org/10.54691/bcpbm.v17i.374

15. Kamaletdinov, S., Aripov, N., Khudayberganov, S., Bashirova, A. M., & Akhmedov, M. D. (2023). Evaluation of data quality based on Bayesian networks in railway rolling stock monitoring systems. E3S Web of Conferences, 460, 04014. https://doi.org/10.1051/e3sconf/202346004014

16. Kaneko, H. (2022). Cross-validated permutation feature importance considering correlation between features. Analytical Science Advances. https://doi.org/10.1002/ansa.202200018

17. Koi-Akrofi, G. Y., Tanye, H., Quist, S. C., Koi-Akrofi, J., Gaisie, E., Agangiba, M., & Yeboah, D. (2024). Towards a software quality factor assessment model for learning management systems (LMS). Indian Journal of Science and Technology, 17(23), 2463–2468. https://doi.org/10.17485/ijst/v17i23.1006

18. Nguyen, P. H., Sen, S., Jourdan, N., Cassoli, B., Myrseth, P., Armendia, M., & Myklebust, O. (2022). Software engineering and AI for data quality in cyber-physical systems: SEA4DQ'21 workshop report. ACM SIGSOFT Software Engineering Notes, 47(1), 26–29. https://doi.org/10.1145/3502771.3502781

19. Oh, H.-S., Kim, D., & Lee, S. (2021). Review on the quality attributes of an integrated simulation software for weapon systems. Journal of the Korea Institute of Military Science and Technology, 24(4), 408–417. https://doi.org/10.9766/kimst.2021.24.4.408

20. Severini, T. A. (2026). Some properties of a parameterization useful in integrated likelihood inference. Metrika, 89, 305–321. https://doi.org/10.1007/s00184-025-01006-1

А. С. Шантир, О. В. Зінченко, К. П. Сторчак, А. П. Бондарчук, Н. О. Лащевська

ВИЗНАЧЕННЯ ДОПУСТИМИХ РІВНІВ НЕВИЗНАЧЕНОСТІ ПРИ ОЦІНЮВАННІ ЯКОСТІ ПРОГРАМНИХ СИСТЕМ

У моніторингу якості програмного забезпечення помилки вимірювання, а також варіації в середовищі виконання, архітектурі програмного забезпечення та поведінці користувачів створюють серйозні проблеми для кількісної оцінки якості програмного забезпечення. Для вирішення вищезазначених проблем у статті пропонується інтегрований підхід, що включає байєсівський підхід корекції середнього, робастні ймовірнісні моделі, авторегресивні моделі та моделі виявлення аномалій. Композитний підхід є адаптивним і визначає так звану допустиму невизначеність на основі історичних даних та поточного стану системи. У статті представлено приклад застосування байєсівського підходу корекції середнього та підходу фільтрації аномалій, протестованих на реальному наборі даних з операційної системи планування ресурсів підприємства (ERP). Завдяки застосуванню підходів для фільтрації аномалій та байєсівської корекції відносна невизначеність розглянутого показника часу відгуку зменшилася з 84,1% до 4,1% і залишалася стабільною, в середньому 290–330 мс. Крім того, середня задоволеність користувачів зросла на 3,8 процентних пункти до 87,7% з 83,9%.

Ключові слова: програмні системи, оцінювання якості, невизначеність, моніторинг, оптимізація, моделювання, байєсівська корекція.

Надійшла до редакції: 09.09.2026
Прийнята до друку: 29.09.2026
Опубліковано: 30.10.2026
© 2026 Shantyr A., Zinchenko O., Storchak K., Bondarchuk A., Lashevska N.
Цей матеріал ліцензовано за умовами CC BY 4.0.https://creativecommons.org/licenses/by/4.0/