

FORMALIZATION OF THE STRUCTURAL COMPLEXITY OF SOFTWARE AS A FACTOR IN THE DESIGN ASSESSMENT OF ITS RELIABILITY

A. P. Nyrkov, V. V. Korotkov, V. D. Gaskarov

Admiral Makarov University of Maritime and Inland Shipping,
St. Peterburg, Russian Federation

This study is devoted to one aspect of designing automated control systems for water transport facilities, such as logistics centers, container terminals, ports, ship repair yards, shipping companies, and lock navigation channels on regulated inland waterways. The problem of developing a methodology for the quantitative design assessment of the structural complexity of software under development is considered, since this complexity is a significant factor in assessing its reliability, and increasing complexity inevitably leads to various types of vulnerabilities in program code. A set of factors on which structural complexity depends is defined. A corresponding set of terms is introduced to designate these factors assuming the use of a modular development principle: module call spectrum, module alternative spectrum, module depth, module cyclicity, weighted module depth, weighted module cyclicity. The expediency of transitioning to the corresponding dimensionless indicators is justified. Building on the previously formulated multiplicative method of aggregating a general system integral indicator from a set of partial indicators, a final formula for the design indicator of structural complexity of automated control system software is provided, and a method for including it in the design assessment of the overall reliability of the control loop is proposed. A metric for vector evaluation of differences in structural complexity among alternative software products is introduced.

Key words: automated control system, water transport, project, software module, structure, complexity, reliability.

For citation:

Nyrkov, Anatoliy P., V. V. Korotkov and V. D. Gaskarov. "Formalization of the structural complexity of software as a factor in the design assessment of its reliability." *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S. O. Makarova*. 17.4 (2025): 613–620. DOI: 10.21821/2309-5180-2025-17-4-613-620.

УДК 004.6:656.6

ФОРМАЛИЗАЦИЯ СТРУКТУРНОЙ СЛОЖНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ КАК ФАКТОРА ПРОЕКТНОЙ ОЦЕНКИ ЕГО НАДЕЖНОСТИ

А. П. Нырков, В. В. Коротков, В. Д. Гаскаров

ФГБОУ ВО «ГУМРФ имени адмирала С. О. Макарова»,
Санкт-Петербург, Российская Федерация

Исследование посвящено одному из аспектов проектирования автоматизированных систем управления объектами водного транспорта, такими как контейнерные терминалы, порты, судоремонтные предприятия, судоходные компании, шлюзованные судоходные каналы на регулируемых внутренних водных путях. Рассмотрена проблема создания методики проектной количественной оценки структурной сложности разрабатываемого программного обеспечения, поскольку указанная сложность является существенным фактором для оценки его надежности, а повышение сложности неизбежно приводит к возникновению различного рода уязвимостей в программном коде. Определена совокупность факторов, от которых зависит структурная сложность. Введена совокупность терминов для обозначения этих факторов при использовании модульного принципа разработки: «спектр модуля по вызовам», «спектр модуля по альтернативам», «глубина модуля», «цикличность модуля», «взвешенная глубина модуля», «взвешенная цикличность модуля». Обоснована целесообразность перехода к соответствующим безразмерным показателям. В развитие сформулированного ранее мультипликативного способа агрегирования общесистемного интегрального показателя из набора частных показателей приведена итоговая формула показателя проектной структурной

сложности программного обеспечения автоматизированной системы управления и предложен способ его включения в проектную оценку общей надежности контура управления. Введена метрика для векторной оценки различия структурной сложности альтернативных программных продуктов.

Ключевые слова: автоматизированная система управления, водный транспорт, проект, программный модуль, структура, сложность, надежность.

Для цитирования:

Нырков А. П. Формализация структурной сложности программного обеспечения автоматизированной системы управления как фактора проектной оценки его надежности / А. П. Нырков, В. В. Коротков, В. Д. Гаскаров // Вестник Государственного университета морского и речного флота имени адмирала С. О. Макарова. — 2025. — Т. 17. — № 4. — С. 613–620. DOI: 10.21821/2309-5180-2025-17-4-613-620. — EDN ZFQGEC.

Введение (Introduction)

Оценка надежности автоматизированной системы управления является многофакторной. Эта многофакторность системно проявляется во взаимодействии аппаратного контура, программного обеспечения и больших данных, а внутри программ — взаимодействие их алгоритмически различных фрагментов. Понятие *сложность технической системы* концептуально связано с понятием *большая система*. Это последнее в рамках «общей теории систем» в 1930-е гг. исследовал в своих работах Л. фон Берталанфи, в 1950–70-е гг. предлагалось множество различных подходов к определению системы (М. Месарович, Л. Заде, Р. Акофф, Ю. А. Урманцев и др.). Исследования, связанные с этим понятием, не потеряли актуальность до настоящего времени.

Формальное определение такой сложной технической системы, как АСУ в целом, так и ее подсистемы — программного обеспечения, с учетом возможных вариантов предполагает следующие ключевые особенности (например, [1], [2]):

- *структурированность* — применительно к ПО АСУ это выражается в модульном принципе его построения
- *несводимость* функциональных свойств целостной системы к формальному объединению свойств составляющих элементов (эмерджентность);
- *многообразие* связей между элементами; применительно к ПО АСУ это выражается в многообразии и глубине последовательных (каскадных) вызовов модулей;
- *стохастичность* как проявление частотной устойчивости некоторых событий (закон больших чисел);
- *случайность* как наступление редких (нежелательных) событий, не имеющих частотной характеристики
- *реактивность* по отношению к внешней среде — в случае ПО в качестве таковой следует считать информационные потоки, поступающие в аппаратный контур.

Проблема влияния структурной сложности большого программного комплекса на надежность его работы представляется недостаточно изученной. Важной вехой является монография [3], в которой обобщен большой массив отечественных и зарубежных исследований, получивших дальнейшее развитие в ряде более поздних публикаций [4]–[13]. В работах [3]–[5] доминирующими по надежности ПО являются ошибки в его проектировании. Разработка проекта ПО предполагает формирование модульной структуры с ее многообразными связями по вызовам и данным. Этот тезис развивается в данной работе в следующей трактовке: структурная сложность ПО, оцениваемая адекватно сформированным интегральным количественным показателем, может и должна рассматриваться как важный фактор прежде всего проектной надежности (в противопоставлении с эксплуатационной). В ряде публикаций, в частности, [6]–[8], не рассматривается в явной форме структурная сложность ни в качественном, ни в содержательном плане, ни в плане формализованной количественной оценки.

Авторы публикаций [9–12] для оценки структурной сложности программного обеспечения предлагают использовать вероятностный подход. В [9] для учета влияния «архитектуры», т. е. структуры, ПО предложена формула для определения среднего времени простоя из-за программ-

ных сбоев, которая имеет вид математического ожидания и использует априорные либо статистически проявленные вероятности появления ошибок программирования, независимо сгруппированные по уровням. Таким образом, та часть общей надежности, которая относится к архитектуре ПО, имеет размерность времени. Представляется, что соответствующий показатель должен быть безразмерным, так как это упрощает его включение в общий интегральный показатель (тоже безразмерный) в виде коэффициента одного из множителей [14]. Кроме того, использование вероятностей программных сбоев предполагает тестовую эксплуатацию системы управления, что неприемлемо для ранних стадий проектирования ПО АСУ.

Таким образом, целью работы является агрегирование количественной оценки структурной сложности программного обеспечения на стадии проектирования для прогноза его надежности. В рамках поставленной цели решаются задачи выделения факторов структурной сложности и учета их совместного влияния.

Методы и материалы (Methods and Materials)

Методологические предпосылки. Рассмотрим априорные предпосылки для формального определения интегрального показателя. При построении количественного показателя структурной сложности ПО АСУ будем исходить из предпосылки систематического использования разработчиками принципов структурного программирования (в частности, модульного принципа и отказа от операторов безусловного перехода типа *go to*). В этом случае структуру ПО можно представить связным ориентированным графом G без кратных ребер с одним шарниром (входом в систему), с выходящими из него ребрами к не взаимодействующим между собой функциональным подсистемам (за исключением, возможно, доступа к общим данным). Отдельным вершинам при этом соответствуют программные модули, ребрам — обращения к другим модулям.

Представляется, что при отказе от использования рекурсий (укорачивающих код, но увеличивающих расход памяти и затрудняющих отладку) сложность структуры ПО должна учитывать следующие характеристики каждого модуля, входящего в программный комплекс АСУ:

- *размер программного кода модуля* — количество операторов исходного текста на проектно-ориентированном языке высокого уровня либо объем соответствующего исполняемого EXE-файла; эта характеристика должна вычисляться единообразно для всех модулей, чтобы сохранить сопоставимость результатов;

- *спектр модуля по вызовам*, т. е. количество непосредственных обращений из него к другим модулям (количество ребер, выходящих из представляющей его вершины);

- *спектр модуля по общему количеству используемых альтернатив*;

- *глубину модуля*, характеризующую «вертикальную», или «каскадную» вложенность других модулей по вызовам из данного (максимальную по количеству ребер длину *полного* пути, т. е. нециклического (ввиду предположенного отсутствия рекурсий) ориентированного пути из ребер, начинающегося в вершине данного модуля;

- *цикличность модуля* — количество циклов в теле модуля (поскольку вложенность циклов означает явное усложнение его структуры, это должно отражаться в количественном значении цикличности);

- *взвешенную глубину модуля* — суммарную длину программных кодов, образующих путь, длина которого равна глубине модуля;

- *взвешенную цикличность модуля* — суммарную длину программных кодов, образующих циклы, при этом длина вложенных циклов оказывается включенной с соответствующей кратностью в соответствии со сформулированным ранее требованием.

Методологическим основанием для включения в рассмотрение взвешенных показателей является не формализованное, но бытующее в профессиональном программистском сообществе в течение нескольких десятилетий представление о том, что «в среднем» сложность программы, а значит, и частота ошибок программирования пропорциональны длине программы (разумеется, с большой дисперсией) [14].

Математическая модель показателя структурной сложности. Опишем параметры модели. Пусть n — количество (пронумерованных) программных модулей (вершин графа G). В данном случае из рассмотрения исключены стандартные библиотечные подпрограммы и функции используемых систем программирования, предполагая их свободными от ошибок. $A_{n \times n} = (a_{ij})$ — его бинарная матрица смежности [15], [16]: $a_{ij} = 1$, если из i в j ведет ребро, и $a_{ij} = 0$ в противном случае.

Введем обозначения для частных характеристик структурной сложности модуля i : l_i — размер модуля (вес i -й вершины), измеряемый его длиной, т. е. числом операторов; s_i — спектр; γ_i — количество альтернатив, введенных всеми условными операторами; h_i — глубина; c_i — цикличность; h'_i — взвешенная глубина; c'_i — взвешенная цикличность.

Отдельный полный нециклический путь модуля i задается упорядоченным набором вершин $\pi(i) = (i, u_2, \dots, u_t)$, соответствующих последовательности вложенных вызовов при его исполнении, а именно:

– спектру модуля по вызовам:

$$s_i = \sum_{j \neq i} a_{ij}; \quad (1)$$

– спектру модуля по альтернативам:

$$\gamma_i = n_i^{if} + n_i^{choice}, \quad (2)$$

где n_i^{if} и n_i^{choice} — количество альтернатив, введенных всеми условными операторами модуля типа *if* и *choice*, соответственно;

– глубине модуля:

$$h_i = \max_{\pi(i)} t \quad (3)$$

(если при этом спектр модуля $s_i = 0$, то полагаем $h_i = 0$);

– взвешенной глубине:

$$h'_i = l_i + l_{u_2} + \dots + l_{u_t}, \quad (4)$$

где суммирование ведется по тем вершинам графа G , которые задают глубину модуля.

Если значение t реализуется одновременно на нескольких полных путях: $\pi_1(i), \dots, \pi_k(i)$, то полагаем h'_i равным максимальному из соответствующего набора значений:

$$h'_i = \max_{\pi_j(i)} \{h'_i(\pi_j(i))\}. \quad (5)$$

Пусть $\zeta(i)$ — какой-либо цикл, содержащийся в модуле i , $l[\zeta(i)]$ — его длина. Тогда, в соответствии с определением взвешенной цикличности, ее значение

$$c'_i = \sum_{\zeta(i)} l[\zeta(i)]. \quad (6)$$

Характеристики структурной сложности (1)–(6) являются размерными, т. е. измеряемыми в принятых единицах длины программы. В то же время желательно иметь аналогичный набор безразмерных характеристик для сравнения разных программ в указанном аспекте. Априори понятно, что короткая программа может с содержательной точки зрения иметь сложную, «запутанную» структуру, и в то же время длинная программа может оказаться структурно простой, например, ее граф может иметь структуру бинарного дерева или даже линейного списка.

Пусть L — объем ПО в целом, N — общее количество модулей.

Определим:

– средний спектр модуля по вызовам:

$$S = \left(\sum_i s_i \right) / N; \quad (7)$$

– среднюю глубину модуля по альтернативам:

$$\Gamma = \left(\sum_i \gamma_i \right) / L; \quad (8)$$

– среднюю взвешенную глубину модуля:

$$H' = \left(\sum_i h'_i \right) / L ; \quad (9)$$

– среднюю взвешенную цикличность модуля:

$$C' = \left(\sum_i c'_i \right) / L . \quad (10)$$

Теперь, исходя из мультипликативного принципа построения общего показателя проектной надежности [14] и его частных показателей-множителей, получим проектный показатель структурной сложности ПО АСУ в виде

$$K_{\text{стр}} = S \cdot H \cdot H' \cdot C' \cdot \Gamma . \quad (11)$$

Если на первом этапе проектирования ПО, определяя его архитектуру [14], примем, что длина каждого модуля i в формулах (3)–(6) равна $1 + s_i + l_i$ (операторы между вызовами других модулей отсутствуют), получим по формуле (11) значение $K_{\text{кар}}$, которое назовем *каркасной сложностью*. Эта модификация дает простой механизм для оценки и сравнения сложности любых программных продуктов без содержательного анализа их полного текста.

Вектор частных структурных характеристик: $X = (L, S, H, H', C', \Gamma)$, может быть использован для сравнения альтернативных проектных вариантов, характеризующихся значениями структурной сложности ПО $X^{(1)}$ и $X^{(2)}$, с применением для задания расстояния между строками евклидовой метрики $\rho_E(X^{(1)}, X^{(2)})$:

$$\rho_E(X^{(1)}, X^{(2)}) = \sqrt{(L_1 - L_2)^2 + (S_1 - S_2)^2 + (H_1 - H_2)^2 + (H'_1 - H'_2)^2 + (C'_1 - C'_2)^2 + (\Gamma_1 - \Gamma_2)^2} . \quad (12)$$

Если проектировщики имеют основания различать степень важности отдельных компонент вектора X , то возможна модификация формулы (12), использующая соответствующие веса p_j , $p_j \geq 0$, $\sum_{j=1}^6 p_j = 1$ для квадратов соответствующих разностей.

При вычислении глубины h_i и взвешенной глубины h'_i для каждой вершины i может применяться алгоритм перечисления полных путей из фиксированной вершины в ациклическом ориентированном графе, описанный, например, в [13] и [15, 16].

Построенный в работе [14] интегральный показатель H проектной оценки надежности АСУ имеет вид

$$H = H_{\text{апп}} \cdot H_{\text{пр}} \cdot H_{\text{дан}} , \quad (13)$$

где $H_{\text{апп}}$ — надежность аппаратного контура,

$H_{\text{пр}}$ — надежность программного обеспечения,

$H_{\text{дан}}$ — надежность хранения данных.

Теперь можно сделать множитель $H_{\text{пр}}$ более информативным. Поскольку увеличение структурной сложности ПО должно уменьшать показатель его надежности в соответствии с используемым *принципом мультипликативности*, прежнее значение $H_{\text{пр}}$ следует разделить на $K_{\text{стр}}$:

$$H_{\text{пр}} \leftarrow H_{\text{пр}} / K_{\text{стр}} . \quad (14)$$

Таким образом, построенный показатель применим не только на стадии проектирования, но и к уже функционирующему ПО АСУ, так как он позволяет динамически сопоставлять временную динамику количества выявленных в ходе эксплуатации ошибок с финальным показателем сложности.

Результаты исследования и их обсуждение (Results and Discussion)

Соблюдение принципа системности при построении проектной оценки надежности программного обеспечения разрабатываемой АСУ транспортной системы с рассредоточенной сетью удаленных объектов, сетевого аппаратного контура, облачных технологий хранения и передачи данных предполагает наряду с другими факторами учет, который необходим на каждой стадии про-

движения проекта структурной сложности программного продукта. До настоящего времени отсутствовали формализованные количественные показатели и методики их построения, которые можно оценить как *системные*, учитывающие различные аспекты структуры.

В данной работе приведен перечень соответствующих факторов, которые позволяют сделать вывод о том, что наряду с очевидным влиянием на надежность длины программы и ее составных частей, необходим учет следующих параметров: спектр модуля по вызовам, спектр модуля по альтернативам, глубина модуля, цикличность модуля. Для обеспечения сопоставимости результатов оценки для разных вариантов ПО следует перейти к безразмерным показателям и использовать мультипликативный способ их агрегирования. Сформированный показатель структурной сложности включается в оценку проектной надежности в степени -1 .

Заключение (Conclusion)

Частью решения общей задачи разработки методики проектной оценки системной надежности АСУ для объектов инфраструктуры водного транспорта является построение оценки надежности программного обеспечения. Методика построения может использовать мультипликативный принцип агрегирования частных характеристик и должна учитывать размер ПО, модульную структуру, глубину последовательных вызовов модулей, а также их количество, набор альтернативных разветвлений, количество циклов. Получаемый при этом показатель структурной сложности влияет на интегральный показатель проектной оценки надежности программного обеспечения по схеме обратной пропорциональной зависимости.

СПИСОК ЛИТЕРАТУРЫ

1. *Цветков В. Я.* Сложные технические системы / В. Я. Цветков // Образовательные ресурсы и технологии. — 2017. — № 3 (20). — С. 86–92.
2. *Цветков В. Я.* Ресурсность и интегративность сложной организационно технической системы / В. Я. Цветков // Международный журнал прикладных и фундаментальных исследований. — 2016. — № 5–4. — С. 676. — EDN VVTJIV.
3. *Липаев В. В.* Надежность программных средств. Сер.: Информатизация России на пороге XXI века / В. В. Липаев. — М: СИНТЕГ, 2018. — 296 с.
4. *Климов С. М.* Методика оценки функциональной надежности компонент программно-аппаратной встраиваемой микропроцессорной системы управления / С. М. Климов, Ю. В. Сосновский, Д. Р. Чачиев // Надежность. — 2025. — Т. 25. — № 1. — С. 58–66. DOI: 10.21683/1729-2646-2025-25-1-58-66. — EDN TXVYZH.
5. *Боровиков С. М.* Оценка ожидаемой надежности прикладных программных средств для компьютерных информационных систем / С. М. Боровиков, В. О. Казючиц, В. В. Хорошко [и др.] // Информатика. — 2021. — Т. 18. — № 1. — С. 84–95. DOI: 10.37661/1816-0301-2021-18-1-84-95. — EDN AHDRQC.
6. *Мякишев Д. В.* Принципы и методы создания надежного программного обеспечения / Д. В. Мякишев. — М: Инфра-Инженерия, 2025. — 112 с.
7. *Звонарева А. А.* Основные аспекты надежности программного обеспечения систем управления / А. А. Звонарева, А. О. Толоконский // Вестник Национального исследовательского ядерного университета «МИФИ». — 2021. — Т. 10. — № 5. — С. 429–435. DOI: 10.1134/S2304487X21050126. — EDN HZYDBT.
8. *Наумов А. А.* Надежность программного обеспечения и методы ее повышения / А. А. Наумов, А. Р. Айдинян // Инженерный вестник Дона. — 2018. — № 2(49). — С. 98. — EDN VAQUWA.
9. *Тынченко В. В.* К вопросу оценки надежности программного обеспечения с многоуровневой архитектурой / В. В. Тынченко, Р. Ю. Царев // Современные проблемы науки и образования. — 2015. — № 2–1. — С. 204. — EDN UHWZRD.
10. *Рогова Д. В.* Особенности использования структурного подхода при разработке надежного программного обеспечения / Д. В. Рогова, А. Г. Суворов // Актуальные проблемы авиации и космонавтики: Сборник материалов VII Международной научно-практической конференции, посвященной Дню космонавтики: в 3 томах, Красноярск, 12–16 апреля 2021 года — Красноярск: Федеральное государственное бюд-

жетное образовательное учреждение высшего образования «Сибирский государственный университет науки и технологий имени академика М. Ф. Решетнева», 2021. — С. 358–360. — EDN COMMSF.

11. Nyrkov A. Mathematical Models for Solving Problems of Reliability Maritime System / A. Nyrkov, K. Goloskokov, E. Koroleva, S. Sokolov, A. Zhilenkov, S. Chernyi // *Advances in Systems, Control and Automation: ETAEERE-2016* — Springer Singapore, 2018. — С. 387–394. DOI: 10.1007/978-981-10-4762-6_37.

12. Shipunov I. S. Developing a Reliable Information Storage Scheme Within a Partially Unmanned Maritime Convoy / I. S. Shipunov, A. P. Nyrkov, D. A. Evtushenko, A. V. Kostenkova, I. V. Li, A. A. Nyrkov // *2022 Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)* — 2022. — С. 439–442. DOI: 10.1109/ElConRus54750.2022.9755534.

13. Рванова А. С. Разработка модели оценки структурной сложности программ / А. С. Рванова, Н. С. Кольева, М. В. Панова // *Цифровые модели и решения.* — 2024. — Т. 3. — № 2. — С. 5–16. DOI: 10.29141/2949-477X-2024-3-2-1. — EDN UEZYBI.

14. Коротков В. В. Математическая модель формирования интегральной оценки надежности автоматизированной системы управления на стадии проектирования / В. В. Коротков // *Вестник государственного университета морского и речного флота имени адмирала С. О. Макарова.* — 2025. — Т. 17. — № 3. — С. 449–458. DOI: 10.21821/2309-5180-2025-17-3-449-458. — EDN WVZEVZ.

15. Юмашева Е. С. Интеграция графовых моделей для обнаружения аномалий в сетевом трафике / Е. С. Юмашева, А. П. Нырков // *Перспективы науки.* — 2025. — № 6(189). — С. 102–107.

16. Куделя В. Н. Методы перечисления путей в графе / В. Н. Куделя // *Научно-технические исследования в космических исследованиях Земли.* — 2023. — Т. 15. — № 5. — С. 28–38. DOI: 10.36724/2409-5419-2023-15-5-28-38. — EDN HQEASN.

REFERENCES

1. Tsvetkov, V. Ya. “Complex technical systems.” *Educational Resources And Technologies* 3(20) (2017): 86–92. DOI: 10.21777/2500-2112-2017-3-86-92.

2. Tsvetkov, V. Ya. “Resursnost’ i integrativnost’ slozhnoy organizatsionno tekhnicheskoy sistemy.” *Mezhdunarodnyy zhurnal prikladnykh i fundamental’nykh issledovaniy* 5–4 (2016): 676.

3. Lipaev, V. V. *Nadezhnost’ programmnykh sredstv. Ser.: Informatizatsiya Rossii na poroge XXI veka* M: SINTEG, 2018: 296.

4. Klimov, S. M., Yu. V. Sosnovskiy and D. R. Chachiev. “A methodology for evaluating the functional dependability of the components of an embedded software and hardware microprocessor-based control system.” *Dependability* 25.1 (2025): 58–66. DOI: 10.21683/1729-2646-2025-25-1-58-66.

5. Borovikov, S. M., V. O. Kazyuchits and V. V. Khoroshko [i dr.]. “Assessment of expected reliability of applied software for computer-based information systems.” *Informatics* 18.1 (2021): 84–95. DOI: 10.37661/1816-0301-2021-18-1-84-95.

6. Myakishev, D. V. *Printsipy i metody sozdaniya nadezhnogo programmnoy obespecheniya* M: Infra-Inzheneriya, 2025: 112.

7. Zvonareva, A. A. and A. O. Tolokonskiy. “Basic aspects of the reliability of control system software.” *Vestnik Natsional’nogo issledovatel’skogo yadernogo universiteta “MIFI”* 10.5 (2021): 429–435. DOI: 10.1134/S2304487X21050126.

8. Naumov, A. A. and A. R. Aydiyanyan. “Software reliability and methods of its improvement.” *Engineering Journal Of Don* 2(49) (2018): 98.

9. Tynchenko, V. V. and R. Yu. Tsarev. “Toward the problem of evaluation of the reliability of software with multiple level architecture.” *Sovremennyye problemy nauki i obrazovaniya* 2–1 (2015): 204.

10. Rogova, D. V. and A. G. Suvorov. “Features of using a structural approach in developing reliable software.” *Aktual’nye problemy aviatsii i kosmonavтики: Sbornik materialov VII Mezhdunarodnoy nauchno-prakticheskoy konferentsii, posvyaschennoy Dnyu kosmonavтики: v 3 tomakh, Krasnoyarsk, 12–16 aprelya 2021 goda* Krasnoyarsk: Federal’noe gosudarstvennoe byudzhethoe obrazovatel’noe uchrezhdenie vysshego obrazovaniya “Sibirskiy gosudarstvennyy universitet nauki i tekhnologii imeni akademika M. F. Reshetneva”, 2021: 358–360.

11. Nyrkov, A., K. Goloskokov, E. Koroleva, S. Sokolov, A. Zhilenkov and S. Chernyi. “Mathematical Models for Solving Problems of Reliability Maritime System.” *Advances in Systems, Control and Automation: ETAEERE-2016* Springer Singapore, 2018: 387–394. DOI: 10.1007/978-981-10-4762-6_37.

12. Shipunov, I. S., A. A. Nyrkov, et al. “Developing a Reliable Information Storage Scheme Within a Partially Unmanned Maritime Convoy.” *2022 Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)* — 2022: 439–442. DOI: 10.1109/ElConRus54750.2022.9755534.

13. Rvanova, A. S., N. S. Kol’eva and M. V. Panova. “Development model for assessing the structural complexity of programs.” *Digital Models And Solutions 3.2* (2024): 5–16. DOI: 10.29141/2949-477X-2024-3-2-1.

14. Korotkov, V. V. “Methodology for forming an integral assessment of the reliability of automated control systems of transport infrastructure at the design stage.” *Vestnik gosudarstvennogo universiteta morskogo i rechnogo flota im. admiral S. O. Makarova 17.3* (2025): 449–458. DOI: 10.21821/2309-5180-2025-17-3-449-458.

15. Yumasheva, E. S. and A. P. Nyrkov. “Integrating Graph Models to Detect Anomalies in Network Traffic.” *Science Prospects 6*(189) (2025): 102–107.

16. Kudelya, V. N. “Methods for enumerating paths in a graph.” *High Technologies In Earth Space Research. H&Es Research 15.5* (2023): 28–38. DOI: 10.36724/2409-5419-2023-15-5-28-38.

ИНФОРМАЦИЯ ОБ АВТОРАХ

Ныркoв Анатолий Павлович —

доктор технических наук, профессор
ФГБОУ ВО «ГУМРФ имени адмирала
С. О. Макарова»

198035, Российская Федерация, Санкт-Петербург,
ул. Двинская, 5/7

e-mail: kaf.koib@gmail.com, nyrkowap@gumrf.ru

Коротков Виталий Валерьевич —

доцент кафедры комплексного обеспечения
информационной безопасности ГУМРФ им.
адмирала С. О. Макарова

198035, Россия, Санкт-Петербург,
ул. Двинская, д. 5/7.

e-mail: korotkovvv@gumrf.ru

Гаскаров Вагиз Дилляурович —

доктор технических наук, профессор
ФГБОУ ВО «ГУМРФ имени адмирала
С. О. Макарова»

198035, Российская Федерация, Санкт-Петербург, ул.
Двинская, 5/7

e-mail: GaskarovVD@gumrf.ru

INFORMATION ABOUT THE AUTHORS

Nyrkov, Anatoliy P. — Grand PhD of Technical
Sciences, professor

Admiral Makarov State University of Maritime and
Inland Shipping

5/7 Dvinskaya str., St. Petersburg, 198035, Russian
Federation

e-mail: kaf.koib@gmail.com, nyrkowap@gumrf.ru

Korotkov Vitaliy V. — Associate Professor of
Department of Integrated Information Security,
Admiral Makarov State University of Maritime and
Inland Shipping

5/7 Dvinskaya str., St. Petersburg, 198035, Russian
Federation

e-mail: korotkovvv@gumrf.ru

Gaskarov Vagiz D. — Grand PhD of Technical
Sciences, professor

Admiral Makarov State University of Maritime and
Inland Shipping

5/7 Dvinskaya str., St. Petersburg, 198035, Russian
Federation

e-mail: GaskarovVD@gumrf.ru

Статья поступила в редакцию 18 июля 2025 г.

Received: July 18, 2025.