

## Метод повышения степени автономности сложных распределенных вычислительных систем

И. В. Сеницын\*, Ю. А. Воронцов, Е. К. Михайлова

ФГБОУ ВО «МИРЭА – Российский технологический университет», г. Москва, Российская Федерация

Адрес: 119454, Российская Федерация, г. Москва, пр. Вернадского, д. 78

\* sinicyn\_i@mirea.ru

### Аннотация

В данной работе рассматривается проблема повышения автономности функционирования кластерных распределенных вычислительных систем (КРВС). Целью данной исследовательской работы является повышение автономности функционирования кластерных распределенных вычислительных систем. В соответствии с поставленной целью, в работе решается научная задача – на основании анализа применения существующих кластерных распределенных вычислительных систем, их архитектур и особенностей, а также проведенного анализа тенденций развития отечественных разработок в этой области разработать методiku определения и задания основных эксплуатационных требований к перспективным кластерным распределенным вычислительным системам, обеспечивающих повышение их автономности. Первоначально в работе описывается текущее состояние вопроса автономности систем, включая рассмотрение существующих критериев соответствия компьютерной системы понятию автономности, а также классификация систем по степени их автономности, определяется необходимый перечень задач автономного функционирования. На основании этого производится построение формализованного аппарата, выделяются 2 группы задач, необходимых для решения вопроса автономного функционирования КРВС, даётся определение автономной компьютерной системы через вероятность работоспособности и функциональности технического, программного и информационного обеспечения, а также определение элементарного и сложного объекта. Представляется функция признаков, влияющая на выполнение целевой задачи сложного объекта и характеризующая интенсивность неисправностей и функциональных отказов в единицу времени. Вводится функция коэффициента автономности элементарного объекта, на основании которой определяется функция коэффициента автономности сложного объекта. На основании введенных определений и функций реализуется формализованное решение задачи повышения автономности путем максимизации комплексного показателя автономности, на основе прогнозирования периода автономного выполнения задач без внешнего вмешательства в процесс его функционирования.

**Ключевые слова:** автономность компьютерных систем, кластерные распределенные вычислительные системы, функциональная автономность, задачи функционирования КРВС, интервал автономности, комплексный показатель автономности

**Конфликт интересов:** авторы заявляют об отсутствии конфликта интересов.

**Для цитирования:** Сеницын И. В., Воронцов Ю. А., Михайлова Е. К. Метод повышения степени автономности сложных распределенных вычислительных систем // Современные информационные технологии и ИТ-образование. 2022. Т. 18, № 4. С. 774-780. doi: <https://doi.org/10.25559/SITITO.18.202204.774-780>

© Сеницын И. В., Воронцов Ю. А., Михайлова Е. К., 2022



Контент доступен под лицензией Creative Commons Attribution 4.0 License.  
The content is available under Creative Commons Attribution 4.0 License.



## Method of Increasing the Autonomy of Complex Distributed Computing Systems

I. V. Sinitsyn\*, Yu. A. Vorontsov, E. K. Mikhailova

MIREA – Russian Technological University, Moscow, Russian Federation

Address: 78 Vernadsky Ave., Moscow 119454, Russian Federation

\* sinicy\_n\_i@mirea.ru

### Abstract

This paper considers the problem of increasing the autonomy of the functioning of cluster distributed computing systems (CDCS). The goal of this research work is to increase the autonomy of functioning of cluster distributed computing systems. In accordance with the goal, the article solves the following scientific task: taking into account the analysis of the application of existing cluster distributed computing systems, their architectures and features, as well as the analysis of development trends of domestic developments in this area, develop a methodology for determining and setting the main operational requirements for promising cluster distributed computing systems that increase their autonomy. Initially, the paper describes the current state of the issue of autonomy of systems, including consideration of the existing criteria for the compliance of a computer system with the concept of autonomy, as well as the classification of systems according to the degree of their autonomy, and determines the necessary list of tasks for autonomous functioning. On the basis of this, a formalized apparatus is constructed, 2 groups of tasks are identified that are necessary to solve the issue of autonomous functioning of the CDCS, a definition of an autonomous computer system is given through the probability of operability and functionality of technical, software and information support, as well as the definition of an elementary and complex object. A feature function affecting the performance of the target task of a complex object and characterizing the intensity of malfunctions and functional failures in a unit of time is presented. The function of the coefficient of autonomy of an elementary object is introduced, on the basis of which the function of the coefficient of autonomy of a complex object is determined. On the basis of the introduced definitions and functions the formalized solution of the task of increasing autonomy is realized by maximizing the complex indicator of autonomy, based on predicting the period of autonomous task execution without external interference in the process of its functioning.

**Keywords:** autonomy of computer systems, cluster distributed computing systems, functional autonomy, tasks of functioning of the CDCS, interval of autonomy, complex indicator of autonomy

**Conflict of interests:** The authors declare no conflict of interest.

**For citation:** Sinitsyn I.V., Vorontsov Yu.A., Mikhailova E.K. Method of Increasing the Autonomy of Complex Distributed Computing Systems. *Modern Information Technologies and IT-Education*. 2022;18(4):774-780. doi: <https://doi.org/10.25559/SITITO.18.202204.774-780>



## Введение

На текущий момент для решения задач, подразумевающих обработку больших объемов данных активно применяются кластерные распределенные вычислительные системы (КРВС), позволяющие производить параллельную обработку подмассивов исходных данных для ускорения их обработки. При этом при потере работоспособности одним из узлов, возможно возникновение проблем с работоспособностью всей системы. Следовательно, необходимо повышение автономности каждого узла и системы в целом.

## Текущее состояние вопроса автономности

Вопрос автономности изучается на протяжении длительного времени, как развитие автоматизированных систем, способных поддерживать функциональную работоспособность на протяжении длительного времени. Существуют наборы критериев, согласно которым компьютерная система будет считаться автономной. Например, такой набор предложила компания IBM.

Автономная компьютерная система – компьютерная система, отвечающая следующим критериям:

1. «Знание самой себя» – компоненты системы должны обладать самоидентификацией. Обусловлено это тем, что система существует на нескольких уровнях, следовательно, она должна знать всё о своих компонентах, их текущем статусе, возможностях и связях с другими системами, от которых она зависит. На основании этого система должна также знать об объеме доступных её ресурсов, которыми она может управлять.
2. Система должна иметь возможность управления своей конфигурацией, перестраивать её при изменениях во внешней среде. При этом, при изменении внешних условий переконфигурация должна совершаться автоматически.
3. Система постоянно находится в процессе оптимизации своей внутренней структуры и процедур для достижения заранее установленных целей системы. Отслеживаются не только процедуры целиком, но и их части, для оптимизации работы.
4. Система должна иметь возможность обнаружения нарушений в процессе выполнения стандартной деятельности системы и нахождения альтернативного способа использования ресурсов, и в результате чего производить переконфигурацию.
5. Система должна быть способной сама себя защитить, следовательно, иметь способы обнаружения, идентификации и защиты от различного рода негативных внешних воздействий для сохранения своей стабильности.
6. Компьютерная система должна получать информацию о внешней среде, особенно в контексте влияния её собственной деятельности на внешнюю среду, и действовать соответственно, что означает быть адаптивной.
7. Автономная компьютерная система является открытой системой.
8. Система скрывает внутреннюю сложность предоставляе-

мых ресурсов. Это необходимо для уменьшения разрыва между требованиями пользователя, логики (программной части) и технологической реализацией системы.

Данные критерии были разработаны в рамках одноименного проекта «Автономные компьютерные системы». Целью проекта было создание систем, способных работать в автоматическом режиме. Это можно считать попыткой создания распределенных компьютерных систем, способных к самостоятельному реагированию на внезапные ситуации. Предполагается, что подобные системы производят постоянную самооптимизацию и контроль своего состояния и состояния окружающей среды.

Помимо критериев оценки автономности компьютерных систем, осуществлялись и другие исследования, нацеленные на формирование требований и подходов к созданию автономных систем. Например, для более четкого разграничения классов систем, была разработана обобщенная классификация автономности в различных сферах применения. В ней существенные аспекты автоматизации разделены на четыре класса:

1. Функциональные автоматизированные системы могут автономно выполнять ограниченные, четко сформулированные задачи, как например, автоматическая парковка, приземление или полностью автоматизированное производство одного определенного продукта. Эти системы не могут обучаться в процессе эксплуатации; кооперация с другими системами ограничена лишь обменом контекстной информацией.
2. Системы, ориентированные на выполнение заданий, могут решать незапланированную последовательность знаковых и выполнимых задач независимо от имеющейся ситуации. Это может быть, например, автономная езда на машине по трассе или работа одной линии метро. При реализации этого могут играть роль различные критерии оптимизации, как например, минимизация времени или ресурсных затрат. Расчеты для планирования и оптимизации производятся системой динамически в момент выполнения задачи. Эти системы также не могут обучаться в процессе работы; кооперация с другими системами ограничена обменом информацией о контексте задачи и о самой системе.
3. Коллаборативными системами являются такие системы как роботы, группы из транспортных средств или летательных дронов, которые двигаются в определенном порядке или кооперируются друг с другом во избежание столкновений. В целях выполнения своей миссии, такие системы в состоянии взаимодействовать с другими системами и людьми, а также динамически согласовывать друг с другом свое восприятие, интерпретацию, цели, планы и действия. Эти системы обмениваются со своими партнерами релевантной информацией, но, все же, не являются обучаемыми.
4. Самодостаточные (аутопоэтические) системы – это системы, которые в состоянии самостоятельно развивать свои восприятие, интерпретацию, действия и способности совместной работы, а также обмениваться этим с другими системами (включая воспроизведение изученного поведения). Эти системы демонстрируют таким образом чело-



векоподобное поведение и, на сегодняшний день, все еще не реализованы на практике. Способность к бесконтрольному обучению является основной характеристикой этого класса систем [1-11].

Проведенный анализ существующего научно-методического аппарата оценки функционирования современных сложных вычислительных систем целевого назначения, в отечественной и в зарубежной литературе показал, что вопросы увеличения автономности функционирования кластерных распределенных вычислительных систем и отдельные аспекты формирования требований к таким системам различного целевого назначения, а также задача увеличения автономности за счет устойчивости функционирования, исследованы недостаточно [12-20].

Таким образом, в настоящее время, актуальным является решение задачи увеличения автономности функционирования сложных вычислительных систем в общем и кластерных распределенных систем, в частности.

Учитывая вышеизложенное, **целью** данной исследовательской работы является повышение автономности функционирования кластерных распределенных вычислительных систем.

В соответствии с поставленной целью, в работе решается **научная задача** – на основании анализа применения существующих кластерных распределенных вычислительных систем, их архитектур и особенностей, а также проведенного анализа тенденций развития отечественных разработок в этой области разработать методику определения и задания основных эксплуатационных требований к перспективным кластерным распределенным вычислительным системам, обеспечивающих повышение их автономности.

## Решение задачи повышения автономности

Решение указанных вопросов требует создания теоретической основы, т.е. построения формализованного аппарата для получения требуемых количественных оценок и поиска оптимальных решений, и определения методологии исследований с помощью этого аппарата. С этой целью при проведении исследований развивается один подход, связанный с использованием метода оценки информационных процессов для описания функционирования исследуемой системы<sup>1</sup>. Для решения вопроса автономности функционирования КРВС необходимо решить 2 группы задач:

1. Задачи обеспечения функционирования КРВС;
2. Целевые задачи.

К задачам обеспечения функционирования КРВС относятся:

- программное обеспечение контроля состояния и управления вычислительными системами;
- техническое обеспечение контроля состояния и управления вычислительными системами;
- информационное обеспечение контроля состояния и

управления вычислительными системами;

- автоматизация распределения системных ресурсов КРВС;
  - обеспечение интерфейсов взаимодействия человек-машина и машина-машина;
  - обеспечение безопасности информации;
  - обеспечение задач по восстановлению и сохранению.
- К целевым задачам функционирования КРВС относятся:
- решение задач, связанных с технологиями Big Data;
  - решение задач, которые хорошо распараллеливаются;
  - решение задач искусственного интеллекта;
  - решение задач, связанных с реализацией самообучаемых автоматизированных систем обучения для подготовки технических специалистов.

Программное, техническое, информационное обеспечение контроля состояния и управления вычислительными системами заключается в создании автономной системы мониторинга КРВС.

Переход от содержательного описания научной задачи к формальной ее постановке требует учета технико-экономических характеристик процесса создания КРВС, обоснования определения совокупности показателей автономности системы, концепции выбора искомого решения, формирования системы исходных данных и установление ограничений, накладываемых на область допустимых решений [21-25].

Автономность КРВС является сложным комплексным понятием и с дальнейшим развитием компьютерных технологий определяющую роль в ее повышении имеет КРВС, т.к. в ней происходит сбор, хранение информации, выработка управляющих воздействий и решение целевых задач. В дальнейшем будем рассматривать пути повышения автономности КРВС через повышение автономности отдельных структурных элементов.

Автономность компьютерной системы – есть вероятность того, что при функционировании системы в течение некоторого периода времени она способна выполнять определенные действия полностью самостоятельно без внешнего управляющего изменения ее технической, программной и информационной структуры в этот промежуток времени. Она является функцией вероятностей работоспособности и функциональности технического, программного и информационного обеспечения и определяется как:

$$P_A = P_{\Pi} * P_T * P_{\text{и}} \quad (1)$$

При этом предполагается, что потеря автономности технического, программного или информационного обеспечения приводит к потере автономности КРВС в целом.

Где:

$P_T = P_T (P_{\text{тк}}, P_{\text{г}}, P_{\text{в}}, P_{\text{мт}}, P_{\text{мп}}, P_{\text{ми}}, P_{\text{д}}, t)$  – автономность технического обеспечения (ТО) КРВС, вероятность работоспособности и функциональности технического обеспечения КРВС;

$P_{\Pi} = P_{\Pi} (P_{\text{пк}}, P_{\text{г}}, P_{\text{в}}, P_{\text{мт}}, P_{\text{мп}}, P_{\text{ми}}, P_{\text{д}}, t)$  – автономность программного обеспечения (ПО) КРВС, вероятность работоспособности и функциональности программного обеспечения КРВС;

$P_{\text{и}} = P_{\text{и}} (P_{\text{ик}}, P_{\text{г}}, P_{\text{в}}, P_{\text{мт}}, P_{\text{мп}}, P_{\text{ми}}, P_{\text{д}}, t)$  – автономность информацион-

<sup>1</sup> ГОСТ Р ИСО/МЭК 9126-93 Информационная технология. Оценка программной продукции. Характеристики качества и руководства по их применению: государственный стандарт РФ: издание официальное: утвержден и введен в действие Постановлением Госстандарта России от 28 декабря 1993 г. № 267: введен впервые: дата введения 1994-07-01. М. : Госстандарт России, 1994; IEEE Recommended Practice for Software Design Descriptions. In: IEEE Std 1016-1998. P. 1-23. IEEE Computer Society, 1998. doi: <https://doi.org/10.1109/IEEESTD.1998.88828>



ного обеспечения (ИО) КРВС, вероятность работоспособности и функциональности информационного обеспечения КРВС, где

$P_T$  – вероятность работоспособности при температурном воздействии;

$P_V$  – вероятность работоспособности при вибрационном воздействии;

$P_D$  – вероятность работоспособности при деструктивном воздействии;

$P_{MT}$  – вероятность работоспособности при модернизации технического обеспечения;

$P_{MP}$  – вероятность работоспособности при модернизации программного обеспечения;

$P_{MI}$  – вероятность работоспособности при модернизации информационного обеспечения.

Параметры, влияющие на автономность КРВС можно разделить на две группы:

- конструкционные, т.е. те границы значений, которых закладываются в объект при его изготовлении согласно существующих требований –  $P_{TK}, P_{PK}, P_{IK}$ ;
- внешние, которые в значительной степени могут влиять на работоспособность и функциональность КРВС –  $P_T, P_V, P_D, P_{MT}, P_{MP}, P_{MI}, P_D$ .

Имеем сложный объект – КРВС, выполняющая определенную целевую задачу и состоящий из множества элементарных объектов  $O_i$  (где  $i=1$  до  $n$ ),  $O_i \in G$  ( $G$  – множество элементарных объектов).

Элементарный объект  $O_i$  – объект, не имеющий в составе других любых объектов, принимающих независимое участие в выполнении целевой задачи сложного объекта КРВС. Элементарные объекты в нашем случае – объекты технического обеспечения, программные и информационные комплексы КРВС. Каждый  $O_i$  (где  $i=1 \dots n$  – номер элементарного объекта) характеризуется функцией признаков:

$\lambda_{ij}(t)$  – функция  $j$  (где  $j=1 \dots m$  – номер признака  $j \in W$  ( $W$  – множество признаков)) признака для  $i$  элементарного объекта, влияющая на выполнение целевой задачи сложного объекта и характеризующая интенсивность неисправностей и функциональных отказов в единицу времени.

Представим функцию признаков в виде:

$$\lambda_{ij}(t) = d_{ij}(t) + r_{ij}(t), \quad (2)$$

где:

$d_{ij}(t)$  – детерминантная составляющая функции признаков;

$r_{ij}(t)$  – стохастическая составляющая функции признаков.

Введем функцию коэффициента автономности элементарного объекта

$$K_{Ai}(t) = \frac{N_{O_i}(t)}{N_{S_i}(t)}, \quad (3)$$

Где:

$N_{O_i}(t)$  – функция количества простых операций реализуемых на  $O_i$  элементарном объекте;

$N_{S_i}(t)$  – число всех простых операций необходимых для выполнения целевой задачи  $O_i$  элементарного объекта.

Определим функцию коэффициента автономности сложного объекта А:

$$K_A(t) = \frac{1}{n} * \sum_{i=1}^n \frac{N_{O_i}(t)}{N_{S_i}(t)}, \quad (4)$$

$K_A(t)$  – весовая функция, учитывающая функциональный вклад в прогноз:

$$0 \leq K_A(t) \leq 1 \quad (5)$$

Введем функцию интервала автономности  $i$  элементарного объекта:

$$T_i(t) = 1 / \sum_{j=1}^m \lambda_{ij}(t), \quad (6)$$

Определим функцию интервала автономности сложного объекта.

$T(t) = \min(T_i(t))$  – интервал автономности которого имеет физический смысл – прогноз продолжительности времени, которое сложный объект способен, выполнять целевые задачи без управляющего внешнего воздействия.

Тогда задачу повышения автономности сложного объекта целесообразно привести к решению нахождения максимального значения комплексного показателя автономности:

$$\int_0^{T_A} K_A(t) T(t) dt \rightarrow \max, \quad (7)$$

при  $\lambda_i = (\lambda_{i1}, \dots, \lambda_{im})$  определяющей интенсивность отказов для  $i$ -го объекта по следующим признакам:

$\lambda_{i1}$  – при температурном воздействии на КРВС;

$\lambda_{i2}$  – при вибровоздействии на КРВС;

$\lambda_{i3}$  – при ошибках программного обеспечения КРВС;

$\lambda_{i4}$  при ошибках информационного обеспечения КРВС;

$\lambda_{i5}$  при ошибках технического обеспечения КРВС;

$\lambda_{i6}$  при преднамеренных деструктивных воздействиях на КРВС.

## Заключение

Представленная задача максимизации позволяет повысить автономность сложных вычислительных систем, с учетом воздействия внешних и внутренних факторов. Помимо рассмотренных параметров, могут быть учтены и дополнительные, актуальные в каждой конкретной области применения кластерных распределенных вычислительных систем.

## References

- [1] Damm W., Kalmar R. Autonome Systeme – Fähigkeiten und Anforderungen. *Informatik Spektrum*. 2017;40(5):400-408. doi: <https://doi.org/10.1007/s00287-017-1063-0>
- [2] Kortenkamp D., Bonasso R.P., Ryan D., Schreckenghost D. Traded Control with Autonomous Robots as Mixed Initiative Interaction. In: Papers from the 1997 AAAI Spring Symposium. AAAI; 1997. p. 89-94. Available at: <https://cdn.aaai.org/Symposia/Spring/1997/SS-97-04/SS97-04-016.pdf> (accessed 11.08.2022).
- [3] Nicolescu M.N., Maticiћ M.J. A hierarchical architecture for behavior-based robots. In: Proceedings of the first international joint



- conference on Autonomous agents and multiagent systems: part 1 (AAMAS '02). New York, NY, USA: Association for Computing Machinery; 2002. p. 227-233. doi: <https://doi.org/10.1145/544741.544798>
- [4] Nowostawski M., Purvis M. The Concept of Autonomy in Distributed Computation and Multi-agent Systems. In: 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'07). Fremont, CA, USA: IEEE Computer Society; 2007. p. 420-423. doi: <https://doi.org/10.1109/IAT.2007.23>
- [5] Pellkofer M., Luetzeler M., Dickmanns E.D. Interaction of perception and gaze control in autonomous vehicles. In: Casasent D.P., Hall E.L. (eds.) Proceedings of SPIE. Vol. 4572. Intelligent Robots and Computer Vision XX: Algorithms, Techniques, and Active Vision. SPIE; 2001. p. 219-230. doi: <http://dx.doi.org/10.1117/12.444186>
- [6] Tianfield H., Unland R. Towards autonomic computing systems. *Engineering Applications of Artificial Intelligence*. 2004;17(7):689-699. doi: <https://doi.org/10.1016/j.engappai.2004.08.029>
- [7] Neema S., Bapty T., Shetty S., Nordstrom S. Autonomic fault mitigation in embedded systems. *Engineering Applications of Artificial Intelligence*. 2004;17(7):711-725. doi: <https://doi.org/10.1016/j.engappai.2004.08.031>
- [8] Sterritt R. Autonomic networks: engineering the self-healing property. *Engineering Applications of Artificial Intelligence*. 2004;17(7):727-739. doi: <https://doi.org/10.1016/j.engappai.2004.08.028>
- [9] Jin X., Liu J. Characterizing autonomic task distribution and handling in grids. *Engineering Applications of Artificial Intelligence*. 2004;17(7):809-823. doi: <https://doi.org/10.1016/j.engappai.2004.08.034>
- [10] Fisher M., Ferrando A., Cardoso R.C. Increasing confidence in autonomous systems. In: Proceedings of the 5th ACM International Workshop on Verification and mOnitoring at Runtime EXecution (VORTEX 2021). New York, NY, USA: Association for Computing Machinery; 2021. p. 1-4. doi: <https://doi.org/10.1145/3464974.3468452>
- [11] McKee D.W., Clement S.J., Almutairi J., Xu J. Survey of advances and challenges in intelligent autonomy for distributed cyber-physical systems. *CAAI Transactions on Intelligence Technology*. 2018;3(2):75-82. doi: <https://doi.org/10.1049/trit.2018.0010>
- [12] Shang J., Chen P., Wang Q., Lu L. Information System Reliability Quantitative Assessment Method and Engineering Application. In: 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C). Lisbon, Portugal: IEEE Computer Society; 2018. p. 182-186. doi: <https://doi.org/10.1109/QRS-C.2018.00044>
- [13] Talmale G., Shrawankar U. Cluster Based Real Time Scheduling for Distributed System. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*. 2021;10(2):137-156. doi: <https://doi.org/10.14201/ADCAIJ2021102137156>
- [14] Li J., Woodside M., Chinneck J., Litiou M. Adaptive Cloud Deployment Using Persistence Strategies and Application Awareness. *IEEE Transactions on Cloud Computing*. 2017;5(2):277-290. doi: <https://doi.org/10.1109/TCC.2015.2409873>
- [15] Häring I. Models for Hardware and Software Development Processes. In: Technical Safety, Reliability and Resilience. Singapore: Springer; 2021. p. 179-192. doi: [https://doi.org/10.1007/978-981-33-4272-9\\_10](https://doi.org/10.1007/978-981-33-4272-9_10)
- [16] Bouquet F., Grandpierre C., Legear B., Peureux F. A test generation solution to automate software testing. In: Proceedings of the 3rd international workshop on Automation of software test (AST '08). New York, NY, USA: Association for Computing Machinery; 2008. p. 45-48. doi: <https://doi.org/10.1145/1370042.1370052>
- [17] Thayer R.H. Software System Engineering: A Tutorial. *Computer*. 2002;35(4):68-73. doi: <https://doi.org/10.1109/MC.2002.993773>
- [18] Yamada S., Tamura Y. Software Reliability. In: OSS Reliability Measurement and Assessment. Springer Series in Reliability Engineering. Cham: Springer; 2016. p. 1-13. doi: [https://doi.org/10.1007/978-3-319-31818-9\\_1](https://doi.org/10.1007/978-3-319-31818-9_1)
- [19] Pavlov N., Iliev A., Rahnev A., Kyurkchiev N. Some Transmuted Software Reliability Models. *Journal of Mathematical Sciences and Modelling*. 2019;2(1):64-70. doi: <http://dx.doi.org/10.33187/jmsm.434277>
- [20] Pham H. System Software Reliability. In: Springer Series in Reliability Engineering. London: Springer; 2006. 440 p. doi: <https://doi.org/10.1007/1-84628-295-0>
- [21] Pham H. A new software reliability model with Vtub-shaped fault-detection rate and the uncertainty of operating environments. *Optimization*. 2014;63(10):1481-1490. doi: <https://doi.org/10.1080/02331934.2013.854787>
- [22] In Hong Chang, Hoang Pham, Seung Woo Lee, Kwang Yoon Song. A testing-coverage software reliability model with the uncertainty of operating environments. *International Journal of Systems Science: Operations & Logistics*. 2014;1(4):220-227. doi: <https://doi.org/10.1080/23302674.2014.970244>
- [23] Hidaka S., Hu Z., Litiou M., Liu L., Martin P., Peng X., Wang G., Yu Y. Design and Engineering of Adaptive Software Systems. In: Yu Y., Bandara A., et al. (eds.) Engineering Adaptive Software Systems. Singapore: Springer; 2019. p. 1-33. doi: [https://doi.org/10.1007/978-981-13-2185-6\\_1](https://doi.org/10.1007/978-981-13-2185-6_1)
- [24] Narendra K.S., Parthasarathy K. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*. 1990;1(1):4-27. doi: <https://doi.org/10.1109/72.80202>
- [25] Timofeev R.A., Shlychkov V.V., Nestulaeva D.R. Methods of economic reliability assessment for industrial enterprise in the market economy conditions. *SHS Web of Conferences*. 2017;35:01125. doi: <https://doi.org/10.1051/shsconf/20173501125>

Поступила 11.08.2022; одобрена после рецензирования 19.10.2022; принята к публикации 16.11.2022.

Submitted 11.08.2022; approved after reviewing 19.10.2022; accepted for publication 16.11.2022.



**Об авторах:**

**Синицын Иван Васильевич**, доцент кафедры математического обеспечения и стандартизации информационных технологий Института информационных технологий, ФГБОУ ВО «МИРЭА – Российский технологический университет» (119454, Российская Федерация, г. Москва, пр. Вернадского, д. 78), кандидат технических наук, **ORCID: <https://orcid.org/0000-0002-4302-1616>**, sinicyn\_i@mirea.ru

**Воронцов Юрий Алексеевич**, ассистент кафедры математического обеспечения и стандартизации информационных технологий Института информационных технологий, ФГБОУ ВО «МИРЭА – Российский технологический университет» (119454, Российская Федерация, г. Москва, пр. Вернадского, д. 78), **ORCID: <https://orcid.org/0000-0002-1732-6002>**, voroncov\_yu@mirea.ru

**Михайлова Евгения Константиновна**, ассистент кафедры математического обеспечения и стандартизации информационных технологий Института информационных технологий, ФГБОУ ВО «МИРЭА – Российский технологический университет» (119454, Российская Федерация, г. Москва, пр. Вернадского, д. 78), **ORCID: <https://orcid.org/0000-0002-0105-9325>**, mihajlova\_e@mirea.ru

*Все авторы прочитали и одобрили окончательный вариант рукописи.*

**About the authors:**

**Ivan V. Sinitsyn**, Associate Professor of the Software and IT-standard Department, Institute of Information Technology, MIREA – Russian Technological University (78 Vernadsky Ave., Moscow 119454, Russian Federation), Cand. Sci. (Eng.), **ORCID: <https://orcid.org/0000-0002-4302-1616>**, sinicyn\_i@mirea.ru

**Yuriy A. Vorontsov**, Assistant Professor of the Software and IT-standard Department, Institute of Information Technology, MIREA – Russian Technological University (78 Vernadsky Ave., Moscow 119454, Russian Federation), **ORCID: <https://orcid.org/0000-0002-1732-6002>**, voroncov\_yu@mirea.ru

**Evgenia K. Mikhailova**, Assistant Professor of the Software and IT-standard Department, Institute of Information Technology, MIREA – Russian Technological University (78 Vernadsky Ave., Moscow 119454, Russian Federation), **ORCID: <https://orcid.org/0000-0002-0105-9325>**, mihajlova\_e@mirea.ru

*All authors have read and approved the final manuscript.*

