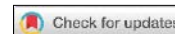


# ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

## INFORMATION TECHNOLOGIES



УДК [004.4 + 004.9]-047.58  
<https://doi.org/10.37661/1816-0301-2021-18-1-84-95>

Оригинальная статья  
Original Paper

### Оценка ожидаемой надежности прикладных программных средств для компьютерных информационных систем

С. М. Боровиков<sup>✉</sup>, В. О. Казючиц, В. В. Хорошко, С. С. Дик, К. И. Клинов

Белорусский государственный университет информатики и радиоэлектроники,  
ул. П. Бровки, 6, 220013, Минск, Беларусь

<sup>✉</sup>E-mail: [bsm@bsuir.by](mailto:bsm@bsuir.by)

**Аннотация.** Надежность компьютерной информационной системы во многом определяется надежностью разрабатываемого прикладного программного средства. В качестве показателя надежности прикладного программного средства рассматривается интенсивность отказов его компьютерной программы. Для определения ожидаемой надежности прикладного программного средства, планируемого к разработке (до написания кода компьютерной программы), предлагается модель, в которой используются некоторые параметры будущей компьютерной программы, данные о влиянии на ее надежность различных факторов и предполагаемое в дальнейшем тестирование компьютерной программы. Модель учитывает отрасль применения программного средства и быстродействие процессора компьютера. Анализируется процесс получения параметров модели. С помощью предложенной модели можно определить прогнозную интенсивность отказов планируемой к разработке прикладной компьютерной программы, а далее и надежность компьютерной информационной системы в целом. При необходимости могут быть разработаны организационно-технические мероприятия по обеспечению требуемого уровня надежности компьютерной информационной системы.

**Ключевые слова:** интенсивность отказов компьютерной программы, модель надежности, ранние этапы разработки, прогнозируемая плотность ошибок, ожидаемое число ошибок

**Благодарность.** Работа проводилась в рамках договора № Ф20МВ-021 на выполнение научно-исследовательской работы в соответствии с решением научного совета Белорусского республиканского фонда фундаментальных исследований (протокол № 1 от 22.04.2020).

**Для цитирования.** Оценка ожидаемой надежности прикладных программных средств для компьютерных информационных систем / С. М. Боровиков [и др.] // Информатика. – 2021. – Т. 18, № 1. – С. 84–95. <https://doi.org/10.37661/1816-0301-2021-18-1-84-95>

**Конфликт интересов.** Авторы заявляют об отсутствии конфликта интересов.

## Assessment of expected reliability of applied software for computer-based information systems

Sergei M. Borovikov<sup>✉</sup>, Vladislav O. Kaziuchyts, Vitaliy V. Khoroshko, Sergei S. Dick, Konstantin I. Klinov

*Belarusian State University of Informatics and Radioelectronics,  
st. P. Brovki, 6, 220013, Minsk, Belarus*

<sup>✉</sup>E-mail: bsm@bsuir.by

**Abstract.** The reliability of computer-based information systems is largely determined by the reliability of the developed application software. The failure rate of its computer program is considered as an indicator of the reliability of the application software. To determine the expected reliability of the application software planned for the development (until writing the code of a program), the model is proposed that uses some parameters of the future computer program, data on the influence of various factors on its reliability, and further testing of the program. The model takes into account the field of software application and computer processor performance. The process of model parameters obtaining is analyzed. It is possible by use of proposed model to determine the predicted failure rate of the planned application computer program, and then the reliability of the computer-based information system as a whole. If necessary, the measures can be developed to ensure the required level of reliability of the computer-based information system.

**Keywords:** failure rate of computer program, reliability model, early stages of development, predicted error density, expected number of errors

**Acknowledgements.** This work was carried out in the framework of contract no. F20MV-021 for the implementation of research work in accordance with the decision of the Scientific Council of the Belarusian Republican Foundation for Basic Research (protocol no. 1 of 04.22.2020).

**For citation.** Borovikov S. M., Kaziuchyts V. O., Khoroshko V. V., Dick S. S., Klinov K. I. Assessment of expected reliability of applied software for computer-based information systems. *Informatics*, 2021, vol. 18, no. 1, pp. 84–95 (in Russian). <https://doi.org/10.37661/1816-0301-2021-18-1-84-95>

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

**Введение.** В компьютерных информационных системах можно выделить три основные части:

- аппаратные средства ЭВМ и оборудование телекоммуникационных сетей;
- программное обеспечение;
- технический персонал и пользователи.

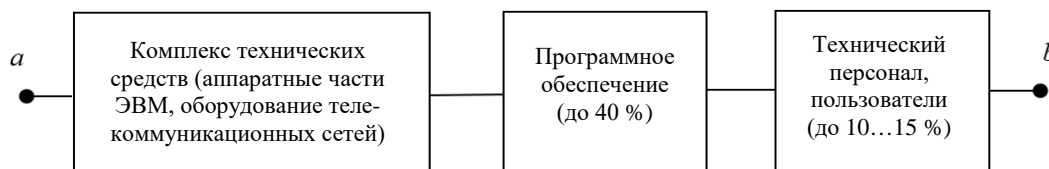
Аппаратные средства ЭВМ включают устройства ввода (клавиатуру, сканер и т. д.), устройство обработки данных (центральный процессор), устройства хранения информации (модули оперативной памяти, жесткие диски и т. д.) и вывода (принтеры, экраны мониторов и др.). Телекоммуникационная связь позволяет организациям связывать компьютерные системы в эффективные сети.

Программное обеспечение включает совокупность системных и прикладных компьютерных программ и определяет команды, которые выполняют аппаратные средства. Прикладные программы, в свою очередь, определяют команды компьютеру, которые следует выполнять в ответ на действия человека, работающего за компьютером.

Технический персонал компьютерных информационных систем включает всех, кто программирует и обслуживает компьютерную информационную систему, следит за работоспособностью ее аппаратных средств и оборудования телекоммуникационных сетей. Пользователи – это люди, которые применяют компьютерную информационную систему для получения информации или решения других задач.

Структурная схема надежности (ССН) компьютерной информационной системы, в которой отсутствует избыточность, изображена на рисунке. Эта схема является моделью надежности компьютерной информационной системы. Если хотя бы одна из частей системы неработоспособна, то при переходе из точки *a* в точку *b* на схеме отсутствует замкнутый путь. Отсутствие

замкнутого пути означает, что отказ любой из указанных частей приводит к отказу системы в целом. Это символизируется разрывом цепи на ССН компьютерной информационной системы. Функциональные части системы на схеме обозначены прямоугольниками, в которые записан примерный вклад соответствующей функциональной части в ненадежность компьютерной информационной системы [1, 2].



Структурная схема надежности компьютерной информационной системы

При оценке ожидаемой надежности проектируемых компьютерных информационных систем важно учесть надежность всех трех ее составных функциональных частей.

**Актуальность разработки.** Методы оценки надежности комплекса технических средств достаточно хорошо разработаны. Для ознакомления с подходами и методиками расчета надежности аппаратных частей ЭВМ и оборудования телекоммуникационных сетей можно обратиться к зарубежным источникам [3–5], российскому справочнику [6] или отечественным работам [7, 8]. Достигнут определенный уровень в разработке подходов к оценке надежности технического персонала и пользователей при их взаимодействии с комплексом технических средств и программным обеспечением. С методологией и описанием этих подходов можно ознакомиться в зарубежной [9, 10] и российской [11, 12] литературе.

В настоящее время проведены серьезные исследования в области оценки ожидаемой надежности прикладных компьютерных программ, входящих в состав программного обеспечения компьютерных информационных систем, однако предлагаемые модели для оценки надежности компьютерных программ носят ограниченный характер. Под моделью надежности компьютерной программы понимается, как правило, математическая модель, построенная для оценки зависимости надежности программного обеспечения от некоторых определенных параметров. Значения этих параметров либо предполагаются известными, либо могут быть измерены в ходе наблюдений или экспериментального исследования процесса функционирования компьютерных программ. С начала 1970-х гг. было создано более 200 моделей, но вопрос о том, как количественно оценить надежность компьютерных программ, остается в основном нерешенным. Не существует отдельной модели, которая могла бы использоваться во всех ситуациях. Ни одна модель не является полной или даже представительной [13].

Большинство моделей надежности (оценочные и измерительные) исходят из того, что имеются определенные данные о результатах тестирования разработанной компьютерной программы после устранения в ней ошибок, вызванных нарушениями правил языка программирования [1, 14–16]. Прогнозирующие модели надежности, такие как модель Холстеда и модель Мотли – Брукса, основаны на экспертной оценке характеристик создаваемой компьютерной программы: размера, количества операторов, количества и типов переменных (операндов), характеристик интерфейсов и др. [1]. Эти модели позволяют прогнозировать ожидаемый уровень надежности планируемых к разработке компьютерных программ. Однако на практике в большинстве случаев возникает проблема с определением характеристик будущей прикладной компьютерной программы. Обычно перед разработкой программного средства для компьютерной информационной системы удается относительно четко указать функциональное назначение и область применения программного средства и спрогнозировать примерный объем реализующей его прикладной компьютерной программы в строках программного кода. Актуальным является вопрос, как оценить надежность прикладных компьютерных программ в таких случаях.

**Подход к оценке ожидаемой надежности программных средств.** Согласно ГОСТ 27.205–1997 «Надежность в технике. Проектная оценка надежности сложных систем с учетом технического и программного обеспечения и оперативного персонала. Основные положения» об ожидаемой надежности разрабатываемой компьютерной программы (программного средства) будем су-

дить по интенсивности проявления скрытых ошибок, оставшихся в компьютерной программе после ее разработки и тестирования. Такую интенсивность (обозначим ее через  $\lambda$ ) будем рассматривать в качестве интенсивности отказов компьютерной программы.

Большинство моделей надежности компьютерных программ исходят из того, что интенсивность отказов  $\lambda$  прямо пропорциональна числу оставшихся в программе ошибок [1, 16]:

$$\lambda \approx C_{\text{п}} \cdot N_{\text{ош}} = C_{\text{п}} \cdot F \cdot L, \quad (1)$$

где  $C_{\text{п}}$  – коэффициент пропорциональности, определяемый экспериментально;  $N_{\text{ош}}$  – число ошибок, оставшихся в компьютерной программе;  $F$  – плотность ошибок в программе (среднее число ошибок, приходящихся на строку кода:  $F = N_{\text{ош}} / L$ );  $L$  – объем компьютерной программы в исполняемых строках программного кода.

В работе [17] эксплуатационная интенсивность отказов  $\lambda_{\text{экс}}$  компьютерной программы в соответствии с выражением (1) представлена в виде равенства

$$\lambda_{\text{экс}} = C_{\text{п}} F_0 L = K_{\text{пр}} F_0 = K_{\text{пр}} \frac{N_0}{L}, \quad (2)$$

где  $F_0$  – плотность ошибок компьютерной программы после написания ее кода и устранения нарушений правил языка программирования;  $N_0$  – число ошибок (до выполнения тестирования);  $K_{\text{пр}}$  – коэффициент преобразования, показывающий, как начальное число ошибок  $N_0$  (или, что то же самое, плотность ошибок  $F_0$ ) трансформируется в эксплуатационную интенсивность отказов  $\lambda_{\text{экс}}$ .

На основе экспериментальных данных, приводимых в работе [17], для компьютерных программ различных областей применения получены значения коэффициента преобразования  $K_{\text{пр}}$  (табл. 1). Эти значения рассчитаны на основе выражения (2) как  $K_{\text{пр}} = \lambda_{\text{экс}} / F_0$ . Например, среднее значение коэффициента  $K_{\text{пр}} = 0,1 / 0,0094 \approx 10,6$ .

Таблица 1

Экспериментальные данные о надежности компьютерных программ

Область применения компьютерной программы	Среднее число исполняемых строк кода, приходящееся на одну программу $L$	Средняя плотность ошибок $F_0$ для одной программы, ошибка/строка кода	Средняя эксплуатационная интенсивность отказов $\lambda_{\text{экс}}$ , 1/ч	Значение $K_{\text{пр}}$ , строка кода/ч
Авиация	77 231,0	0,013	0,08	6,28
Мониторинг и обеспечение безопасности	85 420,5	0,009	0,0109	1,2
Телекоммуникации, мобильные электронные устройства	17 650,4	0,008	0,108	13,8
Управление производственными процессами	70 045,0	0,002	0,0076	3,8
Автоматизированные системы управления (в различных областях и сферах деятельности людей)	214 618,9	0,009	0,198	23
Инструменты разработки программ, моделирование, испытательные стенды, обучение	32 239,2	0,014	0,236*	16,9
Среднее	100 597,2	0,0094	0,1	10,6

\* – значение получено на основе интенсивности отказов, соответствующей этапу тестирования.

Приведенные в табл. 1 коэффициенты  $K_{пр}$  соответствуют случаю выполнения программ на компьютере с 32-разрядным процессором и быстродействием 10 млн операций в секунду (10 million instructions per second) [17].

Для компьютерных программ определенной области деятельности людей значение  $K_{пр}$  считают примерно постоянным независимо от этапа разработки или использования прикладной программы. Изменение значений интенсивности отказов компьютерной программы происходит из-за изменения числа оставшихся ошибок  $N_{ош}$  в программе или изменения плотности ошибок  $F$ .

Объем прикладных программ, разрабатываемых для компьютерных информационных систем, включает от ста тысяч до миллионов строк кода, и поэтому невыявленные на этапе тестирования ошибки в программах всегда существуют. Механизм преобразования между невыявленными ошибками  $N_{ош}$  и интенсивностью отказов программы  $\lambda$  основан на том, что оператор, имеющий скрытые ошибки, не приведет к отказу программы ни при каких обстоятельствах, пока он не будет выполнен, т. е. до тех пор, пока не повлияет на содержимое памяти или на управление компьютером.

Известно [17], что на вероятность возникновения отказа программы при ее использовании по функциональному назначению влияют три внешние характеристики среды выполнения программы:

- быстродействие процессора (производительность компьютера);
- степень изменчивости входных данных и состояний управления компьютером;
- рабочая нагрузка, которую эксплуатационная среда будет оказывать на компьютерную программу.

**Модель определения ожидаемой интенсивности отказов планируемой к разработке компьютерной программы.** Для количественной оценки ожидаемой начальной интенсивности отказов компьютерной программы  $\lambda_0$  используем модель, аналогичную выражению (2). Как показали экспериментальные данные [17], для рассматриваемых компьютерных программ при суммарной продолжительности их тестирования, составляющей примерно 40 % общего времени, отводимого на разработку, и использовании общепринятых методов тестирования интенсивность отказов компьютерных программ за счет выполнения тестирования уменьшалась примерно в 50 раз относительно их начальной интенсивности отказов  $\lambda_0$ , т. е.  $\lambda_0 / \lambda_{экс} \approx 50$ . Таким образом, можно записать равенство

$$\lambda_0 = 50 K_{пр} \frac{N_0}{L} = 50 K_{пр} F_0. \quad (3)$$

С другой стороны, интенсивность отказов компьютерной программы до начала тестирования с учетом быстродействия процессора компьютера может быть определена по модели Му-са [1]:

$$\lambda_0 = C V N_0 = 3600 C \frac{0,7R}{B} N_0 = 2520 C \frac{R}{B} N_0 = 2520 C V_{пик} N_0, \quad (4)$$

где  $C$  – коэффициент пропорциональности, называемый коэффициентом проявления ошибок компьютерной программы;  $V$  – средняя скорость выполнения команд компьютерной программы, представляющая собой возможное число прогонов компьютерной программы в течение одного часа;  $R$  – пиковое быстродействие процессора, указываемое производителем в технической документации, размерность «операций в секунду»;  $B$  – количество команд (операторов) в компьютерной программе; множитель 3600 – коэффициент перевода быстродействия процессора в размерность «операций в час»; множитель 0,7 – средний коэффициент, учитывающий реальное быстродействие процессора относительно пикового значения  $R$ ;  $V_{пик}$  – пиковая скорость выполнения команд компьютерной программы (скорость для пикового быстродействия процессора  $R$ ), определяемая отношением  $V_{пик} = R/B$ .

Примерное значение числа команд  $B$  можно определить следующим образом:

$$B = E_L L,$$

где  $E_L$  – коэффициент расширения кода (увеличения числа команд компьютерной программы относительно числа строк кода).

Согласно работе [18] будем считать, что одна строка кода компьютерной программы транслируется в 10 машинных команд, т. е.  $E_L = 10$ .

С помощью выражений (3), (4) и с учетом значений коэффициентов преобразования  $K_{пр}$ , приведенных в табл. 1, определены средние значения коэффициента проявления ошибок  $C$  для компьютерных программ различных областей применения (табл. 2).

Таблица 2

Значения коэффициентов для компьютерных программ различных областей применения

Область применения компьютерной программы	Коэффициент $C$ , 1/ошибка	Коэффициент $K_{\Sigma}$	Значение $A$ , ошибка / строка кода
1. Авиация	$1,246 \cdot 10^{-7}$	5,23	0,0128
2. Мониторинг и обеспечение безопасности	$0,238 \cdot 10^{-7}$	1,00	0,0092
3. Телекоммуникации, мобильные устройства	$2,738 \cdot 10^{-7}$	11,50	0,0078
4. Управление производственными процессами	$0,754 \cdot 10^{-7}$	3,17	0,0018
5. Автоматизированные системы управления	$4,563 \cdot 10^{-7}$	19,20	0,0085
6. Разработка программ, моделирование, обучение	$3,353 \cdot 10^{-7}$	14,10	0,0123
Среднее	$2,103 \cdot 10^{-7}$	8,83	0,0087

Представим коэффициент  $C$  в виде произведения

$$C = C_B \cdot K_{вх} \cdot K_n = C_B \cdot K_{\Sigma}, \quad (5)$$

а величины, входящие в выражение (5), обозначим следующим образом:  $C_B$  – базовое значение коэффициента проявления ошибок программы;  $K_{вх}$  – коэффициент увеличения интенсивности отказов, обусловленный изменчивостью входных данных;  $K_n$  – коэффициент увеличения интенсивности отказов, обусловленный рабочей нагрузкой на программу;  $K_{\Sigma}$  – суммарный коэффициент увеличения интенсивности отказов, обусловленный совместным действием изменчивости входных данных и рабочей нагрузки на компьютерную программу.

В качестве базового значения коэффициента проявления ошибок  $C_B$  будем рассматривать коэффициент  $C$  для компьютерных программ, предназначенных для мониторинга и обеспечения безопасности. Пользуясь табл. 2 (строка 2), можно записать:  $C_B = 0,238 \cdot 10^{-7}$  1/ошибка. Для программ этого назначения суммарный коэффициент увеличения интенсивности отказов принят равным  $K_{\Sigma} = 1,0$ . В табл. 2 (столбец 3) приведены рассчитанные значения коэффициента  $K_{\Sigma}$  для компьютерных программ других областей применения.

Будем предполагать, что интенсивность отказов компьютерной программы после выполнения ее тестирования уменьшается в  $Q$  раз относительно начальной интенсивности отказов  $\lambda_0$ . С учетом выражения (1) величина  $Q$  также показывает, во сколько раз за счет тестирования уменьшится число оставшихся ошибок в компьютерной программе относительно начального их числа (до выполнения тестирования), поэтому величину  $Q$  будем называть коэффициентом эффективности тестирования компьютерной программы. Используя выражения (4) и (5) и принимая во внимание значение коэффициента  $C_B = 0,238 \cdot 10^{-7}$  1/ошибка, модель определения ожи-

даемой интенсивности отказов компьютерной программы, прошедшей тестирование ( $\lambda_{\text{экс}}$ ), может быть представлена в виде

$$\lambda_{\text{экс}}^{(i)} = \frac{1}{Q} (2520 C^{(i)} V_{\text{пик}} N_0^{(i)}) = \frac{60}{Q} K_{\Sigma}^{(i)} V_{\text{пик}} F_0^{(i)} L \cdot 10^{-6} \text{ (ч}^{-1}\text{)}, \quad (6)$$

где верхний индекс  $i$  указывает на то, что соответствующие характеристики относятся к компьютерным программам  $i$ -й области применения.

**Определение параметров модели надежности компьютерной программы.** Значение параметра  $F_0$  модели (6) может быть найдено с помощью работы [19], в которой прогнозируемую плотность ошибок  $F$  предлагается определять в соответствии с моделью RL-92-52 [17] как произведение

$$F = A \cdot D \cdot S, \quad (7)$$

где  $A$  – средняя (базовая) плотность ошибок для компьютерных программ, используемых в данной отрасли или области применения (см. табл. 2);  $D$  – метрика, учитывающая производственную среду разработки программного обеспечения: особенность организации, разрабатывающей компьютерную программу; характеристику группы программистов, их квалификацию;  $S$  – метрика, учитывающая характеристики планируемой к разработке компьютерной программы.

Метрику  $D$  из выражения (7) предлагается определять в виде произведения двух коэффициентов:

$$D = K_{\text{орг}} \cdot K_{\text{кв.прог}},$$

где  $K_{\text{орг}}$  – коэффициент, характеризующий особенность организации, которая разрабатывает компьютерную программу;  $K_{\text{кв.прог}}$  – коэффициент, учитывающий квалификацию и опыт программистов.

В качестве рекомендуемых значений коэффициента  $K_{\text{орг}}$  могут быть взяты данные, приводимые в работе [17], о метрике, учитывающей среду разработки программного обеспечения. Если характеристики группы программистов (организации) заметно отличаются от указанных в [17], можно дать экспертную оценку значению коэффициента  $K_{\text{орг}}$  с учетом рекомендованного в этой работе условия

$$0,5 \leq K_{\text{орг}} \leq 2.$$

Для определения коэффициента  $K_{\text{кв.прог}}$  предлагается пользоваться данными табл. 3, полученными методом экспертной оценки на кафедре проектирования информационно-компьютерных систем Белорусского государственного университета информатики и радиоэлектроники [19].

Таблица 3  
Результаты экспертной оценки

Квалификация и опыт программиста	Значение $K_{\text{кв.прог}}$
Студент, освоивший программирование на уровне программы учебной дисциплины высшего технического учебного заведения	2,0
Младший программист (Junior Developer)	1,3
Программист (Middle Developer)	1,0
Ведущий программист (Senior Developer)	0,7

Для определения метрики  $S$  из выражения (7) предлагается уточненная модель вида

$$S = K_{\text{слож}} \cdot K_{\text{С.Р}} \cdot K_{\text{нов}} \cdot K_{\text{мод}}, \quad (8)$$

где  $K_{\text{слож}}$  – коэффициент, учитывающий категорию сложности компьютерной программы;  $K_{\text{С.Р}}$  – коэффициент, учитывающий используемые средства разработки компьютерной программы;  $K_{\text{нов}}$  – коэффициент, учитывающий степень новизны компьютерной программы;  $K_{\text{мод}}$  – коэффициент, характеризующий степень использования стандартных модулей в компьютерной программе.

В табл. 4 в качестве примера выбора поправочных коэффициентов модели (8) показано, как выбирать значения коэффициента  $K_{\text{С.Р}}$ , учитывающего используемые средства разработки компьютерной программы, и коэффициента  $K_{\text{мод}}$ , учитывающего степень применения стандартных модулей в разрабатываемой компьютерной программе. Значения коэффициентов могут быть получены с помощью постановления Министерства труда и социальной защиты Республики Беларусь № 91 от 27.06.2007 «Об утверждении укрупненных норм затрат труда на разработку программного обеспечения» с учетом гипотезы о том, что число возможных ошибок в компьютерной программе прямо пропорционально затратам труда на ее разработку.

Таблица 4  
 Выбор значения коэффициента  $K_{\text{С.Р}}$  в зависимости от операционной среды

Средства разработки программного обеспечения	IBM-PC, Windows	Функционирование программного обеспечения в сетях	
		локальных	глобальных
Языки высокого уровня (Си++, Паскаль)	1,0	1,2	1,3
Языки 4GL (Visual Basic, Delphi)	0,8	0,95	1,1
Системы программирования на основе СУБД типа Foxpro	0,45	0,55	0,65
Системы программирования на основе СУБД типа Oracle, SQLServer	0,4	0,5	0,6
Объектно-ориентированные технологии (COM/DCOM, CORBA)	0,55	0,6	0,7
Прочие CASE-средства	0,19	0,22	0,25

В табл. 5 указываются значения коэффициента  $K_{\text{мод}}$ , учитывающего степень использования стандартных модулей в разрабатываемой компьютерной программе.

Таблица 5  
 Выбор значения коэффициента  $K_{\text{мод}}$

Степень охвата реализуемых функций компьютерной программы стандартными модулями, %	Значение $K_{\text{мод}}$
От 60 и выше	0,55
От 40 до 60	0,65
От 20 до 40	0,77
До 20	0,90
Не используются стандартные модули	1,0



Аналогично, используя указанное выше постановление, могут быть получены значения коэффициентов, которые принимают во внимание сложность ( $K_{\text{слож}}$ ) и степень новизны ( $K_{\text{нов}}$ ) разрабатываемой компьютерной программы.

**Некоторые рекомендации по применению модели оценки надежности прикладных компьютерных программ.** Для использования модели необходимо прежде всего уточнить область (отрасль) применения компьютерной программы. Это позволит более обоснованно выбрать значение базовой плотности ошибок в компьютерной программе (величину  $A$ , используемую в модели определения плотности ошибок  $F$ ). Рекомендуется воспользоваться данными последнего столбца табл. 2. Если сложно сориентироваться с выбором области применения компьютерной программы, можно выбрать строку «среднее» в табл. 2. Далее, принимая во внимание функциональное назначение компьютерной программы, необходимо уточнить язык программирования и определить прогнозируемое число исполняемых строк программного кода  $L$ .

Необходимо также уточнить быстродействие процессора  $R$ , входящего в состав компьютера, на котором предполагается выполнять разрабатываемую программу, и оценить прогнозируемое число команд компьютерной программы  $B$ , а также значение пиковой скорости ( $V$ ) их выполнения.

Важно с учетом языка программирования и особенностей компьютерной программы (числа и размера циклов, объема тела циклов, числа условных переходов, особенностей интерфейса и т. д.) оценить значение коэффициента расширения программного кода  $E_L$ . Согласно работе [18] в случае неопределенности можно считать, что одна строка кода компьютерной программы транслируется в 10 машинных команд ( $E_L = 10$ ).

При получении метрики  $S$ , входящей в модель (7), следует обратить особое внимание на выбор значений коэффициентов, характеризующих разрабатываемую компьютерную программу. Выбор значений коэффициентов  $K_{\text{с.р}}$  и  $K_{\text{мод}}$  был описан выше. Для определения коэффициентов  $K_{\text{слож}}$  и  $K_{\text{нов}}$  можно использовать постановление Министерства труда и социальной защиты Республики Беларусь № 91 от 27.06.2007 «Об утверждении укрупненных норм затрат труда на разработку программного обеспечения» или обратиться к работе [20].

Значение коэффициента эффективности тестирования  $Q$  может быть получено экспертным методом в зависимости от предполагаемой продолжительности и технологии тестирования, а также опыта и квалификации тестировщиков. Согласно работе [17] при использовании общепринятых методов тестирования продолжительностью не менее 40 % общего времени, отводимого на разработку компьютерной программы, может быть принято  $Q \approx 50$ .

**Заключение.** Полученная модель надежности дает возможность приближенно оценить ожидаемую надежность планируемой к разработке компьютерной программы. Во многих случаях такой ориентировочный расчет полезен, так как позволяет на раннем этапе разработки компьютерной программы получить представление о ее ожидаемой надежности и, следовательно, оценить надежность компьютерной информационной системы. В случае несоответствия проектной надежности компьютерной информационной системы требованиям технического задания могут быть разработаны мероприятия по повышению уровня надежности компьютерной программы, что приведет к увеличению надежности системы в целом.

### Список использованных источников

1. Шубинский, И. Б. Функциональная надежность информационных систем. Методы анализа / И. Б. Шубинский. – М. : Надежность, 2012. – 296 с.
2. Чуканов, В. О. Методы обеспечения аппаратно-программной надежности вычислительных систем [Электронный ресурс] / В. О. Чуканов, В. В. Гуров, Е. В. Прокопьева. – 2014. – Режим доступа: <http://www.mcst.ru/metody-obespecheniya-apparatnoprogrammnoj-nadezhnosti-vychislitelnykh-sistem>. – Дата доступа: 21.04.2020.
3. Reliability Prediction Models. Notice 1: HDBK-217Plus™ / W. Denson [et al.]. – 2nd ed. – Washington : Quanterion Solutions Incorporated, 2015. – 186 p.

4. Gullo, L. J. The Revitalization of MIL-HDBK-217 [Electronic resource] / L. J. Gullo. – 2008. – Mode of access: [https://www.researchgate.net/publication/229021234\\_The\\_Revitalization\\_of\\_MIL-HDBK-217](https://www.researchgate.net/publication/229021234_The_Revitalization_of_MIL-HDBK-217). – Date of access: 20.04.2020.
5. A comparative overview of electronic devices reliability prediction methods-applications and trends / F. O. Ehiagwina [et al.] // *Majlesi J. of Telecommunication Devices*. – 2016. – Vol. 5, no. 4. – P. 129–137.
6. Прытков, С. Ф. Надежность электрорадиоизделий : справ. / С. Ф. Прытков, В. М. Горбачева, А. А. Борисов. – М. : ФГУП «22 ЦНИИИ МО РФ», 2008. – 641 с.
7. Разработка методики прогнозирования надежности электронных устройств для системы АРИОН / С. М. Боровиков [и др.] // *Доклады БГУИР*. – 2011. – № 4(58). – С. 93–100.
8. Боровиков, С. М. Расчет показателей надежности радиоэлектронных средств : учеб. пособие / С. М. Боровиков, И. Н. Цырельчук, Ф. Д. Троян. – Минск : БГУИР, 2010. – 68 с.
9. A Technique for Human Error Analysis. Technical Basis and Methodology Description (NUREG/CR-6350, BNL-NUREG-52467) / S. Cooper [et al.]. – Washington : U. S. Nuclear Regulatory Commission, 1996. – 114 p.
10. Hollnagel, E. Human Reliability Analysis: Context and Control / E. Hollnagel. – London : Academic Press, 1993. – 336 p.
11. Бодров, В. А. Психология и надежность: человек в системах управления техникой / В. А. Бодров, В. Я. Орлов. – М. : Ин-т психологии РАН, 1998. – 288 с.
12. Щербанов, В. Ю. Надежность деятельности человека в автоматизированных системах и ее количественная оценка / В. Ю. Щербанов, А. Ф. Бобров // *Психологический журнал*. – 1990. – № 3. – С. 36–40.
13. Sridevi, E. Software reliability – an overview / E. Sridevi, B. Aruna, P. Sowjanya // *Intern. J. of Computer Science and Technology*. – 2012. – Vol. 3, iss. 1. – P. 471–473.
14. Lyu, M. R. Handbook of Software Reliability Engineering / M. R. Lyu. – N. Y. : McGraw-Hill, 1996. – 850 p.
15. Prasad, C. M. Overview of software reliability models / C. M. Prasad, L. Flowrence, C. V. Srikrishna // *Intern. J. of Engineering and Management Research*. – 2013. – Vol. 3, iss. 5. – P. 11–15.
16. Karanta, I. Methods and Problems of Software Reliability Estimation: VTT Working Papers 63 / I. Karanta. – Espoo : VTT Technical Research Centre of Finland, 2006. – 57 p.
17. Software Reliability, Measurement, and Testing Guidebook for Software Reliability Measurement and Testing [Electronic resource] / J. A. McCall [et al.]. – 1992. – Mode of access: <https://apps.dtic.mil/dtic/tr/fulltext/u2/a256164.pdf>. – Date of access: 20.04.2020.
18. Чуканов, В. О. Надежность программного обеспечения и аппаратных средств систем передачи данных атомных электростанций : учеб. пособие / В. О. Чуканов. – М. : МИФИ, 2008. – 168 с.
19. Модель прогнозирования надежности планируемых к разработке прикладных компьютерных программ / С. М. Боровиков [и др.] // *Интернаука*. – 2020. – № 12(141). – Ч. 1. – С. 68–72.
20. Боровиков, С. М. Метод оценки ожидаемой надежности прикладных компьютерных программ систем медицинской электроники / С. М. Боровиков, С. С. Дик // *Доклады БГУИР*. – 2018. – № 7(117). – С. 112–117.

---

## References

1. Shubinsky I. B. Funktsionalnaya nadezhnost informatsionnykh sistem. Metody analiza. *Functional Reliability of Information Systems. Analysis Methods*. Moscow, Nadezhnost, 2012, 296 p. (in Russian).
2. Chukanov V. O., Gurov V. V., Prokop'eva E. V. Metody obespecheniya apparatno-programmnoj nadyozhnosti vychislitel'nykh system. *Methods for Ensuring the Hardware and Software Reliability of Computing Systems*, 2014. Available at: <http://www.mcst.ru/metody-obespecheniya-apparatnoprogrammnoj-nadezhnosti-vychislitelnykh-sistem> (accessed 21.04.2020).
3. Denson W., Nicholls D., Lein P., Wisnewski J. Reliability Prediction Models. Notice 1: HDBK-217Plus™. 2nd ed., Washington, Quanterion Solutions Incorporated, 2015, 186 p.
4. Gullo L. J. *The Revitalization of MIL-HDBK-217*, 2008. Available at: [https://www.researchgate.net/publication/229021234\\_The\\_Revitalization\\_of\\_MIL-HDBK-217](https://www.researchgate.net/publication/229021234_The_Revitalization_of_MIL-HDBK-217) (accessed 20.04.2020).
5. Ehiagwina F. O., Adewunmi T. O., Seluwa E. O., Kehinde O. O., Abubakar N. S. A comparative overview of electronic devices reliability prediction methods-applications and trends. *Majlesi Journal of Telecommunication Devices*, 2016, Vol. 5, no. 4, pp. 129–137.

6. Prytkov S. F., Gorbacheva V. M., Borisov A. A. Nadyozhnost elektroradioizdely. *Reliability of Radio and Electronic products*. Moscow, Federal'noe gosudarstvennoe unitarnoe predpriyatie "22 Central'nyj nauchno-issledovatel'skij ispytatel'nyj institut Ministerstva oborony Rossijskoj Federacii", 2008, 641 p. (in Russian).
7. Borovikov S. M., Shnejderov E. N., Matyushkov V. E., Cyrel'chuk I. N. Razrabotka metodiki prognozirovaniya nadyozhnosti elektronnykh ustroystv dlya sistemy ARION [Development of a method for predicting the reliability of electronic devices for the ARION system]. *Doklady Belorusskogo gosudarstvennogo universiteta informatiki i radioelektroniki [Reports of the Belarusian State University of Informatics and Radioelectronics]*, 2011, no. 4(58), pp. 93–100 (in Russian).
8. Borovikov S. M., Cyrel'chuk I. N., Troyan F. D. Raschyot pokazateley nadyozhnosti radioelektronnykh sredstv. *Calculation of Reliability Indicators of Electronic Equipment*. Minsk, Belorusskij gosudarstvennyj universitet informatiki i radioelektroniki, 2010, 68 p. (in Russian).
9. Cooper S. E., Ramey-Smith A. M., Wreathall J., Parry G. W., Bley D. C., ..., Barriere M. T. *A Technique for Human Error Analysis. Technical Basis and Methodology Description (NUREG/CR-6350, BNL-NUREG-52467)*. Washington, U. S. Nuclear Regulatory Commission, 1996, 114 p.
10. Hollnagel E. *Human Reliability Analysis: Context and Control*. London, Academic Press, 1993, 336 p.
11. Bodrov V. A., Orlov V. Ya. Psikhologiya i nadezhnost: chelovek v sistemakh upravleniya tekhnikoy. *Psychology and Reliability: Man in Control Systems of Technique*. Moscow, Institut psikhologii Rossijskoj akademii nauk, 1998, 288 p. (in Russian).
12. Shcheblanov V. Yu., Bobrov A. F. Nadezhnost deyatelnosti cheloveka v avtomatizirovannykh sistemakh i ee kolichestvennaya otsenka [Reliability of human activity in automated systems and its quantitative assessment]. *Psikhologicheskij zhurnal [Psychological Journal]*, 1990, no. 3, pp. 36–40 (in Russian).
13. Sridevi E., Aruna B., Sowjanya P. Software reliability – an overview. *International Journal of Computer Science and Technology*, 2012, vol. 3, iss. 1, pp. 471–473.
14. Lyu M. R. *Handbook of Software Reliability Engineering*. New York, McGraw-Hill, 1996, 850 p.
15. Prasad C. M., Flowrence L., Srikrishna C. V. Overview of software reliability models. *International Journal of Engineering and Management Research*, 2013, Vol. 3, iss. 5, pp. 11–15.
16. Karanta I. *Methods and Problems of Software Reliability Estimation: VTT Working Papers 63*. Espoo, VTT Technical Research Centre of Finland, 2006, 57 p.
17. McCall J. A., Randel W., Dunham J., Lauterbach L. *Software Reliability, Measurement, and Testing Guidebook for Software Reliability Measurement and Testing*, 1992. Available at: <https://apps.dtic.mil/dtic/tr/fulltext/u2/a256164.pdf> (accessed 20.04.2020).
18. Chukanov V. O. Nadyozhnost programmnoho obespecheniya i apparatnykh sredstv sistem peredachi dannykh atomnykh elektrostantsy. *Reliability of Software and Hardware for Data Transmission Systems of Nuclear Power Plant*. Moscow, Moskovskij inzhenerno-fizicheskij institut, 2008, 168 p. (in Russian).
19. Borovikov S. M., Dik S. S., Le Van Tam, Klinov K. I. Model prognozirovaniya nadyozhnosti planiruyemykh k razrabotke prikladnykh kompyuternykh programm [A model for predicting the reliability of software applications planned for development]. *Internauka [Internauka]*, 2020, no. 12(141), part 1, pp. 68–72 (in Russian).
20. Borovikov S. M., Dik S. S. Metod otsenki ozhidayemoy nadezhnosti prikladnykh kompyuternykh programm sistem meditsinskoj elektroniki [Method for assessing the expected reliability of applied computer programs of medical electronics systems]. *Doklady Belorusskogo gosudarstvennogo universiteta informatiki i radioelektroniki [Reports of the Belarusian State University of Informatics and Radioelectronics]*, 2018, no. 7(117), pp. 112–117 (in Russian).

### Информация об авторах

*Боровиков Сергей Максимович*, кандидат технических наук, доцент кафедры проектирования информационно-компьютерных систем, Белорусский государственный университет информатики и радиоэлектроники.  
E-mail: bsm@bsuir.by

*Казючич Владислав Олегович*, магистр технических наук, аспирант, Белорусский государственный университет информатики и радиоэлектроники.  
E-mail: vladkaz@bsuir.by

### Information about the authors

*Sergei M. Borovikov*, Cand. Sci. (Eng.), Associate Professor of the Department of Information and Computer Systems Design, Belarusian State University of Informatics and Radioelectronics.  
E-mail: bsm@bsuir.by

*Vladislav O. Kaziuchyts*, M. Sci. (Eng.), Postgraduate Student, Belarusian State University of Informatics and Radioelectronics.  
E-mail: vladkaz@bsuir.by

*Хорошко Виталий Викторович*, кандидат технических наук, заведующий кафедрой проектирования информационно-компьютерных систем, Белорусский государственный университет информатики и радиоэлектроники.

E-mail: khoroshko@bsuir.by

*Дик Сергей Сергеевич*, магистр технических наук, аспирант, Белорусский государственный университет информатики и радиоэлектроники.

E-mail: sdick@bsuir.by

*Клинов Константин Игоревич*, студент, Белорусский государственный университет информатики и радиоэлектроники.

E-mail: kost9klinov@gmail.com

*Vitaliy V. Khoroshko*, Cand. Sci. (Eng.), Head of the Department of Information and Computer Systems Design, Belarusian State University of Informatics and Radioelectronics.

E-mail: khoroshko@bsuir.by

*Sergei S. Dick*, M. Sci. (Eng.), Postgraduate Student, Belarusian State University of Informatics and Radioelectronics.

E-mail: sdick@bsuir.by

*Konstantin I. Klinov*, Student, Belarusian State University of Informatics and Radioelectronics.

E-mail: kost9klinov@gmail.com