

ПРАКТИКА ОЦЕНКИ ОЖИДАЕМОЙ НАДЁЖНОСТИ ПРИКЛАДНЫХ КОМПЬЮТЕРНЫХ ПРОГРАММ ДЛЯ ИНФОРМАЦИОННЫХ СИСТЕМ



Ван Там Лэ
Научный сотрудник,
Институт
технологий, Ханой,
Вьетнам, Магистр
техники и технологии
luct.tle94@gmail.com



С.М. Боровиков
Доцент кафедры
проектирования
информационно-
компьютерных систем
БГУИР, кандидат
технических наук,
доцент
bsm@bsuir.by



А.В. Будник
Декан факультета
инжиниринга и
технологий БГАС,
кандидат технических
наук, доцент
A.Budnik@bsac.by



С.К. Дик
Заведующий кафедрой
инженерной и
компьютерной графики
БГУИР, кандидат
физ.-мат. наук, доцент
sdick@bsuir.by

Ван Там Лэ

Окончил Белорусский государственный университет информатики и радиоэлектроники. Область научных интересов: информационные технологии; надёжность прикладных компьютерных программ; исследование, разработка и технология изготовления интегральных схем; исследования и разработка технологии изготовления магнитомягких материалов и оптического стекла.

С.М. Боровиков

Окончил Минский радиотехнический институт. Основная область научных интересов – прикладные математические методы в проектировании изделий радиоэлектроники, включая алгоритмы статистического прогнозирования надёжности изделий электронной техники и оценку надёжности прикладного программного обеспечения на ранних этапах его разработки.

А.В. Будник

Окончил Белорусский государственный университет информатики и радиоэлектроники. Основные направления научной деятельности – микроэлектроника, защита информации в технических системах.

С.К. Дик

Окончил Минский радиотехнический институт по специальности «Радиотехника», руководит научными исследованиями в области лазерной медицины и биомедицинской оптики.

Аннотация. Рассмотрена практика оценки эксплуатационной надёжности планируемых к разработке прикладных компьютерных программ, предназначенных для информационных систем. На конкретном примере показана оценка ожидаемой надёжности компьютерной программы, используемой для моделирования и исследования проектных решений в учебном процессе студентов технического вуза.

Ключевые слова: информационные системы, прикладные компьютерные программы, ожидаемая надёжность, тестирование.

Введение. Эффективное использование прикладных компьютерных программ во многом определяется их эксплуатационной надёжностью. Проблема обеспечения надёжности разрабатываемых компьютерных программ является острой [1–3], поскольку вклад программного обеспечения в ненадёжность современных сложных компьютерных информационных систем может составлять до 40 и более процентов. Недостаточный уровень надёжности и обусловленные этим частые сбои в использовании прикладных компьютерных программ раздражают пользователей, что вынуждает их отказываться от использования таких программ и предпринимать шаги по замене программ более надёжными версиями или делать попытки по поиску совершенно новых программ у заслуживающих доверия разработчиков и IT-организаций. Разработчикам программ, а также менеджерам IT-организаций очень важно получить информацию об ожидаемом уровне эксплуатационной надёжности планируемой к разработке будущей компьютерной программы ещё до написания её кода на языке программирования. Это позволит сориентироваться в целесообразности дальнейшего вкладывания и расходования денежных средств на разработку компьютерной программы с учётом бюджета времени и количества программистов, используемых для написания кода программы, а также определить требуемое календарное рабочее время и количество специалистов, которых следует задействовать для тестирования, испытания и отладки компьютерной программы. В данной работе на конкретном примере показано, как оценить ожидаемый уровень эксплуатационной надёжности компьютерной программы, планируемой к разработке, и определить, какое примерно рабочее календарное время потребуется для её тестирования с целью обеспечения заданного уровня эксплуатационной надёжности.

Основные принципы и подходы к оценке ожидаемой надёжности планируемых к разработке прикладных компьютерных программ. Эти принципы и подходы обсуждались в работах [1, 4–11] и в систематизированном виде описаны ниже:

1. Вклад программного обеспечения в ненадёжность сложных компьютерных информационных систем в ряде случаев может превышать вклад, вносимый техническими средствами (компьютерами), поскольку используемые программой входные данные могут быть сложными и их формат всё время меняется [3].

2. Единственно приемлемой количественной характеристикой измерения объёма (размера) компьютерной программы является число исполняемых строк программного кода (в англоязычном варианте LOC – Lines Of Code) [3]. Во многих случаях используют размерность KLOC – тысяч строк кода. Размер (объём) прикладных компьютерных программ современных информационных систем может достигать миллионов строк программного кода и оцениваться 1...100 млн и более команд.

3. В компьютерных программах большого объёма (тысячи–десятки тысяч и более строк кода) после тестирования в течение отводимого бюджетного времени всегда остаются скрытые ошибки, которые могут себя иногда проявлять в зависимости от характера набора

исходных данных и нагрузки на компьютерную программу со стороны эксплуатационной среды.

4. В прикладных компьютерных программах, которые протестированы только на предмет обеспечения функциональных возможностей, присутствует примерно от 5 до 30...50 ошибок на KLOC (1000 выполняемых строк программного кода) [3]. Для ответственного программного обеспечения (ПО), к которому можно отнести операционные системы и другое системное ПО, к моменту поставки программы клиенту в нём может содержаться до 0,04...0,15 ошибок на 1000 строк кода программы.

5. После написания кода компьютерной программы и устранения в ней ошибок, вызванных нарушением правил языка программирования, она характеризуется начальной интенсивностью проявления ошибок (λ_0), иначе – интенсивностью отказов. При типовых исходных данных обеспечивается функциональность компьютерной программы, но значение λ_0 обычно не отвечает требованиям пользователей. И кроме того, из-за наличия скрытых неявных ошибок в зависимости от изменчивости исходных данных и нагрузки на программу со стороны эксплуатационной среды (запись промежуточных данных на электронные носители, вывод данных на принтер, ожидание операций в очереди и т.д.) может не всегда обеспечиваться функциональность компьютерной программы.

6. Компьютерная программа после тестирования характеризуется эксплуатационной интенсивностью проявления ошибок (интенсивностью отказов) — $\lambda_{\text{экс}} < \lambda_0$. Задача тестирования состоит в том, чтобы в рамках отводимого бюджетного времени выявить и устранить наиболее критичные ошибки с точки зрения функциональности программы, свести к минимуму долю скрытых ошибок, оставшихся в компьютерной программе, и в итоге обеспечить приемлемый уровень её эксплуатационной интенсивности отказов $\lambda_{\text{экс}}$.

7. Переход от значения λ_0 к значению $\lambda_{\text{экс}}$ выполняют с учётом коэффициента тестирования Q [1]:

$$\lambda_{\text{экс}} = \frac{\lambda_0}{Q} \quad (1)$$

Достижимое численное значение Q зависит от бюджетного времени T_b , отводимого на тестирование, и области (сферы) применения компьютерной программы [4]:

$$Q = \exp[60 \cdot 10^{-6} \cdot V \cdot K_{\Sigma} \cdot T_b \cdot (r/100)], \quad (2)$$

где V – скорость выполнения команд компьютерной программы, размерность – с⁻¹; K_{Σ} – коэффициент увеличения интенсивности отказов из-за суммарного действия изменчивости входных данных и рабочей нагрузки на компьютерную программу со стороны эксплуатационной среды; T_b – бюджетное календарное рабочее время (в часах), отводимое на тестирование программы; r – средний процент времени выполнения (прогона) компьютерной программы при её тестировании в течение календарного рабочего дня для прикладной компьютерной программы рассматриваемой области применения.

8. Предлагаемая модель прогнозирования начальной интенсивности отказов компьютерной программы (λ_0) [1, 4, 10] учитывает следующие факторы:

- область (отрасль) применения компьютерной программы, что определяет изменчивость реального потока наборов исходных данных, вводимых в программу и рабочую нагрузку на компьютерную программу со стороны эксплуатационной среды;
- особенность организации, которая будет разрабатывать компьютерную программу, квалификация и опыт задействованных программистов;
- прогнозируемый размер (объём) компьютерной программы в KLOC;
- основные характеристики компьютерной программы: сложность, новизна, средства разработки, использование стандартных модулей;

- степень полноты учёта факторов, определяющих качество написания программного кода;
- быстродействие процессора в составе используемого компьютера.

$$\lambda_0 = 60 \cdot V \cdot K_{\Sigma} \cdot F_0 \cdot L \cdot 10^{-6} = 60 \cdot \left(\frac{R}{B}\right) \cdot K_{\Sigma} \cdot F_0 \cdot L \cdot 10^{-6} \text{ ч}^{-1}, \quad (3)$$

где F_0 – начальная плотность ошибок в компьютерной программе; L – прогнозируемое (в KLOC) число строк компьютерной программы.

Размерность параметров модели (3), квадратные скобки символизируют размерность: $[V] = \text{с}^{-1}$, $[F_0] = \text{ошибок/KLOC}$; $[L] = \text{KLOC}$; $[R] = \text{операций (команд) в секунду}$; $[B] = \text{команд}$.

9. В качестве основы определения F_0 использована «Римская модель» RL-92-52 [13], включающая метрику D , учитывающую особенность организации, которая будет разрабатывать компьютерную программу, и метрику S , учитывающую основные характеристики компьютерной программы. Для метрики S рекомендуется учесть такие её характеристики как сложность, новизна, средства разработки, степень использования стандартных модулей. Авторами статьи предлагается саму модель дополнить метрикой U , учитывающей, не принятые во внимание, но гипотетически влияющие на F_0 факторы, связанные с организацией, разрабатывающей программу, программистами и характеристиками самой будущей компьютерной программы. Её (метрику U) будем называть метрикой полноты учёта факторов. Вид предлагаемой модели определения F_0

$$F_0 = F_b \cdot D \cdot S (2 - U) \text{ ошибок/KLOC}, \quad (4)$$

где F_b – метрика области применения компьютерной программы, представляет собой среднюю по отрасли или базовую плотность отказов, размерность «ошибок/KLOC».

Метрика U безразмерная и должна определяться методом экспертной оценки из условия

$$0 < U \leq 1, \quad (5)$$

причём $U = 1$ – соответствует учёту практически всех факторов, определяющих значение F_0 (качество написания программного кода), а $U < 1$ – случаю, когда какая-то часть влияющих факторов не принята во внимание в метриках F_b , D и S .

10. Метрики D , S определяют как произведение коэффициентов K_i , каждый из которых учитывает конкретные факторы, влияющие на значения этих метрик [1, 4]:

$$D, S = \prod_i K_i \quad (6)$$

11. Эксплуатационная интенсивность отказов ($\lambda_{\text{о.к.}}$) характеризует надёжность компьютерной программы, но значение этой характеристики зависит от быстродействия процессора R , используемого в составе компьютера. Согласно ГОСТ 27.205–97, в качестве основной характеристики безотказного выполнения программой своих функций рекомендуется использовать вероятность того, что прикладная компьютерная программа безотказно выполнит обработку одного произвольного усреднённого набора исходных данных из числа тех наборов, которые могут поступать в условиях использования компьютерной программы в составе информационной системы. В работе [12], для расчёта этой вероятности (p_1) рекомендуется использовать экспоненциальную функцию:

$$p_1 = p_1(t_1) = \exp(-\lambda_{\text{о.к.}} \cdot t_1), \quad (7)$$

где $\lambda_{\text{жк}}$ – эксплуатационная интенсивность отказов в размерности 1/ч; t_1 – среднее процессорное время (в часах) обработки компьютерной программой одного усреднённого набора исходных данных; может быть найдено как

$$t_1 \approx \frac{B}{3600(0,7 \cdot R)}, \text{ ч}, \quad (8)$$

где B – прогнозная оценка числа команд, необходимых для обработки одного усреднённого набора исходных данных; R – пиковое быстродействие процессора компьютера, указанное в технической документации, размерность – «операций (команд) в секунду».

12. Вероятность того, что оставшиеся после тестирования скрытые ошибки в компьютерной программе не проявятся в течение заданного календарного рабочего времени τ , рассчитывают (с учётом ГОСТ 27.205–97) по формуле

$$P(\tau) = [p_1(t_1)]^{\eta \cdot \tau}, \quad (9)$$

где η – интенсивность обращения к компьютерной программе в течение периода τ ; η представляет собой среднее число сеансов («прогонов») программы за каждый час в течение календарного рабочего периода τ .

Пример оценки надёжности компьютерной программы. Для иллюстрации метода расчёта ожидаемой надёжности планируемой к разработке прикладной компьютерной программы использованы следующие исходные данные:

- область применения компьютерной программы: моделирование проектных решений и их использование в учебном процессе студентов учебного заведения;
- прогнозный объём компьютерной программы L , определяется на основе анализа блок-схемы алгоритма работы компьютерной программы; в случае затруднений следует воспользоваться рекомендациями документа [14]; получено $L = 40$ KLOC;
- язык программирования для написания программного кода: Java;
- суммарное календарное рабочее время тестирования компьютерной программы $T_b = 2100$ ч (5 студентов, для каждого $T = 420$ ч);
- средний процент времени прогона компьютерной программы (r) при её тестировании в течение календарного рабочего времени, $r = 3\%$ [4, 8];
- заданный календарный период эксплуатации τ , для которого определяют проектный уровень эксплуатационной надёжности компьютерной программы, $\tau = 176$ ч (1 месяц = 22 учебных дня по 8 часов);
- прогнозное число обращений в час к компьютерной программе (η) в течение заданного календарного периода эксплуатации $\tau = 176$ ч, принято $\eta = 4$ ч⁻¹;
- указанное в технической документации пиковое быстродействие процессора, используемого в компьютере, $R = 5 \cdot 10^8$ операций в секунду.

Для выбора коэффициентов, принимаемых во внимание при определении метрик D (особенность организации, разрабатывающей компьютерную программу) и S (характеристики компьютерной программы) использованы работы [1, 11, 14]. Последовательность оценки ожидаемой надёжности планируемой к разработке компьютерной программы:

1. Уточняем метрику производственной среды разработки компьютерной программы (D):

$$D = K_{\text{орг}} \cdot K_{\text{квал}} = 1,3 \cdot 2,0 = 2,6, \quad (10)$$

где $K_{\text{орг}}$ – коэффициент, учитывающий особенность проектной организации; принято $K_{\text{орг}} = 1,3$ (высшее учебное заведение как организация, эксплуатирующая компьютерную информационную систему) [1, 4, 10, 13];

$K_{\text{квал}}$ – коэффициент, учитывающий квалификацию программистов, принято $K_{\text{квал}} = 2,0$ (программист – студент технического вуза, освоивший программирование в рамках соответствующей учебной дисциплины) [1, 4, 10].

2. Получаем коэффициенты, включаемые в метрику S , учитывающую характеристики планируемой к разработке компьютерной программы:

2.1. Рассчитываем, используя [4, 10, 14], коэффициент сложности компьютерной программы ($K_{\text{слож}}$), учитывающий наличие факторов, влияющих на трудоёмкость разработки программы:

$$K_{\text{слож}} = 1 + \sum_{i=1}^m k_i = 1 + (k_1 + k_2 + k_3) = 1 + (0,08 + 0,06 + 0,12) = 1,26, \quad (11)$$

где k_i – коэффициент повышения сложности прикладной компьютерной программы за счёт наличия у неё i -й характеристики; принято $m = 3$.

В соответствии с документом [14] выбраны коэффициенты повышения сложности ($m = 3$):

$k_1 = 0,8$ – связь с другими компьютерными программами;

$k_2 = 0,06$ – хранение и поиск данных в сложных структурах;

$k_3 = 0,12$ – наличие одновременно двух характеристик сложности из числа описанных в документе [14]: а) интеллектуальный языковой интерфейс с пользователем, б) моделирование объектов и процессов.

2.2. Находим значение коэффициента $K_{\text{нов}}$, учитывающего новизну компьютерной программы. Выбрано $K_{\text{нов}} = 0,81$ [4, 14], что соответствует компьютерным программам, являющимся развитием определённого функционального назначения компьютерных программ.

2.3. Выбираем значение коэффициента $K_{\text{ср.разр}}$, учитывающего используемые средства разработки компьютерной программы. Принято $K_{\text{ср.разр}} = 1,2$ – язык программирования высокого уровня, функционирование программы в локальной сети [1, 4, 14].

2.4. Определяем значение коэффициента $K_{\text{мод}}$, учитывающего процент использования стандартных модулей. Выбрано $K_{\text{мод}} = 0,77$, что соответствует степени охвата стандартными модулями реализуемых функций от 20 до 40 % [1, 4, 14].

3. Рассчитываем саму метрику S с учётом коэффициентов $K_{\text{слож}}$, $K_{\text{нов}}$, $K_{\text{ср.разр}}$ и $K_{\text{мод}}$:

$$S = K_{\text{слож}} \cdot K_{\text{нов}} \cdot K_{\text{ср.разр}} \cdot K_{\text{мод}} = 1,26 \cdot 0,81 \cdot 1,2 \cdot 0,77 \approx 0,943. \quad (12)$$

4. Выбираем значение метрики U , учитывающей степень полноты учёта факторов, влияющих на качество написания программного кода. Путём экспертной оценки получено $U = 0,8$.

5. Определяем начальную плотность ошибок в компьютерной программе (F_0), характеризующую ожидаемое количество скрытых дефектов в программном коде после устранения ошибок, вызванных нарушением правил языка программирования:

$$F_0 = F_{\text{б}} \cdot D \cdot S \cdot (2 - U) = 12,3 \cdot 2,6 \cdot 0,943 \cdot (2 - 0,8) = 36,189 \text{ ошибок/KLOC}, \quad (13)$$

где $F_{\text{б}} = 12,3$ ошибок/KLOC – базовая (усреднённая) плотность ошибок для прикладных компьютерных программ, предназначенных для моделирования и обучения [1, 4, 10].

6. Даём оценку прогнозному значению количества команд, выполняемых программой:

$$B = L \cdot E_L \cdot E_{\text{ц}} = 40000 \cdot 10 \cdot 20 = 8,0 \cdot 10^6 \text{ команд}, \quad (14)$$

где E_L – коэффициент расширения кода программы относительно числа строк кода L , определяется используемым языком программирования. Согласно [15], принято $E_L = 10$. $E_{\text{ц}}$ – коэффициент увеличения числа выполняемых процессором команд за счёт наличия в программе условных переходов, ветвлений и других особенностей (ввод-вывод данных, считывание информации с внешних носителей, ожидание в очереди, обработка прерываний).

Значение $E_{\text{ц}}$ следует выбирать на основе экспертной оценки с учётом алгоритма выполнения компьютерной программой своих функций, экстремального характера входных данных, числа ветвлений в программе, записи и считывания данных с внешних носителей и т.п. В случае неопределённости рекомендуется выбирать значение от 20 до 50. Принято $E_{\text{ц}} = 20$.

7. Определяем скорость выполнения команд компьютерной программы (V) для пикового быстродействия R процессора, размерность R – «операций (команд) в секунду»:

$$V = \frac{R}{B} = \frac{5 \cdot 10^8}{8 \cdot 10^6} = 62,5 \text{ с}^{-1}. \quad (15)$$

8. Получаем, используя модель (3), начальную интенсивность отказов λ_0 (интенсивность проявления скрытых ошибок), т.е. интенсивность отказов до начала тестирования программы:

$$\lambda_0 = 60 \cdot K_{\Sigma} \cdot V \cdot F_0 \cdot L \cdot 10^{-6} = 60 \cdot 14,1 \cdot 62,5 \cdot 36,189 \cdot 40 \cdot 10^{-6} = 76,54 \text{ ч}^{-1}, \quad (16)$$

где K_{Σ} – коэффициент увеличения интенсивности отказов из-за суммарного действия изменчивости входных данных и рабочей нагрузки на компьютерную программу со стороны эксплуатационной среды (ввод данных, использование принтера, ожидание операций в очереди и т. д); для компьютерных программ, используемых для моделирования и обучения для рассматриваемой области применения компьютерной программы, $K_{\Sigma} = 14,1$ [1, 4, 10].

Размерности параметров: $[V] = \text{с}^{-1}$; $[F_0] = \text{ошибка/KLOC}$; $[L] = \text{KLOC}$.

9. Определяем по формуле (2) проектное значение коэффициента тестирования Q компьютерной программы в предположении, что тестирование программы выполняется в течение суммарного бюджетного календарного рабочего времени $T_{\text{б}} = 2100$ ч:

$$Q = \exp[60 \cdot 10^{-6} \cdot K_{\Sigma} \cdot V \cdot T_{\text{б}} \cdot (r/100)] = \exp[60 \cdot 10^{-6} \cdot 14,1 \cdot 62,5 \cdot 2100 \cdot (3/100)] = 27,97. \quad (17)$$

10. Находим эксплуатационную интенсивность отказов компьютерной программы ($\lambda_{\text{экс}}$):

$$\lambda_{\text{экс}} = \frac{\lambda_0}{Q} = \frac{76,54}{27,97} = 2,7365 \approx 2,74 \text{ ч}^{-1}. \quad (18)$$

11. По формуле (8) определяем среднее процессорное время (в часах) обработки компьютерной программой одного усреднённого набора исходных данных (значение R должно подставляться в размерности «операций (команд) в секунду»:

$$t_1 \approx \frac{B}{3600(0,7 \cdot R)} = \frac{8 \cdot 10^6}{3600 \cdot (0,7 \cdot 5 \cdot 10^8)} = 6,349 \cdot 10^{-6} \text{ ч.} \quad (19)$$

12. Вероятность безотказной работы программы в течение календарного времени τ (приняты во внимание ГОСТ 27.205–97 и работы [4, 11]):

$$P(\tau) = (P_1)^{n \cdot \tau} = [\exp(-\lambda_{\text{экс}} \cdot t_1)]^{n \cdot \tau} = [\exp(-2,74 \cdot 6,349 \cdot 10^{-6})]^{4 \cdot 176} = 0,988, \quad (20)$$

где P_1 – вероятность безотказной обработки одного усреднённого набора исходных данных.

Заключение. Оценка ожидаемой надёжности планируемой к разработке прикладной компьютерной программы, предназначенной для работы в составе информационной системы, в том числе и при обработке Big Data, является ответственной задачей, так как позволяет ещё до начала написания программного кода сориентироваться в целесообразности вкладывания средств в разработку компьютерной программы и оценить время, необходимое для обеспечения её заданной эксплуатационной надёжности.

Иллюстрация применения метода расчёта надёжности, рассматривая конкретный пример, позволит лучше осмыслить подход, используемый для оценки ожидаемой надёжности планируемой к разработке прикладной компьютерной программы.

Список литературы

- [1] 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. – P. 84–95 (in Russian). <https://doi.org/10.37661/1816-0301-2021-18-1-84-95>.
- [2] Чуканов В.О., Гуров В.В., Прокопьева Е.В. Методы обеспечения аппаратно-программной надёжности вычислительных систем [Электронный ресурс]. – 2014. – URL: http://www.mcst.ru/files/5357ec/dd0cd8/50af39/000000/seminar_metody_obespecheniya_apparatno-programmnoy_nadezhnosti_vychislitelnyh_sistem.pdf (дата обращения 14.03.2026).
- [3] Программное обеспечение – источник всех проблем [Электронный ресурс]. – 2018. – URL: <http://www.williamspublishing.com/PDF/5-8459-0785-3/part1.pdf> (дата обращения 14.03.2026).
- [4] Боровиков С.М., Будник А.В., Лэ В.Т. Надёжность разрабатываемых прикладных компьютерных программ для информационных систем = Reliability of developed application computer programs of information systems // Проблемы инфокоммуникаций: научный журнал. – 2025. – № 1. – С. 25–34.
- [5] Боровиков С.М., Дик С.С., Лэ В.Т., Клинов К.И. Анализ и оценка надёжности прикладных компьютерных программ // BIG DATA и анализ высокого уровня: Сб. матер. VI Междунар. науч.-практ. конф. / Минск (20–21 мая 2020 г.). – Ч. 1. – Минск: Бестпринт, 2020. – С. 382–390.
- [6] Боровиков С.М., Дик С.С., Лэ В.Т., Клинов К.И. Модель прогнозирования надёжности планируемых к разработке прикладных компьютерных программ // Интернаука: научный журнал. – 2020. – № 12 (141), ч. 1. – С. 68–72.
- [7] Боровиков С.М., Дик С.С., Лэ В.Т., Клинов К.И. Возможный подход к оценке надёжности разрабатываемых программных средств на ранних этапах проектирования информационно-компьютерных систем // Globus: технические науки – от теории к практике [Электронный ресурс]: Сб. науч. публ. – 2020. – Вып. 1 (32). – С. 4–9.
- [8] Боровиков С.М., Казючиц В.О., Дик С.К., Дик С.С. Модель прогнозирования времени тестирования прикладных компьютерных программ для технологий BIG DATA / BIG DATA и анализ высокого уровня: Сб. науч. статей VII Междунар. науч.-практ. конф. / Минск (19–20 мая 2021 г.). – Минск: Бестпринт, 2021. – С. 404–411.
- [9] Боровиков С.М., Казючиц В.О., Дик С.С., Юренин К.В. Модель прогнозирования времени тестирования прикладных компьютерных программ для автоматизированных систем управления // Информационные технологии и системы 2021: Матер. Международ. науч. конф. ИТС-2021 / Минск (24 ноября 2021 г.). – Минск, 2021. – С. 28–29.
- [10] Боровиков С.М., Казючиц В.О., Дик С.К., Дик С.С. Методика обеспечения эксплуатационной надёжности планируемых к разработке прикладных компьютерных программ для информационных систем //

BIG DATA и анализ высокого уровня: Сб. науч. статей VIII Междунард. науч.-практ. конф. / Минск (11–12 мая 2022 г.). – Минск: Бестпринт, 2022. – С. 162–173.

[11] Лэ В.Т., Боровиков С.М., Дик С.К., Будник А.В. Рекомендации по оценке и обеспечению надёжности планируемых к разработке прикладных компьютерных программ для информационных систем // BIG DATA и анализ высокого уровня: Сб. науч. статей IX Междунард. науч.-практ. конф. / Минск (17–18 мая 2023 г.). – Ч. 2. – Минск, 2023. – С. 78–86.

[12] Шубинский И.Б. Функциональная надёжность информационных систем. Методы анализа – М.: «Журнал Надёжность», 2012. – 296 с.

[13] McCall J.A., Randell W., Dunham J., Lauterbach L. Software reliability, measurement, and testing guidebook for software reliability measurement and testing // RL-TR-92-52: Final Technical Report, Vol II (of two) / Science Applications International Corp., Research Triangle Institute. – Rome Laboratory, NY 13441-5700, 1992. – 256 p. – URL: <https://apps.dtic.mil/sti/tr/pdf/ADA256164.pdf> (дата обращения 14.03.2026).

[14] Постановление Министерства труда и социальной защиты Республики Беларусь 27 июня 2007 г. № 91 «Об утверждении укрупнённых норм затрат труда на разработку программного обеспечения» [Электронный ресурс]. – URL: <https://news-newsby-org.narod.ru/doc-razn/pos05/postn05236/index.htm> (дата обращения 14.03.2026).

[15] Чуканов В.О. Надёжность программного обеспечения и аппаратных средств систем передачи данных атомных электростанций: учеб. пособ. – М.: МИФИ, 2008. – 168 с.

Авторский вклад

Лэ Ван Там – уточнение модели для оценки ожидаемой начальной интенсивности проявления ошибок прикладной компьютерной программой.

Боровиков Сергей Максимович – общее руководство исследованиями; подготовка данных для расчёта метрики, учитывающей характеристики планируемой к разработке компьютерной программы.

Будник Артур Владимирович – подготовка данных для расчёта метрики, учитывающей характеристику производственной среды разработки компьютерной программы, в том числе для мониторинга и обеспечения безопасности, телекоммуникации и мобильных устройств, систем автоматизированного управления.

Дик Сергей Константинович – подготовка справочных параметров для оценки надёжности прикладных компьютерных программ для разных областей применения.

PRACTICE OF ASSESSING THE EXPECTED RELIABILITY OF APPLIED COMPUTER PROGRAMS FOR INFORMATION SYSTEMS

Van Tam Le
Researcher, Institute of Technology, Hanoi, Vietnam, Master of Engineering and Technology

S.M. Borovikov
Associate Professor, Department of Information Computer Systems Design, BSUIR, PhD of Technical sciences, Associate Professor

A.V. Budnik
Dean of the Faculty of Engineering and Technology of the BSAC, PhD of Technical sciences, Associate Professor

S.K. Dick
Head of the Department of Engineering and Computer Graphics at BSUIR, PhD of Technical Sciences, Associate Professor

Abstract. This article examines the practice of assessing the operational reliability of planned computer applications designed for information systems. A specific example is provided for assessing the expected reliability of a computer program used for modeling and researching design solutions in the educational process for students at a technical university.

Keywords: computer information systems, applied software, expected reliability of software, testing.