

УДК 004.4' 242:004.6-027.45

УСКОРЕННАЯ ОЦЕНКА НАДЁЖНОСТИ РАЗРАБАТЫВАЕМЫХ ПРИКЛАДНЫХ КОМПЬЮТЕРНЫХ ПРОГРАММ



Ван Там Лэ
Ассистент-исследователь
Ханойского технологического института,
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



С.К. Дик
Доцент кафедры электронной техники и технологии БГУИР,
кандидат технических наук,
доцент
sdick@bsuir.by



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

С.К. Дик

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

А.В. Будник

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

Аннотация. Уточнены и систематизированы результаты исследований по оценке эксплуатационной надёжности разрабатываемых компьютерных программ, предназначенных для использования в различных сферах деятельности людей, в том числе в сфере *Big Data*.

Приводятся сведения по оценке ожидаемой начальной надёжности компьютерных программ и рекомендации по оценке продолжительности тестирования с целью обеспечения эксплуатационной надёжности планируемых к разработке прикладных компьютерных программ для сложных информационных систем.

Ключевые слова: прикладные компьютерные программы, прогнозируемая плотность ошибок, модели надёжности, тестирование.

Введение. В настоящее время проблема обеспечения надёжности разрабатываемых компьютерных программ стоит остро, поскольку по мнению ряда специалистов вклад программного обеспечения в ненадёжность сложных информационно-компьютерных систем может составлять 40 и более процентов [1].

Формы проявления ненадёжности прикладных компьютерных программ разнообразны: «зависание» программы, переполнение памяти, невозможность продолжения выполнения из-за некорректности входных данных, деление на нуль и т.д. Ненадёжность прикладных компьютерных программ большого размера (объёма) является следствием наличия в них скрытых неявных ошибок, оставшихся после выполнения процедуры её тестирования в течение отводимого времени на разработку программы. Специалисты по программированию пришли к выводу, что при оценке надёжности компьютерных программ в качестве единственно приемлемой количественной характеристикой измерения объёма программы является число исполняемых строк программного кода (в англоязычном варианте *LOC – Lines Of Code*) [2].

Прикладные компьютерные программы для сложных систем могут включать сотни тысяч и даже миллионы строк кода. Как пример, большой объём современного программного обеспечения иллюстрируется данными, приведёнными в таблице 1.

Таблица 1. Объём программного обеспечения

Программное средство (объект)	Космическая станция	Космический корабль	Boeing 777	Windows NT5	Windows 95	Windows XP
Количество строк кода, млн	40	10	7	35	5	40

Среднее число ошибок, приходящихся на тысячу строк кода (*KLOC*) варьируется от 5 до 50 ошибок [1]. Даже в системах, которые прошли строгий контроль качества (в англоязычном наименовании *Quality Assurance – QA*) все равно содержатся приблизительно 5 ошибок на 1000 строк кода [2]. Для ответственного программного обеспечения (ПО), к которому можно отнести операционные системы и другое системное ПО, к моменту поставки системы клиенту в нём может содержаться 0,04...0,15 ошибки на 1000 строк кода программы.

В прикладных компьютерных программах, которые протестированы только на предмет работоспособности функциональных возможностей, присутствует около 50 ошибок на 1000 выполняемых строк кода (рисунок 1).

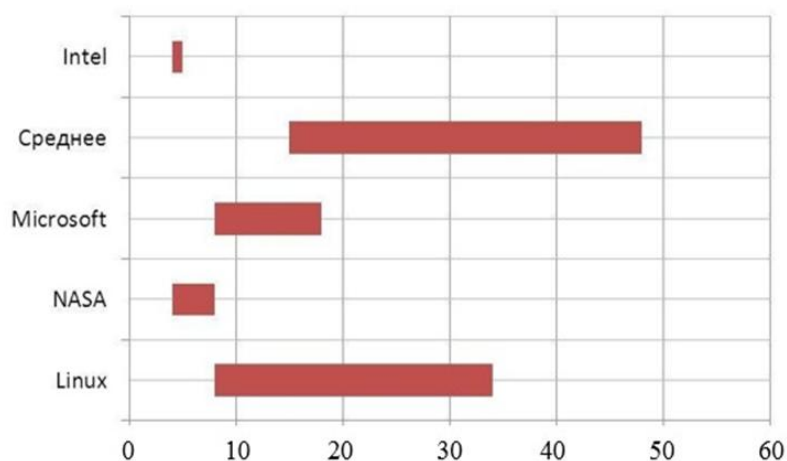


Рисунок 1. Среднее число ошибок, приходящихся на 1000 строк кода,

для прикладных программных средств, прошедших тестирование

По данным [4] отношение числа ошибок, не выявленных при проектировании прикладной компьютерной программы, к общему числу команд программы (обозначим общее число команд программы через B) лежит в диапазоне от 0,25 до 10 на 1000 команд.

Примерное значение числа команд B можно определить как [6]

$$B = E_L L, \quad (1)$$

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

Согласно работе [6] можно считать, что одна строка кода компьютерной программы транслируется в 10 машинных команд, т. е. $E_L = 10$.

Актуальность. В соответствии с ГОСТ 27.205-1997 [3] о прогнозной надёжности разрабатываемой компьютерной программы будем судить по интенсивности проявления ошибок (λ), обусловленных дефектами её проектирования. Поскольку проявление ошибки при использовании программы приводит к невыполнению задачи пользователя, что равносильно отказу программы, то далее будем употреблять термин «интенсивность отказов программы». Использование λ в качестве характеристики надёжности программных средств удобно тем, что её можно оценивать по прогнозируемой плотности ошибок F в компьютерной программе [5]. Плотность ошибок F это отношение вида

$$F = \frac{\text{Количество возможных ошибок в компьютерной программе}}{\text{Число строк кода компьютерной программы}}.$$

Прогнозирование плотности отказов не требует знания среды исполнения компьютерной программы и, следовательно, подходит для ранних этапов разработки программного обеспечения. Когда информация о предполагаемой среде исполнения становится доступной, прогнозируемая плотность отказов может быть преобразована в прогнозируемую интенсивность отказов. Актуальными являются получение прогнозного значения начальной плотности ошибок, т.е. плотности ошибок компьютерной программы после написания её кода и устранения нарушений правил языка программирования (обозначим через F_0) и алгоритма преобразования её в начальную интенсивность отказов λ_0 разрабатываемой компьютерной программы.

Прогнозное значение начальной плотности ошибок. Определение прогнозного значения начальной плотности ошибок F_0 рассматривалось в [7–10]. На основе анализа установлено, что возможным является использование модели RL-92-52 [5], согласно которой F_0 можно представить в виде произведения

$$F_0 = F_B \cdot D \cdot S = F_B \cdot \prod_{i=1}^m D_i \cdot \prod_{j=1}^n S_j, \quad (2)$$

где F_B – базовая (средняя) плотность ошибок для компьютерных программ, используемых в данной отрасли, области применения; D – метрика, учитывающая производственную среду разработки программного обеспечения, в том числе особенность организации, разрабатывающей компьютерную программу, характеристику группы программистов, их квалификацию; S – метрика, учитывающая характеристики разрабатываемой компьютерной программы; D_i – поправочный коэффициент, учитывающий влияние на значение F_0 i -го фактора, описывающего производственную среду разработки компьютерной программы; S_j – поправочный коэффициент, учитывающий j -ю

характеристику компьютерной программы; m – число факторов, принимаемых во внимание при описании метрики D ; n – число характеристик компьютерной программы, выбранных для описания метрики S .

Коэффициенты метрик D и S могут принимать значения как меньше единицы, так и больше неё в зависимости от того, как фактор влияет на плотность ошибок.

Метрику D модели (2) в соответствии с [8] предлагается определять в виде произведения двух коэффициентов:

$$D = D_{\text{орг}} \cdot D_{\text{квал}},$$

где $D_{\text{орг}}$ – коэффициент, характеризующий особенность организации, разрабатывающей прикладную компьютерную программу; $D_{\text{квал}}$ – коэффициент, учитывающий квалификацию и опыт программистов.

Вопросы выбора коэффициента $D_{\text{орг}}$ рассматриваются в [5]. Для выбора коэффициента $D_{\text{квал}}$ можно использовать рекомендации работы [10] (таблица 1).

Таблица 1. Значения коэффициента $K_{\text{квал}}$

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

Выбор коэффициентов S_j , принимаемых во внимание для компьютерной программы, удобно сделать в соответствии с документом «Укрупнённые нормы затрат труда на разработку программного обеспечения. Постановление Министерства труда и социальной защиты Республики Беларусь 27.06.2007 № 91». Этот документ позволяет определить нормы времени на разработку компьютерных программ с учётом основных их характеристик. С учётом документа [11] рекомендуется принять во внимание коэффициенты S_j , учитывающие следующие характеристики разрабатываемой компьютерной программы:

S_1 – категорию сложности компьютерной программы;

S_2 – используемые средства разработки компьютерной программы (язык программирования; операционная система компьютера; сеть функционирования: локальная, глобальная; другие факторы);

S_3 – степень новизны разрабатываемой компьютерной программы;

S_4 – степень использования стандартных модулей в компьютерной программе.

Для выбора значений коэффициентов $S_1 \dots S_4$ рекомендуется использовать данные документа [11] с учётом гипотезы о том, что число возможных ошибок в компьютерной программе, следовательно, и плотность ошибок, прямо пропорционально затратам труда на её разработку.

Преобразование плотности ошибок в интенсивность отказов программы. В работе [10] на основе экспериментальных данных о надёжности прикладных компьютерных программ [5], показано, как по прогнозному значению начальной плотности ошибок F_0 , объёму программы в исполняемых строках программного кода L ,

области применения программы, а также быстродействию R процессора компьютера, получить ожидаемый (примерный) уровень начальной интенсивности отказов λ_0 :

$$\lambda_0^{(l)} = (2520 C^{(l)} V_{\text{пик}} N_0^{(l)}) = 60 K_{\Sigma}^{(l)} V_{\text{пик}} F_0^{(l)} L 10^{-6}, \text{ч}^{-1}, \quad (3)$$

где верхний индекс l означает, что соответствующие параметры относятся к прикладной компьютерной программе l -й области применения; F_0 – плотность ошибок компьютерной программы после написания её кода и устранения нарушений правил языка программирования, размерность: ошибка / строка кода; $C^{(l)}$ – коэффициент проявления ошибок программы, размерность: 1/ошибка; $V_{\text{пик}}$ – пиковая скорость выполнения команд компьютерной программы, представляющая собой возможное число прогонов компьютерной программы в течение одного часа для пикового быстродействия процессора; N_0 – начальное число ошибок (до выполнения тестирования); $K_{\Sigma}^{(l)}$ – суммарный коэффициент увеличения интенсивности отказов, обусловленный совместным действием изменчивости входных данных и рабочей нагрузки на компьютерную программу; L – прогнозное значение числа выполняемых строк программного кода (LOC).

В случае затруднений в определении прогнозного значения L разрабатываемой компьютерной программы рекомендуется обратиться к документу [11].

Коэффициент $C^{(l)}$ модели (3) показывает, как значение начальных ошибок $N_0^{(l)}$ в прикладной компьютерной программе, предназначенной для её применения в l -й области, трансформируется в начальную интенсивность отказов λ_0 при пиковой скорости $V_{\text{пик}}$ выполнения команд компьютерной программы, определяемой как

$$V_{\text{пик}} = \frac{R_{\text{пик}}}{B}, \quad (4)$$

где $R_{\text{пик}}$ – пиковое значение быстродействия процессора компьютера, указанное в технической документации, размерность: «операций в секунду»; B – прогнозное число команд (операций), которое будет выполняться программой в течение одного прогона для экстремального набора входных данных и экстремальной рабочей нагрузки на компьютерную программу со стороны эксплуатационной среды.

В первом произведении модели (3) коэффициент (число) 2520 получен с учётом того, что рассматривается скорость $V_{\text{пик}}$ выполнения команд компьютерной программы в час, а реальное быстродействие R процессора принято, как $R = 0,7 \cdot R_{\text{пик}}$.

Достоверность начальной интенсивности отказов компьютерной программы λ_0 во многом определяется точностью оценки значения B . Использование равенства (1) для получения значения B даёт слишком ориентировочную оценку. Рекомендуется делать экспертную оценку числа команд B , выполняемых прикладной компьютерной программой за один прогон для случая экстремального режима входных данных и экстремальной нагрузки со стороны эксплуатационной среды (загрузка и выгрузка модулей прикладной компьютерной программы из памяти, нахождение операций ввода-вывода в очереди, наличие состояний ожидания и т.д.). Этап экспертной оценки B является достаточно сложным.

Используя экспериментальные данные о надёжности прикладных компьютерных программ, в [10] получены значения коэффициентов $C^{(l)}$ и $K_{\Sigma}^{(l)}$ для разных областей применения, для которых разрабатываются прикладные компьютерные программы (таблица 2). Эти коэффициенты могут рассматриваться в качестве справочных, используемых для оценки надёжности разрабатываемых прикладных компьютерных программ.

В таблицу 2 также включены справочные значения базовых плотностей ошибок, полученные на основе исследования надёжности прикладных компьютерных программ разных областей (сфер) их применения [5].

Таблица 2. Значения коэффициентов для компьютерных программ различных областей применения

Область применения компьютерной программы	Коэффициент $C, \times E-07, 1/\text{ошибка}$	Коэффициент K_{Σ}	Значение $F_{\text{б}}, \text{ошибка} / \text{строка кода}$	Средний процент времени прогона программы при её тестировании в течение календарного времени $T_{\text{календ}}$
1 Авиация, космос	1,246	5,23	0,0128	8
2 Мониторинг и обеспечение безопасности	0,238	1,00	0,0092	43
3 Телекоммуникации, мобильные устройства	2,738	11,5	0,0078	3,5
4 Управление производственными процессами	0,754	3,17	0,0018	14
5 Автоматизированные системы управления, технологии <i>Big Data</i>	4,563	19,2	0,0085	2,5
6 Разработка программ, моделирование, обучение	3,353	14,1	0,0123	3
Среднее	2,103	8,83	0,0087	12

Значение λ_0 , полученное по модели (3), обычно не отвечает требованиям заказчика, поэтому выполняют тестирование компьютерной программы с целью поиска и устранения скрытых ошибок, которые гипотетически могут иногда проявляться в зависимости от набора входных данных и уровня нагрузки программы со стороны эксплуатационной среды. Используя работу [12], требуемое процессорное время тестирования можно определить по формуле

$$T_{\text{календ}} = \frac{B \ln(Q)}{60 K_{\Sigma}^{(l)} R r^{(l)}} 10^{-4} = \frac{L K_B \ln(Q)}{60 K_{\Sigma}^{(l)} R r^{(l)}} 10^{-4}, \text{ ч}, \quad (5)$$

где Q – коэффициент тестирования; $r^{(l)}$ – средний процент времени выполнения (прогона) компьютерной программы при её тестировании в течение рабочей календарной продолжительности для прикладных компьютерных программ l -й области применения (см. таблицу 2); K_B – коэффициент увеличения числа команд компьютерной программы относительно числа выполняемых строк программного кода L .

Коэффициент K_B учитывает, как язык программирования, так и специфику процедур, выполняемых компьютерной программой.

В формулу (5) пиковое быстроедействие процессора R следует подставлять в размерности «операций в секунду», показатель L – в строках кода (LOC).

Значение Q находят как

$$Q = \frac{\lambda_0}{\lambda_{\text{экс}}}, \quad (6)$$

где $\lambda_{\text{экс}}$ – допустимая эксплуатационная интенсивность отказов компьютерной программы (интенсивность проявления ошибок).

В качестве основной характеристики безотказности выполнения программой своих функций, согласно [4], можно использовать вероятность того, что прикладная компьютерная программа безотказно выполнит обработку одного произвольного набора исходных данных из числа тех наборов, которые могут поступать в условиях функционирования компьютерной программы в составе информационной системы. Согласно работе [3], эту вероятность (обозначим через P_1) можно определить, как

$$P_1 = \exp(-\lambda_{\text{экс}} t_1), \quad (7)$$

где $\lambda_{\text{экс}}$ – ожидаемая (прогнозируемая) интенсивность отказов компьютерной программы, размерность $1/\text{ч} = \text{ч}^{-1}$; t_1 – среднее время (в часах) обработки прикладной компьютерной программой одного набора исходных данных.

В формуле (7) в качестве времени t_1 следует рассматривать процессорное время обработки данных (без учёта времени ожидания).

Важным вопросом является уточнение частоты обращения к прикладной компьютерной программе, т.е. среднего числа сеансов («прогонов») программы за один час в процессе функционирования информационной системы (обозначим это среднее число сеансов через η). Тогда вероятность того, что оставшиеся ошибки в прикладной компьютерной программе не проявятся в течение заданного календарного времени τ , можно определить, используя выражение

$$P(\tau) = (P_1)^\eta \tau. \quad (8)$$

Если проектное значение вероятности $P(\tau)$ не отвечает требованиям функционирования информационной системы, то следует увеличить проектное время тестирования прикладной компьютерной программы.

Список литературы

- [1] Чуканов, В. О., Методы обеспечения аппаратно-программной надёжности вычислительных систем / В. О. Чуканов, В. В. Гуров, Е. В. Прокопьева [Электронный ресурс]. – 2014. – Режим доступа: http://www.mcst.ru/files/5357ec/dd0cd8/50af39/000000/seminar_metody_obespecheniya_apparatno-programmnoy_nadezhnosti_vychislitelnyh_sistem.pdf. – Дата доступа: 31.01.2024.
- [2] Программное обеспечение – источник всех проблем. [Электронный ресурс]. – 2018. – Режим доступа: <http://www.williamspublishing.com/PDF/5-8459-0785-3/part1.pdf>. – Дата доступа: 31.01.2024.
- [3] Шубинский, И. Б. Функциональная надёжность информационных систем. Методы анализа / И. Б. Шубинский. – М. : «Журнал Надёжность», 2012. – 296 с.
- [4] Надёжность в технике. Проектная оценка надёжности сложных систем с учётом технического и программного обеспечения и оперативного персонала. Основные положения : ГОСТ 27.205-97. – Введен в действие с 01.10.2005. – Минск : Госстандарт Республики Беларусь, 2005. – 22 с.
- [5] McCall, J. A. Software reliability, measurement, and testing guidebook for software reliability measurement and testing / J. A. McCall [et al.] // Final Technical Report, Vol II (of two) ; RL-TR-92-52. – Science Applications International Corp., Research Triangle Institute : Rome Laboratory, NY 13441-5700, 1992. – 256 p.

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

[7] Модель прогнозирования надёжности планируемых к разработке прикладных компьютерных программ / С. М. Боровиков [и др.] // Интернаука: научный журнал. – 2020. – № 12 (141). Ч. 1. – С. 68-72.

[8] Возможный подход к оценке надёжности разрабатываемых программных средств на ранних этапах проектирования информационно-компьютерных систем / С. М. Боровиков [и др.] // Globus: технические науки – от теории к практике [Электронный ресурс] : сборник научных публикаций. – 2020. – Вып. 1 (32). – С. 4-9.

[9] Анализ и оценка надёжности прикладных компьютерных программ / С. М. Боровиков [и др.] // BIG DATA and Advanced Analytics = BIG DATA и анализ высокого уровня : сб. материалов VI Междунар. науч.-практ. конф., Минск, 20-21 мая 2020 года: в 3 ч. Ч. 1. – Минск : Бестпринт, 2020. – С. 382-390.

[10] Оценка ожидаемой надёжности 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>

[11] 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]. – Режим доступа: <https://zakonrb.com/npa/ob-utverzhdenii-ukrupnennyh-norm-zatrat-truda>. – Дата доступа: 31.01.2024.

[12] Методика обеспечения эксплуатационной надёжности планируемых к разработке прикладных компьютерных программ для информационных систем / С. М. Боровиков [и др.] // BIG DATA and Advanced Analytics = BIG DATA и анализ высокого уровня: сборник научных статей VIII Международной научно-практической конференции, Республика Беларусь, Минск, 11-12 мая 2022 года. – Минск : Бестпринт, 2022. – С. 162–173.

Авторский вклад

Лэ Ван Там – разработка моделей по оценке ожидаемой надёжности прикладных компьютерных программ разных областей применения

Боровиков Сергей Максимович – общее руководство исследованиями и разработка моделей по оценке надёжности прикладных компьютерных программ.

Дик Сергей Константинович – постановка задачи и разработка модели определения прогнозного значения начальной плотности ошибок в прикладной компьютерной программе.

Будник Артур Владимирович – постановка задачи по обеспечению эксплуатационной надёжности прикладных компьютерных программ, подготовка справочных параметров для оценки надёжности прикладных компьютерных программ для разных областей применения.

ACCELERATED ASSESSMENT OF THE RELIABILITY OF DEVELOPED APPLICATION SOFTWARE

Van Tam Le
*Research Assistant,
Hanoi Institute of
Technology, Vietnam,
Master of Engineering
and Technology*

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

S.K. Dick
*Associate Professor,
Department of
Electronic Engineering
and Technology, PhD of
Technical sciences,
Associate Professor*

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

Abstract. The results of research to assess the operational reliability of the software being developed, intended for use in various fields of human activity, including in the field of Big Data, have been clarified and systematized.

Information is provided on assessing the expected initial reliability of software and recommendations on assessing the duration of testing in order to ensure the operational reliability of application software planned for development for complex information systems.

Keywords: application software, predicted density fault, reliability models, testing.