

УДК 004.4' 242 : 004.6-027.45

АНАЛИЗ И ОЦЕНКА НАДЁЖНОСТИ ПРИКЛАДНЫХ КОМПЬЮТЕРНЫХ ПРОГРАММ



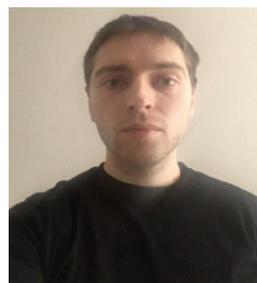
С.М. Боровиков
Доцент кафедры ПИКС БГУИР,
кандидат технических наук



С.С. Дик
Аспирант кафедры ПИКС БГУИР



В.Т. Лэ
Магистрант кафедры ПИКС БГУИР



К.И. Клинов
Студент ФКП БГУИР

Белорусский государственный университет информатики и радиоэлектроники,
Республика Беларусь
E-mail: bsm@bsuir.by

С.М. Боровиков

Доцент кафедры проектирования информационно-компьютерных систем БГУИР. 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. Руководитель разработки программных комплексов по автоматизированному расчёту и обеспечению надёжности электронных устройств: система АРИОН (2008-2009 гг.), система АРИОН-плюс (2011-2015 гг.).

С.С. Дик

Окончил Белорусский государственный университет информатики и радиоэлектроники (2016 г.), в настоящее время является аспирантом этого университета, магистр технических наук. Работает в ООО «Itransition» в должности Project Manager. Проводит научные исследования по созданию и внедрению программного обеспечения в процессы подготовки спортсменов.

Ван Там Лэ

Гражданин Вьетнама. Окончил Белорусский государственный университет информатики и радиоэлектроники (2018 г.), в настоящее время является магистрантом кафедры проектирования информационно-компьютерных систем этого университета. Выполняет магистерскую диссертацию по разработке методики, предназначенной для оценки надёжности прикладных программных средств на ранних этапах их проектирования.

К.И. Клинов

Окончил филиал Белорусского национального технического университета «Борисовский государственный политехнический колледж» (факультет «Приборостроение»). В настоящее время является

студентом заочной формы обучения Белорусского государственного университета информатики и радиоэлектроники. Работает разработчиком программного обеспечения в ЧП «Гузикс» (г. Минск).

Аннотация. Развитие и совершенствование технологий Big Data обуславливают необходимость разработки новых прикладных компьютерных программ по обработке больших объемов данных. Заинтересованные специалисты хотели бы знать ожидаемый уровень надёжности компьютерных программ до начала написания их кода на языке программирования. В работе предложен возможный подход к оценке надёжности прикладных компьютерных программ, основываясь на прогнозном объеме будущей программы (числе строк кода) и области её применения. На основе рассмотрения экспериментальных данных об эксплуатационной надёжности используемых компьютерных программ предложена модель оценки ожидаемой надёжности планируемых к разработке компьютерных программ. Модель учитывает основные характеристики будущей компьютерной программы, квалификацию программистов, выполнение процедуры тестирования, а также основные факторы, определяющие эксплуатационную надёжность компьютерных программ: быстродействие используемых компьютеров, изменчивость входных данных и рабочую нагрузку, которую эксплуатационная среда будет оказывать на компьютерную программу.

Ключевые слова: прикладные компьютерные программы, модель надёжности, оценка надёжности до написания кода.

Введение. При создании аналитических систем для анализа больших данных (big data) возникает необходимость выполнения ряда этапов, среди которых важнейшими являются следующие:

- выбор целевой переменной (англоязычный вариант – target), описывающей эффективность деятельности компаний по организации и оказанию управленческих, образовательных, коммерческих, медицинских и других услуг, производству продукции;
- определение набора признаков (англоязычный вариант – features), позволяющих предсказывать (прогнозировать) значения целевой переменной с высокой достоверностью и минимальными последующими рисками для компании или организации;
- получение данных в виде значений признаков и целевой переменной; эти данные будут обрабатываться с помощью аналитической системы, а по результатам обработки будет прогнозироваться целевая переменная, которую можно будет использовать для оперативного принятия решений в соответствующей сфере деятельности людей;
- выбор алгоритма анализа данных для принятия решений;
- выбор имеющихся приложений или разработка специальных прикладных программ для обработки данных;
- оценка эффективности разработанной аналитической системы.

Отметим, что все указанные этапы создания аналитических систем важны и от успешного выполнения этапа зависит в конечном итоге эффективность создаваемой системы обработки больших данных. Например, подбор данных для обработки и алгоритмов анализа может стать не меньшей проблемой, чем другие этапы, так как отсутствует понимание, какие данные следует собирать и хранить, а какие можно игнорировать.

Некоторые возможные подходы к созданию аналитических систем обработки больших данных и прогнозирования целевой переменной были рассмотрены в [1, 2]. Со специальными методами и алгоритмами анализа больших данных можно ознакомиться в [3]. Используя рекомендации специалистов в области обработки больших данных, в конечном итоге можно определиться с данными и алгоритмами их обработки. Далее возникает следующая непростая задача: выбор приложений для обработки данных и/или разработка прикладного программного средства. Причём, используемые компьютерные программы должны отвечать высоким требованиям надёжности, поскольку надо быстро обрабатывать большие данные (big data), получать достоверные результаты и оперативно их использовать для управленческих и других действий.

В данной работе рассмотрены подходы к оценке надёжности прикладных программ на ранних этапах их разработки, до написания кода на языке программирования.

Актуальность. Аналитические системы, используемые для обработки больших данных относятся к классу сложных информационно-компьютерных систем, в которых вклад программного обеспечения в ненадёжность систем может составлять 40 и более процентов [4]. Для принятия решения о целесообразности создания и эффективности функционирования аналитических систем необходимо на ранних этапах их проектирования оценить ожидаемую надёжность разрабатываемого программного обеспечения. Методы оценки надёжности программного обеспечения, описанные в технической литературе, исходят из того, что написана компьютерная программа, устранены ошибки, обусловленные нарушениями правил языка программирования, и имеются определённые данные о тестировании программного средства (компьютерной программы и документации к ней). Но разработчиков аналитических систем и программного обеспечения к ним во многих случаях интересует прогнозный уровень надёжности компьютерных программ ещё до написания кода на языке программирования. Метод оценки надёжности компьютерных программ до написания их кода был предложен в [5, 6]. Метод использует статистические модели оценки ожидаемого числа возможных ошибок в компьютерной программе (некоторые специалисты используют термин «дефекты проектирования программы»). Однако метод нуждается в уточнении и адаптации к быстродействию используемых компьютеров, учёту изменчивости входных данных, а также рабочей нагрузки, которую будет воспринимать компьютерная программа со стороны эксплуатационной среды (ввод-вывод данных и нахождение этих операций в очереди и наличие состояний ожиданий, загрузка-выгрузка программы и/или её модулей из памяти и т.д.).

Исходные положения и предпосылки. В соответствии с ГОСТ 27.205-1997 [7] 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 (то есть 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 [8]. 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.

Вероятность безошибочного выполнения компьютерной программы (P) в течение компьютерного (процессорного) времени $t_{\text{комп}}$ обработки данных может быть определена в предположении экспоненциального распределения времени проявления ошибок:

$$P(t_{\text{комп}}) = \exp(-\lambda t_{\text{комп}}),$$

где λ – ожидаемая интенсивность отказов компьютерной программы; представляет собой среднее число появлений ошибок в единицу времени; в качестве единицы времени в данном случае следует принимать один час компьютерного (процессорного) времени.

Согласно техническому отчёту [8], для оценки ожидаемой интенсивности отказов компьютерной программы может быть использована модель

$$\lambda = K_{\text{тр}} \frac{N_{\text{ош}}}{L} = K_{\text{тр}} F, \quad (1)$$

где λ – ожидаемая интенсивность отказов компьютерной программы (размерность: 1/ч); $K_{\text{тр}}$ – коэффициент, показывающий, как при использовании компьютерной программы по назначению скрытые дефекты проектирования программы количественно трансформируются (преобразуются) в интенсивность отказов (размерность: строка кода/ошибок в час); $N_{\text{ош}}$ – число оставшихся ошибок в компьютерной программе; L – объём текста кода компьютерной программы в строках кода; F – ожидаемая плотность оставшихся ошибок в компьютерной программе (размерность: ошибок/строка кода).

В [8] приводятся экспериментальные данные, на основе которых получены значения коэффициента преобразования $K_{\text{тр}}$ в зависимости от назначения, области применения компьютерной программы (таблица 1).

Таблица 1. – Экспериментальные данные о надёжности компьютерных программ

Область применения компьютерной программы	Число программ	Суммарное число строк кода	Средняя плотность ошибок F на одну программу, ошибок/строка кода	Средняя эксплуатационная интенсивность отказов λ , 1/ч	Значение $K_{\text{тр}}$, строка кода/ошибок в час
1. Авиация	7	540617	0,013	0,08	6,28
2. Мониторинг и обеспечение безопасности	21	1793831	0,009	0,0109	1,2
3. Телекоммуникации, мобильные электронные устройства	5	88252	0,008	0,108	13,8
4. Управление производственными процессами	2	140090	0,002	0,0076	3,8
5. Автоматизированные системы управления (в различных областях и сферах деятельности людей)	12	2575427	0,009	0,198	23
6. Инструменты разработки программ, моделирование, испытательные стенды, обучение	6	193435	0,014	0,236*	16,9
Среднее	53	5331652	0,0094	0,1	10,6

Примечание. * – получено пересчётом средней интенсивности отказов во время тестирования, составляющей 11,8 ч⁻¹.

Значения коэффициентов преобразования $K_{\text{тр}}$, полученные по данным таблицы 1, приведены в последнем столбце таблицы. Эти значения рассчитаны на основе выражения (1) как $K_{\text{тр}} = \lambda / F$. Например, для строки «Среднее» коэффициент преобразования $K_{\text{тр}}$ равен $0,1 / 0,0094 \approx 10,6$.

Приведённые в таблице 1 коэффициенты $K_{\text{тр}}$ соответствуют случаю выполнения программ на компьютере с 32-разрядным процессором, быстродействие: 10 миллионов

операций в секунду (англоязычный вариант: 10 million instructions per second, кратко – 10 MIPS).

Механизм преобразования между плотностью ошибок и интенсивностью отказов программы основан на следующем. Оператор с ошибками не приведёт к отказу ни при каких обстоятельствах, пока он не будет выполнен, то есть до тех пор, пока он не повлияет на содержимое памяти или на управление компьютером. Будем предполагать, что в больших компьютерных программах (объём: тысячи–десятки тысяч строк кода) ошибки всегда существуют. Тогда на вероятность возникновения отказа влияют три характеристики среды выполнения программы:

- производительность компьютера (быстродействие, пропускная способность);
- изменчивость данных и состояний управления;
- рабочая нагрузка, которую эксплуатационная среда будет оказывать на компьютерную программу.

При прочих равных условиях программа, постоянно выполняющаяся на компьютере с большим быстродействием, будет показывать более высокую интенсивность отказов, чем та же программа, выполняемая на более медленном компьютере. Показатели отказов программ, выраженные с учётом компьютерных или процессорных часов, являются наиболее полезными показателями надёжности в данной эксплуатационной среде. Но следует не забывать, что подверженность программ проявлениям ошибок (отказам) зависит от числа прогонов, а не от течения времени. Таким образом, если один прогон программы с заданным набором данных занимает 1 секунду на компьютере *A* и 0,1 секунды на компьютере *B*, то проявление ошибки в единицу времени (интенсивность отказов) для программы, выполняемой на быстром компьютере (*B*), будет в десять раз больше, чем в случае её выполнения на медленном компьютере (*A*). При прочих равных условиях можно ожидать, что интенсивность отказов (выраженная в единицах общего времени) для компьютера *B* будет в десять раз больше, чем при запуске программы на компьютере *A*.

Неисправная программа, выполняемая даже на очень быстром компьютере, не будет иметь отказов программного обеспечения, если она постоянно работает с набором данных, который не приводил к появлению ошибок. С другой стороны, введение преднамеренной изменчивости во входные данные, как это делают при тестировании, ускорит возникновение отказов программы.

Рабочая нагрузка компьютерной системы влияет на частоту отказов программного обеспечения. Было обнаружено, что при очень высокой рабочей нагрузке интенсивность отказов может увеличиться более чем на порядок по сравнению с базовой, низкой рабочей нагрузкой программы. Экспериментально установлено, что между нагрузкой компьютерной программы и интенсивностью отказов имеет место очень тесная положительная корреляция [8]. Базовая концепция, лежащая в основе этого явления, заключается в том, что в условиях большой рабочей нагрузки встречаются более необычные ситуации (загрузка и выгрузка программы из памяти, нахождение операций ввода-вывода в очереди, наличие состояний ожидания и т.д.). И программист компьютерной программы, возможно, не предвидел все эти ситуации. Кроме того, системное программное обеспечение будет чаще выходить из строя при более частом его использовании.

Результаты. Интенсивность отказов компьютерной программы до начала этапа тестирования может быть определена на основе модели Муса по формуле [9]

$$\lambda_0 = CVN_{\text{нач}} = C \frac{R}{LE_R} N_{\text{нач}}, \quad (2)$$

где C – коэффициент пропорциональности; V – возможное непрерывное число прогонов компьютерной программы (команд программы) в течение одного часа; $N_{\text{нач}}$ – начальное число ошибок компьютерной программы (до начала тестирования); R – производительность компьютера (быстродействие); L – объём компьютерной программы в строках кода; E_R – коэффициент расширения кода (увеличения числа команд программы), зависит от используемого языка программирования (для Assembler $E_R = 1,0$; для Macro Assembler, $E_R = 1,5$; для Си $E_R = 2,5$; для Fortran, Cobal $E_R = 3,0$; для Ada $E_R = 4,5$; для Си++ $E_R = 6,0$).

Авторы работ в области надёжности программного обеспечения называют коэффициент пропорциональности C по-разному. В работах [9, 10] коэффициент C назван коэффициентом проявления ошибок программы и приводятся его примерные численные значения: от $1,5 \cdot 10^{-7}$ до $4 \cdot 10^{-7}$. В работе [11] этот коэффициент называют постоянной, связанной с динамическим строением программы и её выполнением на различных платформах вычислительных систем. Указывается диапазон возможных значений от $1,4 \cdot 10^{-7}$ до $10,6 \cdot 10^{-7}$ и отмечается, что по умолчанию принимают значение $C = 4,2 \cdot 10^{-7}$. Размерность коэффициента C : 1/ошибка.

С учётом того, что для числа оставшихся в компьютерной программе ошибок N справедливо равенство

$$N = LF, \quad (2)$$

выражение (2) для λ_0 можно представить в виде

$$\lambda_0 = C \frac{R}{E_R} F, \quad (3)$$

где F – ожидаемая плотность оставшихся ошибок в компьютерной программе до выполнения тестирования программного средства.

При выполнении тестирования компьютерной программы и устранения части скрытых ошибок уменьшается ожидаемая плотность оставшихся ошибок F выражения (3). С учётом этого при окончании тестирования значение λ_0 трансформируется в эксплуатационную интенсивность отказов $\lambda_{\text{эсп}}$.

Коэффициент C обычно считают постоянным для конкретной компьютерной программы на всех этапах её разработки.

Из сравнения выражений (1) и (3) видно, что

$$K_{\text{тр}} = C \frac{R}{E_R}.$$

Используя данные таблицы 1, определим средние значения коэффициента пропорциональности C для компьютерных программ различных отраслей (областей применения). Будем считать, что программы были написаны на языке программирования Си, поэтому примем $E_R = 2,5$. Согласно [8], при выполнении (прогоне) программ использовались компьютеры с быстродействием процессора $R = 10$ MIPS (million instructions per second – миллионов операций в секунду). Подсчитанные значения коэффициента C для разных областей применения компьютерных программ приведены в таблице 2.

Таблица 2. – Значения коэффициента пропорциональности C

Область применения компьютерной программы	Авиация	Мониторинг и обеспечение безопасности	Телекоммуникации, мобильные устройства	Управление производственными процессами	Автоматизированные системы управления	Разработка программ, моделирование, обучение	Среднее
Коэффициент C , 1/ошибка	4,36E-10	8,33E-11	9,58E-10	2,64E-10	1,60E-09	1,17E-09	7,36E-10
Коэффициент K_{Σ}	5,2	1,0	11,5	3,2	19,2	14,1	8,84

Значения коэффициента C оказались разными в зависимости от назначения компьютерной программы. Это является закономерным поскольку коэффициент C учитывает влияние на интенсивность отказов программ степени изменчивости входных данных и рабочую нагрузку на программу.

Представим коэффициент C в виде произведения

$$C = C_B \cdot K_{вх} \cdot K_n = C_B \cdot K_{\Sigma}, \quad (4)$$

а величины, входящие в выражение (4), будем называть как: 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_{Σ} – 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 примем значение коэффициента C для компьютерных программ, используемых для мониторинга и обеспечения безопасности. Пользуясь таблицей 2 (3-й столбец), можно записать: $C_B = 8,33 \cdot 10^{-11}$ 1/ошибка. Для программ этого назначения суммарный коэффициент увеличения интенсивности отказов принят $K_{\Sigma} = 1,0$.

Для определения ожидаемой надёжности компьютерной программы после выполнения её тестирования будем предполагать, что процедура тестирования выполняется общепринятыми методами, и при этом её суммарная продолжительность составляет не менее 40 процентов общего времени, отводимого на разработку компьютерной программы. Согласно работе [8], экспериментальные данные показали, что в таких случаях интенсивность отказов компьютерных программ за счёт выполнения тестирования уменьшается примерно в семь раз относительно начальной интенсивности отказов λ_0 . Тогда модель для определения ожидаемой интенсивности отказов компьютерной программы, прошедшей тестирование, может быть представлена в виде

$$\lambda_{\text{экс}}^{(i)} = 0,14 C_B V N_{\text{нач}} K_{\Sigma}^{(i)}, \quad (5)$$

где $\lambda_{\text{экс}}^{(i)}$ – ожидаемая (для i -й области применения, см. таблицу 2) эксплуатационная интенсивность отказов компьютерной программы, прошедшей тестирование; $K_{\Sigma}^{(i)}$ – суммарный коэффициент увеличения интенсивности отказов для компьютерных программ i -й области применения (см. таблицу 2); множитель 0,14 – коэффициент, учитывающий

уменьшение интенсивности отказов в семь раз за счёт выполнения процедуры тестирования компьютерной программы ($1/7 \approx 0,14$).

Прогнозное число возможных прогонов программы (команд программы) в течение одного часа определяется в соответствии с выражением (2):

$$V = \frac{R}{LE_R}.$$

Начальное число ошибок компьютерной программы (до начала тестирования) $N_{\text{нач}}$ можно определить в соответствии с работой [2] как

$$N_{\text{нач}} = F \cdot L = A \cdot D \cdot S \cdot L, \quad (6)$$

где A – базовая плотность дефектов для компьютерных программ, используемых в данной отрасли; D – метрика, учитывающая специфику организации, разрабатывающей программное обеспечение, а также опыт и квалификацию программистов; S – метрика, учитывающая характеристики программного средства (категорию сложности, степень новизны, язык программирования, используемую операционную систему, компьютерную сеть, степень использования стандартных модулей).

Выбор и/или определение величин A , D , и S формулы (6) рассмотрены в работе [2].

Заключение. Предложенный подход определения ожидаемой надёжности компьютерных программ основан на использовании статистических (экспериментальных) данных о свойствах используемых программ и даёт возможность приближённо оценить ожидаемую надёжность планируемого к разработке программного средства, в том числе используемого для анализа больших данных. Такой расчёт полезен, так как позволяет ещё до разработки программного средства получить представление о его ожидаемой надёжности и при необходимости разработать мероприятия по обеспечению требуемого уровня надёжности.

Список литературы

[1]. Batura M., Dzik S., Tsyrelchuk I., Borovikov S. Big Data Volumes and Some Approaches to the Creation of Corporate Analytical Systems / BIG DATA and Advanced Analytics. Использование BIG DATA для оптимизации бизнеса и информационных технологий: сб. материалов II Междунар. науч.-практ. конф. (Минск, Республика Беларусь, 15–17 июня 2016 года) / редкол. : М.П. Батура [и др.]. – Минск : БГУИР, 2016. – С. 74-80.

[2]. Боровиков, С. М. Большие данные и принципы разработки аналитических систем / С. М. Боровиков, С. К. Дик, С. С. Дик // BIG DATA and Advanced Analytics = BIG DATA и анализ высокого уровня : сборник материалов V Международной научно-практической конференции, Минск, 13–14 марта 2019 г. В 2 ч. Ч. 2 / Белорусский государственный университет информатики и радиоэлектроники; редкол. : В. А. Богуш [и др.]. – Минск, 2019. – С. 167-171.

[3]. Что такое Big Data (большие данные) в маркетинге: проблемы, алгоритмы, методы анализа [Электронный ресурс]. – Режим доступа : <http://lpgenerator.ru/blog/2015/11/17/chto-takoe-big-data-bolshie-dannye-v-marketinge-problemy-algoritmy-metody-analiza/> (дата обращения: 27.02.2020).

[4.] Боровиков, С. М. Метод оценки ожидаемой надёжности прикладных компьютерных программ систем медицинской электроники / С. М. Боровиков, С. С. Дик // Доклады БГУИР. – 2018. – № 7 (117). – С. 112-117.

[5]. Боровиков, С. М. Прогнозирование ожидаемой надёжности прикладных программных средств с использованием статистических моделей их безотказности / С. М. Боровиков, С. С. Дик // BIG DATA Advanced Analytics: collection of materials of the fourth international scientific and practical conference, Minsk, Belarus, May 3 – 4, 2018 / editorial board: M. Batura [etc.]. – Minsk, BSUIR, 2018. – P. 348 - 354.

[6]. Боровиков, С. М. Возможный подход к оценке надёжности прикладных программных средств для технологий Big Data / С. М. Боровиков, Лэ Ван Там, С. С. Дик // BIG DATA and Advanced Analytics = BIG DATA и анализ высокого уровня : сборник материалов V Международной научно-практической конференции, Минск, 13–14 марта 2019 г. В 2 ч. Ч. 2 / Белорусский государственный университет информатики и радиоэлектроники; редкол. : В. А. Богуш [и др.]. – Минск, 2019. – С. 77-83.

[7]. ГОСТ 27.205-97. Надёжность в технике. Проектная оценка надёжности сложных систем с учётом технического и программного обеспечения и оперативного персонала. Основные положения. – Минск : Госстандарт Республики Беларусь, 2005. – 22 с.

[8]. Software reliability, measurement and testing guidebook for software reliability measurement and testing: RL-TR-92-52, Vol II (of two) Final technical report April 1992/ Science Applications International Corp. (SAIC), Research Triangle Institute (RTI). Rome Laboratory Air Force Systems Command Griffiss Air Force Base NY 13441-5700.

[9]. Программирование и основы алгоритмизации. Исследование методов оценки и повышения надёжности программного обеспечения [Электронный ресурс]. – Режим доступа : <http://www.azovikdip.ru/index18.php> (дата обращения: 27.02.2020).

[10]. Модели надёжности программного и информационного обеспечения [Электронный ресурс]. – Режим доступа : <https://refdb.ru/look/2279745-pall.html> (дата обращения: 27.02.2020).

[11]. Модели оценки надёжности программных средств [Электронный ресурс]. – Режим доступа : <https://helpiks.org/4-73503.html> (дата обращения: 27.02.2020).

RELIABILITY OF APPLIED SOFTWARE

S.M. Borovikov,

PhD, associate professor of the department of Information and Computer Systems Design of the Belarusian State University of Informatics and Radioelectronics

S.S. Dick,

master of engineering, PG student of the Belarusian state university of informatics and Radioelectronics

Van Tam LE

master student of the department of Information and Computer Systems Design of the Belarusian State University of Informatics and Radioelectronics

K.I. Klinov

student of the Belarusian State University of Informatics and Radioelectronics

*Belarusian State University of Informatics and Radioelectronics, Republic of Belarus
E-mail: bsm@bsuir.by*

Abstract. The development and improvement of Big Data technologies necessitates the development of new applied computer programs for processing large amounts of data. Interested specialists would like to know the expected level of reliability of computer programs before writing their code in a programming language. A possible approach to assessing the reliability of applied computer programs based on the predicted volume of the future program (number of lines of code) and the area of its application is proposed. Based on the consideration of experimental data on the operational reliability of the computer programs used, a model for evaluating the expected reliability of computer programs planned for development is proposed. The model takes into account the main characteristics of the future computer program, the qualifications of programmers, and the implementation of the testing procedure..

Keywords: applied computer programs, reliability model, reliability assessment before writing code.