

УДК 004.4' 242 : 004.6-027.45

ВОЗМОЖНЫЙ ПОДХОД К ОЦЕНКЕ НАДЁЖНОСТИ ПРИКЛАДНЫХ ПРОГРАММНЫХ СРЕДСТВ ДЛЯ ТЕХНОЛОГИЙ BIG DATA



С.М. Боровиков

Доцент кафедры проектирования информационно-компьютерных систем Белорусского государственного университета информатики и радиоэлектроники, кандидат технических наук



Ван Там Лэ

Магистрант кафедры проектирования информационно-компьютерных систем Белорусского государственного университета информатики и радиоэлектроники



С.С. Дик

Аспирант кафедры проектирования информационно-компьютерных систем Белорусского государственного университета информатики и радиоэлектроники

*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.).

Ван Там Лэ

Гражданин Вьетнама. Окончил Белорусский государственный университет информатики и радиоэлектроники (2018 г.), 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.

С.С. Дик

Окончил БГУИР (2016 г.), в настоящее время является аспирантом этого университета, магистр технических наук. Работает в ООО «Syber Genesis» в должности начальника отдела разработки программного обеспечения. Проводит научные исследования по созданию и внедрению информационно-компьютерных систем в процессы подготовки спортсменов.

Аннотация. Развитие и совершенствование технологий Big Data обуславливают необходимость разработки новых подходов и методов по обработке больших объёмов структурированных и/или неструктурированных данных. Практическая реализация новых методов требует наличия прикладных программных средств, которые должны обладать достаточным уровнем надёжности. Заинтересованные специалисты хотели бы знать ожидаемый уровень надёжности программных средств до начала написания кода компьютерных программ на языке программирования. В работе предложен возможный подход к оценке надёжности прикладных программных средств на ранних этапах их разработки.

Ключевые слова: прикладные программные средства, ранние этапы разработки, прогнозируемая надёжность.

Введение. Одной из задач технологии обработки больших данных (big data) является создание различных аналитических отчётов и получение прогнозов, которые будут использованы компаниями и организациями в своей деятельности. Эффективность деятельности компаний по производству продукции, оказанию услуг (коммерческих, образовательных, медицинских и др.) может быть описана целевой переменной (англоязычный вариант – target), одной или несколькими. Во многих случаях в задачах прогнозирования с использованием больших данных целевая переменная может предсказываться на основе набора признаков (англоязычный вариант – features). Причём этот набор должен быть исчерпывающим с точки зрения прогнозирования целевой переменной с высокой достоверностью [1]. Это позволит оперативно принимать решения в соответствующей сфере деятельности людей. Подбор данных для обработки, выбор алгоритма их анализа и получения прогноза целевой переменной могут вызвать определённые проблемы, так как в большинстве случаев отсутствует чёткое понимание, какие данные следует собирать и хранить, а какие можно игнорировать. С помощью специалистов с используемыми данными и алгоритмами их обработки в конечном итоге можно определиться. После этого возникает следующая непростая задача: разработка программного средства. Причём, это средство должно отвечать требованию надёжности, поскольку, используя большие данные (big data), надо быстро получать результаты и оперативно их применять.

Актуальность. В сложных информационно-компьютерных системах, к которым относятся аналитические системы, использующие большие данные, вклад программного обеспечения в ненадёжность систем может составлять 40 и более процентов [2].

Известны десятки моделей надёжности программных средств, разработанные в 1970-2006 гг. [3]. Однако, сейчас ситуация такова, что каждая фирма-разработчик программного средства или иная организация, заинтересованная в количественной оценке надёжности программного средства, вынуждены решать проблему для себя заново, учитывая опыт и квалификацию программистов и тестировщиков, а также специфику создаваемых аналитических систем, использующих большие данные. Необходимость решения данной задачи связана с узкой специализацией большинства предложенных моделей.

Разработанные ранее модели и методы оценки надёжности прикладных программных средств носят ограниченный характер, предполагают наличие определённых данных о тестировании разрабатываемой компьютерной программы после устранения в ней ошибок, вызванных нарушениями правил языка программирования. В большинстве случаев разработчиков программного обеспечения для аналитических систем интересует ожидаемый уровень надёжности прикладных компьютерных программ ещё до написания их программного кода. В [4, 5] был предложен метод оценки надёжности прикладных программ на ранних этапах их разработки. Метод основан на использовании статистических моделей определения ожидаемого числа дефектов (некоторые специалисты используют термин «ошибки») в компьютерной программе. Достоверность результатов, которые обеспечивает метод, в значительной степени зависит от используемых коэффициентов, характеризующих опыт и квалификацию программистов и тестировщиков. Необходимость получения значений этих коэффициентов и отсутствие данных о их достоверности ограничивают применение метода для оценки ожидаемой надёжности прикладных программных средств и аналитических систем в целом. Это обуславливает необходимость совершенствования подходов и методов оценки надёжности прикладного программного обеспечения аналитических систем, использующих большие данные.

Подход к оценке ожидаемой надёжности прикладных программных средств. Согласно ГОСТ 27.205-1997 [6] о проектной надёжности разрабатываемого программного

средства будем судить по интенсивности проявления дефектов, обусловленных ошибками его проектирования.

Для оценки ожидаемой надёжности прикладного программного средства, прошедшего тестирование, предлагаемая модель

$$\lambda_{п.тест} = \lambda_0 \cdot K_{тест}(S_1, S_2, S_3), \quad (1)$$

где $\lambda_{п.тест}$ – ожидаемая интенсивность проявления дефектов программного средства, прошедшего тестирование; λ_0 – ожидаемая начальная интенсивность проявления оставшихся дефектов в программном средстве (до начала этапа тестирования); $K_{тест}(S_1, S_2, S_3)$ – поправочный коэффициент уменьшения интенсивности проявления дефектов программного средства за счёт выполнения его тестирования, коэффициент учитывает три метрики: технологию тестирования (S_1), время тестирования (S_2), опыт и квалификацию тестирующих (S_3).

Величину λ_0 в выражении (1) предлагается оценивать с учётом предполагаемого объёма компьютерной программы в тысячах строк кода (англоязычный вариант – KLOC), используя модель

$$\lambda_0 = C \cdot F \cdot L_{KLOC}, \quad (2)$$

где C – коэффициент пропорциональности, показывающий как оставшиеся в программе ошибки трансформируются в интенсивность проявления дефектов при использовании программного средства по назначению, размерность коэффициента – 1/ошибок за единицу времени; F – ожидаемая плотность оставшихся дефектов, приходящаяся на 1000 строк кода компьютерной программы после устранения в ней ошибок программирования (до выполнения тестирования); L_{KLOC} – предполагаемый объём компьютерной программы в тысячах строк кода.

Коэффициент C выражения (2) может быть получен по модели Муса [7], либо по данным о надёжности действующих программных средств, к которым предъявлялись требования (к написанию кода и к выполнению тестирования) как и к программам, используемым в аналитических системах, обрабатывающих большие данные. Для наглядности покажем примером, как получить прогнозное значение коэффициента C , используя информацию о программном средстве Oracle Content Server for Windows 8. По данным [8] исходный код этого программного средства составляет $L = 1\,300$ тысяч строк. Количество ошибок в тексте программы принято на уровне минимального из оценочных значений, а именно, 0,04 ошибки на 1000 строк кода после выполнения тестирования, полученная средняя интенсивность проявления дефектов (ошибок) программного средства после завершения этапа тестирования составила $\lambda_{ср} = 0,00124 \text{ ч}^{-1}$ [8]. Для определения значения C воспользуемся общепринятым выражением для интенсивности проявления дефектов λ [3-5, 7-9]:

$$\lambda = C \cdot N_{\text{ош}}, \quad (3)$$

где $N_{\text{ош}}$ – число дефектов (ошибок), оставшихся в программном средстве.

Можно убедиться, что для рассматриваемого приложения Oracle Content Server число дефектов (ошибок), оставшихся в программном средстве после выполнения процедуры тестирования составит $N_{п.тест} = 52$ ошибки.

Принимая $\lambda = \lambda_{ср} = 0,00124 \text{ ч}^{-1}$, а $N_{\text{ош}} = N_{п.тест} = 52$ ошибки, с учётом выражения (3) получим

$$C = 2,385 \cdot 10^{-5} \text{ 1/ошибок в час.}$$

Этим значением коэффициента C можно пользоваться при оценке ожидаемой надёжности разрабатываемых программных средств для аналитических систем, обрабатывающих большие данные.

Значение плотности дефектов F выражения (2) предлагается получать с помощью модели RL-92-52 («Римская модель») [9], основанной на использовании метрик программного обеспечения:

$$F = A \cdot D \cdot S, \quad (4)$$

где F – прогнозируемая плотность дефектов, приходящихся на 1000 строк кода разрабатываемой программы; A – метрика, учитывающая тип или назначение программного средства; D – метрика, учитывающая среду разработки программного обеспечения; S – метрика, учитывающая характеристики программного средства.

Метрика A представляет собой среднюю или базовую плотность дефектов для программных средств, используемых в данной отрасли (сфере деятельности людей), и рассматривается в качестве отправной точки для получения прогноза значения F (таблица 1). Метрики D и S являются фактически поправочными коэффициентами для метрики A , и каждая из них может иметь значение меньше единицы (1,0), если окружающая среда или реализация программного средства имеют тенденцию уменьшать плотность дефектов или значение больше единицы, если они имеют тенденцию увеличивать плотность. Эти метрики эквивалентны «пифакторам» в методике расчёта надёжности электронных устройств по MIL HDBK 217F, а также поправочным коэффициентам K_i при прогнозировании эксплуатационной надёжности элементов с использованием систем АСРН (Россия) или АРИОН (Республика Беларусь) [10-12].

Таблица 1

Плотность дефектов проектирования на 1000 строк программного кода [9]

Область применения (тип) программного обеспечения	Значение A , дефектов/тыс. строк кода
1. Электронные системы, используемые в авиации	12,8
2. Телекоммуникационные системы, системы экологического мониторинга	9,2
3. Мобильные электронные устройства	7,8
4. Управление производственными процессами	1,8
5. Автоматизированные системы управления (в различных областях и сферах деятельности людей)	8,5
6. Инструменты разработки программного обеспечения, моделирование, испытательные стенды, обучение	12,3

Значение метрики A для программного обеспечения, разрабатываемого для аналитических систем, использующих большие данные, может заметно превышать числа, указанные в строках 1 или 5 таблицы 1 (примерно в 1,5 ... 3 раза – по мнению авторов). Для каждой конкретной аналитической системы требуется экспертная оценка выбираемого значения метрики A .

Значение метрики D может быть выбрано с учётом того, что организация, заинтересованная в создании аналитической системы, заказывает разработку программного обеспечения специализированной IT-компании, программисты которой имеют опыт по разработке программ, но прямо не связаны с пользователями программного средства. В этом случае необходимо принять $D = 1$. В других случаях необходимо воспользоваться рекомендациями документа RL-92-52 или (при необходимости) дать экспертную оценку значению этой метрики с учётом условия $0,5 \leq D \leq 2$ [9].

Для определения метрики S выражения (4) предлагается модель вида

$$S = K_{\text{нов}} \cdot K_{\text{слож}} \cdot K_{\text{С.Р}} \cdot K_{\text{мод}} \cdot K_{\text{кв.прог}} \cdot K_{\text{вх.дан}}, \quad (5)$$

где K_i – поправочный коэффициент (иначе – подметрика), учитывающий влияние конкретного фактора (обозначен как i -й фактор) на значение метрики S и, следовательно, на интенсивность проявления дефектов программного средства; смысл i -го фактора указывает нижний индекс при коэффициенте.

В таблице 2 записаны поправочные коэффициенты вида K_i , даётся их пояснение, приводятся экстраполированные значения и/или рекомендации их получения.

Таблица 2

Поправочные коэффициенты K_i

Обозначение	Учитываемый фактор программного средства	Пояснение	Значение коэффициента
$K_{\text{слож}}$	Категория сложности	Три категории, шесть характеристик повышения сложности	От 1,00 до 1,47 по [13]
$K_{\text{нов}}$	Степень новизны	Три категории, включающие восемь разновидностей	От 0,63 до 1,58 по [13]
$K_{\text{С.Р}}$	Средства разработки (язык программирования, операционная система, компьютерная сеть)	Семь категорий, три разновидности	От 0,16 до 1,3 по [13]
$K_{\text{мод}}$	Степень использования стандартных модулей	Пять категорий	От 0,55 до 1,00 по [13]
$K_{\text{кв.прог}}$	Квалификация и опыт программиста	Три уровня	Примечание 1
$K_{\text{вх.дан}}$	Характер структурированности и однородности используемых входных данных	Четыре категории: структурированные однородные (коэффициент вариации до 33%), неструктурированные однородные, структурированные неоднородные, неструктурированные неоднородные	От 1 до 8 (экстраполированные значения). Для уточнения значений рекомендуется использовать экспертные оценки
$K_{\text{тест}} (S_1, S_2, S_3)$	Тестирование программного средства	Используются метрики, учитывающие технологию и условия тестирования	Определять методом, описанным в [9], с учётом примечания 2

Примечания: 1. Предлагается три уровня квалификации и опыта согласно [14]: Junior Developer (младший программист), Developer или Middle Developer (программист), Senior developer (ведущий разработчик). Значение $K_{\text{кв.прог}}$ для программиста (Middle Developer) соответствует $K_{\text{кв.прог}} = 1$. Для двух других категорий коэффициент определяется методом экспертных оценок.

2. Предлагается два уровня квалификации тестировщиков [15]: низкая квалификация, высокая квалификация. Определение значения метрики S_3 по каждой из квалификаций выполняется методом экспертной оценки отдельно в зависимости от размера фирмы (небольшая, большая)

В таблице 3 в качестве примера для выбора поправочных коэффициентов указываются значения коэффициента $K_{\text{мод}}$, учитывающего степень использования стандартных модулей в разрабатываемом программном средстве [13].

Таблица 3

Коэффициент, учитывающий степень использования стандартных модулей ($K_{\text{мод}}$)

Степень охвата реализуемых функций разрабатываемого программного обеспечения стандартными модулями	Значение $K_{\text{мод}}$
1. От 60% и выше	0,55
2. От 40 до 60%	0,65
3. От 20 до 40%	0,77
4. До 20%	0,9
5. Не используются стандартные модули для реализации функций разрабатываемого программного обеспечения	1

Отметим, что значения поправочных коэффициентов K_i , выбираемые по таблице 2, следует использовать в случаях, когда методом экспертной оценки затруднительно дать более достоверные оценки этим коэффициентам.

Заключение. Предложенный подход даёт возможность приближённо оценить ожидаемую надёжность разрабатываемого программного средства, используемого для обработки больших данных. Даже такой ориентировочный расчёт полезен, так как позволяет на раннем этапе разработки программного средства получить представление о его надёжности и сориентироваться в целесообразности проектирования и/или эффективности применения аналитической системы, использующей это программное средство. При необходимости могут быть разработаны организационно-технические мероприятия по обеспечению требуемого уровня надёжности программного средства и аналитической системы в целом.

Литература

- [1]. Фрэнк, Б. Укрощение больших данных. Как извлекать знания из массивов / Б. Фрэнк ; пер. с англ. – М. : Изд-во «Технологии развития ООО», 2014. – 352 с.
- [2]. Программное обеспечение – источник всех проблем [Электронный ресурс]. – Режим доступа: <http://www.williamspublishing.com/PDF/5-8459-0785-3/part1.pdf> (дата обращения: 20.01.2019).
- [3]. Модели оценки надёжности программных средств [Электронный ресурс]. – Режим доступа : <https://helpiks.org/4-73503.html> (дата обращения 20.01.2019).
- [4]. Боровиков, С. М. Оценка ожидаемой надёжности прикладных программных средств на начальном этапе их разработки / С. М. Боровиков, А. В. Будник, М. Н. Закривашевич // Современные средства связи : материалы XXII Междунар. науч.-техн. конф., 19–20 окт. 2017 года, Минск, Респ. Беларусь ; редкол. : А. О. Зеневич [и др.]. – Минск : Белорусская государственная академия связи , 2017. – С. 197-199.
- [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]. 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.
- [7]. Модели надёжности программного и информационного обеспечения [Электронный ресурс]. – Режим доступа : <https://refdb.ru/look/2279745-pall.html> (дата обращения: 25.01.2019).
- [8]. Чуканов, В. О. Надёжность программного обеспечения и аппаратных средств систем передачи данных атомных электростанций / В. О. Чуканов. – М. : МИФИ, 2008. – 168 с.

- [9]. 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.
- [10]. Reliability prediction of electronic equipment : Military Handbook MIL-HDBK-217F. – Washington : Department of defense DC 20301, 1995. – 205 p.
- [11]. Надёжность электрорадиоизделий, 2006 : справочник / С. Ф. Прытков [и др.] // научн. руководитель авторского коллектива С. Ф. Прытков. – М. : ФГУП «22 ЦНИИ МО РФ», 2008. – 641 с.
- [12]. Разработка методики прогнозирования надёжности электронных устройств для системы АРИОН / С. М. Боровиков [и др.]. Доклады БГУИР : электроника, материалы, технологии, информатика. – 2011. – № 4 (58). – С. 93-100.
- [13]. Укрупнённые нормы затрат труда на разработку программного обеспечения [Электронный ресурс]. – Режим доступа : <http://www.levonevski.net/pravo/norm2013/num28/d28024.html> (дата обращения: 25.01.2019).
- [14]. Ранги в профессии программист или профессиональный рост в IT карьере [Электронный ресурс]. – Режим доступа : <https://www.trn.ua/articles/6733> (дата обращения: 25.01.2019).
- [15]. Тестирование программного обеспечения. Базовый курс : версия книги 1.0.3 от 23.09.2015 [Электронный ресурс]. – Режим доступа : https://www.bsuir.by/m/12_108786_1_98216.pdf (дата обращения: 25.01.2019).

POSSIBLE APPROACH TO ESTIMATION OF RELIABILITY OF APPLIED SOFTWARE FOR BIG DATA TECHNOLOGY

S.M. BOROVIKOV

PhD, associate professor of the department of Information and Computer Systems Design 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

S.S. DZICK

Master of engineering, PG 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 approaches and methods for processing large volumes of structured and / or unstructured data. The practical implementation of new methods requires the availability of application software, which must have a sufficient level of reliability. Interested experts would like to know the expected level of reliability of software before writing computer code in a programming language. The paper proposes a possible approach to assessing the reliability of application software at the early stages of their development.

Keywords: application software, early development stages, expected reliability.