

УДК 004.45:615.47

МЕТОД ОЦЕНКИ ОЖИДАЕМОЙ НАДЕЖНОСТИ ПРИКЛАДНЫХ КОМПЬЮТЕРНЫХ ПРОГРАММ СИСТЕМ МЕДИЦИНСКОЙ ЭЛЕКТРОНИКИ

С.М. БОРОВИКОВ, С.С. ДИК

Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь

Поступила в редакцию 15 ноября 2018

Аннотация. Предлагается метод оценки ожидаемой надежности прикладных компьютерных программ для медицинских информационно-компьютерных систем на ранних этапах их разработки. Для получения ожидаемой эксплуатационной надежности компьютерной программы в методе используют поправочные коэффициенты, учитывающие важнейшие факторы, определяющие начальный уровень ее надежности (до начала тестирования), а учет процедуры тестирования на надежность программы выполняют с помощью поправочного коэффициента, зависящего от метрик, принимающих во внимание технологию тестирования, время тестирования, опыт и квалификацию специалистов.

Ключевые слова: компьютерная программа, ожидаемое число ошибок, тестирование, надежность.

Abstract. The authors propose a method for estimating the expected reliability of applied computer programs for medical computer systems in the early stages of their design. To obtain the expected operational reliability of a computer program, the method uses correction factors that take into account the most important factors determining the initial level of program reliability (before testing begins). Accounting for the impact on the reliability of the program testing procedures are performed using a correction factor that depends on metrics that take into account testing technology, testing time, experience and qualifications of specialists.

Keywords: computer program, expected number of errors, testing, reliability.

Doklady BGUIR. 2018, Vol. 117, No. 7, pp. 112-117
Method of evaluating the expected reliability
of applied computer programs of medical electronics systems
S.M. Borovikov, S.S. Dick

Введение

Под надежностью информационно-компьютерной системы, к которым относятся системы медицинского назначения, понимают свойство системы выполнять предписанные функции в течение заданного времени в определенных условиях эксплуатации. В общем случае надежность подобных систем определяется надежностью следующих трех разнородных частей:

- комплекса технических средств, включающего компьютеры, датчики, контрольно-измерительную, специальную контрольно-диагностическую, лечебную и другую аппаратуру;
- программного обеспечения, представляющего собой совокупность компьютерных программ (системных и прикладных) и документации по их использованию, служащих для управления работой компьютеров и специальной контрольно-диагностической и лечебной аппаратурой;
- оперативного персонала, функции которого выполняет медицинский технический персонал.

Примерный вклад этих частей в ненадежность информационно-компьютерной системы указан на рис. 1.

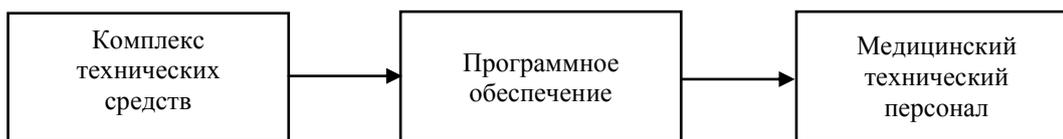


Рис. 1. Структурная схема надежности медицинской информационно-компьютерной системы

С точки зрения надежности части системы обычно соединены последовательно. Это означает: отказ любой части приводит к отказу системы в целом, что символизируется разрывом цепи, показанной на рис. 1. На этапе проектирования медицинской информационно-компьютерной системы важно оценить ее ожидаемую надежность с учетом наличия трех составных частей.

Актуальность разработки

Надежность комплекса технических средств ограничивается ошибками проектирования, производственными дефектами и частотой сбоев (зависит от физических процессов). Методы оценки надежности технических устройств достаточно хорошо разработаны (например, [1, 2]). Достигнут также определенный уровень в разработке методов и подходов в оценке надежности оператора при его работе в составе технических систем [3].

В сложных информационно-компьютерных системах, к которым относятся системы медицинского назначения, вклад программного обеспечения в ненадежность систем может составлять до 40 % и более [4]. Специалистами проделана немалая работа по определению ожидаемой надежности прикладных компьютерных программ, однако разработанные модели и методы носят ограниченный характер, предполагают наличие определенных данных о тестировании разработанной компьютерной программы после устранения в ней ошибок, вызванных нарушениями правил языка программирования. В большинстве случаев проектировщиков информационно-компьютерных систем и разработчиков программного обеспечения для этих систем интересует ожидаемый уровень надежности прикладных компьютерных программ еще до написания их программного кода. В [5] авторами был предложен метод оценки надежности прикладных программ на ранних этапах их разработки. Метод основан на использовании статистических моделей определения ожидаемого числа дефектов в компьютерной программе (некоторые специалисты используют термин ошибки). Достоверность результатов, которые обеспечивает метод, в значительной степени зависит от используемых коэффициентов, характеризующих опыт и квалификацию программистов и тестировщиков. Необходимость получения значений этих коэффициентов создает определенные проблемы при оценке ожидаемой надежности компьютерных программ и информационно-компьютерных систем в целом. Это обуславливает необходимость совершенствования подходов и методов оценки надежности прикладного программного обеспечения.

Теоретический анализ

Согласно ГОСТ 28806-90 под надежностью (en reliability) компьютерной программы понимают совокупность свойств, характеризующую способность программы сохранять заданный уровень пригодности в определенных условиях в течение заданного интервала времени. Ограничения ее уровня пригодности являются следствием дефектов, возникших в результате ошибок, допущенных при разработке программы. Количество и характер отказов компьютерной программы являются следствием этих дефектов (иначе, ошибок) и зависят от способа применения программы и выбираемых вариантов ее функционирования.

Очевидным является то, что реальный уровень надежности компьютерной программы зависит от ее объема. Среди специалистов по программированию в качестве единицы измерения объема программ используют количество строк кода (в англоязычном варианте: Lines Of Code – LOC).

Известны десятки моделей надежности компьютерных программ, разработанных в 1970–2017 гг. Однако в настоящее время ситуация такова, что каждая фирма –

разработчик компьютерной программы или иная организация, заинтересованная в количественной оценке надежности программы, вынуждены решать проблему для себя заново, учитывая опыт и квалификацию программистов и проектировщиков, специфику создаваемых систем, требования к их надежности и безопасности, экономические аспекты. Необходимость решения данной задачи связана с узкой специализацией большинства существующих моделей и трудностью выбора одной из них для конкретной компьютерной программы.

Для ориентировочной оценки ожидаемой надежности прикладных компьютерных программ на ранних этапах их разработки (до написания кода программ) авторами предлагается метод, в котором используются некоторые параметры будущей программы и данные о влиянии на ее надежность разных факторов. Подход, используемый в методе, основан на следующих основных гипотезах и допущениях:

- интенсивность проявления ошибки λ в программе прямо пропорционально числу оставшихся в ней ошибок $N_{\text{ош}}$;
- поправочные коэффициенты, учитывающие влияние фактора на надежность программы, пропорциональны трудозатратам;
- поправочные коэффициенты и/или метрики, учитывающие уровень квалификации и опыт специалистов, уменьшаются в геометрической прогрессии с повышением этого уровня.

О проектной надежности компьютерных программ согласно ГОСТ 27.205-2007 будем судить по интенсивности проявления дефектов, вызванных ошибками разработки.

Метод оценки ожидаемой надежности прикладного программного средства

Для оценки ожидаемой интенсивности проявления дефектов в компьютерной программе, прошедшей тестирование (обозначим через $\lambda_{\text{п.тест}}$), предлагается модель

$$\lambda_{\text{п.тест}} = \lambda_0 \cdot (K_{\text{нов}} \cdot K_{\text{слож}} \cdot K_{\text{С.Р}} \cdot K_{\text{мод}} \cdot K_{\text{кв.прог}} \cdot K_{\text{вх.дан}}) \cdot [K_{\text{тест}}(S_1, S_2, S_3)], \quad (1)$$

где λ_0 – ожидаемая начальная интенсивность проявления оставшихся дефектов в программе (до начала этапа тестирования), написанной программистом, имеющим достаточный опыт работы (не менее 3–5 лет); K_i – поправочный коэффициент, учитывающий влияние конкретного фактора (обозначен как i -й фактор) на число оставшихся дефектов в программе и, следовательно, на начальную интенсивность проявления дефектов; смысл i -го фактора указывает нижний индекс при коэффициенте K ; $K_{\text{тест}}(S_1, S_2, S_3)$ – поправочный коэффициент уменьшения интенсивности проявления дефектов в программе за счет выполнения ее тестирования, зависящий от трех метрик, в которых учитывается технология тестирования (S_1), время тестирования (S_2), опыт тестировщиков (S_3).

Для получения ожидаемого значения начальной интенсивности проявления дефектов в программе λ_0 используются модели Муса [6] и RL-92-52 («Римская модель») [7]:

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

где C – коэффициент пропорциональности, показывающий, как оставшиеся в программе дефекты трансформируются в интенсивность проявления дефектов при использовании программы по назначению; размерность коэффициента – $1/c$; F – ожидаемая плотность оставшихся дефектов, приходящаяся на 1000 строк кода программы до начала этапа тестирования (после устранения ошибок, обусловленных нарушением правил языка программирования); L_{KLOC} – объем программы в тысячах строк кода.

Согласно модели Муса [6] коэффициент C определяется как

$$C = f \cdot k, \quad (3)$$

где k – коэффициент проявления дефектов k программы (1/дефект). Значение k может быть получено эмпирическим путем по однотипным программам; обычно это значение изменяется от $1,5 \times 10^{-7}$ до $10,6 \times 10^{-7}$ (по умолчанию принимают $k = 4,2 \times 10^{-7}$); f – средняя скорость выполнения компьютером одной машинной команды для рассматриваемой программы (команд/с); зависит от архитектуры компьютера, производительности процессора, иерархии памяти, а также особенностей программы (числа итераций в циклах или условных переходах и т. п.).

Величина f выражения (3) определяется отношением

$$f = \frac{v}{0,001L_{KLOC}E}, \quad (4)$$

где v – средняя скорость выполнения команд компьютером (команд/с); в англоязычном варианте – IPS (operations per second); E – коэффициент расширения кода программы (команд/строка кода), показывающий, во сколько машинных команд транслируется одна строка исходного кода программы, например для языка C++, $E = 6,0$).

Значение плотности ошибок F может быть найдено с помощью модели RL-92-52 [7], основанной на использовании метрик программного обеспечения:

$$F = A \cdot D, \quad (5)$$

где F – прогнозируемая плотность дефектов, приходящаяся на 1000 строк кода (табл. 1); A – метрика, учитывающая тип (назначение) компьютерной программы; D – метрика, учитывающая среду разработки программного обеспечения (табл. 2).

Таблица 1. Плотность дефектов проектирования на 1000 строк программного кода [7]

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. Тактические ракеты, 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

При выборе значения метрики D для систем медицинского назначения авторы рекомендуют использовать строку под номером 2.

Таблица 2. Значения метрики D , учитывающей среду разработки программного обеспечения [7]

Характеристика среды разработки	Значение метрики D
1. Группа программистов работает в организации, обеспечивает ее потребности в программном обеспечении и отвечает за программу	0,76
2. Группа программистов имеет опыт работы по разработке программ, но не связана с пользователем программы	1,00
3. Группа программистов обладает опытом работы с компьютером, но не знакома с приложением, для которого разрабатывается программа. Программа должна использоваться во взаимосвязанном комплексе технических средств, программного обеспечения и действий оператора	1,30

В табл. 3 приводятся поправочные коэффициенты вида K_i , дается их пояснение, указываются экстраполированные значения и использованные источники их получения.

Предложенный метод позволяет по модели (1) определить ожидаемую интенсивность проявления дефектов $\lambda_{п.тест}$ в программе, прошедшей тестирование. Размерность величины $\lambda_{п.тест}$ – 1/с или с⁻¹. Рассчитав время выполнения программой одного набора данных (среднее время «прогона» программы t_1), можно определить вероятность p того, что программа правильно (безошибочно) выполнит обработку одного произвольного набора данных:

$$p = \exp(-\lambda_{п.тест} \cdot t_1). \quad (6)$$

Уточнив интенсивность обращения к программе в процессе функционирования информационно-компьютерной системы медицинского назначения, т. е. среднее число «прогонов» программы за единицу времени η (1/ч), можно рассчитать вероятность того, что оставшиеся дефекты в программе не проявятся в течение заданного времени τ , например, рабочей смены (6 ч):

$$P_{кп}(\tau) = p^{n\tau}. \quad (7)$$

Таблица 3. Пояснение поправочных коэффициентов K_i

Обозначение	Учитываемый фактор	Пояснение	Значения коэффициента
$K_{\text{слож}}$	Категория сложности ПС	Три категории сложности	От 1,00 до 1,77 по [8]
$K_{\text{нов}}$	Степень новизны ПС	Три категории новизны	От 0,63 до 1,58 по [8]
$K_{\text{С.Р}}$	Средства разработки (язык программирования, операционная система, компьютерная сеть)	Семь категорий, три разновидности	От 0,16 до 1,3 по [8]
$K_{\text{мод}}$	Степень использования стандартных модулей	Пять категорий	От 0,55 до 1,00 по [8]
$K_{\text{кв.прог}}$	Квалификация и опыт программиста	Три уровня	См. *
$K_{\text{вх.дан}}$	Степень изменчивости входных исходных данных (использовать максимальное значение ν – коэффициент вариации операндов)	Четыре градации: непредсказуемая, $\nu \gg 100\%$; очень сильная, $\nu = 60 \dots 100\%$; сильная, $\nu = 33 \dots 60\%$; слабая, ν до 33 %	От 8 до 1 (экстраполированные значения), шаг геометрической прогрессии $q = 0,5$
$K_{\text{тест}} (S_1, S_2, S_3)$	Тестирование программы	Используются метрики, учитывающие технологию и условия тестирования	Использовать метод, аналогичный описанному в [7]. См. **
<p>* Предлагаются три категории уровня квалификации и опыта (согласно [9]): 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.</p> <p>** Предлагаются два уровня квалификации тестировщиков (в соответствии с [10]): 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).</p>			

Зная вероятность безотказной работы за время τ комплекса технических средств $P_{\text{кТС}}(\tau)$, вероятность безошибочных действий медицинского персонала в течение времени τ $P_{\text{МП}}(\tau)$ и значение $P_{\text{кп}}(\tau)$, рассчитанное по формуле (7), можно оценить ожидаемую надежность медицинской информационно-компьютерной системы в целом, пользуясь структурной схемой надежности (см. рис. 1).

Заключение

Предложенный метод дает приближенные результаты ожидаемой надежности разрабатываемой компьютерной программы. Однако даже такой ориентировочный расчет полезен, так как позволяет получить представление о надежности медицинской информационно-компьютерной системы на раннем этапе ее разработки и при необходимости определить организационно-технические мероприятия по обеспечению требуемой надежности системы.

Список литературы

1. Надежность технических систем: справочник / Под ред. И.А. Ушакова. М.: Радио и связь, 1985. 608 с.
2. Боровиков С.М., Цырельчук И.Н., Троян Ф.Д. Расчет показателей надежности радиоэлектронных средств. Минск: БГУИР, 2010. 68 с.
3. Оценка надежности системы «человек-машина». [Электронный ресурс]. URL: <https://poznayka.org/s62329t1.html> (дата общения: 18.10.2018).
4. Программное обеспечение – источник всех проблем. [Электронный ресурс]. URL: <http://www.williamspublishing.com/PDF/5-8459-0785-3/part1.pdf> (дата общения: 20.10.2018).
5. Боровиков С.М., Дик С.С. Прогнозирование ожидаемой надежности прикладных программных средств с использованием статистических моделей их безотказности // Сб. матер. IV Междунар. науч.-практ. конф. «BIG DATA and Advanced Analytics». Минск, 3–4 мая 2018 г. С. 348–354.
6. Программирование и основы алгоритмизации. Исследование методов оценки и повышения надежности программного обеспечения. [Электронный ресурс]. URL: <https://refdb.ru/look/2279745-pall.html> (дата обращения: 29.10.2018).
7. Software reliability, measurement and testing guidebook for software reliability measurement and testing: RL-TR-92-52. Vol II. Final technical report April 1992 / Science Applications International Corp.,

Research Triangle Institute (RTI). Rome Laboratory Air Force Systems Command Griffiss Air Force Base NY 13441-5700.

8. Укрупненные нормы затрат труда на разработку программного обеспечения. [Электронный ресурс]. URL: <http://www.levonevski.net/pravo/norm2013/num28/d28024.html> (дата обращения: 18.10.2018).
9. Ранги в профессии программист или профессиональный рост в IT карьере. [Электронный ресурс]. URL: <https://www.trn.ua/articles/6733> (дата обращения: 18.10.2018).
10. Тестирование программного обеспечения. Базовый курс: версия книги 1.0.3 от 23.09.2015. [Электронный ресурс]. URL: https://www.bsuir.by/m/12_108786_1_98216.pdf (дата обращения: 19.10.2018).

References

1. Nadezhnost' tehniceskikh sistem: spravochnik / Pod red. I.A. Ushakova. M.: Radio i svjaz', 1985. 608 s. (in Russ.)
2. Borovikov S.M., Cyrel'chuk I.N., Trojan F.D. Raschet pokazatelej nadezhnosti radiojelektronnyh sredstv. Minsk: BGUIR, 2010. 68 s. (in Russ.)
3. Ocenka nadezhnosti sistemy «chelovek-mashina». [Electronic resource]. URL: <https://poznayka.org/s62329t1.html> (date of access: 18.10.2018). (in Russ.)
4. Programmnoe obespechenie – istochnik vseh problem. [Electronic resource]. URL: <http://www.williamspublishing.com/PDF/5-8459-0785-3/part1.pdf> (date of access: 20.10.2018). (in Russ.)
5. Borovikov S.M., Dik S.S. Prognozirovanie ozhidaemoj nadezhnosti prikladnyh programnyh sredstv s ispol'zovaniem statisticheskikh modelej ih bezotkaznosti // Sb. mater. IV Mezhdunar. nauch.-prakt. konf. «BIG DATA and Advanced Analytics». Minsk, 3–4 maja 2018 g. S. 348–354. (in Russ.)
6. Programirovanie i osnovy algoritimizacii. Issledovanie metodov ocenki i povysheniya nadezhnosti programmnoho obespechenija. [Electronic resource]. URL: <https://refdb.ru/look/2279745-pall.html> (date of access: 29.10.2018). (in Russ.)
7. Software reliability, measurement and testing guidebook for software reliability measurement and testing: RL-TR-92-52. Vol II. Final technical report April 1992 / Science Applications International Corp., Research Triangle Institute (RTI). Rome Laboratory Air Force Systems Command Griffiss Air Force Base NY 13441-5700.
8. Ukpurnennye normy zatrat truda na razrabotku programmnoho obespechenija. [Electronic resource]. URL: <http://www.levonevski.net/pravo/norm2013/num28/d28024.html> (date of access: 18.10.2018). (in Russ.)
9. Rangi v professii programmist ili professional'nyj rost v IT kar'ere. [Electronic resource]. URL: <https://www.trn.ua/articles/6733> (date of access: 18.10.2018). (in Russ.)
10. Testirovanie programmnoho obespechenija. Bazovyy kurs: versija knigi 1.0.3 ot 23.09.2015. [Electronic resource]. URL: https://www.bsuir.by/m/12_108786_1_98216.pdf (date of access: 19.10.2018). (in Russ.)

Сведения об авторах

Боровиков С.М., к.т.н., доцент кафедры проектирования информационно-компьютерных систем Белорусского государственного университета информатики и радиоэлектроники.

Дик С.С., аспирант Белорусского государственного университета информатики и радиоэлектроники.

Information about the authors

Borovikov S.M., PhD, associate professor of department of information and computer systems design of Belarusian state university of informatics and radioelectronics.

Dick S.S., PG student of Belarusian state university of informatics and radioelectronics.

Адрес для корреспонденции

220013, Республика Беларусь,
г. Минск, ул. П. Бровки, д. 6,
Белорусский государственный университет
информатики и радиоэлектроники
тел. +375-29-610-24-34;
e-mail: bsm@bsuir.by
Боровиков Сергей Максимович

Address for correspondence

220013, Republic of Belarus,
Minsk, P. Brovka st., 6,
Belarusian state university
of informatics and radioelectronics
tel. +375-29-610-24-34;
e-mail: bsm@bsuir.by
Borovikov Sergei Maksimovich