

**МОДЕЛЬ ПРОГНОЗИРОВАНИЯ НАДЁЖНОСТИ ПЛАНИРУЕМЫХ  
К РАЗРАБОТКЕ ПРИКЛАДНЫХ КОМПЬЮТЕРНЫХ ПРОГРАММ*****Боровиков Сергей Максимович****канд. техн. наук, доц., Белорусский государственный университет информатики и радиоэлектроники,  
Республика Беларусь, г. Минск****Дик Сергей Сергеевич****аспирант, Белорусский государственный университет информатики и радиоэлектроники,  
Республика Беларусь, г. Минск****Лэ Ван Там****магистрант, Белорусский государственный университет информатики и радиоэлектроники,  
Республика Беларусь, г. Минск****Клинов Константин Игоревич****студент, Белорусский государственный университет информатики и радиоэлектроники,  
Республика Беларусь, г. Минск***RELIABILITY FORECASTING MODEL PLANNED  
TO DEVELOP APPLIED COMPUTER PROGRAMS*****Sergey Borovikov****Candidate of Sciences, Associate Professor, Belarusian State University of Informatics and Radioelectronics,  
Republic of Belarus, Minsk****Sergey Dick****graduate student, Belarusian State University of Informatics and Radioelectronics,  
Republic of Belarus, Minsk****Le Van Tam****master's degree, Belarusian State University of Informatics and Radioelectronics,  
Republic of Belarus, Minsk****Konstantin Klinov****student, Belarusian State University of Informatics and Radioelectronics,  
Republic of Belarus, Minsk***АННОТАЦИЯ**

Оценка надёжности планируемых к разработке компьютерных программ является актуальной задачей. Для оценки надёжности компьютерной программы надо располагать моделью прогнозирования надёжности программы. В статье излагаются исходные предпосылки и положения, используемые для получения модели, и приводится сама модель прогнозирования надёжности прикладных компьютерных программ. С помощью полученной модели можно решать практические задачи по оценке ожидаемой надёжности компьютерных программ, планируемых к разработке, до написания кода программ на языках программирования.

**ABSTRACT**

Assessment of the reliability of computer programs planned for development is an urgent task. To assess the reliability of a computer program, you must have a model for predicting the reliability of the program. The article sets out the initial premises and provisions used to obtain the model and provides the model for predicting the reliability of applied computer programs. Using the obtained model, it is possible to solve practical problems in assessing the expected reliability of computer programs planned for development before writing the program code in programming languages.

**Ключевые слова:** прикладная компьютерная программа, модель надёжности, прогнозирование.

**Keywords:** applied computer program, reliability model, forecasting.

Разработчики компьютерных информационных систем во многих случаях хотели бы знать ожидае-

мый уровень надёжности программных средств на раннем этапе их разработки, до написания кода

компьютерных программ на языке программирования. Для получения ответа на интересующий вопрос надо иметь модель надёжности компьютерной программы.

В соответствии с ГОСТ 27.205-1997 [1] о проектной надёжности разрабатываемой компьютерной программы (программного средства) будем судить по интенсивности проявления скрытых ошибок, оставшихся в компьютерной программе после её разработки и тестирования. Эту интенсивность будем называть интенсивностью отказов компьютерной программы (обозначим как  $\lambda$ ).

В качестве исходных данных для оценки надёжности компьютерной программы будем рассматривать:

- предполагаемый объём программы в строках кода;
- назначение прикладной компьютерной программы (сфера, область деятельности людей);

- технические характеристики используемого компьютера: быстродействие процессора (производительность компьютера).

Предпосылки и положения, принятые во внимание при разработке модели надёжности компьютерных программ:

1. Значение интенсивности отказов  $\lambda$  считают прямо пропорциональным числу оставшихся в компьютерной программе ошибок  $N_{\text{ош}}$  [2, с. 26], [3, ч. 2, с. 79]:

$$\lambda \cdot C_{\text{пр}} \cdot N_{\text{ош}}, \quad (1)$$

где  $C_{\text{пр}}$  – коэффициент пропорциональности, определяемый экспериментально.

2. Для прикладных программ различного функционального назначения известны коэффициенты трансформации  $K_{\text{тр}}$  плотности ошибок  $F$  в интенсивность отказов  $\lambda$  [4, с. 100-7] (таблица 1, 2-й столбец). Плотность ошибок  $F$  это отношение вида

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

Таблица 1.

Значения коэффициентов для компьютерных программ разного функционального назначения

Область применения компьютерной программы	Значение $K_{\text{тр}}$ , строка кода/ошибка в час	Коэффициент $C$ , 1/ошибка	Коэффициент $K_{\Sigma}$	Значение $A$ , ошибок/строка кода
Авиация	6,28	$1,74 \cdot 10^{-9}$	5,23	0,0128
Мониторинг и обеспечение безопасности	1,2	$3,33 \cdot 10^{-10}$	1	0,0092
Телекоммуникации, мобильные устройства	13,8	$3,83 \cdot 10^{-9}$	11,5	0,0078
Управление производственными процессами	3,8	$1,06 \cdot 10^{-9}$	3,18	0,0018
Автоматизированные системы управления	23	$6,39 \cdot 10^{-9}$	19,2	0,0085
Разработка программ, моделирование, обучение	16,9	$4,69 \cdot 10^{-9}$	14,1	0,0123
Среднее	10,6	$2,94 \cdot 10^{-9}$	8,83	0,0087

Данные таблицы 1 (2-й столбец) 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).

3. На интенсивность отказов  $\lambda$  влияют три характеристики эксплуатационной среды выполнения компьютерной программы [4, с. 100-5]:

- производительность компьютера (быстродействие процессора);
- изменчивость входных данных и состояний управления компьютером;
- рабочая нагрузка, которую эксплуатационная среда будет оказывать на компьютерную программу при её использовании по назначению.

4. Интенсивность отказов компьютерной программы до начала этапа тестирования может быть

определена на основе модели Муса [2, с. 26], [5, с. 2] по формуле

$$\lambda_0 = C \cdot V \cdot N_{\text{нач}} = 3600 \cdot C \cdot \frac{R}{E_R \cdot L} \cdot N_{\text{нач}}, \quad (2)$$

где:  $C$  – коэффициент пропорциональности, некоторые авторы называют его коэффициентом проявления ошибок;

$V$  – скорость выполнения команд компьютерной программы, представляет собой возможное число прогонов компьютерной программы (команд программы) в течение одного часа;

$N_{\text{нач}}$  – начальное число ошибок в компьютерной программе (до начала тестирования);

$R$  – быстродействие процессора (производительность компьютера), размерность: операций в секунду;

$L$  – объём компьютерной программы в строках кода;

$E_R$  – коэффициент расширения кода (увеличения числа команд программы относительно числа строк кода);

множитель 3600 – коэффициент перевода быстродействия процессора в размерность «операций в час».

Согласно [6, с. 132] будем считать, что одна строка кода компьютерной программы транслируется в 10 машинных команд ( $E_R = 10$ ).

5. Используя значения коэффициента  $K_{тр}$ , определены средние значения коэффициента пропорциональности  $C$  для компьютерных программ разных областей применения (см. таблицу 1, 3-й столбец).

Представим коэффициент  $C$  в виде произведения

$$C = C_B \cdot K_{вх} \cdot K_n = C_B \cdot K_{\Sigma}, \quad (3)$$

а величины, входящие в выражение (3), 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;  $K_{\Sigma}$  – суммарный коэффициент увеличения интенсивности отказов, обусловленный совместным действием изменчивости входных данных и рабочей нагрузки на компьютерную программу.

В качестве базового значения коэффициента проявления ошибок  $C_B$  принято значение коэффициента  $C$  для компьютерных программ, используемых для мониторинга и обеспечения безопасности. Пользуясь таблицей 1 (3-я строка), можно записать:  $C_B = 3,33 \cdot 10^{-10}$  1/ошибка. Для программ этого назначения суммарный коэффициент увеличения интенсивности отказов принят  $K_{\Sigma} = 1,0$ . Рассчитанные значения коэффициентов  $K_{\Sigma}$  для компьютерных программ другого функционального назначения приведены в таблице 1 (4-й столбец).

6. Предполагаем, что процедура тестирования программного средства выполняется общепринятыми методами, и при этом её суммарная продолжительность составляет не менее 40 процентов общего времени, отводимого на разработку компьютерной программы. Согласно работе [4, с. 300-3], экспериментальные данные показали, что в таких случаях интенсивность отказов компьютерных программ за счёт выполнения тестирования уменьшается примерно в семь раз относительно начальной интенсивности отказов  $\lambda_0$ .

**Результаты.** С учётом изложенных предпосылок и положений модель для определения ожидаемой интенсивности отказов компьютерной программы, прошедшей тестирование ( $\lambda_{эсп}$ ), может быть получена в виде

$$\begin{aligned} \lambda_{эсп}^{(i)} &= 0,14 \cdot C_B \cdot \frac{3600R}{10L} \cdot N_{нач}^{(i)} \cdot K_{\Sigma}^{(i)} = \\ &= 50,4 \cdot C_B \cdot R \cdot F_{нач}^{(i)} \cdot K_{\Sigma}^{(i)}, \end{aligned} \quad (4)$$

где: верхний индекс ( $i$ ) указывает на то, что соответствующие характеристики относятся к компьютерным программам  $i$ -й области применения (см. таблицу 1);

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

$F_{нач}$  – начальная плотность ошибок в компьютерной программе.

Значение  $F_{нач}$  может быть найдено с использованием работ [3, ч. 2, с. 80], [4, с. 100-3] по модели RL-92-52 («Римская модель»), основанной на метриках программного обеспечения:

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

где:  $F$  – прогнозируемая плотность ошибок, приходящихся на строку кода разрабатываемой программы;

$A$  – метрика, учитывающая тип или назначение компьютерной программы;

$D$  – метрика, учитывающая среду разработки программного обеспечения (характеристику проектной организации и опыт программистов);

$S$  – метрика, учитывающая характеристики компьютерной программы.

Метрика  $A$  представляет собой среднюю или базовую плотность ошибок для компьютерных программ, используемых в данной отрасли (сфере деятельности людей), и рассматривается в качестве отправной точки для получения прогноза значения  $F$  [4, с. В-4] (см. таблицу 1, 5-й столбец). 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 компьютерной программы имеют тенденцию уменьшать плотность ошибок или значение больше единицы, если они имеют тенденцию увеличивать плотность. Эти метрики эквивалентны «пи-факторам» в методике расчёта надёжности электронных устройств по MIL HDBK 217F, а также поправочным коэффициентам  $K_i$  при прогнозировании эксплуатационной надёжности элементов с использованием систем АСПН (Россия) или АРИОН (Республика Беларусь) [7-9].

Значение метрики  $D$  может быть выбрано с учётом того, что организация, заинтересованная в создании компьютерной информационной системы, заказывает разработку программного обеспечения специализированной ИТ-компании, программисты которой имеют опыт по разработке программ, но прямо не связаны с пользователями компьютерной программы. 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  $0,5 \leq D \leq 2$  [4, с. 102-3]. Выбирая значение метрики  $D$ , необходимо также принять во внимание квалификацию и опыт программистов (коэффициент  $K_{кв.прог}$ , см. таблицу 2).

Для определения метрики  $S$  выражения (5) предлагается уточнённая модель вида

$$S = K_{\text{нов}} \cdot K_{\text{слож}} \cdot K_{\text{С.Р}} \cdot K_{\text{мод.}} \quad (6)$$

где:  $K_i$  – поправочный коэффициент (иначе – подметрика), учитывающий влияние конкретного фактора (обозначен как  $i$ -й фактор) на значение метрики  $S$  и, следовательно, на интенсивность отка-

зов компьютерной программы; 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 [3, ч. 2, с. 81].

Таблица 2.

Поправочные коэффициенты  $K_i$ 

Обозначение	Учитываемый фактор компьютерной программы	Пояснение	Значение коэффициента
$K_{\text{слож}}$	Категория сложности	Три категории, шесть характеристик повышения сложности	От 1,00 до 1,47 по [10, приложения 2-4]
$K_{\text{нов}}$	Степень новизны	Три категории, включающие восемь разновидностей	От 0,63 до 1,58 по [10, приложение 5]
$K_{\text{С.Р}}$	Средства разработки (язык программирования, операционная система, компьютерная сеть)	Семь категорий, три разновидности	От 0,16 до 1,3 по [10, приложение 7]
$K_{\text{мод}}$	Степень использования стандартных модулей	Пять категорий	От 0,55 до 1,00 по [10, приложение 6]
$K_{\text{кв.прог}}$	Квалификация и опыт программиста	Три уровня	Смотрите примечание
<i>Примечание:</i> Предлагается три уровня квалификации и опыта согласно [11]: 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.			

**Выводы:** Полученная в работе модель вида (4) даёт возможность приближённо оценить ожидаемую надёжность разрабатываемой компьютерной программы. Даже такой ориентировочный расчёт полезен, так как позволяет на раннем этапе разработки

программного обеспечения получить представление о его надёжности и сориентироваться в целесообразности проектирования и/или эффективности применения компьютерной информационной системы, использующей это программное обеспечение.

#### Список литературы:

- ГОСТ 27.205-1997. 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.
- Модели оценки надёжности программных средств [Электронный ресурс]. – Режим доступа : <https://helpiks.org/4-73503.html> (дата обращения: 22.03.2020).
- 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. Богущ [и др.]. – Минск, 2019. – С. 77-83.
- 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.
- Динамические модели надёжности программного обеспечения [Электронный ресурс]. – Режим доступа : <https://student.zoomru.ru/informat/dinamicheskie-modeli-nadezhnosti-programmnogo-obespecheniya/189982.1594166.s1.html> (дата обращения: 26.03.2020).
- Чуканов, В. О. Надёжность программного обеспечения и аппаратных средств систем передачи данных атомных электростанций : учеб. пособие. М.: МИФИ, 2008. 68 с
- Reliability prediction of electronic equipment : Military Handbook MIL-HDBK-217F. – Washington : Department of defense DC 20301, 1995. 205 p.

8. 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.
9. 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.
10. Укрупнённые нормы затрат труда на разработку программного обеспечения [Электронный ресурс]. – Режим доступа : <https://zakonrb.com/npa/ob-utverzhdennii-ukrupnennyh-norm-zatrat-truda> (дата обращения: 25.03.2020).
11. Ранги программистов [Электронный ресурс]. – Режим доступа : <http://www.ardecs.com/blog/23-02-2017/?lang=ru> (дата обращения: 25.03.2020).