



<http://dx.doi.org/10.35596/1729-7648-2022-20-7-72-80>

Оригинальная статья
Original paper

УДК [004.415.53+004.42]:621.382.2/3-027.45

МОДЕЛЬ ПРОГНОЗИРОВАНИЯ ВРЕМЕНИ ТЕСТИРОВАНИЯ КОМПЬЮТЕРНОЙ ПРОГРАММЫ АВТОМАТИЗИРОВАННОЙ ОЦЕНКИ НАДЕЖНОСТИ ПОЛУПРОВОДНИКОВЫХ ПРИБОРОВ

В. О. КАЗЮЧИЦ, С. М. БОРОВИКОВ, Е. Н. ШНЕЙДЕРОВ

*Белорусский государственный университет информатики и радиоэлектроники
(г. Минск, Республика Беларусь)*

Поступила в редакцию 20.07.2022

© Белорусский государственный университет информатики и радиоэлектроники, 2022

Аннотация. Планируемая к разработке компьютерная программа KLASS предназначена для работы в составе программного комплекса АРИОН-плюс и позволяет выполнять автоматизированную оценку надежности изделий электроники, включая полупроводниковые приборы. На этапе планирования работ по созданию программы KLASS, как модуля комплекса АРИОН-плюс, возник вопрос о рабочем времени, отводимом на процедуру тестирования компьютерной программы. Описанные в научной литературе подходы, используемые для оценки эксплуатационной надежности компьютерных программ с учетом их тестирования, исходят из того, что написан и отлажен программный код и имеются определенные данные о результатах тестирования компьютерной программы. Разработчики программного обеспечения еще до начала выполнения работ по написанию программного кода хотели бы знать прогнозное время тестирования, обеспечивающее заданную эксплуатационную надежность компьютерной программы. На основе анализа экспериментальных данных о надежности компьютерных программ разных областей применения предложена модель определения времени тестирования, необходимого для обеспечения эксплуатационной надежности программ. Модель использована для планируемой к разработке компьютерной программы KLASS и учитывает язык программирования, объем программного кода, быстродействие процессора компьютера и область применения программы. На основе полученной модели построена номограмма с двумя бинарными полями, позволяющая быстро определить прогнозное время тестирования компьютерных программ.

Ключевые слова: компьютерные программы, эффективность тестирования, время тестирования, эксплуатационная надежность программ, полупроводниковые приборы.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Благодарности. Работа выполнена в рамках договоров (проектов) на выполнение научно-исследовательских работ № Т20МВ-026 на тему «Прогнозирование эксплуатационной надежности мощных полупроводниковых приборов с использованием методов и алгоритмов машинного обучения» и № Ф20МВ-021 «Статистические модели надежности прикладных программных средств и их использование для оценки ожидаемой безотказности компьютерных программ на ранних этапах их разработки» в соответствии с решением Научного совета Белорусского республиканского фонда фундаментальных исследований (протокол № 1 от 22.04.2020).

Для цитирования. Казючиц В. О., Боровиков С. М., Шнейдеров Е. Н. Модель прогнозирования времени тестирования компьютерной программы автоматизированной оценки надежности полупроводниковых приборов. Доклады БГУИР. 2022. 20 (7). С. 72–80.

MODEL FOR PREDICTION OF TESTING TIME OF A COMPUTER PROGRAM FOR AUTOMATED RELIABILITY EVALUATION OF SEMICONDUCTOR DEVICES

VLADISLAV O. KAZIUCHYTS, SERGEI M. BOROVIKOV, EVGENI N. SHNEIDEROV

Belarusian State University of Informatics and Radioelectronics (Minsk, Republic of Belarus)

Submitted 20.07.2022

© Belarusian State University of Informatics and Radioelectronics, 2022

Abstract. The KLASS computer program planned for development is designed to work as a part of the ARION-plus software package and allows you to perform an automated assessment of the reliability of electronic products, including semiconductor devices. At the stage of work planning on the creation of the KLASS program, as a module of the ARION-plus complex, the question arose about the working time allotted for the procedure for testing a computer program. The approaches described in the scientific literature used to assess the operational reliability of computer programs, taking into account their testing, proceed from the fact that the program code has been written and debugged and there are certain data on the results of testing the computer program. Software developers would like to know the predicted testing time, which ensures a given operational reliability of a computer program, even before starting work on writing program code. Based on the analysis of the experimental data on the reliability of computer programs in various fields of application, a model is proposed for determining the testing time required to ensure the operational reliability of programs. The model was used for the computer program KLASS planned for development and takes into account the programming language, the amount of program code, the speed of the computer processor, and the scope of the program. Based on the obtained model, a nomogram with two binary fields was constructed, which allows one to quickly determine the predicted time for testing computer programs.

Keywords: computer programs, testing efficiency, testing time, operational reliability, semiconductor devices.

Conflict of interests. The authors declare no conflict of interests.

Gratitude. The article was prepared as part of the project No. T20MB-026 on the topic “Predicting the operational reliability of powerful semiconductor devices using machine learning methods and algorithms” and No F20MV-021 “Statistical models of the reliability of applied software and their use to assess the expected reliability of computer programs at the early stages of their development” in accordance with the decision approved by the Scientific Council of the Belarusian Republican Foundation for Fundamental Research (Protocol No 1 dated 22.04.2020).

For citation. Kaziuchyts V. O., Borovikov S. M., Shneiderov E. N. Model for Prediction of Testing Time of a Computer Program for Automated Reliability Evaluation of Semiconductor Devices. Doklady BGUIR. 2022. 20 (7), 72–80.

Введение

Компьютерные программы, разрабатываемые для современных информационных систем, могут содержать тысячи и даже миллионы строк кода. После написания программного кода и устранения явных ошибок (синтаксических, например), вызываемых нарушением грамматики языка программирования, программа характеризуется начальным уровнем надежности, который обычно не отвечает требованиям пользователя. Поэтому необходима процедура тестирования программы. В программах указанного объема даже после выполнения их тестирования всегда содержатся скрытые ошибки, наличие которых определяет эксплуатационную надежность программ. Обеспечить идеальную эксплуатационную надежность компьютерных программ при большом объеме кода не представляется возможным из-за того, что при тестировании не удастся в проектные сроки проверить функционирование алгоритма компьютерной программы с учетом всех возможных трасс и ветвей проведения вычислительного процесса, особенности вводимых исходных данных. При тестировании стремятся устранить критические ошибки с точки зрения правильности результата работы программы, уменьшить долю скрытых ошибок и, тем самым, обеспечить требования заказчика к эксплуатационной надежности разрабатываемой компьютерной программы.

Алгоритм компьютерной программы

Планируемая к разработке компьютерная программа KLASS предназначена для функционирования в качестве модуля в составе программного комплекса АРИОН-плюс, созданного в рамках выполнения НИР на кафедре проектирования информационно-компьютерных систем в Белорусском государственном университете информатики и радиоэлектроники [1]. В качестве информационной основы алгоритма компьютерной программы KLASS использованы описанные в [2–6] методы и модели прогнозирования надежности полупроводниковых приборов (ППП) по их информативным параметрам с определением класса работоспособности экземпляров на заданную наработку: K_1 – класс работоспособных экземпляров, K_0 – класс неработоспособных экземпляров.

Основные функциональные задачи, возлагаемые на компьютерную программу KLASS, понятны из описания этапов автоматизированного решения задачи прогнозирования надежности ППП по их информативным параметрам.

1. Ввод с клавиатуры или электронных носителей результатов обучающего эксперимента для рассматриваемого типа ППП в виде матрицы, включающей результаты измерений электрических параметров, предполагаемых на информативность, с указанием для каждого экземпляра номера класса с учетом его работоспособности по результатам испытаний для заданной наработки t_n : 1 – класс работоспособных экземпляров, 0 – класс неработоспособных экземпляров. Размер матрицы $n \times (q+1)$, где n – объем обучающей выборки (до 1000 экземпляров); q – число электрических параметров, проверяемых на информативность (до 100 электрических параметров).

2. Определение для электрических параметров (обозначим их через x_i) степени информативности по выбираемому критерию ($i = 1, 2, \dots, q$):

- коэффициенту линейной парной корреляции электрического параметра x_i с номером класса экземпляра (1 или 0) по результатам испытаний;
- информационной мере Кульбака, определяемой по [3, 4];
- количеству информации о классе экземпляра K_S ($S = 1; 0$), получаемой от контроля (измерения) электрического параметра x_i [3, 4].

В последнем случае информативный электрический параметр x_i вначале преобразуется по определенным правилам [3–6] в кодовый сигнал τ_i с последующим определением вероятностей принадлежности экземпляра к классам K_1 и K_0 при условии, что по результатам преобразования x_i в код получено конкретное значение кодового сигнала τ_i .

3. Формирование обучающей выборки в виде матрицы размером $n \times (k + 1)$, где k – число электрических параметров, выбранных в качестве информативных. Элементами этой матрицы являются кодовые сигналы, полученные по результатам преобразования x_i в кодовые сигналы τ_i ($i = 1, 2, \dots, k$) по правилам, приводимым в [3–6]. В случае использования третьего критерия этапа 2 кодовые сигналы уже получены. При использовании первых двух критериев, указанных в этапе 2, получение кодовых сигналов τ_i и указанных условных вероятностей выполняется на данном этапе.

4. Подсчет для каждого экземпляра обучающей выборки частной информации о принадлежности экземпляра обучающей выборки к классу K_1 (обозначим как $I(K_1 | \tau_1^{(j)}, \tau_2^{(j)}, \dots, \tau_k^{(j)})$) и к классу K_0 (обозначим как $I(K_0 | \tau_1^{(j)}, \tau_2^{(j)}, \dots, \tau_k^{(j)})$), содержащейся в сообщении о том, что по результатам преобразования информативных параметров j -го экземпляра $x_1^{(j)}, x_2^{(j)}, \dots, x_k^{(j)}$ в кодовые сигналы $\tau_1^{(j)}, \tau_2^{(j)}, \dots, \tau_k^{(j)}$ для этого экземпляра получен конкретный набор кодовых сигналов.

Этот этап следует рассматривать в качестве машинного обучения, поскольку для каждого экземпляра обучающей выборки определяется количество информации о близости экземпляра к классам K_1 и K_0 . В данном случае особенность машинного обучения, включающего преобразование информативных параметров с учетом их закономерностей в кодовые сигналы, и определение для каждого экземпляра обучающей выборки частной информации о принадлежности его к классам K_1 и K_0 , состоит в следующем:

– для экземпляров в действительности класса K_1 частная информация $I(K_1 | \tau_1^{(j)}, \tau_2^{(j)}, \dots, \tau_k^{(j)})$ о принадлежности к классу K_1 оказывается в основном положительной, а частная информация $I(K_0 | \tau_1^{(j)}, \tau_2^{(j)}, \dots, \tau_k^{(j)})$ о принадлежности к классу K_0 – в основном отрицательной;

– для экземпляров в действительности класса K_0 положительной в основном оказывается информация $I(K_0 | \tau_1^{(j)}, \tau_2^{(j)}, \dots, \tau_k^{(j)})$, а отрицательной – частная информация $I(K_1 | \tau_1^{(j)}, \tau_2^{(j)}, \dots, \tau_k^{(j)})$.

5. Определение для каждого j -го экземпляра обучающей выборки значения прогнозирующей функции $F^{(j)}$:

$$F^{(j)} = I(K_1 | \tau_1^{(j)}, \tau_2^{(j)}, \dots, \tau_k^{(j)}) - I(K_0 | \tau_1^{(j)}, \tau_2^{(j)}, \dots, \tau_k^{(j)}). \quad (1)$$

Получение на основе функции (1) прогнозирующего правила, показывающего, как по набору кодовых сигналов j -го экземпляра $(\tau_1^{(j)}, \tau_2^{(j)}, \dots, \tau_k^{(j)})$ принимать решение о классе этого экземпляра для заданной наработки t_n . Теоретически порогом разделения классов является значение $F = 0$. При $F^{(j)} > 0$ должно приниматься решение о принадлежности экземпляра к классу K_1 , а при $F^{(j)} < 0$ – к классу K_0 .

6. Определение класса экземпляра обучающей выборки по прогнозу, используя прогнозирующую функцию $F^{(j)}$ при значении порога $F = 0$.

7. Подсчет ожидаемых характеристик качества прогнозирования с использованием для обучающей выборки данных о действительном классе и классе экземпляра по прогнозу. Оценка интересующих характеристик выполняется методами, описанными в [3, 4].

8. Корректировка прогнозирующего правила, полученного при выполнении этапа 5, и улучшение характеристик прогнозирования путем смещения порога F влево или вправо от нуля (оптимизация порога F).

9. Применение полученного прогнозирующего правила для прогнозирования класса однотипных экземпляров, не являющихся представителями обучающей выборки. Это позволит из новых выборок (партий) однотипных ППП отбирать экземпляры, отвечающие требованию надежности для заданной наработки t_n (экземпляры класса K_1), или же выполнять отбраковку экземпляров, не отвечающих требованию надежности для наработки t_n (экземпляры класса K_0).

Оценка надежности компьютерной программы

В качестве показателя надежности компьютерной программы будем рассматривать (согласно ГОСТ 27.205–1997) интенсивность проявления скрытых ошибок, оставшихся в программе при ее разработке. Оценка ожидаемой надежности прикладной компьютерной программы KLASS, реализующей описанные этапы процедуры прогнозирования надежности ППП, может быть дана с помощью подхода, изложенного в [7].

Для оценки прогнозной эксплуатационной интенсивности проявления скрытых ошибок прикладной компьютерной программы ($\lambda_{\text{экс}}$) с учетом быстродействия R процессора компьютера, изменчивости исходных данных и рабочей нагрузки, которую будет воспринимать компьютерная программа со стороны эксплуатационной среды, использована модель [7]:

$$\lambda_{\text{экс}} = \frac{60}{Q} K_{\Sigma} \frac{R}{B} F_0 L \cdot 10^{-6}. \quad (2)$$

В этой модели:

– коэффициент Q назван эффективностью тестирования, показывающей, во сколько раз уменьшается интенсивность отказов компьютерной программы после выполнения ее тестирования ($\lambda_{\text{экс}}$) относительно начальной интенсивности отказов λ_0 ;

– коэффициент K_{Σ} показывает увеличение интенсивности отказов из-за суммарного действия изменчивости входных данных и рабочей нагрузки на компьютерную программу со стороны факторов эксплуатационной среды (ввод данных, использование принтера, ожидание операций в очереди и т. д.);

– величина B представляет количество выполняемых процессором команд (операторов) компьютерной программы;

– произведение $F_0 L = N_0$ является прогнозным значением числа скрытых ошибок в компьютерной программе до выполнения ее тестирования; F_0 представляет начальную плотность ошибок в компьютерной программе, а L – прогнозное значение объема компьютерной программы в строках исполняемого кода.

Прогнозное число строк кода L компьютерной программы KLASS составляет примерно 20 тыс. строк (в англоязычном варианте $L = 20$ KLOC). Программу планируется использовать на компьютере, имеющем процессор с пиковым быстродействием $R = 500 \cdot 10^6$ операций/с.

Прогнозное количество команд B компьютерной программы найдено как

$$B = LE_L E_{ц}, \quad (3)$$

где E_L – коэффициент расширения кода программы, который определяется языком программирования; $E_{ц}$ – коэффициент увеличения числа выполняемых процессором команд за счет наличия в программе циклов, ветвлений и других особенностей.

Согласно [7], принято $E_L = 10$. Путем экспертной оценки для разрабатываемой компьютерной программы KLASS с учетом наихудшего пути ее выполнения принято $E_{ц} = 20$. В качестве K_{Σ} выберем среднее значение, характерное для прикладных компьютерных программ: $K_{\Sigma} = 8,83$ [7]. Значение базовой (усредненной) плотности ошибок A , используемой для расчета F_0 , примем равной 0,0087 ошибок / строка кода (строка «Среднее» [8]). Само значение F_0 найдено по модели

$$F_0 = A \prod_i K_i, \quad (4)$$

где K_i – коэффициенты (их называют метриками), учитывающие влияние на плотность ошибок особенностей проектной организации, квалификации и опыта программистов, характеристик компьютерной программы (новизна, сложность, средства разработки, степень использования стандартных модулей).

С учетом [7] и постановления Министерства труда и социальной защиты Республики Беларусь от 27.06.2007 № 91 «Об утверждении укрупненных норм затрат труда на разработку программного обеспечения» для компьютерной программы KLASS выбраны коэффициенты K_i и получены значения $F_0 \approx 0,056$ ошибка/строка кода, $\lambda_0 = 74,2 \text{ ч}^{-1}$.

Среднее время одного прогона компьютерной программы в часах (ч) определим как

$$t_{\text{пр}} = \frac{LE_L E_{ц}}{3600 \cdot 0,7R}, \quad (5)$$

где множитель 3600 – коэффициент перевода быстродействия процессора в размерность «операций/ч»; множитель 0,7 – средний коэффициент, учитывающий реальное быстродействие процессора относительно пикового значения R .

В формулу (5) значение R следует подставлять в размерности «операций/с». В данном случае $L = 20\,000$ строк кода, $E_L = 10$, $E_{ц} = 20$, $R = 500 \cdot 10^6$ операций/с. По формуле (5) получим $t_{\text{пр}} = (20\,000 \cdot 10 \cdot 20) / (3600 \cdot 0,7 \cdot 500 \cdot 10^6) \approx 3,17 \cdot 10^{-6}$ ч.

С учетом нагрузки эксплуатационной среды на программу (ввод и корректировка данных, вывод информации на печать, запись информации на электронные носители и т. п.) примем прогнозное число прогонов компьютерной программы KLASS при решении задач прогнозирования надежности ППП в течение трех рабочих месяцев, равным 1500 раз, что соответствует процессорному времени $t_{\text{пр,мес}} = 3,17 \cdot 10^{-6} \cdot 1500 = 0,00477$ ч. Вероятность отсутствия факта проявления ошибки программы (до выполнения тестирования) в течение трех рабочих месяцев составит

$$P(t_{\text{пр,мес}}) = \exp(-\lambda_0 \cdot t_{\text{пр,мес}}) = \exp(-74,2 \cdot 0,00477) \approx 0,702,$$

что в данном случае не является удовлетворительным результатом.

Актуальность разработки модели прогнозирования времени тестирования

Для практического применения компьютерной программы KLASS значение вероятности $P(t_{\text{пр,мес}})$ должно быть не менее 0,9, что обеспечивается при $\lambda_{\text{экс}} \leq 22,1 \text{ ч}^{-1}$. Значение $\lambda_{\text{экс}}$ может быть достигнуто при коэффициенте эффективности тестирования $Q = \lambda_0 / \lambda_{\text{экс}} = 74,2 / 22,1 \approx 3,36$.

Возникает вопрос, как спрогнозировать процессорное время тестирования программы KLASS, необходимое для получения требуемого коэффициента эффективности тестирования $Q = 3,36$ и, следовательно, какое календарное время в рабочих человеко-днях потребуется для выполнения тестирования компьютерной программы, чтобы достичь значения $\lambda_{\text{экс}} \leq 22,1 \text{ ч}^{-1}$, после чего компьютерная программа предположительно будет отвечать требованию эксплуатационной надежности и ее можно будет использовать для решения практических задач?

Методика исследований. Модель прогнозирования времени тестирования

После написания кода компьютерной программы и устранения нарушений правил языка программирования она характеризуется начальной интенсивностью проявления ошибок λ_0 . При завершении тестирования интенсивность проявления ошибок уменьшается и должна соответствовать эксплуатационному значению $\lambda_{\text{экс}}$. Для достижения $\lambda_{\text{экс}}$ необходимо при тестировании обеспечить требуемый коэффициент эффективности тестирования Q :

$$Q = \frac{\lambda_{\text{экс}}}{\lambda_0} = \frac{N_0}{N_{\text{п.тест}}}, \quad (6)$$

где $N_{\text{п.тест}}$ – число скрытых ошибок, оставшихся в компьютерной программе после выполнения ее тестирования.

Величина $N_{\text{п.тест}}$ в (6) определяется по формуле

$$N_{\text{п.тест}} = N_0 - n_{\text{обн}}, \quad (7)$$

где $n_{\text{обн}}$ – общее число ошибок, выявленных на этапе тестирования компьютерной программы, зависит от времени выполнения тестирования.

Методика определения времени тестирования, которое для компьютерной программы позволит достичь значения Q , основана на том, что величину $n_{\text{обн}}$, входящую в (7), необходимо выразить через время тестирования t и затем полученное уравнение решить относительно t .

В [9] отмечается, что значение $n_{\text{обн}}$ может быть определено с помощью экспоненциальной функции суммарного времени t прогона компьютерной программы при ее тестировании:

$$n_{\text{обн}} = N_0 \left[1 - \exp\left(-\frac{\lambda_0}{N} t\right) \right], \quad (8)$$

где N – общее число ошибок в компьютерной программе, которые теоретически могут проявиться при ее тестировании, $N \approx N_0$.

В формуле (8) величина t является процессорным временем выполнения компьютерной программы. Зная значение t , можно найти календарное время.

Принимая во внимание (6)–(8) и учитывая примерное равенство $N \approx N_0$, получим

$$Q = \frac{N_0}{N_{\text{п.тест}}} = \frac{N_0}{N_0 \exp\left(-\frac{\lambda_0}{N_0} t\right)} = \exp\left(\frac{\lambda_0}{N_0} t\right). \quad (9)$$

Записывая начальную интенсивность проявления ошибок λ_0 по модели Муса [9] с учетом быстродействия R процессора компьютера и приводимого в [7] описания коэффициента C этой модели, выражение (9) примет вид

$$Q = \exp\left(\frac{\lambda_0}{N_0} t\right) = \left| \begin{array}{l} \lambda_0 = C \frac{R}{L E_L E_{\text{ц}}} N_0, \\ C = C_{\text{б}} K_{\Sigma} \end{array} \right| = \exp\left(\frac{C_{\text{б}} K_{\Sigma} \frac{R}{L E_L E_{\text{ц}}} N_0}{N_0} t\right) = \exp\left(\frac{C_{\text{б}} K_{\Sigma} R}{L E_L E_{\text{ц}}} t\right), \quad (10)$$

где $C_{\text{б}}$ – базовое значение коэффициента проявления ошибок программы, не зависящее от области применения компьютерной программы.

Из выражения (10) получим модель определения процессорного времени t , необходимого для обеспечения требуемого коэффициента эффективности тестирования Q :

$$t = \frac{L E_L E_{\text{ц}} \ln(Q)}{C_{\text{б}} K_{\Sigma} R}. \quad (11)$$

Представим модель (11) в другом, более удобном виде, подставив в нее приводимое в [7] значение $C_{\text{б}} = 0,238 \cdot 10^{-7}$ 1/ошибка и следующие коэффициенты:

– $3600 \cdot 10^6$ – множитель для R , обеспечивающий для процессорного времени t размерность в часах (ч) и использование для R размерности «млн операций/с», то есть, если $R = 500 \cdot 10^6$ операций/с, то в качестве R следует подставлять число 500;

– 0,7 – множитель, учитывающий действительное быстродействие процессора относительно его пикового значения R , приводимого в технической документации.

Тогда модель (11) примет вид

$$t = \frac{L E_L E_{\text{ц}} \ln(Q)}{60 K_{\Sigma} R}. \quad (12)$$

Модель (12) позволяет определить процессорное время t , необходимое для обеспечения требуемого коэффициента эффективности тестирования Q и, следовательно, заданной эксплуатационной интенсивности $\lambda_{\text{экс}}$ проявления скрытых ошибок компьютерной программы.

Результаты исследований и их обсуждение

Для быстрого определения и анализа влияния параметров модели (12) на значение процессорного времени тестирования t была построена номограмма с двумя бинарными полями (рис. 1), которую можно использовать как для планируемой к разработке компьютерной программы KLASS, так и для других прикладных компьютерных программ, выбирая в зависимости от их назначения усредненное значение коэффициента K_{Σ} [7].

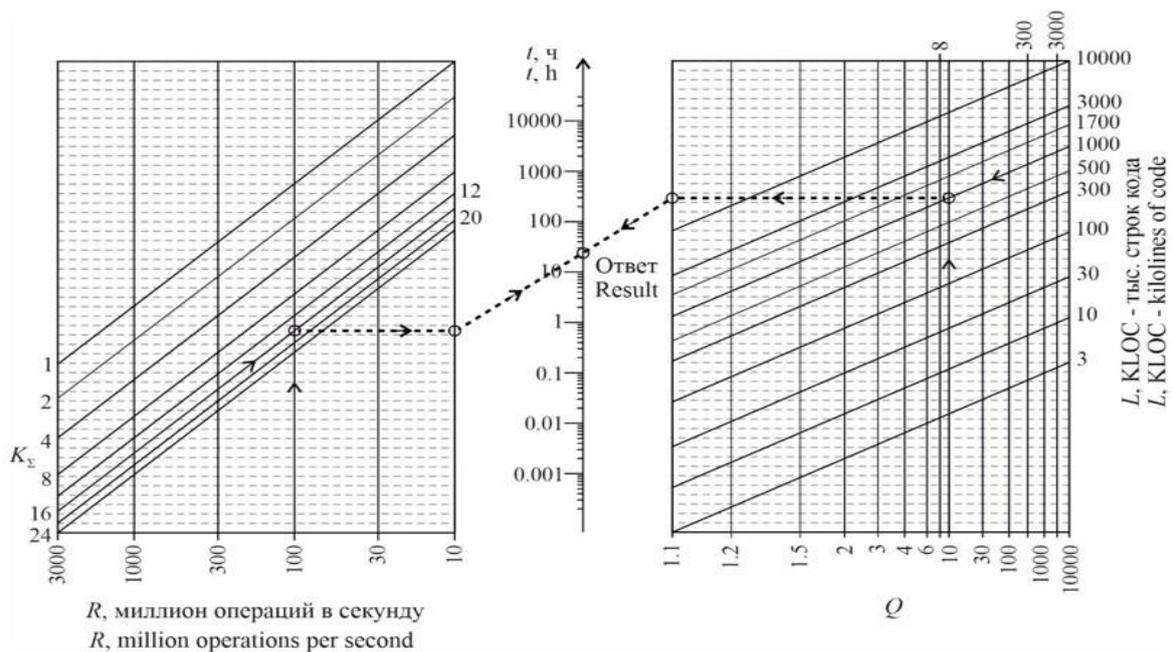


Рис. 1. Номограмма определения времени тестирования прикладных компьютерных программ
Fig. 1. Nomogram for determining the time for testing applied computer programs

Правила использования номограммы для получения значения процессорного времени t понятны из примера (рис. 1), в котором использованы следующие значения величин: $K_{\Sigma} = 16$; $R = 100$ млн операций/с, $L = 1000$ тыс. строк кода, $Q = 10$. Значение t , полученное по номограмме, примерно равно 23...25 ч и соответствует коэффициентам $E_L = 1$ и $E_{\text{ц}} = 1$. Выбор E_L определяется языком написания программного кода, например для Си $E_L \geq 2,5$; для Fortran, Cobal $E_L \geq 3,0$; для Ada $E_L \geq 4,5$; для Си++ $E_L \geq 6,0$. В случае неопределенности рекомендуется принять $E_L = 10$ [7]. Значение $E_{\text{ц}}$ выбирается на основе экспертной оценки с учетом особенностей входных данных, наличия в компьютерной программе циклов, ветвлений и условных переходов. При использовании номограммы найденное значение t следует умножить на произведение $E_L E_{\text{ц}}$.

Экспериментальные данные, приведенные в [8], позволили получить средний процент процессорного времени в общем календарном времени этапа тестирования прикладных

компьютерных программ в предположении, что для тестирования используются общепринятые методы и технологии и общее календарное время тестирования составляет 40 и более процентов от всего календарного времени, планируемого на разработку программы. Для прикладных компьютерных программ, используемых для управления производственными процессами, этот показатель составил примерно 14 %, а среднее значение по прикладным компьютерным программам всех областей применения – 12 %, что хорошо согласуется с числом (примерно 15 %), указанным IT-специалистами НАН Беларуси.

Календарное рабочее время тестирования $T_{\text{календ}}$ для компьютерной программы, используемой в сфере управлениями производственными процессами, определится как

$$T_{\text{календ}} = \frac{100t}{14} \approx 7,1t. \quad (13)$$

Используя модель (12) и формулу (13), для планируемой к разработке программы KLASS получено: $t = 25,5$ ч; $T_{\text{календ}} = 181,0$ ч, что примерно соответствует одному месяцу при пятидневной рабочей неделе. Числа получены при следующих значениях параметров: $K_{\Sigma} = 8,83$; $R = 500$ млн операций/с; $L = 20\,000$ строк кода; $E_L = 10$; $E_{\Sigma} = 20$; $Q = 3,36$.

Заключение

Предложена модель определения времени тестирования прикладных компьютерных программ для обеспечения их заданной эксплуатационной надежности. Модель использована для определения времени тестирования планируемой к разработке компьютерной программы KLASS, предназначенной для работы в составе программного комплекса АРИОН-плюс для автоматизированной оценки надежности изделий электроники, включая полупроводниковые приборы, по значениям их информативных параметров. На основе предложенной модели построена номограмма с двумя бинарными полями, позволяющая при необходимости проследить изменение процессорного времени тестирования в зависимости от быстродействия процессора компьютера, области применения программы (коэффициента K_{Σ}), объема программного кода, коэффициента эффективности тестирования.

Список литературы

1. Разработка методики прогнозирования надежности электронных устройств для системы АРИОН / С. М. Боровиков [и др.] // Доклады БГУИР. 2011. № 4. С. 93–100.
2. Боровиков, С.М. Индивидуальное прогнозирование надежности транзисторов большой мощности для электронных устройств медицинского назначения / С. М. Боровиков, В. О. Казючиц // Доклады БГУИР. 2021. Т. 19, № 1. С. 88–95. <http://dx.doi.org/10.35596/1729-7648-2021-19-1-88-95>.
3. Боровиков, С. М. Статистическое прогнозирование для отбраковки потенциально ненадежных изделий электронной техники / С. М. Боровиков. М.: Новое знание, 2013.
4. Прогнозирование надежности изделий электронной техники / С. М. Боровиков [и др.]. Минск: МГВРК, 2010.
5. Метод прогнозирования надежности изделий электронной техники / С. М. Боровиков [и др.] // Доклады Национальной академии наук Беларуси. 2006. Т. 50, № 4. С. 105–109.
6. Прогнозирование надежности изделий электронной техники методом пороговой логики / С. М. Боровиков [и др.] // Доклады БГУИР. 2006. № 2. С. 49–56.
7. Оценка ожидаемой надежности 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>.
8. Software Reliability, Measurement, and Testing Guidebook for Software Reliability Measurement and Testing [Electronic Resource] / J. A. McCall [et al.]. 1992. Mode of access: <https://apps.dtic.mil/dtic/tr/fulltext/u2/a256164.pdf>. Date of access: 20.04.2022.
9. Шубинский, И. Б. Функциональная надежность информационных систем. Методы анализа / И. Б. Шубинский. М.: Журнал Надежность, 2012.

References

1. Borovikov S. M., Shneiderov E. N., Matyushkov V. E., Tsyrelchuk I. N. (2011) Development of a Method for Predicting the Reliability of Electronic Devices for the ARION System. *Doklady BGUIR = Doklady BGUIR*. (4), 93–100 (in Russian).
2. Borovikov S. M., Kazyuchicz V. O. (2021) Individual Prediction of the Reliability of Semiconductor Devices for Electronic Devices of Medical Purposes. *Doklady BGUIR = Doklady BGUIR*. 19 (1), 88–95. <http://dx.doi.org/10.35596/1729-7648-2021-19-1-88-95> (in Russian).
3. Borovikov S. M. (2010) *Statistical Forecasting for the Rejection of Potentially Unreliable Electronic Products*. Moscow, New Knowledge Publ. (in Russian).
4. Borovikov S. M., Tsyrelchuk I. N., Shneiderov E. N., Beresnevich A. I. (2010) *Predicting the Reliability of Electronic Products*. Minsk, MGVRK Publ. (in Russian).
5. Borovikov S. M., Beresnevich A. I., Khmyl A. A., Emelyanov A. V., Tsyrelchuk I. N. (2006) Method for Predicting the Reliability of Electronic Products. *Doklady Natsional'noi Akademii Nauk Belarusi = Doklady of the National Academy of Sciences of Belarus*. 50 (4), 105–109 (in Russian).
6. Borovikov S. M., Beresnevich A. I., Khmyl A. A., Emelyanov A. V., Tsyrelchuk I. N. (2006) Predicting the Reliability of Electronic Products Using the Threshold Logic Method. *Doklady BGUIR = Doklady BGUIR*. (2), 49–56 (in Russian).
7. Borovikov S. M., Kazyuchits V. O., Khoroshko V. V., Dik S. S., Klinov K. I. (2021) Estimation of the Expected Reliability of Applied Software for Computer Information Systems. *Informatics = Informatics*. 18 (1), 84–95. <https://doi.org/10.37661/1816-030.1-2021-18-1-84-95> (in Russian).
8. McCall J. A. et al. (1992) *Software Reliability, Measurement, and Testing Guidebook for Software Reliability Measurement and Testing*. Available at: <https://apps.dtic.mil/dtic/tr/fulltext/u2/a256164.pdf> (Accessed 20 April 2022).
9. Shubinsky I. B. (2012) *Functional Reliability of Information Systems. Analysis Methods*. Moscow, Magazine Reliability Publ. (in Russian).

Вклад авторов

Все авторы в равной степени внесли вклад в написание статьи.

Authors' contribution

All authors have equally contributed to writing the article.

Сведения об авторах

Казючиц В. О., м.т.н., аспирант Белорусского государственного университета информатики и радиоэлектроники.

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

Шнейдеров Е. Н., к.т.н., доцент, проректор по учебной работе Белорусского государственного университета информатики и радиоэлектроники.

Information about the authors

Kazyuchyts V. O., M. Sci, Postgraduate of the Belarusian State University of Informatics and Radioelectronics.

Borovikov S. M., Cand. of Sci., Associate professor of the Department of Information and Computer Systems Design of Belarusian State University of Informatics and Radioelectronics.

Shneiderov E. N., Cand. of Sci., Vice-Rector for Academic Affairs of the Belarusian State University of Informatics and Radioelectronics.

Адрес для корреспонденции

220013, Республика Беларусь,
г. Минск, ул. П. Бровки, 6
Белорусский государственный университет
информатики и радиоэлектроники
Тел. +375 17 293-88-38
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 17 293-88-38
E-mail: bsm@bsuir.by
Borovikov Sergei Maksimovich