

# МОДЕЛЬ ПРОГНОЗИРОВАНИЯ ВРЕМЕНИ ТЕСТИРОВАНИЯ ПРИКЛАДНЫХ КОМПЬЮТЕРНЫХ ПРОГРАММ ДЛЯ АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ

Боровиков С. М., Казючиц В. О., Дик С. С., Юрениа К. В.  
Кафедра проектирования информационно-компьютерных систем,  
Белорусский государственный университет информатики и радиоэлектроники  
Минск, Республика Беларусь  
E-mail: {bsm, vladkaz}@bsuir.by, kytech@yandex.by

*Прикладные компьютерные программы, используемые в автоматизированных системах управления, содержат сотни тысяч-миллионы строк программного кода и после написания этих программ в них всегда имеются скрытые ошибки, определяющие уровень надёжности программ. Уровень надёжности компьютерной программы перед началом её тестирования обычно не отвечает требованиям практики. Задача этапа тестирования состоит в том, чтобы выявить и устранить наиболее критичные ошибки и свести долю скрытых ошибок к минимуму. Возникает вопрос, сколько времени необходимо выполнять тестирование. На основе статистических данных о тестировании и надёжности компьютерных программ разного функционального назначения, в том числе используемых в автоматизированных системах управления, получена модель, позволяющая определять прогнозируемое время тестирования, обеспечивающее для разрабатываемой программы заданный уровень её эксплуатационной надёжности.*

Работа проводилась в рамках договора № Ф20МВ-021 на выполнение НИР «Статистические модели надёжности прикладных программных средств и их использование для оценки ожидаемой безотказности компьютерных программ на ранних этапах их разработки» в соответствии с решением научного совета БРФФИ по результатам конкурса «БРФФИ-Минобразование М-2020 (протокол № 1 от 22.04.2020).

## ВВЕДЕНИЕ

Автоматизированные системы управления (АСУ) относятся к сложным информационно-компьютерным системам, в которых вклад программного обеспечения в ненадёжность систем может составлять 40 и более процентов [1]. Качество работы АСУ зависит от надёжности используемых прикладных компьютерных программ. Для определения целесообразности разработки и оценки эффективности работы проектируемой АСУ в заданных условиях необходимо на ранних этапах её проектирования оценить ожидаемую эксплуатационную надёжность планируемых к разработке в составе АСУ прикладных компьютерных программ. Под эксплуатационной надёжностью компьютерной программы будем понимать тот уровень надёжности, который она покажет на начальном этапе эксплуатации, то есть после выполнения этапа тестирования и отладки программы.

### I. АКТУАЛЬНОСТЬ

Прикладные компьютерные программы, используемые в АСУ, содержат сотни тысяч-миллионы строк программного кода и поэтому после написания программ в них всегда имеются скрытые ошибки. Наличие этих ошибок определяют уровень надёжности компьютерных программ. После написания кода компьютерной программы и устранения нарушений правил языка программирования программа характеризуется начальным уровнем надёжности, который

как правило не отвечает требованиям практики. Поэтому далее выполняют этап тестирования и отладки программы. Задача этого этапа состоит в том, чтобы выявить и устранить наиболее критичные ошибки, свести к минимуму долю скрытых ошибок, оставшихся в компьютерной программе, и в итоге обеспечить приемлемый уровень эксплуатационной надёжности компьютерной программы. Разработчики АСУ хотели бы знать ожидаемый уровень эксплуатационной надёжности разрабатываемой прикладной компьютерной программы ещё до написания кода программы и выполнения её тестирования. Разработчиков АСУ интересуют также финансовые затраты и/или затраты времени, которые потребуются для достижения заданного уровня надёжности прикладной компьютерной программы.

### II. ИСХОДНЫЕ ПОЛОЖЕНИЯ И ПРЕДПОСЫЛКИ

Оценка надёжности прикладной компьютерной программы до написания её программного кода была рассмотрена в [2]. Подход основан на статистических моделях ожидаемого числа возможных ошибок в компьютерной программе (некоторые специалисты используют термин «дефекты проектирования программы»), и использовании некоторых параметров будущей компьютерной программы, а также данных о влиянии на её надёжность различных факторов и предполагаемое в дальнейшем тестирова-

ние компьютерной программы:

$$\lambda_{exp} = \frac{60}{Q} K_s \frac{R}{B} F_0 L * 10^{-6} = \frac{60}{Q} K_s V N_0 * 10^{-6}, \quad (1)$$

где  $\lambda_{exp}$  - ожидаемая эксплуатационная интенсивность отказов ( $ч^{-1}$ ) прикладной компьютерной программы;  $Q$  - коэффициент эффективности тестирования компьютерной программы;  $K_s$  - суммарный коэффициент увеличения интенсивности отказов, обусловленный совместным действием изменчивости входных данных и рабочей нагрузки на компьютерную программу;  $R$  - пиковое быстродействие процессора, указываемое производителем в технической документации, размерность: операций в секунду;  $B$  - количество команд (операторов) в компьютерной программе;  $F_0$  - начальная плотность ошибок в программе;  $L$  - объём компьютерной программы в исполняемых строках программного кода;  $V$  - скорость выполнения команд компьютерной программы (скорость для пикового быстродействия процессора  $R$ ), определяемая отношением  $V = R/B$ ;  $N_0$  - число ошибок в компьютерной программе до выполнения тестирования. Параметр  $F_0$  в выражении (1) - это плотность ошибок компьютерной программы после написания её кода и устранения нарушений правил языка программирования, представляет собой среднее число ошибок, приходящихся на строку кода:  $F_0 = N_0/L$ . Модель (1) учитывает отрасль применения компьютерной программы, быстродействие процессора компьютера, изменчивость входных данных, а также рабочую нагрузку, которую будет воспринимать компьютерная программа со стороны эксплуатационной среды (ввод-вывод данных и нахождение этих операций в очереди, наличие состояний ожиданий, загрузка-выгрузка программы и/или её модулей из памяти и т.д.). Рассматривается получение параметров модели. Из работы [2] следует, что для достижения заданного уровня эксплуатационной надёжности компьютерной программы необходимо обеспечить нужное значение коэффициента эффективности тестирования  $Q$ , определяемого отношением [2].

$$Q = \frac{N_0}{N_t}, \quad (2)$$

где  $N_t$  - число оставшихся ошибок в прикладной компьютерной программе после завершения её тестирования.

### III. МОДЕЛЬ ПРОГНОЗИРОВАНИЯ ВРЕМЕНИ ТЕСТИРОВАНИЯ КОМПЬЮТЕРНОЙ ПРОГРАММЫ

Для получения модели прогнозирования, используемой для определения времени, требуемого для проведения тестирования планируемой к разработке прикладной компьютерной программы, необходимо величину  $N_{п.тест}$  в отношении (2) выразить через время тестирования  $t$  и далее решить полученное уравнение относительно  $t$ . В докладе с использованием модели (1), отношения (2) и положений, приводимых в [3], модель для прогнозирования времени тестирования компьютерной программы получена в виде

$$t = \frac{LE_L \ln(Q)}{60K_s R}, \quad (3)$$

где  $E_L$  - коэффициент расширения кода, показывающий увеличение числа команд компьютерной программы относительно числа строк программного кода  $L$ . Значение  $E_L$  зависит от используемого языка программирования, например для *C*  $E_L = 2,5$ ; для *Fortran*, *Cobal*  $E_L = 3,0$ ; для *Ada*  $E_L = 4,5$ ; для *C++*  $E_L = 6,0$ . Коэффициент  $K_s$ , рассчитанный по экспериментальным данным [4] для компьютерных программ, используемых для АСУ, принял значение  $K_s = 19,2$ . В модели (3) в качестве  $t$  рассматривается процессорное время выполнения компьютерной программы. В дальнейшем при необходимости время выполнения компьютерной программы  $t$  может быть преобразовано в календарное время.

### IV. СПИСОК ЛИТЕРАТУРЫ

1. Чуканов, В. О. Методы обеспечения аппаратно-программной надежности вычислительных систем / В. О. Чуканов, В. В. Гуров, Е. В. Прокопьева [Электронный ресурс]. - 2014. - Режим доступа : <http://www.mcst.ru/metody-obespecheniya-apparatnoprogrammnoj-nadezhnosti-vychislitelnykh-sistem>. - Дата доступа: 4 апреля 2021.
2. Borovikov S. M., Kaziuchyts V. O., Khoroshko V. V., Dick S. S., Klinov K. I. Assessment of expected reliability of applied software for computer-based information systems. *Informatics*, 2021, vol. 18, no. 1, pp. 84–95 (in Russian). <https://doi.org/10.37661/1816-0301-2021-18-1-84-95>
3. Шубинский, И. Б. Функциональная надёжность информационных систем. Методы анализа / И. Б. Шубинский. - М.: «Журнал Надежность», 2012. - 296 с.
4. McCall, J. A. 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: 20.10.2021.