

МОДЕЛЬ ПРОГНОЗИРОВАНИЯ ВРЕМЕНИ ТЕСТИРОВАНИЯ ПРИКЛАДНЫХ КОМПЬЮТЕРНЫХ ПРОГРАММ ДЛЯ СИСТЕМ ТЕЛЕКОММУНИКАЦИЙ

¹Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники», г. Минск, Республика Беларусь

²Учреждение образования «Белорусская государственная академия связи», г. Минск, Республика Беларусь

Работа проводилась в рамках договора № Ф20МВ-021 на выполнение НИР «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 БРФФИ по результатам конкурса «БРФФИ–Минобразование М-2020» (протокол № 1 от 22.04.2020).

Современные системы телекоммуникаций относятся к классу сложных информационно-компьютерных систем, в которых на долю программного обеспечения может приходиться 40 и более процентов общей ненадежности систем [1]. Важно на ранних этапах проектирования систем оценить ожидаемую эксплуатационную надежность планируемых к разработке прикладных компьютерных программ.

В работе [2] модель оценки ожидаемой эксплуатационной интенсивности отказов прикладной компьютерной программы ($\lambda_{\text{экс}}$) получена в виде:

$$\lambda_{\text{экс}} = \frac{60}{Q} K_{\Sigma} \frac{R}{B} F_0 L \cdot 10^{-6} = \frac{60}{Q} K_{\Sigma} V N_0 \cdot 10^{-6}, \text{ ч}^{-1}, \quad (1)$$

где Q – коэффициент эффективности тестирования компьютерной программы; K_{Σ} – суммарный коэффициент увеличения интенсивности отказов, обусловленный совместным действием изменчивости входных данных и рабочей нагрузки на компьютерную программу (для прикладных компьютерных программ, используемых в телекоммуникациях, $K_{\Sigma} \approx 11,5$ [3]); R – пиковое быстродействие процессора, указываемое производителем в технической документации; B – количество команд (операторов) в компьютерной программе; F_0 – начальная плотность ошибок в компьютерной программе; L – объем компьютерной программы в исполняемых строках программного кода; V – скорость выполнения команд компьютерной программы (скорость для пикового быстродействия 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 ее кода и устранения нарушений правил языка программирования, представляет собой среднее число ошибок, приходящихся на строку кода: $F_0 = N_0 / L$.

Модель (1) получена для случая, когда пиковое быстродействие процессора R будет подставляться в размерности «миллионов операций в с (секунду)», а реальное быстродействие процессора относительно пикового значения R , указываемого производителем, составляет примерно 70 процентов.

Согласно [2] коэффициент Q показывает, во сколько раз за счет тестирования уменьшится число оставшихся ошибок в компьютерной программе относительно начального их числа N_0 (до выполнения тестирования):

$$Q = \frac{N_0}{N_{\text{п.тест}}} = \frac{\lambda_0}{\lambda_{\text{экс}}}, \quad (2)$$

где $N_{п.тест}$ – число оставшихся ошибок в прикладной компьютерной программе после завершения ее тестирования; λ_0 – интенсивность отказов прикладной компьютерной программы до начала тестирования.

Вопрос о длительности этапа тестирования прикладных компьютерных программ остался открытым. Актуальной является задача по определению времени выполнения тестирования, которое обеспечит необходимое значение коэффициента эффективности тестирования Q , при достижении которого прикладная компьютерная программа предположительно будет отвечать требованию эксплуатационной надежности и ее можно будет применять по функциональному назначению.

Модель прогнозирования, которая может использоваться для определения времени, требуемого для проведения тестирования планируемой к разработке прикладной компьютерной программы, получена в предположении, что суммарное число ошибок, обнаруженных в компьютерной программе при ее тестировании $n_{обн}$ ($n_{обн} = N_0 - N_{п.тест}$), описывается экспоненциальной функцией суммарного времени функционирования программы при ее тестировании [4]:

$$n_{обн} = N_0 \left[1 - \exp\left(-\frac{\lambda_0}{N} t\right) \right], \quad (3)$$

где t – суммарное время выполнения компьютерной программы при ее тестировании; N – общее число ошибок, которые гипотетически могут проявиться в процессе тестирования компьютерной программы (для простоты принято $N \approx N_0$).

Здесь в качестве t рассматривается процессорное время выполнения компьютерной программы. В дальнейшем при необходимости процессорное время выполнения программы t может быть преобразовано в календарное время.

С учетом выражения (3) и формулы определения λ_0 по модели Муса [4], а также положений, вводимых в работе [2] для коэффициента проявления ошибок компьютерной программы, модель для расчета времени t выполнения компьютерной программы при ее тестировании для обеспечения требуемого значения коэффициента эффективности тестирования Q получена в виде:

$$t = \frac{L E_L \ln(Q)}{60 K_{\Sigma} R}, \quad (4)$$

где E_L – коэффициент расширения кода, показывающий увеличение числа команд программы относительно числа строк программного кода L (зависит от используемого языка программирования, например для «Си» – $E_L = 2,5$; для «Си++» – $E_L = 6,0$).

В модель (4) пиковое быстродействие процессора компьютера R должно подставляться в размерности «миллионов операций в с». Прогнозное время t выполнения компьютерной программы при ее тестировании будет получено в ч (часах).

Согласно экспериментальным данным [5], при общепринятом подходе к тестированию компьютерных программ средний процент их выполнения (прогона) в течение рабочего календарного времени $T_{календ}$, отводимого на тестирование компьютерной программы, составляет примерно 12 %. Следовательно, $T_{календ} \approx 100 \cdot t / 12$.

Модель (4) позволяет спрогнозировать требуемое время тестирования планируемой к разработке компьютерной программы для обеспечения заданного уровня ее эксплуатационной надежности. Такое прогнозирование полезно, так как позволяет еще до использования прикладной компьютерной программы получить представление о ее ожидаемой надежности и оценить предполагаемые для этого затраты времени.

ЛИТЕРАТУРА

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: 18.09.2021.
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. Модель оценки надежности прикладных компьютерных программ для систем телекоммуникаций / С. М. Боровиков [и др.] // Современные средства связи : материалы XXV Международной научно-технической конференции, Минск, 22–23 октября 2020 г. / Белорусская государственная академия связи ; редкол.: А. О. Зеневич [и др.]. – Минск, 2020. – С. 163–165.

Инфокоммуникации и информационные технологии

4. Шубинский, И. Б. Функциональная надежность информационных систем. Методы анализа / И. Б. Шубинский. – М. : «Журнал Надежность», 2012. – 296 с.

5. 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: 18.09.2021.