

МЕТОД ОЦЕНКИ ОЖИДАЕМОЙ НАДЁЖНОСТИ ПРИКЛАДНЫХ КОМПЬЮТЕРНЫХ ПРОГРАММ

Силенкова Д.С., Зацепина М.И.

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

Научный руководитель: Боровиков С.М. – к.т.н., доцент, доцент кафедры ПИКС

Аннотация. В работе рассматривается метод оценки ожидаемой надёжности прикладной компьютерной программы до начала написания кода на языке программирования. Метод учитывает особенности организации, разрабатывающей программное обеспечение, квалификацию программистов и основные характеристики планируемой к разработке компьютерной программы. На конкретном примере проиллюстрировано применение метода.

Ключевые слова: надёжность прикладной компьютерной программы, интенсивность проявления скрытых дефектов, вероятность безотказной работы программы

Введение. Оценка эксплуатационной надёжности ПО на этапе проектирования критична для определения рентабельности разработки и итоговых характеристик системы.

Цель работы – расчет ожидаемой надёжности прикладного ПО на конкретном примере..

Основная часть. Для иллюстрации метода расчета ожидаемой надёжности прикладной компьютерной программы использованы следующие исходные данные:

– прогнозный объём компьютерной программы $L = 30$ KLOC (KLOC – англоязычный вариант размерности «тысяч строк программного кода»);

– календарное рабочее время тестирования компьютерной программы $T = 700$ ч;

– средний процент времени прогона компьютерной программы (r) при её тестировании, зависит от области применения программы, принято $r = 2,5\%$ [1, 3];

– заданный период эксплуатации τ , для которого определяют проектный уровень эксплуатационной надёжности компьютерной программы, $\tau = 720$ ч;

– прогнозное число обращений в час к компьютерной программе (η) в течение заданного периода эксплуатации $\tau = 720$ ч, принято $\eta = 2 \text{ ч}^{-1}$;

– указанное в технической документации пиковое быстродействие процессора, используемого в компьютере, $R = 5 \cdot 10^8$ операций в секунду;

– язык программирования для написания программного кода: C++.

Для выбора коэффициентов, определяющих метрики компьютерной программы использованы работы [1–3]. Последовательность оценки ожидаемой надёжности программы:

1 Уточняем метрику производственной среды разработки компьютерной программы (D):

$$D = K_{\text{орг}} \cdot K_{\text{квал}} = 0,76 \cdot 1,3 = 0,988. \quad (1)$$

где $K_{\text{орг}}$ – коэффициент, учитывающий особенность проектной организации; принято $K_{\text{орг}} = 0,76$ (организация, эксплуатирующая компьютерную информационную систему) [1, 3];

$K_{\text{квал}}$ – коэффициент, учитывающий квалификацию программистов, принято $K_{\text{квал}} = 1,3$ (младший программист – Junior Developer) [1, 3].

2 Определяем коэффициент сложности программного обеспечения ($K_{\text{слож}}$), учитывающий наличие факторов (m), влияющих на трудоёмкость разработки и тестирования компьютерной программы:

$$K_{\text{слож}} = 1 + \sum_{i=1}^m k_i = 1 + k_1 + k_2 = 1 + 0,08 + 0,06 = 1,14, \quad (2)$$

где k_i – коэффициент повышения сложности прикладной компьютерной программы за счёт наличия у нее i -й характеристики; принято $m = 2$, коэффициенты k_i получены по [2].

3 Находим метрику S , учитывающую характеристики разрабатываемой компьютерной программы: влияние сложности ($K_{\text{слож}}$), новизны ($K_{\text{нов}}$), используемых средств разработки ($K_{\text{ср.разр}}$) и степени использования в программе готовых программных модулей ($K_{\text{мод}}$). Коэффициенты выбраны с учётом документа [2].

$$S = K_{\text{слож}} \cdot K_{\text{нов}} \cdot K_{\text{ср.разр}} \cdot K_{\text{мод}} = 1,14 \cdot 0,81 \cdot 1,2 \cdot 0,77 \approx 0,853. \quad (3)$$

4 Определяем начальную плотность ошибок в компьютерной программе (F_0), характеризующую ожидаемое количество скрытых дефектов в программном коде после устранения ошибок, вызванных нарушением правил языка программирования:

$$F_0 = F_B \cdot D \cdot S = 8,5 \cdot 0,988 \cdot 0,853 = 7,163 \text{ ошибок/KLOC}, \quad (5)$$

где $F_B = 8,5$ ошибок/KLOC – базовая (усреднённая) плотность ошибок для прикладных компьютерных программ, предназначенных для АСУ [1, 3].

5 Даём оценку прогнозируемому значению количества команд, выполняемых программой:

$$B = L \cdot E_L \cdot E_{\text{ц}} = 30000 \cdot 6 \cdot 30 = 5,4 \cdot 10^6 \text{ команд}, \quad (6)$$

где E_L – коэффициент расширения кода программы относительно числа строк кода L , определяется используемым языком программирования, для (C++) $E_L \geq 6,0$.

$E_{\text{ц}}$ – коэффициент увеличения числа выполняемых процессором команд за счёт наличия в программе условных переходов, ветвлений и других особенностей (ввод-вывод данных, считывание с внешних носителей, ожидание в очереди, обработка прерываний).

Значение $E_{\text{ц}}$ выбирается на основе экспертной оценки с учетом алгоритма программы, характера входных данных и сложности операций, а при неопределенности рекомендуется диапазон 20–50 (принято $E_{\text{ц}} = 30$).

6 Определяем пиковую скорость выполнения команд компьютерной программы ($V_{\text{пик}}$) для пикового быстродействия процессора:

$$V_{\text{пик}} = \frac{R}{B} = \frac{5 \cdot 10^8}{5,4 \cdot 10^6} = 92,5926 \text{ с}^{-1}. \quad (7)$$

7 Находим начальную интенсивность отказов λ_0 (интенсивность проявления скрытых ошибок), т.е. интенсивность отказов программы до начала её тестирования:

$$\lambda_0 = 60 \cdot K_{\Sigma} \cdot V_{\text{пик}} \cdot F_0 \cdot L \cdot 10^{-6} = 60 \cdot 19,2 \cdot 92,5926 \cdot 7,163 \cdot 30 \cdot 10^{-6} = 22,9216 \text{ ч}^{-1}, \quad (8)$$

где K_{Σ} – коэффициент увеличения интенсивности отказов из-за суммарного действия изменчивости входных данных и рабочей нагрузки на компьютерную программу со стороны эксплуатационной среды (ввод данных, использование принтера, ожидание операций в очереди и т. д); для компьютерных программ, используемых для АСУ, $K_{\Sigma} = 19,2$ [1, 3].

8 Находим проектное значение коэффициента тестирования Q компьютерной программы в предположении, что тестирование программы выполняется в течение заданного (бюджетного) календарного рабочего времени T [1]:

$$Q = \exp \left[\frac{60 \cdot 10^{-6} \cdot K_{\Sigma} \cdot R \cdot T \cdot (r/100)}{B} \right] = \exp \left[\frac{60 \cdot 10^{-6} \cdot 19,2 \cdot (5 \cdot 10^8) \cdot 700 \cdot (2,5/100)}{5,4 \cdot 10^6} \right] = 6,47. \quad (9)$$

9 Определяем эксплуатационную интенсивность отказов компьютерной программы ($\lambda_{\text{экс}}$):

$$\lambda_{\text{экс}} = \frac{\lambda_0}{Q} = \frac{22,9216}{6,47} = 3,5428 \text{ ч}^{-1}. \quad (10)$$

10 Вероятность безотказной работы программы в течение календарного времени τ (принята во внимание работа [1] и документ [4]):

$$P(\tau) = (P_1)^{\eta \tau} = \left[\exp \left(-\lambda_{\text{экс}} \cdot \frac{B}{(3600 \cdot 0,7 \cdot R)} \right) \right]^{\eta \tau} = \left[\exp \left(-3,5428 \cdot \frac{5,4 \cdot 10^6}{(3600 \cdot 0,7 \cdot 5 \cdot 10^8)} \right) \right]^{2,720} = 0,978373 \approx 0,98. \quad (11)$$

где P_1 – вероятность безотказной обработки одного усредненного набора исходных данных.

Заклучение. Иллюстрация применения метода примером позволит лучше осмыслить подход, используемый для оценки ожидаемой надёжности планируемой к разработке прикладной компьютерной программы.

Список литературы

1. Боровиков, С. М. Надёжность разрабатываемых прикладных компьютерных программ для информационных систем = Reliability of developed application computer programs of information systems / С. М. Боровиков, А. В. Будник, В. Т. Лё // Проблемы инфокоммуникаций : научный журнал. – 2025. – № 1. – С. 25–34.
2. Постановление министерства труда и социальной защиты Республики Беларусь 27 июня 2007 г. № 91 «Об утверждении укрупнённых норм затрат труда на разработку программного обеспечения» [Электронный ресурс]. – URL: <https://news-newsby-org.narod.ru/doc-rzn/pos05/postn05236/index.htm> (дата обращения 24.02.2025).
3. McCall J.A., Randell W., Dunham J., Lauterbach L. Software reliability, measurement, and testing guidebook for software reliability measurement and testing // RL-TR-92-52: Final Technical Report, Vol II (of two) / Science Applications International Corp., Research Triangle Institute. – Rome Laboratory, NY 13441-5700, 1992. – 256 p. – URL: <https://apps.dtic.mil/sti/tr/pdf/ADA256164.pdf> (дата обращения 24.02.2025).
4. ГОСТ 27.205-97 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.

UDC УДК 004.4' 242:004.6-027.45

METHOD FOR ASSESSING THE EXPECTED RELIABILITY OF APPLIED COMPUTER PROGRAMS

Silenkova D.S., Zatsepina M.I.

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

Borovikov S.M. – Cand. of Sci., associate professor, associate professor of the department of ICSD

Annotation. This paper examines a method for assessing the expected reliability of an application program before coding it in a programming language. The method takes into account the characteristics of the software development organization, the qualifications of the programmers, and the key characteristics of the planned computer program. A specific example illustrates the application of the method.

Keywords: software reliability, the intensity of the manifestation of hidden defects, the probability of failure-free operation of the program