

МЕТОД РАСЧЕТА ОЖИДАЕМОЙ НАДЕЖНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

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 надежности программного обеспечения, основанный на использовании метрик производственной среды разработки и характеристик программного продукта, а также на модели перехода от прогнозной плотности ошибок к вероятности безотказной работы в течение заданного периода времени. Приводится последовательность вычислений и демонстрируется применение метода на примере исходных данных.

Ключевые слова: надёжность программного обеспечения, интенсивность отказов, вероятность безотказной работы, методика расчёта.

Введение. Оценка надежности программного обеспечения является важной задачей анализа качества систем. При ограниченных времени и ресурсах экспериментальные методы не всегда применимы, поэтому используются расчетные подходы, позволяющие получить прогнозные показатели на основе формализованных моделей и исходных данных.

Целью данной работы является демонстрация порядка последовательного определения ожидаемой надежности программного обеспечения с использованием нормативных коэффициентов и заданных исходных данных.

Основная часть. Для демонстрации метода расчета ожидаемой надежности программного обеспечения используется следующий набор исходных данных: прогнозный объём программного обеспечения $L=28$ KLOC, календарное время тестирования $T=400$ ч, средний процент времени прогона программы $\tau=2,5\%$, расчетный период эксплуатации $t=720$ ч, прогнозируемое число обращений к программе $\eta=1$ ч⁻¹, производительность вычислительной системы $R=5 \cdot 10^6$ операций в секунду. Язык реализации C++, коэффициенты расширения и ветвления принимаются согласно справочным данным. Базовая плотность ошибок принята равной $F_0=6,0$ ошибок на KLOC. Коэффициенты организационного уровня и квалификации разработчиков приняты: $K_{орг}=1,00$ и $K_{кв}=1,00$.

Метрика производственной среды разработки определяется по формуле:

$$D = K_{орг} * K_{квал} = 1 * 1 = 1, \quad (1)$$

где $K_{орг}$ – коэффициент, учитывающий особенность проектной организации;
 $K_{квал}$ – коэффициент, учитывающий квалификацию и опыт программистов.

Коэффициент сложности программного обеспечения формируется с учетом наличия факторов, влияющих на трудоёмкость разработки и тестирования, и вычисляется по формуле:

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

где k_i – коэффициент повышения сложности прикладной компьютерной программы за счет наличия у нее i -й характеристики.

Метрика характеристик программного продукта отражает совокупное влияние сложности, новизны и используемых средств разработки и определяется выражением:

$$S = K_{\text{слож}} * K_{\text{нов}} * K_{\text{ср.разр}} * K_{\text{мод}} = 1,14 * 0,81 * 1,2 * 0,65 = 0,72 \quad (3)$$

где $K_{\text{нов}}$ – коэффициент, учитывающий степень новизны компьютерной программы;
 $K_{\text{ср.разр}}$ – коэффициент средств разработки компьютерной программы;
 $K_{\text{мод}}$ – коэффициент, учитывающий процент использования стандартных модулей в компьютерной программе.

Прогнозная начальная плотность ошибок характеризует ожидаемое количество скрытых дефектов в программном коде и определяется по формуле:

$$F_0 = F_B * D * S = 8,5 * 1 * 0,72 = 6,122 \text{ ошибок/KLOC}, \quad (4)$$

где $F_B = 8,5$ ошибок/KLOC – базовая (усредненная) плотность ошибок для АСУ;
 D – метрика, учитывающая характеристику производственной среды разработки программы;
 S – метрика, учитывающая характеристики разрабатываемой компьютерной программы.

Полученное значение используется в качестве исходного параметра при дальнейшем анализе эксплуатационных характеристик, поскольку наличие скрытых дефектов оказывает непосредственное влияние на вероятность возникновения отказов в процессе функционирования. На следующем этапе методики учитываются динамические характеристики выполнения программы, позволяющие связать статическую оценку дефектности с условиями эксплуатации.

Число выполняемых команд программы определяется выражением:

$$B = L * E_L * E_{\text{ц}} = 28000 * 6 * 40 = 3,36 * 10^6 \text{ команд}, \quad (5)$$

где L – прогнозный объем будущей компьютерной программы в выполняемых строках кода, LOC;
 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 счет наличия в компьютерной программе условных переходов, ветвлений и других особенностей (ввод-вывод данных, считывание информации с внешних носителей, ожидание в очереди, обработка прерываний).

Значение $E_{\text{ц}}$ следует выбирать на основе экспертной оценки с учетом алгоритма выполнения компьютерной программой своих функций, экстремального характера входных данных, числа ветвлений в программе, записи и считывания данных с внешних носителей и т. п. В случае неопределенности рекомендуется выбрать значение от 20 до 50.

Пиковая скорость выполнения программы вычисляется по формуле:

$$V_{\text{пик}} = \frac{R}{B} = \frac{5 * 10^8}{3,36 * 10^6} = 148,81 \text{ с}^{-1}, \quad (6)$$

где R – скорость пикового быстродействия компьютера, операций в секунду;
 B – число команд программы.

Начальная интенсивность отказов программного обеспечения описывает ожидаемую частоту отказов на начальном этапе эксплуатации и определяется выражением:

$$\lambda_0 = 60 * K_{\Sigma} * V_{\text{пик}} * F_0 * L * 10^{-6} = 29,39 \text{ ч}^{-1} \quad (7)$$

Эксплуатационная интенсивность отказов программного обеспечения:

$$\lambda_{\text{экс}} = \frac{\lambda_0}{Q} = \frac{29,3862}{5,55} = 5,29 \text{ ч}^{-1}, \quad (8)$$

где Q – коэффициент эффективности тестирования, который рассчитывается по формуле:

$$Q = \exp\left(\frac{60 \cdot 10^{-6} \cdot K_{\Sigma} \cdot R \cdot t_{\Sigma}}{B}\right) = \exp\left(\frac{60 \cdot 10^{-6} \cdot 19,2 \cdot 5 \cdot 10^8 \cdot 10}{3,36 \cdot 10^6}\right) = 5,55. \quad (9)$$

Коэффициент эффективности тестирования показывает снижение отказов после испытаний и учитывает влияние объема и длительности тестирования на эксплуатационные показатели. Вероятность безотказной работы программного обеспечения в течение календарного времени τ определяется по формуле:

$$P(\tau) = (P_1)^{\eta \cdot \tau} = (\exp(-\lambda_{\text{экс}} \cdot t_1))^{\eta \cdot \tau} = 0,999985887^{720} = 0,990, \quad (10)$$

где η – прогнозное число обращений к программе в час;

P_1 – вероятность безотказной работы одного усредненного набора исходных данных.

Заключение. Рассмотрен метод последовательной оценки ожидаемой надежности программного обеспечения. Приведена последовательность расчёта основных показателей и получено численное значение вероятности безотказной работы за заданный интервал. Метод применим для прогнозной оценки эксплуатационных характеристик ПО в учебных и выпускных работах.

Список литературы

1. Боровиков С.М. Теоретические основы конструирования, технологии и надёжности. – Минск: Дизайн ПРО, 1998. – 336 с.
2. Боровиков С.М., Цырельчук И.Н., Троян Ф.Д. Расчёт показателей надёжности радиоэлектронных средств [Электронный ресурс]. Режим доступа: <https://libeldoc.bsuir.by/handle/123456789/25651>. Дата доступа: 05.01.2026.
3. Надёжность в технике. Термины и определения: ГОСТ 27.002-2015 [Электронный ресурс]. Режим доступа: <https://megaporm.ru/Data2/1/4293754/4293754027.pdf>. Дата доступа: 06.01.2026.

UDC 004.052

METHOD FOR CALCULATING EXPECTED SOFTWARE RELIABILITY

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 considers a computational method for estimating the expected reliability of software. The method is based on the use of development environment metrics and software product characteristics, as well as on a model for transforming the predicted defect density into the probability of failure free operation over a specified time interval. The calculation procedure is presented and illustrated using a set of input data. The proposed approach can be applied in coursework and graduation projects.

Keywords: software reliability, failure intensity, probability of failure free operation, calculation methodology.