

УДК 004.9:004.52

## ПРОГНОЗИРОВАНИЕ ОЖИДАЕМОЙ НАДЁЖНОСТИ ПРИКЛАДНЫХ ПРОГРАММНЫХ СРЕДСТВ С ИСПОЛЬЗОВАНИЕМ СТАТИСТИЧЕСКИХ МОДЕЛЕЙ ИХ БЕЗОТКАЗНОСТИ



**С.М. Боровиков**

*Доцент кафедры проектирования информационно-компьютерных систем, кандидат технических наук, доцент*



**С.С. Дик**

*Аспирант БГУИР, магистр технических наук*

*Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь  
E-mail: bsm@bsuir.by, sdick@bsuir.by*

**Аннотация.** Для прогнозирования ожидаемой надёжности разрабатываемого прикладного программного обеспечения предлагается подход, основанный на использовании статистической модели безотказности программных средств. Модель базируется на взятых из отечественной и зарубежной печати обобщённых статистических данных об ожидаемом числе ошибок в разрабатываемом прикладном программном средстве в зависимости от его объёма и динамики уменьшения числа ошибок от времени тестирования и с учётом профессионального опыта программистов и тестировщиков. Приводится пример прогнозирования ожидаемой надёжности прикладного программного средства.

**Ключевые слова:** прикладное программное средство, ожидаемая надёжность, прогнозирование, статистическая модель безотказности

Современное программное обеспечение довольно сложное, и есть все предпосылки считать, что оно станет ещё сложнее в ближайшем будущем. Например, в 1983 году программа Microsoft Word состояла только из 27 000 строк кода, но, согласно данным Nathan Myhrvold [1], к 1995 году эта программа увеличилась уже до 2 млн. строк кода! Программисты потратили годы на то, чтобы придумать единицы измерения для программного обеспечения. Но только одна единица измерения позволяет установить соотношение с числом ошибок – количество строк кода (LOC). И действительно, в некоторых кругах специалистов по программированию число строк кода стало единственным приемлемым средством измерения объёма программных средств [2].

Количество ошибок на тысячу строк кода (KLOC) изменяется для каждой конкретной программы. Достоверное значение варьируется от 5 до 50 ошибок на 1000 строк кода [3]. Даже в системах, которые прошли строгий контроль качества (в англоязычном наименовании Quality Assurance – QA) все равно содержатся ошибки – приблизительно 5 ошибок на 1000 строк кода [2]. В программе, которая прошла тестирование только на предмет работоспособности функциональных возможностей, что справедливо для большей части коммерческого программного обеспечения, присутствует намного больше ошибок – около 50 ошибок на 1000 строк кода. Большая часть программ попадает в последнюю категорию. Многие поставщики программного обеспечения неверно предполагают, что они выполняют контроль качества в соответствии с QA. В действительности их методы тестирования

оказываются весьма поверхностными. Строгий контроль качества программного обеспечения заключается не только в тестировании возможностей программ, но и должен включать в себя тестовое внесение неисправностей и анализ ошибок.

Чтобы оценить всю сложность современного программного обеспечения, рекомендуется проанализировать информацию, приведённую в таблице 1.

Таблица 1

Объём программного обеспечения

Программное средство (объект)	Космическая станция	Космический корабль	Boeing 777	Windows NT5	Linux	Windows 95	Windows XP
Количество строк кода, млн	40	10	7	35	1,5	5	40

В сложных информационно-компьютерных системах, к которым относятся системы для анализа больших данных (Big Data), вклад программного обеспечения в ненадёжность систем составляет до 40 процентов. По мнению некоторых авторов этот вклад может превышать вклад, вносимый техническими средствами (компьютерами), поскольку входные данные могут быть сложнее и их формат всё время меняется [2]. Надёжность аппаратуры ограничивается ошибками проектирования, производственными дефектами и частотой сбоев (зависит от физических процессов). По своей природе программное обеспечение сложнее технических средств систем. Объём программных средств (далее – ПС) для современных информационно-компьютерных систем оценивается в  $10^6 \dots 10^8$  и более команд или информационных слов.

Применительно к надёжности программного обеспечения ошибка – это погрешность или искажение кода программы, неумышленно внесённое в неё в процессе разработки, которые при определённом наборе входных данных в ходе функционирования этой программы могут вызвать отказ или снижение эффективности функционирования.

Ошибками в ПС являются все возможные несоответствия между демонстрируемыми характеристиками его качества и предписанными требованиями и, иногда, ожиданиями пользователей. Степень некорректности ПС определяется вероятностью попадания реальных исходных данных в область, которая задана требованиями спецификации и технического задания, однако не была проверена при тестировании и испытаниях.

Отказ ПС – это проявление ошибки в нём. Все ошибки в ПС носят детерминированный характер, но вероятностным оказывается процесс внесения ошибок. Проявление ошибок также носит вероятностный характер.

Поскольку программное обеспечение в процессе эксплуатации не изнашивается, его поломка и ремонт в общепринятом смысле не производится, то надёжность программного обеспечения имеет смысл характеризовать только с точки зрения безотказности его функционирования и возможности восстановления функционирования после отказов, вызванных проявлениями ошибок [2].

Согласно стандарту [4] под надёжностью (en reliability) программного средства понимают совокупность свойств, характеризующую способность программного средства сохранять заданный уровень пригодности в заданных условиях в течение заданного интервала времени.

Наиболее приемлемыми показателями, характеризующими безотказность (завершённость, стабильность) программного обеспечения, представляются показатели сходные с показателями безотказности технических систем, такие как интенсивность проявления ошибок (отказов)  $\lambda_{ПС}$  или средняя наработка на проявление ошибки ПС  $T_0$ . Эти показатели связаны между собой и, зная один из них, можно определить другие [5].

Надёжность ПС определяется качеством отладки программы, глубиной её тестирования.

Целью тестирования является не тотальное обнаружение всех ошибок, что принципиально невозможно, а выявление наибольшего количества наиболее критичных ошибок.

При отладке ПС происходит локализация и устранение синтаксических ошибок и явных ошибок кодирования. В процессе же тестирования проверяется работоспособность программы, не содержащей явных ошибок. С помощью тестирования должно быть выявлено как можно больше смысловых ошибок с учётом возможного формата изменения исходных данных. Тестирование требует значительного времени, и даже после его завершения некоторые ошибки в программном средстве остаются необнаруженными.

Известные методы оценки надёжности прикладных ПС исходят из того, что имеются определённые данные о тестировании уже написанного ПС с использованием кода и устранены ошибки языка программирования, т.е. выполнена отладка программы. Однако в большинстве случаев проектировщиков информационно-компьютерных систем и разработчиков программного обеспечения для анализа с помощью этих систем больших данных (Big Data) интересует ожидаемый уровень надёжности прикладных ПС ещё до написания их программного кода. Возникает вопрос, как спрогнозировать ожидаемый уровень надёжности ПС на этом этапе.

Для выполнения оценки ожидаемой надёжности прикладных ПС предлагается специальная методика, использующая в качестве модели надёжности разрабатываемой программы статистический подход к оценке надёжности прикладных ПС. Обобщённые сведения о надёжности прикладных ПС взяты из опыта их использования по назначению в различных областях деятельности людей. Ниже приводятся основные данные, используемые в моделях безотказности разрабатываемых прикладных ПС.

Профессиональные программисты со стажем 10 и более лет на 1000 строк кода допускают в среднем 131,3 ошибки. Из них до 50% выявляется на этапе компиляции (если транслятор имеет развитую систему предупреждений). На этапе тестирования отдельных модулей обнаруживается 50% и более оставшихся ошибок [3].

В ПС, прошедших тестирование, по различным оценкам содержится от 5 до 100 ошибок на 1000 строк исполняемого кода [3].

Согласно [2], даже в ПС, которые прошли строгий контроль качества (Quality Assurance – QA) все равно содержатся ошибки, приблизительно 5 ошибок на 1000 строк кода. В ПС, которые прошли тестирование только на предмет работоспособности функциональных возможностей, что справедливо для большей части коммерческого программного обеспечения, присутствует намного больше ошибок: около 50 ошибок на 1000 строк кода (рисунок 1)

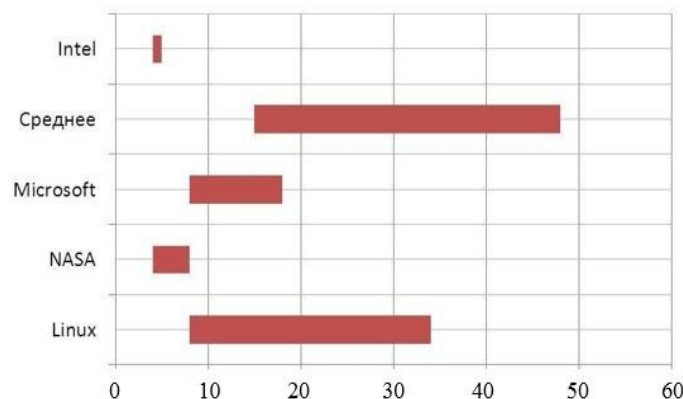


Рисунок 1. Среднее число ошибок на 1000 строк кода для прикладных программных средств, прошедших тестирование

Авторами предлагается алгоритм оценки ожидаемой надёжности прикладного ПС на раннем этапе его разработки, а именно: до написания программного кода и отладки ПС. В

алгоритме в качестве модели оценки ожидаемой надёжности прикладных ПС использованы усреднённые статистические данные об ожидаемом числе ошибок в них в зависимости от их объёма, динамики уменьшения числа ошибок от профессионального опыта программистов и тестировщиков, а также времени тестирования. В качестве основы определения итоговой (прогнозной) надёжности прикладного ПС выбрана модель Шумана [6, 7] в предположении, что вся продолжительность процедуры тестирования программного средства рассматривается как один этап тестирования ( $i = 1$ ). Кроме того, при оценке ожидаемой надёжности ПС с использованием этой модели приняты следующие основные допущения:

- написание программного кода, отладка и дальнейшее тестирование ПС будут выполняться профессиональными специалистами со стажем не менее 10 лет;
- интенсивность проявления ошибок  $\lambda(t)$  для времени  $t$  прямо пропорциональна оставшемуся числу ошибок в программном средстве в момент времени  $t$ ;
- время до следующей ошибки распределено экспоненциально;
- устранение ошибок на этапе тестирования осуществляется без внесения в программу других ошибок.

В качестве исходных данных рассматриваются:

- предполагаемый объём ПС в строках кода;
- время, отводимое на тестирование ПС;
- количество специалистов, привлекаемых к тестированию ПС.

Характеристика основных этапов алгоритма.

1. Определяется прогнозное значение начального числа ошибок  $N_{\text{нач}}$ , содержащихся в ПС после его написания на языке программирования и устранения ошибок трансляции (компиляции, работы интерпретатора и т.п.):

$$N_{\text{нач}} = \frac{131,3 \cdot S(1 - K_{\text{отл}})}{1000}, \quad (1)$$

где:  $S$  – объём ПС в строках кода;  $K_{\text{тр}}$  – коэффициент, показывающий долю ошибок в ПС, устраняемых при его отладке, по умолчанию для профессиональных программистов со стажем не менее 10 лет принимают  $K_{\text{отл}} = 0,5$  [5].

2. Рассчитывается ожидаемое число ошибок, оставшихся в ПС после проведения тестирования  $N_{\text{п.тест}}$ :

$$N_{\text{п.тест}} = (1 - K_{\text{тест}})N_{\text{нач}}, \quad (2)$$

где:  $K_{\text{тест}}$  – коэффициент, показывающий долю ошибок, выявляемых при тестировании; для профессиональных специалистов со стажем не менее 10 лет может быть принято  $K_{\text{тест}} \approx 0,5$  [6]. Для специалистов со стажем менее 10 лет коэффициент  $K_{\text{тест}}$  подлежит уточнению, но в любом случае  $K_{\text{тест}} < 0,5$ .

3. Определяется общее время тестирования  $t_{\text{тест}}$ :

$$t_{\text{тест}} = n_{\text{дн}} m_{\text{т}} t_{\text{дн}} K_{\text{ПС}}, \quad (3)$$

где:  $n_{\text{дн}}$  – число рабочих дней, отводимое для тестирования ПС;  $m_{\text{т}}$  – количество задействованных тестировщиков;  $t_{\text{дн}}$  – продолжительность рабочего дня тестировщика;  $K_{\text{ПС}}$  – коэффициент, показывающий, какая часть времени в день в среднем используется тестировщиком для прогона (исполнения на компьютере) ПС; по умолчанию можно принять  $K_{\text{ПС}} = 0,15$  (по данным

объединённого института проблем информатики НАН Беларуси).

4. Вся продолжительность процедуры тестирования ПС рассматривается как один этап тестирования ( $i = 1$ ). Для интенсивности проявления ошибок на  $i$ -м этапе  $\lambda_i$ , согласно модели Шумана [6, 7], справедливо выражение

$$\lambda_i = (N_{\text{нач}} - n_{i-1})C, \quad 1/\text{ч}, \quad (4)$$

где:  $n_{i-1}$  – число ошибок, исправленных к началу  $i$ -го этапа тестирования; поскольку  $i = 1$ , следовательно,  $n_{i-1} = n_0 = 0$ ;  $C$  – коэффициент пропорциональности, рассчитывается в данном случае как

$$C = \frac{n_{\text{тест}}}{N_{\text{нач}} - n_0} = \left. \begin{array}{l} C \text{ учётом того, что} \\ n_{\text{тест}} = N_{\text{нач}} - N_{\text{п.тест}} \end{array} \right| = \frac{N_{\text{нач}} - N_{\text{п.тест}}}{N_{\text{нач}} \cdot t_{\text{тест}}}, \quad 1/\text{ч}, \quad (5)$$

где:  $t_{\text{тест}}$  – планируемое время тестирования ПС, представляющее собой суммарное время прогона ПС (выполнения на ЭВМ) на этапе тестирования;  $n_{\text{тест}}$  – прогнозное число обнаруживаемых и исправляемых ошибок при тестировании ( $n_{\text{тест}} = N_{\text{нач}} - N_{\text{п.тест}}$ ).

5. Определяется прогнозное значение ожидаемой интенсивности проявления ошибок  $\lambda$  (интенсивности отказов ПС) после завершения процедуры тестирования и исправления выявленных ошибок, т.е. значение  $\lambda$ , соответствующее начальному этапу эксплуатации:

$$\lambda_{\text{эксп}} = (N_{\text{нач}} - n_{\text{тест}})C = N_{\text{п.тест}} \cdot C, \quad (6)$$

где:  $\lambda_{\text{эксп}}$  – интенсивность проявления ошибок (интенсивность отказов), соответствующая начальному этапу эксплуатации программного средства.

С учётом выражения (5) для коэффициента  $C$ , прогнозное значение  $\lambda_{\text{эксп}}$  может быть сразу рассчитано по формуле

$$\lambda_{\text{эксп}} = \frac{N_{\text{п.тест}}(N_{\text{нач}} - N_{\text{п.тест}})}{N_{\text{нач}} \cdot t_{\text{тест}}}, \quad 1/\text{ч}. \quad (7)$$

6. Определяется прогнозное значение средней наработки на проявление ошибки  $T_{0.\text{ПС}}$  (средней наработки на отказ) для начального этапа эксплуатации ПС в предположении экспоненциального распределения времени использования ПС до проявления ошибки:

$$T_{0.\text{ПС}} = \frac{1}{\lambda_{\text{эксп}}} = \frac{1}{N_{\text{п.тест}} \cdot C} = \frac{N_{\text{нач}} \cdot t_{\text{тест}}}{N_{\text{п.тест}}(N_{\text{нач}} - N_{\text{п.тест}})}, \quad \text{ч}. \quad (8)$$

Пример. Разрабатываемое прикладное ПС предположительно будет содержать 10 000 строк кода. Время, отводимое на тестирование программного средства четырьмя специалистами с опытом работы более 10 лет, определено в размере 15 недель при пятидневной рабочей неделе (75 рабочих дней). Среднее время работы программиста в день составляет 8 ч. Определим среднюю наработку на проявление ошибки  $T_{0.\text{ПС}}$  для начального этапа эксплуатации ПС.

Решение. 1. По формуле (1) определяем прогнозное значение числа ошибок  $N_{\text{нач}}$ , содержащихся в ПС после написания кода и устранения ошибок трансляции (отладки программы):

$$N_{\text{нач}} \approx 657 \text{ ошибок.}$$

2. Ожидаемое число ошибок, оставшихся в ПС после проведения тестирования  $N_{\text{п.тест}}$ , находим по выражению (2). Принимаем  $K_{\text{тест}} = 0,5$ . Получим

$$N_{\text{п.тест}} \approx 329 \text{ ошибок.}$$

3. По выражению (3) определяем прогнозное время тестирования ПС на ЭВМ:

$$t_{\text{тест}} = 75 \cdot 4 \cdot 8 \cdot 0,15 = 360 \text{ ч.}$$

4. По формуле (5) рассчитываем значение коэффициента  $C$  с учётом того, что  $n_{\text{тест}} = N_{\text{нач}} - N_{\text{п.тест}} = 328$  ошибок,  $n_0 = 0$ :

$$C = 328 / (657 \cdot 360) \approx 0,0014 \text{ 1/ч.}$$

5. По выражению (8) определяем среднюю наработку на проявление ошибки  $T_{0.\text{ПС}}$  для начального этапа эксплуатации ПС:

$$T_{0.\text{ПС}} \approx 2,0 \text{ ч.}$$

Следует отметить, что предложенный подход даёт весьма приближённые результаты ожидаемой надёжности разрабатываемого прикладного ПС. Однако даже такой ориентировочный расчёт полезен, так как позволяет получить представление о надёжности ПС на раннем этапе его разработки, до написания программного кода. Кроме того, используя описанный подход, можно ориентировочно определить требуемое время на тестирование, необходимое для обеспечения заданного уровня надёжности разрабатываемого прикладного ПС.

При оценке ожидаемой надёжности программного обеспечения, используемого для обработки больших данных (Big Data) с помощью информационно-компьютерной системы, следует принять во внимание и надёжность системного программного обеспечения с учётом также надёжности используемых драйверов. Модель безотказности этого программного обеспечения и алгоритм прогнозирования его ожидаемой надёжности будут отличаться от модели и алгоритма, предложенных для прикладных программных средств. Пользователи вычислительных систем, в том числе для обработки больших данных (Big Data), хотели бы знать, как оценить ожидаемую надёжность используемых в практике системных программных средств. Отметим, что решение этой задачи является предметом отдельного исследования.

#### Список литературы

- [1]. The physicist. Trending now. [Электронный ресурс]. – 2018. – Режим доступа: [http://www.wired.com/wired/archive/3.09/myhrvold.html?person=gordon\\_moore&topic\\_set=wiredpeople](http://www.wired.com/wired/archive/3.09/myhrvold.html?person=gordon_moore&topic_set=wiredpeople)
- [2]. Программное обеспечение – источник всех проблем. [Электронный ресурс]. – 2018. – Режим доступа: <http://www.williamspublishing.com/PDF/5-8459-0785-3/part1.pdf>
- [3]. Методы обеспечения аппаратно-программной надёжности вычислительных систем. Д.т.н., проф. Чуканов В.О., к.т.н., доц. Гуров В.В. [Электронный ресурс]. – 2018. – Режим доступа : [http://www.mcst.ru/files/5357ec/dd0cd8/50af39/000000/seminar\\_metody\\_obespecheniya\\_apparatno-programmnoy\\_nadezhnosti\\_vychislitelnyh\\_sistem.pdf](http://www.mcst.ru/files/5357ec/dd0cd8/50af39/000000/seminar_metody_obespecheniya_apparatno-programmnoy_nadezhnosti_vychislitelnyh_sistem.pdf)
- [4]. ГОСТ 28806-90. Качество программных средств. Термины и определения. – М. : Изд-во стандартов, 1990.
- [5]. ГОСТ 27.002-89. Надёжность в технике. Основные понятия, термины и определения. – М. :

Изд-во стандартов, 1990.

[6]. Чуканов, В. О. Надёжность программного обеспечения и аппаратных средств систем передачи данных атомных электростанций : учеб. пособие / В. О. Чуканов. – М. : МИФИ, 2008. – 168 с.

[7]. Shooman, M. L. Software engineering: Reliability, Development and Management / M. L. Shooman. – McGraw-Hill, International. Book Co, 1983.

## **FORECASTING EXPECTED RELIABILITY OF APPLIED SOFTWARE BY USING THEIR STATISTICAL MODELS OF FAILURE-FREE OPERATION**

***S.M. BOROVIKOV, PhD***

*Associate Professor, Department of Information and Computer Systems Design BSUIR*

***S.S. DICK***

*Postgraduate student BSUIR, Master of engineering*

*Belarusian State University of Informatics and Radioelectronics, Republic of Belarus  
E-mail: bsm@bsuir.by, sdick@bsuir.by*

**Abstract.** To predict the expected reliability of the developed application software, we propose an approach based on the use of a statistical model of software fail-safe. The model is based on generalized statistical data about the expected number of errors in the developed application software, taken from the domestic and foreign press, depending on its volume and the dynamics of reducing the number of errors from the time of testing and taking into account the professional experience of programmers and testers. An example of predicting the expected reliability of an application software is given.

**Key words:** application software, expected reliability, forecasting, statistical model failure-free operation.