

## Надежность программного обеспечения и методы ее повышения

*А.А. Наумов, А.Р. Айдинян*

*Донской государственной технической университет, Ростов-на-Дону*

**Аннотация:** Стабильная работа компьютерной техники важна для эффективного ее использования. Существенная роль в обеспечении стабильности зависит от надежности программного обеспечения. В статье сделан анализ состояния вопроса по повышению надежности программного обеспечения на основе известных математических моделей. Рассмотрены существующие способы увеличения надежности программного обеспечения, использующие временную и программную избыточность. Делается вывод об отсутствии универсального решения проблемы надежности программного обеспечения при наличии множества частных решений.

**Ключевые слова:** надежность программного обеспечения, модели оценки надежности программного обеспечения, классификация моделей надежности программного обеспечения, методы повышения надежности программного обеспечения.

Компьютеры играют важную роль в нашей повседневной жизни и важно, чтобы они работали без сбоев, безотказно. Как известно, компьютер состоит из аппаратного обеспечения и устанавливаемого на него программного обеспечения (ПО) [1]. Из них ключевым элементом, во многом определяющим надежность всей системы, является ПО. В отличие от аппаратуры программное обеспечение не ломается, не изнашивается и не ржавеет, но оно может содержать в себе ошибки, что приводит к сбоям в работе, потере информации и т.д. [2-7].

Ненадежность ПО возникает вследствие ошибок проектирования и дальнейшего кодирования программы, т.е. ошибок, внесенных в процессе разработки.

Для программного обеспечения частота появления ошибок является наиболее высокой на начальном этапе испытаний и тестирования. В процессе тестирования выявляются и удаляются ошибки. Нахождение ошибок продолжается и в дальнейшем – на этапе сопровождения программы. Но чем позже обнаружена ошибка, тем больше будет стоить ее корректировка [8-11]. Но даже пройденное тестирование не дает гарантии, что ПО не содержит в себе ошибок и в дальнейшем будет работать без сбоев.

---

Естественно, чем интенсивнее пользуются программой, чем больше у нее пользователей, тем быстрее обнаруживаются дефекты в программе. На рис. 1 показана зависимость числа обнаруженных ошибок с течением времени  $t$  от числа пользователей  $K$  ( $K_1 > K_2 > K_3$ ) [9].

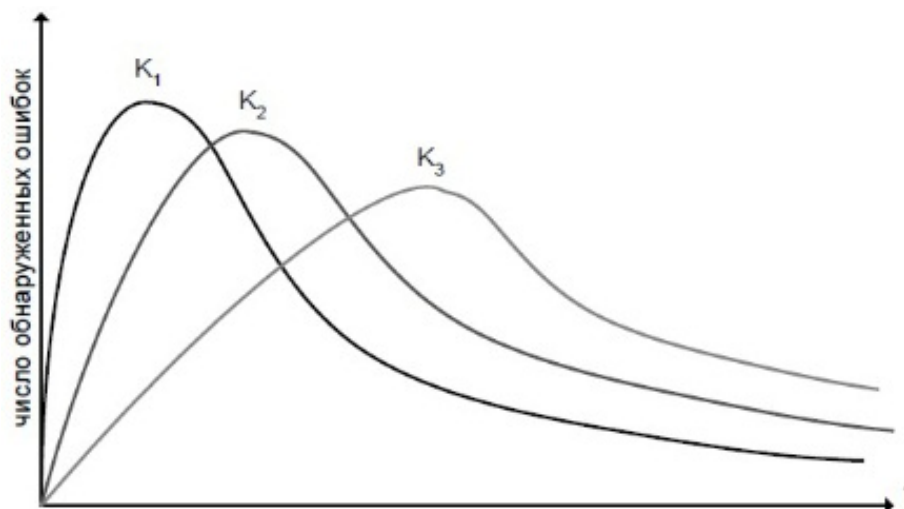


Рис. 1 Интенсивность обнаружения ошибок от числа пользователей

В нижеприведенной таблице показана частота появления различных типов ошибок [9]. Как видно, большинство ошибок возникает из-за неверной спецификации или технического задания.

Таблица

Процентные частоты появления ошибок в ПО

Тип ошибки	Частота появления, %
Не полная или ошибочная спецификация	28
Отклонение от спецификации	12
Ошибочная логика или последовательность операций	12
Пренебрежение правилами программирования	10
Ошибочная выборка данных	10
Ошибочные арифметические операции	9
Неточная запись кода	8
Нехватка времени для решения	4
Ошибка обработки прерываний	4
Ошибка в исходных данных	3

При необходимости дать модели надежности ПО количественную оценку применяют специально разработанные математические модели для оценки взаимосвязи надежности программы от известных наперед или выявленных в процессе исполнения задачи показателей.

В монографии [2] авторы классифицировали существующие модели надежности ПО (рис. 2).

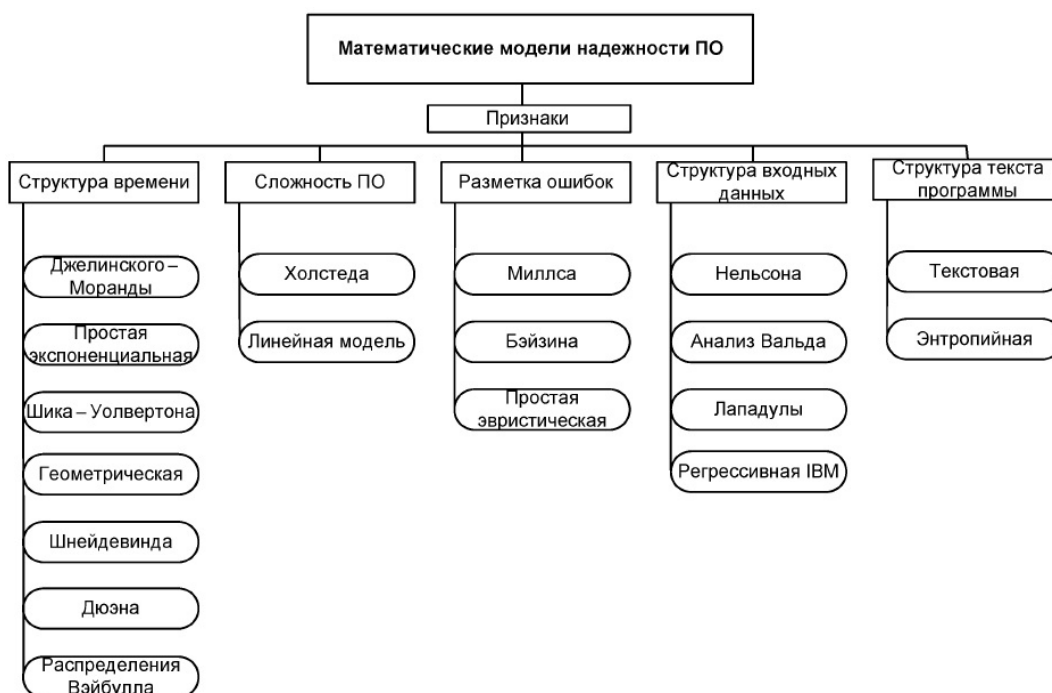


Рис. 2 Математические модели надежности ПО

Представленные модели объединены по следующим ключевым особенностям:

- по временной структуре процесса выявления ошибок в ПО (время появления ошибки, их количество за определенный промежуток времени);
- по сложности программы (критерием сложности могут быть длина, количество функций или модулей, входных данных и т. п.);
- по разметке ошибок (намеренное добавление в программу ошибок);
- по структуре пространства входных данных;
- по структуре кода программы (по тому, как распределены ошибки в программе).

Сравнительный анализ показал, что использование моделей Джелинского-Моранды и Шика-Уолвертона обосновано при определении надежности ПО маленького размера, а улучшенная модель Шика-Уолвертона подходит для огромных программ. Если разработчику нужно получить показатель надежности (в частности, время наработки программы до отказа), то лучше применять геометрическую модель. Определенные модели не выдают ожидаемых от них ответов при некоторых входных данных. Если известны интервалы времени между найденными ошибками, то предпочтительно пользоваться геометрической моделью, при установлении количества ошибок на единицу времени модель Шнейдевинда покажет лучший результат. В работе [7] на реальных задачах показано, что экспоненциальная и дискретная модели являются эффективными для оценки надежности ПО.

Можно констатировать, что в настоящее время нельзя выявить какую-то одну лучшую модель из большого количества имеющихся и вследствие этого рекомендуется пользоваться несколькими моделями одновременно и совмещать получаемые результаты.

Для повышения стабильности работы программ можно использовать следующие методы [2-4]:

- методы, которые используют временную избыточность: авторизация доступа операторов к программе, проверка ресурсов, которые доступны оператору, выделение ресурсов пропорционально уровню подготовки операторов, разграничение прав доступа операторов к определенным программам, модулям управления, таблицам баз данных.

- методы, применяющие информационную избыточность: проверка корректности ссылок баз данных обеспечивается системой внутренних уникальных ключей для всех информационных записей системы, открытая система кодирования, которая позволяет оператору в любое время

---

оптимизировать код программы, инструменты проверки значений контрольных сумм записей системы (позволяют выявить все незаконные изменения в программе).

Использование указанных методов обеспечит повышение надежности ПО с помощью следующих мероприятий [2-4]:

- усовершенствование технологии программирования (например, формальное описание этапов программирования);
- применение алгоритмов, которые не чувствительны к сбоям вычислительного процесса (алгоритмическая избыточность);
- резервирование программ – N-версионное программирование (использование программной избыточности – модули имеют несколько версий с реализацией различных алгоритмов);
- верификация и валидация программ с последующей коррекцией.

Сложность задач, решаемых с помощью компьютерных технологий, постоянно растет. Без компьютеров современный человек уже не представляет свою жизнь, они используются во всех областях жизнедеятельности – от моделирования космических полетов до поддержки сайта госуслуг для пенсионеров. И при всем этом известные способы оценки надежности ПО еще не могут считаться универсальными. Для получения корректной оценки надежности ПО нужно применять несколько методов, что существенно увеличивает финансовые и временные затраты разработчиков. К тому же при создании новых программ применяются самые передовые технологии. В связи с этим весьма важным является разработка универсальных методов оценки надежности ПО, нацеленных на создание программ, применяющих последние достижения в программировании.

### **Литература**

1. Айдинян А.Р., Цветкова О.Л. Информационные технологии: учебное пособие. Ростов н/Д: Издательский центр ДГТУ, 2011. 132 с.



2. Полонников Р.И., Никандров А.В. Методы оценки надежности программного обеспечения. СПб: Политехника, 1992. 80 с.
3. O'Connor J., McDermott I. The Art of Systems Thinking: Essential Skills for Creativity and Problem Solving. Thorsons, 1997. 288 p.
4. Майерс Г. Надёжность программного обеспечения. М.: Мир, 1980. 360 с.
5. Trachtenberg M. Discovering How to Ensure Software Reliability // RCA Engineer. Jan/Feb. 1982. Pp. 53-57.
6. Москат Н.А., Станкевич Е.А. Показатели качества информационно-вычислительных систем железнодорожного транспорта // Инженерный вестник Дона, 2013, №3. URL: [ivdon.ru/ru/magazine/archive/n3y2013/1789](http://ivdon.ru/ru/magazine/archive/n3y2013/1789)
7. Штрик А.А., Осовецкий Л.Г., Мессих И.Г. Структурное проектирование надежных программ встроенных ЭВМ. Л.: Машиностроение, 1989. - 296 с.
8. Любицын В.Н. Необходимость разработки надежного программного обеспечения как вызов современности // Вестник ЮУрГУ, 2012, № 23. С. 26-29.
9. Липаев В.В. Надёжность программных средств. М.: СИНТЕГ, 1998. 232 с.
10. Ивутин А.Н., Суслин А.А. О некоторых применениях статистических распределений в оценке надежности программного обеспечения // Известия ТулГУ. Технические науки. 2011. №2. С. 568-575.
11. Айдинян А.Р., Цветкова О.Л. Подход к оценке DLP-систем с использованием средств нечеткой логики // Инженерный вестник Дона, 2017, №4. URL: [ivdon.ru/ru/magazine/archive/n4y2017/4432](http://ivdon.ru/ru/magazine/archive/n4y2017/4432).

### References

1. Ajdinyan A.R., Cvetkova O.L. Informacionnyye tekhnologii [Information technology]: uchebnoe posobie. Rostov n/D: Izdatel'skij centr DGTU, 2011. 132 p.
-



2. Polonnikov R.I., Nikandrov A.V. Metody ocenki nadezhnosti programmnoho obespecheniya [Methods for assessing the reliability of the software]. SPb: Politekhnik, 1992. 80 p.
3. O'Connor J., McDermott I. The Art of Systems Thinking: Essential Skills for Creativity and Problem Solving. Thorsons, 1997. 288 p.
4. Mayers G. Nadezhnost' programmnoho obespecheniya [Software reliability]. M.: Mir, 1980. 360 p.
5. Trachtenberg M. RCA Engineer. Jan/Feb. 1982. pp. 53-57.
6. Moskat N.A., Stankevich E.A. Inzhenernyj vestnik Dona (Rus), 2013, № 3. URL: [ivdon.ru/ru/magazine/archive/n3y2013/1789](http://ivdon.ru/ru/magazine/archive/n3y2013/1789).
7. SHtrik A.A., Osoveckij L.G., Messih I.G. Strukturnoe proektirovanie nadezhnykh programm vstroennykh EHVM [Structural design of reliable embedded computer programs]. L.: Mashinostroenie, 1989. 296 p.
8. Lyubicyn V.N. Vestnik YUUrGU. 2012. № 23. pp. 26-29.
9. Lipaev V.V. Nadezhnost' programmnykh sredstv [Software reliability]. M.: SINTEG, 1998. 232 p.
10. Ivutin A.N., Suslin A.A. Izvestiya TulGU. Tekhnicheskie nauki. 2011. №2. pp. 568-575.
11. Ajdinyan A. R., Tsvetkova O.L. Inzhenernyj vestnik Dona (Rus), 2017, № 4. URL: [ivdon.ru/ru/magazine/archive/n4y2017/4432](http://ivdon.ru/ru/magazine/archive/n4y2017/4432).