

## НЕОБХОДИМОСТЬ РАЗРАБОТКИ НАДЕЖНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ КАК ВЫЗОВ СОВРЕМЕННОСТИ

*В.Н. Любицын*

## THE NECESSITY OF RELIABLE SOFTWARE DEVELOPMENT AS A CHALLENGE OF MODERN AGE

*V.N. Lyubitsyn*

Проводится анализ современного состояния в проблематике повышения надежности программного обеспечения на основе моделей оценки его надежности различного типа, которые классифицированы по четырем признакам. Делаются выводы об отсутствии общего решения проблемы надежности программного обеспечения при наличии множества частных решений, не учитывающих такие существенные факторы, как интенсивность внесения и устранения ошибок в программе, время разработки, а также о том, что лучший результат может дать модель надежности, не рассматривающая конкретный язык программирования или объем и сложность текста программ.

*Ключевые слова: надежность программного обеспечения, прогнозирование надежности программного обеспечения, классификация моделей надежности программного обеспечения.*

The analysis of the current state of the problems of software reliability increase on the basis of reliability assessment models of various types, which are classified in accordance with four characteristics, is carried out. Conclusions of the lack of general solution to the problem of software reliability in the presence of a number of particular solutions, in spite of such important factors as the intensity of errors control and introduction to the program, time of development, and of the fact that better results can be achieved by the use of reliability model, without considering the specific programming language or the volume and complexity of the program code, are made in the article.

*Keywords: software reliability, software reliability prediction, classification of software reliability models.*

### **Введение**

В наше время стали создаваться сверхсложные системы в информатике, энергетике, транспорте и в других отраслях народного хозяйства. Причем это не просто системы, которые характеризуются большим числом входящих в их состав элементов, сложными структурами и алгоритмами функционирования. Это системы, пронизывающие всю инфраструктуру современного общества на государственном уровне, а это приводит не только к чисто структурному и функциональному их усложнению, но и резкому повышению требований к надежности, живучести и безопасности функционирования. Современный период развития техники характеризуется уже не только лозунгами о важности проблемы надежности, но и бурным развитием методов обеспечения высокой надежности систем на всех этапах: при проектировании, производстве, испытаниях и эксплуатации. При этом

ключевым элементом подобных систем, во многом определяющим их надежность, является программное обеспечение (ПО). Сегодня можно без преувеличения утверждать, что мировая экономика становится все более зависимой от надежности ПО.

Действительно, проектирование и реализацию сложных программно-технических систем, на создание которых в течение многих лет затрачивались огромные людские и материальные ресурсы, уже невозможно осуществлять «на глазок». Требуется строгий математический расчет всех технических параметров, включая различные показатели надежности, нужны обоснованные технико-экономические решения. При этом, учитывая огромную ответственность задач, решаемых программно-техническими сверхсистемами на уровне национальной экономики, национальной безопасности, а также порою непредвидимые экономические и

---

**Любицын Владимир Николаевич** – канд. техн. наук, доцент кафедры информационно-аналитического обеспечения управления в социальных и экономических системах, Южно-Уральский государственный университет; lvn\_iaou@mail.ru

---

**Lyubitsyn Vladimir Nickolaevich** – Candidate of Science (Engineering), Associate Professor of Information and Analytical Support in Social and Economic Systems Management Department, South Ural State University; lvn\_iaou@mail.ru

морально-политические последствия от возможных ошибок и отказов в этих системах, необходимо не только обеспечить технические возможности этих систем вообще, но и, что самое главное, сохранить и поддержать работоспособность подобных систем в течение очень длительного времени эксплуатации [1]. Можно привести примеры многих современных программно-технических систем, для которых решение проблемы надежности в самом прямом смысле означает быть или не быть данной системе. К ним можно отнести различные информационные системы, в том числе государственные, региональные, отраслевые и корпоративные автоматизированные системы управления; системы управления воздушным движением для гражданской авиации; автоматизированные системы управления технологическими процессами; сеть центров управления и слежения за космическими объектами; сети и системы передачи данных и т. д.

Усложнение систем идет в различных направлениях. С одной стороны, в состав в их входит все большее число комплектующих элементов. С другой стороны, усложняется структура систем, определяющая соединение отдельных элементов и их взаимодействие в процессе функционирования и поддержания работоспособности. Понятно, что усложнение систем является прямым следствием постоянно возрастающей ответственности выполняемых ими функций, сложности и многообразия этих функций, что, в свою очередь, диктуется прогрессом науки и техники. Любая система, состоящая из большого числа комплектующих элементов и имеющая сложную структуру, сложный алгоритм функционирования и, следовательно, сложный программный комплекс, требует разработки специальных методов обеспечения, повышения и поддержания надежности таких систем, включая разработку математических методов априорных расчетов и экспериментальной оценки.

Инженеры и математики приложили немало совместных усилий для разработки современной теории надежности. Были предприняты гигантские усилия для создания более надежных компонентов, более простых и надежных схем и конструкций, улучшения условий эксплуатации. Были разработаны соответствующие методы, позволяющие осуществлять анализ и синтез разрабатываемых программно-технических систем на этапе проектирования, проводить обоснованные оценки показателей надежности этих средств во время испытаний и эксплуатации. Однако проблема надежности этих систем и, особенно, их ПО продолжает оставаться одной из основных. Дело, видимо, объясняется не столько тем, что достигнутая надежность современных технических систем слишком низка, сколько тем, что непрерывно усложняются решаемые задачи и одновременно повышаются требования к надежности их выполнения. В любом случае сегодня прогнозирование надежности ПО в

процессе его эксплуатации осуществляется на основе математических моделей надежности программ, реализуемых на основе довольно разных подходов.

Так, в работе [2] приведены вероятностные модели надежности. Так как теория надежности аппаратуры развита довольно хорошо, естественно попытаться применить ее и к надежности ПО. В этих моделях ищется число ошибок, оставшихся в программе (хотя, на наш взгляд, лучше искать вероятность возникновения ошибки за время  $t$ , так как и при небольшом количестве ошибок, но при заданном режиме работы ПО может выходить из строя очень часто, и, наоборот, при большом количестве ошибок – редко «зависать»). Это полезно знать для завершения процесса тестирования и можно оценить стоимость сопровождения, которая пропорциональна количеству оставшихся в программе ошибок. Эти модели позволяют находить в том числе и надежность программы, которая понимается как вероятность, что программа будет функционировать без ошибок в течение заданного интервала времени, а также среднее время между отказами программы.

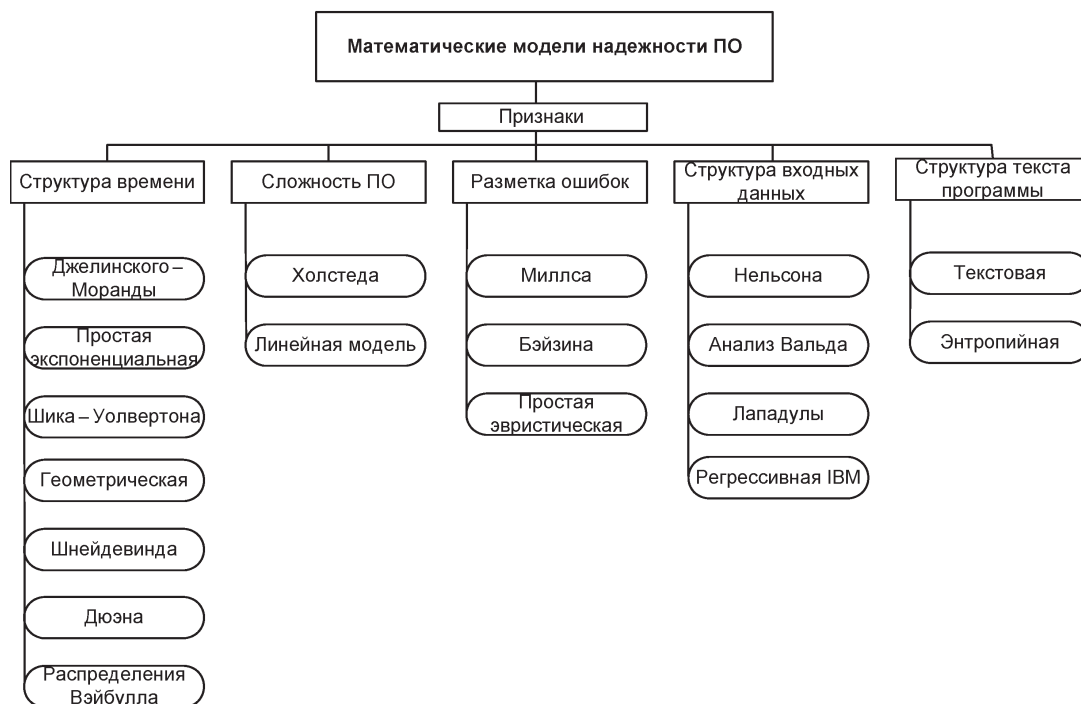
В [3] дается классификация моделей надежности ПО. Наиболее известных моделей надежности ПО в настоящее время существует около двух десятков (см. рисунок), поэтому в данной работе они сгруппированы по ряду классификационных признаков, в качестве которых выбраны следующие:

- временная структура процессов проявления ошибок в ПО (время появления ошибки, количество ошибок за заданный интервал времени);
- сложность программы (мера сложности ПО – длина, количество функций или модулей, данных и т. п.);
- разметка ошибок (искусственное внесение в ПО известных ошибок);
- структура пространства входных данных;
- структура текста программы (распределение ошибок по тексту программы).

Как показано в [4], на практике простейшие, элементарные ошибки программ и данных могут приводить к катастрофическим последствиям при функционировании ПО. В то же время крупные системные дефекты могут только несколько ухудшать эксплуатационные характеристики ПО. Поэтому невозможно ранжировать типы первичных ошибок по степени влияния на надежность и следует одинаково тщательно относиться к их обнаружению и устранению.

Статистика ошибок в комплексных программах и их характеристики могут служить ориентиром для разработчиков при распределении усилий на отладку и предохранять их от излишнего оптимизма при оценке достигнутого качества и надежности ПО. Они помогают:

- оценивать реальное состояние проекта и планировать необходимые трудоемкость и длительность до его завершения;



#### Математические модели надежности ПО

– выбирать методы и средства проектирования, программирования и тестирования.

Регистрация, сбор и анализ характеристик ошибок в программах – сложный и трудоемкий процесс. Поэтому имеется относительно небольшое число работ, в которых опубликованы реальные характеристики ошибок. Однако можно обоснованно констатировать, что при автономной и в начале комплексной отладки доля системных ошибок невелика (~ 10 %), но она существенно возрастает (до 20–25 %) на завершающих этапах комплексной отладки. В процессе сопровождения системные ошибки являются преобладающими (до 80 % от всех ошибок). При этом ни одна из этих моделей надежности ПО не имеет явных преимуществ по точности аппроксимации распределений и прогнозирования числа ошибок в программах по сравнению с простейшей экспоненциальной моделью. Более подробные сведения о некоторых моделях надежности ПО приведены в [3–6].

В работе [3] дается сравнение моделей. Модели Джелинского – Моранды и Шика – Уолвертона целесообразны при моделировании надежности ПО небольшого объема, а модифицированная модель Шика – Уолвертона – для ПО больших проектов. Если при моделировании необходимо получить значения надежности (например, среднюю наработку до отказа), то лучше использовать геометрические модели. Некоторые модели не имеют решений (то есть расходятся при определенных входных условиях). Если имеются данные об интервалах времени между ошибками, то лучше воспользоваться геометрической моделью, а если имеются данные о числе ошибок, приходящихся

на единицу времени, то лучше применять модель Шнейдевинда. Экспоненциальная и дискретная модели были проверены при тестировании реальных программ и хорошо соответствуют действительности [5]. При этом в [3] делается вывод, что на сегодняшний день невозможно выбрать наилучшую модель среди десятка предложенных.

Характерно, что большинство моделей надежности ПО сложны в применении, так как требуют знаний таких характеристик ПО, которые можно вычислить только после длительной его эксплуатации, но в этом случае характеристики надежности ПО уже известны и так! Кроме того, из-за значительных неопределенностей во всех указанных моделях в [2] рекомендуется использовать несколько моделей одновременно и объединить их результаты.

В [4] говорится, что модели дают удовлетворительный результат при относительно высоких уровнях интенсивности проявления ошибок, то есть при невысокой надежности ПО. В этих условиях математические модели предназначены для приближенной оценки:

- потенциально возможной надежности функционирования программ в процессе испытаний и эксплуатации;
- числа обнаруженных ошибок;
- времени тестирования, требуемого для обнаружения следующей ошибки;
- времени, необходимого для обнаружения с заданной вероятностью большинства имеющихся ошибок.

Модели, основанные на сложности исходных текстов программ, в данной статье не рассматри-

ваются, поскольку, на наш взгляд, нет прямой связи между количеством операторов, количеством циклов, ветвлений, переходов и количеством ошибок в программе, так как основные ошибки закладываются на этапах определения требований к системе и проектирования, а вовсе не кодирования. Это особенно ясно в последнее время, когда почти все среды разработки приближаются к языкам 4-го поколения, автоматически генерируют исходный текст и делают это, конечно, без ошибок. При этом современные компиляторы четко отслеживают все ошибки программиста (такого рода, как неинициализированная переменная, недоступный код и т. п.). А также весьма трудоемко считать количество входов и ветвей в программе. Бесплезно это и для тестирования программ, так как показано, что протестировать все ветви практически невозможно.

Можно согласиться с мнением [4, 5] о том, что оценки, приведенные для нескольких конкретных систем, позволят прогнозировать эти характеристики для других проектов, а гипотеза статистической независимости временных интервалов между ошибками достаточно хорошо подтверждается экспериментальными данными (хотя, говоря строго, это верно только для хорошо отлаженной программы, а для «сырой» – одна обнаруженная ошибка может вскрыть целый пласт ошибок).

Наверное, можно описывать надежность ПО логарифмически-нормальным распределением, так как в этом случае интенсивность отказов  $\lambda(t)$  сначала возрастает, а затем убывает до нуля. То есть логарифмически-нормальное распределение времени безотказной работы применимо к объектам, имеющим свойство «упрочняться» с течением времени эксплуатации. Именно к таким объектам относится и ПО. При этом распределении логарифм времени безотказной работы (или времени восстановления) подчиняется нормальному закону [6].

Наконец, следует заметить, что любая математическая модель надежности ПО дает плохой результат при низкой интенсивности отказов, то есть обычно, когда программа отлажена и передается в эксплуатацию. Попытки аналитически рас-

считать вероятности проявления ошибок оказались безуспешными вследствие низкой достоверности результатов. Это объясняется невозможностью определения средней интенсивности отказов элементов ПО – операций или модулей. Они зависят от индивидуальных характеристик разработчика и сложности ПО.

#### **Заключение**

На основе сделанного обзора можно констатировать: отсутствует общее решение проблемы надежности ПО и есть много частных решений, не учитывающих такие существенные факторы, как интенсивность внесения и устранения ошибок в программе, время разработки ПО. При этом объем и сложность самого текста программ в настоящее время играют все меньшую роль при определении надежности ПО. Ни одна из моделей не может считаться достаточной для оценки надежности. Вероятно, что лучший результат может дать модель надежности ПО, не рассматривающая конкретный язык программирования или объем и сложность текста программ. Как бы там ни было, задача построения более совершенных моделей надежности ПО продолжает оставаться актуальной.

#### **Литература**

1. *Соммервилл, И. Инженерия программного обеспечения / И. Соммервилл. – М.: Вильямс, 2002. – 624 с.*
2. *Майерс, Г. Надежность программного обеспечения / Г. Майерс. – М.: Мир, 1980. – 360 с.*
3. *Полонников, Р.И. Методы оценки надежности программного обеспечения / Р.И. Полонников, А.В. Никандров. – СПб: Политехника, 1992. – 80 с.*
4. *Липаев, В.В. Надежность программных средств / В.В. Липаев. – М.: СИНТЕГ, 1998. – 232 с.*
5. *Штрик, А.А. Структурное проектирование надежных программ встроенных ЭВМ / А.А. Штрик, Л.Г. Осовецкий, И.Г. Мессих. – Л.: Машиностроение, 1989. – 296 с.*
6. *Пальчун, Б.П. Оценка надежности программного обеспечения / Б.П. Пальчун, Р.М. Юсупов. – СПб.: Наука, 1994. – 84 с.*

*Поступила в редакцию 13 апреля 2012 г.*