УДК 519.95:539.376

В. П. ПОШИВАЛОВ, Ю. Ф. ДАНИЕВ

О МОДЕЛЯХ НАДЕЖНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ЭРГАТИЧЕСКИХ СИСТЕМ

Институт технической механики
Национальной академии наук Украины и Государственного космического агентства Украины,
ул. Лешко-Попеля, 15, 49005, Днепр, Украина; e-mail: Poshivalov.V.P@nas.gov.ua

Мета роботи – аналіз моделей і розробка підходів до забезпечення надійності програмної складової ергатичних систем. Проаналізовано моделі надійності програмного забезпечення і відзначено особливості його відмов. Розглянуто основні підходи до забезпечення надійності програмного забезпечення та відзначено, що вони повною мірою не дають можливості практичного застосування для оцінки надійності програмного забезпечення. Виявлено фактори, що впливають на кількість помилок програмного забезпечення, і фактори, що сприяють підвищенню його надійності. Запропоновано шляхи, що дозволяють підвищити надійність програмного забезпечення. Показано, що у якосі основних показників надійності програмного забезпечення ергатичної системи необхідно використовувати функцію надійності й середній час між відмовами.

Цель работы – анализ моделей и разработка подходов к обеспечению надежности программной составляющей эргатических систем. Проанализированы модели надежности программного обеспечения и отмечены особенности его отказов. Рассмотрены основные подходы к обеспечению надёжности программного обеспечения и отмечено, что они в полной мере не дают возможности практического применения для оценки надежности программного обеспечения. Выявлены факторы, влияющие на количество ошибок программного обеспечения, и факторы, способствующие повышению его надежности. Предложены пути, позволяющие повысить надежность программного обеспечения. Показано, что в качестве основных показателей надежности программного обеспечения эргатической системы необходимо использовать функцию надежности и среднее время между отказами.

The aim of this work is to analyze models of the reliability of the ergative system software component and to develop approaches to the assurance thereof. Software reliability models are analyzed, and the features of software failures are pointed out. Basic approaches to software reliability assurance are considered, and it is pointed out that in actual practice they do not allow one to assess software reliability in full measure. Factors that have an effect on the number of software errors and factors that enhance software reliability are identified. Ways to enhance software reliability are suggested. It is shown that the reliability function and the average time between failures must be used as the basic software reliability indices.

Ключевые слова: *эргатическая система, надежность, модели надежности, программное обеспечение.*

Эргатическая система (ЭС) представляет собой систему, в состав которой входят технические устройства, программное обеспечение (ПО) и человек, выполняющий операторские функции.

Особенность таких систем состоит в потенциальной опасности нарушения их функциональной стабильности, поскольку полный или частичный отказ системы может привести к значительным экономическим, экологическим или другим убыткам. Поэтому обеспечение функциональной стабильности эргатических систем является актуальной проблемой. В связи с этим в состав требований на разработку таких систем включаются требования по надежности [1, 2].

Сама проблема надежности ПО включает два аспекта: обеспечение и оценка (измерение) надежности [3 – 6]. Надежность программы гораздо важнее таких традиционных ее характеристик, как время исполнения или требуемый объем оперативной памяти, однако никакой общепринятой количественной меры надежности программ не существует.

К основным проблемам надёжности ПО можно отнести:

– разработку методов оценки и прогнозирования надёжности ПО;

– определение факторов, влияющих на достижение заданного уровня надёжности ПО;

– совершенствование методов повышения надёжности ПО в процессе проектирования и эксплуатации.

Целью работы является анализ моделей и разработка подходов к обеспечению надежности программной составляющей эргатических систем.

Существует достаточно много разных моделей, которые дают возможность прогнозировать надежность ПО на разных этапах его жизненного цикла [7 – 9].

При построении моделей надёжности ПО можно выделить два подхода. Первый подход – программы рассматриваются как аналоги невосстанавливаемых систем и находятся значения соответствующих показателей надежности (вероятность безотказной работы программ на протяжении определенного интервала наработки или числа прогонов при эксплуатации программ в расчетном режиме). Второй подход – программы рассматриваются как аналоги восстанавливаемых систем.

Анализ отказов эргатических систем по вине ПО позволяет выделить следующее:

– отказ техники обусловлен разрушением элементов, а программный отказ происходит при выходе программы на участок, содержащий ошибку;

– безотказность ПО целиком зависит от наличия в нём ошибок, внесенных на этапе его создания, а безотказность технических средств определяется в основном отказами, зависящими от изменения их параметров во время эксплуатации;

– после устранения отказа техники не исключается повторение такого же отказа при дальнейшей работе, а обнаружение и устранение программного отказа означает, что такой отказ в дальнейшем не повторится;

– прогнозировать отказ техники можно статистическими методами, а предвидеть появление программных отказов (выход на участок, содержащий ошибку), во многих случаях не представляется возможным;

– вероятность возникновения программного отказа не зависит от времени работы системы;

– надёжность работы ПО зависит от правильности и формирования используемой входной информации.

Несмотря на то, что отказы ПО имеют совершенно другую физическую природу, чем отказы техники, это не является причиной невозможности использования некоторых терминов и показателей надежности техники при исследовании качества ПО.

Модели надежности программных средств подразделяются на аналитические и эмпирические.

Аналитические модели дают возможность рассчитать количественные показатели надежности, основываясь на данных о поведении программы в процессе тестирования. Они представлены двумя группами: динамические и статические.

В динамических моделях поведение ПО (появление отказов) рассматривается во времени. Если фиксируются интервалы каждого отказа, то получается непрерывная картина появления отказов во времени (модели с непрерывным временем), а когда поведение ПО анализируется только в дискретных точках имеем модели с дискретным временем. К таким аналитическим динамическим моделям относятся: модель Шумана; модель Ла Падула; модель Джелинского–Мораны; модель Шика–Волвертона; модель Муссы; модель переходных вероятностей [4, 7].

Для модели Шумана исходные данные собираются в процессе тестирования ПО в течение фиксированных или случайных временных интервалов. Использование модели Шумана предполагает, что тестирование проводится в несколько этапов. Каждый этап представляет собой выполнение программы на полном комплексе разработанных тестовых данных. Выявленные ошибки регистрируются, но не исправляются. По завершении этапа на основе собранных данных модель Шумана может быть использована для расчета количественных показателей надежности. После этого исправляются ошибки, обнаруженные на предыдущем этапе, при необходимости корректируются тестовые наборы и проводится новый этап тестирования. При использовании модели Шумана предполагается, что исходное количество ошибок в программе постоянно и в процессе тестирования может уменьшаться по мере того, как ошибки выявляются и исправляются (новые ошибки не вносятся). Базовые понятия модели заимствованы из теории надежности аппаратных средств.

В модели Ла Падула выполнение последовательности тестов производится в m этапов. Каждый этап заканчивается внесением изменений (исправлений) в ПО. Возрастающая функция надежности базируется на числе ошибок, обнаруженных в ходе каждого тестового прогона. Основным показателем надежности является функция надежности.

Рассмотрим аналитическую динамическую модель Джелинского–Моранды. Основное положение, на котором базируется модель, заключается в том, что в процессе тестирования ПО значение интервалов времени тестирования между обнаружением двух ошибок имеет экспоненциальное распределение с интенсивностью отказов, пропорциональной числу еще не выявленных ошибок. Каждая обнаруженная ошибка устраняется, число оставшихся ошибок уменьшается на единицу.

Функция плотности распределения времени обнаружения i-й ошибки, отсчитываемого от момента выявления $(i-1)$-й ошибки, имеет вид

$$g(t_i) = \lambda_i \exp(-\lambda_i t_i), \qquad (1)$$

где λ_i – интенсивность отказов, которая пропорциональна числу еще не выявленных ошибок в программе:

$$\lambda_i = C(N - i + 1), \qquad (2)$$

где N – число ошибок, первоначально присутствующих в программе; C – коэффициент пропорциональности.

Наиболее вероятные значения величин N и C определяются на основе данных, полученных при тестировании. Для этого фиксируют время выполнения программы до очередного отказа $\Delta t_1, \Delta t_2, \Delta t_3, ..., \Delta t_K$, где K – число экспериментально полученных интервалов между отказами. На основе метода максимального правдоподобия значение N можно получить как решение нелинейного уравнения

$$K\frac{\sum_{i=1}^{K}\Delta t_i}{\sum_{i=1}^{K}\frac{1}{N+1-i}}=\sum_{i=1}^{K}(N+1-i)\Delta t_i\ , \qquad (3)$$

а коэффициент пропорциональности C определяется как

$$C=\frac{\sum_{i=1}^{K}\left(\frac{1}{N+1+i}\right)}{\sum_{i=1}^{K}\Delta t_i}.$$

Учитывая (2) и (3) можно получить числовые значения a_i, и определить вероятность безотказной работы на различных временных интервалах.

Модель Шика–Волвертона является модификацией модели Джелинского–Моранды для случая возникновения на рассматриваемом интервале более одной ошибки. При этом считается, что исправление ошибок производится лишь после истечения интервала времени, на котором они возникли. В основе модели Шика–Волвертона лежит предположение, согласно которому частота ошибок пропорциональна не только количеству ошибок в программах, но и времени тестирования, т. е. вероятность обнаружения ошибок с течением времени возрастает. Интенсивность обнаружения ошибок предполагается постоянной в течение интервала времени и пропорциональна числу ошибок, оставшихся в программе по истечении интервала. Она пропорциональна также и суммарному времени, уже затраченному на тестирование. Расчеты надежности ПО аналогичны расчетам модели Джелинского–Моранды.

В модели переходных вероятностей процесс тестирования ПО рассматривается как марковский процесс.

Статические модели учитывают зависимость количества ошибок либо от числа тестовых прогонов (модели по области ошибок), либо от характеристики входных данных (модели по области данных), т. е. появление отказов не связывают со временем. К аналитическим статистическим моделям относятся: простая интуитивная модель, модель Коркорэна, модель последовательных испытаний Бернулли, модель Нельсона [4].

В простой интуитивной модели проводится тестирование двумя группами программистов, использующими независимые тестовые наборы, независимо одна от другой. В процессе тестирования каждая из групп фиксирует все найденные ошибки. При оценке числа оставшихся в программе ошибок результаты тестирования обеих групп собираются и сравниваются.

В модели Коркорэна не используются параметры времени тестирования, а учитывается только результат N испытаний, в которых выявлено N_i ошибок i-го типа. Модель использует изменяющиеся вероятности отказов для различных типов ошибок. В отличие от рассмотренной выше статистической модели, модель Коркорэна определяет вероятность безотказного выполнения программы на момент оценки.

В модели последовательности испытаний Бернулли пространство элементарных событий содержит $2n$ точек, где n – число испытаний (в данном случае под испытанием подразумевается запуск программы). Каждый запуск программы имеет не менее двух исходов: правильный и неправильный. Обозначим вероятность неправильного исхода p, а вероятность правильного – $(1-p)$. Вероятность того, что из n запусков k приведут к неправильному результату, выражается формулой биномиального распределения

$$B(p,n,k) = C(n,k)p^k(1-p)^{n-k}, \quad (4)$$

где $C(n,k)$ – число сочетаний. Вероятность p априори неизвестна, но по результатам запусков известны n и k.

Функция B имеет максимум при $p = k/n$. В качестве меры надежности программы принимается величина

$$R = 1 - k/n. \quad (5)$$

Модель Нельсона при расчете надежности ПО учитывает вероятность выбора определенного тестового набора для очередного выполнения программы. Предполагается, что область данных, необходимых для выполнения тестирования программного средства, разделяется на k взаимоисключающих подобластей Z_i, $i = 1,2,...,k$. Пусть P_i – вероятность того, что набор данных Z_i будет выбран для очередного выполнения программы. Если к моменту оценки надежности было выполнено N_i прогонов программы на Z_i наборе данных и из них n_i прогонов закончились отказом, то надежность ПО определяется равенством

$$R = 1 - \sum_{i=1}^{k} \frac{n_i}{N_i} P_i. \quad (6)$$

Эмпирические модели базируются на анализе структурных особенностей программ. К этим моделям относятся модель сложности и модель, определяющая время доводки программ.

В модели сложности ПО характеризуется его размером, количеством программных модулей, количеством и особенностью межмодульных интерфейсов. Под программным модулем в данном случае следует понимать программную единицу, выполняющую определенную функцию и взаимосвязанную с другими модулями ПО. Существует несколько разновидностей моделей сложности. В каждой из них дается некая оценка сложности программы, которая считается пропорциональной ее надежности. В модели, определяющей время доводки программ, анализ модульных связей ПО строится на том, что каждая пара модулей имеет конечную (возможно, нулевую) вероятность того, что изменения в одном модуле вызовут изменения в другом. Данная модель позволяет на этапе тестирования определять возможное число необходимых исправлений и время, необходимое для доведения ПО до рабочего состояния.

Таким образом, к настоящему времени разработаны различные модели надежности ПО, представляющие собой математические модели, построенные для оценки зависимости надежности программного обеспечения от некоторых определяющих параметров. Однако ни одна из предложенных моделей не находит широкого использования на практике по той причине, что их применение неудобно или нецелесообразно. Некоторые модели, например модель Джелинского–Моранды, дают достаточно грубую оценку надежности. Поэтому дальнейшее развитие моделей надёжности ПО является актуальной задачей.

Известно, что абсолютно надежных программ не существует, так как абсолютная степень надежности не может быть теоретически доказана и, следовательно, недостижима. Однако важно знать, насколько надежно конкретное ПО. Описанные модели представляют теоретический подход и, как правило, имеют ограниченное применение.

Практика разработки ПО предполагает приоритет задачи обеспечения надежности над задачей ее оценки. Ситуация выглядит парадоксально: совершенно очевидно, что прежде чем обеспечивать надежность, следует научиться ее измерять. Но для этого нужно иметь практически приемлемую единицу измерения надежности ПО и модель ее расчета.

Анализируя вышесказанное, можно выбрать в качестве основных показателей надежности следующие:

– функция надежности – вероятность того, что ни одна ошибка не появится на определённом интервале;

– среднее время между отказами.

Количественная оценка надежности программных средств (ПС), несмотря на наличие большого числа вероятностных моделей (моделей роста надежности ПО [3]), далека от реального практического применения. Это связано со многими причинами. Одна из них состоит в том, что при разработке этих моделей принимаются достаточно грубые допущения, которые не учитывают реалий процесса разработки, тестирования и сопровождения и не могут распространяться на большую часть программных проектов. В то же время, как показано в работе [4], перечень допущений определяющим образом влияет на выбор моделей надежности ПО. Для удобства анализа показателей надежности ПО сложных систем целесообразно представить систему в виде совокупности менее сложных составляющих, обычно называемых программными модулями (ПМ). Программный модуль, в свою очередь, может быть разделен на более мелкие части и т. д. Таким образом, ПМ является аналогом элемента, для которого в дальнейшем определяются показатели надежности. Обычно он представляет собой логически самостоятельную программу. При исследовании надежности функционирования ПО необходимо решить две задачи. Первая (прямая задача) – по заданной структуре ПО, состоящего из некоторой совокупности ПМ, имеющих показатели надежности, необходимо найти показатель надежности ПО. Вторая задача (обратная задача) состоит в определении достижения максимального (минимального) значения показателя надежности ПО при ограничениях на ресурсы (время, стоимость и др.) или в минимизации величины ограничения на ресурсы.

Чтобы уменьшить число ошибок, необходимо учитывать факторы, влияющие на надёжность ПО:

– точность математической формализации задач обработки данных;

– полнота и обоснованность требований к программному обеспечению;

– степень удовлетворения требований при разработке ПО;

– степень отлаженности программ;

– качество структуры общего алгоритма обработки данных и степень согласованности отдельных программ в каждом программном модуле.

Основными путями, повышающими надёжность ПО, могут быть:

– использование системного принципа проектирования ПО (программное обеспечение проектируется по модульно-иерархической структуре).

– применение библиотечных процедур для реализации стандартных функций;

– резервирование программ и другие методы структурной избыточности.

Выводы. Проанализированы модели надежности и отмечены особенности отказов ПО. Рассмотрены основные подходы к обеспечению надёжности программного обеспечения. Выявлены факторы, влияющие на количество ошибок ПО. Показано, что рассмотренные модели надежности ПО в полной мере не дают возможности практического применения для оценки надежности ПО и необходимо продолжать исследования по этой проблеме.

1. *Даниев Ю. Ф., Пошивалов В. П. Резниченко Л. В.* Общие принципы обеспечения надежности эргатических систем. Системные технологии: міжвуз. сб. наук. праць. Днепропетровск, 2014. С. 121–126.
2. *Даниев Ю. Ф., Пошивалов В. П., Резниченко Л. В.* Системный подход к обеспечению надежности сложных систем. Системні технології: міжвуз. сб. наук. праць. Дніпро, 2017. Вип. 2 (109). С. 27–34.
3. *Липаев В. В.* Надежность программных средств. М.: Синтег. 1998. 220 с.
4. *Майерс Г.* Надежность программного обеспечения: пер. с англ. / под ред. В. Ш. Кауфмана. М.: Мир, 1980. 360 с.
5. *Тейер Т., Липов М., Нельсон Э.* Надежность программного обеспечения. М.: Мир, 1981. 323 с.
6. Оценка надёжности программной составляющей эргатической системы управления: 4.02 / *Даниев Ю. Ф., Пошивалов В. П.* // Информационные технологии в управлении сложными системами – 2013: научная конференция (19–20 июня 2013 г.): сборник докладов. Секция 2 / под редакцией академика НАН Украины В. В. Пилипенко. Днепропетровск: ИТМ НАН Украины и ГКА Украины, 2013. URL: http://www.itm.dp.ua/. 3 стор.
7. *Шапоров В. Н.* Надежность информационных систем. Сыктывкар. 2013. 86 с.
8. *Пальчун Б. П., Юсупов Р. М*. Оценка надежности программного обеспечения. СПб: Наука, 1994. 84 с.
9. *Штрик А. А., Осовецкий Л. Г., Мессих И. Г.* Структурное проектирование надежных программ встроенных ЭВМ. Л.: Машиностроение, 1989. 296 с.

Получено 04.12.2017,
в окончательном варианте 08.12.2017