УДК 004. 052

О.В. Мельник*

АНАЛИЗ МОДЕЛЕЙ ОЦЕНКИ НАДЕЖНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫХ И УПРАВЛЯЮЩИХ СИСТЕМ

Анализ надежности программного обеспечения состоит, как правило, из двух основных этапов: этап испытаний и тестирования и этап эксплуатации. На первом этапе обычно по результатам выборочных данных или субъективных наблюдений строится математическая модель, при помощи которой можно выполнить прогноз надежности программы на этапе эксплуатации. Анализ поведения на этапе эксплуатации предусматривает учет различных факторов и ограничений работы, таких как используемые аппаратные средства, способы обнаружения ошибок, времена исправления ошибок, допустимая область исходных данных, параллельное или последовательное совместное использование других программ, возможное использование избыточности [1].

---

* Работа выполнена под руководством д-ра техн. наук, проф., доцента ФГБОУ ВПО «ТГТУ» Ю.Т. Зырянова.

Возможно рассматривать надежность программного обеспечения как продукт производственно-технического назначения. Исходя из этого, надежность программного обеспечения – это комплексное свойство, состоящее, как и в случае технических объектов, из набора характеристик. Это корректность, устойчивость, восстанавливаемость и исправляемость программного обеспечения.

Для количественной оценки показателей надежности программного обеспечения используют модели надежности, под которыми понимаются математические модели, построенные для оценки зависимости надежности от заранее известных или определенных в ходе выполнения задания параметров. Эти модели можно разделить на две основные группы: эмпирические и аналитические. Выбирая методику оценки надежности, необходимо учитывать ее пригодность для различных стадий жизненного цикла, установления порядка ее совместного использования для определения надежности программного обеспечения на протяжении всего его жизненного цикла [2].

МОДЕЛИ ОЦЕНКИ НАДЕЖНОСТИ

Эмпирические модели предназначены для оценки сложности программного обеспечение. Это модели оценивают множество характеристик программного обеспечения, таких как длина программы, информационное содержание, число подсистем, число операторов, сложность интерфейса и т.п. Примером эмпирической модели может являться модель Холстеда, которая оценивает количество оставшихся в программе ошибок после окончания ее разработки:

$$N_{\text{ош}} = K_{\text{но}} V_{\text{оп}} \log_2(\eta_1 + \eta_2),$$

где $N_{\text{ош}}$ – число ошибок в программе; $K_{\text{но}}$ – коэффициент пропорциональности; $V_{\text{оп}}$ – число операторов в программе; η_1 – число операторов в программном средстве; η_2 – число операндов в программном средстве.

Преимущество эмпирических моделей в том, что они не содержат сложных формул и вычисления по ним просты. Недостатком же является их грубость и приблизительность. Кроме того, они не отражают динамики вычислительного процесса при эксплуатации программ.

Аналитические модели разделяются на статические и динамические. Среди динамических также можно выделить непрерывные и дискретные. При использовании непрерывной динамической модели предполагается, что функционирование программного обеспечения описывается набором последовательных состояний, переход между которыми происходит в случае возникновения отказа, за которым также следует восстановление.

Примерами дискретных моделей могут послужить модель Шумана и модель Мусса. В модели Шумана предполагается, что тестирование проводится в несколько этапов. Каждый этап представляет собой выполнение программы по набору тестовых данных. Выявленные в течение этапа тестирования ошибки регистрируются, но не исправляются. По завершении этапа исправляются все обнаруженные на этом этапе ошибки, корректируются тестовые наборы и проводится новый этап тестирования.

Предполагается, что при корректировке новые ошибки не вносятся и что интенсивность обнаружения ошибок пропорциональна числу оставшихся ошибок.

К преимуществам модели относится то, что по ней можно определить все неизвестные параметры, т.е. нет необходимости обращаться к другим моделям, что сокращает время расчета надежности. К недостаткам относится предположение, что при корректировке не вносятся новые ошибки, а это не всегда имеет место в реальных программах. Кроме того, в процессе тестирования необходимо регистрировать большое количество данных, необходимых для расчета по формулам этой модели.

Статические модели отличаются от динамических прежде всего тем, что в них не учитывается время появления ошибок. К статическим моделям относятся модель Миллса, модель Нельсона и модель Коркорэна.

Модель Коркорэна предполагает наличие в программном обеспечении многих источников программных отказов, связанных с различными типами ошибок, и разную вероятность их появления.

Преимуществом модели является то, что она учитывает существование в программном обеспечении нескольких источников ошибок, а также то, что расчет надежности с математической точки зрения проще, чем в других моделях.

Недостаток – необходимость определения статистическим методом вероятность того, что для очередного прогона программы будет выбран набор данных из предполагаемой области, что затрудняет расчеты.

К непрерывной динамической модели относятся модель Джелински – Моранды и модель переходных вероятностей Маркова.

Модель Джелински – Моранды основана на допущениях, что время до следующего отказа распределено экспоненциально, а интенсивность отказов программы пропорциональна количеству оставшихся в программе ошибок.

Согласно этим допущениям вероятность безотказной работы программного обеспечениякак функция времени t_i равна:

$$P(t_i) = e^{-\lambda t_i},$$

где интенсивность отказов $\lambda_i = C_D(N - (i-1))$; C_D – коэффициент пропорциональности; N – первоначальное количество ошибок;

$$C_D = \frac{\sum_{i=1}^{k-1} \frac{1}{N-i+1}}{\sum_{i=1}^{k-1} t_i},$$

где k – номер прогнозируемого отказа.

При известных значениях k; t_1, t_2, …, t_k можно найти значения параметров модели C_D и N, а затем интенсивность отказов, время от последнего до следующего отказа t_{k+1}, вероятность безотказной работы через время t_{k+1} после последнего отказа.

Основным преимуществом модели является простота расчетов. Недостаток этой модели состоит в том, что при неточном определении величины N интенсивность отказов программы может стать отрицательной, что приводит к бессмысленному результату. Кроме того, предполагается, что при исправлении обнаруженных ошибок не вносятся новые ошибки, что тоже не всегда выполняется.

ВЫВОДЫ

Таким образом, большинство моделей надежности программного обеспечения определяют надежность на начальных стадиях жизненного цикла. Применение рассмотренных моделей для оценки завершающих стадий жизненного цикла программного обеспечения ограничено по следующим причинам: на фазах производства и тестирования программного обеспечения информация о процессе отладки, обнаружении и устранении ошибок, как правило, недоступна; отказы при приемосдаточных испытаниях малоинтенсивны или отсутствуют. Поэтому для определения надежности программного обеспечения на всех стадиях его жизненного цикла целесообразно применять, как минимум, две модели надежности программного обеспечения. Модель надежности программного обеспечения для фазы разработки выбирается для каждой конкретной программы. Для этого нужно собрать данные об ошибках, на основании имеющихся данных выбрать модель надежности, а затем выполнить тесты, показывающие, насколько эта модель подходит. Для определения надежности программного обеспечения на завершающих стадиях наиболее эффективно применять модели надежности с системно-независимым аргументом.

СПИСОК ЛИТЕРАТУРЫ

1. Майерс, Г. Надежность программного обеспечения / Г. Майерс ; пер. с англ. Ю.Ю. Галимова ; под ред. В.Ш. Кауфмана. – М. : Мир, 1980. – 356 с.

2. Карповский, Е.Я. Надежность программной продукции / Е.Я. Карповский, С.А. Чижов. – Киев : Изд-во Техника, 1990. – 160 с.

Кафедра «Конструирование радиоэлектронных и микропроцессорных систем» ФГБОУ ВПО «ТГТУ»