

МЕТОДОЛОГИЯ РАСЧЕТА НАДЕЖНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ АВТОМАТИЗИРОВАННЫХ СИСТЕМ

А.В. Юров, Леонов С.В.
СТИ НИЯУ МИФИ
yurov-anton@mail.ru

Введение

Современные тенденции развития промышленной отрасли, дают представление о том, что для успешного проектирования АСУ ТП необходимо применять высокопроизводительные контроллеры, а также использовать современные методы оценки надежности проектируемых систем. Немаловажную роль в оценке надежности АСУ ТП играет повышение безопасности и работоспособности алгоритмов управления применяемых в контроллерах управления.

Целью данной работы является анализ существующих методов оценки надежности программного обеспечения на предмет их применимости для оценки алгоритмов управления.

Понятие надежности ПО

Согласно определению: программа, не содержащая ошибок, считается правильной, а следовательно – абсолютно надежной, поскольку она не дает неверные результаты. На основании данного факта сформировалось представление о том, что количество ошибок в программе можно рассматривать как наиболее естественную меру надежности [1]. Надежность программы не зависит от количества ошибок, находящихся в ней, поскольку:

1. Число ошибок в программе - величина «ненаблюдаемая», т.е. проявляется результат ошибок, сами ошибки мы видеть не можем.

2. Программа может функционировать неправильно не из-за одной, а сразу нескольких ошибок.

3. Ошибки могут быть взаимосвязаны между собой и компенсировать друг друга, так что после исправления одной ошибки программа может начать «работать хуже».

4. Надежность является характеристикой и показывает частоту проявления ошибок, в то же время ошибки могут появляться с различной частотой: некоторые ошибки могут оставаться невыявленными даже после нескольких месяцев и лет эксплуатации. Также существуют примеры программ, обладающих нулевой надежностью, в которых одна единственная ошибка приводит к неверной работе программы при любых исходных данных.

Следует отметить тот факт, что если число ошибок рассматривать как меру надежности, то по терминологии теории вероятности — данное число есть случайная величина, однако вопрос, на каком пространстве элементарных событий она задана, нигде не затрагивается. Важно отметить, что при исправлении ошибки или коррекции кода

получается программа с новым показателем надежности [2].

Следовательно, можно прийти к выводу о том, что количество ошибок в программе описывает не саму программу, а ее изготовителей, а также используемый инструментарий.

Модели оценки надежности ПО

Согласно общепринятой классификации все модели оценки надежности ПО можно разделить на следующие: эмпирические и аналитические. К аналитическим моделям в свою очередь относятся: статические, непрерывные динамические, а также дискретные модели [3]. Преимущество эмпирических моделей в том, что они не содержат сложных формул и вычисления по ним просты. К недостаткам эмпирических моделей относится то, что они очень грубы, весьма приблизительны. Кроме того, они не отражают динамики вычислительного процесса при эксплуатации программ [4]. **Модель Джелинского-Моранды.** Метод относится к непрерывным динамическим моделям. Данная модель основана на допущении о том, что время до следующего отказа распределено экспоненциально, а интенсивность отказов программы является величиной пропорциональной количеству ошибок, оставшихся в программе. Согласно данным допущениям вероятность безотказной работы ПО как функции времени t равна (1):

$$P(t_i) = e^{-\lambda t}, \quad (1)$$

интенсивность отказов (2):

$$\lambda_i = C_D(N - (i - 1)). \quad (2)$$

где: N – первоначальное количество ошибок;

C_D – коэффициент пропорциональности.

Основным достоинством 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.

Модель Шумана. В данной модели существует несколько этапов тестирования. На каждом этапе осуществляется выполнение программы по набору тестовых данных. Найденные ошибки в процессе тестирования регистрируются, однако не исправляются. На стадии завершения этапа все найденные ошибки исправляются, тестовые наборы корректируются, после чего осуществляется следующий этап тестирования. Модель Шумана предполагает программное обеспечение на i -м этапе тестирования и имеет

следующую функцию надежности (3,4):

$$R_i(t) = e^{-\lambda t}, \quad (3)$$

$$\lambda_i = C(N - n_{i-1}). \quad (4)$$

где N – первоначальное количество ошибок в программном обеспечении;

$N - n_{i-1}$ – количество ошибок, оставшихся к началу i -го этапа;

C – коэффициент пропорциональности, равный (5):

$$C = \frac{\sum_{i=1}^k \frac{m_i}{N - n_{i-1}}}{\sum_{i=1}^k t_i}. \quad (5)$$

Считается, что частота обнаружения ошибок пропорциональна количеству оставшихся ошибок, а также что при исправлении программы новые ошибки не вносятся. К преимуществам модели можно отнести то, что по ней можно определить все неизвестные параметры, то есть нет необходимости обращаться к другим моделям, что сокращает время расчета надежности.

Модель Миллса. Использование данной модели предполагает необходимость искусственного внесения в программу перед началом тестирования некоторого количества известных ошибок. Ошибки вносятся случайным образом, после чего фиксируются в протоколе искусственно внесенных ошибок. Специалист, который осуществляет тестирование, как правило не должен знать ни количество, ни характер внесенных ошибок. Предполагается, что все ошибки, как естественные, так и искусственные равновероятно могут быть найдены в процессе тестирования. Программа тестируется в течение некоторого времени, после чего собираются статистические данные об обнаруженных ошибках.

Допустим, что после тестирования программы обнаружено n собственных ошибок программы и v искусственно внесенных ошибок. Тогда первоначальное количество ошибок в программе N можно оценить по следующей формуле (6):

$$N = n \frac{S}{v}, \quad (6)$$

где S – количество искусственно внесенных ошибок. Достоинством модели Миллса является простота применяемого математического аппарата и наглядность. Недостатки: необходимость внесения искусственных ошибок (этот процесс плохо формализуем), достаточно вольное допущение величины K , которое основывается исключительно на интуиции и опыте человека, производящего оценку, то есть допускает большое влияние субъективного фактора.

Заключение

Проводя анализ моделей для определения надежности программного обеспечения, можно заключить, что большинство из них определяет надежность ПО только на начальных стадиях жизненного цикла программы. Применение данных моделей для оценки надежности ПО на завершающих стадиях жизненного цикла программы имеет ограничения по следующим причинам:

– недоступность на стадии производства и использования программного обеспечения информации о процессе отладки, а также обнаружении и устранении ошибок;

– малая интенсивность отказов при приемосдаточных испытаниях.

Соответственно имеет смысл рекомендовать для определения надежности программного обеспечения на всех стадиях его жизненного цикла использование по меньшей мере, двух моделей оценки надежности ПО. Модель по оценке надежности ПО должна подбираться для каждой стадии разработки и для каждой конкретной программы. Для этого необходимо наличие данных о имеющихся ошибках, на основании имеющихся данных целесообразно осуществлять подбор модели надежности, после чего также необходимо выполнить тесты, показывающие, пригодность выбранной модели. Для определения надежности программного обеспечения на завершающих стадиях целесообразно применять модели надежности с системно-независимым аргументом, поскольку данные модели дают наиболее точную оценку надежности ПО. Исходя из выше представленной информации, можно сделать вывод о том, что наибольшее влияние на надежность программного обеспечения оказывают внутренние ошибки, а также ошибки, которые находятся на первых этапах эксплуатации программ. Соответственно для оценки надежности ПО целесообразно использовать следующие модели: Модель Миллса, Модель Шумана, Модель Джелинского-Моранды. Данные модели позволят наиболее полно оценить надежность алгоритмов управления проектируемой АСУ ТП.

Список использованных источников

1. Г. Майерс. Надежность программного обеспечения. Москва, Мир, 1980 год.
2. Карповский Е.Я., Чижов С.А. Надежность программной продукции. – К.: Техника, 1990. – 160 с.
3. Фатуев В.П. Надежность автоматизированных информационных систем: Учебное пособие // Высоцкий В.И., Бушинский В.И. – Т.: ТГУ, 1998. – 104 с.
4. Шураков В.В. Надежность программного обеспечения систем обработки данных: Учебник для вузов. – М.: Статистика, 1981. – 216 с.