Лекция 7. Модели надежности программного обеспечения

Термин **модель надежности программного обеспечения**, как правило, относится к математической модели, построенной для оценки зависимости надежности программного обеспечения от некоторых определенных параметров. Значения таких параметров либо предполагаются известными, либо могут быть измерены в ходе наблюдений или экспериментального исследования процесса функционирования программного обеспечения. Данный термин может быть использован также применительно к математической зависимости между определенными параметрами, которые хотя и имеют отношение к оценке надежности программного обеспечения, но тем не менее не содержат ее характеристик в явном виде. Например, поведение некоторой ветви программы на подмножестве наборов входных данных, с помощью которых эта ветвь контролируется, существенным образом связано с надежностью программы, однако характеристики этого поведения могут быть оценены независимо от оценки самой надежности. Другим таким параметром является частота ошибок, которая позволяет оценить именно качество систем реального времени, функционирующих в непрерывном режиме, и в то же время получать только косвенную информацию относительно надежности программного обеспечения (например, в предположении экспоненциального распределения времени между отказами).

Одним из видов модели надежности программного обеспечения, которая заслуживает особого внимания, является так называемая феноменологическая, или эмпирическая, модель. При разработке моделей такого типа предполагается, что связь между надежностью и другими параметрами является статической. С помощью подобного подхода пытаются количественно оценить те характеристики программного обеспечения, которые свидетельствуют либо о высокой, либо о низкой его надежности. Так, например, параметр сложность программы характеризует степень уменьшения уровня ее надежности, поскольку усложнение программы всегда приводит к нежелательным последствиям, в том числе к неизбежным ошибкам программистов при составлении программ и трудности их обнаружения и устранения. Иначе говоря, при разработке феноменологической модели надежности программного обеспечения стремятся иметь дело с такими параметрами, соответствующее изменение значений которых должно приводить к повышению надежности программного обеспечения.

Модели надежности программных средств (МНПС) подразделяются на аналитические и эмпирические. Аналитические модели дают возможность рассчитать количественные показатели надежности, основываясь на данных о поведении программы в процессе тестирования (измеряющие и оценивающие модели). Эмпирические модели базируются на анализе структурных особенностей программ. Они рассматривают зависимость показателей надежности от числа межмодульных связей, количества циклов в модулях, отношения количества прямолинейных участков программы к количеству точек ветвления и т.д. Часто эмпирические модели не дают конечных результатов показателей надежности, однако они включены в классификационную схему, так как развитие этих моделей позволяет выявлять взаимосвязь между сложностью ПС и

его надежностью. Эти модели можно использовать на этапе проектирования ПС, когда осуществлена разбивка на модули и известна его структура.

1 Аналитические модели надежности

Аналитическое моделирование надежности ПС включает четыре шага:

- определение предположений, связанных с процедурой тестирования ПС;
- разработка или выбор аналитической модели, базирующейся на предположениях о процедуре тестирования;
- выбор параметров моделей с использованием полученных данных;
- применение модели — расчет количественных показателей надежности по модели.

1.1 Динамические модели

1.1.1 Модель Шумана

Исходные данные для модели Шумана, которая относится к динамическим моделям дискретного времени, собираются в процессе тестирования ПС в течение фиксированных или случайных временных интервалов. Каждый интервал — это стадия, на которой выполняется последовательность тестов и фиксируется некоторое число ошибок.

Модель Шумана может быть использована при определенным образом организованной процедуре тестирования. 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 (собирается статистика об ошибках), но не исправляются. По завершении этапа на основе собранных данных о поведении ПС на очередном этапе тестирования может быть использована модель Шумана 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 при корректировке не вносятся. Скорость обнаружения ошибок пропорциональна числу оставшихся ошибок. Общее число машинных инструкций в рамках одного этапа тестирования постоянно.

1.1.2 Модель Джелинского - Моранды

Модель Джелинского - Моранды относится к динамическим моделям непрерывного времени. Исходные данные для использования этой модели собираются в процессе тестирования ПС. При этом фиксируется время до очередного отказа. Основное положение, на котором базируется модель, заключается в том, что значение интервалов времени тестирования между обнаружением двух ошибок имеет экспоненциальное распределение с частотой ошибок (или интенсивностью отказов), пропорциональной числу еще не выявленных ошибок. Каждая обнаруженная ошибка устраняется, число оставшихся ошибок уменьшается на единицу.

1.1.3 Модель Шика-Волвертона

Модификация модели Джелинского - Моранды для случая возникновения на рассматриваемом интервале более одной ошибки предложена Волвертоном и Шиком. 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. Частота ошибок (интенсивность обнаружения ошибок) предполагается постоянной в течение интервала времени ti и пропорциональна числу ошибок, оставшихся в программе по истечении (i-1)-го интервала; но она пропорциональна также и суммарному времени, уже затраченному на тестирование (включая среднее время выполнения программы в текущем интервале).

В данной модели наблюдаемым событием является число ошибок, обнаруживаемых в заданном временном интервале, а не вре-мя ожидания каждой ошибки, как это было для модели Джелинского - Моранды. В связи с этим модель относят к группе дискретных динамических моделей.

1.1.4 Модель Муса

Модель Муса относят к динамическим моделям непрерывного времени. Это значит, что в процессе тестирования фиксируется время выполнения программы (тестового прогона) до очередного отказа. Но считается, что не всякая ошибка ПС может вызвать отказ, поэтому допускается обнаружение более одной ошибки при выполнении программы до возникновения очередного отказа.

В модели Муса различают два вида времени:

1) суммарное время функционирования т, которое учитывает чистое время тестирования до контрольного момента, когда проводится оценка надежности;

2) оперативное время t — время выполнения программы, планируемое от контрольного момента и далее при условии, что дальнейшего устранения ошибок не будет (время безотказной работы в процессе эксплуатации).

Для суммарного времени функционирования т предполагается:

- интенсивность отказов пропорциональна числу неустраненных ошибок;
- скорость изменения числа устраненных ошибок, измеряемая относительно суммарного времени функционирования, пропорциональна интенсивности отказов.

Один из основных показателей надежности, который рассчитывается по модели Муса, — средняя наработка на отказ. Этот показатель определяется как математическое ожидание временного интервала между последовательными отказами и связан с надежностью.

1.1.5 Модель переходных вероятностей

Эта модель основана на марковском процессе, протекающем в дискретной системе с непрерывным временем.

Процесс, протекающий в системе, называется марковским (или процессом без последствий), если для каждого момента времени вероятность любого состояния системы в будущем зависит только от состояния системы в настоящее время (to) и не зависит от того, каким образом система пришла в это состояние. Процесс тестирования ПС рассматривается как марковский процесс.

1.2 Статические модели

Статические модели принципиально отличаются от динамических прежде всего тем, что в них не учитывается время появления ошибок в процессе тестирования и не используется никаких предположений о поведении функции риска X(t). Эти модели строятся на твердом статистическом фундаменте.

1.2.1 Модель Миллса

Использование этой модели предполагает необходимость перед началом тестирования искусственно вносить в программу («засорять») некоторое количество известных ошибок. Ошибки вносятся случайным образом и фиксируются в протоколе искусственных ошибок. Специалист, проводящий тестирование, не знает ни количества, ни характера внесенных ошибок до момента оценки показателей надежности по модели Миллса. Предполагается, что все ошибки (как естественные, так и искусственно внесенные) имеют равную вероятность быть найденными в процессе тестирования.

Тестируя программу в течение некоторого времени, собирают статистику об ошибках. В момент оценки надежности по протоколу искусственных ошибок все ошибки делятся на собственные и искусственные.

1.2.2 Модель Липова

Липов модифицировал модель Миллса, рассмотрев вероятность обнаружения ошибки при использовании различного числа тестов. Если сделать то же предположение, что и в модели Миллса.

Модель Липова дополняет модель Миллса, давая возможность оценить вероятность обнаружения определенного количества ошибок к моменту оценки.

1.2.3 Простая интуитивная модель

Использование этой модели предполагает проведение тестирования двумя группами программистов (или двумя программистами в зависимости от величины программы) независимо друг от друга, использующими независимые тестовые наборы. В процессе тестирования каждая из групп фиксирует все найденные ею ошибки. При оценке числа оставшихся в программе ошибок результаты тестирования обеих групп собираются и сравниваются.

Получается, что первая группа обнаружила N1 ошибок, вторая — N2, а N12— это ошибки, обнаруженные обеими группами.

1.2.4 Модель Коркорэна

Модель Коркорэна относится к статическим моделям надежности ПС, так как в ней не используются параметры времени тестирования и учитывается только результат N испытаний, в которых выявлено N ошибок i-го типа. Модель использует изменяющиеся вероятности отказов для различных типов ошибок.

2 Эмпирические модели надежности

Эмпирические модели в основном базируются на анализе структурных особенностей программного средства (или программы). Как указывалось ранее, эмпирические модели часто не дают конечных результатов показателей надежности, однако их использование на этапе проектирования ПС полезно для прогнозиро-ания требующихся ресурсов тестирования, уточнения плановых сроков завершения проекта и т.д.

2.1 Модель сложности

В литературе неоднократно подчеркивается тесная взаимосвязь между сложностью и надежностью ПС. Если придерживаться упрощенного понимания сложности ПС, то она может быть описана такими характеристиками, как размер ПС (количество программных модулей), количество и сложность межмодульных интерфейсов.

Под программным модулем в данном случае следует понимать программную единицу, выполняющую определенную функцию (ввод, вывод, вычисление и т.д.) взаимосвязанную с другими модулями ПС. Сложность модуля ПС может быть описана, если рассматривать структуру программы.

В качестве структурных характеристик модуля ПС используются:

1) отношение действительного числа дуг к максимально возможному числу дуг, получаемому искусственным соединением каждого узла с любым другим узлом дугой;

2) отношение числа узлов к числу дуг;

3) отношение числа петель к общему числу дуг.

Для сложных модулей и для больших многомодульных программ составляется имитационная модель, программа которой «засоряется» ошибками и тестируется по случайным входам. Оценка надежности осуществляется по модели Миллса.

При проведении тестирования известна структура программы, имитирующей действия основной, но не известен конкретный путь, который будет выполняться при вводе определенного тестового входа. Кроме того, выбор очередного тестового набора из множества тест- входов случаен, т.е. в процессе тестирования не обосновывается выбор очередного тестового входа. Эти условия вполне соответствуют реальным условиям тестирования больших программ.

Полученные данные анализируются, проводится расчет показателей надежности по модели Миллса (или любой другой из описанных выше), и считается, что реальное ПС, выполняющее

аналогичные функции, с подобными характеристиками и в реальных условиях должно вести себя аналогичным или похожим образом.

Преимущества оценки показателей надежности по имитационной модели, создаваемой на основе анализа структуры будущего реального ПС, заключаются в следующем:

- модель позволяет на этапе проектирования ПС принимать оптимальные проектные решения, опираясь на характеристики ошибок, оцениваемые с помощью имитационной модели;
- модель позволяет прогнозировать требуемые ресурсы тестирования;
- модель дает возможность определить меру сложности программ и предсказать возможное число ошибок и т.д.

К недостаткам можно отнести высокую стоимость метода, так как он требует дополнительных затрат на составление имитационной модели, и приблизительный характер получаемых показателей.

2.2 Модель, определяющая время доводки программы

Эта модель используется для ПС, которые имеют иерархическую структуру, т.е. ПС как система может содержать подсистемы, которые состоят из компонентов, а те, в свою очередь, состоят из V модулей. Таким образом, ПС может иметь V различных уровней композиции. На любом уровне иерархии возможна взаимная зависимость между любыми парами объектов системы. Все взаимозависимости рассматриваются в терминах зависимости между парами модулей.

Анализ модульных связей строится на том, что каждая пара модулей имеет конечную (возможно, нулевую) вероятность, изменения в одном модуле вызовут изменения в другом модуле.

Данная модель позволяет на этапе тестирования, а точнее при тестовой сборке системы, определять возможное число необходимых исправлений и время, необходимое для доведения ПС до рабочего состояния.

Основываясь на описанной процедуре оценки общего числа изменений, требуемых для доводки ПС, можно построить две различные стратегии корректировки ошибок:

- фиксировать все ошибки в одном выбранном модуле и устранить все побочные эффекты, вызванные изменениями этого модуля, отрабатывая таким образом последовательно все модули;
- фиксировать все ошибки нулевого порядка в каждом модуле, затем фиксировать все ошибки первого порядка и т.д.

Исследование этих стратегий доказывает, что время корректировки ошибок на каждом шаге тестирования определяется максимальным числом изменений, вносимых в ПС на этом шаге, а общее время — суммой максимальных времен на каждом шаге.

Это подтверждает известный факт, что тестирование обычно является последовательным процессом и обладает значительными возможностями для параллельного исправления ошибок, что часто приводит к превышению затрачиваемых на него ресурсов над запланированными.