

КЛАССИФИКАЦИЯ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ ОЦЕНКИ НАДЁЖНОСТИ ПО

Трубач Н.А.

Научный руководитель – к.т.н., доцент Попова Ю.Б.

Одним из важнейших показателей качества программного обеспечения является его надёжность.

Надёжность характеризует способность ПС в конкретных областях применения выполнять заданные функции в соответствии с программными документами в условиях возникновения отклонений в среде функционирования, вызванных сбоями технических средств, ошибками во входных данных, ошибками обслуживания и другими дестабилизирующими воздействиями (ГОСТ 28195-99).

В области надёжности аппаратуры достигнут уровень, когда создан ряд математических методов. Эти математические методы (главным образом представлены в форме **математических моделей**), построенные для оценки зависимости надёжности от заранее известных или оцененных в ходе создания программного средства параметров.

Математическая модель – приближённое описание объекта моделирования, выраженное с помощью **математической** символики.

Одним из самых важных параметров программного обеспечения является число ошибок, оставшихся в программе. Если знать это число, то можно оценить стоимость работ по сопровождению и определить уровень доверия к программе.

Известные математические модели оценки надёжности программного обеспечения делятся на два вида – аналитические и эмпирические модели. Аналитические математические модели представлены двумя группами: динамические и статические.

В динамических моделях поведение программного обеспечения (появление отказов) рассматривается во времени. В то время, как в статических математических моделях появление отказов не связывают со временем, а учитывается только зависимость количества ошибок от числа тестовых прогонов. Среди динамических математических моделей можно выделить дискретные и непрерывные модели.

К динамическим математическим моделям относятся следующие: **модель Шумана**– данные для модели Шумана собираются в процессе тестирования ПО в течение фиксированных или случайных временных интервалов; **модель LaPadula**– по этой модели выполнение

последовательности тестов производится в m этапов, каждый из которых заканчивается внесением изменений (исправлений) в ПО; **модель Шика-Волвертона**—эта модель 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; **модель Джелинского-Маранды**—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); **модель Муса** предполагает, что в процессе тестирования фиксируется время выполнения программы до очередного отказа; **модель переходных вероятностей** —предполагается, что в процессе тестирования выявляется по одной ошибке.

Среди статических математических моделей можно выделить модели по области ошибок и модели по области данных.

К статическим математическим моделям относятся следующие модели: **модель Миллса**— использование этой модели предполагает необходимость перед началом тестирования искусственно «засорять» программу, т.е. вносить в нее некоторое количество известных ошибок; **модель Липова** — в модели предполагается, что собственные и искусственные ошибки имеют равную вероятность быть найденными; **простая интуитивная модель** - использование этой модели предполагает проведение тестирования двумя группами программистов, использующими независимые тестовые наборы, независимо одна от другой (в процессе тестирования каждая из групп фиксирует все найденные ею ошибки, и при оценке числа оставшихся в программе ошибок результаты тестирования обеих групп собираются и сравниваются); **модель Коркорэна**— в этой модели не используются параметры времени тестирования и учитывается только результат испытаний, в которых выявлено N_i ошибок i -го типа; **модель Нельсона** — данная модель при расчете надежности ПО учитывает вероятность выбора определенного тестового набора для очередного выполнения программы.

Эмпирические математические модели базируются на анализе структурных особенностей программ. Предполагается, что связь между надежностью и другими параметрами является статической. Они часто не дают конкретных значений параметров надежности программы. При этом их использование считается полезным на этапе проектирования программ для прогнозирования ресурсов тестирования.

К эмпирическим математическим моделям относятся следующие модели: **модель сложности**—сложность ПО характеризуется его размером (количеством программных модулей), количеством и сложностью межмодульных интерфейсов, также модель сложности позволяет на этапе проектирования ПО принимать оптимальные проектные решения, опираясь на характеристики ошибок; **модель, определяющая время доводки программы**— данная модель позволяет на этапе тестирования, а точнее при тестовой сборке системы, определять возможное число необходимых исправлений и время, необходимое для доведения ПО до рабочего состояния.

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

Следует признать, 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 математические модели представляют теоретический подход и, как правило, имеют ограниченное применение. Для определения надежности программного обеспечения на всех стадиях его жизненного цикла целесообразно применять, как минимум, две математические модели надежности программного обеспечения.

Литература

1. Майерс, Г. Надежность программного обеспечения / Г. Майерс – 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.
3. Василенко, Н.В. Модели оценки надёжности программного обеспечения / Н.В. Василенко, В.А. Макаров // Вестник Новгородского государственного университета. – 2004. – №28 – С.126 – 132.

4. Василенко, Н.В. Оценка надёжности программного обеспечения / Н.В. Василенко, В.А. Макаров // Вестник Новгородского государственного университета. – 2005. – №30 – С.88 – 92.

УДК 681.3.06

МНОГОУРОВНЕВАЯ АРХИТЕКТУРА

А.А. Кислач

Научный руководитель – Н.А. Разоренов, к.т.н. доцент

Многоуровневая архитектура - одна из архитектурных парадигм разработки ПО, при которой разбиение приложения на самостоятельные составные части происходит по реализуемой ими функциональности.

Характерные особенности многоуровневой архитектуры:

- требуемая функциональность реализуется в одном уровне и не дублируется в других;
- разделение возможностей приложения по функциональным областям: пользовательский интерфейс, сервисный слой, бизнес-логика, доступ к данным и др.;
- каждый уровень должен четко реализовывать ту функциональность, к области которой он относится, не совмещая код других функциональных областей;
- организация передачи данных между слоями, через компоненты доступа к данным, далее через бизнес-логику, с передачей через контролирующие сервисы;
- уровни слабо связаны, данными обмениваются явно;
- описана как перевернутая пирамида повторного использования: каждый слой агрегирует ответственности и абстракции уровня, расположенного непосредственно под ним;
- физически все слои могут быть развернуты на одном компьютере или распределены по разным компьютерам;
- использование в других системах (за счёт слабого связывания и чётко определённой задачи);

Уровни данной архитектуры:

- уровень сущностей (данных) содержит все сущности, используемые в проектах приложения;