УДК 681.3

МОДЕЛЬ ОЦЕНКИ НАДЕЖНОСТИ ОБЪЕКТНО-ОРИНТИРОВАНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Зозуля Е.Н.
Научный руководитель - к.т.н., доцент Царев Р.Ю.

Сибирский федеральный университет

В работе предложена модель оценки надежности объектно-ориентированного программного обеспечения, проведены ее исследования и выявлены особенности ее применения для оценки надежности программного обеспечения с различной архитектурой

Исходя из идеологии объектно-ориентированного программирования, класс является капсулой, содержащей данные и методы их обработки, доступ к которой возможен через специально разрешенный интерфейс – данные и методы, описанные как Public. Класс можно рассматривать как отдельный модуль, который значительно легче исследовать – связь между модулями возможна через интерфейс или через наследование. Наследование является одним из базовых принципов объектно-ориентированного программирования, суть которого заключается в том, что, если в данном классе не переопределены (не определены) какие-либо члены, то они будут взяты из родительского класса. При оценке надежности классов-потомков необходимо учитывать влияние классов-предков.

Для объектно-ориентированного подхода архитектура программного обеспечения - совокупность иерархий классов. Каждый класс - совокупность свойств (переменных) и методов (функций) объекта. Процесс - последовательный переход от метода одного класса к методу другого класса.

Обозначения:

F - общее число компонентов в архитектуре *ПО*;

R_i - коэффициент надежности компонента i, $i=1,\ldots,F$;

C_i - стоимость разработки компонента i, $i=1,\ldots,F$;

PU_i - вероятность использования компонента i, $i=1,\ldots,F$;

PF_i - вероятность сбоя в компоненте i, $i=1,\ldots,F$;

PL_{ij} - условная вероятность сбоя в компоненте i при появление сбоя в компоненте j, $i=1,\ldots,F$, $j=1,\ldots,F$;

TA_i - относительное время доступа к компоненту i, $i=1,\ldots,F$;

TU_i - относительное время использования компонента i, $i=1,\ldots,F$;

S - коэффициент готовности программного обеспечения;

R_S - коэффициент надежности системы;

TR - среднее время простоя программного обеспечения;

MTTF - среднее время появления сбоя.

Для данной модели обязательно выполнение условия:

$$\sum_{i=1}^{F} PU_i = 1.$$

Параметр TR вычисляется как:

$$TR = \sum_{i=1}^{F} [PU_i \times PF_i \times TA_i + \sum_{j=1, j\neq i}^{F} [PL_{ji} \times TA_j]].$$

Параметр MTTF вычисляется как:

$$MTTF = \sum_{i=1}^{F}[PU_i \times (1 - PF_i) \times [TU_i + \sum_{j=1, j \neq i}^{F}[(1 - PL_{ji}) \times TU_j]]].$$

Коэффициент готовности системы вычисляется как:

$$S = MTTF / (MTTF + TR).$$

Коэффициент надежности системы вычисляется как:

$$R_S = \sum_{i=1}^{F} PU_i \times R_i \text{ , где } R_i = 1 - \prod_{k \in Zi} PF_{ik}.$$

Выбор уровня детализации компонент - очень важный вопрос. Под компонентом в объектно-ориентированной модели будем понимать класс или метод с операционными профилями, участвующих в финальной сборке системы. Класс рассматривается как компонент для оценки общей надежности системы - совокупности классов для решения задач из определенной проблемной области.

В операционный профиль входит множество подобластей входных диапазонов параметров, с указанием надежности для каждого диапазона. Для каждого компонента обязательно составляется хотя бы один диапазон.

Выход компонента может являться входным профилем для другого компонента, таким образом, можно говорить о входных и выходных профилях. Для построения модели достаточно использовать только входные профили.

Для описания профиля используется плотность вероятности. На практике очень трудно достигнуть точных данных о профиле от пользователей ПО. Лучшее, что можно сделать, это описать его как гистограмму вероятностей над обширными классами входов.

Применение основных понятий теории надежности для объектно-ориентированного программного обеспечения позволяет получить ряд четких, хорошо измеряемых интегральных показателей качества программ. Приведенные критерии используются в основном при испытании объектно-ориентированного программного обеспечения и на завершающих фазах комплексной отладки. Использование данной модели для оценки качества программных модулей и групп программ, крайне затруднено.

Процесс разработки объектно-ориентированного программного обеспечения происходит во времени, и характеристики его классов в каждой конкретной версии могут служить частными конструктивными критериями для оценки качества программы. В данной модели надежность объектно-ориентированного программного обеспечения определяется в зависимости от надежности его классов (компонент) и вероятности их использования. При этом ошибками программного обеспечения будем называть ошибки, приводящие к нарушению работоспособности программ, и ошибки, искажающие результаты, но не влияющие на надежность функционирования программ. В этом случае для интенсивности тестирования характерна положительная обратная связь: чем больше выявляется ошибок в программе на некотором интервале времени, тем шире должно быть варьирование тестовых данных и больше тестов. По мере устранения ошибок в программе частота их обнаружения снижается, и специалисты, осуществляющие отладку, попадают в область низкого темпа обнаружения ошибок.

Наработка на отказ учитывает ситуации потери работоспособности программного обеспечения, когда длительность восстановления велика и превышает допустимое значение времени, разделяющее события сбоя и отказа. При этом большое значение имеют средства оперативного контроля и восстановления (рестарта). Качество программного обеспечения более полно отражает среднее время простоя программного обеспечения и среднее время появления сбоя.

Таким образом, выполненные исследования выявили особенности предложенной модели оценки надежности объектно-ориентированного программного обеспечения, а также показали особенности применения данной модели для практических задач.