МЕТОД ОЦЕНКИ НАДЕЖНОСТИ ФУНКЦИОНИРОВАНИЯ СЛОЖНЫХ ВЫЧИСЛИТЕЛЬНЫХ ПРОГРАММНЫХ СРЕДСТВ

А.Н. Коварцев, Д.А. Попова-Коварцева

Самарский государственный аэрокосмический университет имени академика С.П. Королёва (национальный исследовательский университет)

Известно, что понятие надежности программных продуктов имеет ряд специфических особенностей и существенных отличий от понятия надежности технических систем. Это в значительной степени усложняет оценку системной надежности сложного программного продукта. В работе предлагается новый показатель надежности - средняя надежность, а так же приводится метод оценки средней надежности сложной программной системы на основе известных характеристик составляющих её модулей. Приводится теоретическое обоснование предложенного подхода.

Ключевые слова: надежность программных средств, ненадежность, маршрут исполнения, программный модуль.

Введение

Надежность является важным и естественным требованием, предъявляемым к качеству современных аппаратно-программных комплексов (АПК). В области обеспечения надежности технических устройств достигнут очень высокий уровень. Например, в работе [1] показано, что вероятность отказов вычислителей пилотажно-навигационного комплекса самолетов типа ТУ-324 составляет 10^{-10}, что соответствует современным требованиям к безотказности аппаратуры, применяемой на космических аппаратах и в авиации. Что же касается оценки надежности программного обеспечения АПК, то положение дел в этой области обстоит не вполне удовлетворительно.

Долгое время основное внимание уделялось непосредственно технике программирования, а к проблеме тестирования и оценки надежности программных продуктов относились как к «неизбежному злу. О наличии ошибок в программе каждый раз убеждались на практике. Более серьезное отношение к оценке надежности программных средств (ПС) проявилось в конце 90-х годов в связи с существенным усложнением ПС и пониманием, что в АПК конечный результат обеспечивается совместной работой как аппаратной, так и его программной частей.

Однако наиболее весомой причиной отсутствия методов оценки надежности ПС является отсутствие соответствующей теоретической базы, приемлемой для исследования именно надежности ПС.

1. Понятие надежности функционирования программных средств

Первоначально определимся, что такое *надежность* ПС. В работе [2] надежность определяется как *«...вероятность того, что при функционировании системы в течение некоторого периода времени не будет обнаружено ни одной ошибки».* Далее выясняется, что понятие отказа программы субъективно: одна и та же программа может устроить одного пользователя и не устроить другого. Более того, программные ошибки имеют разную степень воздействия на конечный результат: одни приводят к катастрофе, другие

выводят на экран дисплея тексты с орфографическими ошибками. Поэтому Г.Майерс [2] уточняет понятие надежности следующим образом:

Надежность программного обеспечения есть вероятность его работы без отказов в течение определенного периода времени, рассчитанная с учетом стоимости для пользователя каждого отказа.

Из последнего определения следует, что понятие надежности применительно к программным средствам является сослагательным и, говоря о надежности ПС, мы должны упоминать, какие типы ошибок имеются в виду. Понятие надежность элементов технических устройств обычно не связывают с типами отказов.

Однако в данном определении надежности ПС имеется заимствование из терминологии надежности технических устройств - фактор времени. Основной постулат теории надежности технических устройств: зависимость количества отказов элемента от времени его эксплуатации, что абсолютно бессмысленно применительно к понятию надежности программ.

В ПС ошибки возникают при конкретных, вполне определенных сочетаниях исходных данных. Бесчисленные запуски программы при одном или ограниченном количестве наборов исходных данных, где она работоспособна, в течение любого интервала времени не породят ни одного отказа. Более того, надежность программных модулей со временем может только увеличиваться за счет устранения выявляемых ошибок. Теоретически может возникнуть ситуация, когда при тестировании программного модуля ошибки уже не будут обнаруживаться.

Такое положение вещей делает совершенно бессмысленным для программирования понятия наработки на отказ или вероятности отказа за определенное время.

Аксиоматически классическая теория надежности строится на предположении о том, что существует пусть маленькая, но не нулевая вероятность отказа p_0 любого технического устройства. Данная гипотеза подтверждается экспериментально и связана либо с ошибками проектирования, либо эксплуатационным износом устройства. Программы не изнашиваются и не ломаются, следовательно, ненадежность ПС - следствие исключительно ошибок проектирования, внесенных в процессе разработки. В то же время, можно разработать любое количество простых программ, для которых вероятность отказа $p_0 = 0$. Но в этом случае весь аппарат классической теории надежности становится непригодным для практического использования.

Наиболее общее определение *надежности* ПП предлагает Б. Мейер [3], он определяет надежность как *способность программы давать разумные результаты во всех возможных окружениях и, в частности, в аномальных условиях*.

Тогда ненадежность ПС можно трактовать как соотношение мощности множества ошибочных ситуаций $|\Omega_E|$ и мощности множества исходных данных $|\Omega_{In}|$ ($\Omega_E \subset \Omega_{In}$). Применительно к ЭВМ это - соотношение между количеством сочетаний исходных данных программы, когда возникает ошибочная ситуация, к общему числу сочетаний исходных данных. В этом случае ненадежность можно вычислить по формуле $q = |\Omega_E| / |\Omega_{In}|$.

Однако количество сочетаний исходных данных программных модулей (в общем случае) настолько велико, что перебрать все сочетания для современной ЭВМ практически невозможно. Для сравнительно простых программных модулей, имеющих несколько входных параметров, в настоящее время получены положительные результаты. Для

некоторых модулей удается доказать их корректность [3], т.е. $q=0$. В других случаях q можно определить экспериментально [4,5].

Для современных сложных программных средств число исходных данных измеряется сотнями и тысячами переменных, поэтому прямые методы тестирования становятся неэффективными и практически нереализуемыми. Кажется естественным оценивать надежность сложных ПС, используя известные характеристики составляющих его компонент так же, как это делается для технических систем. Однако и здесь обнаруживаются неожиданности.

Во-первых, как бы ни была сложна структура ПС, она всегда имеет (в смысле надежности) последовательную схему соединения элементов (программных модулей ПС).

Во-вторых, надежность ПС зависит не только от надежности составляющих ее программных модулей, но и корректности организации логики функционирования ПС [6].

В-третьих, надежность ПС не может быть вычислена непосредственно, исходя из надежностей составляющих ее элементов (программных модулей).

Более подробно разберем последний случай.

2. Проблемы оценки надежности сложной программной системы

Предположим, что нам известны характеристики надежности всех программных модулей, из которых составлено ПС. Пусть q_i - ненадежность i-го модуля $A_i : X_i^{In} \to Y_i^{Out}$, вычисленная с учетом всевозможных применений модуля, т.е. на достаточно широкой области значений исходных данных $X_i^{In} = (x_{1i}^{In}, x_{2i}^{In}, ..., x_{n_i i}^{In}) \in \Omega_{In}^i$, где Ω_{In}^i - область значений вектора исходных данных i-го модуля.

Допустим, что логические ошибки в организации логики функционирования ПС отсутствуют (данная проблема требует отдельного рассмотрения см. [4]).

Пусть некоторое ПС имеет L маршрутов ее исполнения [6]. Каждый из маршрутов обозначим $M_j = i_{j_0} i_{j_1} ... i_{j_k}$, где i_{j_k} - номер программного модуля ПС.

Пусть каждый i-й объект на j-м маршруте вызывается в среднем $\tilde{m}_{ij}$ раз.

Ненадежность i-го модуля на j-м маршруте за $\tilde{m}_{ij}$ вызовов можно оценить величиной:

$$Q(\tilde{m}_{ij}) = 1 - (1 - q_i)^{\tilde{m}_{ij}} = Q_{ij}. \qquad (1)$$

Очевидно, что, если в любом из модулей маршрута возникнет ошибка, то весь маршрут можно считать ошибочным. Тогда ненадежность маршрута можно оценить величиной:

$$Q_{M_j} = 1 - \prod_{i=1}^{j_k} (1 - Q_{ij}). \qquad (2)$$

Пусть каждый из маршрутов реализуется с вероятностью r_j, причем справедливо $\sum r_j = 1$. Тогда ненадежность ПС можно оценить величиной:

$$Q_{ПС} = \sum_{j=1}^{L} r_j Q_{M_j}. \qquad (3)$$

Формула (3) справедлива, если оценки ненадежности программных модулей неизменны при различных использованиях модулей, но это не так.

В качестве примера рассмотрим программу термогазодинамического расчета двухвального ТРД (см. рис.1).

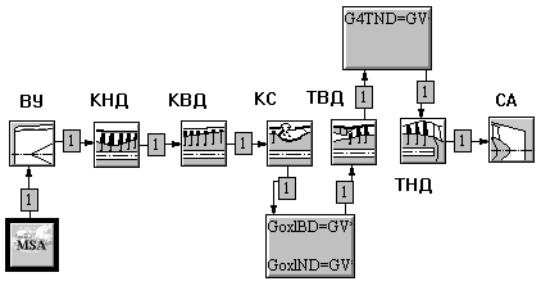

Рис. 1. Программа термогазодинамического расчета двухвального ТРД

Представленная программа достаточно проста и имеет единственный маршрут. Однако при ее выполнении за счет функциональных преобразований изменяются области изменения исходных данных комплектующих ее модулей.

На рисунках 2а-2е показаны области изменения исходных данных ПС расчета ТРД и его узлов (компрессора высокого давления, камеры сгорания, турбин высокого и низкого давления и соплового аппарата).

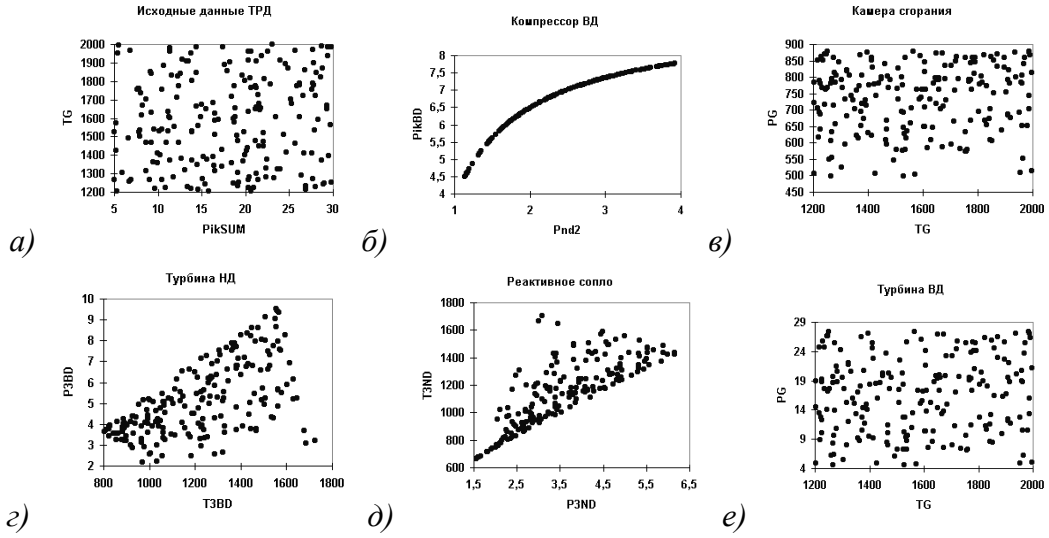

Рис. 2. Области изменения исходных данных ПС расчета ТРД и его узлов

Как видно из рисунков, изменилась топология областей исходных данных модулей, особенно для турбины низкого давления и соплового аппарата, а для компрессора низкого давления область исходных данных выродилась в кривую. Первоначально, при тестировании области исходных данных имели прямоугольный вид. Кроме того, изменились (уменьшились) размеры областей исходных данных. Все это способно увеличить характеристику ненадежности программных модулей за счет уменьшения знаменателя формулы $q_i = V(\Omega_E^i)/V(\Omega_{In}^i)$. Но еще большую проблему представляет изменение законов распределения исходных данных модулей за счет соответствующих функциональных преобразований первоначально равномерного распределения исходных данных ПС. На рис.3 показаны гистограммы распределения входных параметров P3ND, T3ND (входные для соплового аппарата) и P3BD (входного для модуля расчета турбины низкого давления).

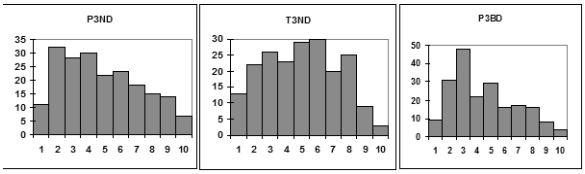

Рис. 3. Распределения входных параметров P3ND, T3ND и P3BD

Как видно из рисунков, эти параметры распределены неравномерно, что вызывает необходимость использования вероятностных мер при оценке ненадежности программных модулей. Рассмотрим влияние перечисленных факторов на изменение в оценке надежности программных модулей.

3. Метод оценки надежности сложной программной системы

Пусть некоторый модуль, имеющий единственных входной параметр x, «прошел» серию испытаний (из N испытаний), в которых не было обнаружено ни одной ошибки, причем $x \in [a,b]$. Из чего можно сделать вывод, что $q \le 1/(N+1)$ и $V(\Omega_E) \approx (b-a)/(N+1)$. Пусть теперь при использовании модуля в ПС изменился диапазон значений входного параметра модуля $x \in [c,d] \subset [a,b]$ без изменения его закона распределения. Итоговую ненадежность модуля, с учетом вероятности $P\{\Omega_E \subset [c,d]\}$ можно оценить следующим образом: $\tilde{q} = \frac{V(\Omega_E)}{d-c} P\{\Omega_E \in [c,d]\} = \frac{V(\Omega_E)}{d-c} \cdot \frac{d-c}{b-a} = \frac{V(\Omega_E)}{b-a} = q,$ т.е. для равномерного закона распределения параметра x характеристика ненадежности модуля не изменяет своего значения.

Для простоты положим $\mathrm{a} = \mathrm{c} = 0$. Пусть теперь на отрезке $[0, \mathrm{d}]$ параметр x имеет квазиэкспоненциальное распределение с функцией плотности распределения $f(x) = K\lambda e^{-\lambda x}$, где $K = \frac{1}{1-e^{-\lambda d}}$. Положим $V(\Omega_E) = \Delta$. Ненадежность модуля можно определить из выражения

$$\tilde{q}(x) = P\{\Delta \in [0,d]\} \int_x^{x+\Delta} K\lambda e^{-\lambda x} dx = \frac{d}{b} \frac{e^{-\lambda x}}{1-e^{-\lambda d}} (1-e^{-\lambda \Delta}).$$

Ненадежность модуля теперь зависит от того, куда попадет область ошибочных ситуаций и не равна полученной в результате автономного тестирования величине q (рис. 4).

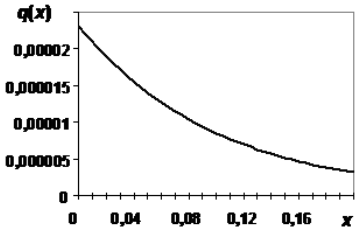

Рис. 4. Зависимость ненадежности модуля от параметра x

Что же тогда считать характеристикой ненадежности модуля?

В предположении равновероятности попадания области ошибочных ситуаций в любое место отрезка [0, d] вычислим среднюю величину $\tilde{q}(x)$

$$q_{c\,\mathrm{p}} = \int_0^d \tilde{q}(x) \frac{1}{d} dx = \frac{P\{\Delta \in [0,d]\} K (1-e^{-\lambda\Delta})}{\lambda d} \int_0^d \lambda e^{-\lambda x} dx =$$

$$= \frac{d}{b} \frac{(1-e^{-\lambda\Delta})K}{\lambda d} (1-e^{-\lambda d}) = \frac{(1-e^{-\lambda\Delta})}{\lambda b} \qquad (4)$$

Введенная характеристика не зависит от параметра x и определяется только размерами области ошибок (Δ) и параметром экспоненциального закона распределения (λ). Как показывают расчеты (см. таблицу) существенные изменения значений λ (степени неравномерности распределения) практически не сказываются на величине q_{cp}. Более того, $q_{cp} \approx q$. В данном примере, $q = 0.00001, \mathrm{d} = 0.2, \ \mathrm{b} = 1$. Для больших неравномерностей $\lambda > 10$ условие $q(x) > q_{c\mathrm{p}}$ наблюдается на сравнительно небольших участках области определения

входного параметра $x-\Delta x$, а в остальных случаях отмечается $q(x) \le q_{ср}$. При небольших значениях λ справедливо $\max q(x) \approx q_{ср}$.

Табл. Влияние неравномерности закона распределения исходных данных на $q_{ср}$

λ	$q_{ср}$	max q(x)	Δx
0,1	9,99999E-06	1,01E-05	0,1
1	9,99995E-06	1,1E-05	0,0975
10	9,9995E-06	2,31E-05	0,0825
100	9,995E-06	0,0002	0,0275
1000	9,95017E-06	0,00199	0,005

Если $q_{ср} \approx q$ справедливо для любых законов распределения, то оценки q_i можно использовать для вычисления надежности ПС, интерпретируя $Q_{ПС}$ как среднюю ожидаемую величину ненадедежности программного средства. Однако данный результат необходимо еще обобщить на многомерный случай.

Для произвольного закона распределения $F(x)$, имеющего плотность распределения $\varphi(x)$ можно выявить следующие общие закономерности.

Определим условие нормировки из уравнения:

$$\int_0^d K\varphi(x)dx = 1 \text{ или } K = 1/\int_0^d \varphi(x)dx .$$

Далее

$$\tilde{q}(x) = P\{\Delta \in [0,d]\} \int_x^{x+\Delta} K\varphi(x)dx =$$
$$= KP\{\Delta \in [0,d]\}(F(x+\Delta) - F(x)).$$

При $\Delta \to 0$ имеем

$$\lim_{\Delta\to 0} \tilde{q}(x) = K\Delta P\{\Delta \in [0,d]\} \lim_{\Delta\to 0} \frac{(F(x+\Delta) - F(x))}{\Delta} =$$
$$= K\Delta P\{\Delta \in [0,d]\}\varphi(x).$$

Тогда

$$q_{ср} = \int_0^d \tilde{q}(x)\frac{1}{d}dx = \frac{\Delta P\{\Delta \in [0,d]\}K}{d}\int_0^d \varphi(x)dx = \quad (5)$$
$$= \frac{d}{b}\frac{\Delta}{d} = \frac{\Delta}{b} = q.$$

Т.е. осредненная вероятность совпадает в пределе с характеристикой ненадежности модуля системы q (при высоких показателях надежности).

Заключение

Таким образом, оценивание надежности ПС существенно отличается от решения аналогичной задачи для технических устройств и требует разработки соответствующей теоретической базы.

Литература

1. Авакян А.А. Концепция построения высоконадежных вычислителей для авиационной и ракетной техники / Авакян А.А., Искандаров Р.Д., Новиков Н.Н. и д.р. //Надежность и качество 2001: Сб. докладов межд. Симпоз Пенза, 2001, с.33-37.
2. Майерс Г. Надежность программного обеспечения. - М.: Мир, 1980. - 360 с.

3. Мейер Б., Бодуэн К. Методы программирования: В 2-х томах. Т.2.- М.: Мир, 1982.- 368 с.
4. Коварцев А.Н. Автоматизация разработки и тестирования программных средств.- Самар. гос. аэрокосм. ун-т. Самара, 1999. - 150 с.
5. Коварцев А.Н. Автоматизация тестирования вычислительных модулей //Надежность и качество 2001: Сб. докладов межд. симпоз Пенза, 2001, с.285-288.
6. Липаев В.В. Надежность программных средств. - М.: СИНТЕГ, 1998. - 232 с.