Тема 3.6. Оценка надежности программных средств

Вопросы надежности программного обеспечения имеют высокую значимость как для разработчиков программ, так и для непосредственных пользователей.

Модели надежности приобрели самостоятельное значение и развитие. Для моделей используют собственную классификацию, в которой учитывают их специфические особенности.

В зависимости от назначения модели надежности делят на прогнозные, оценочные и измерительные, а методически – на математические и эмпирические.

Математические модели, в свою очередь, делятся на вероятностные и статистические, а вероятностные – на экспоненциальные и ба́йесовские.

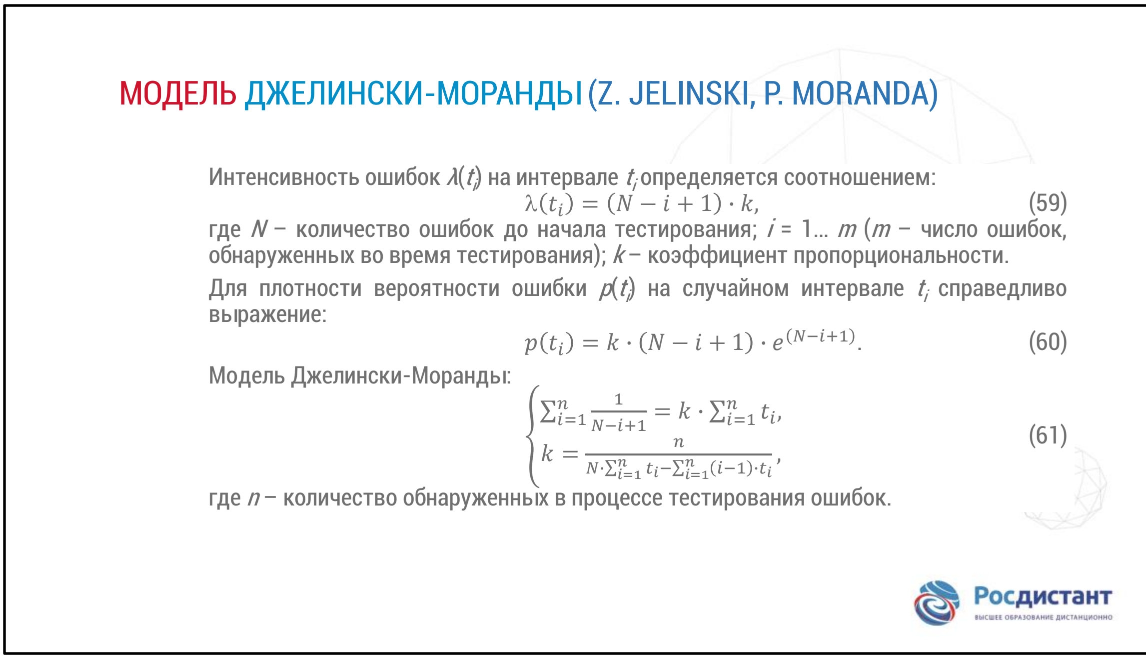
Прогнозные модели используются для определения ожидаемого значения показателей надежности программного средства на этапе его проектирования.

Оценочные модели используются для определения значений показателей надежности на основе анализа результатов тестирования программного средства.

По результатам оценки можно не только судить о надежности программного обеспечения, но и принимать решение о возможности окончания или необходимости продолжения тестирования.

Измерительные модели используются для оценки надежности в фазе испытаний, на этапе сопровождения или эксплуатации. Это возможно при условии, что такие измерения предусмотрены программами испытаний, планами сопровождения и эксплуатации.

Классическими примерами оценочных моделей являются экспоненциальная модель надежности Джели́нски – Мора́нды, статистическая модель Миллса, эвристическая модель.



Модель Джелински – Моранды предложена в 1996 году. «Она основана на предположении об экспоненциальной зависимости плотности вероятности интервалов времени между проявлением ошибок от интенсивности ошибок» [14].

Если допустить, что ошибка после ее каждого появления устраняется и при этом в программный модуль не вносятся новые, то интенсивность ошибок на интервале определяется соотношением 59. Для плотности вероятности ошибки на случайном интервале справедливо выражение 60.

Уравнения, образующие систему 61, представляют модель Джелински – Моранды. Она позволяет оценить количество ошибок в программе до начала тестирования по количеству обнаруженных ошибок.

Продолжительность интервала тестирования измеряется не в единицах времени, а количеством тестов, которое потребовалось для обнаружения очередной ошибки.

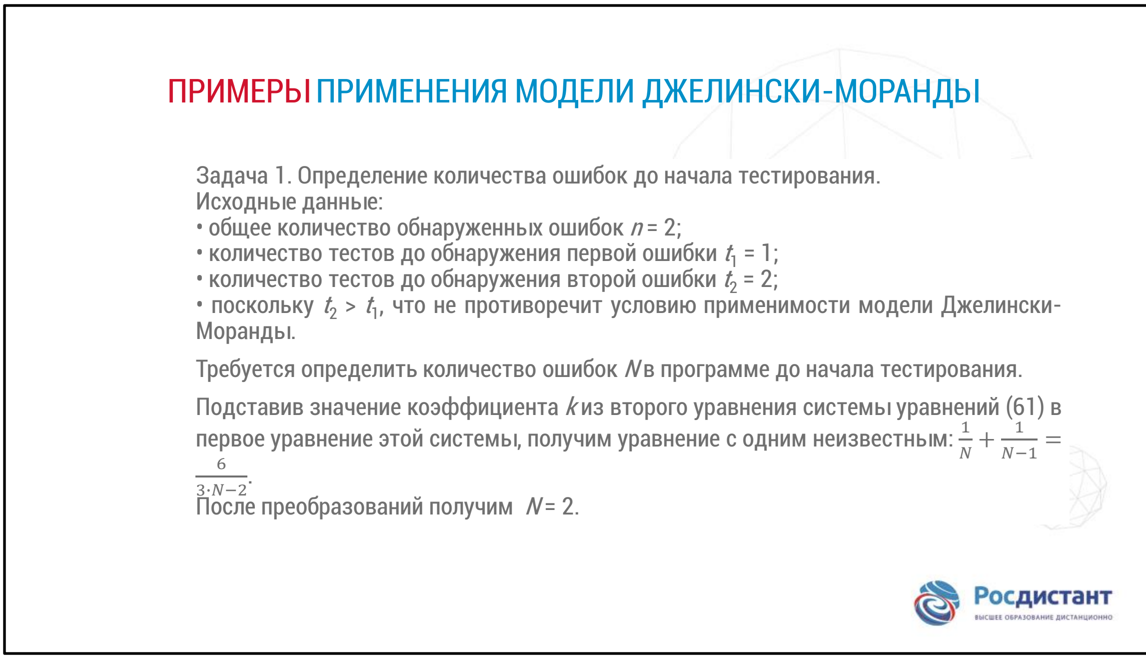
Итак, «модель Джелински – Моранды основывается на соблюдении следующих условий:

– имеет место экспоненциальная зависимость плотности вероятности интервалов времени между появлением ошибок;
– интенсивность ошибок линейно зависит от

количества оставшихся ошибок на любом случайном интервале;

- каждый тест находит только одну ошибку;
- после каждого появления ошибка устраняется и не вносится новая ошибка» [14].

Модель Джелински – Моранды называют моделью роста надежности. Это обусловлено тем, что с течением времени интенсивность ошибок уменьшается и возрастает интервал между проявлением ошибок.



Рассмотрим первую задачу на определение количества ошибок до начала тестирования, используя модель Джелински – Моранды.

Допустим, что в результате тестирования серией из трех тестов обнаружены две ошибки, причем проявились они в первом и третьем тестах.

Требуется определить, сколько ошибок содержала программа до начала тестирования.

Проведем предварительный анализ исходных данных, из которых нам известно следующее:

- общее количество обнаруженных ошибок – 2;
- количество тестов до обнаружения первой ошибки – 1: в первом тесте выявлена ошибка;
- количество тестов до обнаружения второй ошибки – 2: после первого теста во втором ошибки не выявлялись, ошибка проявила себя только в третьем тесте;
- интервал обнаружения второй ошибки больше интервала обнаружения первой ошибки, что не противоречит условию применимости модели Джелински – Моранды.

Подставив значение коэффициента k [ка] из второго уравнения системы уравнений 61 в первое уравнение этой системы, получим уравнение с одним неизвестным. После преобразований найдем значение переменной N [эн], равное двум.

Таким образом, на основе применения модели Джелински – Моранды можно утверждать, что программа до начала тестирования содержала две ошибки.

Тогда, учитывая, что в процессе тестирования программы обнаружены две ошибки, есть основание предполагать, что все ошибки обнаружены и тестирование можно закончить.

ПРИМЕРЫ ПРИМЕНЕНИЯ МОДЕЛИ ДЖЕЛИНСКИ-МОРАНДЫ

Задача 2. Определение количества ошибок в программе, не устраненных после проведения тестирования.

Исходные данные:

- общее количество обнаруженных ошибок $n = 2$;
- интервал продолжительности обнаружения первой ошибки $t_1 = 1$;
- интервал продолжительности обнаружения второй ошибки $t_2 = 3$;
- $t_2 > t_1$, что не противоречит условию применимости данной модели.

Требуется оценить количество оставшихся в программе ошибок.

Подставив значение коэффициента k из второго уравнения в первое уравнение системы (61), получим уравнение с одним неизвестным: $\frac{1}{N} + \frac{1}{N-1} = \frac{8}{4 \cdot N-3}$, $N = 1{,}5 \approx 2$.

В программе осталось $N - n = 0$ необнаруженных ошибок.

Росдистант
высшее образование дистанционно

Рассмотрим вторую задачу на определение количества ошибок в программе, не устраненных после проведения тестирования.

«В результате тестирования программы серией из четырех случайно выбранных из набора тестов обнаружено две ошибки. Ошибки обнаружены первым и четвертым тестами. Все ошибки исправлены сразу после обнаружения.

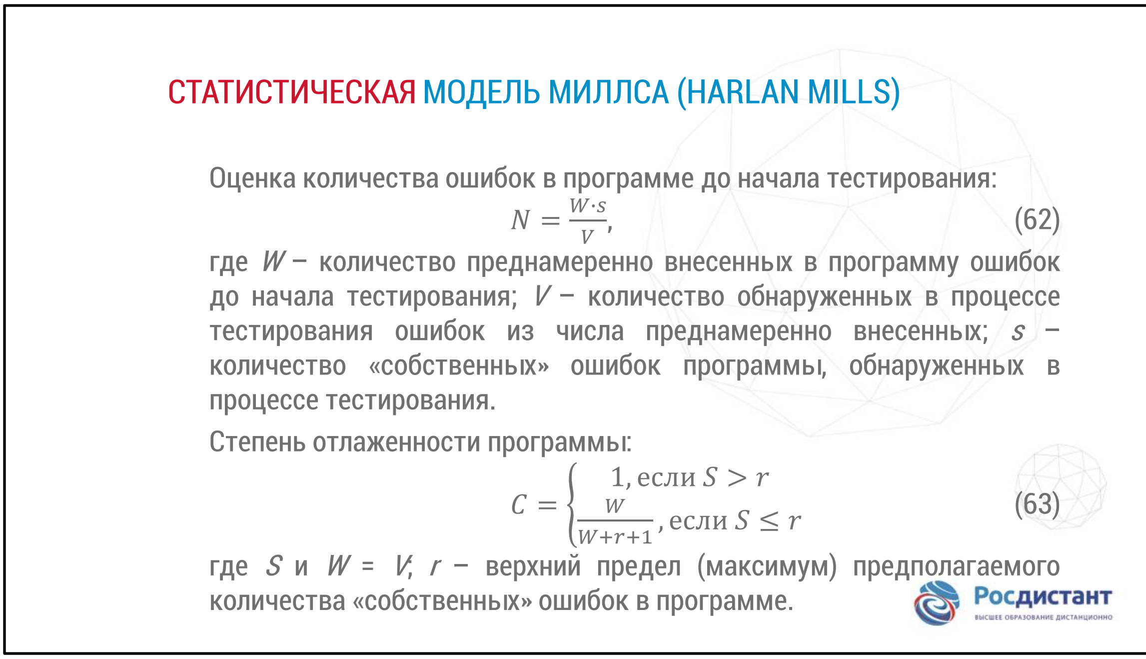
В предположении, что исправление ошибок не повлекло появления новых, требуется оценить количество оставшихся в программе ошибок.

Проанализируем исходные данные поставленной задачи в соответствии с моделью Джелински – Моранды.

- общее количество обнаруженных ошибок равно 2;
- интервал продолжительности обнаружения первой равен 1, так как ошибка обнаружена при проведении одного, причем первого теста;
- интервал продолжительности обнаружения второй ошибки равен 3, так как ошибка обнаружена при проведении четвертого теста;
- интервал обнаружения второй ошибки больше интервала обнаружения первой ошибки, что не противоречит условию применимости модели Джелински – Моранды» [14].

Подставив значение коэффициента *k* [ка] из второго уравнения системы уравнений 61 в первое уравнение этой системы, получим уравнение с одним неизвестным. Полученное уравнение необходимо решить относительно переменной N [эн].

Таким образом, в соответствии с моделью Джелински – Моранды до начала тестирования в программе содержалось две ошибки. В процессе тестирования были обнаружены две ошибки. Следовательно, в программе не осталось необнаруженных ошибок.



Статистическая модель Миллса позволяет оценить количество ошибок до начала тестирования и степень отлаженности программ. Модель предложена программистом компании IBM [ай би эм] Ха́рланом Миллсом в 1972 году.

До начала тестирования в программу преднамеренно вносятся ошибки. Затем обнаружение преднамеренно внесенных и собственных ошибок программы принимается равновероятным. Тестирование продолжается до тех пор, пока все ошибки из числа преднамеренно внесенных не будут обнаружены.

Выражения 62, 63 для оценки количества ошибок в программе до начала тестирования и степени отлаженности программы представляют собой статистическую модель Миллса. К достоинствам статистической модели Миллса можно отнести следующие положения:

– модель Миллса математически проста и интуитивно понятна;
– данная модель может оказывать влияние на группу тестирования. При обнаружении не всех внесенных ошибок программист будет уверен, что программа еще содержит «собственные» ошибки и продолжит тестирование до выявления всех внесенных ошибок.

Однако у модели Миллса есть и недостаток, затрудняющий ее применение. Речь идет о процессе внесения ошибок. Предполагается, что собственные и внесенные ошибки обнаруживаются с одинаковой вероятностью, в то время как она неизвестна.

Из этого следует, что внесенные ошибки должны быть типичными для конкретной программы. Однако сложность и состоит в том, что неизвестно, какой должна быть типичная ошибка для данной программы. Дело в том, что тестирование, как правило, проводит не тот человек, который эту программу разрабатывал.

ПРИМЕР ПРИМЕНЕНИЯ МОДЕЛИ МИЛЛСА

Исходные данные поставленной задачи:
- число преднамеренно внесенных в программу ошибок $W = 10$;
- всего обнаружено семь ошибок;
- количество обнаруженных ошибок из числа внесенных $V = 5$;
- количество «собственных» ошибок в программе $s = 7 - 5 = 2$.

Получим оценку количества ошибок в программе: $N = \frac{10 \cdot 2}{5} = 4.$

Известно, что:
- количество обнаруженных «собственных» ошибок в программе $s = 2$;
- предполагаемое количество ошибок в программе $r = 4$.

Так как $s \leq r$, то степень отлаженности программы:

$$C = \frac{W}{W + r + 1} = \frac{10}{10 + 4 + 1} = 0{,}6(6) \approx 0{,}67 = 67\%$$

Росдистант
высшее образование дистанционно

Рассмотрим задачу на определение количества ошибок до начала тестирования и степени отлаженности программы.

«Допустим, что в программу было преднамеренно внесено десять ошибок. В результате тестирования обнаружено семь ошибок, из которых пять ошибок были внесены преднамеренно. Все обнаруженные ошибки были исправлены. До начала тестирования предполагалось, что программа содержит не более четырех ошибок.

Требуется оценить количество ошибок до начала тестирования и степень отлаженности программы. При этом предполагается, что все преднамеренно внесенные ошибки будут обнаружены, а количество обнаруженных собственных ошибок программы не увеличится» [14].

Проанализируем исходные данные задачи:

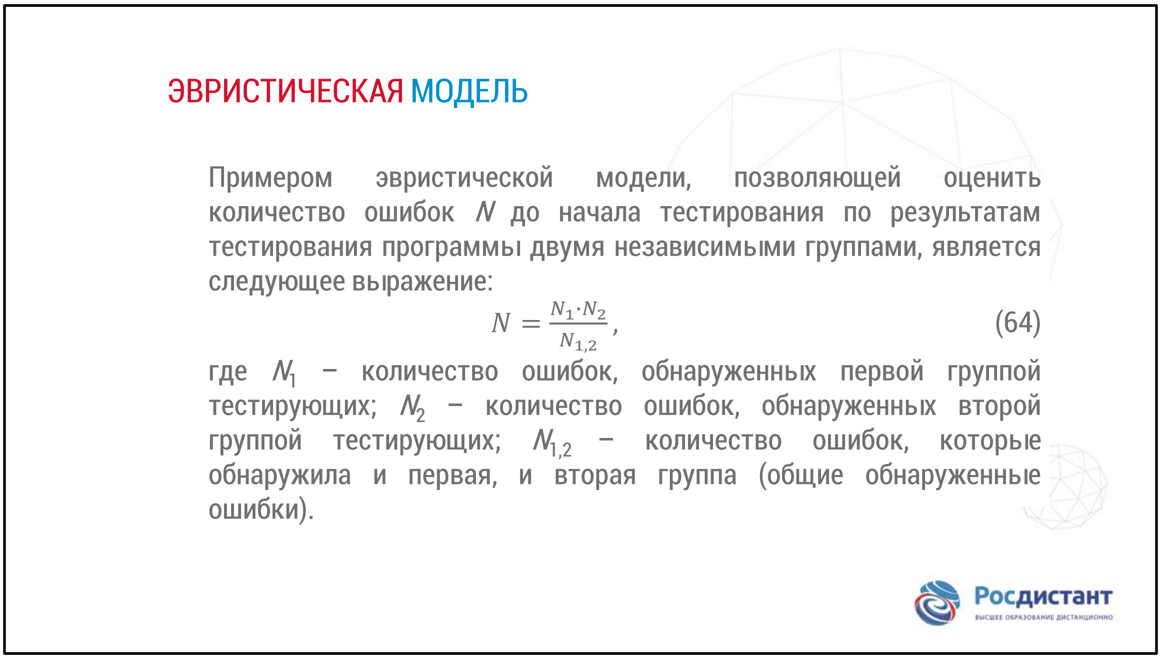
- число преднамеренно внесенных в программу ошибок равно 10;
- всего обнаружено 7 ошибок;
- количество обнаруженных ошибок из числа внесенных равно 5;
- количество собственных ошибок в программе – 2.

Подставив указанные значения в формулу 62, получим оценку количества ошибок в программе до начала тестирования.

Таким образом, из результатов тестирования следует, что до начала тестирования в программе были четыре ошибки. Также известно, что:

- количество обнаруженных собственных ошибок в программе равно 2;
- предполагаемое количество ошибок в программе равно 4;
- количество преднамеренно внесенных и обнаруженных ошибок – 10.

Очевидно, что обнаружено меньшее число собственных ошибок, чем количество предполагаемых ошибок в программе. Из уравнения 63 получим, что степень отлаженности программы составляет 67 %.



Представленное на слайде выражение 64 является примером эвристической модели оценки надежности программных средств. Данная модель «позволяет оценить количество ошибок до начала тестирования по результатам тестирования программы двумя независимыми группами» [14].

Предположим, что в результате тестирования программы силами двух независимых групп первой группой было обнаружено 40 ошибок, а второй группой – 20.

При этом оказалось, что 10 ошибок, обнаруженных первой группой, совпадают с ошибками, обнаруженными второй группой. Совпадение выявленных ошибок является вполне реальным событием при тестировании.

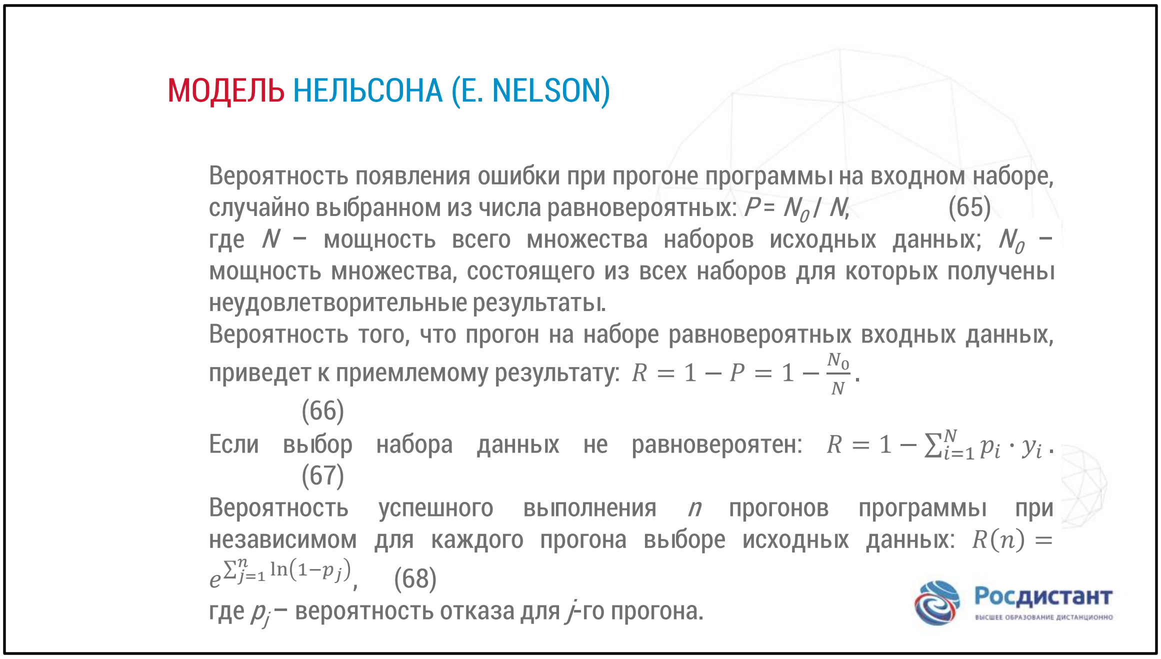
Требуется оценить количество ошибок в программе до начала тестирования.

Из условия задачи нам известны следующие исходные данные:

– «количество ошибок, обнаруженных первой независимой группой тестировщиков, равно 40;
– количество ошибок, обнаруженных второй группой, равно 20;
– количество ошибок, обнаруженных как первой, так и второй группой тестировщиков, равно 10» [14].

Подставив исходные данные в формулу 64, получим оценку количества собственных ошибок в тестируемой программе. Значение составляет 80.

«Эвристическая модель хорошо работает при перекрестном тестировании программ несколькими группами тестировщиков, поскольку обеспечивает достаточно легкую обработку получаемых результатов» [14].



Классической измерительной моделью надежности является модель Не́льсона, предложенная в 1978 году.

«Модель основана на выделении областей исходных данных, покрывающих всё множество вариантов их использования в программе.

Мощность всего множества наборов исходных данных обозначена символом N [эн].

Мощность множества, состоящего из всех наборов исходных данных, для которых получены неудовлетворительные результаты, обозначена как N_0 [эн нулевое].

Совокупность действий, включающих ввод входных данных и выполнение программы, которое заканчивается получением результата или рабочим отказом, считают прогоном программы» [14].

Предполагается, что входные данные, образующие входной набор, необязательно должны подаваться на вход одновременно.

Вероятность появления ошибки при прогоне программы на входном наборе, случайно выбранном из числа равновероятных, определяется отношением мощности N_0 [эн нулевое] к мощности N [эн] по формуле 65.

Тогда по формуле 66 вычисляется вероятность того, что прогон программы на наборе входных данных, случайно выбранном среди равновероятных наборов, приведет к приемлемому результату. Если выбор набора данных не равновероятен, то следует использовать выражение 67.

Также представлена формула 68 для определения вероятности успешного выполнения прогонов программы при независимом для каждого прогона выборе исходных данных.

«Модель Нельсона в наибольшей степени отражает традиционный подход, принятый для определения надежности технических устройств, для измерения надежности программ» [14].

ПРИМЕРЫ ПРИМЕНЕНИЯ МОДЕЛИ НЕЛЬСОНА

Определить надежность программного обеспечения по результатам испытаний.

Задача 1. Дано: $N = 20$, $N_0 = 10$. Решение: $R = 1 - N_0 / N = 1 - 10 / 20 = 0{,}5$.

Задача 2. Исходные данные:

№ теста	Частота выбора теста	Исход прогона теста	№ теста	Частота выбора теста	Исход прогона теста
1	0,04	1	16	0,01	0
2	0,01	0	17	0,02	1
3	0,03	0	18	0,01	0
4	0,05	0	19	0,03	1
5	0,02	1	20	0,19	0
6	0,03	0	21	0,03	1
7	0,05	1	22	0,02	0
8	0,01	0	23	0,04	1
9	0,04	0	24	0,01	1
10	0,01	0	25	0,02	1
11	0,02	1	26	0,01	1
12	0,07	0	27	0,03	1
13	0,01	0	28	0,06	1
14	0,02	1	29	0,02	1
15	0,05	1	30	0,04	1

Решение: По формуле (67) получаем:

$$R = 1 - (0{,}04 \cdot 1 + 0{,}02 \cdot 1 + 0{,}05 \cdot 1 + 0{,}02 \cdot 1 + 0{,}02 \cdot 1 + 0{,}05 \cdot 1 + 0{,}02 \cdot 1 + 0{,}03 \cdot 1 + 0{,}03 \cdot 1 + 0{,}04 \cdot 1 + 0{,}01 \cdot 1 + 0{,}02 \cdot 1 + 0{,}01 \cdot 1 + 0{,}03 \cdot 1 + 0{,}06 \cdot 1 + 0{,}02 \cdot 1 + 0{,}04 \cdot 1 = 0{,}49.$$

Рассмотрим примеры использования модели Нельсона для определения надежности программного средства.

«Условие первой задачи. Для испытания программы использовалось 20 наборов исходных данных, которые равновероятно выбирались для прогона 20 тестов. При этом 10 тестов обнаружили дефекты программного обеспечения. Требуется провести расчет надежности программного обеспечения по результатам испытаний» [14].

Для определения оценки надежности используется формула 66. Получим значение вероятности, составляющее 0,5 [ноль целых пять десятых].

«Условие второй задачи. Для испытания программы использовалось 30 наборов исходных данных. Они выбирались в соответствии с функцией распределения частот, значения которой представлены в таблице.

В 17 тестах были обнаружены ошибки. Все исходы прогонов, закончившиеся отказом, в таблице обозначены единицами. Требуется определить надежность программы по результатам испытаний» [14].

Если набор данных для тестирования программы не равновероятен, то для оценки надежности программы используют соотношение 67.

В результате расчета вероятность события, что прогон программы на заданном наборе исходных данных не приведет к рабочему отказу, составляет 0,49 [ноль целых сорок девять сотых].

Достоинства модели Нельсона достаточно заметны, и рассмотренные примеры хорошо их демонстрируют. Расчеты просты, а результат не требует больших трудозатрат.