# 3 РАСЧЕТ НАДЕЖНОСТИ ПРОГРАММНОГО СРЕДСТВА

## 3.1 Расчет надежности программного средства по модели сложности

Модель сложности программного обеспечения основывается на предположении, что уровень безошибочности продукта может быть предсказан с использованием метрик сложности. Это справедливо для непреднамеренных уязвимостей, так как, чем сложнее и больше программа, тем выше вероятность ошибок при её написании и модификации. Для расчета надежности программного средства по модели сложности используются метрики размера, сложности потока управления, сложности потока данных, а также объектно-ориентированные метрики.

К метрикам размера относят объём (*V*) и потенциальный объём (*V\**) программного средства.

Объём программного средства *V* рассчитывают по формуле (3.1):

|  |  |
| --- | --- |
|  | (3.1) |

где:

– n1 =15 (число уникальных операторов);

– n2 =30 (число уникальных операндов);

– N1 =170 (общее число операторов);

– N2 =85 (общее число операндов).

Потенциальный объем программы определяется по формуле 3.2:

(3.2)

,

где *n*\* – теоретический словарь программы (словарный запас, необходимый для написания программы с учетом того, что необходимая функция уже реализована в языке программирования). Теоретический словарь программы равен сумме количества операторов и операндов программы, при условии того, что нужная функциональность уже реализована в программе. Для разрабатываемого программного средства *n*\*= 136.

Для разрабатываемого программного средства значения объема и потенциального объема равны:

.

Для определения метрик сложности потока управления используются следующие характеристики:

– CL – абсолютная сложность программы, характеризующаяся количеством операторов условия;

– cl – относительная сложность программы, определяемая как отношение CL к общему числу операторов;

– CLI – максимальный уровень вложенности операторов условия.

Для разрабатываемого программного средства значения этих характеристик равны: CL =20, cl = 26, CLI = 42.

Метрика сложности потока данных, предложенная Чепиным, оценивает информационную прочность программного модуля на основе анализа использования переменных из списка ввода-вывода. Все переменные делятся на четыре группы:

– P (вводимые переменные для расчетов и обеспечения вывода);

– M (модифицируемые или создаваемые внутри программы переменные);

– C (переменные, участвующие в управлении работой программного модуля);

– T (не используемые в программе переменные).

Метрика Чепина рассчитывается по формуле 3.3:

(3.3)

Для разрабатываемого программного средства значение метрики Чепина равно:

Теоретическая длина программы N рассчитывается по формуле 3.4:

(3.4)

где n1 – словарь операторов, n2– словарь операндов.

Для разрабатываемого программного средства = 205.

Уровень качества программирования L определяется как отношение потенциального объема к фактическому по формуле 3.5:

(3.5)

Для разрабатываемого программного средства L = 0,69.

Аппроксимированный уровень качества программирования L^ рассчитывается по формуле 3.6:

(3.6)

​

Для разрабатываемого программного средства = 0,05.

Интеллектуальные усилия на разработку программы E рассчитываются по формуле 3.7:

Для разрабатываемого программного средства E = 2035,8

(3.7)

Для оценки объектно-ориентированных характеристик программного средства используются метрики Чидамбера и Кемерера:

– WMC – суммарная сложность всех методов класса;

– DIT – глубина дерева наследования;

– NOC – количество потомков;

– CBO – сцепление между классами;

– RFC – отклик для класса;

– LCOM – недостаток сцепления методов.

Для разрабатываемого программного средства значения этих метрик равны: WMC = 201, DIT = 2, NOC = 3, CBO = 7, RFC = 20, LCOM = 20.

Результаты расчетов представлены в таблице 3.1.

Таблица 3.1 – Результаты расчетов

|  |  |  |  |
| --- | --- | --- | --- |
| Метрики ПС | amin | amax | ai |
| V | 108 | 1364 | 1400,4 |
| V\* | 540 | 83362,41 | 963,9 |
| CL | 7 | 368 | 20 |
| Cl | 25 | 1278 | 26 |
| CLI | 41 | 4213 | 42 |
| Q | 67 | 3589 | 170 |
| N^ | 6 | 186 | 205 |
| L | 0,0054 | 2 | 0,69 |
| L^ | 0 | 3 | 0,05 |
| E | 29 | 967 | 2035,83 |
| WMC | 13 | 299 | 201 |
| DIT | 1 | 8 | 2 |
| NOC | 1 | 32 | 3 |
| CBO | 1 | 27 | 7 |
| RFC | 1 | 163 | 20 |
| LCOM | –39 | 387 | 20 |

Исходя из полученных метрик рассчитаны нормативный и фактический уровни каждой метрики (формулы (3.8) и (3.9) соответственно):

(3.8)

(3.9)

По полученным результатам определен дискриминант каждой метрики, используя формулу (3.10):

(3.10)

Результаты вычислений по формулам (3.8 – 3.10) представлены в таблице 3.2.

Таблица 4.2 – Результаты вычисления

|  |  |  |  |
| --- | --- | --- | --- |
| Метрики ПС | xmin | xфi | di |
| V | 0,07917888563 | 1,026686217 | -0,00223503021 |
| V\* | 0,006477739787 | 0,01156276552 | 0,5573567619 |
| CL | 0,01902173913 | 0,05434782609 | 0,3373961219 |
| Cl | 0,01956181534 | 0,02034428795 | 0,9607710725 |
| CLI | 0,009731782578 | 0,009969143128 | 0,97595649 |
| Q | 0,01866815269 | 0,04736695458 | 0,3825917761 |
| N^ | 0,03225806452 | 1,102150538 | -0,003089430894 |
| L | 0,0027 | 0,345 | 0,005139964862 |
| L^ | 0 | 0,01666666667 | 0 |
| E | 0,02998965874 | 2,105305067 | -0,01623163562 |
| WMC | 0,04347826087 | 0,6722408027 | 0,02216191768 |
| DIT | 0,125 | 0,25 | 0,4285714286 |
| NOC | 0,03125 | 0,09375 | 0,311827957 |
| CBO | 0,03703703704 | 0,2592592593 | 0,1098901099 |
| RFC | 0,006134969325 | 0,1226993865 | 0,04413580247 |
| LCOM | -0,1007751938 | 0,05167958656 | -1,679929577 |

Риск снижения надежности рассчитывается по формуле 3.11:

(3.11)

Где λi – весовые коэффициенты для каждой метрики. Для упрощения расчетов можно считать, что все метрики вносят равный вклад в надежность, т.е. λi = 1/16.

Для разрабатываемого программного средства вероятность безотказной работы равна 3.12:

(3.12)

Таким образом, вероятность безотказной работы программного средства по модели сложности составляет приблизительно 0,6.

## 3**.2 Расчет надежности программного средства по модели Джелинского – Моранды**

Модель Джелинского – Моранды используется для оценки надежности программного обеспечения на основе данных о времени между обнаружением ошибок. Она предполагает, что интенсивность обнаружения ошибок зависит от текущего количества оставшихся ошибок в программе. В данном разделе проведем расчет надежности программного средства на основе модели Джелинского – Моранды. Общее время тестирования программы составило 15 дней. За это время было обнаружено 20 ошибок. Время между обнаружением ошибок (Xi​) приведено в таблице 3.3.

Таблица 3.3 – Время между ошибками (Xi​) в часах:

|  |  |
| --- | --- |
| Номер ошибки (i) | Время между ошибками (Xi) |
| 1 | 4 |
| 2 | 6 |
| 3 | 5 |
| 4 | 7 |
| 5 | 3 |
| 6 | 8 |
| 7 | 6 |
| 8 | 4 |
| 9 | 5 |
| 10 | 7 |
| 11 | 5 |
| 12 | 6 |
| 13 | 4 |
| 14 | 5 |
| 15 | 7 |
| 16 | 3 |
| 17 | 6 |
| 18 | 4 |
| 19 | 5 |
| 20 | 6 |

Для оценки общего числа ошибок B*B* в программном средстве используется метод, основанный на минимизации разности функций f(m) и g(m,A)*.* Функции f(m) и g(m,A) определяются следующим образом по формулам 4.13 и 4.14:

(3.13)

f(m)=n/m,

(3.14)

g(m, A)=A/m,

где A — константа, зависящая от данных. В данном случае предположим, что A=12> (20 + 1)/2.

Для нахождения B выполняется поиск такого значения m, при котором разность f(m)−g(m,A) минимальна. Результаты расчетов для различных значений m показывают, что минимальная разность достигается при m=26, что соответствует общему числу ошибок B=25 (так как B=m−1).

Коэффициент K*K* рассчитывается по формуле 3.14:

(3.14)

K=nB⋅T−∑i=1ni⋅Xi.

Для расчета суммы ∑i=1ni⋅Xi∑*i*=1*n*​*i*⋅*Xi*​ используются времена между отказами Xi​. Подставляя значения, получаем:

K=2025⋅120−1260=203000−1260=201740≈0.0115.

Интенсивность отказов λ(t)после обнаружения 20 ошибок рассчитывается по формуле:

λ(t)=K⋅(B−20) =0.0115⋅(25−20)=0.0115⋅5=0.0575.

Вероятность безотказной работы P(t)*P*(*t*) на интервале t=1*t*=1 час определяется по формуле 3.15:

(3.15)

P(1)=e−λ(t)⋅t=e−0.0575⋅1≈0.944.

Для времени t=1 час:

P(t)=e−0,1196⋅1≈e−0,1196≈0,887.

Среднее время до появления следующей ошибки рассчитывается по формуле 3.16:

(3.16)

X21=1K⋅(B−20) =10.0115⋅5≈17.39 часов.

Время до окончания тестирования оценивается как сумма времени до обнаружения оставшихся ошибок (формула 3.17):

(3.17)

Предположив, что ошибки будут находиться через среднее значение времени, получим t = 90 часов.

Таким образом, модель Джелинского – Моранды позволяет оценить надежность программного средства и спрогнозировать время, необходимое для обнаружения и исправления всех ошибок.

## 4.3 Расчет по модели Муса

Проведем расчет надежности по модели Муса. В этой модели надежность программного средства на этапе эксплуатации оценивается по результатам тестирования. Пусть Т – суммарное время тестирования, М – число отказов, произошедших за время тестирования. Тогда по модели Муса средняя наработка до отказа после тестирования определяется по формуле (3.18).

|  |  |
| --- | --- |
|  | (3.18) |

В этой формуле τ0 – средняя наработка до отказа до начала тестирования, С – коэффициент, учитывающий уплотнение тестового времени по сравнению с временем реальной эксплуатации. Например, если один час тестирования соответствует 12 часам работы в реальных условиях, то С = 12. Неизвестный параметр τ0 можно оценить из соотношения (4.19).

|  |  |
| --- | --- |
|  | (3.19) |

В этой формуле N – первоначальное число ошибок в программной средстве, которое можно оценить с помощью другой модели, позволяющей определить N на основе статистических данных, полученных при тестировании. К – коэффициент проявления ошибок, который определяется эмпирическим путем по однотипным программам, обычно это значение изменяется от 1,5·10–7 до 4·10–7. f – средняя скорость исполнения программы (A), деленная на число команд (операторов (B)).

Средняя скорость исполнения одного оператора программного средства определяется по формуле (3.20).

|  |  |
| --- | --- |
|  | (3.20) |

Длительности этапов тестирования составляют t1 = 50 ч, t2 = 60 ч, t3 = 10 ч. Число отказов на первом этапе m1 = 10, на втором – m2 = 8, на третьем – m3 = 2. Средняя скорость исполнения программного средства = 107 операторов/ч, количество операторов в программном средстве = 1050. Определим надежность системы для периода эксплуатации t = 90 ч.

Найдем среднюю скорость выполнения одного оператора по формуле (3.21):

(3.21)

Первоначальное количество ошибок в программном средстве N равно 20. Коэффициент проявления ошибок K примем равным 3·10-7. Найдем формуле (3.22).

(3.22)

Примем значение коэффициента С = 8. Тогда средняя наработка до отказа после тестирования на этапе эксплуатации программного обеспечения рассчитывается по формуле (3.18).

Найдем надежность программного средства для периода эксплуатации t равному 90 часам по формуле (3.23).

|  |  |
| --- | --- |
|  | (3.23) |

Надежность равна: