**СОДЕРЖАНИЕ**

**ВВЕДЕНИЕ**

**1 РАСЧЕТ НАДЕЖНОСТИ**

**1.1 Модель сложности.**

В ходе расчета модели сложности рассчитываются следующие метрики:

* метрики размера: объем программы *V*; потенциальный объем программы ;
* метрики сложности потока управления (по Джиблу): абсолютная сложность программы *CL*; относительная сложность программы *cl*; максимальный уровень вложенности оператора *CLI*;
* метрики сложности потока данных: метрика Чепина *Q*;
* метрики стилистики и понятности программ (по Холстеду): теоретическая длина программы ; метрика корректности программы *L*; метрика корректности реальной программы ; число элементарных решений, принятых при написании программы *E*;
* объектно-ориентированные метрики: суммарная сложность всех методов класса *WMC*; глубина дерева наследований *DIT*; количество потомков *NOC*; сцепление между классами *CBO*; мощность множества классов *RFC*; недостаток сцепления методов *LCOM*.

Метрики размера программного средства (ПС). В данные метрики включают следующее:

* объем программы *V*;
* потенциальный объем программы *V\**.

Объем V определяется по формуле:

|  |  |
| --- | --- |
| , | (1.1) |

где (*n*1 +*n*2) – словарь ПС; *n*1 – число уникальных операторов программы, включая символы-разделители, имена процедур и знаки операций (словарь операторов); *n*2 – число уникальных операндов программы (словарь операндов); (*N*1 + *N*2) – длина ПС *N*; *N*1 – общее число операторов в программе; *N*2 – общее число операндов в программе.

Подставив значения в формулу (1.1) получим выражение

*V* = (5463+7812) ∙ *log*2(4658 + 6647) = 13275 ∙ 13,465 = 178747,875

Потенциальный объем *V*\* рассчитывается по формуле:

|  |  |
| --- | --- |
|  | (1.2) |

где *n*\* – теоретический словарь ПС.

Подставив значения в формулу (1.2) получим выражение

|  |  |
| --- | --- |
|  |  |

Метрики сложности потока управления. Используются следующие метрики предложенные Джиблом:

* абсолютная сложность управления *CL*, характеризующаяся количеством операторов условия;
* относительная сложность *cl*;
* насыщенность ПС операторами условия определяется отношением *CL* к общему числу операторов;
* максимальный уровень вложенности оператора условия *CLI*.

Суть метода состоит в оценке информационной прочности отдельно взятого программного модуля с помощью анализа характера использования переменных из списка ввода-вывода.

Для этого все переменные, составляющие список ввода-вывода, разбивается на четыре функциональные группы:

1. *Р* – вводимые переменные для расчетов и обеспечения вывода.
2. *М* – модифицируемые, или создаваемые внутри программы переменные.
3. *С* – переменные, участвующие в управлении работой программного модуля (управляющие переменные).
4. *Т* – не используемые в программе («паразитные») переменные.

Поскольку каждая переменная может выполнять одновременно несколько функций, необходимо учитывать ее в каждой соответствующей функциональной группе.

Сложность потока данных *Q* определяется выражением:

|  |  |
| --- | --- |
|  | (1.3) |

Подставив значения в формулу (1.3) получим следующее значение:

|  |  |
| --- | --- |
|  |  |

Метрики стилистики и понятности ПС. В данной группе используются три метрики:

* отклонение реальной *N* от теоретической *N*^ длины ПС ∆*N* (в %);
* уровень качества программирования *L* (уровень ПС);
* оценка интеллектуальных усилий на разработку ПС *E*.

Отклонение *N* от *N^* определяется выражением:

|  |  |
| --- | --- |
| , | (1.4) |

где *N*^ определяется выражением:

|  |  |
| --- | --- |
|  | (1.5) |

где *n*1 – словарь операторов; *n*2–словарь операндов ПС.

Таким образом, теоретическая длина *N*^ информационной системы передачи информации составит:

|  |  |
| --- | --- |
| 141161.336 |  |

Отклонение *N* от *N*^ определим по формуле (1.4):

Уровень качества программирования *L* расчитывается по формуле:

|  |  |
| --- | --- |
|  | (1.6) |

Подставив численные значения получим:

|  |
| --- |
|  |

Исходным для введения этой характеристики является предположение о том, что при снижении стилистического качества программирования уменьшается содержательная нагрузка на каждый компонент программы и, как следствие, расширяется объем реализации исходного алгоритма.

Интеллектуальные усилия на разработку ПС *Е* рассчитывается по формуле:

|  |  |
| --- | --- |
| , | (1.7) |

где *V* – объем программы; *V*\* – потенциальный объем программы.

Подставив численные значения в формулу (1.7) получим:

|  |  |
| --- | --- |
|  |  |

Объектно-ориентированные метрики.Рассчитаем метрики Мартина:

* центростремительное сцепление *Ca* – количество классов вне этой категории, которые зависят от классов внутри этой категории;
* центробежное сцепление *Ce* – количество классов внутри этой категории, которые зависят от классов вне этой категории;
* нестабильность *I*.

Нестабильность определяется по формуле:

|  |  |
| --- | --- |
|  | (1.8) |

Подставив численные значения получим:

Далее рассчитаем нормативный *xmin* и фактический *xфi* уровни для каждой из метрик по формулам:

|  |  |
| --- | --- |
| *,* | (1.9) |
|  | (1.10) | |

где *ai* – значение метрики, рассчитанное для конкретного ПС; *amin* – минимально возможное значение этой метрики для данного типа ПС; *amax* – максимально возможное значение этой метрики для данного ПС.

Также рассчитываем дискриминант *di* для каждой метрики по формуле:

|  |  |
| --- | --- |
| , | (1.11) |

Результаты вычислений приведены в таблице 1.1.

Таблица 1.1 – нормативные пределы метрик, результаты вычислений нормативных и фактических уровней метрик, дискриминантов метрик

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Метрики | Значение *ai* | *amin* | *amax* | *xmin* | *xϕi* | *di* |
| *V* | 178747,875 | 31 000 | 240 000 | 0.129167 | 0.744 | 0.0508 |
| *V\** | 49637,716 | 508 | 69 000 | 0.007362 | 0.71 | 0.0028 |
| *Q* | 1070 | 57 | 3425 | 0.016642 | 0.312 | 0.0372 | |
| *ΔN* | 9,4 | 4 | 35 | 0.114286 | 0.2685 | 0.3514 | |
| *L* | 0,277 | 0,003 | 3,1 | 0.000968 | 0.089 | 0.00987 | |
| *E* | 643679,95 | 23000 | 991000 | 0.023209 | 0.6495 | 0.0128 | |
| *Ca* | 23 | 11 | 141 | 0.078014 | 0.1631 | 0.434 | |
| *Ce* | 25 | 11 | 210 | 0.052381 | 0.1190 | 0.409 | |
| *I* | 0,52 | 0 | 2 | 0 | 0.26 | 0 | |

Рассчитываем риск снижения надежности работы программного средства *R* по формуле:

|  |  |
| --- | --- |
|  | (1.12) |

где *λi* – весовые коэффициенты для конкретных метрик. Отражает на сколько та или иная метрика имеет больший вес для надежности ПС и должна удовлетворять условию:

|  |  |
| --- | --- |
|  | (1.13) |

где *N* – количество метрик, используемое при расчете риска снижения надежности.

Для упрощения задачи примем равнозначным вклад каждой метрики в результат расчета риска снижения надежности.

Таким образом, по формуле (1.12) для данного программного продукта риск снижения надежности равен:

Следовательно, надежность равна:

10,056 = 0,9432

**1.2 Модель Шумана**

Использование модели Шумана для оценки надежности ПС предполагает что тестирование проводится в несколько этапов.

Каждый этап представляет собой выполнение программы на полном комплексе разработанных тестовых данных. Выявленные ошибки регистрируются (собирается статистика об ошибках), но не исправляются.

По завершении этапа на основе собранных данных о поведении ПС на очередном этапе тестирования может быть использована модель Шумана для расчета количественных показателей надежности.

После этого исправляются ошибки, обнаруженные на предыдущем этапе, при необходимости корректируются тестовые наборы и проводится новый этап тестирования.

Предполагается, что до начала тестирования в ПС имеется *ЕT* ошибок. В течение времени тестирования *τ* обнаруживается *ЕС* ошибок в расчете на команду в машинном языке.

Функция надежности, или вероятность безотказной работы на интервале времени от 0 до *t*, равна:

|  |  |
| --- | --- |
|  | (1.14) |

Программа содержит 1485 командных строки, из них, до начала эксплуатации (после периода отладки), 10 командных строк содержат ошибки. После 23 дней работы обнаружено 25 ошибки. Необходимо найти среднее время безошибочной работы программы и интенсивность отказов программы при коэффициенте пропорциональности, равном 0,75.

Исходные данные: *I*=1485; *M* = 10; *t* = 20; *C* = 0,75. Предполагается, что до начала тестирования (т.е. в момент *τ* = 0) имеется *M* ошибок. В течение времени тестирования *τ* обнаруживается *ε1(τ)* ошибок в расчете на одну команду в машинном языке, где *ε1(τ)*:

подставив исходные данные получим:

Тогда удельное число ошибок на одну машинную команду, оставшихся в системе после времени тестирования *τ*, определяется выражением:

таким образом:

Тогда, если время работы программы без отказа *t* отсчитывается от точки *t*=0, *а* *τ* остается фиксированным, функция надежности, или вероятность безотказной работы на интервале от 0 до *t*, определяется по формуле:

таким образом:

Время наработки на отказ рассчитывается по формуле:

таким образом:

Тогда по формуле (1.17) получим вероятность безотказной работы на протяжении 213 дней:

Таким образом, вероятность безотказной работы в течение 220 суток составляет 0,37, что в очередной раз доказывает необходимость тестирования и отладки программного средства.

**1.3 Модель Ла Падула**

Модель основывается на выполнении последовательности тестов в *t* этапов. Каждый этап заканчивается внесением изменений (исправлений) в программном средстве. Возрастающая функция надежности базируется на числе ошибок, обнаруженных в ходе каждого тестового прогона.

Известно:

* число этапов m = 3;
* число тестов на первом этапе – 30, число отказов – 14;
* число тестов на во втором этапе – 36, число отказов – 11;
* число тестов на в третьем этапе – 41, число отказов – 5.

Надежность программного средства на *i*-м этапе определяется по формуле:

|  |  |
| --- | --- |
|  | (1.19) |

где *А* – параметр роста; *Р*(∞) – предельная надежность программного средства, равняется 1.

Параметр роста вычисляется путем решения следующего уравнения:

|  |  |
| --- | --- |
|  | (1.20) |

где *Si* – число тестов; *mi* – число отказов во время *i*-го этапа; *t* – число этапов; *i* = 1, 2, ..., *t*; *Si* – число тестов на этапе; *n* – число этапов; *P*(∞)– предельное значение надежности; *Ai* – параметр роста на *i*-ом этапе тестирования.

Таким образом надежность на *i*-м этапе можно определить по формуле:

|  |  |
| --- | --- |
|  | (1.21) |

Используя исходные данные для решения уравнения (1.20), определим *A*1 = 0,46, *A*2 = 0,611 и *A*3 = 0,365

Подставив найденные значения в (1.21) найдем надежность программного средства после каждого этапа тестирования: *P*1 = 0,54, *P*2 = 0,6945, *P*3 = 0, 87.

Таким образом, вероятность безотказной работы после трех этапов составляет 0,87.

**1.4 Модель Джелинского – Моранды**

Модель Джелинского – Моранды строится на основе следующих допущений:

1 Интенсивность обнаружения ошибок λ(t) пропорциональна текущему числу ошибок в программном средстве, т. е. числу оставшихся ошибок.

2 Все ошибки одинаково вероятны, и их появления независимы.

3 Каждая ошибка имеет один и тот же порядок серьезности.

4 Время до следующего отказа (ошибки) распределено экспоненциально.

5 Программное средство функционирует в среде, близкой к реальной.

6 Ошибки постоянно корректируются без внесения в программное средство новых.

7 λ(t) = const в интервале между двумя соседними ошибками.

В соответствии с этими допущениями интенсивность возникновения ошибок в программном средстве можно представить в виде

|  |  |
| --- | --- |
|  | (1.22) |

где *t* – произвольное время между обнаружением (*i*-1) и *i*-й ошибок; *K* – неизвестный коэффициент; *B* – неизвестное общее число ошибок в программном средстве.

А вероятность безотказной работы программного средства рассчитывается согласно выражению

|  |  |
| --- | --- |
|  | (1.23) |

Для нахождения *B* используют выражения

|  |  |
| --- | --- |
|  | (1.24) |

|  |  |
| --- | --- |
|  | (1.25) |

|  |  |
| --- | --- |
|  | (1.26) |

|  |  |
| --- | --- |
|  | (1.27) |

|  |  |
| --- | --- |
|  | (1.28) |

где *m* ≥ *n*+1 – число прогнозируемых ошибок; *n –* количество выявленных ошибок.

Неизвестный коэффициент *K* рассчитывают по формуле

|  |  |
| --- | --- |
|  | (1.29) |

На этапе отладки программного средства за 179 день было выявлено 10 ошибок. Исходные данные сведены в таблицу 1.2 в виде интервалов времени *Хi* (дни) между соседними ошибками (*i* – номер ошибки). Необходимо найти вероятность безотказной работы программного средства.

Таблица 1.2 – Исходные данные для расчетов

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *i* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| *Xi,* день | 2 | 9 | 11 | 14 | 16 | 18 | 22 | 27 | 33 | 39 |

Подставив численные значения в формулу (1.27) получим *A=*7,1712

Осуществим расчеты по формулам (1.25), (1.26) и сведем их в таблицу 1.3.

Таблица 1.3 – Результаты вычислений

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *i* | *Xi* | *i∙Xi* | *m* | *gn*(*m, A*) | *fn*(*m*) | *|fn*(*m*) – *gn*(*m, A*)| |
| 1 | 2 | 2 | 11 | 2.553191 | 2.928968254 | 0.375777 |
| 2 | 9 | 18 | 12 | 2.033898 | 2.019877345 | 0.01402 |
| 3 | 11 | 33 | 13 | 1.690141 | 1.603210678 | 0.08693 |
| 4 | 14 | 56 | 14 | 1.445783 | 1.346800422 | 0.09898 |
| 5 | 16 | 80 | 15 | 1.263158 | 1.168228993 | 0.09493 |
| 6 | 19 | 114 | 16 | 1.121495 | 1.03489566 | 0.0866 |
| 7 | 22 | 154 | 17 | 1.008403 | 0.930728993 | 0.07767 |
| 8 | 27 | 216 | 18 | 0.916031 | 0.84669538 | 0.06934 |
| 9 | 33 | 297 | 19 | 0.839161 | 0.777250935 | 0.06191 |
| 10 | 39 | 390 | 20 | 0.774194 | 0.718771403 | 0.05542 |

Из таблицы 1.3 видно, что наилучшим решением для уравнения (1.24) является *m=*12, в таком случае согласно выражению (1.28) *B* = 14.

Подставив в формулу (1.29) рассчитанные значения получим

Далее согласно с формулой (1.22) рассчитаем интенсивность возникновения ошибок после того, как обнаружена ( *i –* 1) ошибка.

По формуле (1.23) вероятность безотказной работы программного средства *P*(*t*) = 0,94.

**1.5 Модель Шика – Волвертона**

Модификация модели Джелинского-Моранды. В основе модели Шика-Волвертона лежит формула:

|  |  |
| --- | --- |
|  | (1.30) |

где *K* – коэффициент пропорциональности; *B* – число ошибок, первоначально присутствующих в программе; *ti* – промежуток времени между (*i -* 1)-й и *i*-й ошибками

Вероятность безотказной работы определяется выражением:

|  |  |
| --- | --- |
|  | (1.31) |

В основе модели Шика-Волвертона лежит предположение (гипотеза), согласно которому частота ошибок пропорциональна не только количеству ошибок в программах, но и времени тестирования, т.е. вероятность обнаружения ошибок с течением времени возрастает. Частота ошибок (интенсивность обнаружения ошибок) *λi*, предполагается постоянной в течение интервала времени *ti*, и пропорциональна числу ошибок, оставшихся в программе по истечении (*i* - 1)-го интервала; но она пропорциональна также и суммарному времени, уже затраченному на тестирование (включая среднее время выполнения программы в текущем интервале):

|  |  |
| --- | --- |
|  | (1.32) |
|  |  |

Уравнения для определения *K* и *B* имеют следующий вид:

|  |  |
| --- | --- |
|  | (1.33) |

где *Тi-1* – время, накопленное за (*i* - 1) интервалов:

|  |  |
| --- | --- |
| , T0=0, | (1.34) |

где *ni-1* – суммарное число ошибок, обнаруженных за период от первого до (*i* -1)-го интервала времени включительно:

|  |  |
| --- | --- |
| , n0=0, | (1.35) |

где *М*– общее число временных интервалов;

Суммарное число обнаруженных ошибок:

|  |  |
| --- | --- |
|  | (1.36) |

В ходе отладки программного модуля были зафиксированы интервалы времени *t*1= 19, *t*2= 23, *t*3= 28 часов между отказами программы.

Методом подбора было определено первоначальное количество ошибок *N* = 4, откуда по формуле:

Найдем *λ*4по формуле (1.32):

|  |  |
| --- | --- |
|  |  |

Следовательно, среднее время до следующего отказа составляет:

Тогда, подставляя найденные значения *λ4* и *t4* в формулу (1.31), получим вероятность отсутствия четвертого отказа:

*P*(*t*4) = *e* 0,77 (-1,29) = *е*-0,9933 = 0,37

**1.6 Модель Муса**

В модели Муса надежность ПС на этапе эксплуатации оценивается по результатам тестирования.

Средняя наработка до отказа после тестирования определяется по формуле:

|  |  |
| --- | --- |
| , | (1.37) |

где τ0 – средняя наработка до отказа до начала тестирования; *С* – коэффициент, учитывающий уплотнение тестового времени по сравнению с временем реальной эксплуатации.

Неизвестный параметр τ0 можно оценить из следующего соотношения:

|  |  |
| --- | --- |
|  | (1.38) |

Средняя скорость исполнения одного оператора ПС расчитывается по формуле:

|  |  |
| --- | --- |
|  | (1.39) |

Общее время тестирования *Т + ΔТ* должно удовлетворять соотношению:

|  |  |
| --- | --- |
|  | (1.40) |

Надежность ПС для периода эксплуатации *t* определяются по формуле:

|  |  |
| --- | --- |
|  | (1.41) |
|  |  |

Для данного программного модуля длительности этапов тестирования составляют *t1* = 19 часов, *t2* = 23 часов, *t3* = 28 часов. Число отказов на первом этапе *m*1 = 3, на втором – *m*2 = 5, на третьем – *m*3 = 1. Средняя скорость исполнения ПС *A* = 104 операторов/час, количество операторов в ПС *B* = 740. Период эксплуатации *t* = 189 часов.

Найдем среднюю частоту выполнения одного оператора с помощью формулы (1.39):

|  |  |
| --- | --- |
|  |  |

Первоначальное количество ошибок в программном средстве (из модели Шумана) *N* = 4. Коэффициент проявления ошибок *K* для данного программного средства 1,5 ∙ 10-6, в таком случае τ0 можно рассчитать по формуле (1.38)

Один час тестирования соответствует 6 часам работы, следовательно значение коэффициента *С* = 5. Тогда средняя наработка до отказа после тестирования на этапе эксплуатации ПО согласно с формулой (3.17) равна:

Найдем надежность программного средства для периода эксплуатации *t* = 189 часов по формуле (1.41):

**1.7 Модель Переходных вероятностей**

Данный метод связан с анализом систем, которые можно описать с помощью случайных процессов.

Марковский процесс – случайный процесс, протекающий в какой-либо системе , обладающий свойством, что для любого момента времени вероятность любого состояния системы в будущем (при ) зависит только от ее состояния в настоящем (при ) и не зависит от того, как и когда система пришла в это состояние. Одним из основных видов таких процессов является процесс с дискретными состояниями и дискретным временем, называемый цепью Маркова.

Состояние , состоящее в том, что сразу после k-ого шага система находится в состоянии (*i* = 1,2,3…) является случайным событием. Назвать последовательность таких случайных событий марковской цепью можно при условии, если для каждого шага вероятность перехода из любого состояния в любое состояние не зависит от того, как и когда система пришла в состояние . Начальное состояние может быть случайно или задано заранее.

Основные параметры марковской цепи следующие

Вероятностями состояний цепи называют вероятноститого, что после *k*-ого шага и до *k* + 1-го шага система будет находиться в состоянии (i = 1,2,3…n). Для любого будет выполняться выражение

Начальным распределением вероятностей марковской цепи называется распределение вероятностей состояний в начале процесса .

Переходной вероятностью на *k*-ом шаге из состояний в состояние называется условная вероятность того, что система после *k*-го шага окажется в состоянии при условии, что непосредственно перед этим (после шага) она находилась в состоянии .

Т.к. система может пребывать в одном из состояний , то для каждого момента времени необходимо вероятностей перехода . Их можно представить в виде следующей матрицы:

|  |  |
| --- | --- |
|  | (1.42) |

Данная матрица, где – вероятность перехода за один шаг из состояния состояние , а – вероятность задержки системы в состоянии , называется матрицей переходных вероятностей. Если переходные вероятности не зависят от номера шага, а зависят только от того, из какого состояния в какое осуществляется переход, то соответствующая цепь Маркова носит название однородной.

Если для однородной марковской цепи заданы начальное распределение вероятностей и матрица переходных вероятностей, то вероятности состояний системы () определяются по рекуррентной формуле:

|  |  |
| --- | --- |
|  | (1.43) |

Для однородной цепи Маркова можно поставить задачу нахождения предельных вероятностей состояний:

|  |  |
| --- | --- |
|  | (1.44) |

Для нахождения предельных вероятностей необходимо перейти в формуле (1) к пределу при k стремящемся к бесконечности. Получится система из *n* линейных уравнений:

|  |  |
| --- | --- |
|  | (1.45) |

При составлении системы удобно пользоваться понятием потока вероятности. Потоком вероятности из состояния в называется произведение *PiPij*. Дополнительно в систему добавляется условие нормировки:

|  |  |
| --- | --- |
|  | (1.46) |

В процессе работы модуля можно выделить 3 его состояния: попытка установления соединения с сервисом электронных счетов-фактур, попытка соединиться с базой данных для обработки необходимых данных и формирование файла с результатом обработки программы. Опытным путем определим значения переходных вероятностей и представим их в виде матрицы:

|| *Pij* || = ,

а также вектор начальных состояний системы:

*P*0 = .

Необходимо найти вероятность того, что за 2 временных промежутка модуль не выйдет из состояния успешной передачи информации.

Используя матрицу переходных вероятностей, определим вероятности состояний после первого шага:

*P*1(1) = *P*1(0) *P*11 + *P*2(0) *P*21  + *P*3(0) *P*31 = 0,2∙0,07+0,7∙0,2+0,3∙0 = 0,154

*P*2(1) = *P*1(0) *P*12  + *P*2(0) *P*22  + *P*3(0) *P*32 = 0,2∙0,9+0,7∙0,2+0,3∙0,2 = 0,36

*P*3(1) = *P*1(0) *P*13  + *P*2(0) *P*23  + *P*3(0) *P*33  = 0,2∙0+0,7∙0,7+0,3∙0,7 = 0,28

Вероятности состояний после второго шага соответственно:

*P*1 (2) = *P*1(1) *P*11 + *P*2(1) *P*21  + *P*3(1) *P*31 =0,384∙0,08+0,36∙0,4+0,28∙0 = 0,174

*P*2 (2) = *P*1(1) *P*12  + *P*2(1) *P*22  + *P*3(1) *P*32 = 0,384∙0,11+0,36∙0,4+0,28∙0,2 = 0,242

*P*3 (2) = *P*1(1) *P*13  + *P*2(1) *P*23  + *P*3(1) *P*33  = 0,384∙0+0,36∙0,9+0,28∙0,8 = 0,548

Данный расчет показывает, что вероятность остаться в режиме успешной постоянной передачи информации после второго шага – 0,548.

Для определения вероятности безотказной работы рассчитаем предельные вероятности: составим систему уравнений для предельных вероятностей в условиях матрицы перехода:

Решив данную систему уравнений, например, методом Гаусса, получаем результат:

*p*1 = 0,1011

*p*2 = 0,056

*p*3 = 0,723

говорящий о том, что вероятность безотказной работы, равна 0,723.

**1.8 Модель Миллса.**

Модель Миллса , была создана в 1972 году программистом Харланом Миллсом. Он получил широкое распространение благодаря своей простоте и интуитивной привлекательности. Пусть в ходе тестирования было обнаружено *n* естественных ошибок и *m* искусственных.

Предположим, что вероятность обнаружения для естественных и искусственных ошибок одинакова.Тогда выполняется соотношение

|  |  |
| --- | --- |
|  | (1.47) |

Отсюда количество ошибок в программе определяется выражением

|  |  |
| --- | --- |
|  | (1.48) |

Таким образом, количество необнаруженных естественных ошибок можно найти по формуле

|  |  |
| --- | --- |
|  | (1.49) |

Предположим, что в программе *N* естественных ошибок. Внесем в нее *M* искусственных ошибок. Будем тестировать программу до тех пор, пока не найдем все искусственные ошибки. Пусть к этому моменту найдено *n* естественных ошибок. На основании этих чисел вычислим величину *C*, которая выражает степень доверия к модели:

|  |  |
| --- | --- |
|  | (1.50) |

В программу преднамеренно внесли 6 ошибок. В результате тестирования обнаружено 9 ошибок, из которых 6 ошибок были внесены преднамеренно. Все обнаруженные ошибки исправлены. До начала тестирования предполагалось, что программа содержит не более четырех ошибок. Требуется оценить количество ошибок до начала тестирования и степень отлаженности программы.

Для оценки количества ошибок до начала тестирования используем формулу (1.48).

Нам известно:

* количество внесенных в программу ошибок – *M* = 6;
* количество обнаруженных ошибок из числа внесенных – *m* = 6;
* количество собственных ошибок в программе – *n* = 9 – 6 = 3.

Подставив указанные значения в формулу (1.48), получим оценку количества ошибок:

Таким образом, из результатов тестирования следует, что до начала тестирования в программе имелось три ошибки.

Нам известно:

* количество обнаруженных собственных ошибок в программе – n = 3;
* количество предполагаемых ошибок в программе – r = 4;
* количество преднамеренно внесенных и обнаруженных ошибок – M = 6.

Очевидно, что обнаружено меньшее число собственных ошибок, чем количество предполагаемых ошибок в программе (*n < r*). Для оценки отлаженности программы используем уравнение (1.50):

Степень отлаженности программы равна 0,54. Если мы хотим увеличить степень отлаженности программы, мы должны составить соответствующее уравнение.

Допустим, степень отлаженности должна быть равной 86%, тогда

;

.

Следовательно, нам необходимо внести 31 искусственных ошибок, чтобы получить степень отлаженности равной 86 %.

**1.9 Модель Липова.**

В модели Миллса считается, что собственные и искусственные ошибки имеют равную вероятность быть найденными. В методике Липова предложена формула по нахождению вероятности обнаружения ошибок:

|  |  |
| --- | --- |
|  | (1.51) |

где *N* – первоначальное число ошибок в программе; *m* – количество тестов; *S* – количество искусственно внесенных ошибок; *n* – число найденных собственных ошибок; *V* – число обнаруженных к моменту оценки искусственных ошибок; *q* – вероятность обнаружения ошибки в каждом из тестов, рассчитываемая по формуле:

|  |  |
| --- | --- |
|  | (1.52) |

Применение модели Липова требует выполнения следующих условий:

|  |  |
| --- | --- |
|  | (1.53) |

Достоинством модели Липова является то, что она позволяет оценить вероятность обнаружения некоторого количества ошибок к моменту оценки.

При проведении тестирования были получены следующие исходные данные для расчета:

*– N* = 20 – первоначальное число ошибок в программе;

*– m* = 10 – количество тестов;

*–* *S* = 3 – количество искусственно внесенных ошибок;

*– n* = 2 – число найденных собственных ошибок;

*– V* = 4 – число обнаруженных к моменту оценки искусственных ошибок.

Вероятность обнаружения ошибки в каждом из тестов рассчитывается по формуле (1.52):

Рассчитаем вероятность обнаружения ошибок по формуле (1.51):

Вероятность безотказной работы равна 1 – *Q* = 0,929

**1.10 Простая интуитивная модель**

Использование простой интуитивной модели предполагает проведение тестирования двумя группами специалистов, использующими различные тестовые наборы независимо одна от другой. В процессе тестирования каждая из групп фиксирует все найденные ею ошибки. При оценке числа оставшихся в программном средстве ошибок результаты тестирования обеих групп собираются и сравниваются.

К примеру, первая группа специалистов обнаружила *N*1 ошибок, вторая – *N*2, a *N*12 – это ошибки, обнаруженные дважды (обеими группами). Если обозначить через *N* неизвестное количество ошибок, присутствовавших в программе до начала тестирования, то эффективность тестирования каждой из групп следующая:

|  |  |
| --- | --- |
|  | (1.54) |

|  |  |
| --- | --- |
|  | (1.55) |

Эффективность тестирования можно интерпретировать как вероятность того, что ошибка будет обнаружена. Таким образом, можно считать, что первая группа обнаруживает ошибку в программе с вероятностью , вторая – с вероятностью . Тогда вероятность *p*12 того, что ошибка будет обнаружена обеими группами, можно принять равной . С другой стороны, так как группы действуют независимо друг от друга, то *p*12 = *p*1*p*2.

Тогда

|  |  |
| --- | --- |
|  | (1.56) |

Отсюда получаем оценку первоначального числа ошибок программы:

|  |  |
| --- | --- |
|  | (1.57) |

Предполагая, что возможность обнаружения всех ошибок одинакова для обеих групп, можно допустить, что если первая группа обнаружила определенное количество всех ошибок, она могла бы определить то же количество любого случайным образом выбранного подмножества. В частности, можно допустить

|  |  |
| --- | --- |
|  | (1.58) |

В процессе тестирования программы первая группа специалистов нашла 14 ошибок, вторая группа – 18 ошибок, общих ошибок – 6. Определить надежность по простой интуитивной модели.

В программе всего имеется 48 ошибок, так как первая группа нашла 14 ошибок, тогда вероятность нахождения ошибки первой группой

Аналогично для 2-й группы

Так как группы действуют независимо друг от друга, то *p*12 = *p*1*p*2. Тогда вероятность нахождения ошибки обеими группами

Можно сделать вывод, что безотказность работы программы находится так:

Все результаты расчетов сведены в таблицу 1.4.

Таблица 1.4 – Исходные данные и результаты расчета

|  |  |
| --- | --- |
| Величины | Результаты |
| *N*1 | 14 |
| *N*2 | 18 |
| *N*12 (общие ошибки) | 6 |
| *N* (все ошибки) | 48 |
| *p*1 | 0,291667 |
| *p*2 | 0,375 |
| *p*12 | 0,109375 |
| *p*б | 0,890625 |

**1.11 Модель Коркорэна**

Модель Коркорэна относится к статистическим моделям надежности программных средств, т.к. в ней не используются параметры времени тестирования и учитывается только результат *N* испытаний, в которых выявлено *N* ошибок *i*-го типа. Модель использует изменяющиеся вероятности отказов для различных типов ошибок.

Применение модели предполагает знание следующих ее показателей:

* модель содержит изменяющуюся вероятность отказов для различных источников ошибок и соответственно разную вероятность их исправления;
* в модели используются такие параметры, как результат только *N;*
* испытаний, в которых наблюдается *Ni* ошибок *i*-го типа;
* выявление в ходе *N* испытаний ошибки *i*-го типа появляется с вероятностью *аi*.

Показатель уровня надежности *R* вычисляют по следующей формуле:

где 𝑁0 – число успешных выполнений программы; *N* – число выполнений программы; *К* – известное число типов ошибок; *ai* – вероятность выявления при тестировании ошибок i-го типа; *Yi* – вероятность появления ошибок (при *Ni* > 0, *Yi* = *ai*, при *Ni* = 0, *Yi* = 0).

Было проведено 200 испытаний программы, в результате которых 70 из 200 испытаний прошли безуспешно, а в остальных 130 случаях были получены

данные предоставленные в таблице 1.5.

Таблица 1.5 – Результаты испытаний

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Тип ошибки, *k* | Вероятность появления ошибки, 𝑎𝑖 | Число появления ошибок *Ni* при испытании | при *Ni* > 0,  *Yi* = ai | *(Yi*  *Ni* 1** |
| Ошибки вычисления | 0,065 | 5 | 0,065 | 0,0013 |
| Логические ошибки | 0,29 | 18 | 0,29 | 0,02465 |
| Ошибки  ввода/вывода | 0,07 | 5 | 0,07 | 0,0014 |
| Ошибки манипулирования данными | 0,10 | 0 | 0 | 0 |
| Ошибки  сопряжения | 0,1 | 11 | 0,1 | 0,005 |
| Ошибки  определения данных | 0,0758 | 1 | 0,0758 | 0 |
| Ошибки БД | 0,1 | 4 | 0,1 | 0,0015 |

Далее по данным из результатов тестирования производится вычисление показателя уровня надежности *R* по формуле (1.59):

*R* = 130/200 + 0,0013 + 0,02465 + 0,0014 + 0 + 0,005 +0+ 0,0015 = 0,68385.

Таким образом расчетный уровень надежности программного средства по модели Коркорэна составил 0,86.

**1.12 Модель надежности Нельсона**

Модель основана на выделении областей исходных данных *Еi,* покрывающих все множество вариантов их использования в ПС *Е.*

Вероятность *R* того, что прогон программы на наборе входных данных *Е****i****,* случайно выбранном из *Е* среди равновероятных наборов, приведет к приемлемому результату, определяется выражением:

|  |  |
| --- | --- |
|  | (1.60) |

Для испытания ПС использовалось 42 наборов исходных данных, которые равновероятно выбирались для прогона *N* = 42 тестов. При этом *N*0 = 37 тестов обнаружили дефекты. Проведём расчет надежности ПС.

В соответствии с моделью Нельсона надежность ПС по результатам испытаний определяется вероятностью *R* того, что прогон программы на наборе входных данных *Еi*, случайно выбранном из *Е* среди равновероятных, приведет к приемлемому результату, которая вычисляется в соответствии с выражением (1.60):

Вероятность *R* события, при котором прогон программы на заданном наборе исходных данных не приведет к рабочему отказу, равна 0,88.

**1.13 Модель последовательных независимых испытаний Бернулли**

Пусть в каждом независимом испытании может наступить один из r исходов, причем их вероятности не зависят от номера испытания (однородная схема). Вероятность *P*n(*m*1,*m*2,…,*mr*) того, что в *n* испытаниях полиномиальной схемы исход «1» наступил *m*1 раз, исход «2» – *m*2 раз, …, исход «*r*» – *m*r раз, будет тогда определяться равенством (полиномиальной формулой):

|  |  |
| --- | --- |
| , | (1.61) |

где *pi* – вероятность *i*-го исхода в отдельном испытании (*i* = 1, 2, …, *r*), *m*1, *m*2, …, *m*r – целые неотрицательные числа, удовлетворяющие равенству *m*1 + *m*2 + … + *m*r = *n*.

Частный случай последовательности независимых испытаний, в котором каждое испытание может закончиться одним из двух исходов, называют схемой Бернулли. Обычно один из этих исходов условно называют «успехом» (исход *A*), а другой – «неудачей» (исход *Ā*), а их вероятности обозначают *p* (0 ≤ *p* ≤ 1) и *q* = 1 - *p* соответственно. Для схемы Бернулли часто представляет интерес событие *Bm* = {в *bn* испытаниях наступило ровно m успехов}. Вероятность успеха в n испытаниях наступило ровно m определяется формулой Бернулли:

|  |  |
| --- | --- |
| , , | (1.62) |

Для нормальной работы программного модуля необходима безотказная работа в течение дня, как минимум, 3 компьютеров. Рассчитаем, сколько компьютеров нужно установить, чтобы с вероятностью, не меньшей *P* = 0,99, обеспечить нормальную работу приложения, если вероятность отказа компьютера в течение дня равна 0,06.

Вероятность безотказной работы любого компьютера в течение дня составляет *p* = 1 – *q* = 1 – 0,06 = 0,94.

Если установить *n* = 3 рабочих компьютера, то в течение дня безотказно должны работать они все. По формуле Бернулли:

*P*33 = *C*33 ∙ (0,94)3 ∙ (0,06)0 = 0,7742.

Очевидно, что полученного значения вероятности безотказной работы недостаточно. Следовательно, необходимо увеличить число компьютеров.

Предположим, что установлено *n* = 4 компьютеров. Тогда по теореме сложения вероятностей несовместных событий:

*P*43 = 0,9367.

– вероятность того, что в течение дня безотказно будут работать, как минимум, 3 компьютеров из 4.

Данное значение также меньше требуемой надежности работы. Добавляем еще один:

Пусть установлено *n* = 5 компьютеров. Тогда безотказно должны работать 3, 4 или 5 компьютеров. Найдем вероятность того, что в течение дня безотказно будут работать, как минимум, 3 компьютеров из 5:

*P*53 = 0,9934.

**2 ИССЛЕДОВАНИЕ И РАСЧЕТ ВЕЛИЧИН ПОЖАРНЫХ РИСКОВ**

Исследование и расчет пожарных рисков производится в три этапа: расчет времени эвакуации из здания, времени от начала пожара до блокировки эвакуационных путей и расчет потенциального риска в зданиях объекта.

Расчет пожарного риска будет производиться на примере этажа бизнес-центра «Академия», показанного на рисунке 2.1.

На данном этаже расположены:

– помещение 1: площадь *S* = 100 м2, количество человек: 8;

– помещение 2: площадь *S* = 80 м2, количество человек: 7;

– помещение 3: площадь *S* = 80 м2, количество человек: 10;

– помещение 4: площадь *S* = 400 м2, количество человек: 35.

Высота помещений составляет *H* = 4 м, ширина дверных проемов составляет *а* = 1 м, высота дверных проемов составляет *b* = 2 м.



Рисунок 2.1 – План этажа бизнес-центр «Академия»

**2.1 Определение расчетного времени эвакуации.**

Расчетное время эвакуации людей из помещений и зданий устанавливают по расчету времени движения одного или нескольких людских потоков через эвакуационные выходы от наиболее удаленных мест размещения людей непосредственно наружу или в безопасную зону. Также учитывается весь путь движения людского потока, который разделяют на участки определенной длины и ширины.

Расчетное время эвакуации людей следует определять как сумму времени движения людского потока по отдельным участкам пути по формуле:

, (2.1)

где – время движения людского потока на первом участке, мин; , , …, – время движения людского потока на каждом из следующих после первого участка пути, мин.

Время движения людского потока по первому участку пути , мин, определяется по формуле:

, (2.2)

где – длина *i*-го участка пути, м; – скорость движения на -ом участке пути.

Плотность однородного людского потока на первом участке пути D1, определяется по формуле:

(2.3)

где – число людей на первом участке, чел.; – средняя площадь горизонтальной проекции человека, , принимаемая равной 0,125; – ширина первого участка пути, м.

Скорость движения на участках, следующих после первого, принимаются по таблице 2.1 в зависимости от интенсивности движения людского потока по каждому из этих участков пути , которая определяется по формуле:

, (2.4)

где , – ширина рассматриваемого *i*-го и предшествующего ему участка пути, м; , – интенсивности движения людского потока по рассматриваемому -му и предшествующему участкам пути, м/мин.

Интенсивность движения первого людского потока на первом участке пути определяется из таблицы 2.1 по значению плотности потока на первом участке пути .

Таблица 2.1 – Интенсивность и скорость движения людского потока на разных участках путей эвакуации в зависимости от плотности потока.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Плот-ность потока *D*, м2/м2 | Горизонтальный путь | | Дверной проем, интенсив-ность *q*, м/мин | Лестница вниз | | Лестница вверх | |
| Скорость *v*, м/мин | Интен-сив-ность *q*, м/мин | Скорость *v*, м/мин | Интен-сив-ность *q*, м/мин | Скорость *v*, м/мин | Интен-сив-ность *q*, м/мин |
| 0,01 | 100 | 1,0 | 1,0 | 100 | 1,0 | 60 | 0,6 |
| 0,05 | 100 | 5,0 | 5,0 | 100 | 5,0 | 60 | 3,0 |
| 0,1 | 80 | 8,0 | 8,7 | 95 | 9,5 | 53 | 5,3 |
| 0,2 | 60 | 12,0 | 13,4 | 68 | 13,6 | 40 | 8,0 |
| 0,3 | 47 | 14,1 | 16,5 | 52 | 15,6 | 32 | 9,6 |
| 0,4 | 40 | 16,0 | 18,4 | 40 | 16,0 | 26 | 10,4 |
| 0,5 | 33 | 16,5 | 19,6 | 31 | 15,6 | 22 | 11,0 |
| 0,6 | 28 | 16,3 | 19,05 | 24,5 | 14,1 | 18,5 | 10,75 |
| 0,7 | 23 | 16,1 | 18,5 | 18 | 12,6 | 15 | 10,5 |
| 0,8 | 19 | 15,2 | 17,3 | 13 | 10,4 | 13 | 10,4 |
| 0,9 и более | 15 | 13,5 | 8,5 | 8 | 7,2 | 11 | 9,9 |

Интенсивность движения в дверном проеме при плотности потока 0,9 и более, равная 8,5 м/мин, установлена для дверного проема шириной 1,6 м и более, а при дверном проеме меньшей ширины интенсивность движения следует определять по формуле:

. (2.5)

Время задержки движения на участке из-за образовавшегося скопления людей на границе с последующим участком определяется по формуле:

. (2.6)

Точкой начала эвакуации является помещение 3, т.к. она максимально удалена от выхода. Следовательно, расчетным временем эвакуации является время выхода людей из помещения 3 на улицу.

Плотность потока людей для первого участка пути, а именно из помещения 3 рассчитывается по формуле (2.3):

*.*

Из таблицы 2.1 получим значения исходя из рассчитанной плотности потока и = 5.0 м/мин.

Согласно формуле (2.2), на первом участке пути время эвакуации

Далее происходит слияние потоков людей из помещения 3 и 2.

Определим интенсивность потока людей из кабинета 2. Для этого рассчитаем плотность людского потока по формуле (2.3):

*.*

Из таблицы 2.1 следует, что

Тогда на втором участке пути интенсивность потока людей после слияния равна:

На втором участке пути время эвакуации по формуле (2.2) будет равно:

Далее этот поток сливается с потоком людей из кабинета 4.

Плотность людского потока из кабинета 4 рассчитывается по формуле (2.3):

Исходя из таблицы 2.1, интенсивность данного потока равна

Интенсивность образовавшегося потока равна:

Следовательно, время движения по 3-му участку пути, исходя из таблицы 2.1, рассчитывается по формуле (2.2):

Далее этот поток сливается с потоком людей из кабинета 1.

Следовательно, плотность людского потока из кабинета 1 рассчитывается по формуле (2.3):

Следовательно, = 1 м/мин.

Интенсивность образовавшегося потока равна:

Время движения по 4-му участку пути, исходя из таблицы 2.1, рассчитывается по формуле (2.2):

Далее выясним интенсивность потока людей в дверных проемах при плотности потока 0,9 и более при дверном проеме шириной 1 м:

Исходя из полученной интенсивности потока, из таблицы 5 выясним, что м/мин.

Следовательно, время движения потока в одном проеме составляет:

Исходя из всех расчетов времени, суммируем время движения на каждом участке пути согласно формуле (2.1) для того, чтобы получить расчетное время эвакуации:

.

**2.2 Расчет времени от начала пожара до блокирования эвакуационных путей**

В результате распространения на них опасных факторов пожара*.* Время от начала пожара до блокировки эвакуационных путей в результате воздействия на них опасных факторов пожара определяется путем выбора из полученных в результате расчетов значений критической продолжительности пожара минимального времени:

(2.5)

где – время достижения значения опасного фактора – потери видимости; – время достижения значения опасного фактора – температура; – время достижения значения опасного фактора – потребление кислорода; – время достижения значения опасного фактора-токсичные продукты горения (*СО*2, *СО*, *HCl*).

Критическая продолжительность пожара по каждому из опасных факторов определяется как время достижения этим фактором критического значения на путях эвакуации и оценивается по формулам:

– по повышенной температуре:

(2.6)

– по потере видимости:

(2.7)

– по пониженному содержанию кислорода:

(2.8)

– по каждому из газообразных токсичных продуктов горения:

(2.9)

где *B* – размерный комплекс, зависящий от теплоты сгорания материала и свободного объема помещения, кг; *A* – размерный параметр, учитывающий удельную массовую скорость выгорания горючего материала и площадь пожара, кг/c; *Z* – безразмерный параметр, учитывающий неравномерность распределения опасного фактора пожара по высоте помещения; *t* – начальная температура воздуха в помещении, °C; *n* – показатель степени, учитывающий изменение массы выгорающего материала во времени; *V* – свободный объем помещения, м3; – коэффициент отражения предметов на путях эвакуации; *E* – начальная освещенность, лк; – предельная дальность видимости в дыму, м; – дымообразующая способность горящего материала, Нп∙ м2/кг; *L* – удельный выход токсичных газов при сгорании 1 кг материала, кг/кг; X – предельно допустимое содержание токсичного газа в помещении, кг∙м-3; – удельный расход кислорода, кг/кг.

Параметр B рассчитывается по формуле:

(2.10)

где – низшая теплота сгорания материала, МДж/кг; удельная изобарная теплоемкость газа, МДж/кг; – коэффициент теплопотерь; – коэффициент полноты горения.

Параметр *Z* рассчитывается по формуле:

(2.11)

где высота рабочей зоны, м; – высота помещения, м.

Определяется высота рабочей зоны следующим образом:

(2.12)

где – высота площадки, на которой находятся люди, над полом; – разность высот пола, м.

Параметр вычисляется по формуле:

, (2.13)

где – удельная массовая скорость выгорания жидкости, кг/(м2∙с); – скорость распространения пламени, м/с.

Для проведения дальнейших расчетов определены и представлены в таблице 2.2 необходимые исходные данные, а именно:

* начальная температура воздуха в помещении ;
* показатель степени, учитывающий изменение массы выгорающего материала во времени *n*;
* удельная изобарная теплоемкость газа ;
* коэффициент отражения предметов на путях эвакуации ;
* коэффициент полноты горения ;
* коэффициент теплопотерь ;
* начальная освещенность *E*;
* предельная дальность видимости в дыму ;
* высота помещения *H*;
* разность высот пола δ;
* предельно допустимое содержание токсичного газа в помещении *X*.

Таблица 2.2 – Исходные данные для расчетов

|  |  |
| --- | --- |
| Параметр | Значение |
|  | 20 |
| *n* | 3 |

Продолжение таблицы 2.2

|  |  |
| --- | --- |
| Параметр | Значение |
| , кДж/кг | 1,005 |
|  | 0,3 |
|  | 0,87 |
|  | 0,3 |
| *E*, лк | 500 |
|  | 20 |
| *H* , м | 3,5 |
| δ , м | 0 |
| , кг/м3  , кг/м3  , кг/м3 | 0,11  1,1610-3  2310-6 |

Исходя из вышеописанных формул, можно рассчитать время от начала пожара до блокирования эвакуационных путей в результате распространения на них опасных факторов пожара для следующих сценариев.

Сценарий 1: горит машинное масло и ковролин. Исходя из материала определим следующие параметры для расчета по сценарию:

– низшая теплота сгорания *Q:* 34,5 МДж/кг;

– линейная скорость пламени *v*: 0,007 м/с;

– удельная скорость выгорания Ψ*F*: 0,014 кг/(м2 с);

– дымообразующая способность *Dm*: 499 Нп м2/кг;

– удельный расход кислорода *LO*2: 3,24 кг/кг;

– свободный объем помещения *V* = 85%.

Выделение газа:

– *СО*2: 3,8 кг/кг;

– *СО*: 0,2 кг/кг.

Рассчитаем высоту рабочей зоны по формуле (2.12):

*h* = 0 + 1,7 - 0,5 ∙ 0 = 1,8 м.

Расчет *A*, *B* и Z выполняется по формулам (2.10), (2.11) и (2.13) соответственно:

Отношение величин *B* и *А:*

Расчет критического времени:

– по повышенной температуре (2.6):

– по потере видимости (2.7):

– по пониженному содержанию кислорода (2.8):

– по каждому из газообразных токсичных продуктов горения (2.9):

Определяем время блокировки по формуле (2.5):

Сценарий 2: горит мебель и канцтовары. Исходя, из материала определим следующие параметры для расчета по сценарию:

– низшая теплота сгорания *Q:* 15,4 МДж/кг;

– линейная скорость пламени *v*: 0,01 м/с;

– удельная скорость выгорания Ψ*F*: 0,012 кг/(м2• с);

– дымообразующая способность *Dm*: 145 Нп• м2/кг;

– удельный расход кислорода *LO*2: 3,8 кг/кг;

– свободный объем помещения *V* = 85%.

Выделение газа:

– *СО*2: 1,55 кг/кг;

– *СО*: 0,037 кг/кг;

– *HCI*: 0,0036 кг/кг.

Расчет *A*, *B* и *Z* выполняется по формулам (2.10), (2.11) и (2.13) соответственно:

.

Отношение величин *B* и *А:*

Расчет критического времени:

– по повышенной температуре (2.6):

– по потере видимости (2.7):

– по пониженному содержанию кислорода (2.8):

– по каждому из газообразных токсичных продуктов горения (2.9):

160,38 с;

Определяем время блокировки по формуле (2.5):

.

*Сценарий 3: техника*. Исходя, из материала определим следующие параметры для расчета по сценарию:

– низшая теплота сгорания *Q:* 19,7МДж/кг;

– линейная скорость пламени *v*: 0,013 м/с;

– удельная скорость выгорания Ψ*F*: 0,0068 кг/(м2 с);

– дымообразующая способность *Dm*: 296 Нп м2/кг;

– удельный расход кислорода *LO*2: 1,94 кг/кг;

– свободный объем помещения *V* = 85%.

Выделение газа:

– *СО*2: 0,248 кг/кг;

– *СО*: 0,0445 кг/кг.

Расчет *A*, *B* и Z выполняется по формулам (2.10), (2.11) и (2.13) соответственно:

Отношение величин *B* и *А:*

Расчет критического времени:

– по повышенной температуре (2.6):

– по потере видимости (2.7):

– по пониженному содержанию кислорода (2.8):

– по каждому из газообразных токсичных продуктов горения (2.9):

Определяем время блокировки по формуле (2.5):

.

При всех сценариях *t*бл превышает время эвакуации *tp.*

**2.3 Расчет потенциального пожарного риска.**

Вероятность эвакуации по эвакуационным путям определяется по формуле :

(2.14)

где – время от начала реализации *j*-го сценария пожара до блокирования эвакуационных путей в результате распространения на них опасных факторов пожара, имеющих предельно допустимые для людей значения (время блокирования эвакуационных путей), мин; – расчетное время эвакуации людей из *i*-го помещения при *j*-м сценарии пожара, мин; – интервал времени от начала реализации *j*-го сценария пожара до начала эвакуации людей из *i*-го помещения, мин.

Вероятность эвакуации людей, находящихся в *i*-ом помещении здания при реализации *j*-ой ситуации определяется по формуле:

(2.15)

где – вероятность эвакуации людей, находящихся в *i*-м помещении здания, по эвакуационным путям при реализации *j*-го сценария пожара;   
– вероятность выхода из здания людей, находящихся в *i*-м помещении, через аварийные или иные выходы.

Условная вероятность поражения человека определяется по формуле:

(2.16)

где – вероятность эффективной работы технических средств по обеспечению безопасности людей в i-м помещении при реализации j-го пожара.

Интервал времени от начала реализации *j*-го сценария пожара до начала эвакуации людей из *i*-го помещения примем равным 15 секундам.

Вероятность выхода из здания людей, находящихся в *i*-м помещении, через аварийные или иные выходы берем равной 0,1.

Вероятность эффективной работы технических средств по обеспечению безопасности людей в *i*-м помещении при реализации *j*-го пожара берем равной 0.

Частота возникновения пожара в рассматриваемом помещении  
. Для расчета этого параметра на всю площадь помещения используется следующая формула:

(2.17)

где *S* – площадь помещения, м2.

Тогда потенциальный пожарный риск можно рассчитать по формуле:

(2.18)

Величина индивидуального риска *Rm* год-1, для работника *qi*т при его нахождении в здании объекта, обусловленная опасностью пожаров в здании, определяется по формуле:

|  |
| --- |
| , (2.19) |

где – величина потенциального риска в *i*-м помещении здания, год-1; – вероятность присутствия работника т в *i*-м помещении; – число помещений в здании и сооружении.

Исходя из вышеописанных формул, можно рассчитать время от начала пожара до блокирования эвакуационных путей в результате распространения на них опасных факторов пожара для рассмотренных ранее сценариев.

Сценарий 1. Рассчитываем вероятность эвакуации по пожарным путям согласно с формулой (2.14):

По формуле (2.15) рассчитаем вероятность эвакуации людей для данного сценария, принимая вероятность выхода из здания людей, находящихся в рассматриваемом помещении *Р*д.в*ij*, через аварийные выходы равной *Р*д.в*ij* = 0,03:

.

По формуле (2.16) рассчитаем вероятность поражения человека в данном сценарии, принимая вероятность эффективной работы технических средств по обеспечению пожарной безопасности рассматриваемых помещений *Dij* = 0,2:

.

Частота возникновения пожара согласно формуле (2.17):

год-1.

Рассчитаем риск возникновения пожара согласно с формулой (2.18):

год-1.

Сценарий 2. Рассчитываем вероятность эвакуации по пожарным путям согласно с формулой (2.14):

.

По формуле (2.15) рассчитаем вероятность эвакуации людей для данного сценария, принимая вероятность выхода из здания людей, находящихся в рассматриваемом помещении *Р*д.в*ij*, через аварийные выходы равной *Р*д.в*ij* = 0,03:

.

По формуле (2.16) рассчитаем вероятность поражения человека в данном сценарии, принимая вероятность эффективной работы технических средств по обеспечению пожарной безопасности рассматриваемых помещений *Dij* = 0,126.:

.

Частота возникновения пожара согласно формуле (2.17):

год-1.

Рассчитаем риск возникновения пожара согласно с формулой (2.18):

год-1.

*Сценарий 3*. Рассчитываем вероятность эвакуации по пожарным путям согласно с формулой (2.14):

По формуле (2.15) рассчитаем вероятность эвакуации людей для данного сценария, принимая вероятность выхода из здания людей, находящихся в рассматриваемом помещении *Р*д.в*ij*, через аварийные выходы равной *Р*д.в*ij* = 0,03:

*.*

По формуле (2.16) рассчитаем вероятность поражения человека в данном сценарии, принимая вероятность эффективной работы технических средств по обеспечению пожарной безопасности рассматриваемых помещений .:

Частота возникновения пожара равна согласно формуле (2.17):

год-1.

Рассчитаем риск возникновения пожара согласно с формулой (2.18)::

год-1.

*Потенциальный риск.* Величина потенциального риска *Рi* год-1, в *i*-м помещении здания определяется по формуле:

(2.20)

где – число сценариев возникновения пожара в здании; – частота реализации в течении года -го сценария пожара, год-1; – условная вероятность поражения человека при его нахождении в*i*-м помещении и реализации -го сценария пожара.

Следовательно, равно:

*Pt* = 0,00000098 + 0,000261 + 0,0001291 = 39,108 \* 10-5 год-1

Определения индивидуального пожарного риска для рассматриваемого здания принимаем, что у каждого представителя той или иной профессии 200 рабочих дней в году. В рассматриваемом здании работают представители профессии программист (8 ч в рабочий день = 1600 ч/год = 0,18).

При определении индивидуального риска требуется определить вероятность нахождения работника в здании за год. В среднем работник, в данном случае разработчик, проводит в здании около 1600 часов в год при 8-ми часов рабочем дне. Следовательно, вероятность нахождения разработчика в здании за год является 0,18.

Тогда величина индивидуального риска для работника высчитывается по формуле (2.19):

Исходя, из проведенных расчетов можно подвести следующие итоги:

– время эвакуации меньше времени блокировки при трех рассчитанных сценариях {, , }. Следовательно, люди, находящиеся на данном этаже, успеют эвакуироваться до блокировки путей по трем приведенным сценариям;

– потенциальный пожарный риск разработчика находится на удовлетворительном уровне и составляет . Вмешательств и улучшений не требуется.

**ЗАКЛЮЧЕНИЕ**