И.Б. ШУБИНСКИЙ

Функциональная надежность информационных систем

Методы анализа

ШУБИНСКИЙ И.Б.

ФУНКЦИОНАЛЬНАЯ НАДЕЖНОСТЬ ИНФОРМАЦИОННЫХ СИСТЕМ

Методы анализа

ООО «Журнал Надежность»
Москва, 2012 г.

ББК 39.973
Ш 93
УДК 681.31

Шубинский И.Б.
Функциональная надежность информационных систем. Методы анализа / И.Б. Шубинский. – М.: «Журнал Надежность», 2012, – 296 с., ил.

ISBN……..

В книге впервые представлена теория функциональной надежности информационных систем как составная часть общей теории надежности. Она включает понятия и определения; основные угрозы нарушения функциональной надежности информационных систем; систему показателей; методы оценки функциональной надежности цифровых устройств; методы и модели оценки функциональной надежности программного обеспечения. В отдельной главе рассмотрена функциональная надежность критически важных информационных систем, в том числе понятие критически важной системы, особенности оценки сбойных ошибок, оценки функциональной надежности операторов, оценки опасных отказов и рисков, требования к функциональной надежности и к архитектуре программного обеспечения критически важных информационных систем.

В конце каждой главы содержатся контрольные вопросы по наиболее сложному и значимому материалу главы.

Книга рассчитана, в первую очередь, на специалистов, занимающихся практической работой по разработке, производству, эксплуатации и модификации информационных технологий и информационных систем. Она предназначена научным работникам в области надежности программно-аппаратных средств информационных систем, преподавательскому составу, аспирантам и студентам, специализирующимся в области информационных технологий, а также в области автоматизированных систем управления.

УДК 681.31
ББК 32.972

ISBN ……

Предисловие

Информационные системы применяются для решения широкого спектра научных и производственных задач – от традиционного сбора, обработки, накопления и хранения информации, от решения задач искусственного интеллекта до управления ответственными объектами в реальном масштабе времени. Эти задачи имеют актуальное значение в жизни современного общества. Отсюда высокий уровень требований, предъявляемых к надежности информационных систем. При этом следует различать два класса задач обеспечения их надежности. Первый класс – это задачи структурной надежности. Эти задачи и методы анализа рассмотрены нами в работе [1].

Второй класс – это задачи функциональной надежности информационных систем. Необходимость выделения класса *«функциональная надежность»* в общей теории надежности была замечена специалистами еще на начальном этапе создания информационных систем. Ни с технической, ни с математической точки зрения не представляется возможным объединить отказы аппаратуры и ошибки человека – оператора, и тем более, отказы аппаратуры, ошибки операторов, сбойные и программные ошибки. Здесь под термином *ошибка* понимается: *в широком смысле – непреднамеренное отклонение от истины или правил; в узком смысле – отклонение значения измеряемой или теоретически определяемой величины от ее настоящего значения.*

Были попытки ввести некие критерии цены ошибки – если больше допустимой величины, то следует классифицировать как событие нарушения работоспособности и отнести это собы-

тие к отказу системы, введя в схему расчета надежности так называемое «условное устройство». К данному виртуальному устройству относились все события проявления угроз надежности, которые расценивались как отказы системы и которые невозможно было распределить по составным элементам (устройствам, изделиям) системы. В противном случае договорились считать, что в результате ошибки оператора работоспособность системы не нарушается и это событие следует отнести к дефектам и в расчетах надежности системы не учитывать.

В большинстве случаев ошибки операторов трудно и даже невозможно разделить только на две категории: влияющие на надежность («черные» ошибки) или не влияющие на надежность («белые» ошибки). Как правило, имеют место «серые» ошибки, которые невозможно отнести к отказам составных элементов системы, и также недопустимо игнорировать, поскольку они приводят к нарушению некоторых операций, процедур, что может проявиться в виде ошибки управления подчиненными объектами, в том числе через несколько циклов управления. Однако возникновение ошибки управления может быть существенно критичнее, чем отказ отдельного устройства системы, хотя для устранения сбоя или вызванной им сбойной ошибки не требуется ремонт техники, – достаточно исправить или исключить искаженную информацию. Обычно длительность устранения сбоя много меньше длительности устранения отказа техники. Это обстоятельство послужило основанием для некоторых авторов, например [7], разделить программные ошибки на отказы и сбои программного обеспечения по критерию допустимой длительности существования программной ошибки. Такое предложение недостаточно корректно. Это обусловлено тем, что к качеству управления предъявляются не только временные ограничения, но также требования по допустимой точности, пропускной способности, производительности, а также требования к *цене ошибки.*

В целом следует отметить, что в течение полувека на базе методологии классической (структурной) надежности глубоко разработаны количественные характеристики надежности технических средств, методы анализа и синтеза надежности элементов и технических средств, методы повышения надежности аппаратуры на этапах проектирования, рациональные методы технического обслуживания и эксплуатации, методы испытаний аппаратуры на надежность. Вместе с тем, не удалось создать методологию и практические основы анализа и синтеза надежности информационных систем с учетом таких определяющих факторов как сбойные и программные ошибки, ошибки операторов, а также ошибки во входной информации. Основная причина состоит в необходимости исследования не только процессов отказов и восстановлений информационной техники, но и в значительной мере в необходимости исследования безошибочности выполнения информационных технологий на предусмотренной информационной технике с учетом реализуемых в этих технологиях алгоритмов, требований к качеству управления и значимости ошибок. А это уже предмет исследований функциональной надежности.

В предлагаемой читателю книге «*Функциональная надежность информационных систем. Методы анализа*» ***впервые представлена теория функциональной надежности информационных систем как составная часть общей теории надежности***. Она включает понятия и определения; основные угрозы нарушения функциональной надежности информационных систем; систему показателей; методы оценки функциональной надежности цифровых устройств; методы и модели оценки функциональной надежности программного обеспечения. В отдельной главе рассмотрена функциональная надежность критически важных информационных систем, в том числе по-

нятие критически важной системы, особенности оценки сбойных ошибок, оценки функциональной надежности операторов, оценки опасных отказов и рисков, требования к функциональной надежности и к архитектуре программного обеспечения критически важных информационных систем.

Данная книга является второй частью проекта из трех раздельных, но связанных общим замыслом книг. Первая книга «*Структурная надежность информационных систем. Методы анализа*» выпущена издательством «Журнал Надежность» в 2012 г.

В третью книгу данного проекта «*Обеспечение надежности информационных систем*» предполагается включить следующий материал:

– Общие положения (жизненный цикл надежности информационной системы, наблюдаемость и управляемость, избыточность и резервирование, отказоустойчивость, ошибкоустойчивость, информационная защищенность, программа обеспечения

и доказательство надежности).

– Резервирование в информационных системах.

– Обеспечение надежности программных средств.

– Адаптивная отказоустойчивость информационно – управляющих систем.

– Обеспечение функциональной надежности и безопасности критически важных информационных систем.

– Испытания и эксплуатация информационных систем.

Разделы, пункты, рисунки и таблицы первой, второй и третьей взаимосвязанных книг пронумерованы латинскими цифрами I, II и II соответственно.

Во всех трех книгах в конце каждой главы содержатся контрольные вопросы по наиболее сложному и значимому материалу главы.

Все три указанные книги рассчитаны, в первую очередь, на специалистов, занимающихся практической работой по разработке, производству, эксплуатации и модификации информационных систем. Они предназначены научным работникам в

области надежности программно – аппаратных систем, профессорско-преподавательскому составу, аспирантам и студентам, специализирующимся в области информационных технологий, в области информационных систем, а также в области автоматизированных систем управления.

Часть II. Функциональная надежность

Глава II.1. Основные понятия. Угрозы функциональной надежности информационных систем

II.1.1. Понятие функциональной надежности

Функциональная надежность. Это понятие до настоящего времени трактуется различными исследователями неоднозначно. В нормативном документе [2] под функциональной надежностью понимается по существу готовность системы к выполнению предусмотренных задач. Эта позиция сформулирована следующим образом: «вероятность отказа любой части системы будет определять, будет ли система пригодна, когда и сколько необходимо, во время любого эксплуатационного использования и в любое заданное (случайное) время. Факторы, которые влияют на функциональную надежность, включают в себя среднее время между сбоями, среднее время, требуемое на ремонт и время административного простоя».

Другой, более распространенный подход, который закреплен в стандарте [3], состоит в том, что для многофункциональной информационно – управляющей системы (автоматизированной системы управления – АСУ) рассчитывается надежность относительно каждой функции. С этой целью устанавливается перечень функций и видов их отказов, а также критериев этих отказов. Уровень надежности системы оценивается в зависимости от надежности и других свойств технических средств, програм-

много обеспечения и персонала, участвующего в функционировании системы. Для расчета надежности АСУ из ее состава выделяются функциональные подсистемы (ФП), каждая из которых решает одну конкретную задачу и содержит необходимые для этого технические, программные средства и определенный персонал. Анализ надежности всей системы проводят для каждой ФП с учетом надежности ее составных средств. В качестве показателей надежности используют показатели надежности реализации функций. Так, в качестве единичного показателя безотказности системы относительно непрерывно – выполняемой функции вводится вероятность безотказной работы *i-й* ФП в течение заданного времени, а также показатели средней наработки до отказа, наработки на отказ, интенсивности отказов и параметра потока отказов. В качестве комплексных показателей надежности используют коэффициенты готовности, технического использования и сохранения эффективности каждой *i-й* ФП (рис. II.1.1). Здесь в каждой функциональной подсистеме

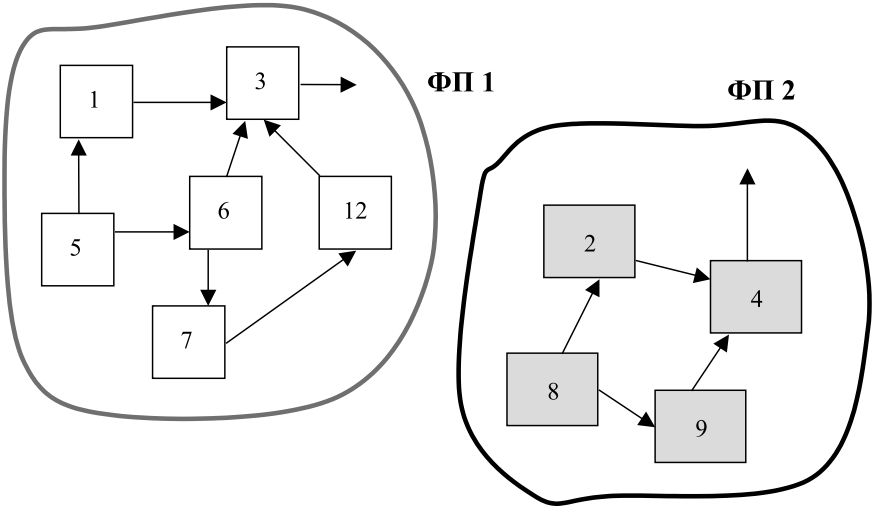


Рис. II.1.1. Распределение технических средств информационной системы по функциональным подсистемам

квадратами обозначены объекты, участвующие в выполнении функции этой подсистемы. Например, в выполнении функции ФП 1 участвуют объекты 1, 3, 5, 6, 7, 12 информационной системы. Стрелки – это связи между объектами. Так, в ФП 1 входной информацией для объекта 3 являются выходные результаты работы объектов 1, 6, 12.

Рассмотренный подход есть не что иное как попытка с позиций структурной надежности объединить надежность технических средств и надежность выполнения информационных процессов в АСУ. Эти позиции хорошо известны, результат объединения получается неудачным. Причины:

1. Разделение многофункциональной информационной системы на ряд функциональных подсистем (ФП) не подкрепляется обоснованными критериями разделения, кроме одного – каждая ФП должна обеспечивать решение одной предусмотренной функциональной задачи. Однако элементы ФП могут практически одновременно участвовать в решении нескольких функциональных задач, т.е. взаимодействовать с другими ФП. Это означает взаимную коррелированность ФП, а, следовательно, и коррелированность их показателей надежности. Реальные уровни надежности ФП могут быть (как правило) далеки от расчетных значений. Эта проблема не обсуждается авторами.

2. В современных информационно – управляющих системах (в частности АСУ) оперативность обработки информации настолько высока, что в доли секунды по случайным запросам могут решаться потоки задач. Понятие непрерывно – выполняемой функции становится неактуальным. В подавляющем большинстве функции выполняются по запросам. Эти запросы поступают в дискретные моменты времени. Интервалы между моментами времени, как правило, носят случайный характер. Таким образом, информационно-управляющая система – это система массового обслуживания запросов. Авторы работы [3]

не исключали этого обстоятельства и ввели понятие дискретно – выполняемых функций (Д-функций) и предположили, что в составе системы наряду с множеством ФП с Н-функциями есть некоторые ФП с Д-функциями. Основным показателем надежности таких систем предлагается вероятность успешного выполнения заданной процедуры при поступлении запроса. Возникают вопросы: процедура – это функция или часть ее? Что означает успешное выполнение – во время ее выполнения не было отказов или процедура выполнена качественно или что-либо еще? Ответы на поставленные вопросы авторами не предусмотрены.

В течение всей истории развития теории надежности был и остался открытым вопрос о том, как рассчитывать надежность системы с динамично изменяющейся структурой. Сейчас можно утверждать, что эта задача не относится к возможностям теории и практики структурной надежности.

В последнее время ряд исследователей исходят из того, что необходимо изучать надежность выполнения информационных технологий на предусмотренной информационной технике с учетом таких угроз как неисправности, ошибки, отказы [71]. В указанной работе такая надежность расценивается как фундаментальное свойство информационной системы и определяется как ***общая надежность***. Здесь под общей надежностью понимается способность информационной системы поставлять обслуживание, которому можно доверять. *Обслуживание,* поставляемое системой, представляет ее свойства или поведение в том виде, в котором это воспринимается *пользователем*. В свою очередь, пользователь является другой системой (физической или человеческой), которая взаимодействует с данной системой через *интерфейс обслуживания*. В трактовке авторов данной работы дерево *общей надежности* имеет следующий вид (рис. I.1.2).

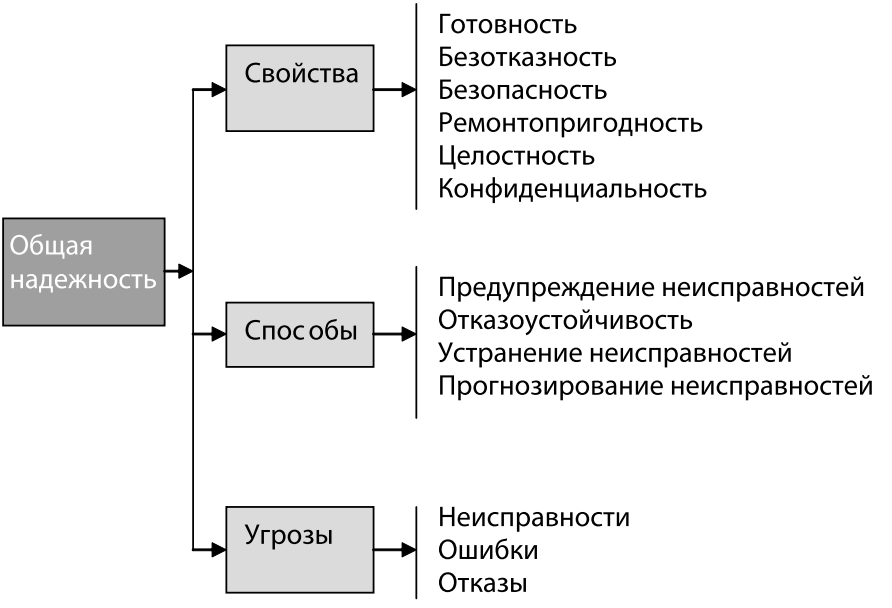


Рис. II.1.2. Дерево «общей надежности»

В указанной работе [71] применены следующие понятия:

- *правильное или корректное обслуживание* осуществляется, когда обслуживание реализует функцию (функции) системы;

- *отказ* системы – событие отклонения осуществляемого обслуживания от правильного обслуживания, т.е. отказ – это переход от правильного обслуживания к *неправильному обслуживанию*, когда не осуществляется функция системы;

- переход от правильного обслуживания к неправильному называется *восстановлением обслуживания*. Временной интервал, в течение которого происходит неправильное обслуживание, называется *простоем* обслуживания;

- *ошибка* является состоянием системы, которое может вызвать последующий отказ.

Отказы, вызванные атаками на систему, классифицируются в работе [71] как «преднамеренно совершенные со злым умыслом» отказы.

Таким образом «общая надежность» представляет собой интегральное понятие и охватывает следующие свойства или атрибуты: *готовность* для правильного обслуживания, *надежность* (понимается *безотказность*), т.е непрерывность правильного обслуживания, *безопасность* – отсутствие катастрофических последствий для пользователя и окружающей среды, *конфиденциальность* – отсутствие неуполномоченного раскрытия информации, *целостность* – отсутствие ненадлежащего изменения состояния системы, *ремонтопригодность* – способность системы к ремонту и обновлениям. Здесь уместно использовать атрибут *информационная защищенность* как совокупность свойств конфиденциальности, целостности и готовности (доступности для зарегистрированных пользователей).

Развитие данного подхода нашло отражение в материалах пятилетних исследований рабочей группы WG 10.4 Международной Федерации (IFIP WG-10.4) по обработке информации [72]. Однако вместо термина «Общая надежность» специалисты этой рабочей группы вводят термин ***«гарантоспособность»***, которая в указанной работе рассматривается как «достоверность вычислительной системы, способной предоставлять требуемые услуги, которым можно оправданно доверять». В явном виде гарантоспособность – это свойство *обслуживания* и зависит от характера использования системы. Здесь подразумевается сочетание аппаратной части, программного обеспечения и человека-оператора информационной системы. Особое внимание авторы обращают на программное обеспечение с гарантией высокой надежности (HDCP), в котором должна быть предусмотрена защищенность от сбойных ошибок (fault-tolerance), защита от опасных отказов (safety), информационная защищенность (security).

Материалы указанных работ [71] и [72] и им близких по концептуальному подходу к надежности функционирования вычислительных систем находятся на понятийном уровне. Вмес-

те с тем, в научном сообществе не выработана единая система взглядов на понимание предмета, целей и задач функциональной надежности информационных систем. Так, в работе [71] акцент сделан на полноту охвата проблемы надежности, о чем, в частности, свидетельствует название «Общая надежность» и перечень рекомендованных свойств общей надежности (например, свойств готовности, ремонтопригодности, безопасности, конфиденциальности и др.). При этом не учтено исключительно важное, на наш взгляд, свойство безошибочности, в перечне угроз отсутствуют информационные атаки. В работах [71,72] и в близких к ним исследованиях акцент сделан на надежность с особыми требованиями к обеспечению безопасности, хорошую информационную защищенность, непрерывность работы и человеко – машинное взаимодействие.

К сожалению, нам не известны работы, содержащие модели (включая показатели функциональной надежности), методы анализа и прогнозирования функциональной надежности информационных систем, а указанные выше и другие известные понятийного плана публикации носят фрагментарный характер. В связи с этим целесообразно остановиться на некоторых основных понятиях функциональной надежности.

Теория и практика функциональной надежности предназначены для исследования потоков функциональных отказов, которые возникают в ходе динамично изменяющихся процессов выполнения задач в информационных системах (ИС), а также для исследования потоков восстановлений информационных процессов. Функциональный отказ – это событие невыполнения функциональной задачи вследствие нарушения информационного процесса.

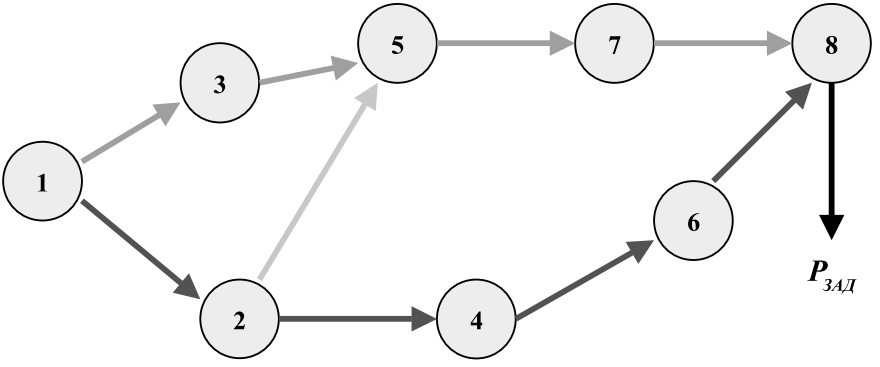
Одной из центральных проблем нарушений информационных процессов, наряду с ошибками в программных средствах, ошибками операторов, ошибками во входной информации, была

и существует *проблема сбойных ошибок цифровой аппаратуры ИС*, частота которых на порядки превышает частоту отказов технических средств. Сбойные ошибки есть результат неправильного (искаженного) выполнения дискретными элементами логических функций, которые затем трансформируются в ошибки в выполнении операций, процедур, программ, задач.

Методология структурной надежности при всей обширности решаемых проблем не ориентирована на расчеты безошибочности выполнения информационных процессов и их составных процедур – она оперирует только процессами отказов и восстановлений технических средств. Эта методология не учитывает содержание алгоритмов выполняемых в системе задач. Она также не учитывает влияние проявленных ошибок в программном обеспечении, ошибок операторов и ошибок во входной информации на результаты выполнения предусмотренных алгоритмов. Все эти факторы (угрозы) ненадежности представляют собой предмет рассмотрения функциональной надежности как составной части общей теории надежности.

Функциональная надежность информационных систем определяется *правильностью и безошибочностью* выполнения информационных процессов. Термин «правильность» означает, что информационные процессы реализуются в соответствии с заданной совокупностью правил и предписаний, т.е. по существу в соответствии с предусмотренными в системе алгоритмами выполнения информационных процессов. Понятие «правильность» в функциональной надежности аналогично понятию «работоспособность» в структурной надежности. Всякое отклонение от заданных правил и предписаний приводит к нарушению *правильности* функционирования информационной системы.

Допустим, что система правильно выполняет предусмотренные задачи. Значит ли, что она функционально надежна? Нет – обеспечение *правильной работы необходимо, но недостаточно.*



$$P_{ЗАД} = P_7(P_5, P_3, P_2, P_1) + P_6(P_4, P_2, P_1)$$

Рис. II.1.3. Фрагмент графа алгоритма задачи. Зависимость вероятности безошибочного выполнения подзадачи от функциональной надежности выполнения составных процедур

Так, под воздействием сбойных ошибок промежуточные и/или выходные результаты правильного выполнения информационных процессов оказались искаженными, что привело, например, к ошибкам в управлении.

На рис. II.1.3 показан фрагмент графа алгоритма задачи. Он содержит процедуры (вершины графа) и связи между ними (ребра графа). Все заданные процедуры задачи (их можно рассматривать как предписания) должны быть выполнены в соответствии с заданными правилами (их можно рассматривать как связи между вершинами графа). Если в данной задаче все действия выполнены строго по правилам и предписаниям, то следует полагать, что информационный процесс решения функциональной задачи реализован правильно. К вершинам графа отнесены вероятности безошибочного выполнения процедур алгоритма задачи. Результирующая вероятность безошибочного выполнения задачи в целом рассчитывается с учетом связей между процедурами выполнения задачи. Свойство безошибочности – комплексное свойство. Оно будет обеспечено как при

условии безошибочности выполнения всех процедур решения функциональной задачи, так и при условии правильности алгоритма задачи, тем более, что нарушения правильности выполнения задач во многом представляют собой результаты воздействия перечисленных выше ошибок в реализации информационных процессов.

От приведенной выше иллюстрации понятия функциональной надежности относительно фрагмента задачи перейдем к рассмотрению причинно – следственных связей, приводящих к нарушению информационного процесса в выполнении задачи в целом. Дело в том, что информационный процесс состоит из совокупности иерархически упорядоченных информационных процессов и реализуется от самого низшего уровня иерархии (этот уровень иерархии процесса обозначим как $i_{\max}$) до процесса самого высшего уровня иерархии (этот уровень иерархии принято обозначать как $i_0 \Rightarrow 0$). Ошибки, возникшие на любом уровне иерархии, распространяются до самого высшего уровня и могут привести к функциональному отказу информационной системы.

Причинно-следственная цепочка ошибок применительно к информационному процессу показана на рис. II.1.4. В результате возникновения сбойной ошибки или проявления программной ошибки при выполнении процесса на *i*-м уровне иерархии в этом процессе закладывается функциональная ошибка, которая распространяется на результаты выполнения процессов последующих уровней иерархии, включая самый высокий. Результаты выполнения процесса нулевого уровня иерархии являются выходными результатами выполнения данного информационного процесса в целом. Они могут привести к функциональному отказу относительно данной задачи. Не исключено также и то, что распространяемая по процессам функциональная ошибка окажется недостаточно существенной в соответствии с

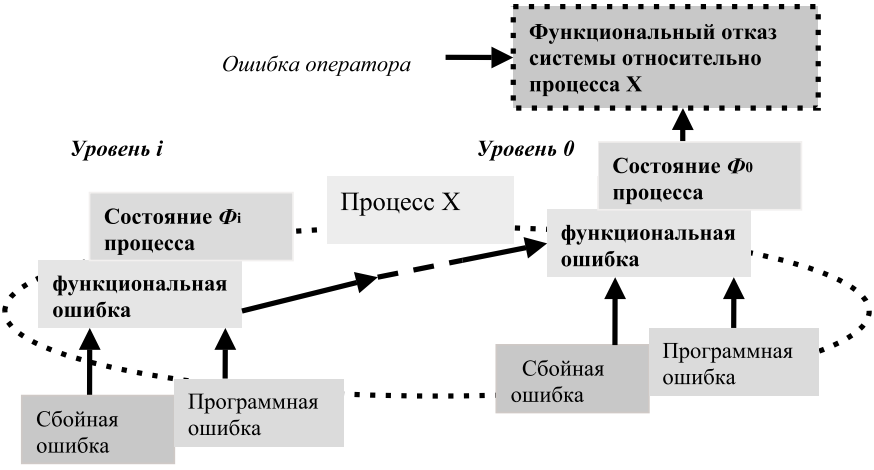


Рис.II.1.4. Концептуальное представление взаимосвязи функциональной ошибки и функционального отказа системы относительно данного информационного процесса

установленным для данной задачи критерием функционального отказа и не приведет к невыполнению этой задачи. В этом случае не следует ожидать возникновения функционального отказа. С другой стороны, на этом уровне иерархии возможно воздействие на результаты выполнения задачи ошибки оператора, которая окажет на них существенное влияние и приведет к возникновению функционального отказа системы.

II.1.2. Определение функционального отказа

Пусть ИС в текущий момент времени выполняет q функциональных задач. Каждая задача реализуется одной или группой программ и описывается набором параметров. Совокупность возможных значений параметров i-й задачи (i=1, …, q) обозначим Y_i. Множество Y_i включает в себя множество x_i параметров надежности технических средств, множество y_i параметров надежности программ, множество l_i параметров надежности

операторов, а также множество k_i параметров надежности информационных каналов ИС, используемых для решения данной задачи, т.е. $Y_i = \{x_i, y_i, l_i, k_i\}$.

Множество $R = (Y_1, Y_2, ..., Y_q)$ значений параметров всех задач, выполняемых в текущий момент времени, представляет собой мгновенный образ информационно – вычислительной среды ИС, характеризующий состояние ее функционирования в этот момент времени. Обработка этого образа может осуществляться различными способами. Типичный вариант обработки – это нахождение центра тяжести $\overline{R}$ образа. В простейшем случае центр тяжести может быть вычислен как средневзвешенное значение мгновенных характеристик Y_i. Координаты центра тяжести $\overline{R}$ образа определяют состояние функционирования вычислительной среды ИС. Вследствие возникновения и устранения ошибок при выполнении любой из q функциональных задач координаты центра тяжести меняются во времени – имеет место случайный процесс $\overline{R}(t)$. Отдельные реализации этого процесса будем называть траекториями процесса смены состояний g, а множество траекторий обозначим G, т.е. $g \in G$. Отсутствие ошибок в результате выполнения любой из функциональных задач на всем интервале времени t соответствует траектории g_0.

Помимо образа $\overline{R}$, характеризующего влияние на функциональную надежность внутренних факторов ИС, следует учитывать параметры внешних факторов, обусловленные воздействием внешней среды на систему. К ним относятся: параметры потоков заявок, которые в случайные моменты поступления определяют требуемое количество технических средств для обслуживания заявок, каналов для передачи сообщений; параметры оперативности обработки и передачи информации и др. Совокупность возможных значений параметров j-го внешнего фактора ($j \in m$, где m – количество учитываемых внешних фак-

торов) обозначим z_j. Вектором $\boldsymbol{Z}=(z_1,z_2,...,z_m)$ определяются значения параметров внешних факторов, которые известны в текущий момент времени.

Введем функцию безошибочности $\Phi(\boldsymbol{R}, \boldsymbol{Z}, t)$, которая характеризует способность ИС в течение времени t безошибочно выполнять различные группы составных процессов (функциональных задач), безошибочно принимать и передавать сообщения в соответствии с изменяющимися во времени параметрами внешних факторов.

Все множество состояний S системы разделяется на два непересекающихся множества $S_\phi \cup \overline{S_\phi} = S$, где S_ϕ – множество приемлемых по безошибочности состояний функционирования ИС, а $\overline{S}_\Phi$ – множество состояний с уровнями безошибочности ИС ниже допустимого. Множество S_ϕ также разделяется на два непересекающихся множества $S_0 \cup S_1 = S_\phi$, где S_0 – состояния, в которых обеспечивается номинальная безошибочность ИС вследствие того, что все запрошенные процессы выполнялись правильно и в полном объеме, а также безошибочно принималась и передавалась вся предусмотренная информация. Это объясняется отсутствием ошибок в системе или хотя бы в тех средствах, которые привлекались для выполнения запрошенных задач; S_1 – состояния пониженной, хотя и приемлемой, безошибочности ИС. Состояния S_1 также можно разделить на группы непересекающихся множеств, упорядочив их по степени снижения уровней приемлемой безошибочности

$$S_{11} \supset S_{12} \supset S_{13} \supset ... \supset S_{1Z},$$

где S_{11} и S_{1Z} – граничные с множествами S_0 и соответственно $\overline{S}_\Phi$ множества состояний пониженной безошибочности. При этом множества состояний $S_{1K}, S_{1j} (k < j)$ являются промежуточными между множествами состояний S_{11} и S_{1Z}.

В принятых терминах под *частичным функциональным отказом* ИС понимается переход процесса $\overline{R}(t)$ из одного множества S_{1K} в другое S_{1j} ($S_{1j} \subset S_{1K}$) со значением функции безошибочности $\Phi_j < \Phi_k$ системы ниже допустимого уровня относительно одного процесса. Уровню номинальной безошибочности Φ_0 соответствует множество состояний S_0.

Полный функциональный отказ ИС наступает при ее переходе из множества состояний S_ϕ в множество состояний $\overline{S}_\Phi$, в котором уровень безошибочности системы меньше допустимого.

Таким образом, *функциональная надежность ИС есть ее способность правильно выполнять предусмотренные функциональные задачи с приемлемым уровнем безошибочности в реальных условиях эксплуатации при взаимодействии с внешними объектами.*

II.1.3. Сбойные ошибки

В информационных системах (ИС) такой тип сбоев как самоустраняющиеся отказы, а также перемежающиеся отказы не оказывают существенного влияния на надежность функционирования ИС. Это объясняется следующими основными причинами.

Многие ИС относятся к категории критически важных систем. Поэтому при создании средств информационной техники применяются специальные меры по стабилизации в приемлемых границах температуры и влажности, обеспечивается качественное кондиционирование помещений и стативов, тщательное проектирование электрических режимов, исключение событий гонок и состязаний сигналов и др.

Как уже отмечалось во введении, многие информационные системы работают в реальном масштабе времени и в директивные сроки выполняют строго определенные информационные технологии. Для этих систем приоритетное влияние на резуль-

таты функционирования оказывают сбои функционального характера, т.е. такие сбои техники, которые при определенных обстоятельствах могут привести к ошибкам в управлении и иметь серьезные последствия для работы всей системы в целом и даже для окружающей среды.

Практика испытаний и эксплуатации информационных систем различного назначения показала, что наиболее опасными угрозами функциональной надежности являются сбои информационной техники, вызванные внутренними или внешними дестабилизирующими факторами – помехами. Именно помехи, главным образом по цепи питания, по заземлению и по входу, в сочетании со входными сигналами, передаточными и амплитудно – временными характеристиками интегральных схем являются в совокупности теми факторами, которые приводят к сбоям, которые, в свою очередь, влияют на правильность выполнения логических функций, микроопераций, операций, процессов и информационных технологий в целом [4,5]. Эта, на первый взгляд, длинная цепочка условий возникновения ошибок в выполнении информационных технологий кажется маловероятной. Однако весь вопрос в том, что понимается под сбоем функционального характера и как часто возникают такие события.

Сбой функционального характера – событие, заключающееся в однократном искажении перерабатываемой или хранящейся в информационной технике информации, возникающее под воздействием внутренних или внешних дестабилизирующих факторов (помех).

В середине 60-х годов XX столетия автором при испытаниях автоматизированных систем управления было установлено, что интенсивность сбоев функционального характера средств информационной техники, работающей в реальном масштабе времени, более чем на два порядка выше интенсивности их отказов [3]. По мере перехода к интегральным схемам, по мере

создания больших и сверхбольших интегральных схем, значительно выросло быстродействие информационной техники, что повлекло за собой повышение интенсивности сбоев функционального характера. Это связано со снижением их помехоустойчивости. Остановимся на этом моменте более подробно.

Помехоустойчивость цифровых интегральных схем определяется по отношению к статическим и импульсным помехам. Статическая помехоустойчивость характеризуется передаточной характеристикой схемы, под которой подразумевается зависимость выходного напряжения схемы от напряжения на одном из входов при постоянных напряжениях на других входах и заданном числе нагрузок.

Импульсная помехоустойчивость ЦИС характеризуется их устойчивостью к воздействию импульсных помех. Причины возникновения этого типа помех в цифровых устройствах обусловлены искажением формы сигнала в ЦИС и линиях связи, гонками и состязаниями сигналов, перекрестными наводками между сигнальными линиями связи и паразитными связями между ЦИС по цепям питания и заземления, отражениями в сигнальных линиях связи [5] .

Оценка импульсной помехоустойчивости схем обычно осуществляется с помощью амплитудно-временной характеристики и показателя вольт-секундной площади допустимых помех. Чем выше этот показатель, тем выше импульсная помехоустойчивость схемы. Чем выше быстродействие схемы, тем меньше время задержки сигналов и, следовательно, меньше вольт-секундная площадь допустимых помех. Импульсные помехи обусловлены, главным образом, помехами по цепям питания и заземления.

Таким образом, ***за повышение быстродействия цифровых интегральных схем приходится расплачиваться увеличением интенсивности их сбое***в. Этот вывод особенно характерен

для триггеров, которые являются базовыми элементами ЦИС средней и большой интеграции. Для повышения помехоустойчивости применяется стробирование триггеров. В результате сбой триггера может произойти из-за помех, возникающих во время его переключения [5]. Однако, чем выше быстродействие устройства, тем выше частота переключения триггеров и тем выше вероятность сбойных ошибок, которые вызваны сбоями логической части устройства и самих триггеров и зафиксированы ошибочными состояниями триггеров. Отсюда следует, что ***интенсивность сбойных ошибок в выходных результатах работы устройства тем больше, чем быстрее и чаще выполняются предусмотренные операции.***

Экспериментальные данные по сбоям интегральных схем отсутствуют. Это объясняется, главным образом тем, что для проведения экспериментов в лабораторных условиях требуются измерительные приборы (генераторы тактовых и информационных импульсов, регулируемых импульсов помех, коммутатор сигналов и импульсов помех, осциллограф), быстродействие которых должно быть на порядок выше быстродействия испытываемых ЦИС. Это означает, что быстродействие элементов измерительных приборов должно определяться временем задержки меньшим 0,1 нс. Построение таких уникальных приборов принципиально возможно. Однако и этого недостаточно. Для получения достоверных данных требуется параллельно испытывать десятки и сотни тысяч однотипных ЦИС в течение двух-трех лет, чтобы оценить характеристики потока сбоев в зависимости от основных видов помех и наиболее характерных условий эксплуатации. Отсюда необходимость в серийном производстве уникальных измерительных приборов для проведения экспериментов по сбоям.

Особенно сложны эксперименты по определению сбоев сверхбольших интегральных схем (СБИС). Это объясняется

тем, что они оперируют со словами и поэтому следует значительно увеличить объем регистрирующей аппаратуры, а обработку текущих данных экспериментов выполнять с помощью быстродействующих ЭВМ. Отсюда следует, что опасность сбоя измерительных приборов и устройств может быть выше, чем испытуемых СБИС. Возможности получения экспериментальных данных по сбоям в производственных условиях также ограничены. Это объясняется организационными трудностями, недостаточной подготовкой в этом направлении и незаинтересованностью обслуживающего персонала, малой представительностью выборки идентичных ЦИС в цифровом устройстве.

Суммируя сказанное, можно определить два направления работ по оценке интенсивности сбоев ЦИС.

Первое направление состоит в том, чтобы наращивать усилия в преодолении указанных выше препятствий для экспериментального определения характеристик потоков сбоев ЦИС.

Второе направление заключается в разработке и развитии экспериментально – расчетных методов прогнозирования сбоев ЦИС. Поскольку помехоустойчивость ЦИС зависит от многих факторов, то выделяются наиболее существенные из них и определяются максимальные границы изменения передаточной характеристики, длительности и амплитуды импульсных помех. Эти данные, а также экспериментальные данные о входных сигналах и помехах, являются исходными для расчета диапазона возможных вероятностей сбоя цифрового логического элемента в интегральном исполнении, который часто именуют вентилем по аналогии с транзистором в элементной базе второго поколения. По полученным экспериментально – расчетным путем значениям сбоев вентилей, известной структуре и реализуемым логическим функциям оцениваются уровни вероятности и интенсивности сбоев функционального характера ЦИС различного уровня интеграции.

По характеру влияния на результаты выполнения и информационных технологий сбои подразделяются на две группы: *защищенные сбои и сбойные ошибки.* ***Защищенные сбои*** – часть сбоев информационной техники, которая своевременно (в пределах допустимого времени перерыва в работе) обнаружена системой контроля и автоматически устранена с помощью предусмотренной в системе избыточности.

Сбойные ошибки **–** это результаты возмущающих воздействий незащищенных сбоев на процессы выполнения программ, информационных технологий и на информационный процесс в целом. Эти воздействия проявляются в виде искажений, подмены или потерь данных, ошибок в промежуточных и/или в выходных результатах. Основное деструктивное влияние на результаты информационного процесса оказывают сбои функционального характера. Они приводят к ошибкам в выполнении микроопераций, а те, в свою очередь, к ошибкам в выполнении операций, которые влияют на микропроцессы, а затем и на процесс. Сбои функционального характера, в отличие от сбоев типа кратковременных самоустраняющихся отказов, обнаруживаются в редких случаях. Они проявляются в основном через сбойные ошибки. Цена таких ошибок в нарушении надежности выполнения информационной технологии в информационно – управляющей системе может быть очень высока, поскольку они приводят к серьезным негативным последствиям в результатах управления подчиненными объектами.

Следует отметить, что оба типа незащищенных сбоев могут оказать деструктивное влияние на информационный процесс в целом. Это негативное влияние проявляется в виде зацикливаний, остановов, перегрузок информационной техники, искажаются ответственные таблицы в общей памяти, искажается информация обмена с внешними абонентами. Зацикливания связаны с нарушениями процессов переходов в управляющих

программах, искажениями или подменами значений циклов и др. Причиной останова может быть искажение кода операции, совпадающее с командой останова. При большой интенсивности потока заявок возрастает интенсивность ошибок в выполнении обмена информацией с внешними абонентами. В результате воздействия сбойных ошибок в ходе выполнения информационной технологии возможны искажения таблиц, а также запись в постоянную память искаженных данных, которые при повторном выполнении программ, реализующих эту информационную технологию, опять приводят к ошибочным результатам.

II.1.4. Ошибки в программном обеспечении

Свойства надежности программ принципиально отличаются от свойств надежности технических средств информационной техники. Основные отличия состоят в следующем:

- Программа физически не стареет и не изнашивается. Возможно только моральное старение программы;
- Программа не требует ремонта и профилактического обслуживания;
- Программные ошибки есть результат создания и корректировки ПО. Они могут не проявляться, если входные данные не инициируют выполнение тех участков программ, в которых содержатся ошибки. В отличие от технических средств некоторые входные потоки данных могут привести к частым ошибкам, в то время как другие потоки могут вообще не привести к ошибкам;
- Каждая программа является уникальным продуктом. Этот факт можно объяснить следующим образом. Во – первых, любые две копии программы абсолютно идентичны. Во – вторых, программа никогда не дает ухудшение характеристик без внешнего вмешательства. В-третьих, процесс отладки программы никогда не повторяется;

▪ Любая корректировка программы (даже если это относится только к устранению обнаруженной ошибки) приводит к новой версии программы, поскольку обязательно приводит к изменению хотя бы одного элемента объектного кода программы. Это важное обстоятельство исключает возможность использования ранее имевшихся статистических данных и принципиально не позволяет производить статистическую оценку надежности программ;

▪ При исправлении обнаруженной ошибки программист может внести новые ошибки, которые могут со временем проявиться при определенном наборе входных данных, что отличается от последствий ремонта технических средств.

Причины ошибок целесообразно увязывать с процессом создания программного обеспечения информационных систем. В работе [6] предложено рассматривать создание программного обеспечения как ряд процессов перевода (трансляции), начинающихся с постановки задачи и заканчивающихся текстом программы и документацией на нее. Отсюда следует, что *программирование* – это процесс решения задачи, а *программное обеспечение* (ПО) – некоторый специальный набор информационных элементов, описывающих реализацию решения этой задачи в соответствии с установленным алгоритмом. Тогда процесс производства ПО можно представить как набор процессов трансляции. Трансляция начинается с момента преобразования постановки задачи до получения промежуточных результатов и заканчивается детальным набором машинных инструкций.

При таком подходе к разработке ПО основной причиной его ненадежности являются ошибки, возникающие в процессе трансляции информации на различных стадиях технологического процесса. Укрупненная модель процессов возникновения ошибок в ходе разработки ПО показана на рис. II.1.5. Рассмотрим кратко каждый из этих процессов.

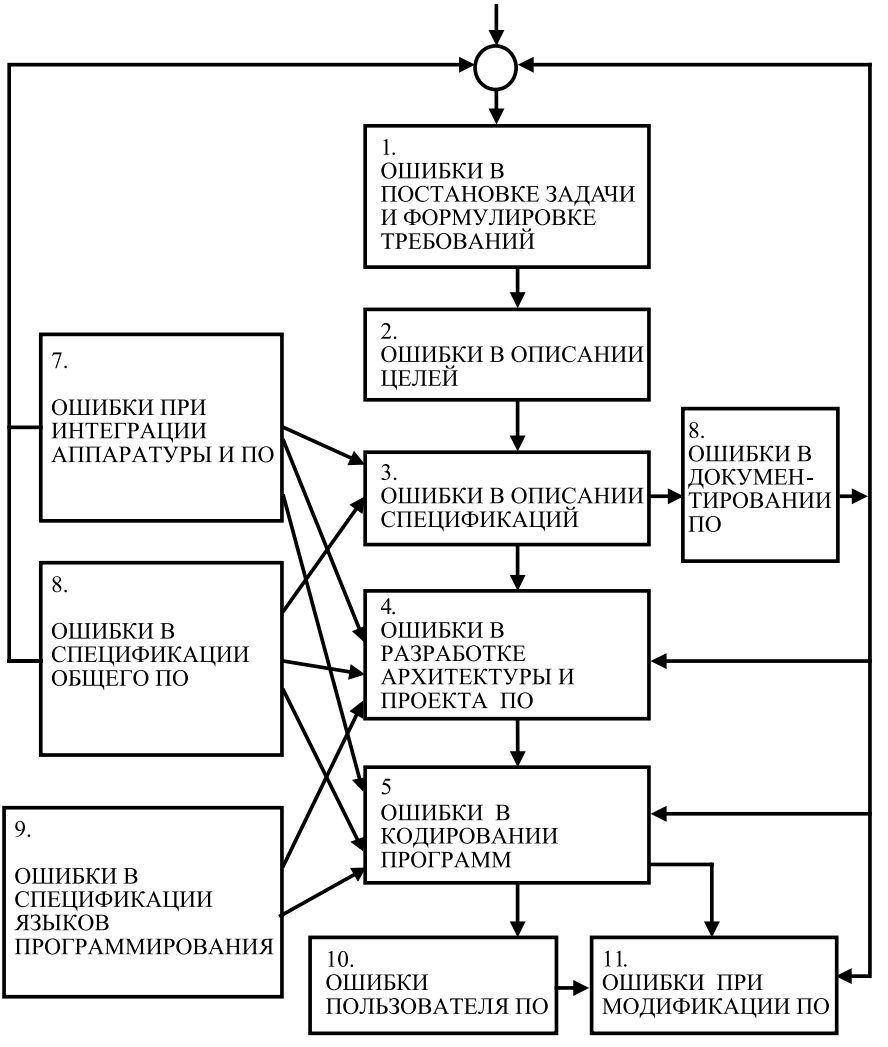


Рис. II.1.5. Укрупненная модель процессов возникновения ошибок в ходе разработки ПО

1. Ошибки в постановке задачи и формулировке требований. Первый процесс начинается с разработки описания решаемой задачи, которое представляет собой перечень требований пользователя. Этот перечень разумнее всего составлять разработчику и пользователю совместно. При составлении данного

перечня всегда имеются большие возможности для ошибок – например, потребности пользователя неверно поняты и учтены разработчиком, либо не учтены в полном объеме, либо пользователь не сумел адекватно выразить свои потребности. Ошибки этого уровня обходятся чрезвычайно дорого.

2. Ошибки в описании целей. Второй процесс состоит в переводе требований пользователя в цели программы. Ошибки на этом этапе возникают, когда неверно интерпретируются требования, не удается выявить все требующие компромиссного решения проблемы или приняты неправильные решения. Они могут возникнуть также в случае, когда не сформулированы цели, необходимые, но не поставленные явно в требованиях пользователя.

3. Ошибки в описании спецификаций. Третий процесс предназначен для точного описания поведения информационной системы с точки зрения пользователя. Иначе говоря, должна решаться задача преобразования целей программы в ее внешние спецификации. Это самый объемный и трудоемкий этап разработки ПО. Он наиболее подвержен ошибкам – они бывают и наиболее многочисленными и наиболее серьезными.

4. Ошибки в разработке архитектуры и проекта ПО. Четвертый этап представляет собой несколько процессов трансляции – от внешнего описания готового продукта до получения детального проекта. На этом этапе внешнее описание переводится в структуру компонентов ПО и создаются алгоритмы программных модулей и составных программ. Поскольку приходится иметь дело с большими объемами информации, то возможности внесения ошибок становятся чрезвычайно высокими.

5. Ошибки в кодировании программ. Пятый процесс проектирования – перевод алгоритма в предложения языка программирования, а также компиляция, т.е. перевод представления программы на языке программирования в объектный код.

Ошибки на этом этапе трансляции имеют место, однако их вероятность возникновения в связи с автоматизацией программирования много меньше, чем на предыдущих этапах проектирования ПО.

6. Ошибки при документировании ПО. Шестой этап завершает разработку ПО как законченного продукта, который может быть передан пользователю. На этом этапе создается документация на ПО, в том числе руководства по инсталляции программ и их эксплуатации. Качество документации оказывает существенное влияние на возможность возникновения программных ошибок. В работе [7] подчеркнуто, что «прочитав руководство, пользователь начнет работать с программой и обнаружит, что она ведет себя не так, как он ожидал, – это и является, по определению, ошибкой в программе». К сожалению, часто качество документации не соответствует предъявляемым к нему требованиям. Это связано с тем, что разработчики программ редко обладают качествами технических писателей, не любят и не всегда умеют кратко, четко и понятно описать программу, правила и порядок работы с ней. В свою очередь, технические писатели недостаточно знают и понимают данную программу.

7. Ошибки при интеграции аппаратуры и ПО. В процессе разработки программы системные спецификации оборудования используются в качестве входных данных. Решения, принимаемые разработчиками программ, во многом зависят от характеристик компьютерных и периферийных средств информационной техники, от характеристик телекоммуникационных средств. Незнание или неправильное толкование этих данных может привести к ошибкам в проектируемом ПО.

8. Ошибки в спецификации общего ПО. Решения разработчиков прикладных программ ограничены возможностями общего ПО, зависят от их средств динамического распределения ресурсов, диспетчеризации вычислений, ввода – вывода инфор-

мации. Незнание или неправильное толкование возможностей общего ПО может привести к ошибкам в проектируемом ПО.

9. Ошибки в спецификации языков программирования. Написание программы обработки информации может быть выполнено на одном или нескольких языках программирования. Конечный продукт формируется с помощью одного языка программирования. Неправильное использование языковых конструкций, синтаксиса и семантики языка (языков) программирования может послужить серьезной причиной возникновения программных ошибок.

10. Ошибки пользователя ПО. Если работает невнимательный пользователь, то вероятность того, что он сделает ошибку, велика. Ошибки пользователя часто приводят к возникновению новых, непредвиденных состояний системы, что, в свою очередь, может повлечь за собой повторение в полном или частичном объеме всех работ, связанных с созданием программ. Следовательно, повторяются ситуации, которые приводят к корректировке целей, спецификаций, алгоритмов и, как следствие, к возникновению новых ошибок.

11. Ошибки при модификации ПО. Эксплуатация ПО является продолжением этапов разработки. На этом этапе жизненного цикла решаются задачи устранения обнаруженных ошибок, которые не направлены на совершенствование программы, но по существу приводят к созданию новой ее версии при каждом случае восстановления работоспособности программы. Процедуру устранения обнаруженных ошибок принято классифицировать как *модификацию* данной программы. В процессе эксплуатации программы возможно изменение некоторых целевых установок, приводящих к необходимости ее доработки. Кроме того, возникают обстоятельства, требующие совершенствования программы. В обоих указанных случаях изменяются некоторые функции программ. Этот процесс

изменения программы называется *модернизацией.* Часто указанные понятия объединяются одним термином – **модификация.** При выполнении как модификации, так и модернизации программы ***программист может не только не исправить ошибку, но и внести новые ошибки,*** которые только понизят надежность программы. Причинами этого являются:

- маскируемые ошибки, появляющиеся после исправления ошибки, которая их маскировала;
- случайные изменения корректного кода программы;
- недостаточный опыт и знания программиста;
- существенное усложнение программы в процессе исправления ошибки.

Значительная часть от общего объема программных ошибок возникает в процессе внесения изменений в рабочую программу. По этой причине ПО, как правило, не является конечным продуктом процесса его проектирования. Очевидно, что еще не раз придется переходить к новым версиям данной программы с целью исправления обнаруженных ошибок и корректировки или добавления новых функций. Приведенное описание модели позволяет выявить большую часть угроз надежности ПО.

Применительно к информационным системам ошибки, возникающие на различных этапах процесса разработки, группируются следующим образом:

– ***системные ошибки***. К ним относятся ошибки в формулировании требований, описании целей, описании спецификаций, включая спецификации оборудования и общего программного обеспечения. Эти ошибки проявляются в результате отклонения характеристик функционирования программного обеспечения в системе и характеристик взаимодействующих объектов от предполагаемых при проектировании. При автономной и начальной отладке программ доля системных ошибок невелика (примерно 10% [6]), но она существенно возрастает (до 35 – 40%) на завер-

шающих этапах комплексной отладки. В процессе эксплуатации системные ошибки являются преобладающими;

– ***алгоритмические ошибки***. К ним относятся ошибки детального проектирования, спецификации оборудования и общего ПО, а также ошибки в спецификации языка программирования. К алгоритмическим ошибкам в значительной мере относятся и ошибки, выявленные при модификации программы и связанные с изменениями в алгоритме при корректировке существующих или добавлении новых функций. Алгоритмические ошибки – это, в первую очередь, ошибки, обусловленные некорректной постановкой функциональных задач, когда не полностью оговорены условия, необходимые для получения правильного результата. Они являются наиболее частыми и составляют около 70% всех алгоритмических ошибок. Около 20% алгоритмических ошибок составляют ошибки сопряжения функциональных подпрограмм. Оставшаяся часть алгоритмических ошибок приходится на просчеты в использовании ресурсов компьютерных сетей и манфреймов в информационных системах;

– ***программные ошибки.*** К ним относятся ошибки кодирования программ, ошибки в спецификации оборудования, общего ПО и языка программирования, а также ошибки, возникшие при модификации ПО.

В работе [8] систематизированы формы проявления программных ошибок. Их можно подразделить на следующие пять групп:

1. Ошибки управления:

- зацикливание – бесконечное повторение одной и той же части программы;
- нарушение последовательности прохождения участков программы;
- обращение к запрещенной области памяти, попытки выполнить запрещенную программу и др.

2. Ошибки в выполнении арифметических операций и стандартных функций:

• деление на 0;

• расхождение результатов арифметических операций с ожидаемыми значениями;

• обращение к стандартным функциям с недопустимыми значениями параметров.

3. Ошибки переполнения:

• переполнение разрядной сетки;

• переполнение целого числа;

• переполнение буфера, в том числе переполнение стека и «кучи» (динамически выделяемой во время выполнения приложения области памяти).

4. Ошибки ввода/вывода:

• несанкционированные выводы на печать, монитор и т.д.;

• несанкционированные сообщения об ошибках от системных программ ввода/вывода.

5. *Ошибки, приводящие к прекращению решения функциональных задач, а также ошибки, приводящие к искажениям результатов, выдаваемых управляющим алгоритмом:*

• останов информационного процесса;

• искажение или потеря накопленной информации о текущем состоянии управляемого процесса;

• прекращение или снижение темпа решения некоторых задач вследствие перегрузки процессора по производительности;

• искажение процессов взаимного прерывания подпрограмм, приводящих к блокировке возможности некоторых типов прерываний;

• пропуск подпрограмм или их существенных частей;

• обработка ложных или искаженных сообщений;

• выход на подпрограммы или их части, искажающие результаты выполнения процесса.

На начальных этапах разработки и автономной отладки подпрограмм вес программных ошибок в информационных системах достигает одной трети от общего количества ошибок. К этапу эксплуатации в результате верификаций модулей, подпрограмм и программ ПО вес программных ошибок вначале падает в два раза, а затем в процессе эксплуатации ИС устанавливается на постоянном уровне примерно в (3-5)% от общего числа ошибок [7].

II.1.5. Ошибки человека – оператора

В реальных условиях в большинстве систем независимо от степени их автоматизации требуется в той или иной мере участие человека. Технические системы становятся взаимосвязанными только благодаря наличию такого основного звена, как человек. Можно утверждать, что там, где работает человек, появляются ошибки [9]. Они возникают независимо от уровня подготовки, квалификации или опыта. Примерно 20-30 % отказов прямо или косвенно связаны с ошибками человека; 10-15 % всех отказов непосредственно связаны с ошибками человека. Ошибка человека определяется как невыполнение поставленной задачи (или выполнение запрещенного действия), которое может явиться причиной повреждения оборудования или имущества либо нарушения нормального хода запланированных операций.

Ошибки по вине человека могут возникнуть в тех случаях, когда:

- оператор или какое-либо лицо стремится к достижению ошибочной цели;
- поставленная цель не может быть достигнута из-за неправильных действий оператора;
- оператор бездействует в тот момент, когда его участие необходимо.

Виды ошибок, допускаемых человеком на различных стадиях взаимодействия в системе “человек – машина” можно классифицировать следующим образом.

1. *Ошибки проектирования*: обусловлены неудовлетворительным качеством проектирования. Например, управляющие устройства и индикаторы могут быть расположены настолько далеко друг от друга, что оператор будет испытывать затруднения при одновременном пользовании ими.

2. *Операторские ошибки*: возникают при неправильном выполнении обслуживающим персоналом установленных процедур или в тех случаях, когда правильные процедуры вообще не предусмотрены.

3. *Ошибки изготовления*: имеют место на этапе производства вследствие: (а) неудовлетворительного качества работы, например неправильной пайки, (б) неправильного выбора материала, (в) изготовления изделия с отклонениями от конструкторской документации.

4. *Ошибки технического обслуживания*: возникают в процессе эксплуатации и обычно вызваны некачественным ремонтом оборудования или неправильным монтажом вследствие недостаточной подготовленности обслуживающего персонала, неудовлетворительного оснащения необходимой аппаратурой и инструментами.

5. *Внесенные ошибки*: как правило, это ошибки, для которых трудно установить причину их возникновения, т.е. определить, возникли они по вине человека или же связаны с оборудованием.

6. *Ошибки контроля*: связаны с ошибочной приемкой как годного элемента или устройства, характеристики которого выходят за пределы допусков, либо с ошибочной отбраковкой годного устройства или элемента с характеристиками в пределах допусков.

7. *Ошибки организации рабочего места*: теснота рабочего помещения, повышенная температура, шум, недостаточная освещенность и т.п.

8. *Ошибки управления коллективом*: недостаточное стимулирование специалистов, их психологическая несовместимость, не позволяющие достигнуть оптимального качества работы.

Свойство человека ошибаться является функцией его психофизиологического состояния. Интенсивность ошибок во многом определяется параметрами внешней среды, в которой человек работает.

Самое распространенное рабочее определение ошибки человека-оператора – это ненамеренное отклонение выполнения его действия от стандарта. Ошибка – это результат действия, совершенного неточно или неправильно, вопреки плану, но самое главное, что результат, который получен, не соответствует намеченному или заданному, требуемому. В терминах этого определения ошибки должно существовать четыре нормативных типа выполнения действия. Этим типам как раз и соответствуют типы выполнения действий, обсуждаемые в инженерно-психологической (human factors engineering) литературе [10]:

- правильные (идеальные) действия, приводящие к оптимальным результатам;
- латентные ошибки или отклонения, находящиеся в пределах допусков и, возможно, даже не замечаемые оператором, благодаря удовлетворительным результатам и способностям адаптации к действиям оператора и способностям фильтрации некоторых типов ошибок оператора;
- скорректированные ошибки или отклонения, выходящие за пределы допусков, замеченные и исправленные оператором во избежание отрицательных последствий;
- собственно ошибки или отклонения, либо не исправленные, либо неисправляемые, но в обоих случаях имеющие неудовлетворительные результаты и отрицательные последствия.

Ошибки, имеющие неудовлетворительные результаты и отрицательные последствия могут возникнуть в тех случаях, когда:

- оператор стремится к достижению ошибочной цели;
- поставленная цель не может быть достигнута из-за неправильных действий оператора;
- оператор бездействует в тот момент, когда его участие необходимо.

Деятельность операторов информационных систем характеризуется значительной динамикой, высоким уровнем информационных воздействий, временной неопределенностью, дефицитом времени оперативного вмешательства, высокой ответственностью и высокой «ценой» последствий этого вмешательства. Деятельность операторов сопровождается изменением состояния их физиологической и психологической систем. С одной стороны, условия гиподинамии и монотонности приводят к снижению функционального тонуса этих систем. Физиологические обследования, проведенные на операторах в производственных условиях, свидетельствуют о понижении артериального давления, понижении частоты дыхания и частоты сердечных сокращений. С другой стороны, необходимость вмешательства в управление при внезапном возникновении значимого рассогласования в системе заставляет оператора постоянно поддерживать на достаточно высоком уровне состояние готовности к экстренному действию. Самостоятельная регуляция готовности к экстренному действию в условиях монотонности и гиподинамии достигается ценой значительных нервных нагрузок и психического напряжения.

В качестве примера приведем причины типичных ошибок операторов информационных систем сортировочных станций на железнодорожном транспорте. К ним относятся: дефицит времени и информации для принятия правильного решения, перегрузка операторов, утомление, а также недостаточная их

квалификация. Действия оператора в период подготовки к роспуску характеризуются невысокой моторной и большой информационной загрузкой. При контроле за ходом программного роспуска основная загрузка горочного оператора – сенсорная, с переработкой в ускоренном темпе большого количества информации, с элементами прогнозирования. В то же время, программно – контрольный режим занимает по времени около 75% всей загрузки оператора и характерен наличием строго определенных действий, повторяющихся при скатывании каждого отцепа с известной закономерностью. Большинство действий оператора при сбоях средств информационной техники характеризуется необходимостью высокой скорости и точности выполнения предусмотренных действий при довольно сложных алгоритмах принятия и реализации решений. Загрузка оператора здесь характеризуется преобладанием (до 70%) оценочных и логических действий над управляющими при дефиците времени (4-6 сек.). При завершении роспуска оператор должен хранить в памяти большое число технологических ситуаций.

Сложные и довольно противоречивые требования к личностным способностям операторов приводят к тому, что по их вине даже при нормальном ходе роспуска возникают следующие нарушения [11]: неправильное торможение отцепов – 20%, несогласованность действий операторов – 15%, ошибки при реализации роспуска и считывании информации – 20%, неправильное управление стрелками – 15%, технологические ошибки – 15% от общего числа ошибок оперативного персонала. Данный пример свидетельствует о том, что оператор в информационно-управляющих системах становится одним из слабых звеньев.

Вместе с тем, в отличие от технических средств каждый оператор в информационной системе – это высокоинтеллектуальное звено системы, способное адаптироваться к изменяющимся условиям функционирования, способное парировать внешние

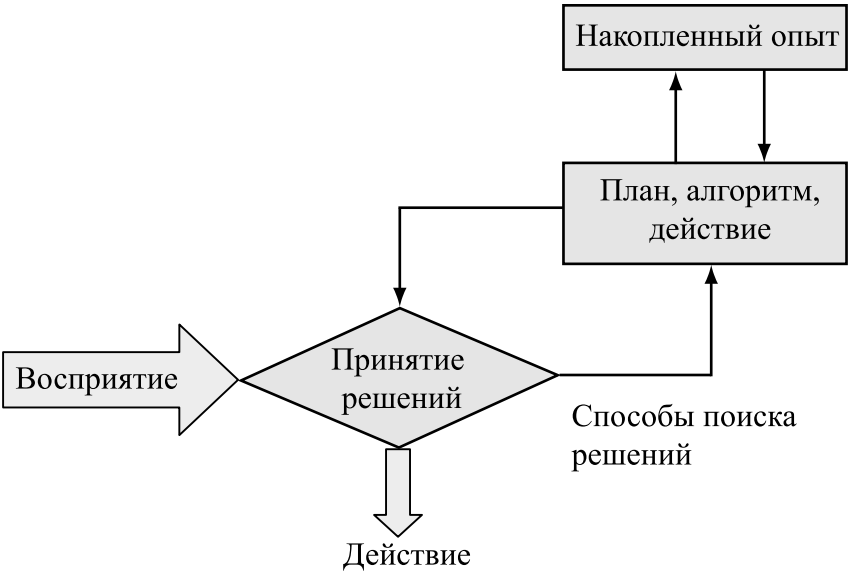


Рис. II.1.6. Познавательно-логическая цепочка обработки информации человеком

возмущающие воздействия. Оператор принимает решение на последующие действия как на основании воспринятой информации, так и с учетом накопленного ранее опыта (рис. II.1.6). Оперативное формирование плана и алгоритма действий оператора все в большей степени зависит от накопленного опыта. Это особенно характерно для операторов с высоким уровнем квалификации и обученности.

При эргономическм анализе выделяют пять классов операторской деятельности [12]:

1. Оператор-технолог. Он непосредственно включен в технологический процесс, работает в режиме немедленного обслуживания, совершает преимущественно исполнительные действия, руководствуясь при этом инструкциями, содержащими, как правило, полный набор ситуаций и решений. Основными в его деятельности являются функции формального перекодирования и передачи информации.

2. Оператор-манипулятор. К числу функций такого оператора относится управление манипуляторами, роботами, машинами – усилителями мышечной энергии.

3. Оператор-наблюдатель, контролер. К ним относятся операторы слежения радиолокационных станций, диспетчеры энергетических, транспортных систем и т.п. Это классический тип оператора, наиболее исследованный и описанный в литературе. Для него характерен большой объем информационных потоков. Он может работать как в режиме немедленного, так и в режиме отсроченного обслуживания.

4. Оператор-исследователь. Для него характерно использование аппарата понятийного мышления и опыта, заложенных в образно-концептуальных моделях. К числу таких операторов относятся пользователи вычислительных систем, программисты, дешифровщики объектов или изображений и т.д.

5. Оператор-руководитель. Он управляет не техническими компонентами системы или машины, а другими людьми. Это управление может осуществляться как непосредственно, так и опосредствованно – с помощью технических средств.

В информационных системах наиболее характерны первый и третий классы операторской деятельности, реже – четвертый и пятый классы.

Операторы – технологи взаимодействуют с вычислительными средствами чаще всего с помощью видеотерминальных устройств. Безошибочность действий операторов зависит от многих рассмотренных выше факторов. Большое влияние на безошибочность работы этого типа операторов оказывают степень обученности и скорость поступления информации. Степень обученности оператора обычно характеризуется частотой промахов. Оператор считается обученным, когда частота промахов достигает установившейся величины. В процессе обучения частота промахов уменьшается.

В работе ***оператора – наблюдателя – контролера*** наиболее часто встречаются ошибки обнаружения, ошибки в ориентации (недостаток информации, избыток информации), снижении скорости при восприятии смысла информации. Воспринятая человеком информация преобразуется. При этом человек старается перевести эту информацию на алфавит собственного языка и идентифицировать с собственным опытом. Словарный запас человека составляет до 15 тыс. слов, тогда как словарный запас языка намного больше (например, русского – около 100 тыс. слов). Переводя запоминание на собственный алфавит (используя слова – синонимы), оператор облегчает себе запоминание. Известно, что часть хранимой в человеческом мозгу информации забывается. Еще в 1885 г. психолог Эббингауз установил, что усвоенная информация наиболее быстро уменьшается за первые 9 ч. – до 30% от первоначального объема.

Для ***операторов – исследователей, операторов – руководителей*** характерны ошибки в восприятии, анализе, обобщении и систематизации информации, оценке ситуации и принятии решения, нечеткость в формулировке цели деятельности, сложность в построении системы логических доказательств, а также ошибки в прогнозировании развития ситуации, недостатки в оценке своих решений, ошибки управления подчиненными и др. Наиболее существенны ошибки в принятии решений. Принятие решения обычно рассматривается как многокритериальный выбор из нескольких альтернатив. К сожалению, соответствующие модели не учитывают характерное свойство человеческого разума: изменчивость критериев, изменчивость точки зрения при переходе в новые диапазоны условий. Не учитывается также непостоянство альтернатив – способность изобретать новые, более правильно отражающие реальность гипотезы, которая также является характерным свойством человека. Поэтому су-

ществующие формальные теории пока еще далеки от реальной действительности, вследствие чего в информационно – управляющих системах автоматизация сводится лишь к подготовке количественных данных и рекомендаций, используемых человеком – оператором при принятии решений.

II.1.6. Ошибки данных

II.1.6.1. Свойства данных в информационных системах

Данные являются продуктом информационной системы. Совокупность их свойств, определяющую пригодность выполнять предусмотренные информационные технологии, принято называть *качеством данных*. Количественные характеристики этих свойств являются *показателями качества данных*.

С точки зрения пользователя качество данных определяется выполнением следующих условий: наличие данных у пользователя в необходимый момент времени и совпадение (в пределах требуемой детализации и степени точности) информационной модели отображаемого явления с реальной действительностью. Кроме того, в информационных системах, связанных с обработкой данных конфиденциального характера или составляющих коммерческую тайну пользователя, требуется защита данных от несанкционированного доступа как со стороны внешних, так и внутренних нарушителей. Кроме перечисленных основных условий, возможны и другие дополнительные условия качества данных. Анализ причин невыполнения указанных условий (ошибки, искажения, потери данных, их запаздывание и др.) приводят к выделению ряда свойств данных, определяющих их качество.

К общим свойствам данных по аналогии с работой [13] можно выделить ***внутренние свойства*** (*достоверность и куму-*

лятивность данных), сохраняющиеся при переносе в другую систему, и ***внешние свойства*** (*временные и защищенность данных)*, которые характерны для данной системы и исчезают при переносе в другую систему.

Под *достоверностью данных* понимается их свойство не иметь скрытых ошибок. В свойстве достоверности выделяется ***безошибочность*** данных (техническая составляющая достоверности) и их ***истинность*** (социально – психологическая составляющая). При анализе *безошибочности* данных рассматриваются случайные ненамеренные искажения, случаи трансформации, недопустимых отклонений или потерь данных вследствие сбойных, программных ошибок, ошибок операторов, ошибок во входных сообщениях. При анализе *истинности* данных рассматриваются намеренные искажения данных человеком – источником сведений (в том числе из – за непонимания сути вопроса). Ранее отмечалось, что информационные системы решают строго определенные задачи, т.е. реализуют хорошо отработанные, доведенные для практического применения информационные технологии. Поэтому сомнения в истинности данных в таких системах в основном отсутствуют. Данное свойство можно при анализе надежности не учитывать еще и по причине того, что предметом анализа функциональной надежности не являются деструктивные действия внешних или внутренних нарушителей. Следовательно, в информационных системах достоверность данных отождествляется со свойством безошибочности.

Под *кумулятивностью данных* понимается свойство данных небольшого объема достаточно полно отображать действительность. Под этим свойством подразумеваются результаты сжатия данных, фильтрации данных, отбора данных, агрегирования данных и т.д. В соответствии с определением кумулятивности

для ее обеспечения важно уменьшить объем перерабатываемых данных, обеспечив пользователя всеми необходимыми сведениями. Это позволит исключить избыточный объем недостаточно систематизированной информации, поставляемой лицу, принимающему решения (ЛПР), и снизить информационную нагрузку на информационную технику. Свойство кумулятивности данных характеризует их *достаточность* для ЛПР и для управления подчиненными объектами. Это свойство данных относится к категории *качества данных*, а все события нарушения этого свойства вследствие влияния ошибок в выполнении предусмотренных процедур являются предметом изучения безошибочности данных.

Под *временными свойствами данных* в информационных системах понимается, в первую очередь, *своевременность* сбора, обработки и передачи информации подчиненным объектам и/или ЛПР. Своевременность данных включает в себя такие характеристики как: *оперативность* – свойство данных, состоящее в том, что время их сбора, обработки и передачи соответствует динамике изменения ситуации; *устойчивость* – свойство данных соответствовать состоянию объекта и сохранять ценность для потребителя с течением времени. Временные свойства данных определяются, в основном, архитектурой системы, а также возможностями информационной среды, в которой осуществляется обмен данными между системой и внешними объектами. Архитектура информационной системы и ее информационная среда сохраняются относительно неизменными в течение всего жизненного цикла системы, требования к временным свойствам данных обеспечиваются в процессе проектирования таких систем. Любые нарушения временных свойств данных, вызванные ошибками в процессе сбора, обработки, передачи информации, относятся к свойству безошибочности данных в информационной системе.

Таким образом, при анализе функциональной надежности информационной системы в качестве определяющего свойства данных учитывается свойство их безошибочности. Ошибки данных могут непосредственно негативно повлиять на другие свойства данных, а именно на кумулятивность данных, на их оперативность и устойчивость. В свою очередь, последние могут косвенно и не существенно проявиться в уровне безошибочности информации.

II.1.6.2.Ошибки во входных сообщениях

Эти ошибки вызваны нарушениями целостности потока сообщений [14]. Поток сообщений определяется как упорядоченный набор сообщений и является уникальным для каждого интервала времени и получателя в сети.

Фактически принимаемый поток сообщений может отличаться от ожидаемого по ряду причин:

- принято большее число сообщений, чем ожидалось;
- принято меньшее число сообщений, чем ожидалось;
- число принятых сообщений равно числу ожидаемых сообщений, но при этом имели место нарушения следования, времени передачи или содержания сообщений.

Принято большее число сообщений, чем ожидалось

В этом случае имело место повторение одного или нескольких сообщений, или же в канал связи было введено постороннее сообщение извне. Поэтому в данном случае основные ошибки в сообщениях следующие: *повторение сообщения, ввод сообщения.*

Принято меньшее число сообщений, чем ожидалось

В этом случае имело место пропадание одного или нескольких сообщений. Поэтому в данном случае основная ошибка в сообщении следующая: *пропадание сообщения.*

Число принятых сообщений равно числу ожидаемых сообщений

В этом случае существует несколько возможностей:

- все сообщения в потоке правильны по содержанию и своевременны по доставке, но порядок следования их неверен: имело место изменение порядка следования сообщений;

- время доставки получателю сообщения в потоке оказалось больше номинального, имела место задержка сообщения;

- сообщение было изменено: имело место искажение сообщения;

- получатель считает, что отправитель сообщения не тот, который должен быть в действительности: имела место имитация сообщения.

В двух последних случаях рассматривается целостность одиночного сообщения. Основные ошибки в сообщениях для всех этих случаев будут следующие: *изменение порядка следования сообщений, задержка, искажение* и *имитация* (“маскарад”).

Приведенные выше основные ошибки в сообщениях не являются взаимно-исключающими: возможно возникновение ошибки в нескольких сообщениях потока, а также возникновение нескольких ошибок в одном сообщении.

II.1.7. Систематические ошибки и отказы по общей причине

В п.II.1.1 рассмотрена упрощенная модель трансформации ошибок информационных процессов в ошибку в выполнении процесса и возникновение возможного функционального отказа. Упрощенность данной модели заключается в том, что она иллюстрирует взаимосвязь процессов различных уровней иерархии, но не отражает реальное влияние среды, в которой эти процессы выполняются. Речь идет о том, что любое действие, любая процедура, любой информационный процесс реализуется с помощью средств информационной техники, которые в

свою очередь могут оказывать возмущающие воздействия на информационные процессы. Эти воздействия проявляются в результатах выполнения процессов в виде случайных или систематических отказов. В первом случае предполагается, что отказы любого компонента структуры ИС, участвующего в выполнении данного процесса, возникают случайно по времени и приводят к случайной ошибке процесса. Это в равной мере относится и к сбоям этого или любого другого средства информационной техники, участвующего в выполнении данного процесса.

Вместе с тем, возможны возмущающие воздействия на каждый информационный процесс, вызванные так называемыми систематическими отказами компонентов структуры ИС. Эти отказы дополняют список тех, которые перечислены нами в главе 1, и отражают роль человека в ненадежности техники. Причины этих отказов в следующем:

– ошибки в проектировании компонентов структуры ИС;

– ошибки в обеспечении климатических режимов работы компонентов;

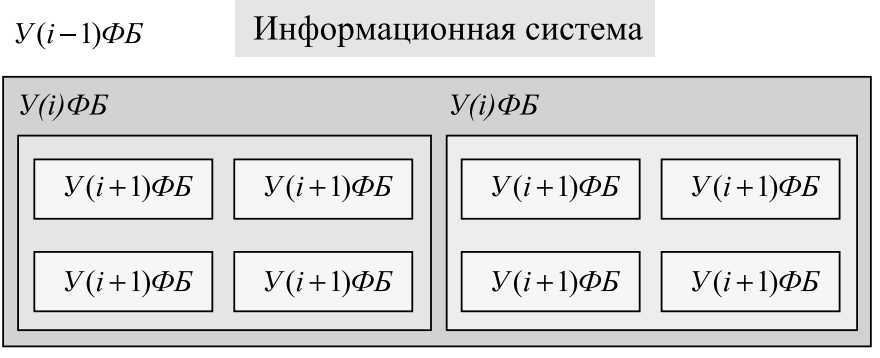
– ошибки в обеспечении их помехозащищенности;

– ошибки обслуживающего персонала и др.

Эти и другие систематические отказы компонентов структуры ИС являются источниками систематических ошибок в выполнении информационных процессов. Термин *систематическая ошибка* введен в конце 90-х годов 20-го столетия разработчиками серии технических стандартов по функциональной безопасности программируемых электронных устройств и систем [15]. Назначение этого термина состоит в том, чтобы подчеркнуть проблемность применения традиционных методов теории структурной надежности к моделированию этой группы ошибок. Это объясняется тем, что отсутствует случайность в возникновении такого рода ошибок и, наоборот, присутству-

ет определенная систематичность проявления первопричин их возникновения.

Определяющее влияние на формирование систематических ошибок оказывают ошибки в программном обеспечении. Ранее в п.1.3 отмечалось, что программные ошибки есть результат создания и корректировки программного обеспечения. Они могут не проявляться, если входные данные не инициируют выполнение тех участков программ, в которых содержатся ошибки. В отличие от технических средств некоторые входные потоки данных могут привести к частым ошибкам, в то время как другие потоки могут вообще не привести к ошибкам. Наличие дефектов в программах не обязательно приводит к программной (систематической) ошибке. Может пройти много времени, прежде чем дефектный код («жук» – термин, принятый в среде инженеров – программистов) может быть задействован при обстоятельствах, приводящих к систематической ошибке. Этот дефект может вызвать ошибку, которая произойдет при достижении в системе некорректного состояния, т.е. когда будет использован хотя бы один бит, отражающий некорректное значение в памяти или в шине.



У – уровень иерархии (i=1,2,3,...); ФБ – функциональный блок

Рис. II.1.7. Конфигурация функциональных блоков

Таким образом, на каждом уровне иерархии информационный процесс реализуется программно – аппаратно. Программно – аппаратное устройство (систему) назовем функциональным блоком. Как показано на рис. II.1.7, функциональный блок может быть представлен в виде многоуровневой иерархической конструкции, каждый из уровней которой может быть назван функциональным блоком. Взаимосвязь ошибок и функциональных отказов показана на рис. II.1.8. На уровне *i* «причина» может проявить себя как ошибка (отклонение от правильного значения или состояния) в пределах функционального блока, соответствующего уровню *i*. Если она не будет исправлена или нейтрализована, то эта ошибка может при определенных условиях привести к отказу функционального блока, который в результате перейдет в состояние Ф невозможности более выполнять предусмотренную функцию. Данное состояние Ф уровня *i*, в свою очередь, может проявиться в виде ошибки на уровне *i-1*, которая, если не будет исправлена или нейтрализо-

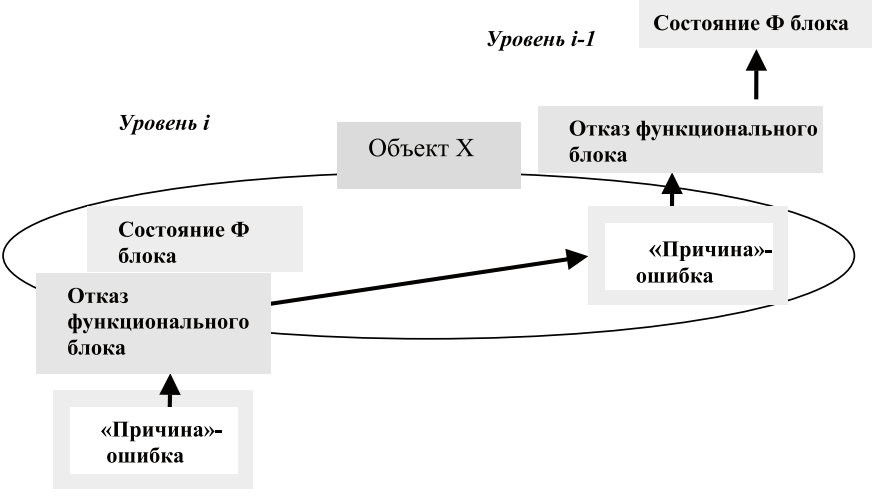


Рис. II.1.8. Концептуальное представление взаимосвязи ошибки и отказа функционального блока

вана, может привести к отказу функционального блока уровня *i-1*. В этой причинно – следственной цепочке один и тот же элемент («объект Х») может рассматриваться как состояние Ф функционального блока уровня *i*, в которое он попадает в результате отказа, а также как причина отказа функционального блока уровня *i-1*.

Итак, мы рассмотрели взаимосвязь ошибок и функционального отказа относительно одной функции ИС (одного информационного процесса). Реально в системе выполняется множество функций, причем некоторые из них с помощью общих функциональных блоков. Ошибки и, тем более, функциональные отказы этих блоков могут быть общей причиной того, что несколько информационных процессов (функций системы) выполнены неверно. Такие ошибки (или отказы функциональных блоков) будем называть *ошибками или функциональными отказами по общей причине (ООП*).

В общем случае под ООП понимаются такие ошибки, которые оказывают одновременное влияние на несколько каналов (в многоканальной ИС), или на несколько функций, или на избыточные устройства или подсистемы. Так, например, возникновение ошибки в операционной системе, повлекшее за собой ошибки в выполнении одной или нескольких прикладных программ. В типичных случаях источниками ООП являются:

– неполнота или противоречивость требований;

– наличие недостаточно надежных общих источников питания;

– производственные недостатки, общие для партии компонентов;

– циклические изменения температуры, электрические наводки, промышленные помехи и т.д.,

– ошибки операторов – диспетчеров и др.

II.1.8. Функциональные отказы вследствие атак на информационную систему

Атака на информацию – что это такое? Дать определение этому действию на самом деле очень сложно, поскольку информация, особенно в электронном виде, представлена сотнями различных видов. Информацией можно считать и отдельный файл, и базу данных, и одну запись в ней, и целиком программный комплекс. И все эти объекты могут подвергнуться и подвергаются атакам со стороны некоторой социальной группы лиц, которые действуют как злоумышленники.

Атака – это совокупность действий злоумышленника, приводящих к нарушению информационной безопасности ИС.

Информация с точки зрения информационной безопасности обладает следующими категориями (рис. II.1.9):

конфиденциальность – гарантия того, что конкретная информация доступна только тому кругу лиц, для кого она предна-

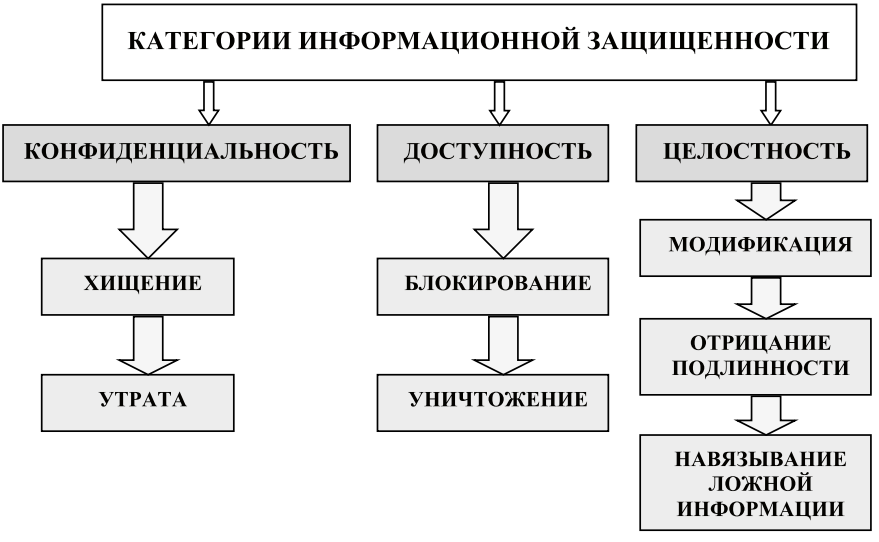


Рис. II.1.9. Категории защищенности информации в информационных системах

значена; нарушение этой категории называется хищением либо раскрытием информации;

целостность – гарантия того, что информация сейчас существует в ее исходном виде, то есть при ее хранении, обработке или передаче не было произведено несанкционированных изменений; нарушение этой категории называется фальсификацией информации (подмена данных, нарушение подлинности (модификация) данных, искажение информации);

доступность – гарантия того, что различные группы лиц имеют различный доступ к информационным объектам, эти ограничения доступа постоянно выполняются и не допускаются случаи блокирования санкционированного доступа к информации. Кроме того, для лиц с разрешенным доступом к информации обеспечивается работоспособность ИС в произвольный момент времени (готовность системы к предоставлению необходимой информации).

Результатом успешной атаки может стать нарушение функциональной надежности, заключающееся в нарушении целостности или доступности информации. В качестве целей атаки могут рассматриваться серверы, рабочие станции пользователей или коммуникационное оборудование ИС. При организации информационных атак злоумышленники часто используют специализированное ПО, позволяющее автоматизировать действия, выполняемые на различных стадиях атаки.

В общем случае в любой атаке можно выделить четыре стадии:

Рекогносцировка. На этой стадии нарушитель старается получить как можно больше информации об объекте атаки, чтобы на ее основе спланировать дальнейшие этапы вторжения. Этим целям может служить, например, информация о типе и версии операционной системы; список пользователей, зарегистрированных в системе; сведения об используемом прикладном ПО и т. д. Для получения информации о структуре и топологии сис-

темы нарушитель может воспользоваться стандартными утилитами, входящими в состав практически любой ОС, которые позволяют сформировать список IP-адресов транзитных маршрутизаторов вплоть до хоста – объекта нападения (хост – компьютер, являющийся элементом сети). Один из наиболее распространенных методов определения типа ОС основан на том, что различные системы по-разному реализуют требования, определяющие правила взаимодействия с сетевыми протоколами. При одних и тех же сетевых запросах разные ОС отправляют в ответ разные данные, и по ним можно с большой долей вероятности определить характеристики атакуемой ОС и даже тип аппаратной платформы. Информацию о типе прикладных сервисов нарушитель может получить, например, путем сканирования открытых портов и анализа заголовков ответов, полученных от этих служб Часто злоумышленники стремятся использовать либо явные промахи создателей, не заметивших возможность очень просто обойти их систему защиты, либо ошибки в реализации программ защиты. Также злоумышленники широко используют психологические приемы для того, чтобы получить нужную им информацию у рядовых сотрудников фирм.

Вторжение. На этом этапе нарушитель получает несанкционированный доступ к ресурсам тех хостов, на которые совершается атака. Обратимся к наиболее популярным и очевидным технологиям несанкционированного доступа. Рассмотрением их нельзя пренебрегать, исходя из очень простого правила: “прочность цепи не выше прочности самого слабого ее звена”. Эта аксиома постоянно цитируется, когда речь идет о компьютерной безопасности. Например, как бы ни была прочна система, если пароль на доступ к ней лежит в текстовом файле в центральном каталоге или записан на экране монитора – это уже не конфиденциальная система. Другой пример – при работе в сети Internet не существует надежного автоматического подтвержде-

ния того, что данный пакет пришел именно от того отправителя (IP-адреса), который заявлен в пакете. А это позволяет даже при применении самого надежного метода идентификации первого пакета подменять все остальные, просто заявляя, что все они пришли тоже с этого же самого IP-адреса.

Атакующее воздействие. На данной стадии реализуются те цели, ради которых и предпринималась атака, – например, нарушение работоспособности ИС, кража конфиденциальной информации из системы, удаление или модификация данных и т.д. При этом атакующий часто выполняет операции, направленные на удаление следов его присутствия в ИС. Всякая атака основана на наличии в ИС уязвимостей, и "правильное" использование хотя бы одной из них открывает злоумышленнику вход в систему. Примеров уязвимостей можно привести немало. Здесь и ошибки при конфигурировании сетевых служб ИС, и ошибки в ПО, и использование "слабых" и "нестойких" паролей, и отсутствие необходимых средств защиты, и многое другое. В результате же нарушитель получает несанкционированный доступ к ресурсам атакованного хоста, что позволяет ему перейти к следующей стадии информационной атаки.

Развитие атаки. Злоумышленник стремится расширить объекты атаки, чтобы продолжить несанкционированные действия на других составляющих ИС. После атакующего воздействия нарушитель может перевести атаку в фазу дальнейшего развития. Для этого в систему обычно внедряется программа, с помощью которой можно организовать атаку на другие хосты ИС. Такая программа называется вредоносной.

В настоящее время не существует общепринятой классификации вредоносных программ и каждая компания, разрабатывающая антивирусный «софт», использует собственные критерии и наименования для определения вредоносного ПО. Существует множество вредоносных программ различного назначения. Ос-

новные угрозы функциональной надежности информационных систем создают следующие группы вредоносных программ:

• ***DoS-атака*** (от англ. Denial of Service, отказ в обслуживании) – атака на информационную систему с целью довести её до отказа, то есть создание таких условий, при которых легитимные (правомерные) пользователи системы не могут получить доступ к предоставляемым системой ресурсам (серверам), либо этот доступ затруднён. Отказ «вражеской» системы может быть одним из шагов к овладению системой (если во внештатной ситуации ПО выдаёт какую-либо критическую информацию – например, версию, часть программного кода и т.д.).

Если DoS-атака связана с большим количеством обычно бессмысленных или сформированных в неправильном формате запросов к компьютерной системе или сетевому оборудованию, то ее принято называть флуд (англ. flood – “наводнение”, “переполнение”). Такая атака предназначена для создания отказа в работе ИС из-за исчерпания ресурсов системы – процессора, памяти либо каналов связи.

Существуют различные причины, по которым в самой ИС может возникнуть DoS-условие. Так, ошибка в программном коде приводит к обращению к неиспользуемому фрагменту адресного пространства, выполнению недопустимой инструкции или другой необрабатываемой исключительной ситуации, когда происходит аварийное завершение серверного приложения. Другая причина – недостаточная проверка данных пользователя, приводящая к бесконечному либо длительному циклу или повышенному длительному потреблению процессорных ресурсов (исчерпанию процессорных ресурсов) либо выделению большого объёма оперативной памяти (исчерпанию памяти).

Если атака выполняется одновременно с большого числа компьютеров, говорят о ***DDoS-атаке*** (от англ. Distributed Denial of Service, распределённая атака типа «отказ в обслуживании»).

• ***Троянские программы.*** Эти вредоносные программы после внедрения в систему нарушают целостность данных и программ. Они также могут собрать сведения о хранящихся на компьютере профилях пользователей, паролях и другую конфиденциальную информацию и затем переслать ее в руки злоумышленников. Такое ПО обычно называют троянами или троянскими программами. Непосредственно после запуска они становятся активными и, например, запускают форматирование жесткого диска. Дроппер является специальным видом троянской программы. Эта программа рассаживает вирусы в системе. В некоторых случаях они не только просматривают хранящуюся на жестком диске информацию, но и запоминают, на какие клавиши нажимает пользователь при работе с клавиатурой, фиксируя данные авторизации. Подобные троянские программы называются кейлоггерами. Еще один вариант действия вредоносной программы – предоставление злоумышленникам удаленного доступа к компьютеру, такое ПО называется бэкдор (от англ. backdoor – черный ход).

• ***Программы несанкционированного управления компьютерами ИС.*** Основные из них:

– *Утилиты администрирования (Backdoor).* С помощью утилит администрирования (задняя дверь, черный ход) можно, обходя системы защиты от НСД, получить компьютер под свой контроль. Программа, работающая в скрытом режиме, дает пользователю практически неограниченные права. Чаще всего эти программы используются для инфицирования системы компьютерными вирусами и установки на нее вредоносных программ.

– *Загрузочные вирусы*. Загрузочный и главный загрузочный сектор жесткого диска заботливо инфицируются загрузочными вирусами. Эти вирусы изменяют важную информацию, необходимую для запуска системы. Одно из последствий – невозможность загрузки операционной системы.

– *Bot-сеть*. Под Bot-сетью понимается удаленно управляемая сеть (в Интернет), состоящая из отдельных персональных компьютеров, коммуницирующих между собой. Контроль сети достигается с помощью вирусов или троянских программ, инфицирующих компьютер. Они ожидают дальнейших указаний злоумышленника, не принося вреда инфицированным компьютерам. Эти сети могут применяться для рассылки СПАМа или организации DDoS-атак. Пользователи участвующих компьютеров могут и не догадываться о происходящем. Основной потенциал bot-сетей заключается в том, что такие сети могут достигать численности в несколько тысяч элементов, чья совокупная пропускная способность может поставить под угрозу любую систему обработки запросов.

– *Эксплойт*. Эксплойт (брешь в безопасности) – это компьютерная программа или скрипт, использующий специфические уязвимости операционной системы или программы. Так, в информационную систему могут проникать программы, с помощью которых могут быть получены расширенные права доступа.

• *Макровирусы.* Макровирусы – это маленькие программы, написанные на макроязыке приложений (напр., WordBasic), которые распространяются только среди документов, созданных для этого приложения. Поэтому они еще называются документными вирусами. Для того, чтобы они стали активными, требуется запуск соответствующего приложения и выполнение инфицированного макроса. В отличие от “нормальных” вирусов, макровирусы инфицируют не исполняемые файлы, а документы, созданные определенным приложением-хозяином.

• *Программные вирусы*. Компьютерный вирус – это программа, обладающая способностью после своего запуска самостоятельно прикрепляться к другим программам, инфицируя их таким образом. Вирусы размножаются самостоятельно, что от-

личает их от логических бомб и троянских программ. В отличие от червя, вирусу всегда необходима программа, внутри которой он может записать свой вредоносный код.

• *Руткит* – набор программных средств, которые устанавливаются в систему, обеспечивая сокрытие логина злоумышленника, процессов и делая копии данных: то есть, делают их администратора невидимым. Вы пытаетесь обновить уже установленную шпионскую программу или установить удаленное шпионское ПО.

• *Сетевые черви.* Данную категорию вредоносных программ характеризует возможность самостоятельного распространения по локальной сети или в Интернете. Для этого «черви» используют самые разнообразные методы и каналы обмена данными такие как электронная почта, сети обмена мгновенными сообщениями, сетевые ресурсы с общим доступом, уязвимости в ОС и программах и др.

Информационные атаки могут быть классифицированы как внешние или внутренние. Внешние сетевые атаки проводятся извне ИС, т.е. с тех узлов, которые не входят в состав системы. Примером внешней сетевой атаки является вторжение нарушителя в локальную вычислительную сеть ИС из сети Интернет. Внутренние атаки проводятся изнутри ИС с одного из её серверов или рабочих станций. В качестве примера такой атаки можно привести действия обиженного сотрудника компании, направленные на нарушение целостности информации.

Контрольные вопросы

1. Что является объектом исследований функциональной надежности?

2. Покажите причинно – следственные связи, приводящие к ошибкам в выполнении информационного процесса.

3. Приведите определение функционального отказа.

4. Что понимается под сбоем функционального характера и сбойной ошибкой?

5. В чем состоят принципиальные отличия между надежностью программ и надежностью технических средств?

6. Опишите модель процессов возникновения ошибок в ходе разработки ПО.

7. Опишите пять групп проявления программных ошибок.

8. Перечислите случаи, когда ошибки оператора приводят к серьезным негативным последствиям.

9. Опишите свойства данных в информационной системе.

10. Приведите причины, по которым фактически принимаемый поток сообщений может отличаться от ожидаемого потока сообщений.

11. Объясните понятие «отказ по общей причине».

12. Что означает термин «систематическая ошибка» и к какой области надежности (структурной или функциональной) он относится?

13. Назовите и поясните категории безопасности информации.

14.Какие стадии информационной атаки вам известны и в чем они заключаются?

15.Поясните суть вредоносных программ вида *DoS-атака* и *DDoS-атака.*

16. Поясните назначение *троянских программ.*

17. Перечислите и поясните назначение основных вредоносных программ *несанкционированного управления компьютерами ИС.*

Глава II.2. Методы расчета функциональной надежности

II.2.1. Требования и принципы формирования системы показателей функциональной надежности

Информационные системы (ИС) активно развиваются. Примерно с такой же активностью различные исследователи предлагают новые показатели надежности и эффективности функционирования этих систем. Результат скорее отрицательный, поскольку обилие показателей и разноголосица в подходах не способствуют объединению специалистов в создании функционально надежных ИС. Причины этого явления имеют как объективный, так и субъективный характер. Объективные причины связаны с тем, что современные ИС обладают громадными возможностями и непредсказуемым поведением. Поэтому попытки исследователей измерить свойства функционирования этих систем традиционными мерками теории надежности не приводят к успеху. Отсюда и возникает стремление приспособить эти мерки к информационным системам путем ввода дополнительных количественных показателей. Субъективные причины увеличения числа показателей обусловлены тем, что научные и практические интересы тех или иных исследователей ограничены рамками отдельных объектов, входящих в состав ИС. Ими могут быть пакеты прикладных программ или каналы передачи информации, или операционные системы, или терминалы и др. Применительно к каждому объекту у со-

ответствующих исследователей сформировались свои взгляды на надежность, привычки в подходах, стремление к преемственности.

По нашему глубокому убеждению прежде, чем компилировать существующие показатели или предлагать систему показателей функциональной надежности ИС, следует вначале выработать требования к этой системе показателей и установить минимально необходимые принципы ее формирования.

II.2.1.1. Требования к системе показателей

Система показателей функциональной надежности, как и любая система показателей качества технических систем, должна отвечать ряду естественных требований:

1) Каждый показатель функциональной надежности должен быть измерим.

2) Показатель функциональной надежности должен допускать возможность экспериментальной проверки во время специально проводимых испытаний или в процессе реальной эксплуатации информационной системы.

3) Система показателей должна отражать дискретность случайных процессов возникновения сбойных и внешних ошибок, проявления ошибок собственных программных средств, а также заявок на выполнение информационных процессов и др.

4) Система показателей должна быть удобной в практическом применении, наглядной и сравнимой.

5) Каждый показатель должен быть простым в физическом смысле и естественным с точки зрения оценки выполняемых в информационной системе функций.

6) Показатели функциональной надежности системы должны иметь единую количественную меру расчета надежности выполнения процессов на всех уровнях их иерархии.

7) Система показателей должна быть достаточно гибкой, чтобы обеспечивать свертывание модулей расчета от низшего к высшему уровню.

8) Система показателей должна обеспечивать комплексную оценку функциональной надежности информационной системы в условиях проявления всех видов угроз (см. главу II.1).

9) Система показателей должна содержать как единичные, так и комплексные показатели. При этом единичные показатели предназначены для анализа безошибочности однократного выполнения информационных процессов, а комплексные показатели – для оценки достигнутого уровня функциональной надежности информационной системы. Расчет единичных показателей должен базироваться на критерии ошибки. Расчет комплексных показателей должен базироваться на критерии функционального отказа.

При первом ознакомлении с приведенными требованиями проявляются кажущиеся противоречия между отдельными требованиями. Например, между требованиями 3) и 4), между 5) и 8) и др. В действительности выполнение требования 6) о применении единой количественной меры на всех уровнях иерархии позволит существенно упростить расчеты при сохранении требуемых границ математической строгости. В противном случае от исследователей потребуются большие усилия, чтобы свести концы с концами, т.е. состыковать между собой различные количественные меры ошибок в программах, сбойных ошибок и ошибок во входной информации, от совместного влияния которых зависит безошибочность выполнения соответствующего информационного процесса. Например, сбойные ошибки измеряются их интенсивностью, ошибки во входной информации – вероятностью ошибки в передаваемом символе или сообщении, а надежность программы – средним временем до проявления очередной программной ошибки.

В связи с этим возникает множество вопросов: какая длительность реализации программы, какова вероятность заявки на выполнение этой программы, как влияют сбои аппаратуры на безошибочность выполнения программы и др. Все эти вопросы естественны. Действительно, если какие-либо информационные средства на определенном отрезке времени (пусть даже незначительном) не привлекаются для выполнения программы, то их сбоями, существовавшими на данном интервале времени можно и нужно пренебречь. Поэтому мера интенсивности сбойных ошибок может носить только локальный характер. Для учета влияния сбойных ошибок на безошибочность выполнения информационного процесса целесообразнее выбрать какой-либо другой показатель, например, безошибочность формирования информационным средством выходных результатов. Приведенный пример подтверждает необходимость придерживаться единой методологии в формировании показателей безошибочности выполнения информационных процессов.

II.2.1.2. Принципы формирования показателей

Информационные системы – это сложные дискретные системы со всеми присущими им особенностями:

– динамично изменяющейся структурой в зависимости от характеристик потоков заявок (запросов) и организации их обслуживания, содержания алгоритмов реализуемых информационных процессов;

– неодновременной работой составных дискретных устройств при выполнении предусмотренных операций;

– нарушениями правильности выполнения последующих процессов при возникновении ошибок в результатах выполнения предыдущих процессов;

– трансформацией сбоев дискретных устройств в ошибки комплекса программ и т.д.

С учетом этих особенностей целесообразно руководствоваться следующими принципами формирования системы показателей:

1. Принцип соответствия структуры алгоритма расчета структуре алгоритма функционирования исследуемой информационной системы при выполнении каждой функции.

Алгоритм функционирования при выполнении любой функции (процесса) состоит из набора подпроцессов, процедур, операций, микроопераций, логических действий. Все эти действия и операции структурно и информационно связаны между собой. Алгоритм расчета функциональной надежности ИС относительно конкретной задачи должен быть адекватен алгоритму функционирования ИС при выполнении этой задачи.

Основой для описания алгоритма расчета являются (рис. II.2.1):

– исходные объекты (информация), представляющие собой: характеристики сбоев составных дискретных устройств, ошибок во входных сообщениях, собственных ошибок в программах, характеристики средств диагностики, оперативного контроля и восстановления информационного процесса, а также характеристики отказов и восстановлений аппаратуры ИС. Исходные объекты записываются системой знаков A;

– результирующие объекты – показатели безошибочности выполнения на аппаратуре ИС информационных процессов описываются системой знаков P.

2. Принцип выделения в алгоритме расчета иерархических уровней, объединенных в иерархическую структуру.

На рис. II.2.1 показана иерархическая структура алгоритма расчета, в которой системой знаков Φ_7, Φ_6 обозначены логические функции и структуры микроопераций, непосредственно реализующихся на средствах информационной техники ИС. Системой знаков $\Phi_5, \Phi_4, \Phi_3, \Phi_2$ обозначены функции связи

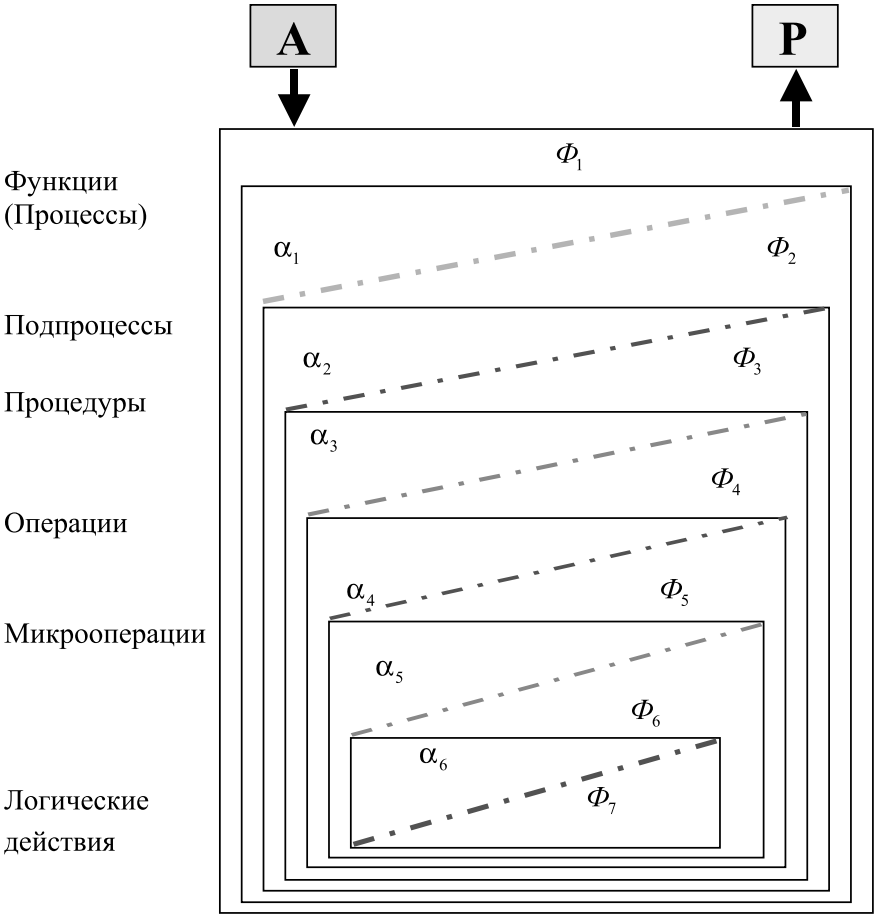


Рис. II.2.1. Иерархическая схема алгоритма расчета функциональной надежности информационных систем

соответственно между микрооперациями в пределах операции, между операциями в пределах процедуры, между процедурами в пределах подпроцесса, между подпроцессами в пределах процесса (функции) и, в конечном итоге, функции связи между задачами Φ_1. Системой знаков $\alpha_6, \alpha_5, ..., \alpha_1$ обозначен набор процессов – операторов (например, набор процедур, набор процессов), которые связаны между собой функциями $\Phi_k (k = 1, ..., 7)$.

3. Принцип совмещения пространства состояний и пространства событий исследуемого процесса.

Пространство состояний – это конечное множество последовательных или последовательно – параллельных действий (например, процедур), осуществляемых в ИС при выполнении конкретного процесса. Пространство событий – это конечное множество фактов возникновения или отсутствия ошибок в результатах действий. Количество событий всегда равно количеству действий независимо от меры детализации исследуемого информационного процесса.

Из данного принципа следует:

• Если детерминированный процесс *i*-го уровня иерархии выполняется с помощью *m* несовместных процессов (действий) *i+1*-го уровня иерархии, описываемых знаками α_{i+1}^{k} таких, что $\alpha_i^k = f(\alpha_{i+1}^1,...,\alpha_{i+1}^m)$, то вероятность безошибочного выполнения данного процесса не зависит от характера связей составных действий на *i+1* уровне иерархии и определяется выражением

$$P_k^{(i)} = \prod_{k=1}^{m} P_k^{(i+1)},$$

где $P_k^{(i+1)}$ – вероятность безошибочного выполнения *k-го* подпроцесса *i+1-го* уровня иерархии. Эта формула справедлива как для зависимых, так и независимых процессов предыдущего уровня, участвующих в выполнении данного процесса при условии из несовместности. Доказательство справедливости этой формулы основывается на принципе двойственности операций сложения и умножения несовместных событий.

• Распараллеливание процесса *i*-го уровня иерархии не приводит к уменьшению количества составных процессов *i+1*-го уровня, так как в противном случае это привело бы к нарушению алгоритма данного процесса. Более того, для распаралле-

ливания нужны дополнительные (служебные) действия. Отсюда следует, что безошибочность выполнения распараллеленного процесса не может быть выше безошибочности исходного, т.е. $P_{расп}^{(i)} \le P^{(i)}$.

4. Принцип разделения структур информационных систем на иерархически упорядоченные множества функциональных структур и частей.

Функциональная часть (ФЧ) – это материальный объект определенного функционального назначения (дискретные элементы и устройства). К категории ФЧ относятся также микрооперации. В любой дискретной системе всегда содержится конечное множество ФЧ. Обозначим его $R_{ФЧ}$.

Функциональная структура (ФС) – это материальное воплощение определенного информационного процесса путем формирования совокупности L множеств $S_l (l = 1,2,...,L)$ ФЧ, объединенных согласно алгоритму данного процесса. Причем, $S_l \le R_{ФЧ}$. Из одного и того же множества ФЧ может быть согласно заданному алгоритму ФС сформирован набор различных множеств $S_1, S_2, ..., S_l$. Связи между этими множествами опять-таки определяются заданным алгоритмом ФС.

Из данного принципа следует:

• Любая одна ФЧ может включаться в несколько множеств S_l, т.е. несколько раз привлекаться для выполнения данного информационного процесса;

• Не все ФЧ информационной системы содержатся в составных множествах S_l данной ФС и, следовательно, не привлекаются для выполнения данного информационного процесса;

• ФС случайного процесса формируется с помощью результирующего множества ФЧ $R_{ФС} = \bigcup_{j=1}^{J} (\prod_{i=1}^{L_j} S_i)$, где $J \le A$. Система знаков $A = const$ вследствие свойства результативности ал-

горитма информационного процесса и адекватности алгоритма расчета исследуемому алгоритму (первый принцип);

• Безошибочность выполнения информационного процесса соответствует безошибочности работы его ФС.

II.2.2. Показатели функциональной надежности информационной системы

II.2.2.1. Единичные показатели

Единичные показатели – это показатели функциональной надежности ИС относительно каждой отдельной функции. К ним относятся:

1. Вероятность безошибочного выполнения процесса

$$P_i = \prod_{j=0}^{m_i-1} P_{ППj},\ i = 1,2,...,n\ ;\ m_i \in M\ , \qquad (2.1)$$

где m_i – количество уровней иерархии при выполнении i-го процесса; $P_{ППj}$ – вероятность безошибочного выполнения процесса j-го уровня иерархии в составе данного процесса; n – количество процессов, выполняемых в ИС в текущий момент времени; M – конечное множество возможных процессов, выполняемых в информационной системе.

2. Вероятность ошибки в выполнении процесса

Это вероятность того, что будет выполнен с ошибкой хотя бы один составной процесс любого уровня иерархии, т.е.

$$G_i = 1 - P_i = 1 - \prod_{j=0}^{m_i-1} P_{ППj}. \qquad (2.2)$$

В свою очередь, показатели безошибочности выполнения каждого составного процесса определяются на основе изложенного ранее в п. II.2.1 принципа *соответствия структуры алгоритма расчета структуре алгоритма функционирования исследуемой информационной системы при выполнении каждой функции,* а также принципа *выделения в алгоритме расчета иерархических уровней, объединенных в иерархическую структуру.*

Пример II.2.1. На железнодорожном транспорте широко применяется определенный класс информационных систем реального времени, которые принято называть системами диспетчерской централизации. В этих системах содержатся разнесенные в пространстве центральный (ЦП) и линейный посты (ЛП). Оператор – диспетчер с ЦП выдает команду на ЛП. Эта команда в виде сообщения передается по каналам связи. Сообщение представляет собой n последовательно передаваемых бит информации.

Предполагается:

– канал биномиальный, т.е. вероятность k ошибок на длине сообщения n определяется как $P(k,n) = C_n^k p^k (1-p)^{n-k}$, где p – вероятность ошибки на бит;

– в целях обеспечения функциональной надежности системы относительно передаваемого сообщения в этом сообщении наряду с информационными битами формируются поверочные (контрольные) биты. Эта процедура осуществляется с помощью CRC-кодирования путем деления блока информационных бит на образующий полином $g(x) = x^{16} + x^{15} + x^2 + 1$;

– длина блока информационных бит $m = 96 + 8N$;

– длина всего сообщения в битах $n = 112 + 8N$, где N – количество байт данных сообщения;

– количество контрольных бит – 16;

– сообщение «упаковывается» в пакет, содержащий адреса отправителя и получателя.

При приеме сообщения должен быть реализован информационный процесс, включающий следующие информационные процессы:

1. Проверка правильности типа пакета;
2. Определение адреса отправителя;
3. Определение адреса получателя;
4. Проверка длины сообщения;
5. Проверка правильности контрольных бит (проверка контрольной суммы).

Вероятности ошибок в выполнении указанных процессов:

Вероятность ошибки в определении типа пакета (трансформации типа пакета)

$G_1 = \frac{K_1 - 1}{2^{16}}(1-(1-p)^{16})$, где K_1 – число типов пакета;

Вероятность ошибки в определении адреса отправителя

$G_2 = \frac{K_2 - 1}{2^{16}}(1-(1-p)^{16})$, где K_2 – число адресов отправителя;

Вероятность ошибки в определении адреса получателя

$G_3 = \frac{K_3 - 1}{2^{16}}(1-(1-p)^{16})$, где K_3 – число адресов получателя;

Вероятность ошибки в определении длины сообщения

$G_4 = \frac{K_4 - 1}{2^{16}}(1-(1-p)^{16})$, где K_4 – число возможных длин сообщений;

Вероятность ошибки в контрольной сумме

$$G_5 = \frac{1}{2^{n-m}} \sum_{i=r+1}^{n} P(i,n) = \frac{1}{2^{16}} \sum_{i=r+1}^{n} C_n^i p^i (1-p)^{n-i},$$

где r – число обнаруживаемых с помощью кода CRC ошибок.

Вероятность ошибки (трансформации) всего сообщения в соответствии с формулой (2.2) определяется следующим приближенным выражением:

$$G_C = 1 - \prod_{j=1}^{5} (1 - G_i) \approx$$

$$\approx 1 - (1 - \frac{1}{2^{16}} \sum_{i=r+1}^{n} C_n^i p^i (1-p)^{n-i}) \prod_{j=1}^{4} \frac{2^{16} + 1 - K_j}{2^{16}}.$$

Данное выражение получено в предположении, что вероятность $1-(1-p)^{16} \approx 1$. Это справедливо для каналов связи с низкой помехоустойчивостью. Для оптоволоконных каналов такое предположение не корректно. Однако оно сделано нами с единственной целью более наглядного представления иллюстративного примера.

Рассмотрим особенности определения данного показателя при многократном выполнении информационного процесса. Здесь уместно оценить вероятность безошибочного выполнения этого процесса при следующих противоположных условиях:

А) *Поток заявок на выполнение данного информационного процесса регулярный* с интервалом времени между заявками T.

Вероятность безошибочного выполнения информационного процесса при данном условии задается следующим выражением:

$$P_i(T, 2T, ..., nT) = P_i^n \text{, где } n = 1, 2, ..., j, ...$$

Вероятность ошибки при регулярном многократном выполнении процесса оценивается как вероятность того, что хотя бы в одном случае из n процесс выполнится с ошибкой, т.е.

$$G_i(T, 2T, ..., nT) = 1 - P_i(T, 2T, ..., nT) = 1 - P_i^n \text{, где } n = 1, 2, ..., j, ...$$

При этих обстоятельствах случайная величина ошибки моделируется геометрическим распределением с математическим ожиданием времени до наступления ошибки

$$T_i(n) = \frac{T(1 - P_i^n)}{1 - P_i}. \qquad (2.3)$$

При большом количестве реализаций процесса ($(n \to \infty)$ формула (2.3) преобразуется к виду

$$T_i(\infty) = T_i = \frac{T}{G_i}$$

Б) *Поток заявок на выполнение данного информационного процесса случайный* и моделируется в виде простейшего потока с интенсивностью Ω_i.

Вероятность безошибочного выполнения информационного процесса в течение времени t задается следующим выражением:

$$P_i(t) = \sum_{n=0}^{\infty} P_i(n,t) P_i^n = \sum_{n=0}^{\infty} \frac{(\Omega_i t)^n}{n!} e^{-\Omega t} P_i^n = \exp(-\Omega_i G_i t)$$

а вероятность ошибки

$$G_i(t) = 1 - P_i(t) = 1 - \exp(-\Omega_i(1 - P_i)t) \approx \Omega_i G_i t$$

В свою очередь, средняя наработка до ошибки в выполнении процесса вычисляется по формуле:

$$T_i = \int_0^\infty \exp(-\Omega_i(1 - P_i)t)dt = \frac{1}{\Omega_i G_i} \qquad (2.4)$$

2. Вероятность частичного функционального отказа системы

Определим критерий трансформации ошибки в выходных данных выполнения задачи (процесса) в функциональный отказ системы относительно этого процесса. Такой отказ относится к категории частичных функциональных отказов, поскольку приводит к невыполнению функции или к ее выполнению, но с неприемлемыми результатами. В соответствии с определением функционального отказа (п. II.1.2) частичный функциональный отказ системы наступает в том случае, когда уровень безошибочности информационного процесса опускается ниже уровня, допустимого для этого процесса.

В первую очередь итогом ошибки является то, что результат R_i выполнения процесса не относится к множеству разрешенных кодовых слов R, т.е. $R_i \notin R$.

Для ряда информационных систем неприемлемость выходного результата в выполнении информационного процесса связана с его погрешностью, когда погрешность δ_i выходных данных выше допустимой $\delta_{ДОП}$.

В некоторых системах приоритет отдается оперативности выдачи результата, т.е. ошибка трансформируется в отказ, если время задержки в выдаче результата τ_i больше допустимого $\tau_{ДОП}$.

Указанные условия трансформации ошибки в частичный функциональный отказ формализуются в виде вероятности сложного события

$$g_{i\Phi T} = P\{(R_i \notin R) \vee (\delta_i > \delta_{ДОП}) \vee (\tau_i > \tau_{ДОП})\}.$$

Обозначим вероятность первого события (несоответствие результата выполнения процесса одному из разрешенных кодовых слов) как g_{i1}, вероятность того, что погрешность результата выше допустимой как g_{i2}, а вероятность того, что время задержки результата больше допустимой как g_{i3}. Тогда условная вероятность трансформации ошибки в частичный функциональный отказ может быть описана следующим выражением:

$$g_{i\Phi T} = g_{i1} \vee g_{i2} \vee g_{i3} = g_{i1} + g_{i2} + g_{i3}.$$

Если в информационной системе, подобной, например, системе диспетчерской централизации, принципиальным является только событие несоответствия полученного результата одному из разрешенных кодовых слов, то в этом случае условная вероятность трансформации ошибки в отказ записывается так $g_{i\Phi T} = g_{i1}$.

Таким образом, вероятность частичного функционального отказа информационной системы равна

$$G_{i\Phi} = (1 - P_i) g_{i\Phi T} = G_i g_{i\Phi T}. \qquad (2.5)$$

Важно понимать, что условная вероятность трансформации ошибки в функциональный отказ может, в свою очередь, быть функцией от этой вероятности ошибки.

При регулярном потоке поступления заявок на выполнение данного информационного процесса с интервалом времени

между заявками T вероятность частичного функционального отказа определяется с помощью следующего выражения

$$G_{i\Phi}(T, 2T, ..., nT) = 1 - (1 - G_i g_{i\Phi T})^n, \text{ где } n = 1, 2, ..., j, ...$$

В другом граничном случае, когда поток заявок случайный и моделируется в виде простейшего потока с интенсивностью Ω_i, вероятность частичного функционального отказа системы имеет следующий вид

$$G_{i\Phi}(t) = 1 - P_{i\Phi}(t) = 1 - \exp(-\Omega_i G_i g_{i\Phi T} t) \approx \Omega_i G_i g_{i\Phi T} t$$

Пример II.2.2. Информационная система диспетчерской централизации [18] содержит АРМ поездного диспетчера, сетевые концентраторы, шлюзовые машины, каналы связи (обычно волоконно-оптические линии связи ВОЛС), модули МАСТЕР, внутренние сети CAN, модуль вывода команд управления устройствами железнодорожной автоматики и телемеханики (ЖАТ), а также модули ввода. Укрупненная структурная схема системы показана на рис. II.2.2. По команде диспетчера из модуля МАСТЕР выдается управляющая информация, которая подается на модуль вывода и затем на устройства ЖАТ. Если в результате ошибки в процессе формирования управляющей информации из постоянного запоминающего устройства модуля МАСТЕР на модуль вывода подается кодовое слово, которое не относится к множеству разрешенных, то данная ошибка трансформируется в функциональный отказ системы. Известны следующие исходные данные: $G_{ПР}$ – вероятность ошибки в выполнении процесса формирования управляющей информации; V – объем памяти в модуле МАСТЕР; n – количество бит в ячейке памяти; ω – тактовая частота процессора модуля МАСТЕР; z – число тактов работы процессора по считыванию информации из памяти.

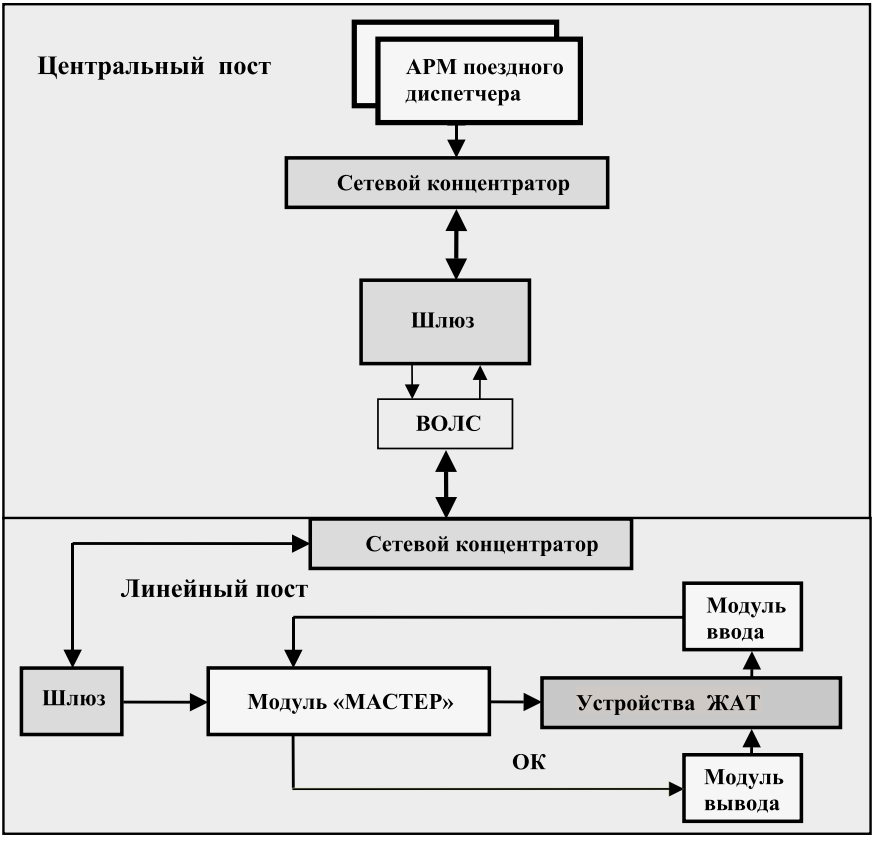


Рис. II.2.2. Упрощенная структурная схема системы диспетчерской централизации

Требуется установить условную вероятность того, что считываемое из модуля МАСТЕР кодовое слово не относится к числу разрешенных слов.

Решение

Пусть в памяти сформировалась ложная информация в k ячейках. Вероятность формирования ложной информации в одной ячейке памяти равна $g_{ОШ} = \frac{G_{ПР}}{V}$,

Тогда вероятность того, что из k ячеек будет сосчитано ровно r ложных слов, равна $p(k,r) = C_k^r g_{ОШ}^r (1 - g_{ОШ})^{k-r}$.

Математическое ожидание количества использованных при считывании информации ячеек памяти с ложной информацией равно $\overline{m} = \sum_{r=0}^{\infty} rp(k,r) = r \cdot g_{ОШ}$.

Поскольку между числом тактов считывания информации и количеством сосчитанных слов существует прямая зависимость $z = \frac{r \cdot n}{\omega}$, то математическое ожидание количества использованных при считывании за один такт информации ячеек памяти с ложной информацией равно $\overline{m}_1 = \frac{r}{z} g_{ОШ} = \frac{\omega}{n} g_{ОШ}$

Параметр $\overline{m}_1$ есть интенсивность считывания ложной информации в единицу времени (в один машинный такт), т.е. $\overline{m}_1 = \lambda_{СЧ}$. Поскольку по определению постоянная интенсивность отказа (ошибки) совпадает с вероятностью отказа (ошибки) за единицу времени или за одно действие. Это означает справедливость равенства $\lambda = G(1)$, что применительно к данной задаче означает совпадение вероятности считывания ложной информации в единицу времени (в один такт) с интенсивностью считывания этой информации $\lambda_{СЧ} = g_{СЧ}$. Отсюда следует, что условная вероятность трансформации ошибки в выполнении процесса формирования управляющей информации в функциональный отказ при условии, что ошибка привела к формированию ложной информации в памяти, может быть вычислена по формуле $g_{ФТ} = g_{СЧ} = \frac{\omega}{n} g_{ОШ}$.

Приведем численные значения.

Частота работы центрального процессора модуля МАСТЕР $\omega = 2 \cdot 10^6$ опер/с. Длина кодового слова на входе памяти модуля МАСТЕР $n = 16$. Объем памяти модуля МАСТЕР $V = 2^{20}$ байт.

Вероятность ошибки в выполнении процесса формирования управляющей информации $G_{ПР} = 10^{-3}$.

Тогда $g_{ОШ} = \frac{G_{ПР}}{V} = \frac{10^{-3}}{10^{6}} = 10^{-9}$, а условная вероятность трансформации ошибки в функциональный отказ $g_{ФТ} = \frac{\omega}{n} g_{ОШ} = \frac{2*10^{6}}{16} 10^{-9} = 1.2*10^{-4}$.

3. Средняя наработка до частичного функционального отказа системы

При периодичности поступления заявок T_i в регулярном потоке в соответствии с формулой (2.3) средняя наработка системы до *i-го* частичного функционального отказа имеет следующий вид:

$$T_{i\Phi}(n) = T_i \frac{1-(1-G_i g_{i\Phi T})^n}{G_i g_{i\Phi T}}, \text{ или при } n \to \infty \; T_{i\Phi} = \frac{T_i}{G_i g_{i\Phi T}}. \quad (2.6)$$

При простейшем потоке заявок с интенсивностью Ω_i

$$T_{i\Phi} = \int_0^{\infty} \exp(-\Omega_i G_i g_{i\Phi T} t) dt = \frac{1}{\Omega_i G_i g_{i\Phi T}}. \quad (2.7)$$

Поскольку интенсивность поступления заявок Ω_i в простейшем потоке величина постоянная (Ω_i=const),то среднее время между заявками также постоянно и равно $\overline{T}_i = \frac{1}{\Omega_i}$. Следовательно, формула (2.7) может быть представлена в виде

$$T_{i\Phi} = \frac{1}{\Omega_i G_i g_{i\Phi T}} = \frac{\overline{T}_i}{G_i g_{i\Phi T}}.$$

Для двух граничных условий получен практически одинаковый результат определения средней наработки до частичного функционального отказа, за тем исключением, что в одном случае имеет место детерминированное время между заявками, во втором – среднее значение случайного времени. Это дает основание воспользоваться данным результатом и в промежуточных случаях поступления потоков заявок с ограниченным последействием.

Следует обратить внимание на то, что между средней наработкой до частичного функционального отказа и средней наработкой до ошибки в выполнении информационного процесса установлена прямая зависимость $\frac{T_{i\Phi}}{T_{i\Phi}} = \frac{1}{g_i}$, что позволяет на практике упростить вычисления одного показателя по известному другому.

4. Среднее время простоя системы при обслуживании одного информационного процесса

Источниками простоя информационной системы относительно отдельного информационного процесса являются следующие причины:

- Задержки заявки в очереди на выполнение процесса;
- Задержки в обнаружении ошибки (включая функциональные отказы);
- Временные затраты на устранение ошибки и восстановление информационного процесса;
- Временные затраты на контроль безошибочности выполнения процесса после его восстановления.

Таким образом, среднее время простоя в техническом обслуживании *i-го* информационного процесса рассчитывается путем суммирования четырех составляющих:

$$T_{ПР} = \sum_{j=1}^{4} T_{ij} \ , \qquad (2.8)$$

где T_{i1} – среднее время задержки заявки в очереди,

T_{i2} – среднее время обнаружения ошибки,

T_{i3} – среднее время устранения ошибки и восстановления информационного процесса,

T_{i4} – среднее время контроля безошибочности выполнения процесса после его восстановления.

Численные значения приведенных составляющих среднего времени простоя в обслуживании информационного процесса зависят от архитектуры информационной системы, от организации обслуживания заявок, от эффективности системы контроля и диагностирования. Они существенно зависят от принятой в системе стратегии устранения ошибки и восстановления информационного процесса, например, путем рестарта, или повторного счета, или путем математического стробирования и исключения возможных ошибочных результатов и др.

5. Коэффициент частичной функциональной готовности системы

Под частичной функциональной готовностью информационной системы понимается ее способность в произвольный момент времени безошибочно выполнять *определенный* информационный процесс. Если в пределе система выполняет только один информационный процесс, то коэффициент частичной функциональной готовности будет равен коэффициенту полной функциональной готовности или для упрощения терминологии – коэффициенту функциональной готовности.

Формульное выражение данного показателя имеет следующий вид:

$$K_{i\,ФГ} = \frac{T_i}{T_i + T_{i\,ПР}}. \qquad (2.9)$$

Следует обратить внимание на то обстоятельство, что в данном показателе содержатся параметры, которые представляют собой только меру времени между ошибками в выполнении определенного процесса и меру времени устранения ошибки. В этом показателе отсутствуют меры, относящиеся к функциональным отказам, как это могло бы быть, если бы мы руководствовались определением коэффициента готовности технических средств. Различие принципиальное. Суть его в том, что функциональный отказ есть последствие ошибки. Большинство ошибок не приводят к функциональным отказам. Вместе с тем, каждая ошибка вызывает простои времени в выполнении данного процесса. Этим обстоятельством нельзя пренебречь. Длительности простоев аналогичны как при устранении ошибки, так и при устранении функционального отказа. Различие было бы только в тех очень редких случаях, когда функциональный отказ вызывает недопустимый ущерб. В этих случаях потребовалось бы применение специальной технологии возвращения системы в исходное состояние. Такие функциональные отказы называют опасными, иногда критическими. Исследование их не относятся к предмету функциональной надежности.

II.2.2.2. Комплексные показатели

Комплексные показатели предназначены для прогнозирования и оценки функциональной готовности системы при выполнении всех предусмотренных функций путем реализации соответствующих информационных процессов. Поскольку речь идет о реализации информационных процессов, то комплексные показатели являются мерой безошибочности выполнения всех этих процессов.

6. Средняя вероятность безошибочного выполнения информационных процессов

Каждый информационный процесс различается частотой исполнения, которая зависит от интенсивности поступления заявок на его выполнение Ω_i, весом (значимостью, вкладом в эффективность работы системы) этого процесса, и, конечно, его функциональной надежностью. Обозначим вес процесса как ω_i. Коэффициенты нормирования частоты исполнения процесса и его значимости при условии выполнения в системе *M* процессов определяются как

$$k_{i1} = \frac{\Omega_i}{\sum_{i=1}^{M} \Omega_i} \text{ и } k_{i2} = \frac{\omega_i}{\sum_{i=1}^{M} \omega_i}.$$

Следовательно, средняя вероятность безошибочного выполнения процесса есть средневзвешенная вероятность, определяемая с помощью произведения коэффициентов k_{i1} и k_{i2}, т.е.

$$\overline{P} = \sum_{i=1}^{M} k_{i1} k_{i2} P_I = \sum_{I=1}^{M} k_i P_i \ . \qquad (2.10)$$

7. Коэффициент функциональной готовности системы

При условии независимости информационных процессов коэффициент функциональной готовности системы определяется следующим образом:

$$K_{\Phi\Gamma} = \sum_{j=0}^{M} \prod_{i=1}^{j} K_{i\Phi\Gamma} \prod_{i=1}^{M-j} (1 - K_{i\Phi\Gamma})(1 - \sum_{i=1}^{M-j} \omega_i), \qquad (2.11)$$

где $\prod_{i=1}^{0} (1 - K_{i\Phi\Gamma}) = 1; \prod_{i=1}^{0} K_{i\Phi\Gamma} = 1; \sum_{i=1}^{0} \omega_i = 0, \sum_{i=1}^{M} \omega_i = 1.$

8. Среднее время простоя системы при обслуживании информационных процессов

При вычислении среднего времени простоя системы относительно составных информационных процессов исходными являются следующие основные сведения:

– средние времена простоев $T_{ПР}$ системы относительно каждого выполняемого процесса ($i = 1,2,...,M$);

– вероятности ошибки в выполнении каждого процесса: чем выше эта вероятность G_i, тем больший вклад среднего времени $T_{ПР}$ в результирующее значение времени простоя системы и наоборот;

– частота выполнения каждого процесса: чем чаще процесс выполняется, тем больший вклад его параметров в значение результирующего показателя.

$$T_{ПР} = \frac{\sum_{i=1}^{M} G_i k_{i1} T_{iПР}}{\sum_{i=1}^{M} G_i k_{i1}}, \qquad (2.12)$$

где $k_{i1} = \frac{\Omega_i}{\sum_{i=1}^{M} \Omega_i}$ – нормирующий коэффициент частоты выполнения каждого информационного процесса.

9. Средняя наработка на ошибку системы при выполнении информационных процессов

Известны следующие показатели: коэффициент функциональной готовности $K_{ФГ}$ и среднее время простоя системы $T_{ПР}$. Следовательно, формульное выражение неизвестного показателя среднего времени до ошибки системы T_C можно найти с помощью формулы коэффициента готовности

$$K_{ФГ} = \frac{T_C}{T_C + T_{ПР}}. \text{ Отсюда } T_C = \frac{T_{ПР} K_{ФГ}}{1 - K_{ФГ}}. \qquad (2.13)$$

10. Средняя наработка на полный функциональный отказ системы

По аналогии с аргументированным ранее переходом от расчетов ошибки в выполнении процесса к расчетам частичного функционального отказа (см. (2.5)) в данном случае также применим переход от ошибки системы в целом к ее полному функциональному отказу путем учета вероятности трансформации этой ошибки в функциональный отказ. С этой целью находим среднюю вероятность трансформации ошибки в функциональный отказ по всем выполняемым в системе информационным процессам

$$\overline{g}_{ФТ} = \frac{\sum_{i=1}^{M} g_{i\,ФТ}}{M}.$$

Ранее при анализе временных показателей системы относительно частичных функциональных отказов было установлено соотношение $\frac{T_{i\Phi}}{T_{i\Phi}} = \frac{1}{g_i}$. С помощью формулы (2.13) и указанного соотношения находим формульное выражение средней наработки между полными функциональными отказами системы

$$T_{\Phi} = \frac{T_{ПР} K_{ФГ}}{(1 - K_{ФГ})\overline{g}_{ФТ}} \qquad (2.14)$$

11. Коэффициент оперативной функциональной готовности системы

Данный показатель определяет вероятность того, что в произвольный достаточно удаленный от начала отсчета момент времени информационная система работает с приемлемым уровнем безошибочности и обеспечивает безотказное выполнение всех предусмотренных информационных процессов в течение заданного оперативного времени $\tau_{ОП}$ управления.

В соответствии с формулой (2.22) [1] находим

$$K_{ОФГ}(\tau_{ОП}) = K_{ФГ} P(\tau_{ОП}) = K_{ФГ} \exp(-\tau_{ОП} / T_{Ф}). \qquad (2.15)$$

Формула (2.15) справедлива при условии простейшего потока функциональных отказов в системе. Такая предпосылка корректна, поскольку согласно доказательствам Б.И. Григелиониса и И.Б. Погожева [16] случайное многократное разрежение произвольного потока событий приводит к преобразованию этого потока к простейшему потоку. Действительно, функциональные отказы системы есть результат трансформации отдельных ошибок из суммарного потока ошибок. Процесс трансформации, характеризуемый выражением $T = T_{Ф} g_{ФТ}$, есть полный аналог многократного случайного разрежения потока ошибок при реальном условии, что $g_{ФТ} << 1$.

В случае короткого оперативного времени управления гарантированно выполняется соотношение $\tau_{ОП} / T_{Ф} << 1$ и формула (2.15) с погрешностью, не более первого порядка малости, упрощается к виду

$$K_{ОФГ}(\tau_{ОП}) \approx K_{ФГ}(1 - \tau_{ОП} / T_{Ф}),$$

или с учетом формулы (2.15) к следующему виду:

$$K_{ОФГ}(\tau_{ОП}) \approx K_{ФГ} - \frac{\tau_{ОП}}{T_{ПР}} \overline{g}_{ФТ}(1 - K_{ФГ}) \qquad (2.16)$$

II.2.3. Метод расчета показателей функциональной надежности с помощью фундаментальной матрицы поглощающих Марковских цепей

Введение. Марковские цепи с поглощающими состояниями не имеют стационарных вероятностей (т.е. все P_i=0). Вместе с тем, с помощью поглощающих Марковских моделей можно нередко получить значения различных практических величин, которые могут быть полезны при расчетах показателей функциональной надежности информационных систем. К их числу относятся:

n_i – среднее число шагов до поглощения при начальном состоянии s_i (умножив n_i на T, найдем среднее время до останова системы);

b_{ik} – вероятность того, что при начальном состоянии s_i процесс будет поглощен в состоянии s_k (вероятность остановки системы).

С помощью этих величин можно определить вероятность и среднюю наработку до функционального отказа системы.

Фундаментальная матрица поглощающей Марковской цепи. При исследовании поглощающих цепей полезно привести матрицу переходных вероятностей к каноническому виду [17]. Для этого нужно выписать поглощающие состояния первыми. Тогда матрица переходных вероятностей принимает следующий вид:

$$\Pi = \begin{array}{c} \\ \\ \text{Не поглощение} \end{array} \begin{array}{c} \begin{array}{cc} \text{Поглощение} & \text{Поглощающие} \end{array} \quad \text{Не поглощающие} \\ \left(\begin{array}{c|c} \boldsymbol{I} & \boldsymbol{0} \\ \hline \boldsymbol{R} & \boldsymbol{Q} \end{array} \right) \end{array}$$

Здесь I – единичная матрица и 0 – нулевая матрица. Это соответствует тому состоянию, что, попав в поглощающее состояние, цепь в нем и остается, и из поглощающих состояний невозможен переход в не поглощающее состояние; R и Q – неотрицательные матрицы, представляющие вероятности переходов из не поглощающих состояний.

Поскольку суммы элементов Π по строкам равны 1, а R не может быть тождественно равно 0, то суммы элементов матрицы Q по строкам меньше или равны 1 (≤ 1), причем по крайней мере в одной строке сумма элементов меньше 1.

Основной результат для конечных поглощающих Марковских цепей состоит в том, процесс смены состояний обязательно попадет в поглощающее состояние, т.е. $Q^n \to 0$. Справедливо даже более сильное утверждение, что степенной ряд $N = I + Q + Q^2 + \ldots$ всегда сходится к пределу, равному обратной матрице для $I - Q$, т.е.

$$N = (I - Q)^{-1}$$

Матрица N называется *фундаментальной матрицей поглощающей Марковской цепи.* С помощью этой матрицы находят вектор – столбец среднего количества шагов до поглощения при каждом возможном начальном состоянии

$$\rho = (n_i) = N1,$$

где 1 – единичный вектор – столбец;
а также вектор – столбец

$$B = (b_{ik}) = NR$$

вероятностей попадания Марковской цепи в поглощающие состояния (в состояния функционального отказа системы).

Пример II.2.3. Пусть информационный процесс в системе организован с рестартом, т.е. при обнаружении ошибки производится повторение процесса для устранения ошибки. Процесс повторяется не более трех раз подряд. Если каждая реализация приводит к ошибочному результату, то выполнение информационного процесса останавливается, фиксируется функциональный отказ и осуществляется диагностика состояний информационной системы. Предполагается, что эффективность контроля безошибочности вычислений равна единице. Вероятность отсутствия ошибок при каждом однократном выполнении вычислительного процесса принимается равной p. Требуется определить среднее время до функционального отказа при условии, что средняя длительность выполнения вычислительного процесса равна τ.

Решение

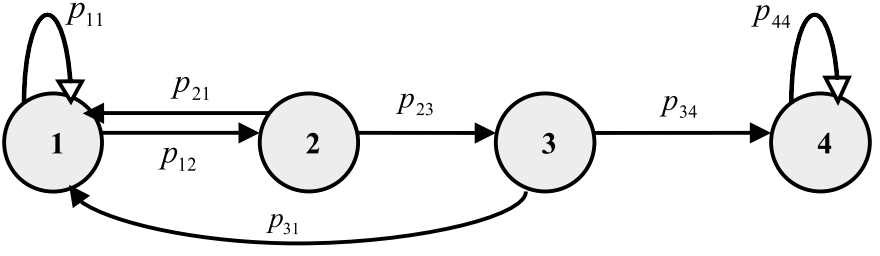


Рис II.2.3. Модель надежности системы с поглощающим состоянием

Граф состояний модели надежности системы показан на рис. II.2.3. В данном примере состояние 4 – поглощающее. Матрица переходных вероятностей Марковской цепи $\Pi = (p_{ij})$ при условии $p_{12} = p_{23} = p_{34} = 1 - p = q$ и $p_{11} = p_{21} = p_{31} = p$; $p_{44} = 1$ имеет следующий вид:

$$\Pi = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} p & q & 0 & 0 \\ p & 0 & q & 0 \\ p & 0 & 0 & q \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Представим матрицу Π в каноническом виде

$$\Pi = \begin{array}{c|ccc|c} \boldsymbol{I} \quad 4 & 1 & 2 & 3 & \\ \hline 4 & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline 1 & 0 & p & q & 0 \\ 2 & 0 & p & 0 & q \\ 3 & q & p & 0 & 0 \\ \hline \boldsymbol{R} & & & & \boldsymbol{Q} \end{array}$$

Фундаментальная матрица

$$N = (I - Q)^{-1} = \begin{pmatrix} 1-p & -q & 0 \\ -p & 1 & -q \\ -p & 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & q & q^2 \\ p(1+q) & q & q^2 \\ p & pq & q^2 \end{pmatrix} \cdot \frac{1}{q^3}$$

Отсюда

$$\rho = N1 = \begin{pmatrix} 1 & q & q^2 \\ p(1+q) & q & q^2 \\ p & pq & q^2 \end{pmatrix} \cdot \frac{1}{q^3} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1+q+q^2}{q^3} \\ \frac{1+q}{q^3} \\ \frac{1}{q^3} \end{pmatrix} \begin{matrix} 1 \\ 2 \\ 3 \end{matrix}$$

Таким образом, среднее время до функционального отказа при условии, что исследуемая система находилась в начальном состоянии 1, определяется выражением

$$T_{\Phi} = \tau\rho_1 = \tau\frac{1+q+q^2}{q^3} = \frac{\tau}{Gg_{\Phi T}}, \text{ где } G = q; g_{\Phi T} = \frac{q^2}{1+q+q^2}$$

Например, вероятность ошибки при однократном выполнении информационного процесса равна $G = 10^{-4}$; средняя длительность однократного выполнения процесса $\tau = 1$ мс. Тогда среднее время до функционального отказа системы относительно выполняемого информационного процесса равно $T_{\Phi} = 2.8 \cdot 10^5$ *ч*

II.2.4. Модифицированный топологический полумарковский метод для расчета функциональной надежности систем

Введение. Функциональная надежность информационной системы определяется безошибочностью выполнения предусмотренных функций. Каждая функция реализуется с помощью информационного процесса. Процесс выполняется по заявкам. Таким образом, информационная система есть система массо-

вого обслуживания. Функциональная надежность системы – это функциональная надежность выполнения совокупности предусмотренных информационных процессов. Заявки на обслуживание имеют разный приоритет. Менее приоритетные заявки обслуживаются реже приоритетных. Вероятность ошибки в выполнении менее приоритетного процесса уменьшается в соответствии с уменьшением частоты обслуживания заявок на данный процесс. Действительно, чем реже мы пользуемся компьютером, тем меньше ошибок получаем в результатах и наоборот.

Этим фактом не ограничиваются особенности расчета функциональной надежности системы при выполнении заданной совокупности информационных процессов. Есть еще одно существенное обстоятельство. Оно заключается в скорости обслуживания заявок (скорости выполнения процессов). Чем выше скорость обслуживания (т.е. меньше время выполнения процесса), тем меньше загруженность каналов обработки информации в системе и тем чаще могут выполняться процессы, а, следовательно, выше вероятность ошибки в их выполнении.

Приведенные рассуждения позволяют перечислить минимально необходимые исходные данные для расчетов показателей функциональной надежности информационной системы в условиях обслуживания разнородных потоков заявок, в том числе в условиях их приоритетного обслуживания [18].

Постановка задачи

Исходные данные:

- функции распределения времени между заявками $F_k(t)$, $k = 1, 2, ..., M$, функции распределения времени выполнения информационных процессов (обслуживания заявок) $Q_k(t)$, $k = 1, 2, ..., M$;

- вероятности ошибок и частичных функциональных отказов $G_i, g_{i\Phi T}, k = 1,2,...,M$;

- «веса» информационных процессов $\omega_k, k = 1,2,...,M$

- средние длительности простоев системы при устранении ошибок в процессах $T_{ПР}$.

- минимальное количество $R \in M$ информационных процессов, которые должны быть выполнены безошибочно в соответствии с критерием полного функционального отказа системы.

Требуется:

Установить правила определения показателей функциональной надежности информационной системы (2.1) – (2.16) в условиях приоритетного выполнения информационных процессов.

Решение задачи

Для определения показателей функциональной надежности системы в условиях приоритетного обслуживания потоков заявок используем некоторые положения изложенного в п. I.4.4 [1] полумарковского топологического метода применительно к приведенным выше исходным условиям. Учет этих условий требует определенной модификации исходного метода. Это связано, главным образом, с необходимостью вычисления средних наработок на частичные функциональные отказы относительно обслуживания каждого информационного потока в условиях прерывания обслуживания или дообслуживания предыдущей заявки, тогда как топологический полумарковский метод предназначен для вычислений показателей структурной надежности.

Этапы решения задачи:

1. Строится граф функционирования системы с приоритетным обслуживанием заявок. В частности, на рис. II.2.4 показана графовая модель системы с абсолютными приоритетами в обслуживании заявок. Для упрощения изложения сути задачи предполагается, что в информационной системе имеется один канал обслуживания. Вершины графа характеризуют следующие состояния системы: 0 – все заявки отсутствуют; *k*-обслуживается заявка *k-го* приоритета (в случае занятости канала обслуживания и поступления заявки более высокого (*k-1)-го*

приоритета прерывается выполнение заявки и система переходит в состояние *k-1,* в котором обслуживается заявка более высокого приоритета); состояние *1* – обслуживается заявка самого высокого приоритета.

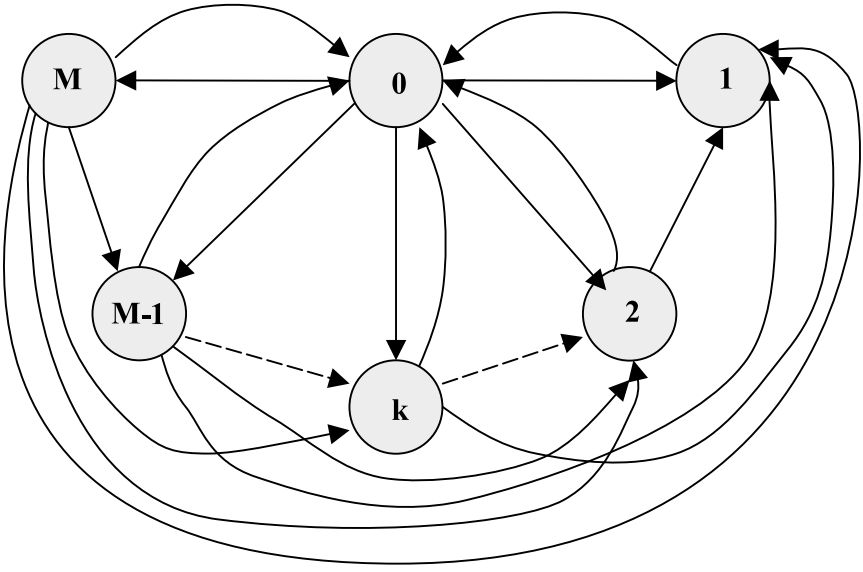


Рис. II.2.4. Графовая модель одноканальной системы с абсолютными приоритетами в выполнении информационных процессов при условии отсутствия очереди

2. Каждой вершине графа системы приписываются те случайные величины, которые определяют возможность выхода системы из данной вершины в соседнее состояние. Так, вершине 0 приписываются случайные величины времени между заявками $\vartheta_1,...,\vartheta_k,...,\vartheta_M$ с функциями распределения времени между заявками $F_1(t)=F_{01}(t),...,F_k(t)=F_{0k}(t),...,F_M(t)=F_{0M}(t)$. Вершине 1 приписывается случайная величина времени обслуживания заявки высшего приоритета η_1 с функцией распределения $Q_1(t)=Q_{10}(t)$. Вершине k – случайные величины $\eta_k,\vartheta_{k-1},\vartheta_{k-2},...,\vartheta_1$ с функциями распределения

$Q_k(t) = Q_{k0}(t); F_{k,k-1}(t); F_{k,1}(t)$. Наконец, вершине M приписываются случайные величины $\eta_M, \vartheta_{M-1}, ..., \vartheta_k, ..., \vartheta_1$ с функциями распределения $Q_M(t) = Q_{M0}(t); F_{M,M-1}(t); F_{M,1}(t)$.

3. Определяются функции плотности $f_k(t)$ на основании формулы (4.9) [1] при условии независимости случайных величин $\eta_1, ..., \eta_k, ..., \eta_M; \vartheta_1, ..., \vartheta_k, ..., \vartheta_M$

$$f_{ik}(t) = \lim_{\Delta t \to 0} \frac{P\{t < \vartheta_k \le t + \Delta t; \vartheta_1, ..., \vartheta_M, \eta_1, ..., \eta_M > t\}}{\Delta t} =$$

$$= \frac{dF_{ik}(t)}{dt} \prod_{j=0}^{M} F_{ij}(t) Q_{ij}(t); \quad i \ne j; i \ne k$$

Следовательно,

$$f_{0k}(t) = \frac{dF_{0k}(t)}{dt} \prod_{i=1}^{M} F_{0i}(t); i \ne k; k = 1...M \ .$$

$$f_{k0}(t) = \frac{dQ_{k0}(t)}{dt} \prod_{i=1}^{k-1} F_{ki}(t); k = 1...M; \prod_{i=1}^{0} F_{ki}(t) = 1 \ .$$

$$f_{kr}(t) = \frac{dF_{kr}(t)}{dt} Q_{k0}(t) \prod_{i=1}^{k-1} F_{ki}(t); k = 1...M; r = 1...k-1; i \ne r \ .$$

4. По формулам (4.10) [1] находятся переходные вероятности p_{ij} и математические ожидания безусловных T_i времен пребывания полумарковского случайного процесса в каждом из состояний

$$p_{0k} = \int_0^\infty f_{ok}(t)dt;\ p_{k0} = \int_0^\infty f_{k0}(t)dt;\ p_{kr} = \int_0^\infty f_{kr}(t)dt;$$

$$T_0 = \int_0^\infty t\sum_{i=1}^{M} f_{0k}(t)dt;\ T_k = \int_0^\infty t[f_{k0}(t) + \sum_{r=0}^{k-1} f_{kr}(t)]dt$$

5. При помощи полученной матрицы $\Pi = (p_{ij})$ и вектора $\vec{T} = (T_k); k = 0...M$ по формуле (4.18) [1] топологического полумарковского метода определяются финальные вероятности π_k пребывания полумарковского процесса в каждом из состояний графа рис. II.2.4.

6. Согласно теореме В.Смита [19] определяется среднее время между повторными попаданиями полумарковского процесса в состояние $k = 0...M$

$$T_{kk} = \tau_k = \frac{T_k}{\pi_k},$$

где τ_k – среднее время между выполнениями *k*-го информационного процесса.

7. По приведенным выше исходным данным на основании формул (2.7), (2.9)-(2.16) определяются все требуемые показатели функциональной надежности информационной системы с приоритетным обслуживанием заявок. Например, средняя наработка на частичный функциональный отказ системы относительно *k*-го информационного процесса равна

$$T_{k\Phi} = \frac{\tau_k}{G_k g_{k\Phi T}} = \frac{T_k}{\pi_k G_k g_{k\Phi T}}.$$

Пример II.2.4. Пусть на вход ИС поступает два потока заявок (M=2). Система одноканальная, очередь на обслуживание отсутствует. Первый поток заявок приоритетный. Функции распределения времени между заявками $F_k(t) = 1 - \exp(-\lambda_k t); k = 1,2$. Функции распределения времени обслуживания заявок $Q_k(t) = 1 - (1 + \mu_k t)\exp(-\mu_k t); k = 1,2$, где λ_k, μ_k – интенсивности поступления или соответственно обслуживания заявок.

Известны значения $G_k, g_{ki\Phi i}; \omega_k; T_{k\Pi P}; k = 1,2.$

Численные значения исходных данных:

$$\lambda_1 = 100\ 1/c; \lambda_2 = 500\ 1/c; \mu_1 = \mu_2 = 10^3\ 1/c; G_1 = 10^{-6}; G_2 = 10^{-7};$$

$$g_{1\Phi T} = g_{2\Phi T} = 10^{-3}; \omega_1 = 0,7; \omega_2 = 0,3; T_{1\Pi P} = T_{2\Pi P} = 10^{-1}\,ч.; \tau_{O\Pi} = 8\,ч$$

Требуется определить численные значения показателей функциональной надежности информационной системы.

Решение задачи осуществляется в соответствии с указанными выше этапами.

Этап. 1. Строится графовая модель одноканальной системы с приоритетным обслуживанием первого из двух потоков заявок

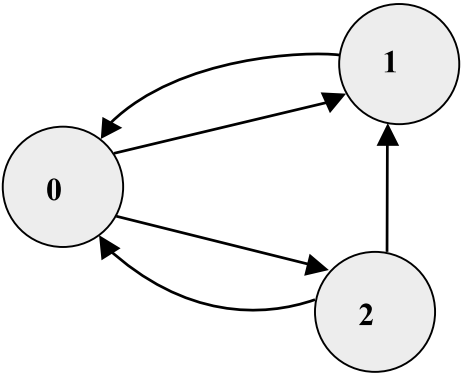


Рис. II.2.5. Графовая модель одноканальной системы без очереди с приоритетным обслуживанием первого из двух информационных процессов

(рис. II.2.5). Состояние 0 – заявки отсутствуют; состояние 1 – выполняется информационный процесс по обслуживанию первого потока заявок; состояние 2 – выполняется информационный процесс по обслуживанию второго потока заявок, в случае поступления заявки первого потока обслуживание прекращается и система переходит к обслуживанию первого потока заявок.

Этап. 2. Каждой вершине графа приписываются случайные величины времени между заявками и времени обслуживания и определяются условные функции распределения времени перехода из вершины (состояния) в другие состояния:

$$F_0(t) = F_{01}(t)F_{02}(t) = [1 - \exp(-\lambda_1 t)][1 - \exp(-\lambda_2 t];$$

$$F_1(t) = Q_{10}(t) = 1 - (1 + \mu_1 t)\exp(-\mu_1 t);$$

$$F_2(t) = Q_{20}(t)F_{21}(t) = [1 - (1 + \mu_2 t)\exp(-\mu_2 t)][1 - \exp(-\lambda_1 t)],$$

где $F_{01}(t) = F_{21}(t)$.

Этап. 3. Определяются функции плотности $f_k(t)$:

$$f_{01}(t) = \frac{dF_{01}(t)}{dt} F_{02}(t) = \lambda_1 \exp(-\lambda_1 - \lambda_2)t;$$

$$f_{02}(t) = \frac{dF_{02}(t)}{dt} F_{01}(t) = \lambda_2 \exp(-\lambda_1 - \lambda_2)t;$$

$$f_{10}(t) = \frac{dQ_{10}(t)}{dt} = \mu_1^2 t \exp(-\mu_1)t;$$

$$f_{20}(t) = \frac{dQ_{20}(t)}{dt} F_{21}(t) = \mu_2^2 t \exp(-\lambda_1 - \mu_2)t;$$

$$f_{21}(t) = \frac{dF_{21}(t)}{dt} Q_{20}(t) = \lambda_1(1 + \mu_2 t)\exp(-\lambda_1 - \mu_2)t.$$

Этап. 4. Определяются переходные вероятности p_{ij} и математические ожидания T_i:

$$p_{01} = \frac{\lambda_1}{\lambda_1 + \lambda_2}; p_{02} = \frac{\lambda_2}{\lambda_1 + \lambda_2}; p_{10} = 1;$$

$$p_{20} = \frac{\mu_2^2}{(\lambda_1 + \mu_2)^2}; p_{21} = \frac{\lambda_1(\lambda_1 + 2\mu_2)}{(\lambda_1 + \mu_2)^2};$$

$$T_0 = \frac{1}{\lambda_1 + \lambda_2}; T_1 = \frac{2}{\mu_1}; T_2 = \frac{(\lambda_1 + 2\mu_2)}{(\lambda_1 + \mu_2)^2}.$$

Этап. 5. Находятся финальные (стационарные) вероятности пребывания системы в каждом из состояний:

$$\pi_0 = \frac{T_0}{T_0 + (p_{01} + p_{02}p_{21})T_1 + p_{02}T_2} = \frac{T_0}{A};$$

$$\pi_1 = \frac{(p_{01} + p_{02}p_{21})T_1}{A}; \pi_2 = \frac{p_{02}T_2}{A}$$

Этап. 6. Определяются средние времена между соседними реализациями первого и второго информационных процессов

$$\tau_1 = \frac{T_1}{\pi_1} = \frac{A}{p_{01} + p_{02}p_{21}};\ \tau_2 = \frac{T_2}{\pi_2} = \frac{A}{p_{02}}.$$

Этап. 7. Определяются численные значения показателей функциональной надежности системы:

- средние наработки между частичными функциональными отказами

$$T_{1Ф} = \frac{\overline{T_1}}{G_1 g_{1ФТ}} = \frac{\tau_1}{G_1 g_{1ФТ}} = \frac{A}{(p_{01} + p_{02}p_{21})G_1 g_{1ФТ}};\ T_{2Ф} = \frac{A}{p_{02}G_2 g_{2ФТ}}.$$

Для вычисления указанных показателей требуется предварительно определить переходные вероятности и математические ожидания согласно этапу 4

$$p_{01} = \frac{\lambda_1}{\lambda_1 + \lambda_2} = \frac{1}{6}; p_{02} = \frac{5}{6}; p_{10} = 1;$$

$$p_{20} = \frac{\mu_2^2}{(\lambda_1 + \mu_2)^2} = 0.91; p_{21} = \frac{\lambda_1(\lambda_1 + 2\mu_2)}{(\lambda_1 + \mu_2)^2} = 0.19$$

$$T_0 = \frac{1}{\lambda_1 + \lambda_2} = 1.7 \cdot 10^{-3} c;\ T_1 = \frac{2}{\mu_1} = 0,2 \cdot 10^{-3} c;$$

$$T_2 = \frac{(\lambda_1 + 2\mu_2)}{(\lambda_1 + \mu_2)^2} = 1.7 \cdot 10^{-3} c.$$

Затем вычисляется параметр $A = T_0 + (p_{01} + p_{02}p_{21})T_1 + p_{02}T_2 = 1.87 \cdot 10^{-3}$ и средние времена между реализациями первого и второго информационных процессов

$$\tau_1 = \frac{A}{p_{01} + p_{02}p_{21}} = \frac{1.87 \cdot 10^{-3}}{0.32} = 5.84 \cdot 10^{-3} c;$$

$$\tau_2 = \frac{A}{p_{02}} = \frac{1.87 \cdot 10^{-3}}{0.83} = 2.25 \cdot 10^{-3} c.$$

Таким образом,

$$T_{1Ф} = \frac{A}{(p_{01} + p_{02}p_{21})G_1 g_{1ФТ}} = \frac{1.87 \cdot 10^{-3}}{0.32 \cdot 10^{-9}} = 5.84 \cdot 10^{6} c = 1623 \text{ ч}.$$

При этом средняя наработка на ошибку равна $T_{1С} = T_{1ФТ} g_{1ФТ} = 1.62$ *ч*;

$$T_{2Ф} = \frac{A}{p_{02}G_2 g_{2ФТ}} = \frac{1{,}87 * 10^{-3}}{0.83 * 10^{-10}} = 2.25 * 10^{7} c = 6258 \text{ ч}.$$

При этом средняя наработка на ошибку $T_{2С} = T_{2ФТ} g_{2ФТ} = 6.26$ *ч*

- вычисляются коэффициенты частичной функциональной готовности

$$K_{1ФГ} = \frac{T_1}{T_1 + T_{1ПР}} = \frac{1.62}{1.62 + 0.1} = 0.94.$$

$$K_{2ФГ} = \frac{T_2}{T_2 + T_{2ПР}} = \frac{6.26}{6.26 + 0.1} = 0.98.$$

- по формуле (2.11) определяется коэффициент функциональной готовности системы

$$K_{ФГ} = K_{1ФГ} K_{2ФГ} + K_{1ФГ}(1 - K_{2ФГ})(1 - \omega_2) +$$

$$+ K_{2ФГ}(1 - K_{1ФГ})(1 - \omega_1) = 0.952.$$

– по формуле (2.12) вычисляется среднее время простоя системы

$$T_{ПР} = \frac{G_1 k_1 T_{1ПР} + G_2 k_2 T_{2ПР}}{G_1 k_1 + G_2 k_2} = \frac{10^{-6} \cdot 0.17 \cdot 0.1 + 10^{-7} \cdot 0.83 \cdot 0.1}{10^{-6} \cdot 0.17 + 10^{-7} \cdot 0.83} = 0.1\ ч$$

- по формуле (2.13) определяется средняя наработка на ошибку системы

$$T_C = \frac{T_{ПР} K_{ФГ}}{1 - K_{ФГ}} = 1.9\ ч;$$

- по формуле (2.14) вычисляется средняя наработка на полный функциональный отказ системы

$$T_Ф = \frac{T}{\overline{g}_{ФТ}} = 1900\ ч, \text{ где } \overline{g}_{ФТ} = \frac{g_{1ФТ} + g_{2ФТ}}{2} = 10^{-3}.$$

- по формуле (2.16) вычисляется коэффициент оперативной функциональной готовности в течение времени работы смены оперативного персонала информационной системы (в течение времени управления 8 час.)

$$K_{ОФГ}(\tau_{ОП}) \approx K_{ФГ} - \frac{\tau_{ОП}}{T_{ПР}} \overline{g}_{ФТ}(1 - K_{ФГ}) =$$

$$= 0.952 - 0.080 \cdot 0.048 = 0.948$$

Контрольные вопросы:

1. Перечислите требования, предъявляемые к системе показателей функциональной надежности.

2. Какие принципы положены в основу формирования системы показателей функциональной надежности?

3. Приведите и поясните вероятностные показатели функциональной надежности информационной системы относительно каждой отдельной функции.

4. Приведите и поясните временные показатели функциональной надежности информационной системы относительно каждой отдельной функции.

5. Приведите и поясните комплексные вероятностные показатели функциональной надежности информационной системы.

6. Приведите и поясните комплексные временные показатели функциональной надежности информационной системы.

7. Объясните суть метода расчета показателей функциональной надежности системы с помощью фундаментальной матрицы поглощающих Марковских цепей.

8. Объясните суть метода расчета показателей функциональной надежности системы с помощью модифицированного полумарковского топологического метода.

Глава II.3. Функциональная надежность цифровых устройств информационных систем

II.3.1. Сбои цифровых устройств

Предмет расчета функциональной надежности цифровых устройств состоит в определении количественной зависимости безошибочности выходных результатов в работе устройств от сбоев аппаратуры при известном наборе выполняемых функций.

Статическая помехоустойчивость цифровых интегральных схем характеризуется передаточной характеристикой схемы, под которой подразумевается зависимость выходного напряжения схемы от напряжения на одном из входов при постоянных напряжениях на других входах и заданном числе нагрузок.

Реальная передаточная характеристика схемы может быть аппроксимирована кусочно-линейной функцией [5]. На рис. II.3.1 приведен пример типовой передаточной характеристики и указаны ее параметры: a_i (i =1,0) – выходной уровень напряжения логической единицы или нуля; b_i – пороговое значение входного сигнала при переключении типа 1⟶ 0 (i =1) или типа 0 ⟶ I (i = 0); $\bar{b} = \frac{b_1 + b_0}{2}$; возможная ширина зоны переключения $\Delta b = / b_1 - b_0 /$. На типовой передаточной характеристике выделяются три области, соответствующие различным состояниям схемы. Область I – закрытое состояние, область III – открытое

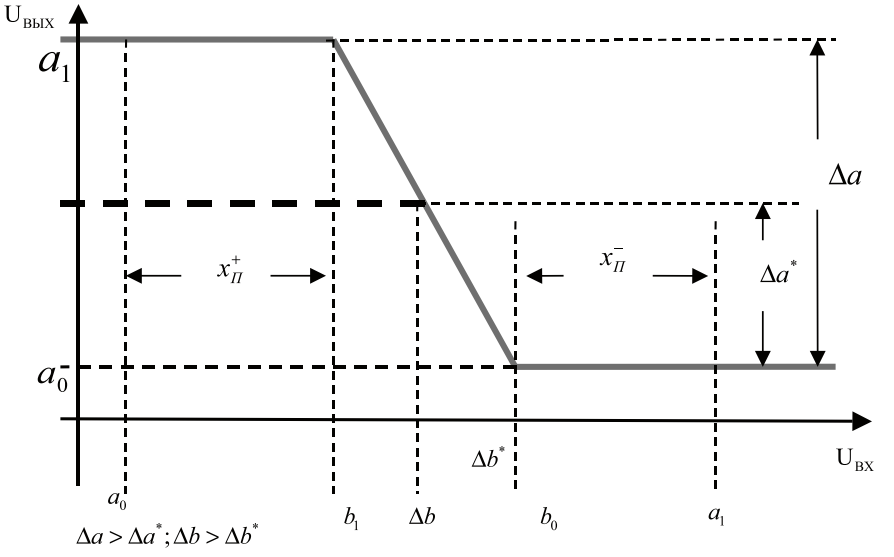


Рис. II.3.1. Типовая передаточная характеристика цифровой интегральной схемы

состояние схемы, а промежуточная область II – состояние переключения. Под воздействием отпирающей помехи возможен переход схемы из области I в область III, т.е. неправильное формирование на выходе схемы кода “I” вместо кода “0” (сбой типа I → 0). Под воздействием запирающей помехи возможен сбой типа 0 → I.

Из рис.II.3.1 следует, что при заданной величине логического перепада Δa технологический разброс пороговых напряжений b_1 и b_0 приводит к уменьшению максимально допустимых амплитуд отпирающей $x_П^+$ и запирающей $x_П^-$ помех, а следовательно, к снижению статической помехоустойчивости схемы. Аналогичный результат имеет место при уменьшении логического перепада Δa, который осуществляется в интересах повышения быстродействия схем. Чем меньше величина логического перепада $\Delta a^* < \Delta a$, тем меньше время переключения схемы $\Delta b^* < \Delta b$ и, следовательно, тем выше ее быстродействие.

Ранее было отмечено, что экспериментальные данные по сбоям интегральных схем отсутствуют. Для оценки характеристик сбоев применяются экспериментально – расчетные методы прогнозирования, основанные на экспериментальных данных о входных сигналах и помехах, передаточных и амплитудно-частотных характеристиках схем. Эти данные являются исходными для расчета диапазона возможных вероятностей сбоя цифрового логического элемента в интегральном исполнении, который часто именуют вентилем по аналогии с транзистором в элементной базе второго поколения. По полученным расчетным значениям сбоев вентилей, известной структуре и реализуемым логическим функциям рассчитываются вероятности и интенсивности сбоев ЦИС.

Рассмотрим более детально суть этого направления в части расчетов вероятностей сбоя вентиля, применяя методику, предложенную в работе [4]. Эта методика основывается на следующих предпосылках:

1) границы правильности однократного срабатывания вентиля полностью определяются границами его передаточной характеристики и диапазонами изменения параметров входных сигналов;

2) каждый входной сигнал представляет собой аддитивную смесь детерминированного полезного сигнала и помехи;

3) помехи по цепям питания и заземления могут быть корректно пересчитаны на вход вентиля и просуммированы со входными помехами.

Поскольку помехоустойчивость ЦИС зависит от многих факторов, то выделяются наиболее существенные из них и определяются максимальные границы изменения передаточной характеристики, длительности и амплитуды импульсных помех. Реальная передаточная характеристика может быть аппроксимирована кусочно-линейной функцией с введением

порогов переключения b_1 и b_0 (рис.II.3.1). В этом случае вероятности правильного появления единичного q и нулевого â сигналов на выходе вентиля соответственно определяются выражениями

$$q = \int_{-\infty}^{b_1} f(x)dx;\ \beta = \int_{b_0}^{\infty} f(x)dx,$$

где $f(x)$ – плотность вероятности смеси сигнала и помехи.

В свою очередь, максимальные вероятности возникновения сбоев “I” и “0” соответственно равны $q_{1\max} = 1 - q = \beta + p_*$, где p_* – вероятность попадания выходного сигнала на участок зоны переключения логического элемента (ЛЭ). Если исключить сбои, обусловленные зоной переключения, то $q_{1\min} = \beta$, $q_{0\min} = q$.

Наличие множества причин, приводящих к случайному изменению смеси сигнала и помехи позволяет предположить, что напряжение этой смеси распределено по нормальному закону с математическим ожиданием M и дисперсией σ^2. Нормированные по выходному напряжению логической “I” значения σ входного сигнала по экспериментальным данным находятся в пределах σ = 0,03-0,1. Анализ технических характеристик вентиля свидетельствует, что случайные значения входных сигналов определяются диапазоном |x|=3 ÷ 20. Поэтому целесообразно определять искомые вероятности по следующим приближенным формулам. При $x_i<0$, когда $M > b_0$,

$$P_i(x_i < 0) = \frac{\exp(-x_i^2/2)}{\sqrt{2\pi}\,|x_i|}, \text{ где } i=1,0.$$

При $x_i>0$, $M < b_1$, $P_i(x_i>0)=1-P_i(x_i<0)$, где i=1,0.

Значения вероятности p_* следующие: $p_* = \beta(x_0<0)$, если осуществляется логический переход типа 0 → I (рис. II.3.1) и $p_* = q(x_1>0)$ при переходе типа I → 0.

По граничным значениям передаточной характеристики вентиля оценивают пределы изменения математического ожидания M смеси входного сигнала и помехи и выбирают приемлемый шаг ΔM его дискретизации. Затем по установленным граничным значениям параметров M и σ^2 входного сигнала и импульсной помехи и граничным значениям результирующих параметров b_1 и b_0 для каждого дискрета ΔM рассчитывают вероятности q_1 и q_0 сбоев относительно логической "I" или "0" соответственно. По полученной совокупности результатов определяют диапазон изменения этих вероятностей. Затем рассчитывают среднюю вероятность сбоя вентиля.

Пример II.3.1. Дан вентиль со сложным инвертором, который является базовым элементом серий ТТЛ. Требуется найти среднюю вероятность сбоя вентиля.

Решение.

Для стандартных условий эксплуатации (t = 20°C и напряжение питания 5 в) нормированные по выходному уровню логической "I" результирующие параметры передаточной характеристики определяются следующими значениями:

$$\bar{b} = 0.375;\ \ \Delta\overline{b/2} = 0.05;\ \ b_1 = \bar{b} - \Delta\overline{b/2} = 0.325;$$
$$b_0 = \bar{b} + \Delta\overline{b/2} = 0.425;$$

Нормированные параметры входного сигнала: $M_c = 0.025 \div 1$; $\Delta M = 0.05$; $\sigma = 0.03 \div 0.05$.

При этих данных диапазон изменения вероятностей сбоев логической "I" и логического "0" вентилей серий ТТЛ соответственно составляют значения

$$1 - q = 10^{-13} \div 10^{-17}.\ 1 - \beta = 10^{-14} \div 10^{-18}.$$

Поскольку эти значения почти симметрично распределяются вокруг средней вероятности сбоя вентиля серий ТТЛ, то средняя вероятность сбоя вентиля равна $\bar{q} = 10^{-16}$.

II.3.2. Расчет правильности выполнения цифровыми устройствами логических функций

II.3.2.1. Методика расчета

В качестве исходных данных для расчета функциональной надежности логического элемента (ЛЭ) предлагается принять вероятности безошибочного прохождения сигналов по каждому входу элемента. Обозначим сигналы на входах ЛЭ в виде $x_1, x_2, ..., x_m$, где m – число входов ЛЭ. Каждый из этих сигналов принимает только два значения «0» и «1» и представляется током или напряжением в импульсной или потенциальной форме различной полярности и/или амплитуды. Тогда вероятность правильного (безошибочного) прохождения сигнала $x_i = 1$ по *i*-му входу ЛЭ ($i = 1,...,m$) обозначается через q, а вероятность правильного прохождения сигнала $x_i = 0$ по *i*-му входу ЛЭ ($i = 1,...,m$) обозначается через β. В свою очередь, $1 - q = \bar{q}$ и $1 - \beta = \bar{\beta}$ – это вероятности возникновения сбоев при поступлении на *i*-ый вход элемента сигналов $x_i = 1$ или $x_i = 0$ соответственно.

Методика содержит три уровня расчета. На третьем (нижнем) уровне определяется безошибочность работы ЛЭ типа И, ИЛИ, НЕ, Усилитель мощности, Запрет и др. На втором уровне рассчитывается безошибочность работы логических схем, в составе которых имеется множество ЛЭ (триггеры, логические узлы равнозначности, дешифраторы, счетчики и др.). На первом (высшем) уровне определяется безошибочность работы многоразрядных дискретных устройств – цифровых устройств (регистров, сумматоров, коммутаторов, мультиплексоров и др.).

II.3.2.2. Расчет правильности работы логических элементов

В качестве основного критерия правильной работы типовых логических элементов применяется вероятность их правильного однократного срабатывания

$$\mathrm{P}_{ЛЭ} = P_{1ЛЭ} + P_{0ЛЭ}, \qquad (3.1)$$

где

$$P_{1ЛЭ} = Bep(f\{x_j, q\} = 1);$$
$$P_{0ЛЭ} = Bep(f\{x_j, \beta\} = 0),$$

f – функция выходного сигнала от всего набора входных сигналов $\{x_j\}$,

где:

$x_j = \begin{pmatrix} 1, \\ 0, \end{pmatrix}$ с вероятностью p_1 или p_0 соответственно,

$j = \overline{1,m}$; $p_1 + p_0 = 1$; m – число входов ЛЭ;

p_1 – вероятность поступления кода “I” на j-й вход элемента;

p_0 – вероятность поступления кода “0” на j-й вход элемента.

Для расчета вероятности $\mathrm{P}_{ЛЭ}$ предложена следующая методика:

а) определяется совокупность всех возможных состояний ЛЭ по причине наличия или отсутствия сбоев в вентилях $\{l_i\}$ и определяется вероятность каждого события $P(l_i)$, где $l_i \in l$.

б) каждому состоянию ЛЭ ставятся в соответствие такие наборы входных сигналов x_j, при поступлении которых на выходе ЛЭ формируется истинное значение выходного сигнала y;

в) вероятность правильного однократного срабатывания ЛЭ рассчитывается по формуле

$$\mathrm{P}_{ЛЭ} = \sum_{j=1}^{2^m} \sum_{l_i=1}^{l_i=k} G_{x_j} A_{ji} P(l_i). \qquad (3.2)$$

где G_{x_j} – вероятность поступления на входы ЛЭ x_j-го набора входных сигналов. Предполагается независимость входных сигналов. Так, если ЛЭ содержит два входа и на первый вход поступает сигнал типа «0», а на второй вход – сигнал типа «1» ($x_j = 0,1$), то $G_{0,1} = p_0 p_1$.

A_{ji} – функция соответствия: $A_{ji} = \begin{cases} 1 \\ 0 \end{cases}$ – при истинном или ложном выходном значении y соответственно; k – число состояний логического элемента; m – число входов.

Пример II.3.2. Рассчитать вероятность правильного однократного срабатывания логического элемента ИЛИ на два входа ($m = 2$).

Решение.

В табл.II.3.1 приведены значения выходного кода y, соответствующие исходам правильного срабатывания логического элемента ИЛИ на два входа. Всего 4 исхода. Вероятность правильного первого исхода $\mathrm{P}_{0,0}$ равна произведению вероятности поступления на входы элемента сигналов $x_1 = 0$ и $x_2 = 0$, которая определяется как $G_{0,0} = p_0 p_0 = p_0^2$, на вероятность правильного их прохождения $P(l_{0,0}) = \beta\beta = \beta^2$, т.е.

$$\mathrm{P}_{0,0} = G_{0,0}\beta^2 = p_0^2\beta^2 \text{ (см. табл. II.3.1).}$$

Табл. II.3.1. Вероятность правильного однократного срабатывания элемента ИЛИ

x_1	x_2	y	G_{x_j}	$P(l_i), i = 1,2,3,4$	$\mathrm{P}_{ИЛИ2}$
0	0	0	$p_0 p_0$	$P(l_{00}) = \beta\beta = \beta^2$	$\mathrm{P}_{0,0} = G_{0,0}\beta^2 = p_0^2\beta^2$
0	1	1	$p_0 p_1$	$P(l_{01}) = \beta q + \bar{\beta} q + \bar{\beta}\bar{q} = 1 - \beta\bar{q}$	$\mathrm{P}_{0,1} = p_0 p_1 (1 - \beta\bar{q})$
1	0	1	$p_0 p_1$	$P(l_{10}) = 1 - \beta\bar{q}$	$\mathrm{P}_{1,0} = p_1 p_0 (1 - \beta\bar{q})$
1	1	1	$p_1 p_1$	$P(l_{11}) = 1 - (1-q)^2$	$\mathrm{P}_{1,1} = p_1^2 [1 - (1-q)^2]$

Вероятность правильного второго исхода будет равна произведению вероятности $G_{0,1}$ на вероятность правильного и неправильного прохождения сигналов в следующих случаях:

а) сигналы $x_1 = 0$ и $x_2 = 1$ проходят правильно – βq;

б) сигнал $x_1 = 0$ проходит неправильно, т.е. приводит к появлению ложного одиночного исхода $y = 1$, а сигнал $x_2 = 1$ проходит правильно и приводит к формированию истинного одиночного исхода $y = 1$. Следовательно, ложный одиночный исход компенсируется истинным исходом и в итоге получен истинный результат с вероятностью $\bar{\beta} q$;

в) оба сигнала $x_1 = 0$ и $x_2 = 1$ проходят неправильно. Это приводит к появлению ложного одиночного исхода $y = 1$ (по входному сигналу $x_1 = 0$) или ложного одиночного исхода $y = 0$ (по входному сигналу $x_2 = 1$). Поскольку по определению $(y = 1) \vee (y = 0) = (y = 1)$, то на выходе ЛЭ будет сформирован истинный результат с вероятностью $\bar{\beta}\bar{q}$.

В итоге получаем $P(l_{01}) = \beta q + \bar{\beta} q + \bar{\beta}\bar{q} = 1 - \beta\bar{q}$. Окончательный результат

$$\mathrm{P}_{0,1} = p_0 p_1 (1 - \beta\bar{q}).$$

Для третьего исхода аналогичным образом получаем следующее выражение

$$\mathrm{P}_{1,0} = p_1 p_0 (1 - \beta \bar{q}) .$$

Заметим, что входы ЛЭ равнозначны и $p_1 p_0 = p_0 p_1$. Следовательно, имеет место равенство вероятностей $\mathrm{P}_{0,1} = \mathrm{P}_{1,0}$.

Вероятность правильного четвертого исхода будет равна произведению вероятности $G_{1,1}$ на вероятность правильного и неправильного прохождения сигналов в следующих случаях:

а) $x_1 = 1$ и $x_2 = 1$ проходят правильно – q^2;

б) сигнал $x_1 = 1$ проходит неправильно, т.е. приводит к появлению ложного одиночного исхода $y = 0$, а сигнал $x_2 = 1$ проходит правильно и приводит к формированию истинного одиночного исхода $y = 1$. Следовательно, ложный одиночный исход компенсируется истинным исходом и в итоге получен истинный результат с вероятностью $\bar{q}q$;

в) сигнал $x_1 = 1$ проходит правильно и приводит к формированию истинного одиночного исхода $y = 1$, а сигнал $x_2 = 1$ проходит неправильно и приводит к появлению ложного одиночного исхода $y = 0$, Следовательно, ложный одиночный исход компенсируется истинным исходом и в итоге получен истинный результат с вероятностью $q\bar{q}$.

Таким образом,

$$\mathrm{P}_{1,1} = p_1^2 [1 - (1 - q)^2] .$$

Учитывая, что

$$G_{0,0} + G_{0,1} + G_{1,0} + G_{1,1} = 1 ,$$

т.е. имеет место полная группа событий, получим формульное выражение для оценки вероятности правильного однократного срабатывания логического элемента ИЛИ на два входа:

$$\mathrm{P}_{ИЛИ2} = \mathrm{P}_{0,0} + 2\mathrm{P}_{0,1} + \mathrm{P}_{1,1} = p_0^2\beta^2 + 2p_0p_1(1-\beta\overline{q}) + p_1^2[1-(1-q)^2]$$

Аналогичным образом, как и в примере II.3.2, можно получить формульное выражение для оценки вероятности правильного однократного срабатывания логического элемента ИЛИ на m входов:

$$\mathrm{P}_{ИЛИm} = p_0^m\beta^m + \sum_{i=1}^{m}\binom{m}{i}p_1^i p_0^{m-i}[1-\beta^{m-i}(1-q)^i], \qquad (3.3)$$

где $P_{0ИЛИm} = p_0^m\beta^m$, $P_{1ИЛИm} = \sum_{i=1}^{m}\binom{m}{i}p_1^i p_0^{m-i}[1-\beta^{m-i}(1-q)^i]$

и $\mathrm{P}_{ИЛИm} = P_{0ИЛИm} + P_{1ИЛИm}$

Проанализируем зависимость правильности работы логического элемента ИЛИ (вероятности $\mathrm{P}_{ИЛИm}$) от числа входов m и вероятностей отсутствия сбоев q и β (рис. II.3.2). При отсутствии сбоев ($\overline{\beta} = 0$) в ходе обработки входного сигнала $x_i = 0$ вероятность $\mathrm{P}_{ИЛИm}$ растет с увеличением числа входов (рис. II.3.2 а – сплошные линии).

Из графиков зависимости $\mathrm{P}_{ИЛИm} = f(p_1, p_0)$ при $q = \beta = 0.5$ (рис. II.3.2 б – сплошные линии) видно, что безошибочность элемента ИЛИ растет при повышении вероятности поступления кода «1» (p_1). Это означает, что элемент ИЛИ обладает функциональной избыточностью по отношению ко входным сигналам $x_j = 1$. Действительно, если подается код «1» на каждый

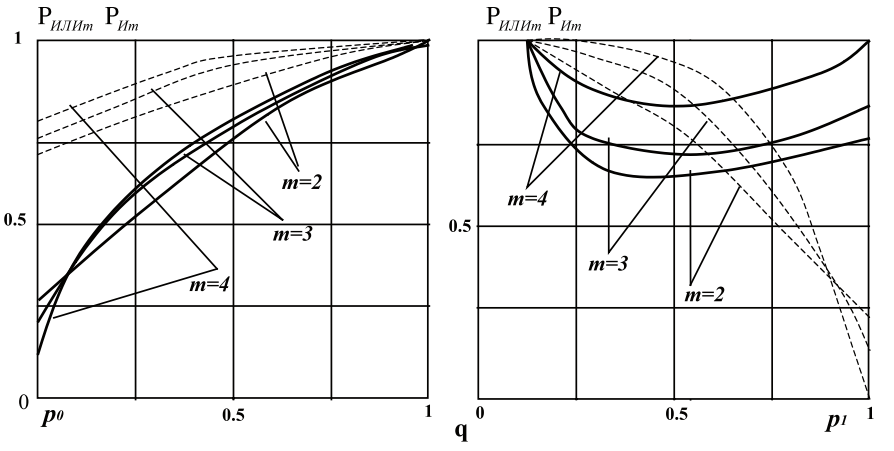


***Рис. II.3.2. Зависимость надежности элементов* ИЛИ *(-) и* И *(---) от: – числа входов m и вероятности q при вероятностях* $\beta = 1$ *и* $p_1 = p_0 = 0{,}5$*; – вероятностей* p_1 *и* p_0 *при* $q = \beta = 0{,}5$.**

вход элемента, то на выходе сформируется правильный код «1» даже при возникновении сбоя при прохождении одного из двух сигналов в элементе. Этот результат тем более справедлив для элементов ИЛИ с большим числом входов.

Вероятность $P_{ИЛИm}$ правильной работы элемента ИЛИ минимальна при максимальной неопределенности о содержании входной информации, т.е. когда $p_1 = p_0 = 0.5$. По мере роста информативности о входных сигналах становится эффективней функциональная избыточность элемента ИЛИ и повышается правильность его работы (рис. II.3.2 б).

Пример II.3.3. Рассчитать вероятность правильного однократного срабатывания логического элемента И на два входа ($m = 2$).

Решение.

В табл. II.3.2 приведены значения выходного кода y, соответствующие 4 исходам правильного срабатывания логического элемента И, а также формульные выражения промежуточных $G_{x_j}, P(l_i)$ и выходных вероятностей $P_{0,0}...P_{1,1}$ безошибочной ра-

боты элемента для каждой из из 4 входных комбинаций сигналов, полученные по аналогии с рассуждениями примера II.3.2.

Табл. II.3.2. Вероятность правильного однократного срабатывания элемента И

x_1	x_2	y	G_{x_j}	$P(l_i), i=1,2,3,4$	$\mathrm{P}_{И2}$
0	0	0	$p_0 p_0$	$P(l_{00}) = 1-(1-\beta)^2$	$\mathrm{P}_{0,0} = G_{0,0}\beta^2 = p_0^2[1-(1-\beta)^2]$
0	1	0	$p_0 p_1$	$P(l_{01}) = 1-q(1-\beta)$	$\mathrm{P}_{0,1} = p_0 p_1[1-q(1-\beta)]$
1	0	0	$p_1 p_0$	$P(l_{10}) = 1-q(1-\beta)$	$\mathrm{P}_{1,0} = p_1 p_0[1-q(1-\beta)]$
1	1	1	$p_1 p_1$	$P(l_{11}) = q^2$	$\mathrm{P}_{1,1} = p_1^2 q^2$

Результирующее выражение вероятности правильного однократного срабатывания элемента И на два входа имеет следующий вид

$$\mathrm{P}_{И2} = p_0^2[1-(1-\beta)^2] + 2p_0 p_1[1-q(1-\beta)] + p_1^2 q^2 .$$

Аналогичным образом, как и в примере II.3.3, можно получить формульное выражение для оценки вероятности правильного однократного срабатывания логического элемента И на m входов:

$$\mathrm{P}_{Иm} = p_1^m q^m + \sum_{i=1}^{m} \binom{m}{i} p_0^i p_1^{m-i}[1-q^{m-i}(1-\beta)^i], \qquad (3.4)$$

где $P_{1Иm} = p_1^m q^m$, $P_{0Иm} = \sum_{i=1}^{m} \binom{m}{i} p_0^i p_1^{m-i}[1-q^{m-i}(1-\beta)^i]$

и $\mathrm{P}_{Иm} = P_{1ИЛИm} + P_{0ИЛИm}$

Анализ графических зависимостей

$\mathrm{P}_{Ит} = f(m, q, \beta, p_1, p_0)$ (рис. II.3.2 – штриховые линии) показывает, что:

- элемент И функционально избыточен относительно входных сигналов $x_j = 0$;

- вероятность $\mathrm{P}_{Ит}$ растет при увеличении числа входов *m*;

- при входных комбинациях длиной *m* предпочтительнее кодировать их большим числом нулей, чем единиц.

II.3.2.3. Расчет правильности работы логических схем

Функциональная надежность логических схем, в состав которых входит несколько логических элементов, определяется также при помощи формул (3.1) и (3.2) следующим образом:

а) схема представляется в виде *n*-уровневой иерархической структуры, на каждом уровне которой сгруппированы равноудаленные от выхода схемы логические элементы. Набор сигналов $\{x_j(n)\}$ поступает на входы элементов периферийного уровня схемы;

б) по формуле (3.2) рассчитывается функциональная надежность каждого элемента периферийного (*n*-го) уровня и согласно формуле (3.1) расчленяется на составляющие $P_{0ЛЭ}(n)$ и $P_{1ЛЭ}(n)$. Это позволяет сформировать набор входных сигналов $\{x_j(n-1)\}$ для очередного (*n*-1) уровня:

$$x_j(n-1) = \begin{cases} 1 \to P_{1ЛЭ}(n); \\ 0 \to P_{0ЛЭ}(n) \end{cases}.$$

Таким образом, информация о функциональной надежности элементов *n*-го уровня содержится в вероятностях поступления кодов «0» и «1» на входы элементов *(n-1)*-го уровня.

В дальнейших расчетах надежности логической схемы отпадает необходимость в определении набора вероятностей $\{\mathrm{P}_{ЛЭ}(n)\}$;

в) последовательно формируется набор входных сигналов для элементов 0-го уровня

$$x_j(0) = \begin{cases} 1 \to P_{1ЛЭ}(1); \\ 0 \to P_{0ЛЭ}(1) \end{cases}.$$

В вероятностях $P_{1ЛЭ}(1)$ и $P_{0ЛЭ}(1)$ содержится вся информация об элементах предыдущих уровней.

г) по формуле (3.2) определяется вероятность правильного однократного срабатывания элементов нулевого уровня $\mathrm{P}_{ЛЭ}(0)$, которая и является показателем безошибочности работы всего устройства.

Пример II.3.4. Определить вероятность правильной работы логического узла равнозначности.

Решение.

Узел равнозначности (рис. II.3.3) содержит два элемента И (&) на два входа, два элемента НЕ и один элемент ИЛИ (1) на

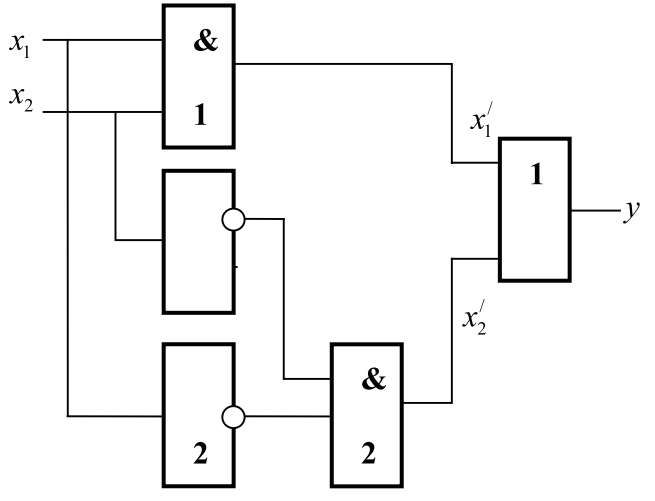


Рис. II.3.3. Логическая схема узла равнозначности

два входа. В таблице истинности II.3.3. приведены четыре возможных исхода выходной величины *y*.

Табл. II.3.3.Таблица истинности узла равнозначности

x_1	x_2	$x_1^{/}$	$x_2^{/}$	y
0	0	0	1	1
1	0	0	0	0
0	1	0	0	0
1	1	1	0	1

Определим вероятности выполнения условий 1 и 4, при которых на выходе логического узла формируется положительный результат «1».

Условие первое.

$$P_1(1) = P(x_1^{/} = 0 / x_1 = x_2 = 0) P(x_2^{/} = 1 / x_1 = x_2 = 0) p_0^2,$$

где $P(x_1^{/} = 0 / x_1 = x_2 = 0) = 1 - (1 - \beta)^2$;

$$P(x_2^{/} = 1 / x_1 = x_2 = 0) = \beta_{HE}^2 q^2,$$

где β_{HE} – вероятность правильного прохождения сигнала $x = 0$ в логическом элементе НЕ.

Следовательно, $P_1(1) = [1 - (1 - \beta)^2] \beta_{HE}^2 q^2 p_0^2$.

– Условие четвертое.

$$P_4(1) = P(x_1^{/} = 1 / x_1 = x_2 = 1) P(x_2^{/} = 0 / x_1 = x_2 = 1) p_1^2,$$

где $P(x_1^{/} = 1 / x_1 = x_2 = 1) = q^2$;

$$P(x_2^{/} = 0 / x_1 = x_2 = 1) = q_{HE}^2 [1 - (1 - \beta)^2],$$

где q_{HE} - вероятность правильного прохождения сигнала $x = 1$ в логическом элементе НЕ

Следовательно, $P_4(1) = q^2 q_{HE}^2 [1 - (1 - \beta)^2 p_1^2$.

Таким образом, вероятность правильного формирования кода «1» на выходе логического узла равнозначности с учетом формулы (3.4) для выходного элемента ИЛИ узла равна

$$P_{1\approx} = [P_1(1) + P_4(1)][1 - \beta(1 - q)].$$

Код «0» формируется на выходе узла равнозначности при втором и третьем исходах. Вероятности указанных условий равны:

– Условие второе

$$P_2(0) = P(x_1^{/} = 0 / x_1 = 1, x_2 = 0) P(x_2^{/} = 1 / x_1 = 1, x_2 = 0) p_0 p_1,$$

$$\text{где } P(x_1^{/} = 0 / x_1 = 1, x_2 = 0) = 1 - q(1 - \beta);$$

$$P(x_2^{/} = 0 / x_1 = 1, x_2 = 0) = \beta_{HE} q_{HE} [1 - q(1 - \beta)].$$

– Условие третье

$$P_3(0) = P(x_1^{/} = 0 / x_1 = 0, x_2 = 1) P(x_2^{/} = 0 / x_1 = 0, x_2 = 1) p_0 p_1,$$

$$\text{где } P(x_1^{/} = 0 / x_1 = 0, x_2 = 1) = 1 - q(1 - \beta);$$

$$P(x_2^{/} = 0 / x_1 = 0, x_2 = 1) = \beta_{HE} q_{HE} [1 - q(1 - \beta)].$$

Таким образом, $P_2(0) = P_3(0) = P(0)$ и вероятность правильного формирования кода «0» на выходе узла равна

$$P_{0\approx} = [P_2(0) + P_3(0)]\beta^2 = 2P(0)\beta^2.$$

Результирующая вероятность правильного однократного срабатывания логического узла равнозначности

$$\mathrm{P}_{\approx} = P_{1\approx} + P_{0\approx} = [P_1(1) + P_4(1)][1 - \beta(1-q)] + 2P(0)\beta^2 .$$

Заметим, что при идеальной функциональной надежности составных элементов И, ИЛИ, НЕ вероятность правильности работы логического узла равнозначности равна 1. Действительно, при этом предположении

$$q = \beta = q_{HE} = \beta_{HE} = 1;$$
$$P_1(1) + P_4(1) + P_2(0) + P_3(0) = p_0^2 + p_1^2 + 2p_0p_1 = 1.$$

Анализ зависимости правильности узла равнозначности от содержания входной информации (от вероятностей p_0, p_1) в предположении, что с поступлением на вход узла кода «0» состояния элементов не меняются, показывает, во – первых, что правильность работы узла максимальна при максимальной энтропии во входной информации ($p_0 = p_1 = 0.5$) и, во-вторых, что правильность работы узла в первую очередь зависит от функциональной надежности составных элементов И.

Пример II.3.5. Определить вероятность правильной однократной работы одноступенного дешифратора на два входа.

Решение.

На рис. II.3.4 показана логическая схема этого дешифратора, включающая четыре идентичных элемента И. Для данного дешифратора возможны четыре исхода. Вероятность правильного однократного срабатывания дешифратора означает: требуемый исход состоялся верно при условии, что остальные исходы не наступили. Например, на входы дешифратора поступили сигналы $\bar{x}_1 = 1, \bar{x}_2 = 1$. В этом случае должно наступить событие A, равное $y_1 = 1; y_2 = y_3 = y_4 = 0$.

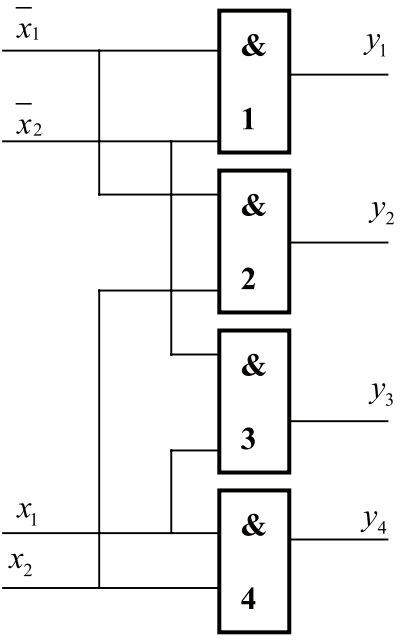


Рис. II.3.4. Логическая схема одноступенного дешифратора на два входа

Вероятность правильного осуществления этого события

$$P(A) = P(y_1 = 1 / y_2 = y_3 = y_4 = 0) =$$
$$= P(y_1 = 1)P(y_2 = 0)P(y_3 = 0)P(y_4 = 0).$$

где в соответствии с формулой (3.4) и табл. II.3.2

$$P(y_1 = 1) = q^2; \quad P(y_2 = 0) = P(y_3 = 0) = 1 - q(1-\beta);$$
$$P(y_4 = 0) = 1 - (1-\beta)^2.$$

Таким образом, вероятность правильного осуществления события A равна

$$P(y_1 = 1) = q^2; \quad P(y_2 = 0) = P(y_3 = 0) = 1 - q(1-\beta);$$
$$P(y_4 = 0) = 1 - (1-\beta)^2$$

$$P(A) = P(y_1) = q^2[1 - q(1-\beta)]^2[1 - (1-\beta)^2].$$

При сигналах $\bar{x}_1 = 1, x_2 = 1$ должно наступить событие, которое обозначим как событие *Б*, которое согласно схеме на рис. II.3.4 должно привести к положительному результату $y_2 = 1 / y_1 = y_3 = y_4 = 0$. Вероятность правильного осуществления события *Б* равна

$$P(Б) = P(y_2) = P(y_2 = 1 / y_1 = y_3 = y_4 = 0) =$$
$$= q^2[1 - q(1-\beta)]^2[1 - (1-\beta)^2].$$

Аналогичные формульные выражения имеют место относительно правильного осуществления событий

$$B\ (P(B) = P(y_3))\ \text{и}\ C\ (P(C) = P(y_4)).$$

Следовательно, при различных фиксированных значениях входных сигналов наблюдается равенство вероятностей всех четырех исходов:

$$P(y_1) = P(y_2) = P(y_3) = P(y_4).$$

В общем случае на входы дешифратора поступают сигналы с вероятностями

$$P(x_1 = 0) = P(\bar{x}_1 = 1);\ \ P(x_2 = 0) = P(\bar{x}_2 = 1);$$

$$P(x_1 = 1) = P(\bar{x}_1 = 0);\ \ P(x_2 = 1) = P(\bar{x}_2 = 0);$$

Вероятности поступления комбинаций входных сигналов следующие

$$G_{0,0} = p_0^2; G_{0,1} = G_{1,0} = p_0 p_1; G_{1,1} = p_1^2 .$$

Таким образом, вероятность правильного однократного срабатывания одноступенного дешифратора с двумя входами равна

$$\mathrm{P}_{DC1}(2) = P(y_i)(G_{0,0} + 2G_{0,1} + G_{1,0}) = q^2[1 - q(1-\beta)]^2[1-(1-\beta)^2]$$

Обобщая формулу расчета безошибочности одноступенного дешифратора на два входа, полученную в примере II.3.5, на произвольное число входов, находим общее выражение безошибочности одноступенных дешифраторов

$$\mathrm{P}_{DC1}(m) = q^m \prod_{i=1}^{m} [1 - q^{m-i}(1-\beta)^i]^{C_m^i} .$$

Для удобства пользования данной формулой представим ее в логарифмической форме

$$\ln \mathrm{P}_{DC1}(m) = m \ln q + \sum_{i=1}^{m} \binom{m}{i} \ln[1 - q^{m-i}(1-\beta)^i] . \qquad (3.5)$$

Полученные аналитические результаты свидетельствуют о том, что правильность работы дешифратора не зависит от содержания входной информации, а следовательно, от ошибок в работе логических схем подчиненных уровней. Дешифраторы не обладают логической избыточностью по отношению к сбоям составных элементов.

II.3.3. Расчет правильности работы цифровых устройств

II.3.3.1. Методика расчета

Рассмотрим особенности расчета вероятности правильной однократной работы многоразрядных функциональных частей (ФЧ) – цифровых устройств.

Многоразрядными ФЧ будем называть дискретные устройства, осуществляющие прием, коммутацию и преобразование многоразрядных кодов. К ним относятся регистры, сумматоры, коммутаторы, мультиплексоры, узлы анализа результатов. Особенностью этих устройств является то, что при выполнении арифметических операций над числами с мантиссой, а также при передаче или анализе их допустимо искажение результатов в младших разрядах с вероятностью $P_{доп}$, начиная с *(k+1)*-го разряда.

Анализ показывает, что если числа нормализованы и имеют одинаковую мантиссу, а порядки разные, то при сбое в одном и том же разряде мантиссы они изменяются на одну и ту же относительную величину. Если же порядки чисел равны, а мантиссы разные, то относительная величина искажения таких чисел в результате сбоя в одном и том же разряде мантиссы зависит от величины мантиссы тем больше, чем меньше мантисса.

Таким образом, при определении значения k необходимо ориентироваться на число с минимально возможной нормализованной мантиссой. Порядок числа при этом значения не имеет, и, следовательно, он может быть и нулевым.

При обработке информации в двоичной системе счисления вес старшего из n разрядов мантиссы равен 2^{-1}, а вес *(k+1)*-го разряда равен 2^{-k}.

Поскольку минимально возможная нормализованная мантисса представляется единицей в самом старшем разряде с нулями

во всех остальных, то очевидно следующее соотношение, устанавливающее зависимость между величиной *k* и максимально допустимым в результате сбоя относительным искажением чисел:

$$2^{-k} > \delta_{ДОП} \geq \sum_{i=0}^{n-k} 2^{-(k-1+i)} \qquad (3.6)$$

При заданном $\delta_{ДОП}$ по формуле (3.6) устанавливается номер разряда мантиссы *(k+1)*, начиная с которого с определенной вероятностью допустимо искажение информации в младших разрядах.

Вероятность $P_{доп}$ может быть определена по следующей формуле:

$$P_{ДОП} = \sum_{s} h_{sy} \Phi_{sy} \qquad (3.7)$$

где h_{sy} – вероятность того, что в результате однократного срабатывания цифровое устройство относительно *n-k* младших разрядов мантиссы может находиться в *s*-ом состоянии, Φ_{sy} – условный показатель отсутствия искажения информации в *s*-ом состоянии; определяется с учетом формулы (3.6). Так, при искажении только одного *(k+1)*-го разряда

$$\Phi_{sy/k+1} = 1 - 2^{-(k-1)}.$$

Полагая, что все разряды устройства независимы и однотипны и каждый из них может находиться лишь в двух состояниях (правильного или ошибочного результата), можно по формуле полной вероятности и при помощи выражения (3.7) установить, что

$$P_{ДОП} = \sum_{i=0}^{n-k} \mathrm{P}_y^{n-k-i} q_y^i [\binom{n-k}{i} - i\sum_{j=0}^{n-k} 2^{-(k-1+j)}], \qquad (3.8)$$

а вероятность правильного однократного срабатывания многоразрядной функциональной части, осуществляющей обработку мантиссы числа, равна

$$\mathrm{P}_{МФЧ} = \mathrm{P}_y^k P_{ДОП}, \qquad (3.9)$$

где P_y – вероятность правильного однократного срабатывания устройства одного разряда.

Формулы (3.8) и (3.9) применимы для расчета безошибочности работы регистров, мультиплексоров, коммутаторов, узлов анализа результатов.

Если все эти устройства принимают, передают или анализируют числа с порядком и мантиссой, то

$$\mathrm{P}_{МФЧ} = \mathrm{P}_y^{r+k} P_{ДОП},$$

где r – количество разрядов порядка.

В том случае, если обрабатываются управляющие числа, то $\mathrm{P}_{МФЧ} = \mathrm{P}_y^{r+k}$ и информационная избыточность отсутствует.

Расчет безошибочности работы сумматоров имеет специфические особенности. Это обусловлено тем, что каждый разряд имеет два выхода – выход суммы S и выход переноса E. Кроме того, достоверность работы устройств последующих разрядов зависит от достоверности работы устройств предыдущих разрядов.

Вероятность правильного однократного срабатывания одного разряда многоразрядного сумматора зависит от вероятности поступления единичного и нулевого значений переноса из пре-

дыдущего разряда. Если предположить, что вероятности нулей и единиц в разрядах слагаемых не зависят от номера разряда, то достоверность работы *n*-разрядного сумматора может быть рассчитана следующим образом.

Полагаем, что для первого разряда сумматора известны вероятности правильного формирования суммы и переноса в следующий разряд при фиксированных равных 0 и I значениях переноса из предыдущего левого разряда. Представим эти вероятности двумя матрицами:

$$S = \begin{pmatrix} p_{0s0}; p_{0s1} \\ p_{1s0}; p_{1s1} \end{pmatrix} \quad E = \begin{pmatrix} p_{0e0}; p_{0e1} \\ p_{1e0}; p_{1e1} \end{pmatrix}$$

где, например, p_{0s1} обозначает вероятность правильного формирования суммы, равной коду 0, при фиксированном равном 1 значении переноса из предыдущего разряда.

Истинные значения вероятностей нулевого и единичного значений переноса из нулевого разряда в первый представляются в виде вектора-столбца $P_0 = \begin{pmatrix} p_{0e}^0 \\ p_{1e}^0 \end{pmatrix}$.

Введем единичный вектор-строку $V = (11)$.

Так как вероятность однократного правильного срабатывания одного разряда сумматора равна произведению вероятностей правильного формирования суммы и переноса, то она выражается матричным соотношением:

$$\mathrm{P}_{s1} = VEP_0VSP_0 .$$

Соответственно для двухразрядного сумматора:

$$\mathrm{P}_{s2} = VEP_1VSP_1VSP_0, \text{ где } P_1 = \begin{pmatrix} p^1_{0e} \\ p^1_{1e} \end{pmatrix}.$$

Наконец, для *n* – разрядного сумматора:

$$\mathrm{P}_{s2} = VEP_{n-1}\prod_{i=0}^{n-1} VSP_i, \text{ где } P_i = \begin{pmatrix} p^i_{0e} \\ p^i_{1e} \end{pmatrix}$$

Принимая во внимание, что

$$P_{n-1} = E^{n-1}P_0 \text{ и } P_i = E^iP_0,$$

получаем окончательно:

$$\mathrm{P}_{sn} = VE^nP_0\prod_{i=0}^{n-1} VSE^iP_0$$

Таким образом, для расчета вероятности правильного однократного срабатывания многоразрядного сумматора достаточно определить вероятности правильного формирования единичного и нулевого значений одного разряда суммы и переноса.

Возможность искажения результатов в работе младших разрядах сумматоров со сквозным переносом, осуществляющих действия над мантиссами, следует учитывать более осторожно, чем в ранее рассмотренных многоразрядных функциональных частях. Это объясняется тем, что сбой в цепи формирования переноса в любом, даже в самом младшем разряде может отразиться на старшем разряде суммы. Учитывая это обстоятельство, сумматоры со сквозным переносом информации следует рассматривать как неизбыточные функциональные части.

II.3.3.2. Принцип построения алгоритма расчета правильности работы цифровых устройств

Расчетчика нельзя освободить от необходимости знать функционирование исследуемого объекта. Однако от непроизводительных затрат времени на выполнение сложных и громоздких расчетов освободить можно и должно. С этой целью следует применять типовую программу расчетов правильности работы цифровых устройств (ФЧ).

Рассмотрим принцип построения типового алгоритма расчета правильности работы ФЧ.

Функциональная часть (логические элементы, логические схемы, цифровые устройства) рассматривается как многоуровневая иерархическая структура. Если на входе ее с вероятностью Q_j действует комбинация сигналов, то она является входной для элементов *n*-го уровня, а комбинация их выходных сигналов, в свою очередь, является входной для элементов (*n-1*)-го уровня. Вероятность правильного формирования сигналов на входах элементов *(n-1)*-го уровня при *j*-ой комбинация входных сигналов:

$$Q_{(n-1)j} = Q_{nj} \prod_{i_n=1}^{J_n} \mathrm{P}_j(i_n),$$

где J_n – число элементов на *n*-ом уровне, $\mathrm{P}_j(i_n)$ – вероятность правильного срабатывания *i*-го логического элемента *n*-го уровня при *j*-ой комбинации входных сигналов.

Вероятность правильного формирования выходной комбинации элементами *(n – 1)*-го уровня, которая является входной для элементов *(n – 2)*-го уровня, определяется аналогично:

$$Q_{(n-2)j} = Q_{(n-1)j} \prod_{i_{n-1}=1}^{J_{n-1}} \mathrm{P}_j(i_{n-1})$$

и так далее.

Наконец, вероятность правильного формирования выходной комбинации элементами *1*-го уровня (элементами 0-го уровня являются выходы ФЧ) определяется соотношением:

$$Q_{0j} = Q_{1j} \prod_{i_1=1}^{J_1} \mathrm{P}_j(i_1)$$

где $Q_{0j} = \mathrm{P}_{\Phi Чj}$ – вероятность правильного однократного срабатывания ФЧ при j -ом наборе входных сигналов.

Выражая Q_{ij} через $Q_{(i+1)j}$ *(i=1,...,n-1)*, можно установить, что

$$\mathrm{P}_{\Phi Чj} = Q_j \prod_{i=1}^{J} \mathrm{P}_j(i),$$

где J – общее число элементов в ФЧ.

С учетом всех комбинаций входных сигналов

$$\mathrm{P}_{\Phi Ч} = \sum_{j=1}^{2^m} Q_j \prod_{i=1}^{J} \mathrm{P}_j i)$$

где m – число входов ФЧ.

При расчете на компьютере ФЧ представляется ориентированным графом *G(X,U)*, вершины которого X соответствуют логическим элементам, а дуги U – связям между ними. Входам и выходам ФЧ ставятся в соответствие фиктивные вершины. Вершины графа G нумеруются в порядке участия соответствующих элементов ФЧ в формировании выходных сигналов. В таком же порядке нумеруются и дуги.

Структура ФЧ задается квадратной матрицей смежности $S = (s_{ij})$ размерности *(N×N),* где N – количество вершин в графе, $s_{ij} = \begin{cases} 1 \\ 0 \end{cases}$, если есть (1) или нет (0) дуги между вершинами i и j соответственно.

а также матрицей элементов $U = (\upsilon_{ij})$ размерности $(k*N)$, где k – количество различных типов логических элементов, учитываемых в алгоритме,

$$\upsilon_{ij} = \begin{cases} 1, i = k, j = k \\ 0, i = k, j \neq k \end{cases},$$

Причем, $i = 1,...,k; j = 1,...,N$.

Кроме того, на основании анализа входных параметров и передаточных характеристик задаются вероятности q, β, $q_{НЕ}$, $\beta_{НЕ}$, $q_{УМ}$, $\beta_{УМ}$, которые отражают вероятности сбоев при смене кодов $0 \rightarrow I$ и $I \rightarrow 0$ соответственно в вентилях, а также в элементах НЕ и усилителях мощности.

Для определения вершин графа и связывающих их дуг вводится вектор-строка r_i с числом элементов, равным числу вершин графа,

$$r_i = (0...010...0),$$

где 1 соответствует номеру i-ой анализируемой вершины.

Номера дуг, сходящихся в i-ой вершине, определяются по результату произведения $r_i S^T$, где S^T – транспонированная матрица S.

Номера дуг, исходящих из i-ой вершины, определяются по результату произведения $r_i S$.

Число и номера входных вершин графа находятся из произведения $VS = (0...0...11...1)$, где V – единичная вектор-строка. Число и место нулей в полученном произведении и есть искомый результат.

С помощью вектора – строки $VS^T = (1...11...0...0)$ можно установить по числу и месту нулей число и место выходных вершин графа.

С помощью данного алгоритма можно также последовательно рассчитать правильность работы многоразрядных ФЧ, имеющих рекуррентный алгоритм работы (например, сумматоров). Для этого граф G, представляющий структуру разряда сумматора, последовательно разбивается на два подграфа G_l $(l=1,2)$, один из которых представляет элементы канала суммы, а другой – элементы канала переноса. Затем по описанному выше алгоритму по каждому подграфу рассчитываются вероятности правильного однократного срабатывания ФЧ и подставляются в приведенную выше формулу расчета правильности работы сумматора в целом.

II.3.3.3. Прогнозирование сбойных ошибок цифровых устройств

Сбойные ошибки – это ошибки в выполнении цифровыми устройствами логических функций, вызванные сбоями их составных элементов. При прогнозирующих расчетах правильности работы цифровых устройств необходимо учитывать неодновременность работы составных интегральных схем при реализации предусмотренных функций. При обработке информации имеет место естественная временная избыточность в работе устройства, присущая алгоритмам функционирования цифровой техники.

Для иллюстрации этого положения рассмотрим пример.

Пусть некоторая функция последовательно и многократно выполняется тремя цифровыми элементами в составе цифрового устройства. Для наглядности предположим, что каждый элемент выполняет треть функции за время $\Delta\tau$ с одинаковой вероятностью правильного однократного выполнения $p = 1 - g$, где g- вероятность возникновения сбойной ошибки при однократной работе элемента устройства в течение времени Δt.

При геометрическом законе распределения средняя наработка до сбойной ошибки элемента устройства равна $T_Э = \Delta\tau / g$, а интенсивность $\lambda_Э = g / \Delta\tau$.

Определяем вероятность однократного правильного выполнения устройством заданной функции $P_y = p^3$. Очередное выполнение функции осуществляется через интервал времени $3\Delta\tau$ с момента начала предыдущего выполнения. При геометрическом распределении случайного количества интервалов времени $3\Delta\tau$ до возникновения сбойной ошибки в решении задачи находим, что искомый показатель равен:

$$T_y = 3\Delta\tau / (1-p^3).$$

Учитывая, что $p = 1 - g$ и вероятность g на много порядков меньше 1, можно, не оговаривая величину погрешности, записать, что

$$T_y = 3\Delta\tau / (1-p^3) = 3\Delta\tau / 3g = \frac{\Delta\tau}{g}.$$

Таким образом, правильность работы цифрового устройства с последовательной обработкой информации соизмерима с правильностью работы одного составного элемента устройства.

Простое суммирование интенсивностей сбойных ошибок трех последовательно соединенных элементов в устройстве приведет нас к следующему выражению средней наработки устройства до сбойной ошибки: $T_y = \Delta\tau / 3g$. Однако этот результат

существенно занижен, так как не учитывает тот факт, что при выполнении функции одним элементом два других простаивают и поэтому сбои в них в течение времени $2\Delta\tau$ не влияют на результаты выполнения функции.

В общем случае формула оценки средней вероятности правильной однократной работы устройства с последовательной обработкой информации имеет следующий вид:

$$P_y = \prod_{i=1}^{N} P_i \text{, где } N \text{ – количество ИС в цифровом устройстве,}$$

а P_i – вероятность правильной однократной работы интегральной схемы в составе устройства без избыточности.

При циклической работе устройства с длительностью цикла $\prod_{i=1}^{N} \Delta\tau_i$ ($\Delta\tau_i$ – время переключения *i*-й интегральной схемы, *i=1,...,N*) случайное количество циклов до возникновения сбойной ошибки описывается геометрическим распределением с математическим ожиданием

$$T_y = \sum_{i=1}^{N} \frac{\Delta\tau_i}{1 - \prod_{i=1}^{N} P_i} \qquad (3.10)$$

С помощью формулы (3.10) можно определить и интенсивность сбойных ошибок цифрового устройства $\lambda_y = 1/T_y$.

Если предположить равными длительности задержек сигналов в работе составных интегральных схем и равными вероятности их правильной однократной работы $\Delta\tau_j = \Delta\tau, P_j = P$, то получим удобные для анализа следующие простые формульные выражения вероятности правильной однократной работы цифрового устройства

$$P_{УВ} = P^N;$$
$$T_{УВ} = \frac{N\Delta\tau}{1-P^N} \approx \frac{\Delta\tau}{g} \qquad (3.11)$$

поскольку с погрешностью, не превышающей второго порядка малости, $P^N \approx 1 - N(1-P) = 1 - Ng$.

Формулы (3.11) определяют верхние граничные значения показателей правильности работы цифрового устройства. Они получены в предположении последовательной обработки информации, что означает работу только одной интегральной схемы в любом произвольно выбранном такте обработки информации.

Нижние граничные значения этих показателей можно установить, приняв, что в устройстве идеально реализована конвейерная обработка информации.

Пусть в рассмотренном ранее устройстве из трех элементов идеально реализована арифметическая конвейерная обработка информации. Тогда в первом цикле времени длительностью $3\Delta\tau$ однократно выполнена вся функция с вероятностью правильного выполнения p^3, две трети функции с вероятностью p^2, одна треть с вероятностью p (рис. II.3.5). Во втором цикле дважды выполнена вся функция и один раз две трети. В третьем и последующих циклах вся функция выполняется по три раза. С третьего цикла начинается установившийся режим работы цифрового устройства с конвейерной обработкой информации. Этот режим обеспечивает максимальную производительность данного устройства. Элементы устройства не простаивают, естественная временная избыточность отсутствует.

В общем случае при *N* элементов в устройстве с конвейерной обработкой информации через *N-1* такт с момента начала обработки информации работают все *N* элементов. Показатели правильности работы устройства при условии, что средняя

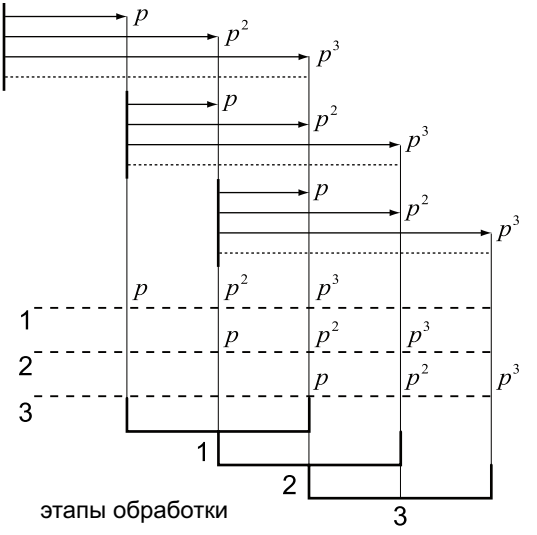


Рис. II.3.5. Иллюстрация схемы определения правильности выполнения процесса конвейерной обработки информации цифровым устройством

вероятность правильности работы элемента равна $\overline{P}$, имеют следующий вид:

$$P_{УН} = (P^N)^N = P^{N^2};\ T_{УН} = \frac{\mathrm{N}\Delta\mathrm{t}}{1\text{-}P^{N^2}} \approx \frac{\Delta t}{Ng}.$$

Таким образом, для прогнозирующих расчетов правильности работы цифровых устройств с учетом возможной параллельной работы их составных элементов применимы следующие формулы:

$$P_У = \prod_{i=1}^{rN} P_i;\ T_У = \frac{\sum_{\mathrm{i=1}}^{\mathrm{N}} \Delta \mathrm{t_i}}{1\text{-}\prod_{\mathrm{i=1}}^{\mathrm{rN}} \overline{\mathrm{P_i}}}, \text{ где } 1 \le r \le N.$$

Коэффициент r определяет, какое количество элементов из общего числа N в устройстве работает параллельно.

II.3.4. Расчет надежности функциональных структур

В п. II.2.1 изложены принципы формирования системы показателей функциональной надежности информационной системы. Была показана целесообразность выделения при расчетах функциональной надежности функциональных частей (ФЧ) и функциональных структур (ФС). При этом подразумевалось, что ФЧ – это материальный объект определенного функционального назначения (например, цифровые устройства). К категории ФЧ относятся также микрооперации. В любой дискретной системе всегда содержится конечное множество ФЧ.

Функциональная структура (ФС) – это материальное воплощение определенного информационного процесса путем формирования совокупности множеств ФЧ, объединенных согласно алгоритму данного процесса. Наиболее распространенным средством изображения функциональной структуры (ФС) является структурная схема, которая представляет собой конфигурацию операционных и решающих элементов, связанных направленными дугами. На структурной схеме можно выделить множество маршрутов, которые исходят из данного начального элемента и заканчиваются в операционных элементах, не содержащих направленных дуг.

Каждый отдельный маршрут – это частичная реализация ФС, которая возникает с определенной вероятностью. Длина маршрута зависит от количества повторений отдельных его участков. Число повторений (циклов) во многих ФС случайно. В целом ФС следует рассматривать как сложную систему со случайной структурой.

Для расчета безошибочности работы ФС предварительно найдем её среднюю структуру, которая по множеству маршрутов и реализаций является математическим ожиданием случайной структуры. Такой подход принят в задачах оценки сложности алгоритма и расчета времени его реализации на вычислительных средствах. Речь идет в первую очередь об операциях перехода от структурной схемы к графу ФС и о сокращении числа вершин в графе. После выполнения указанных операций следует привести граф к ациклическому виду и найти все маршруты в графе. Эти операции разработаны для частных типов циклов и ветвлений. В данной работе покажем возможность их выполнения для произвольных структур графов ФС.

Рассмотрим содержание операций определения средней структуры ФС в порядке их выполнения.

Переход от структурной схемы к графу

По структурной схеме ФС строится конечный взвешенно-ориентированный граф $G(V, E)$, который состоит из конечного множества вершин $V = \{\nu_1, \nu_2, ..., \nu_n\}$, соответствующих элементам схемы (операционным и решающим), и конечного множества дуг $E = \{e_1, e_2, ..., e_F\}$, соответствующих исходным связям между элементами. Дугам графа приписываются веса $\omega_{ij}(i, j = 1, ..., n)$. Вершинам графа приписываются значения безошибочности однократной работы $P_i(i = 1, ..., n)$ ФЧ или ФС подчиненного уровня иерархии.

В целом граф ФС должен удовлетворять следующим условиям:

а) в графе есть два типа отмеченных вершин: входные (соответствуют операциям подготовки исходных данных для данной ФС, из них выходит только одна дуга) и выходные (соответствуют операциям перехода к другим ФС), – из этих вершин не выходит ни одной дуги;

б) из вершины, соответствующей операционному элементу структурной схемы, исходит только одна дуга ($\omega_{ij} = 1$), из вер-

шины, соответствующей решающему элементу – f дуг, так как каждое решение может иметь f исходов; каждый исход – это случайное событие. Вероятность его наступления обозначим как p_{ij};

в) на дугах, выходящих из решающих вершин и означающих переход к циклическому повторению некоторого участка ФС, вес дуги равен сумме p_{ij} и K_C, где K_C – число циклов, которое может быть выбрано из некоторого диапазона – от $K_{C\min}$ до $K_{C\max}$. Максимальное количество циклов должно быть конечным, а информационные факторы или условия выхода из цикла ограничивают количество циклов снизу.

На рис. II.3.6 приведен пример взвешенно-ориентированного графа ФС.

Граф $G(V, E)$ задается взвешенной матрицей смежности $W = (\omega_{ij})$ размерности $(V * V)$. Веса несуществующих дуг при-

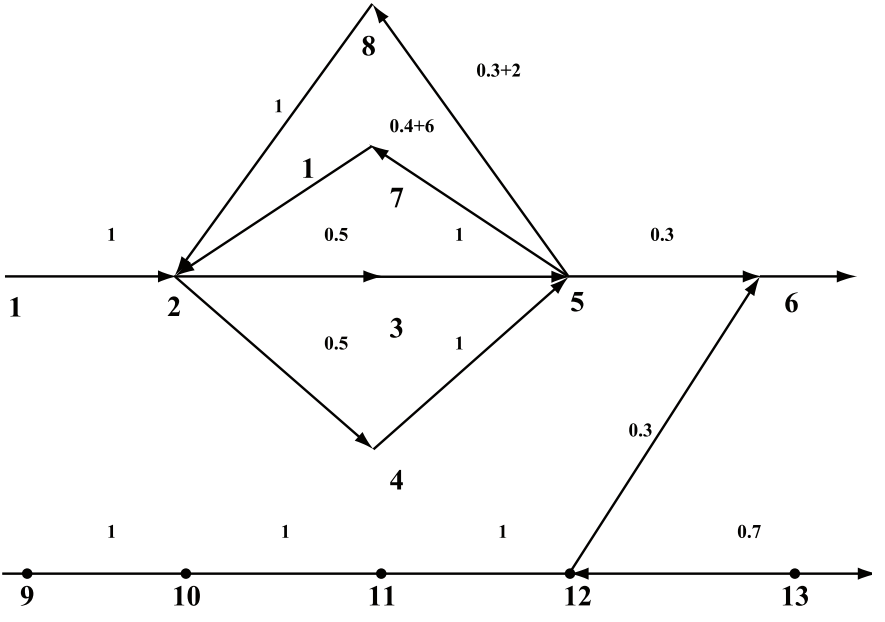


Рис. II.3.6. Пример взвешенного ориентированного графа ФС

нимаются равными 0. Фрагмент взвешенной матрицы смежности для графа, показанного на рис. II.3.6, имеет следующий вид:

$$W = \begin{array}{c|cccccccc} & \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} & \mathbf{6} & \mathbf{7} & \mathbf{8} \\ \hline \mathbf{1} & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{2} & 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 \\ \mathbf{3} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \mathbf{4} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \mathbf{5} & 0 & 0 & 0 & 0 & 0 & 0.3 & 6.4 & 2.3 \\ \mathbf{6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{7} & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{8} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

В этой матрице учтены вершины 1÷8 графа, а также связи между ними. Дугам 5, 7 и 5, 8 приписаны суммы вероятностей переходов и среднего числа циклов.

Сокращение числа вершин в графе

Операция сокращения числа вершин заключается в замене эквивалентной вершиной одного линейного или нескольких линейных параллельных участков графа из простых вершин. Вершина ν_i простая, если имеет одного предшественника и одного потомка, т.е. только одну входную и только одну выходную дугу. Под линейным участком графа понимается участок из *k* простых вершин, которые связаны между собой дугами с весом $\omega_{ij} = 1$, за исключением начальной и конечной вершины линейного участка, которые могут быть непростыми. В примере, приведенном на рис. II.3.6, в линейный участок входят вершины 11-12-13.

Формально для объединения r последовательно расположенных вершин ($r \in n$) необходимо выполнение следующих условий:

$$\omega(l+i; l+i+1) = 1; i = 1, 2, ..., r;$$

$$\omega(d; l+i) = 0;$$

$$l, d, r \in n; d \notin r,$$

где n – множество вершин исходного графа.

Условие $\omega(l+i; l+i+1) = 1$ означает, что в любую вершину из множества объединяемых вершин входит только одна дуга. Условие $\omega(d, l+i) = 0$ при $d \notin r$ означает, что в любую вершину из числа объединяемых не должно входить каких-либо дуг из остальных вершин исходного графа. Вероятность однократного правильного выполнения операций эквивалентной вершины рассчитывается по формуле

$$P_{ЭКВ} = \prod_{i=1}^{k} P_i;$$

где P_i – вероятность правильного однократного выполнения операций i-ой вершины, $к$ – количество вершин в линейном участке.

Линейные участки параллельны, если они связаны с одной и той же парой вершин, одна из которых i является предшественником этих вершин, а другая их потомком. Замена таких вершин эквивалентной вершиной сокращает число циклов, если данный участок входит в цикл. В приведенном примере на рис. II.3.6 параллельными вершинами являются 6, 7, 8, вершиной i – 5, вершиной a – 9.

Вероятность правильного выполнения операций эквивалентной вершины рассчитывается по формуле

$$P_{ЭКВ} = \sum_{j=1}^{r} p_{ij} p_{ja} \prod_{i=1}^{k} P_i ,$$

где r – количество параллельных линейных участков. Если $r = 1$, то $p_{ij} = 1$. Поскольку параллельные j-е вершины простые, то $p_{ja} = 1$ и приведенная формула упрощается к виду:

$$P_{ЭКВ} = \sum_{j=1}^{r} p_{ij} \prod_{i=1}^{k} P_i .$$

Приведение графа к ациклическому виду

Циклами в ФС являются такие ее участки, которые многократно используются в процессе однократной работы ФС. На графе эти участки обозначаются ориентированными контурами.

Введем ряд определений:

1. Каждый ориентированный контур содержит не менее двух непростых вершин.

2. Если пара непростых вершин контура такая, что первая вершина имеет две входные и одну выходную дуги, а вторая – одну входную и две выходных дуги, то первая вершина предшественник, а вторая – потомок контура. Такой контур назовем конечным.

3. Если вершина – предшественник контура содержит более одного выхода или более двух входов, а вершина – потомок контура более одного входа или более двух выходов, то данный контур может быть разложен на два и более конечных контуров.

4. Вершина-предшественник называется корневой вершиной контура.

5. Если в контуре содержится более двух непростых вершин, то в качестве корневой принимается вершина, содержащая наибольшее число входов, а в качестве вершины-потомка – та, которая содержит наибольшее число выходов.

6. Если в контуре содержится более двух непростых вершин, то данный контур может быть разложен на несколько конечных контуров либо на один или несколько конечных контуров и на один или несколько путей от контура к одному или нескольким выходам вычислительного процесса.

В основу поиска контура в орграфе положена стандартная процедура "поиск с возвращением" (*backtracking*) [21], т.е. контуры порождаются в процессе поиска в глубину, в котором дуги добавляются к пути до тех пор, пока не получится контур. Для того чтобы гарантировать, что каждый порожденный контур будет новым и не будет получаться просто в результате перестановки дуг в полученном ранее контуре, полагаем, что каждый контур имеет корень в его наименьшей вершине. Поиск начинается с вершины z, имеющей более одного входа ($z=v_1$) и строится путь $v_1, v_2, \ldots, v_k$, такой, что $\nu_i > z, 1 \le i \le k$. Контур считается построенным только тогда, когда следующая вершина ν_{k+1} равна z. После порождения контура $v_1, v_2, \ldots, v_k, v_1$ анализируется следующая дуга, выходящая из вершины v_k. Если все дуги, выходящие из вершины v_k, уже проанализированы, то происходит возвращение к предыдущей вершине v_{k-1} и анализируются исходящие из нее пути и т.д. Этот процесс продолжается до тех пор, пока будет установлено, что осуществляется попытка вернуться на начальную вершину, когда все контуры, содержащие вершину z, построены. Процедура повторяется для $z = 1, 2, \ldots, \nu$ вершин.

Расчет вероятности правильной однократной работы циклического участка производится по следующей формуле:

$$P_C = p_{v_k v_1} \prod_{i=1}^{V_k - 1} p_{v_i v_{i+1}} (\prod_{i=1}^{V_k} P_i)^{K_C},$$

где V_k – количество вершин в циклическом участке; K_c – количество повторений цикла; $p_{v_k v_1}$ – вероятность перехода из конечной вершины циклического участка в его корневую вершину.

Расчет вероятности правильной однократной работы функциональной структуры

После сокращения линейных участков и параллельных вершин в графе и приведения графа к ациклическому виду нужно построить все маршруты от входной вершины графа к выходной. На данном этапе также применена процедура поиска с возвращением.

Вероятность однократной правильной работы ФС в целом определяется в виде:

$$P_{ФС} = \sum_{n=1}^{N} P_{ФСn},$$

где N – число входов ФС, $n = 1, 2, ..., N$, $P_{ФСn}$ – вероятность правильного однократного выполнения *n-го* маршрута от входной вершины к выходной.

Изложенные методика и алгоритм ориентированы на расчет вероятностей правильной однократной работы ФС на уровне машинных команд и макрокоманд. Расчет показателя $P_{ФС}$ правильности работы ФС для цифровых систем управления и обработки информации, в программах которых содержатся тысячи и

десятки тысяч строк, целесообразно производить эксперимента льно-расчетным путем, основываясь на устойчивости процентного содержания различных машинных команд в исполняемой программе (см. п.II.5.4 применительно к критически важным информационным системам).

Контрольные вопросы

1. Какие и каким образом возмущающие воздействия приводят к нарушению логических условий в результатах срабатывания цифрового логического элемента (вентиля)?

2. Опишите методику определения показателей функциональной надежности вентиля.

3. Опишите методику расчета правильности работы логических элементов.

4. Как зависит вероятность правильной работы логического элемента ИЛИ и/или элемента И от содержания сигналов 1 и 0 во входной информации?

5. Как влияет изменение содержания сигналов 1 и 0 во входной информации на функциональную надежность дешифраторов?

6. Приведите методику расчета правильности работы цифровых устройств.

7. Поясните принцип построения алгоритма расчета правильности работы цифровых устройств.

8. Как зависит правильность работы цифрового устройства от количества операций при последовательной обработке информации?

9. При какой организации обработки информации выше уровень функциональной надежности цифрового устройства: при последовательной или при параллельной обработке информации?

10. Поясните порядок расчета надежности функциональных структур.

Глава II.4. Функциональная надежность программного обеспечения

II.4.1. Классификация программных средств

II.4.1.1. Системное (базовое) программное обеспечение

Программное и аппаратное обеспечение в информационной системе работают в неразрывной связи и взаимодействии. Уровни программного обеспечения представляют собой пирамиду, где каждый высший уровень базируется на программном обеспечении предшествующих уровней [22].

Базовый уровень – низший уровень программного обеспечения, который содержится в составе базового аппаратного обеспечения системы**.** Базовое программное обеспечение содержится в составе базового аппаратного обеспечения и сохраняется в специальных микросхемах постоянного запоминающего устройства (ПЗУ), образуя базовую систему ввода-вывода BIOS. Программы и данные записываются в ПЗУ на этапе производства и не могут быть изменены во время эксплуатации.

Программы *системного уровня* обеспечивают взаимодействие других программ системы с программами базового уровня. От программ этого уровня зависят эксплуатационные показатели всей информационной системы. Другой класс программ системного уровня отвечает за взаимодействие с пользователем. С их помощью можно вводить данные в информационную систему, руководить ее работой и получать результат в удобной форме. Это средства обеспечения пользовательско-

го интерфейса, от них зависит удобство и производительность работы системы.

Совокупность программного обеспечения системного уровня образует ядро операционной системы компьютера. Наличие ядра операционной системы – это необходимое условие для возможности практической работы информационной системы. Ядро операционной системы выполняет такие функции, как: управление памятью, управление процессами ввода-вывода, файловой системой, осуществление организации взаимодействия и диспетчеризации процессов, учет использования ресурсов, обработка команд и т.д.

Служебный уровень программного обеспечения предназначен для автоматизации работ по проверке и настройке информационной системы, а также для улучшения функций системных программ. В состав служебных программ входят: диспетчеры файлов (файловые менеджеры), средства сжатия данных (архиваторы), средства диагностики, программы инсталляции (установки), средства компьютерной безопасности и др.

Совокупность базового, системного и служебного уровней программных средств принято называть *системным (базовым) программным обеспечением.*

II.4.1.2. Прикладное программное обеспечение

Это программное обеспечение содержит комплекс прикладных программ, с помощью которых решаются производственные, научно – технические задачи, а также образовательные и развлекательные задачи. Первые две группы задач имеют непосредственное отношение к теории и практике функциональной надежности информационных систем. Между прикладным и системным программным обеспечением существует тесная взаимосвязь. Функциональные возможности информационной системы и доступность прикладных программ непосредственно

зависят от типа имеющейся операционной системы, системных средств, помещенных в ее ядро, а также от взаимодействий составляющих комплекса человек-программа-оборудование.

Прикладное программное обеспечение (ПО) можно сгруппировать по функциональным признакам в несколько больших групп программ (рис. II.4.1).

К **производственным программам** относятся:

– программы автоматизации обработки текстовых данных и графических изображений;

– программы автоматизации управленческой деятельности;

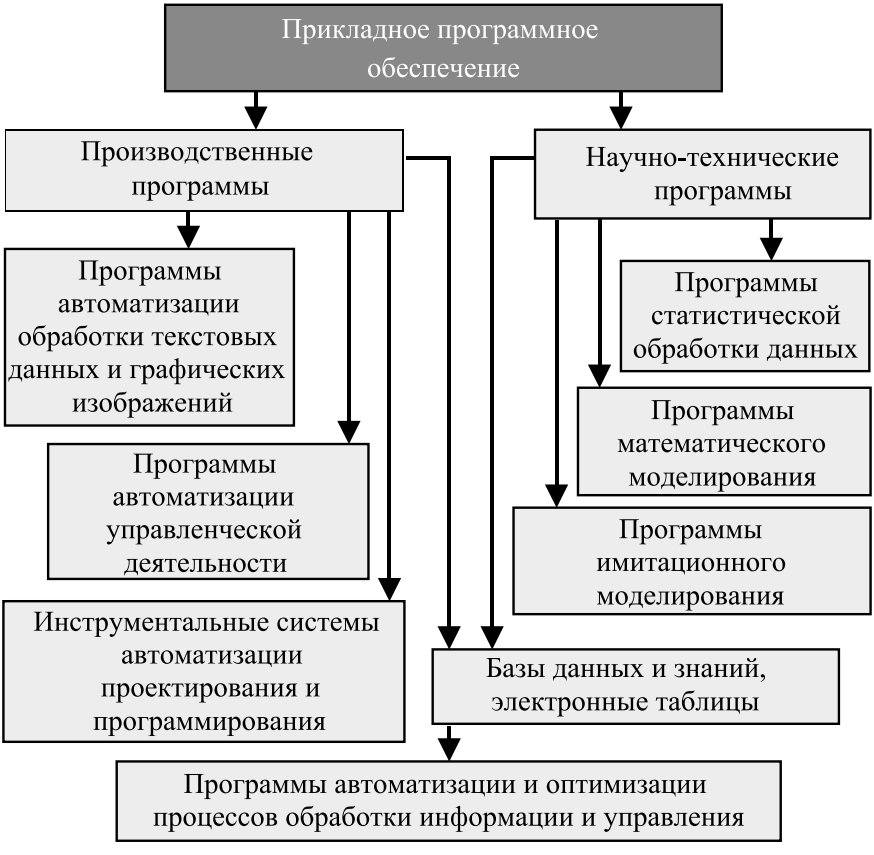


Рис. II.4.1. Классификация прикладного программного обеспечения

– инструментальные системы автоматизации проектирования и программирования;

– программы хранения и обработки данных и знаний, поддержки принятия решений;

– программы автоматизации и оптимизации процессов обработки информации и управления и др.

К классу **научно-технических программ** относятся**:**

– программы статистической обработки данных;

– программы математического моделирования;

– программы имитационного моделирования;

– программы хранения и обработки данных и знаний, поддержки принятия решений;

– программы автоматизации и оптимизации процессов обработки информации и управления и др.

Важно осознавать, что между производственными и научно-техническими программами существует тесная взаимосвязь – производственные программы создаются и развиваются в результате статистической обработки накопленных практических данных, математического и имитационного моделирования. В свою очередь, научно – технические программы создаются с помощью инструментальных средств и систем программирования, а также программ автоматизации обработки текстовых данных и графических изображений.

Группа *программ автоматизации обработки текстовых данных и графических изображений* охватывает достаточно широкий спектр прикладных программ. К ним относятся *текстовые редакторы, текстовые процессоры (в том числе растровые редакторы; векторные редакторы; 3-D редакторы), редакторы HTML* (Web-редакторы), *браузеры* (средства просмотра Web-документов), *настольные издательские системы***.**

Группа ***программ автоматизации управленческой деятельности*** включает в себя *интегрированные системы делопроиз-*

водства – программные средства для автоматизации рабочего места руководителя, *системы автоматизированного перевода, информационно-правовые системы, бухгалтерские системы, финансовые аналитические системы* и др.

К ***инструментальным системам автоматизации проектирования и программирования*** относятся главным образом *системы автоматизированного проектирования* (CAD-системы) и *инструментальные языки и системы программирования*, которые служат для разработки новых программ. В современную систему программирования кроме программы формализации естественного языка и транслятора входит текстовый редактор, программа автоматизации документирования, компоновщик, библиотека стандартных программ, отладчик, визуальные средства автоматизации программирования.

К *прикладным программам хранения и обработки данных и знаний, поддержки принятия решений* относятся *базы данных, системы управления базами данных (СУБД), электронные таблицы, геоинформационные системы (ГИС), экспертные системы.*

Группа *программ автоматизации и оптимизации процессов обработки информации и управления* содержит широкий спектр прикладных программ, ориентированных на конкретные предметные области: *программы преобразования и передачи информации*, *программы оптимальной фильтрации, программы управления процессами и / или объектами, логистические программы, программы оперативного анализа параметров процессов, программы прогнозирования параметров процессов* и многие другие.

Группа *программ статистической обработки данных* предназначена для решения большого числа различных научно – практических задач, связанных *со статистической оценкой параметров случайных процессов, с оценками показа-*

телей качества, надежности и технической эффективности объектов и систем, с выявлением закономерностей в случайных величинах и в случайных процессах, со статистической оценкой погрешностей в результатах обработки информации и управления и др.

Группа *программ математического и имитационного моделирования* предназначена для построения и решения *математических моделей (*в том числе *Марковских и полумарковских, логико-вероятностных, моделей, построенных с помощью теории нечетких множеств или интервальных средних* и многих др.), *имитационных моделей сетей массового обслуживания, сетей связи, функционирования сложных систем* и т.д.

II.4.1.3. Программы встроенных систем

Встроенные системы являются неотъемлемой частью информационно-управляющих систем (ИУС). Они в большинстве своем содержатся в объектах управления, в подсистемах ИУС, непосредственно управляющих объектами управления. Встроенные системы самостоятельно используются в бытовой технике (стиральных машинах, телевизорах, холодильниках и др.).

На практике используется большой класс бортовых информационно – управляющих систем, осуществляющих управление движущимися объектами и системами (автомобилями, локомотивами, самолетами, морскими суднами, ракетами и т.д.). В подавляющем большинстве случаев программное обеспечение встраиваемой системы нельзя рассматривать в отрыве от аппаратного обеспечения, конструкции и особенностей окружения. Необходимо понимать, что проектируется не часть системы, а система целиком. Подходы, используемые современными программистами при создании больших программных систем общего назначения в мире встраиваемых систем, как правило, не пригодны или пригодны с большими ограничениями.

Главной отличительной особенностью встроенной системы является наличие в ней контроллерной сети или хотя бы одного контроллера. Контроллер (англ. *controller* – регулятор, управляющее устройство) представляет собой достаточно сложную компьютерную систему. Обычно контроллер строится с помощью микропроцессорной техники. Контроллер состоит из двух основных частей: ядра и модуля ввода-вывода. Ядро микропроцессорного контроллера составляют микропроцессор, системный контроллер (СК) и устройства памяти. Для хранения программ и данных ядро микроконтроллера содержит ОЗУ (оперативное запоминающее устройство), ПЗУ (постоянное ЗУ) и РПЗУ (репрограммируемое ЗУ). ПЗУ используется только для хранения программ управления. Эти программы, разработанные и отлаженные на специальных средствах отладки, заносятся в ПЗУ в заводских условиях, и пользователь изменять их не может. РПЗУ отличается тем, что пользователь может изменять его содержание, т.е. позволяет производить *программирование микроконтроллеров*. ОЗУ используется для хранения данных, необходимых для выполнения основной программы управления. Обращение к ячейкам памяти адресное. Любой алгоритм управления микроконтроллерной системой реализуется управляющей программой, которая представляет собой цифровые двоичные коды, размещенные в ячейках ПЗУ.

Особенностью управляющих контроллеров является то, что в его состав не входят средства отладки программ, так как основной набор программных модулей, составляющих библиотеку программ контроллера, заносится в его память в заводских условиях и изменению не подлежит. Пользователь имеет только возможность из имеющегося набора программных модулей составить конфигурацию контура направления.

Возрастающая степень интеграции цифровых микросхем определила появление в настоящее время промышленных мик-

роконтроллеров, реализованных на одном кристалле. На кристалле такого контроллера, кроме микропроцессора, находятся модуль памяти, интерфейсные схемы и даже таймер. По сути такие контроллеры – это однокристальные ЭВМ малой производительности. Система команд однокристальных контроллеров позволяет организовать сложную управляющую систему с большим количеством внутрипрограммных ветвлений в соответствии с целью управления и состоянием первичных преобразователей. Существующая возможность их перепрограммирования, а также малые габариты создают предпосылки для создания компактных встраиваемых в оборудование информационно-управляющих систем.

Программы встроенных систем – это класс прикладных программ, созданных с помощью программно – аппаратных средств. Программа создается на *языке функциональных диаграмм* (FBD) – стандартизированном языке программирования для промышленных контроллеров. На экране компьютера формируется графическое описание работы контроллера. В квадратиках выписываются формулы, логика переходов. С левого края обозначаются входные сигналы. С правой стороны графической схемы обозначаются управляющие действия. Программа представлена не в виде алгоритма, а в виде принципиальной схемы: таймеры, счетчики и т.п. Поэтому получается очень наглядное изображение процесса функционирования программы. К тому же отладка программы производится в реальном времени при работающем контроллере, когда подсвечиваются работающие блоки программы, переменные, входы-выходы. Заливка кода в контроллер осуществляется без останова контроллера.

II.4.1.4. Общие соображения

Приведенный краткий обзор системного и прикладного программного обеспечения для информационных систем и специ-

фических встроенных систем позволяет высказать некоторые соображения по поводу стратегии и тактики расчета, прогнозирования и оценивания функциональной надежности программных средств.

– Функциональная надежность программного обеспечения не может моделироваться с помощью детерминированных моделей. Это принципиально важно, поскольку несмотря на систематический характер ошибок в тексте программы, их проявление в процессе исполнения программы носит случайный характер вследствие случайного характера набора входных данных, случайного выбора ветви программы, случайного характера заявок на выполнение программы и др.

– Для исследования функциональной надежности как системного, так и прикладного программного обеспечения, включая программы встроенных систем, наряду с вероятностными моделями перспективно применение моделей, построенных с помощью математических методов нечетких множеств и интервальных средних. Последние типы моделей особенно перспективны при неполной и /или недостоверной информации.

– Стратегия испытаний всех видов программных средств состоит в том, чтобы подавляющий объем испытаний проводился автономно от инструментальной среды. Это особенно характерно для процессов верификации отдельных программ и программных модулей. На завершающем (интеграционном) этапе должна проводиться валидация программно – аппаратных комплексов.

– Наиболее реальный метод испытаний функциональной надежности программных средств встроенных систем – метод ускоренных натурных испытаний на основе внесения искажений в промежуточные результаты выполнения информационных процессов и внесения помех на входы составных цифровых устройств в реальном масштабе времени при условии доказательства подобия результатов ускоренных испытаний практическим данным.

– Для повышения эффективности оценивания функциональной надежности программных средств следует сочетать результаты их испытаний с расширенным аудитом выполнения в проверяемых программных средствах требований соответствующих стандартов в отношении архитектуры этих программных средств, использованных языков программирования, способов обнаружения ошибок и диагностирования, примененных методов верификации и валидации и др.

II.4.2. Определение качества и функциональной надежности программного обеспечения

Основу инженерных методов в программировании составляет повышение качества, для достижения которого сформировались методы определения требований к качеству, подходы к выбору и усовершенствованию моделей метрического анализа показателей качества, методы количественного измерения показателей качества на этапах жизненного цикла.

Качество ПО – это совокупность свойств, определяющих полезность изделия (программы) для пользователей в соответствии с функциональным назначением и предъявленными требованиями. При этом требования могут трактоваться довольно широко, что порождает целый ряд независимых определений понятия. Чаще всего используется определение ISO 9001, согласно которому качество есть «степень соответствия присущих характеристик требованиям». Качество ПО – это относительное понятие, которое имеет смысл только при учете реальных условий его применения, поэтому требования, предъявляемые к качеству, ставятся в соответствии с условиями и конкретной областью их применения.

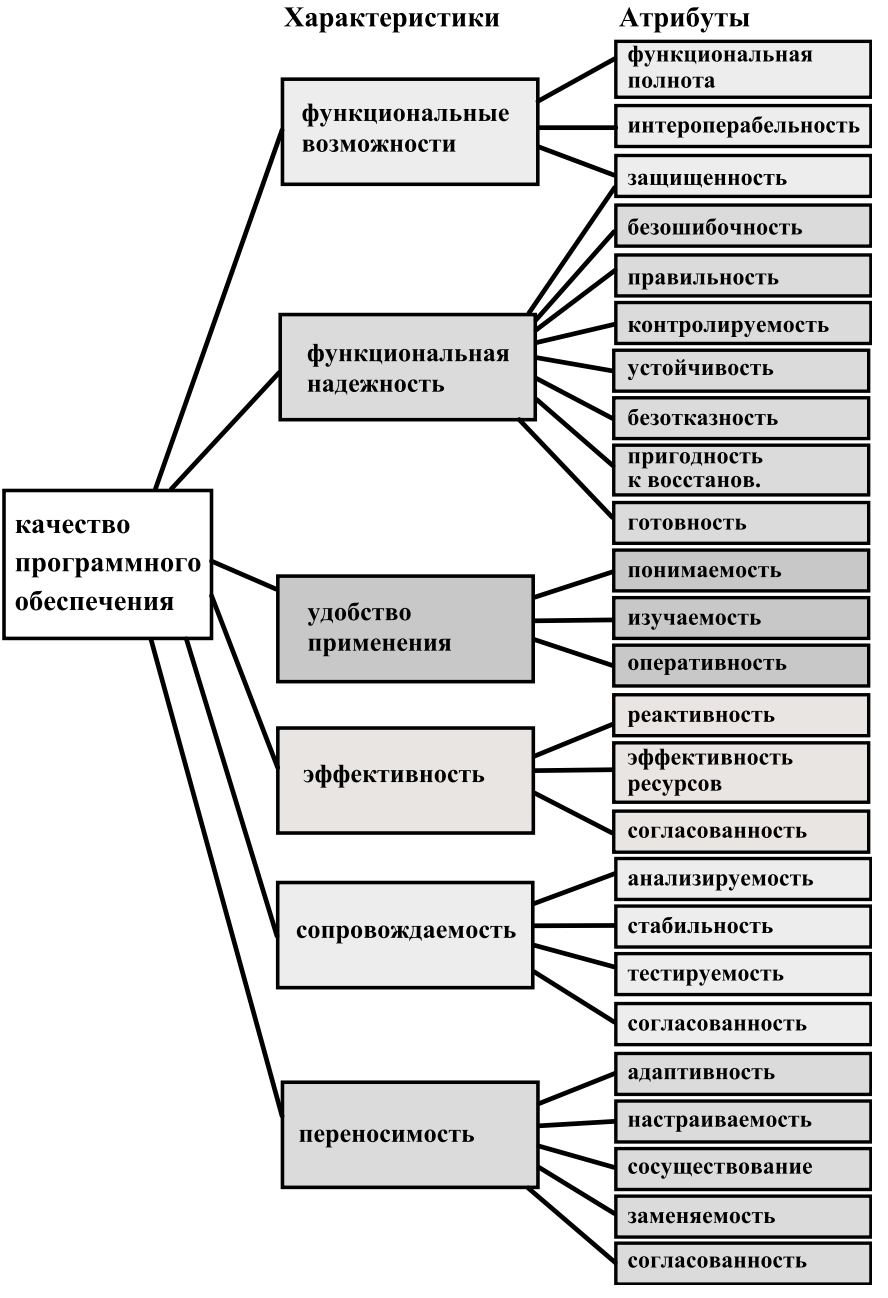


Рис. II.4.2. Характеристики и атрибуты качества программного обеспечения

Модель качества ПО имеет следующие четыре уровня представления.

1. Первый уровень соответствует определению характеристик (показателей) качества ПО, каждая из которых отражает отдельную точку зрения пользователя на качество. Согласно стандартам [23-26, 28-30] в модель качества входит шесть характеристик или шесть основных показателей качеств, которые перечислим в порядке их значимости для большинства пользователей:

1. функциональные возможности;
2. функциональная надежность;
3. удобство применения;
4. эффективность;
5. сопровождаемость;
6. переносимость.

2. Второму уровню соответствуют атрибуты для каждой характеристики качества, которые детализируют разные аспекты конкретной характеристики. Набор атрибутов характеристик качества используется при оценке качества.

Рассмотрим наборы атрибутов для каждого из перечисленных показателей качества (рис. II.4.2).

Функциональные возможности – совокупность свойств, определяющих способность программного обеспечения выполнять предусмотренные функции в заданной среде в соответствии с заданными требованиями. Под *функцией* понимается некоторая упорядоченная последовательность действий для удовлетворения потребительских свойств. Функции бывают целевые (основные) и вспомогательные.

К атрибутам функциональных возможностей относятся:

- *функциональная полнота* – атрибут, который показывает меру достаточности основных функций для решения задач в соответствии с назначением данного ПО;

- *интероперабельность* – атрибут, который показывает возможность взаимодействия данного ПО со специальными системами и средами (операционные системы, вычислительные сети);

- *защищенность* – атрибут, который показывает способность ПО предотвращать несанкционированный доступ (случайный или умышленный) к программам и данным.

Функциональная надежность ПО – это основной предмет обсуждения в данной главе. Поэтому остановимся более подробно на определении этой характеристики качества программ и описании ее атрибутов. Прежде всего, мы исходим из того очевидного факта, что каким – либо одним свойством невозможно описать функциональную надежность. Только совокупность свойств (атрибутов) позволяет выполнять предназначение программ – преобразовывать исходные данные в промежуточные и выходные результаты.

В настоящее время не существует общепринятого строгого понятия надежности программного обеспечения (ПО), и, естественно, понятия функциональной надежности ПО. В сути своей оба эти понятия относятся к одному и тому же предмету – определению одной из наиболее важных характеристик качества функционирования ПО. В различных публикациях определения надежности ПО существенно различаются. Ряд авторов механически переносят понятия, присущие структурной надежности объектов, на надежность ПО. Они рассматривают ПО как неотъемлемую часть цифровой техники, на которой оно реализуется. Так, например, в работе [31] надежность ПО рассматривается «как его свойство сохранять во времени в установленных пределах надежность цифрового вычислительного комплекса, обеспечивающую его функционирование в заданных режимах, условиях применения и технического обслуживания». В трактовке ряда зарубежных авторов [32, 33 и др.]

«надежность системы ПО – вероятность того, что отказ ПО, вызывающий отклонение получаемого выхода от требуемого за допустимые пределы не произойдет при определенных условиях внешней среды в течение заданного периода наблюдения». Это определение, за исключением почему – то типичной для зарубежных авторов понятийной некорректности (надежность – не вероятность, но свойство), близко к определению структурной надежности [34]. В обоих приведенных определениях есть сомнительные условия, которые не определяют надежность ПО. Прежде всего, учет заданного периода наблюдения не характерен для ПО. Время для ПО не имеет такого фатального значения, как для аппаратных средств. Можно показать также, что функциональная надежность ПО существенно зависит не от времени, а от частоты запросов на использование программных средств, т.е. на выполнение информационных процессов (см. главу II.2).

Другое сомнительное условие приведенных выше и аналогичных им определений надежности ПО – это то, что надежность ПО рассматривается с позиции составной части цифровой техники. Такой подход исторически связан с тем, что на ранних стадиях развития информационных систем программное обеспечение было недостаточно функционально и решало в составе цифровой техники узкий класс задач. В процессе эволюции в составе информационных систем возникли сначала аппаратно-программные комплексы, а затем по мере интенсивного развития ПО роль его в информационных системах резко возросла и теперь говорят о программно – аппаратных комплексах. Цифровая техника служит инструментальной средой для ПО. Программное обеспечение в информационных системах имеет определяющее значение и оценивается самостоятельно.

С учетом приведенных рассуждений можно дать следующее определение функциональной надежности ПО.

Функциональная надежность – совокупность свойств, которые определяют способность программного обеспечения с приемлемым уровнем безошибочности правильно преобразовывать исходные данные в результаты при данных условиях, сохраняя выходные результаты в допустимых пределах.

Некоторые положения приведенного определения требуют разъяснения. Прежде всего, нуждается в разъяснении сочетание атрибутов «безошибочность» и «правильность», которые имеют разное смысловое наполнение. Имеется в виду следующее. Если корректно построен алгоритм преобразования, то при отсутствии программных или сбойных ошибок в процессе преобразования возможно получить правильные результаты. Мы говорим «возможно получить», имея в виду, что достижение правильности результатов не исчерпывается только корректно построенным алгоритмом преобразования. Неправильные результаты могут быть получены вследствие нарушений граничных значений данных и/или результатов, граничных значений длин ключей, допустимого числа записей файлов, допустимой длины ключей, допустимого числа критериев поиска и др. Систематические ошибки, вызывающие нарушение правильности преобразования исходных данных в результаты, возникают не только вследствие алгоритмических ошибок или ошибок проектирования граничных значений, но и вследствие того, что сбойные ошибки цифровой техники, в свою очередь, приводят к искажению отдельных предписаний алгоритма или некоторых хранимых граничных значений. Таким образом, свойство *правильности* является одной из важных составляющих *функциональной надежности* программного обеспечения и характеризует достоверность результатов (кодовые комбинации результатов относятся к числу разрешенных).

Определение функциональной надежности ПО было бы неполным без учета условий возможной трансформации ошибки в выходных результатах в функциональный отказ информацион-

ной системы. Надо понимать, что с помощью предусмотренных в программном обеспечении функций выходные результаты сохраняются в границах допусков по точности и своевременности управления. Сохранение выходных результатов в допустимых пределах достигается при: а) наличии в составе ПО качественных средств оперативного обнаружения ошибок и б) наличии в составе ПО средств обеспечения устойчивости к ошибкам. Заметим, что при наличии некоторого, пусть даже незначительного резерва времени, всегда имеется некоторый шанс оперативно устранить обнаруженную ошибку даже при отсутствии средств обеспечения устойчивости к ошибкам. Однако наличие этих средств существенно упрощает задачу, позволяет значительно повысить шансы сохранения правильных выходных результатов и, конечно, позволяет повысить безошибочность процесса преобразования исходных данных в результаты.

К атрибутам **функциональной надежности** ПО относятся:

• *безошибочность* – атрибут, который определяет способность ПО функционировать без ошибок;

• *правильность* – атрибут, который характеризует возможность достижения правильных результатов при функционировании ПО;

• *защищенность* – атрибут, который показывает способность ПО предотвращать несанкционированный доступ (случайный или умышленный) к программам и данным. Этот атрибут характеризует способность ПО функционировать без ошибок в условиях информационных атак;

• *контролируемость* – атрибут, который характеризует полноту и эффективность обнаружения ошибок в промежуточных и выходных результатах;

• *устойчивость к ошибкам* – атрибут, который характеризует способность ПО правильно выполнять функции при аномальных условиях (сбой аппаратуры, ошибки в данных и интерфейсах и др.);

• *безотказность* – атрибут, который характеризует способность ПО не вызывать функциональные отказы информационной системы;

• *пригодность к восстановлению* – атрибут, который характеризует способность программы к устранению программной ошибки и к перезапуску для повторного выполнения и восстановления данных в случае функционального отказа;

• *готовность* – атрибут, который показывает способность программы по произвольной заявке безошибочно выполнить предусмотренное преобразование.

Удобство применения характеризуется множеством атрибутов, которые указывают на необходимые и пригодные условия использования ПО заданным кругом пользователей для получения соответствующих результатов. В стандарте [25] удобство применения определено как специфическое множество атрибутов программного продукта, характеризующих его эргономичность.

К атрибутам удобства применения относятся:

• *понимаемость* – атрибут, который определяет усилия, затрачиваемые на распознавание логических концепций и условий применения ПО;

• *изучаемость* (легкость изучения) – атрибут, который определяет усилия пользователей на определение применимости ПО путем использования операционного контроля, диагностики, а также процедур, правил и документации;

• *оперативность* – атрибут, который показывает на реакцию системы при выполнении операций и операционного контроля.

Эффективность – совокупность атрибутов, которые определяют взаимосвязь уровней выполнения ПО, использования ресурсов (средства, аппаратура, материалы – бумага для печатающего устройства и др.) и услуг, выполняемых штатным обслуживающим персоналом и др.

К атрибутам эффективности ПО относятся:

• *реактивность* – атрибут, который показывает время отклика, обработки и выполнения функций;

• *эффективность ресурсов* – атрибут, показывающий количество и продолжительность используемых ресурсов при выполнении функций ПО;

• *согласованность* – атрибут, который показывает соответствие данной характеристики с заданными стандартами, правилами и предписаниями.

Сопровождаемость – совокупность свойств, которые характеризуют усилия, которые надо затратить на проведение модификаций, включающих корректировку, усовершенствование и адаптацию ПО при изменении среды, требований или функциональных спецификаций.

Сопровождаемость включает атрибуты:

• *анализируемость* – атрибут, определяющий необходимые усилия для диагностики отказов или идентификации частей, которые будут модифицироваться;

• *изменяемость* – атрибут, который определяет удаление ошибок в ПО или внесение изменений для их устранения, а также введение новых возможностей в ПО или в среду функционирования;

• *стабильность* – атрибут, указывающий на постоянство структуры и риск ее модификации;

• *тестируемость* – атрибут, указывающий на усилия при проведении валидации и верификации с целью обнаружения несоответствий требованиям, а также на необходимость проведения модификации ПО и сертификации;

• *согласованность* – атрибут, который показывает соответствие данной характеристики заданным стандартам, правилам и предписаниями качества ПО на всех этапах жизненного цикла.

Переносимость – множество показателей, указывающих на способность ПО адаптироваться к работе в новых условиях среды выполнения. Среда может быть организационной, аппаратной и программной. Поэтому перенос ПО в новую среду выполнения может быть связан с совокупностью действий, направленных на обеспечение его функционирования в среде, отличной от той среды, в которой оно создавалось с учетом новых программных, организационных и технических возможностей.

Переносимость включает атрибуты:

• *адаптивность* – атрибут, определяющий усилия, затрачиваемые на адаптацию к различным средам;

• *настраиваемость* (простота инсталляции) – атрибут, который определяет необходимые усилия для запуска данного ПО в специальной среде;

• *сосуществование* – атрибут, который определяет возможность использования специального ПО в среде действующей системы;

• *заменяемость* – атрибут, который характеризует возможность интероперабельности при совместной работе с другими программами с необходимой инсталляцией или адаптацией ПО;

• *согласованность* – атрибут, который указывает на соответствие стандартам или соглашениям по обеспечению переноса ПО из одной инструментальной среды в другую.

3. Третий уровень предназначен для измерения качества с помощью метрик, каждая из них согласно стандартам [24-26, 28-30] определяется как комбинация метода измерения атрибута и шкалы измерения значений атрибутов. Для оценки атрибутов качества на этапах ЖЦ (при просмотре документации, программ и результатов тестирования программ) используются метрики с заданным оценочным весом для нивелирования результатов метрического анализа совокупных атрибутов конкретного по-

казателя и качества в целом. Атрибут качества определяется с помощью одной или нескольких методик оценки на этапах ЖЦ и на завершающем этапе разработки ПО.

4. Четвертый уровень – это оценочный элемент метрики (вес), который используется для оценки количественного или качественного значения отдельного атрибута показателя качества ПО. В зависимости от назначения, особенностей и условий сопровождения ПО, выбираются наиболее важные характеристики качества и их атрибуты (рис. II.4.2). Выбранные атрибуты и их приоритеты отражаются в требованиях на разработку систем, либо используются соответствующие приоритеты эталона класса ПО, к которому это ПО относится.

II.4.3. Метрики качества программного обеспечения

II.4.3.1. Классификация метрик качества программ

Для измерения характеристик качества используют метрики. *Метрика программного обеспечения* – это мера, позволяющая получить численное значение некоторого свойства программного обеспечения или его спецификаций. Поскольку количественные методы хорошо зарекомендовали себя в других областях, многие теоретики и практики информатики пытались перенести данный подход и в разработку программного обеспечения**.** *Метрика качества программ* – система измерений качества программ. Измерения характеристик можно выполнить объективно и достоверно. Однако не следует исключать того, что оценка качества ПО в целом может быть связана с субъективной интерпретацией получаемых оценок.

В зависимости от характеристик и особенностей применяемых метрик им ставятся в соответствие различные измерительные шкалы:

• *номинальной шкале* соответствуют метрики, классифицирующие программы на типы по признаку наличия или отсутствия некоторой характеристики без учета градаций;

• *порядковой шкале* соответствуют метрики, позволяющие ранжировать некоторые характеристики путем сравнения с опорными значениями, т.е. измерение по этой шкале фактически определяет взаимное положение конкретных программ;

• *интервальной шкале* соответствуют метрики, которые показывают не только относительное положение программ, но и то, как далеко они отстоят друг от друга;

• *относительной шкале* соответствуют метрики, позволяющие не только расположить программы определенным образом и оценить их положение относительно друг друга, но и определить, как далеко оценки отстоят от границы, начиная с которой характеристика может быть измерена.

Все метрики ПО разделяются на два класса:

• метрики, характеризующие наиболее специфические свойства программ, т.е. метрики оценки качества самого ПО;

• метрики оценки технических характеристик и факторов разработки программ, т.е. метрики оценки условий разработки программ.

Предмет обсуждения данного раздела – свойства программ. Условия и специфика разработки программ находятся за рамками обсуждаемых вопросов.

В настоящее время в мировой практике используется несколько сотен метрик программ. Существующие измерения качества программ можно сгруппировать по шести направлениям:

• измерения топологической и информационной сложности программ (производятся с помощью соответствующих метрик и представляют собой косвенные оценки надежности);

• оценки функциональной надежности программных систем, позволяющие прогнозировать проявление ошибок в программе (производятся непосредственно с помощью моделей надежности);

• измерения производительности ПО и оценки повышения его эффективности путем выявления ошибок проектирования;

• измерения уровня языковых средств и оценки их применения;

• измерения восприятия и понимания программных текстов, ориентированные на психологические факторы, существенные для сопровождения и модификации программ;

• измерения производительности труда программистов для прогнозирования сроков разработки программ и планирования работ по созданию программных комплексов.

II.4.3.2. Метрики сложности программ

Кратко рассмотрим метрики сложности. Уменьшение сложности ПО обеспечивает снижение трудоемкости проектирования, разработки, испытаний и сопровождения, обеспечивает простоту и надежность производимого ПО и информационной системы в целом. Целенаправленное снижение сложности ПО представляет собой многошаговую процедуру и требует предварительного исследования существующих показателей сложности, проведения их классификации и соотнесения с типами программ и их местоположением в жизненном цикле. В настоящее время многообразие показателей (в той или иной степени описывающих сложность программ) столь велико, что для их употребления требуется предварительное упорядочение. При оценке сложности программ, как правило, выделяют три основные группы метрик:

• метрики размера программ,

• метрики сложности потока управления программ,

• метрики сложности потока данных программ.

Метрики размера программ

Оценки первой группы наиболее просты и, очевидно, поэтому получили широкое распространение. Традиционной характеристикой размера программ является количество строк исходного текста. Под строкой понимается любой оператор программы, поскольку именно оператор, а не отдельно взятая строка является тем интеллектуальным "квантом" программы, опираясь на который можно строить метрики сложности ее создания. Непосредственное измерение размера программы, несмотря на свою простоту, дает хорошие результаты. Конечно, оценка размера программы недостаточна для принятия решения о ее сложности, но вполне применима для классификации программ, существенно различающихся объемами. При уменьшении различий в объеме программ на первый план выдвигаются оценки других факторов, оказывающих влияние на сложность. Таким образом, оценка размера программы есть оценка по номинальной шкале, на основе которой определяются только категории программ без уточнения оценки для каждой категории. Типичным представителем этой группы метрик является метрика Холстеда [27].

Метрика Холстеда

Основу метрики Холстеда составляют четыре измеряемых характеристики программы:

η_1 – число уникальных операторов программы, включая символы-разделители, имена процедур и знаки операций (словарь операторов);

η_2 – число уникальных операндов программы (словарь операндов);

N_1 – общее число операторов в программе;

N_2 – общее число операндов в программе.

Опираясь на эти характеристики, получаемые непосредственно при анализе исходных текстов программ, Холстед вводит следующие оценки:

– словарь программы $\eta = \eta_1 + \eta_2$, длина программы $N = N_1 + N_2$,

– объем программы $V = N \log_2 \eta$ (бит), где под битом подразумевается логическая единица информации – символ, оператор, операнд. Далее Холстед вводит η^* – теоретический словарь программы, т.е. словарный запас, необходимый для написания программы, с учетом того, что необходимая функция уже реализована в данном языке и, следовательно, программа сводится к вызову этой функции.

Используя длину η^*, Холстед вводит оценку $\hat{V} : \hat{V} = \eta^* \log_2 \eta^*$, с помощью которой описывается потенциальный объем программы, соответствующий максимально компактному тексту программы, реализующей данный алгоритм.

Метрики сложности потока управления программ

Вторая наиболее представительная группа оценок сложности программ – это метрики сложности потока управления программ [6, 22, 28, 35-39]. Типичными представителями этой группы являются метрики Мак – Кейба и их модификации, а также метрика Джилба. Как правило, с их помощью оперируют либо плотностью управляющих переходов внутри программ, либо взаимосвязями этих переходов. И в том и в другом случае стало традиционным представление программ в виде управляющего ориентированного графа *G=(V,E)*, где *V* – вершины, соответствующие операторам, а *E* – дуги, соответствующие переходам.

Метрика Мак – Кейба. Впервые графическое представление программ было предложено Мак-Кейбом. В его основе лежит идея оценки сложности ПО по числу базисных путей в управляющем графе, т.е. таких путей, компонуя которые можно получить всевозможные пути из входа графа в выходы. Основной метрикой сложности он предлагает считать цикломатическую сложность графа программы, или, как ее еще называют, цикломатическое число Мак-Кейба, характеризующее трудоемкость тестирования программы.

Для вычисления цикломатического числа Мак – Кейба *Z(G)* применяется формула

$$Z(G) = m - n + 2r ,$$

где m – число дуг ориентированного графа G;

n – число вершин;

r – число компонентов связности графа.

Число компонентов связности графа можно рассматривать как количество дуг, которые необходимо добавить для преобразования графа в сильно связный. Сильно связным называется граф, любые две вершины которого взаимно достижимы. Для графов корректных программ, т. е. графов, не имеющих недостижимых от точки входа участков и “висячих” точек входа и выхода, сильно связный граф, как правило, получается путем замыкания дугой вершины, обозначающей конец программы, на вершину, обозначающую точку входа в эту программу. По сути *Z(G)* определяет число линейно независимых контуров в сильно связном графе. Иначе говоря, цикломатическое число Мак-Кейба показывает требуемое количество проходов для покрытия всех контуров сильно связного графа или количество тестовых прогонов программы, необходимых для исчерпывающего тестирования по критерию “работает каждая ветвь”.

К достоинствам меры относят простоту ее вычисления и повторяемость результата, а также наглядность и содержательность интерпретации. В качестве недостатков можно отметить: нечувствительность к размеру ПО, нечувствительность к изменению структуры ПО, отсутствие корреляции со структурированностью ПО, отсутствие различия между конструкциями Развилка и Цикл, отсутствие чувствительности к вложенности циклов. Недостатки цикломатической меры привели к появлению ее модификаций, а также принципиально иных мер сложности.

Исходя из этого, Г. Майерс предложил расширение этой метрики. Суть подхода Дж. Майерса (*мера Майерса*) состоит в следующем. Дж. Майерс предложил в качестве меры сложности интервал $[\nu_1, \nu_2]$, где ν_1 – цикломатическая мера, а ν_2 – число отдельных условий плюс единица. При этом, оператор *DO* принимается в качестве одного условия, а оператор *CASE* с *n* – исходами принимается за *n – 1*- условий. Введенная мера получила название интервальной меры.

К мерам сложности, учитывающим вложенность управляющих конструкций, относят *меру Харрисона-Мейджела*, учитывающую уровень вложенности и протяженности ПО, *меру Пивоварского* – цикломатическую сложность и глубину вложенности. Функциональная мера сложности Харрисона-Мейджела предусматривает приписывание каждой вершине графа своей собственной сложности (первичной) и разбиение графа на сферы влияния предикатных вершин. Сложность сферы называют приведенной и слагают ее из первичных сложностей вершин, входящих в сферу ее влияния, плюс первичную сложность самой предикатной вершины. Первичные сложности вычисляются всеми возможными способами. Отсюда функциональная мера сложности ПО есть сумма приведенных сложностей всех вершин управляющего графа.

Мера Пивоварского ставит целью учесть в оценке сложности ПО различия не только между последовательными и вложенными управляющими конструкциями, но и между структурированными и неструктурированными программами. Она выражается соотношением

$$N(G) = r^*(G) + \sum_i U_i \ ,$$

где $r^*(G)$ – модифицированная цикломатическая сложность, вычисленная так же, как и *Z(G),* но с одним отличием: опера-

тор *CASE* с *n*-выходами рассматривается как один логический оператор, а не как *n – 1* операторов. U_i – глубина вложенности *i*-й предикатной вершины. Для подсчета глубины вложенности предикатных вершин используется число сфер влияния. Под глубиной вложенности понимается число всех сфер влияния предикатов, которые либо полностью содержатся в сфере рассматриваемой вершины, либо пересекаются с ней. Глубина вложенности увеличивается за счет вложенности не самих предикатов, а сфер влияния. Сравнительный анализ цикломатических и функциональных мер с обсуждаемой для десятка различных управляющих графов программы показывает, что при нечувствительности прочих мер этого класса, мера Пивоварского возрастает при переходе от последовательных программ к вложенным и далее к неструктурированным.

Метрика Джилба. Одной из наиболее простых, но, как показывает практика, достаточно эффективных мер сложности программ является метрика Т. Джилба, в которой логическая сложность программы определяется как насыщенность программы выражениями типа *IF-THEN-ELSE*. При этом вводятся две характеристики: *CL* – абсолютная сложность программы, характеризующаяся количеством операторов условия; *cl* – относительная сложность программы, характеризующаяся насыщенностью программы операторами условия, т. е. число *cl* определяется как отношение числа *CL* к общему числу операторов:

- количество операторов цикла;
- количество операторов условия;
- число модулей или подсистем;
- отношение числа связей между модулями к числу модулей;
- отношение числа ненормальных выходов из множества операторов к общему числу операторов.

Используя метрику Джилба, можно дополнить ее еще одной составляющей, а именно характеристикой максимального

уровня вложенности оператора *CLI*, что позволяет не только уточнить анализ по операторам типа *IF-THEN-ELSE*, но и успешно применить метрику Джилба к анализу циклических конструкций.

Метрики сложности потока данных

Другая группа метрик сложности программ – метрики сложности потока данных, т.е. использования, конфигурации и размещения данных в программах. Характерными представителями этой группы метрик являются метрики спена, Чепина, Кафура. Рассмотрим эти метрики.

Метрика спена. Определение спена основывается на локализации обращений к данным внутри каждой программной секции. Спен – это число утверждений, содержащих данный идентификатор между его первым и последним появлением в тексте программы. Следовательно, идентификатор, появившийся *n* раз, имеет спен, равный *n-1*. При большом спене усложняется тестирование и отладка.

Метрика Чепина. Суть метрики состоит в оценке информационной прочности отдельно взятого программного модуля с помощью анализа характера использования переменных из списка ввода-вывода. Все множество переменных, составляющих список ввода-вывода, разбивается на 4 функциональные группы:

1. *V* – вводимые переменные для расчетов и для обеспечения вывода. Примером может служить используемая в программах лексического анализатора переменная, содержащая строку исходного текста программы, т.е. сама переменная не модифицируется, а только содержит исходную информацию.

2. *M* – модифицируемые, или создаваемые внутри программы переменные.

3. *U* – переменные, участвующие в управлении работой программного модуля (управляющие переменные).

4. L – не используемые в программе (“паразитные”) переменные. Поскольку каждая переменная может выполнять одновременно несколько функций, необходимо учитывать ее в каждой соответствующей функциональной группе.

Далее вводится значение метрики Чепина:

$$CH = \omega_1 V + \omega_2 M + \omega_3 U + \omega_3 L$$

где $\omega_i (i = 1, 2, 3, 4)$ – весовые коэффициенты.

Весовые коэффициенты использованы для отражения различного влияния на сложность программы каждой функциональной группы. По мнению автора метрики, наибольший вес, равный трем, имеет функциональная группа U, так как она влияет на поток управления программы. Весовые коэффициенты остальных групп распределяются следующим образом: $\omega_1 = 1$, $\omega_2 = 2$, $\omega_4 = 0.5$. Весовой коэффициент группы L не равен 0, поскольку “паразитные” переменные не увеличивают сложность потока данных программы, но иногда затрудняют ее понимание. С учетом весовых коэффициентов предыдущее выражение принимает вид: $CH = V + 2M + 3U + 0.5L$.

Метрика Кафура. Метрика, основанная на учёте потока данных. Вводятся понятия локального и глобального потока:

Локальный поток информации из A в B существует, если:

- модуль A вызывает модуль B (прямой локальный поток);
- модуль B вызывает модуль A и A возвращает модулю B значение, которое используется в B (непрямой локальный поток);
- модуль C вызывает модули A, B и передаёт результат выполнения модуля A в B.

Глобальный поток информации из A в B через глобальную структуру данных D существует, если модуль A помещает информацию в D, а модуль B использует информацию из D. На

основе этих понятий вводится величина I – информационная сложность процедуры:

$$I = length \cdot (fan\_in \cdot fan\_out)^2 .$$

Здесь:

• *length* – сложность текста процедуры (меряется через какую-нибудь из метрик объёма, типа метрик Холстеда, Маккейба, LOC и т.п.);

• *fan_in* – число локальных потоков внутрь процедуры плюс число структур данных, из которых процедура берёт информацию;

• *fan_out* – число локальных потоков из процедуры плюс число структур данных, которые обновляются этой процедурой.

Можно определить информационную сложность модуля как сумму информационных сложностей входящих в него процедур.

Следующий шаг – определяется информационная сложность модуля относительно некоторой структуры данных. Информационная мера сложности модуля относительно структуры данных:

$$I = W \cdot R + W \cdot WrRd + WrRd \cdot R + WrRd \cdot (WrRd - 1) .$$

Здесь:

• W – число процедур, которые только обновляют структуру данных;

• R – число процедур, которые только читают информацию из структуры данных;

• *WrRd* – число процедур, которые и читают и обновляют информацию в структуре данных.

Следует отметить, что рассмотренные метрики сложности программ основаны на анализе исходных текстов программ и графов, что обеспечивает единый подход к автоматизации их расчетов.

II.4.4. Показатели функциональной надежности программного обеспечения

Показатели функциональной надежности информационных систем, изложенные в п.II.2.1, позволяют оценить все компоненты информационных систем, включая программное обеспечение. Вместе с тем, оценка функциональной надежности программного обеспечения должна учитывать индивидуальное влияние свойств каждого составного атрибута. Поэтому целесообразно базироваться на показателях п.II.2.1 и развить их с целью учета количественного влияния свойств составных атрибутов (см.п.II.4.3) на функциональную надежность каждой конкретной программы.

Рассмотрим две группы показателей функциональной надежности программ.

Первая группа показателей учитывает индивидуальное или комплексное влияние атрибутов *безошибочности, контролируемости, правильности, устойчивости к ошибкам и безотказности.* Вторая группа показателей учитывает влияние атрибутов *пригодности к восстановлению программы* и *готовности.*

Итак, первая группа показателей:

▪ *Вероятность безошибочного выполнения программы* – P.

Этот показатель предназначен для оценки отсутствия ошибок в данной программе после ее написания и отладки.

▪ *Вероятность отсутствия ошибки в результатах выполнений программы в течение заданного времени ее эксплуатации* – $P(t)$.

Этот показатель применяется для оценки уровня безошибочности и правильности выполнений программы в течение определенного времени t при следующих условиях:

– программа выполняется при поступлении заявки. Поток заявок случайный. Описывается вероятностью $P(n,t)$ того, что в

течение времени t поступит на вход информационной системы ровно n заявок на выполнение данной программы;

– безошибочность каждой реализации программы оценивается показателем P – его значение принимается постоянным на основании того, что после отладки интенсивность проявляющихся ошибок примерно постоянна, поскольку при устранении обнаруженной ошибки (группы ошибок) возможно внесение в программу новых ошибок;

– правильность результатов оценивается вероятностью $P_{П}$ того, что сбойные ошибки цифровой техники системы не приводят к искажению операндов и/или операторов программы в каждой отдельной ее реализации.

Для получения формульного выражения $P(t)$ представим процесс обслуживания заявок с помощью данной программы как процесс массового обслуживания в одноканальной системе без потерь заявок (предполагается, что при высоком быстродействии современных цифровых устройств очередная заявка на исполнение программы всегда застает ее свободной от обслуживания предыдущей заявки). Тогда по формуле полной вероятности получаем

$$P(t) = 1 - \sum_{n=0}^{\infty} P(n,t)[1 - (PP_{\text{П}})^n]. \qquad (4.1)$$

Для универсальности результата примем в качестве модели распределения времени между поступлениями заявок гамма-распределение с параметрами $\gamma = M / D; a = M^2 / D$, где M и D соответственно математическое ожидание и дисперсия времени между поступлениями заявок на выполнение объектной программы, реализующей функции данной исходной программы; a – порядок гамма-распределения.

Гамма-распределение часто применяется в качестве универсальной априорной модели, когда точная форма распределения неизвестна (см. гл. II.2). Следовательно,

$$P(n,t) = e^{-\gamma t} \sum_{k=na}^{na+a-1} (\gamma t)^k / k!$$

Подставив вероятность $P(n,t)$ в предыдущую формулу, находим

$$P(t) = 1 - \sum_{n=0}^{\infty} e^{-\gamma t} \sum_{k=na}^{na+a-1} (\gamma \cdot t)^k / k! \cdot [1 - (P \cdot P_{\text{П}})^n] .$$

После ряда преобразований находим приближенное выражение вероятности отсутствия ошибки в результатах выполнений программы в течение времени ее эксплуатации

$$P(t) \approx \exp[-\gamma \cdot (1 - \sqrt[a]{P \cdot P_{\text{П}}}) \cdot t] .$$

При значении $a = 1$ (случай экспоненциального распределения времени между заявками) данная формула преобразуется к следующему виду:

$$P(t) = \exp[-\gamma \cdot (1 - P \cdot P_{\text{П}}) \cdot t] =$$

$$= \exp(-\lambda \cdot t) = \exp[-\gamma \cdot (1 - P \cdot P_{\text{П}}) \cdot t], \qquad (4.2)$$

где $\lambda = \gamma \cdot (1 - P \cdot P_{\text{П}})$ – интенсивность ошибок, проявляемых при выполнениях программы.

▪ *Вероятность отсутствия отказов в работе информационной системы при выполнении программы в течение заданного времени ее эксплуатации* – $P_{Б}(t)$.

Этот показатель применяется для оценки уровня безотказности, отказоустойчивости и контролируемости системы при выполнениях программы в течение определенного времени t при следующих условиях:

– условия трансформации ошибки в частичный функциональный отказ в соответствии с п. II.2.2.1 формализуются в виде вероятности сложного события

$$g_{ФТ} = P\{(R \notin R) \vee (\delta > \delta_{ДОП}) \vee (\tau > \tau_{ДОП})\},$$

а вероятность частичного функционального отказа информационной системы относительно данной программы определяется как

$$G_{Ф}(t) = [1 - P(t)]g_{ФТ} = G(t)g_{ФТ};$$

– выходные результаты каждой реализации программы контролируются с помощью встроенных в информационную систему программных, программно – аппаратных и диагностических средств, а также с помощью периодических функциональных контролей. Эффективность своевременного, достоверного и безошибочного контроля отказов системы при выполнениях данной программы оценивается комплексным показателем – вероятностью правильного обнаружения α;

– в информационной системе предусмотрены средства обеспечения отказоустойчивости. В случае обнаружения отказа системы при выполнениях данной программы эти средства позволяют парировать этот отказ со значительным уровнем гарантии, который оценивается вероятностью ρ. Это условная вероятность того, что обнаруженный частичный функциональный отказ системы парирован с вероятностью β при условии безотказности средств обеспечения отказоустойчивости

$$P_{OO}(\mathrm{t}):\ \rho = \beta \cdot P_{OO}(\mathrm{t}).$$

Таким образом, вероятность отсутствия отказа системы относительно данной программы определяется как сумма вероятностей двух событий: А – в течение времени *t* информационная система безотказно выполняла все реализации данной программы и средства обеспечения отказоустойчивости работали безотказно; В – отказ системы относительно данной программы обнаружен и парирован с помощью штатных средств обеспечения отказоустойчивости.

В соответствии с перечисленными условиями вероятность отсутствия частичных функциональных отказов в работе информационной системы равна

$$P_{\text{Б}}(t) = P(A) + P(B) = [1 - G(t)g_{\text{ФТ}}]P_{\text{ОО}}(t) + \alpha \cdot G(t)g_{\text{ФТ}}\beta \cdot P_{\text{ОО}}(\mathrm{t}) =$$

$$= P_{\text{ОО}}(t)[1 - G(t)g_{\text{ФТ}} + \alpha \cdot \beta \cdot G(t)g_{\text{ФТ}}] = \qquad (4.3)$$

$$P_{\text{ОО}}(t)[1 - (1 - \alpha \cdot \beta)g_{\text{ФТ}}(1 - \exp(-\lambda t))].$$

• *Среднее время между частичными функциональными отказами информационной системы относительно данной программы*

$$T(t) = \int_0^\infty P_{Б}(t)dt = T_1 + T_2,$$

$$\text{где } T_1 = \int_0^\infty [1 - G(t)g_{ФФ}]P_{\text{ОО}}(t)dt \qquad (4.4)$$

– среднее время до функционального отказа системы относительно *данной* программы

$$T_2 = \int_0^{\infty} \alpha \cdot G(t) g_{\Phi\Phi} \beta \cdot P_{00}(t) dt \qquad (4.5)$$

– среднее время дополнительной безотказной работы системы относительно *данной* программы за счет средств обеспечения отказоустойчивости и контроля (в предположении идеальной надежности средств контроля, которое допустимо в том случае, если уровень надежности средств контроля не менее, чем на порядок выше уровня надежности системы).

При отсутствии в системе средств обеспечения отказоустойчивости (в этом случае $P_{\text{ОО}}(t) = 1, \beta = 0$) имеет место следующее выражение:

$$T = T_1 = \int_0^{\infty} [1 - G(t) g_{\Phi\Phi}] dt ,$$

которое означает, что предусмотренные в системе средства обеспечения контролеспособности не только не способствуют повышению функциональной надежности системы, но могут снизить ее (это становится очевидным, если вспомнить, что мы ранее анализировали результаты в предположении идеальной надежности средств контроля).

• *Среднее время простоя системы при устранении обнаруженной ошибки в программе*

$$T_{ПР} = \sum_{j=1}^{4} T_j ,$$

где T_1 – среднее время задержки до начала технического обслуживания программы; T_2 – среднее время локализации

ошибки в программе; T_3 – среднее время устранения ошибки и восстановления информационного процесса; T_4 – среднее время, затрачиваемое на контроль безошибочности выполнения процесса после его восстановления.

Численные значения приведенных составляющих временных затрат на устранение ошибки в программе с учетом ее интеграции в информационной системе зависят от архитектуры информационной системы, от организации обслуживания заявок, от эффективности системы контроля и диагностирования. Они су-

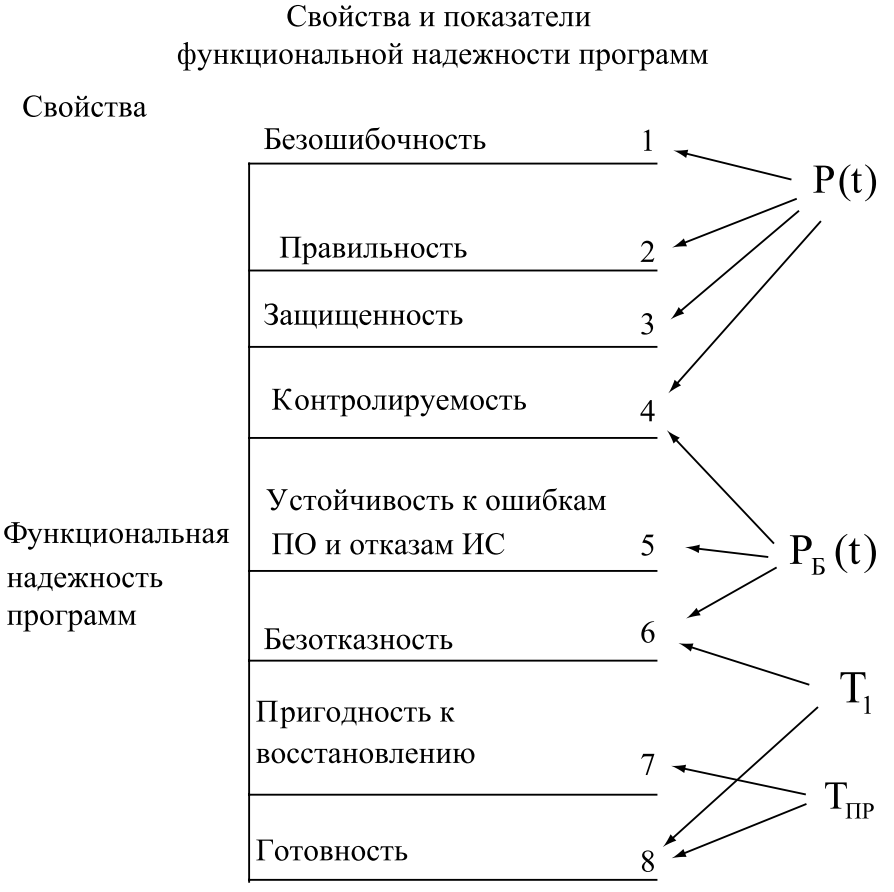


Рис. II.4.3. Связь показателей и свойств надежности программ

щественно зависят от принятой в системе стратегии устранения ошибки и восстановления информационного процесса.

Взаимосвязь свойств и показателей функциональной надежности программ проиллюстрирована на рис. II.4.3.

II.4.5. Модели надежности программного обеспечения

Термин ***модель надежности программного обеспечения,*** как правило, относится к математической модели, построенной для оценки зависимости надежности программного обеспечения от некоторых определенных параметров. Значения таких параметров либо предполагаются известными, либо могут быть измерены в ходе наблюдений или экспериментального исследования процесса функционирования программного обеспечения. Данный термин может быть использован также применительно к математической зависимости между определенными параметрами, которые хотя и имеют отношение к оценке надежности программного обеспечения, но тем не менее не содержат ее характеристик в явном виде. Например, поведение некоторой ветви программы на подмножестве наборов входных данных, с помощью которых эта ветвь контролируется, существенным образом связано с надежностью программы, однако характеристики этого поведения могут быть оценены независимо от оценки самой надежности. Другим таким параметром является **частота ошибок,** которая позволяет оценить именно качество систем реального времени, функционирующих в непрерывном режиме, и в то же время получать только косвенную информацию относительно надежности программного обеспечения (например, в предположении экспоненциального распределения времени между отказами).

Все множество моделей целесообразно разделить на группы по их назначению и по способу построения [31, 41 и др.]. ***По на-***

значению все модели надежности ПО подразделяются на *прогнозирующие, оценочные и измерительные* (рис. II.4.4). Это разделение соответствует последовательности этапов жизненного цикла ПО – этап разработки, этап отладки, этап сопровождения.

Рис. II.4.4. Классификация моделей функциональной надежности программного обеспечения по назначению

II.4.5.1. Прогнозирующие модели

Эти модели надежности основаны на определении технических характеристик создаваемой программы: длина, сложность, число циклов и степень их вложенности, количество ошибок на страницу операторов программы и др. По способу построения (рис. 4.3) прогнозирующие модели в большинстве своем являются ***эмпирическими***. При разработке моделей такого типа предполагается, что связь между надежностью и другими параметрами является статической. С помощью подобного подхода пытаются количественно оценить те характеристики программного обеспечения, которые свидетельствуют либо о высокой, либо о низкой его надежности. Так, например, параметр *сложность программы* характеризует степень уменьшения уровня ее надежности, поскольку усложнение программы всегда приводит к нежелательным последствиям, в том числе к неизбежным ошибкам программистов при составлении программ и трудностям их обнаружения и устранения. Иначе говоря, при разработке эмпирической модели надежности ПО стремятся иметь дело с такими параметрами, соответствующее изменение

значений которых должно приводить к повышению надежности ПО. Эмпирические модели базируются на анализе структурных особенностей программ. Они рассматривают зависимость показателей надежности от числа межмодульных связей, количества циклов в модулях, отношения количества прямолинейных участков программы к количеству точек ветвления и т.д.

Эмпирические модели основаны на анализе накопленной информации о функционировании ранее разработанных программ. Наиболее простая эмпирическая модель связывает число ошибок в программном обеспечении с его объемом. Опытные данные свидетельствуют, что к началу системного тестирования в программном обеспечении на каждые 1000 операторов приходится примерно 10 ошибок. Уровень надежности программного обеспечения считается приемлемым для начала эксплуатации, если тому же объему операторов будет соответствовать одна ошибка.

Часто эмпирические модели не дают конечных результатов показателей надежности, однако они включены в классификационную схему, так как развитие этих моделей позволяет выявлять взаимосвязь между сложностью ПО и его надежностью. Эти модели обычно прогнозируют количество ошибок в программе. Их можно использовать на этапе проектирования ПО, когда осуществлена разбивка на модули и известна его структура.

Типичными статическими моделями надежности ПО являются модели Холстеда и модели фирмы IBM. Например, модель Холстеда оценивает количество оставшихся в программе ошибок после окончания ее разработки:

$$N_{\text{ош}} = (N_1 + N_2)\log_2(\eta_1 + \eta_2)/3000,$$

где $N_{\text{ош}}$ – число ошибок в программе; N_1, N_2 – суммарное число операторов и операндов в ПО; η_1, η_2 – число операторов и операндов в программном средстве.

Определение параметров данного выражения применительно даже к средним по нынешним меркам программам затруднено. На практике применяют установленное в этой же работе соотношение для нормы ожидаемых ошибок:

$$N_{\text{ош}} / V = \kappa / 3000,$$

где V – объем программы, κ -коэффициент разветвленности программы. Для линейных программ κ =1, а для программ с большим числом ветвлений κ =4.

Фирма IBM использует эмпирическую модель, которая оценивает число ошибок в различных редакциях операционной системы:

$$N_{\text{ош}} = 23M_{10} + 2M_1,$$

где $23M_{10}$ – число модулей, потребовавших 10 и более исправлений; M_1 – число модулей, в которых обнаружено меньше 10 ошибок.

Следует отметить, что норма ожидаемых ошибок есть эквивалент вероятности, нормированной на одну операцию ошибки в программе.

II.4.5.2. Оценочные модели

Это наиболее широкий класс моделей надежности ПО. Оценочные модели основываются на серии тестовых прогонов и проводятся на этапах тестирования программы. В тестовой среде определяется вероятность ошибки программы при ее выполнении или тестировании.

В подавляющем большинстве это *динамические модели,* которые предназначены для оценки надежности программ на этапе их отладки. Данные модели подразделяются на две группы

моделей – одна из них без подсчета не выявленных ошибок, вторая с подсчетом таких ошибок (рис. II.4.5). Принято называть не выявленные ошибки в программе *дефектами.* В оценочных моделях, как правило, применяются показатели вероятности безошибочной работы программы в течение заданного времени, интенсивности ошибок и наработки на ошибку. При этом используется гипотеза об экспоненциальном (или близком к нему) законе распределения времени между ошибками в результатах выполнения программы.

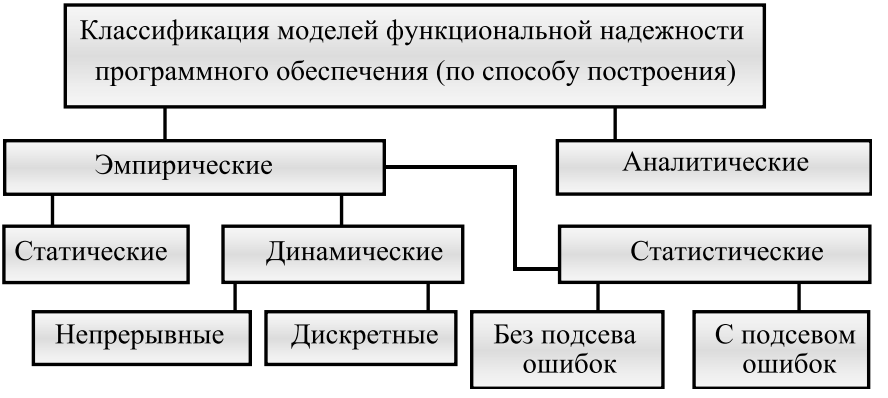


Рис. II.4.5. Классификация моделей функциональной надежности программного обеспечения по способу построения

Модели без подсчета ошибок основаны на измерении интервала времени между отказами и позволяют спрогнозировать количество дефектов, оставшихся в программе. После каждой ошибки оценивается надежность и определяется среднее время до следующей ошибки. К таким моделям относятся модели Джелински – Моранды, Шика – Вулвертона, вероятностно – возможностные [42] и некоторые другие. В модели *Джелински – Моранды* [6, 32, 33] интенсивность ошибок описывается кусочно – постоянной функцией, пропорциональной числу не

устраненных дефектов. После каждой проявившейся и устраненной ошибки интенсивность ошибки скачком изменяется на постоянную величину, а в промежутках между устранениями ошибок остается постоянной. С помощью метода максимального правдоподобия на основе набранной в процессе отладки программы статистики оценивается общее число дефектов в программе и определяется коэффициент пропорциональности, связывающий интенсивность ошибок с числом не устраненных дефектов. В модели *Шика – Уолвертона* предполагается, что интенсивность ошибок пропорциональна произведению числа не устраненных дефектов на время, затраченное на отладку. С увеличением времени отладки растет масштаб изменений интенсивности ошибок. При этом принимается, что исправление ошибок производится лишь после истечения интервала времени, на котором они возникали.

Для моделирования надежности программ *Литтлвуд* [41] применил байесовский подход, т.е. интенсивность ошибок ПО есть случайный процесс, характеристики которого зависят от уже происшедших ошибок. Как и большинство других авторов, Литтлвуд предположил, что отказы информационной системы по вине ПО происходят случайным образом при работе программы, причем интенсивность отказов есть функция числа не устраненных дефектов программы.

Модели с подсчетом ошибок базируются на количестве ошибок, обнаруженных на заданных интервалах времени. Возникновение ошибок в зависимости от времени является стохастическим процессом с непрерывной интенсивностью, а количество ошибок является случайной величиной. Обнаруженные ошибки, как правило, устраняются и поэтому количество ошибок в единицу времени уменьшается. К этому классу моделей относятся модели Шумана, Пуассоновская модель, вероятностно-возможностные и др.

Шуман [32] дополнил модель Джелински – Моранды новыми условиями. Также предполагая пропорциональность интенсивности проявляющихся ошибок числу не выявленных ошибок (дефектов), он вводит условие зависимости интенсивности ошибок от быстродействия компьютерных средств, объема программы и предполагаемого количества ошибок, допущенного при разработке программы. Он ввел масштабный коэффициент, учитывающий структуру программы, например, тот случай, когда одни и те же участки программы повторяются с новыми данными и, следовательно, новые операторы не вызываются и новые ошибки не проявляются. Кроме того, он предположил, что зависимость числа обнаруженных и исправленных ошибок от времени связана с изменением во времени показателей, характеризующих работу программистов. Автором предложено несколько разных функций, описывающих процесс устранения ошибок, выбор которых связан с особенностями программы.

Особое место среди моделей функциональной надежности ПО, ставших уже классическими, занимает модель *Мусы* [33]. Эта модель предназначена для предсказания поведения программы после сдачи ее в эксплуатацию. Как уже отмечалось, оценочные модели предназначены для определения уровня функциональной надежности ПО в процессе отладки, а измерительные модели – для оценки уровня функциональной надежности находящихся в эксплуатации программ. Поэтому модель Мусы занимает промежуточное место между оценочными и измерительными моделями. В основе построения модели лежит уверенность Мусы в том, что время функционирования программы (машинное время, в течение которого выполняется программа) является наиболее объективным временным интервалом, на котором следует оценивать ошибки программы. Отсюда следовал вывод (кстати, полностью разделяемый нами), что теория надежности ПО должна основываться не на календарном времени, а на времени функционирования программы в соответствии с запросами на ее выполнение.

В модели Мусы учитываются два вида времени функционирования: суммарное время t, которое отсчитывается в ходе разработки программы вплоть до контрольного времени, когда производится оценка надежности программы, и оперативное время τ, представляющее собой время исполнения программы, планируемое от контрольного времени и далее при условии, что устранения дефектов не будет.

Для времени функционирования t предполагается:

• Интенсивность ошибок пропорциональна числу не устраненных дефектов;

• Скорость изменения числа дефектов в программе пропорциональна интенсивности проявляемых ошибок в течение времени функционирования. Коэффициент пропорциональности обозначается как коэффициент уменьшения числа дефектов K. Он учитывает три типа дефектов, каждый из которых считается пропорциональным интенсивности ошибок из-за следующих причин: 1) увеличение числа дефектов в программе в связи с появлением новых дефектов в процессе устранения ошибок; 2) выявление дефекта при анализе текста программы, проведенном вслед за обнаружением связанного с ним дефекта во время тестирования; 3) наличие ошибок, причину которых не удается найти, и, следовательно, невозможности исправить дефекты, вызвавшие эти ошибки.

Суммарное число обнаруженных и устраненных дефектов описывается показательной функцией суммарного времени функционирования *t:*

$$M = M_0[1 - \exp(-\frac{\varepsilon \cdot t}{N \cdot T_0})],$$

где M_0 – первоначальное количество дефектов в программе; N – общее число ошибок, которые могут проявиться в процессе

сопровождения ПО; T_0 – средняя наработка на ошибку в начале испытаний; ε – коэффициент сжатия тестов, который учитывает отклонение испытаний от реальных условий. Например, если один час тестирования соответствует 12 ч работы в реальных условиях, то $\varepsilon = 12$.

Неизвестный параметр T_0 можно оценить из следующего соотношения:

$$T_0 = \frac{1}{f \cdot k \cdot M_0},$$

где f – средняя скорость выполнения одного оператора программы, равная отношению средней скорости исполнения программного обеспечения к числу команд (операторов); k – коэффициент проявления ошибок. Значение k определяется эмпирическим путем по однотипным программам. Обычно это значение находится в пределах $(1.5-4)\cdot 10^{-7}$; M_0 – первоначальное количество дефектов в программном обеспечении. Его можно оценить с помощью другой модели, позволяющей определить M_0 на основе статистических данных, полученных при тестировании. Число проявляющихся ошибок связано с числом дефектов в программе следующими соотношениями: $M_0 = \mathrm{K}N; M = \mathrm{K}n$, где n – число зарегистрированных ошибок. При этом коэффициент уменьшения числа дефектов K характеризует число устраненных дефектов, приходящихся на одну проявившуюся ошибку, а число зарегистрированных ошибок зависит от суммарного времени функционирования следующим образом

$$n = N[1 - \exp(-\frac{\varepsilon \cdot t}{N \cdot T_0})].$$

Текущее значение средней наработки до ошибки также зависит от суммарного времени функционирования

$$T = T_0[1 - \exp(-\frac{\varepsilon \cdot t}{N \cdot T_0})].$$

Вероятность безошибочного выполнения программы для оперативного периода оценивается с помощью следующего выражения:

$$P(\tau) = \exp(-\frac{\tau}{T}).$$

Преимущества и недостатки модели. К преимуществам модели можно отнести то, что нет необходимости фиксировать моменты отказов. В случае появления отказов ошибки регистрируются, а исправляются лишь по завершении этапа тестирования.

К недостаткам модели относится то, что для определения первоначального числа ошибок в программном обеспечении необходимо вести расчеты по другой модели, что приводит к дополнительным затратам времени.

II.4.5.3. Измерительные модели

Данный тип моделей предназначен для измерения надежности программного обеспечения в процессе его сопровождения в составе информационной системы. Сопровождение ПО осуществляется в течение длительного времени эксплуатации информационной системы. Отсюда повышенный интерес именно к измерительным моделям надежности. Все они основываются на статистических данных, полученных в результате множества прогонов программы. Среди измерительных моделей наиболее характерны модели *Коркорэна [39], Нельсона [36], коллектива авторов [43]* и др. и *мозаичная модель Пальчуна [41]*, которые базируются на выборе областей входных значений (рис. 4.3). Большой известностью также пользуется модель *Миллса*, в основу которой положен подсев ошибок без изменения входных значений.

Модель, предложенная коллективом авторов под руководством А.С. Шаракшанэ [43], по принципам построения во многом аналогична моделям Коркорэна и Нельсона, но отличается большей глубиной математической проработки, в частности динамических свойств модели. Роль показателя надежности здесь играет введенный авторами коэффициент отлаженности программы *R*, «представляющий собой вероятность того, что программа функционирует безошибочно в соответствии с техническим заданием» [43, стр.137]. Авторы исходили из предположения, что существует некоторая вероятность *p* получения положительного результата, характерная для данных входных условий и состояния программы, и вероятность $q = 1 - p$ получения реализации программы с отрицательным результатом, что означает возникновение ошибки в результатах данной реализации программы. Вероятность устранения обнаруженного дефекта принималась равной γ.

Тогда для *i*-й реализации вероятность отрицательного результата

$$q(i) = q(1 - q\gamma)^{i-1}.$$

Если испытания программы производятся *m* группами прогонов, то в этом случае

$$q_k(m) = \frac{1 - (1 - q\gamma)^k}{k\gamma}[(1 - q\gamma)^k]^{m-1},$$

где *k* – количество реализаций программы в группе.

Достоинством данной модели является наличие достаточно полной методики оценки параметров модели. При этом следует учесть, что авторами допускалось некоторое упрощение схемы самого процесса отладки, так как параметры модели принимались неизменными и не учитывалась возмож-

ность внесения в отлаживаемую программу новых ошибок. Одним из авторов создана модель, в которой сняты указанные ограничения. Сопоставление практических результатов применения указанных моделей позволяет заключить, что особо существенных различий с точки зрения характеристик поведения надежности программного обеспечения при этом не отмечается. Опираясь на практику, авторы решили дать первичную границу коэффициента отлаженности программы (вероятности безошибочности программы) на уровне 0.97-0.98 в зависимости от сложности программы, степени важности решаемых задач и т.д.

Мозаичная модель Б.П. Пальчуна реализует схему обработки дефектоскопической функции $\phi(i) = \sum_{j=1}^{i} I_j, i = 1,2,...,N$, где I_i – протокол результатов *i*-го испытания.

Оценка показателя функциональной надежности программы определяется следующим образом:

$$\hat{P}(N) = 1 - \hat{B}[\hat{A} - \phi \cdot (N-1)],$$

где $\hat{A} = \frac{\hat{V}_0}{V}; \hat{B} = \frac{\Delta\hat{\upsilon}}{V}$, $\hat{V}_0$ – оценка подобластей входной области V программы, которые соответствуют дефектным траекториям до начала испытаний, $\hat{\upsilon}_0$ – оценка алгебраического размера вариации входной дефектной подобласти при исправлении дефектного маршрута программы.

Основное ограничение применимости мозаичной модели – обязательное соответствие распределений вероятностей вектора входных величин при испытаниях и при эксплуатации. В ряде случаев это условие не удается выполнить. Для оценки эффективности применения данной модели требуется широкая практическая апробация.

II.4.5.4. Пример расчета функциональной надежности программы

Исходные условия

На этапе отладки проведено тестирование программы. Результаты тестирования приведены в табл. II.4.1. Известно, что программа в составе информационной системы предназначена для формирования команд управления подчиненным объектом в реальном масштабе времени. Заявки на выполнение программы поступают с интенсивностью $\gamma = 3600$ 1/ч при условии простейшего потока заявок. Вероятность трансформации проявленной ошибки программы в функциональный отказ составляет $g_{ФТ} = 0.01$. Предполагается, что интенсивность отказов средств обеспечения отказоустойчивости системы составляет $\lambda_{ОО} = 10^{-6}\frac{1}{ч}$. Также задано, что вероятность правильного обнаружения отказов оценивается на уровне α =0.99, а вероятность успешного парирования обнаруженного функционального отказа – β =0.95. Среднее время простоя информационной системы вследствие устранения ошибки программы равно τ = 2.5 ч.

Таблица II.4.1. Исходные данные к примеру II.4.5.4

Номер этапа тестирования	**Длительность этапа тестирования *t* [ч]**	**Количество выявленных ошибок программы *m***
1	12	2
2	11	3
3	12	1
4	8	0
5	10	0
6	11	1
7	12	2
8	13	0
9	9	0
10	10	1

Требуется рассчитать показатели функциональной надежности программы при заданном времени работы системы 8 ч в предположении отсутствия сбойных ошибок, а также в предположении, что при исправлении обнаруженных ошибок новые дефекты не вносятся в программу.

Решение.

Прежде всего, следует определить вероятность правильного однократного выполнения программы. Этот показатель используется при расчете большинства показателей функциональной надежности. Однако в исходных условиях задачи отсутствуют статистические данные о реализациях программы в процессе ее сопровождения. Также отсутствуют сведения о структурных характеристиках программы (количестве операторов, операндов, циклов и др.), что не позволяет использовать статические модели надежности типа модели Холстеда или модели IBM и им подобные. Поэтому выберем следующий путь решения задачи. С помощью модели Шумана [16,17] найдем первоначальное количество дефектов в программе, затем интенсивность ошибок, а далее с помощью формулы (4.1) может быть найдена искомая вероятность. С этой целью:

• Определяют первоначальное количество дефектов в программе N по следующей формуле:

$$\sum_{j=1}^{k} m_j \cdot \frac{\sum_{j=1}^{k} t_j}{\sum_{j=1}^{k} \frac{m_j}{N - n_{j-1}}} = \sum_{j=1}^{k} (N - n_{j-1}) t_j ,$$

где N – неизвестное число, k =10, m и n заданы в табл. II.4.1, $n_j = m_1 + m_2 + ... + m_j$.

По данным табл. II.4.1 путем подбора находят N =11.

• Определяют интенсивность ошибок программы по формуле Шумана

$$\lambda = C(N - n_k) = \frac{\sum_{j=1}^{k} \frac{m_j}{N - n_{j-1}}}{\sum_{j=1}^{k} t_j}(N - n_k) = 0.017 \frac{1}{ч}.$$

• Вероятность правильного однократного выполнения программы после ее отладки находят на основании формулы (4.2), где $\lambda = \gamma(1 - PP_{\text{П}})$ и

$$P = (1 - \frac{\lambda}{\gamma}) / P_{П} = (1 - \frac{0.017}{3600}) / 1 = 1 - 4.75 \cdot 10^{-6},$$

Здесь $P_{П} = 1$, так как в условиях задачи принято, что сбойные ошибки отсутствуют.

• Вероятность отсутствия ошибки в результатах выполнения программы в течение заданного времени $t = 8ч$

$$P(t) = \exp(-\lambda t) = \exp(-8 \cdot 0.017) \approx 0.86.$$

• Среднее время до ошибки программы

$$T = \frac{1}{\gamma(1 - P)} = \frac{1}{0.017} = 58.82 \ ч$$

• Вероятность отсутствия отказов в работе информационной системы при выполнении программы в течение заданного времени $t = 8ч$ (формула (4.3))

$$P_{Б}(t) = \exp(-\lambda_{ОО}t)[1-(1-\alpha\beta)g_{ФТ}(1-\exp(-\lambda t))] =$$

$$= \exp(-8\cdot 10^{-6})[1-(1-0.99\cdot 0.95)\cdot 0.01\cdot(1-\exp(-8\cdot 0.017)] =$$

$$= 1-0.88\cdot 10^{-4}$$

• Среднее время до частичного функционального отказа отказоустойчивой информационной системы (формула (4.4))

$$T_{О} = \frac{1-g_{ФГ}(1+\alpha\beta)}{\lambda_{ОО}} + \frac{g_{ФГ}(1+\alpha\beta)}{\lambda+\lambda_{ОО}} =$$

$$= \frac{1-0.01\cdot(1+0.99\cdot 0.95)}{10^{-6}} + \frac{0.01\cdot(1+0.99\cdot 0.95)}{0.017+10^{-6}} = 9.8\cdot 10^{5}\,ч$$

• Коэффициент частичной функциональной готовности информационной системы

$$K_{ФГ} = \frac{T}{T+\tau_{ПП}} = \frac{58.82}{58.82+2.5} = 0.96\,.$$

Контрольные вопросы

1. Приведите классификацию прикладного ПО.
2. Поясните специфические особенности встроенного ПО.
3. Перечислите и поясните содержание характеристик качества ПО.
4. Дайте определение функциональной надежности программ.
5. Какие атрибуты определяют функциональную надежность программ?

6. Приведите классификацию метрик качества программ.

7. Приведите и поясните метрики размера программ.

8. Приведите и поясните метрики сложности потока данных.

9. Приведите и поясните метрики сложности потока управления программ.

10. С помощью каких показателей учитывается влияние атрибутов функциональной надежности программ *безошибочности, контролируемости, правильности, устойчивости к ошибкам?*

11. С помощью каких показателей учитывается влияние атрибутов функциональной надежности программ *пригодности к восстановлению программы* и *готовности?*

12. Перечислите и поясните основные прогнозирующие и оценочные модели функциональной надежности ПО.

13. Перечислите и поясните основные измерительные модели функциональной надежности ПО.

14. Какая модель может быть рекомендована и каким образом использована для оценки ошибок в программе на этапе отладки?

Глава II.5. Функциональная надежность критически важных информационных систем

II.5.1.Понятие критически важной информационной системы

Многие информационно-управляющие системы предназначены для управления ответственными объектами. Этими объектами управления могут быть как различные технические комплексы, так и различные технологические процессы. Информационно-управляющие системы, как правило, применяются в ракетно-космической и авиационной технике военного и гражданского назначения для управления различными подвижными объектами, в робототехнике, а также в качестве АСУ в транспортной отрасли, энергетике, нефтегазовом комплексе, химическом производстве, системах жизнеобеспечения, вооружения и т.п. Такие информационно – управляющие системы принято называть критически важными информационными системами (сокращенно КВС).

Основная особенность процесса функционирования КВС заключается в необходимости формирования управляющей информации в масштабе реального времени. Это предполагает, что скорость обработки информации должна соответствовать динамике функционирования объекта управления, вернее быть выше, чем скорость реально протекающих процессов с целью обеспечения определенного запаса времени для принятия решения и формирования надлежащих управляющих воздействий.

Кроме того, масштаб реального времени определяет жесткость требований к директивным срокам и качеству выполнения возложенных на систему функций в условиях деструктивных воздействий. Как показано, например, в главе I.1 [1], допустимое время перерыва в работе системы соизмеримо с длительностью цикла управления и его превышение приводит к отклонению расчетных значений параметров процесса управления. Если такое отклонение больше некоторого предельного, то происходит срыв управления. При этом инерционность объекта управления однозначно определяет значение допустимого времени перерыва в работе системы: чем она меньше, тем более жесткие требования к частоте выработки управляющих воздействий. Так или иначе, применительно к КВС речь идёт о сохранении управления объектом при деструктивных воздействиях. При этом функциональная надежность данных систем заключается в существовании таких условий функционирования, при которых качество управления объектом не вызывает недопустимого ущерба жизни и здоровью людей, окружающей среде. Объекты, которые могут подвергаться ущербу вследствие снижения качества управления, обобщённо будем рассматривать как внешнюю среду по отношению к КВС. Так как степень возможного ущерба внешней среде зависит как от вида системы, так и от степени снижения качества управления вследствие деструктивного воздействия на неё, то, очевидно, имеет смысл говорить и о различных уровнях функциональной надежности КВС.

Функциональная надежность КВС определяется, в конечном счёте, способностью системы предотвращать или минимизировать последствия нарушения процесса нормального функционирования из-за деструктивных воздействий с целью исключения недопустимого ущерба внешней среде. Поддержание надежных условий обеспечивается наличием механизмов сохранения способности нормального функционирования информационных

систем в условиях деструктивных воздействий, что соответствует требованию обеспечения приемлемого уровня риска.

Вопросы сохранения способности нормального функционирования информационных систем в условиях деструктивных воздействий традиционно решаются с использованием понятий защиты информации от несанкционированного доступа и обеспечения функциональной надёжности и соответствующего им методологического базиса. Рассмотрим насколько возможно и достаточно их использование для создания эффективных механизмов обеспечения надежных условий функционирования КВС.

Защиту информации в системах определим как совокупность таких условий функционирования, при которых обеспечивается их защищённость от угроз нарушения конфиденциальности циркулирующей в системах информации (недопущение хищений, утечки, утрат и несанкционированного использования), целостности информационных ресурсов (модификации информации, отрицания ее подлинности, навязывания ложной информации) и доступности информационных услуг (блокирования и/или несанкционированного уничтожения информации). Это не противоречит существу понятия информационной безопасности, определения которого приведены в различных источниках [45-52]. Конфиденциальность, целостность и доступность отражают разные аспекты защиты информации, вследствие чего и задачи, решаемые для их обеспечения различны. Так, при обеспечении конфиденциальности и целостности, как правило, решаются задачи защиты информации от несанкционированного доступа. Для их эффективного решения широко используются организационные, технологические и правовые методы и технические средства защиты и разграничения доступа к системе, в том числе аутентификация и идентификация пользователей, антивирусная защита, а также защита от целенаправленной пе-

регрузки трафика сетей передачи данных, криптография, аудит и мониторинг состояния защищенности систем, испытания их программных средств на недекларированные возможности, классификация и описание которых приведены в обширной литературе, например в [53-59].

Они ориентированы, в основном, на защиту собственно информационных систем, в силу чего вне поля их применения остаётся решение такой важной для управляющих систем проблемы, как предотвращение или минимизация ущерба внешней среде, который может быть вызван негативным воздействием на последнюю вследствие нарушения нормального процесса функционирования таких систем.

Чрезвычайно важным аспектом обеспечения защиты информации в КВС является обеспечение целостности данных и программ и доступности к ним. Доступность принято понимать как возможность получения потребителями результатов выполнения функций, возложенных на систему, что определяется способностью системы гарантированно выполнять эти функции.

В условиях, когда вследствие деструктивных воздействий нарушается логика функционирования систем, но их элементы остаются технически работоспособными, такая способность обеспечивается применением соответствующих механизмов обнаружения специальных программно-технических воздействий и защиты от них. В условиях отсутствия преднамеренных деструктивных воздействий способность системы безошибочно выполнять возложенные функции обеспечивается поддержанием ее функциональной надежности.

В рамках теории функциональной надежности должны решаться вопросы формирования количественных требований, доказательство надежности (выполнения требований) на основе совместного применения результатов ускоренных натурных испытаний, аналитических методов, стендовых испытаний, эк-

спертных оценок и разработки «паспорта» функциональной надежности критически важной системы на всех этапах жизненного цикла; анализа рисков возникновения состояний опасных функциональных отказов, соответствующих недопустимому ущербу внешней среде, и создания механизмов блокирования перехода в такие состояния.

Обеспечение функциональной надежности достигается применением средств, локализующих развитие неблагоприятных процессов, защищающих систему от выдачи неправильных воздействий, предупреждающих о возможном наступлении экстремальных ситуаций, управляющих функционированием объекта в критических случаях (парирующих развитие отказа и переводящих объекты управления в защитное состояние). Для этих целей используются контролирующие и диагностирующие устройства, которые оценивают значения выходных параметров системы и значения специальных диагностических признаков, а в необходимых случаях и окружающей среды (вибрации, температура, электромагнитная обстановка и др.). Сравнение измеренных сигналов с их заданными значениями, обработка информации и принятие решения о необходимых действиях для предотвращения аварийной ситуации должны осуществляться устройствами, которые сами обладают высокой достоверностью, т.е., в данном случае, отвечающих заданным требованиям.

Основным направлением всех работ и исследований в области функциональной надежности КВС является принцип исключения возможности появления потенциально опасной ситуации (или сведению вероятности появления этого события к минимально допустимой величине).

Проблема обеспечения функциональной надежности КВС может быть рассмотрена с позиции трехуровневой логической организации процесса решения этой проблемы: теоретического, методического, организационно – технического.

Теоретический уровень предполагает формирование теоретических положений функциональной надежности и функциональной безопасности критически важных систем, включая разработку и взаимоувязку понятийного аппарата и подходов к определению надежных и безопасных условий их функционирования, а также возможных рисков нарушения таких условий, принципов их контроля и соответствующего реагирования с целью формирования облика системы обеспечения функциональной надежности и функциональной безопасности КВС в условиях деструктивных воздействий.

Так, для иллюстрации теоретического решения задачи оценки вероятности появления потенциально опасной ситуации в работе КВС можно с помощью модели информационных атак оценить эффективность блокирования этих атак средствами информационной безопасности и функциональной надежности. Идея этого подхода состоит в следующем:

– строится математическая модель потока информационных атак;

– разрабатывается модель защиты от одиночной информационной атаки средствами информационной безопасности и, в случае их неэффективности, средствами функциональной надежности (или функциональной безопасности);

– создается и анализируется комплексная модель, описывающая как поток атак, так и эффективность их блокирования возможными средствами.

Методика построения такой модели состоит в следующем.

Вначале задаются исходные условия. Например:

– поток информационных атак простейший с суммарной интенсивностью λ_A;

– вероятность обнаружения информационной атаки средствами обнаружения, предусмотренными для обеспечения функциональной надежности $\alpha(t \le \bar{t}_Д)$, где $\bar{t}_Д$ – допустимое время обнаружения и устранения опасного функционального отказа в КВС;

– в случае своевременного обнаружения реализованной информационной атаки возможно либо гарантированное устранение результатов атаки (например, блокирование выходных данных, исправление искаженных операндов и операций и др.), либо введение защитного отказа КВС (под защитным отказом понимается перевод КВС в состояние ожидания).

Затем строится сама модель расчета вероятности парирования информационной атаки. Так, при указанных исходных условиях могут выполняться следующие этапы ее построения:

1. В предположении простейшего потока событий задается следующая математическая модель потока информационных атак

$$P(n,t) = \frac{(\lambda_A t)^n}{n!} \exp(-\lambda_A t),$$

где $P(n,t)$ – вероятность того, что за время t создано ровно n информационных атак.

2. Модель защиты от одиночной информационной атаки – вероятность парирования атаки средствами информационной безопасности и функциональной надежности

$$P_З = P(А) + P(Б) = P_И + Q_И \alpha(t \le \bar{t}_Д),$$

где событие $А$ – атака парирована предусмотренными средствами обеспечения информационной безопасности. Вероятность этого события

$$\mathrm{P(A)} = P_И = \sum_{i=1}^{m} p_i - \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} p_i p_j + \ldots + (-1)^{m-1} \prod_{i=1}^{m} p_i .$$

В свою очередь, p_i – вероятность парирования атаки предусмотренным в системе i-м средством (мероприятием) защиты

информации; событие *Б* – атака реализована (предусмотренные средства обеспечения информационной безопасности пропустили атаку с вероятностью $Q_И = 1 - P_И$), но эта атака своевременно обнаружена и парирована средствами обеспечения функциональной надежности. Вероятность этого события

$$P(Б) = Q_И \alpha(t \le \bar{t}_Д).$$

3. Комплексная модель для определения вероятности парирования любой информационной атаки из потока атак

$$P_И(t) = \sum_{n=0}^{\infty} P(n,t)(P_З)^n = \sum_{n=0}^{\infty} P(n,t)[P_И + Q_И \alpha(t \le \bar{t}_Д)]^n .$$

Подставив выражение $P(n,t) = \frac{(\lambda_A t)^n}{n!} \exp(-\lambda_A t)$, получим

$$P_И(t) = \sum_{n=0}^{\infty} \frac{(\lambda_A t)^n}{n!} \exp(-\lambda_A t)[P_И + Q_И \alpha(t \le \bar{t}_o)]^n = \exp(-\lambda_A t) \sum_{n=0}^{\infty} \frac{(z)^n}{n!},$$

где

$$z = \lambda_A \cdot t \cdot [P_И + Q_И \alpha(t \le \bar{t}_Д)].$$

Поскольку $z<1$, то ряд $\sum_{n=0}^{\infty} \frac{(z)^n}{n!}$ является сходящимся и преобразуется к виду

$$\sum_{n=0}^{\infty} \frac{(z)^n}{n!} = \exp(z).$$

Таким образом, формула определения вероятности парирования информационной атаки имеет следующий вид:

$$P_И(t) = \exp[-\lambda_A \cdot t \cdot Q_И \cdot (1 - \alpha(t \le \bar{t}_Д))].$$

При неэффективности средств информационной безопасности остается рассчитывать, что целостность и доступность информации будет обеспечиваться средствами функциональной надежности, т.е. $Q_И = 1, P_И = 0$ и

$$P_И(t) = \exp[-\lambda_A \cdot t \cdot (1 - \alpha(t \le \bar{t}_Д))].$$

В другом предельном случае, когда средства информационной безопасности абсолютно эффективны, т.е. $Q_И = 0, P_И = 1$, получим

$$P_И(t) = \exp[-\lambda_A \cdot t \cdot 0 \cdot (1 - \alpha(t \le \bar{t}_Д))] = 1.$$

Разработка и исследование методов поддержания условий надежного и безопасного функционирования КВС и оценка эффективности их применения составляют ***содержание методического уровня***.

Техническую основу обеспечения функциональной надежности критически важных информационных систем составляет совокупность соответствующих программно-аппаратных средств защиты информации и обеспечения безошибочности выполнения информационных процессов. Создание и организация функционирования таких средств составляет содержание организационно-технического уровня.

Реализация теоретического, методического и организационно-технического уровней должна осуществляться в рамках соответствующей парадигмы обеспечения функциональной надежности критически важных систем. Функциональная надежность КВС определяется их способностью нормально фун-

кционировать в условиях деструктивных воздействий, а также противодействовать опасным функциональным отказам, наносящим недопустимый ущерб внешней среде.

II.5.2. Оценка сбойных ошибок в критически важных информационных системах

II.5.2.1. Оценка сбоев интегральных схем

С помощью изложенных в главе II.3 методик и алгоритма расчета, а также экспериментально – расчетных данных по сбоям базовых вентилей интегральных схем нами установлены возможные значения правильности работы интегральных схем средней (СИС), большой (БИС) и сверхбольшой (СБИС) интеграции для элементной базы типа ТТЛ. Эти значения согласуются с практически наблюдаемой частотой сбоев логических схем в цифровой технике. В качестве показателя функциональной надежности интегральных схем принята вероятность их правильной однократной работы:

Показатель	**СИС**	**БИС**	**СБИС**
$P_{\text{ИС}}$	$1-10^{-15}$	$1-5\cdot10^{-13}$	$1-10^{-10}$

Зная время переключения интегральной схемы и полагая, что цифровая схема в КВС регулярно используется с частотой, обратной времени ее переключения $\Delta\tau$, можно определить интенсивность сбоев интегральной схемы. При указанном выше условии допустимо предположение о геометрическом законе распределения сбоев с функцией распределения $P_{ИС}(n)=1-P_{ИС}^{n}$ и математическим ожиданием времени до сбоя

$$T_{ИС}=\frac{\Delta\tau}{1-P_{ИС}}.$$

Учитывая, что геометрическое распределение есть дискретный аналог экспоненциального распределения, можно выразить интенсивность сбоев через величину обратную средней наработке до сбоя. Следовательно, интенсивность сбоев интегральной схемы есть

$$\lambda_{ИС} = \frac{1 - P_{ИС}}{\Delta\tau}.$$

Так, при времени переключения $\Delta\tau = 10^{-8} c$ и вероятности однократной правильной работы $1 - P_{ИС} = 10^{-15}$ интенсивность сбоев интегральной схемы составит $\lambda = 10^{-7} \; 1/c = 0.3.10^{-4} \; 1/ч$.

Эти результаты свидетельствуют о том, что частота сбоев функционального характера на несколько порядков превышает частоту отказов цифровых устройств.

II.5.2.2. Порядок оценки сбойных ошибок

Структурно-информационная схема расчетов правильности выполнения цифровыми устройствами и КВС предусмотренных операций и процессов показана на рис. II.5.1.

Расчеты проводятся последовательно от периферийных функциональных частей (блоки расчета показателей правильности работы логических устройств и микроопераций) к функциональным структурам (блоки расчета показателей правильности выполнения операций, процессов). Исходными данными для расчетов на уровне функциональных частей являются: характеристики сбоев составных элементов; характеристики сообщений, представленные вероятностями $\{p_1\}$ и $\{p_0\}$ смены кодов 0-1 и 1 – 0 соответственно в каждом разряде сообщений; тип и содержание реализуемых цифровыми устройствами логических функций и микроопераций (представленных системой знаков $Ф$). В результате расчетов правильности работы функциональных частей определяются вероятности $\{P_{МО}\}$ для каждой микрооперации.

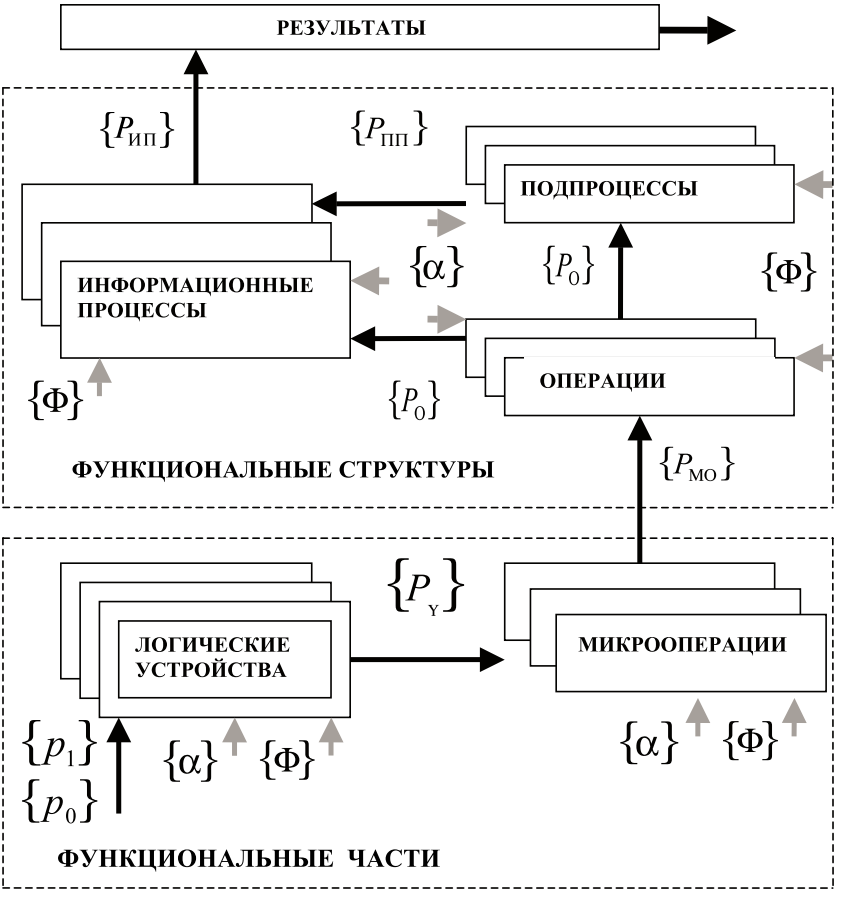


Рис. II.5.1. Структурно-информационная схема расчета вероятности отсутствия сбойных ошибок в результатах работы КВС

Все показатели правильности работы функциональных частей различной сложности рассчитываются по единой методике и на основании алгоритма, приведенного в п.II.3.3.2. Результаты этих расчетов используются в качестве входных данных при оценках правильности выполнения операций, а затем процессов и процессов КВС.

Оценка правильности выполнения функциональных структур начинается с определения вероятностей правильного вы-

полнения каждой операции $\{P_O\}$. С этой целью кроме вероятностей $\{P_{MO}\}$ задаются типы и структуры алгоритмов операций. Для оценки правильности выполнения информационных процессов исходными данными являются соответственно вероятности $\{P_O\}$ и правильности выполнения процессов $\{P_{ПП}\}$, типы и структуры алгоритмов выполнения процессов в целом (обозначен системой знаков α, см. стр. 65), а также организация обслуживания заявок на выполнение процессов.

Изложенные методики и структурно – информационная схема ориентированы на расчет вероятностей правильной работы функциональных структур на уровне операций и процессов. Расчет показателя правильности КВС, программное обеспечение которых содержит миллионы и более операций, целесообразно проводить экспериментально – расчетным путем, основываясь на устойчивости процентного содержания различных операций в процессах.

Пусть P_{Oi} – вероятность правильного выполнения операции *i*- типа, ν_i – отношение количества операций *i*-го типа к среднему количеству операций $\overline{M}$, реализуемых в информационном процессе (определяется по описанию программного обеспечения с помощью смеси Гибсона или путем экспериментального исследования задачи в динамике). Тогда вероятность правильного выполнения информационного процесса оценивается по следующей формуле:

$$\widehat{P}_{\text{П}} = \prod_{i=1}^{n} P_{\text{Oi}}^{\overline{\text{M}}\nu_i},$$

где $i = 1, 2, ..., n$; *n*- число разновидностей операций в процессе.

Для определения величины $\overline{M}$ предварительно подсчитывается среднее время однократного выполнения данного информационного процесса $\overline{t_{\text{П}}}$ и рассчитывается среднее время выполнения одной операции

$$\overline{t_{\text{О}}} = \sum_{i=1}^{n} \nu_i \overline{t_{oi}},$$

где $\overline{t_{\text{Oi}}}$ – среднее время выполнения операции *i*-го типа.

Тогда можно определить среднее количество операций по формуле $\overline{M} = \overline{t_{\text{П}}} / \overline{t_{\text{О}}}$ и подставить ее в формулу расчета вероятности правильного выполнения информационного процесса. В результате получим удобную для практических расчетов формулу оценки искомой вероятности:

$$\ln \widehat{P}_{\text{П}} = \frac{\overline{t_{\text{П}}}}{\overline{t_O}} \sum_{i=1}^{n} \nu_i \ln P_{Oi} . \qquad (5.1)$$

Следует отметить, что количество типов операций в цифровых устройствах КВС зависит от сложности решаемых ими задач и обычно ограничено диапазоном значений 10-200. Поэтому в процессе разработки системы есть возможность предварительно оценить вероятности правильного выполнения различных типов операций и даже уточнить эти оценки путем статических и динамических испытаний программ. Эти результаты могут быть включены в программную документацию системы.

Для ряда КВС есть возможность прогнозировать вероятности правильного выполнения различных типов операций по следующей формуле:

$$\ln P_{Oi} = n_{1i} \ln P_{\text{СИС}} + n_{2i} P_{\text{БИС}} + n_{3i} P_{\text{СБИС}},$$

где $i = 1, 2, ..., n$, $P_{\text{СИС}}(P_{\text{БИС}}, P_{\text{СБИС}})$ – вероятности правильной однократной работы интегральных схем (ИС) средней (большой, сверхбольшой) интеграции, $n_{ri} (r = 1, 2, 3)$ – количество ИС средней (большой, сверхбольшой) интеграции, используемых при выполнении *i*-й операции цифрового устройства.

Пример II.5.1. Оценить вероятность правильного однократного выполнения информационного процесса, среднее время реализации которого в КВС составляет $\overline{t_{\text{П}}} = 1$ мс при исходных данных табл. II.5.1 и при условии однократного выполнения каждого составного процесса.

Таблица II. 5.1. Исходные данные к примеру II.5.1.

Параметры / **Номер процесса**	**Вероятность правильного выполнения процесса ($P_{\text{ПП}}$)**	**Частота использования в программе (ν)**
Первый	$1-4\cdot10^{-8}$	0.1
Второй	$1-8\cdot10^{-9}$	0.4
Третий	$1-3\cdot10^{-9}$	0.5

Решение.

1. Рассчитывают среднее время выполнения составного процесса

$$\overline{t_{\text{ПП}}} = \overline{t_{\text{П}}} / 3 = 0.33 \text{ мс}.$$

2. Определяют искомую вероятность с помощью формулы (5.1)

$$\ln \widehat{P}_{\text{П}} = \frac{\overline{t_{\text{П}}}}{t_{ППП}} \sum_{i=1}^{3} \nu_i \ln P_{ППi} = 3\cdot[0.1\cdot\ln(1-4\cdot10^{-8}) +$$

$$+0.4\cdot\ln(1-8\cdot10^{-9} + 0.5\cdot\ln(1-3\cdot10^{-9});$$

$$\widehat{P}_{\text{П}} = 1-2.6\cdot10^{-8}.$$

II.5.3. Функциональная надежность операторов

II.5.3.1. Интенсивности ошибок операторов

Большинство широко известных аварий и катастроф, таких как катастрофа на Чернобыльской АЭС, на Саяно-Шушенской ГЭС, аварии на электростанции Three Mile Island, в Зибруже, Бхопале и др., напрямую связаны с ошибками человека. Поэтому понятно повышенное внимание в настоящее время разработчиков и, особенно, пользователей критически важных систем к функциональной надежности операторов.

Одновременно персонал рисковых объектов испытывает большую психологическую нагрузку. Факторы, ее обусловливающие, можно рассмотреть на примере [64] работы оперативного персонала традиционной промышленной электростанции: осознание степени опасности и тяжести последствий аварии; высокое давление пара и воды, высокое электрическое напряжение; движущиеся механизмы; вибрация; повышенная температура и пониженная влажность воздуха; монотонность обстановки; медленные изменения показаний приборов; размеренный ритм работы оборудования.

Следствия: расстройство сознания, рост психологической напряженности, потеря бдительности.

Психологи определяют две группы качеств профессиональной подготовки: знания и навыки, психологические, психофизиологические и социально-психологические качества, такие, как стрессоустойчивость, выдержка, добросовестность, ответственность, умение работать в группе. Как правило, основная часть оперативного персонала этому комплексу качеств удовлетворяет. Однако здесь, видимо, требуется полное 100% соответствие оперативного персонала этим качествам, так как неизвестно, на чью долю придется критическая ситуация, из которой придется выходить.

Не следует забывать о социально-психологическом аспекте надежности человеческого фактора в условиях политических столкновений в обществе (тревоги внешнего мира становятся фактором риска, когда у пульта обеспокоенный оператор). В целом сложная картина воздействий на человека, управляющего потенциально опасной техникой, представлена на рис. II.5.2 [69].

При этом для разных людей движущие мотивы профессионального поведения могут быть различные: познавательный интерес к делу, уважение к профессии, осознание ответственности, избежание конфликтов, карьеризм, утилитарный подход (зарплата, премия, жилье, машина, путевка на отдых и т.д.).

В экстремальных ситуации человек, как правило, продолжает ту линию поведения, которая отработана в предыдущий период. Мера воплощения привычных стереотипов зависит от выраженности таких личных качеств, как эмоциональная выдержка,

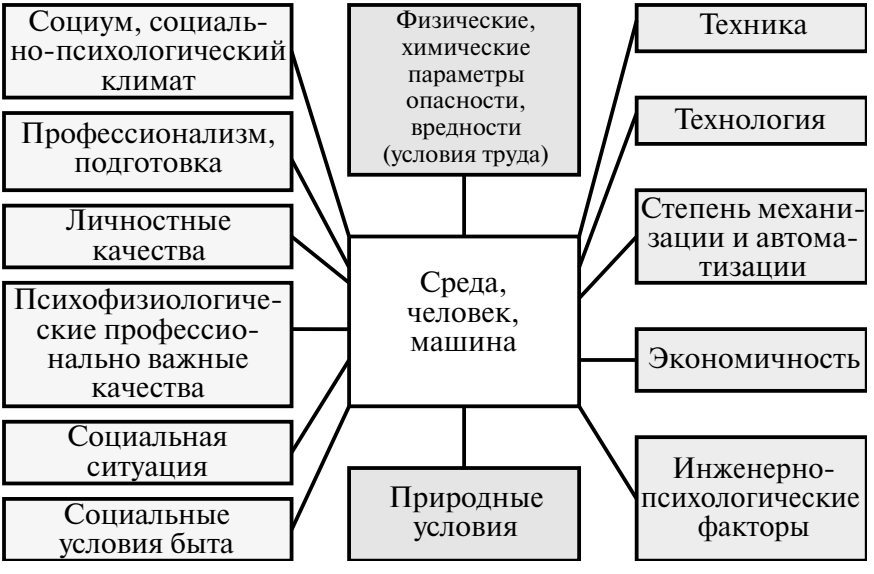


Рис. II.5.2. Факторы, воздействующие на человека, управляющего потенциально опасной техникой

добросовестность, доверчивость, самоконтроль, стрессоустойчивость и доброжелательность.

Поэтому формула безопасности: критическая позиция (I) + строго регламентированный и взвешенный подход (II) + коммуникабельность (III) = безопасность, – будучи внедренной в стереотип поведения оператора, обеспечивает:

– предотвращение (удаление от) аварийной ситуации;

– снижение процента ошибок при управлении аварией.

В возникновении наиболее тяжелой аварии за всю историю атомной энергетики – аварии на ЧАЭС – большую негативную роль сыграл оперативный персонал. Известно, что человеческие ошибки совершаются только в условиях, когда люди не могут их не сделать. Исходя из этого представляется важным оценить психологическую и социально-психологическую обстановку на ЧАЭС [69].

1. ЧАЭС – одна из лучших АЭС. Благоустроенный город Припять. Престижное место работы.

2. Квалификация оперативного персонала на ЧАЭС, и в пятой смене в частности (когда произошла авария), в общем не дают основания для сомнений: образование и практический опыт работы имелись.

3. ЧАЭС – Припять: обособление должностных группировок, внутри которых поддерживались отношения “своих”.

4. Подбор и расстановка кадров осуществлялась в соответствии с п.3.

5. Снижение активности жизненной позиции: определяющий мотив поведения – избежать конфликта с руководством (следствия: “Мне приказано – я делаю”, т.е. буквальное следование инструкциям; равнодушие к производству; уход в мир личных интересов; “позиция винтика”).

6. Традиция сохранения в тайне информации об аварийных случаях, что исключает возможность обучения персонала и воспитания чувства коллективной ответственности.

7. Внутренняя установка на выполнение задания (плана производства электроэнергии, программы испытаний и т.п.), но не на безопасность.

Перечисленное свидетельствует об отсутствии основных элементов культуры безопасности. Следует отметить еще ряд негативных факторов:

1. Работа оператора может быть успешной, если технические характеристики управляемой системы соответствуют возможностям человека (профессиональная подготовка, психофизиологические и психологические характеристики). Это не было обеспечено в данном случае.

2. Управление блоком осуществлялось на основании богатого операторского опыта, знаний физических и теплофизических процессов и интуиции.

Успешный выход из нестандартных ситуаций в прошлом укрепляет уверенность в личных возможностях операторов и способствует потере бдительности у персонала, а иногда порождает и особую “доблесть” риска (“Прорвемся, как и в прошлый раз!”).

3. Блочный щит управления был выполнен без учета требований эргономики (количество и важность информации).

Все эти негативные и позитивные обстоятельства реализовались во время аварии. Оценка масштабов аварии, доступная специалистам, не была доведена до сведения жителей города: соблюдать порядок, не сеять панику, ждать команд свыше – вот тон руководящих указаний, продолжавших линию секретности. Пока дети работников АЭС баловались в лужах города, сами работники АЭС ликвидировали аварию.

Мотивы любого поступка определяются объективными условиями и индивидуальными особенностями человека. Оперативной задачей на ночь с 25 на 26 апреля было завершение испытаний по выбегу ротора турбины. Развитие событий послужило

тому, что положительные личностные качества персонала – дисциплинированность, исполнительность – обратились в свою противоположность – безответственность и небрежность. Причина: привычка к существующему порядку вещей – "главное, чтобы не было конфликта с начальством", пассивная подчиненность, а не критическая позиция и личная ответственность за безопасность.

Возврат к проявлению личностных качеств в неискаженном служебной иерархией виде произошел после аварии. Оперативный персонал 5-й смены и прибывшие по тревоге работники АЭС проявляли выдержку, решительность, мужество, хотя по признакам острой лучевой болезни, появившимся в первые часы после аварии, представление об уровне радиации у них было. Поступки отражали высокую эмоциональную напряженность, активность гражданской позиции, имели целесообразный характер. В основе их лежали ощущения причастности к происшедшему событию, которое может иметь непредсказуемые последствия, ярко проявились чувства ответственности и долга в условиях непосредственной опасности для жизни.

Однако в ряде случаев отмечалась и неадекватная реакция на опасность: демонстрация бесстрашия, легкомысленный интерес к тому, как выглядит помещения 4-го блока, куски реакторного графита, разбросанные внутри и вне здания. В единичных случаях отмечалось и повышенное чувство опасности, нежелание покидать защищенное от радиации помещение даже для выполнения служебного задания.

После завершения первой, наиболее эмоционально напряженной фазы ликвидации аварии, отмечался в ряде случаев уход от инициативной, активной позиции, готовность подчиниться любому решению "сверху".

На развитие опасной ситуации оказывает процесс субъективного восприятия риска. Субъективное восприятие риска – очень

интересный и сложный вопрос. В зависимости от того, как люди воспринимают события катастрофического характера, формируется их поведение при различных формах деятельности.

Как показали исследования, на субъективное восприятие риска влияет множество факторов. Представляется необходимым привести только главные из них:

Оценка вероятностей наступления событий. Оценка вероятности наступления каких-либо событий является наиболее часто используемой операцией как в формальных методах принятия решений в условиях риска, так и в методах, основанных на профессиональных суждениях. Возможности человека правильно определять вероятности неопределенных событий существенно влияют на его способности оценивать степень риска в целом. Его оценки нарушают многие фундаментальные принципы рационального поведения.

Люди часто переоценивают надежность малых выборок, полагая, что их свойства характерны для всей совокупности. Малая выборка рассматривается как репрезентативная для суждения о характеристиках генеральной совокупности (эффект "репрезентативности"). Вероятности того или иного события часто определяются на основе того, как часто люди сталкивались с ними в прошлом (эффект "представительности"). Событие считается более вероятным, если человек может его представить, вспомнить аналогичные примеры. Это ведет к переоценке вероятностей ярких, запоминающихся событий и недооценке других.

Замечено, что *люди плохо учитывают априорные вероятности* и при оценке вероятности стремятся использовать преимущественно свой опыт ("эгоцентризм"), игнорируя и считая ненадежной любую другую априорную информацию. При оценке надежности оборудования технических систем это может приводить к большой переоценке вероятности аварий, если последние имели место, и к недооценке – в случае безотказной работы оборудования.

Известно также, что *человек недостаточно охотно меняет уже сложившиеся представления о вероятностях тех или иных событий под влиянием вновь поступившей информации.* Если информация не согласуется с его представлениями, он склонен считать ее случайной и ненадежной (“консерватизм”).

При *оценке вероятностей двух последовательных независимых событий люди стремятся установить между ними связь (иллюзия “Монте-Карло”).* При оценке вероятности совершения одновременно двух независимых событий люди часто игнорируют тот факт, что эта вероятность не может превосходить вероятности каждого из них в отдельности *(P(A) или P(B*) больше *P(A * B)*).

Значительное влияние на оценки людей оказывают точки отсчета. Когда в экспериментах людям давали разные значения вероятности события в качестве первого приближения и затем просили их скорректировать, ответы существенно отличались друг от друга и тяготели к точкам отсчета (эффект “якоря”).

Исследования показали, что *человек, как правило, недооценивает вероятность очень вероятных событий и переоценивает вероятность маловероятных событий*. Одновременно существует гипотеза, что человек не воспринимает вероятности порядка 10^{-6}, т.е. когда вероятность неблагоприятного исхода составляет один шанс из миллиона.

Личностные характеристики лица, принимающего решения (свойства личности). Этот фактор оказывает влияние как на субъективную оценку вероятностей событий, так и на оценку серьезности возможных последствий. Он же играет существенную роль и при оценке ситуации в целом. Пол, возраст, образование, образ жизни, эмоциональный настрой, социальные нормы и обычаи общества, степень доверия к органам власти, техническим экспертам, средствам массовой информации и другие факторы влияют на поведение человека при оценке уровня риска и безопасности.

Значимость последствий. Большую роль при этом играет то, какие потребности индивидуума могут быть удовлетворены в результате осуществления благоприятного исхода и какую угрозу ему может представлять неблагоприятный исход. Негативные последствия могут быть ранжированы с точки зрения их значимости для человека. Наиболее значимы последствия, ставящие под угрозу жизнь и здоровье человека, далее идут разнообразные последствия, связанные с семейным благополучием, карьерой и т.д.

Распределение угрозы во времени и пространстве. На восприятие риска оказывает большой влияние характер распределения негативных последствий во времени и пространстве. Так, чем ближе местожительство людей к рисковому предприятию, тем больше беспокойства они проявляют. Замечено также, что люди относятся терпимее к частым, распределенным во времени мелким авариям, чем к более редким катастрофам с большим числом жертв, даже если суммарные потери в первом случае гораздо больше, чем во втором.

Связь между возможными последствиями и их вероятностями. Опыт деятельности страховых фирм показывает, что люди по-разному оценивают степень риска от ситуаций с возможно малой вероятностью наступления катастрофических событий (землетрясения, наводнения) и ситуаций с большей вероятностью менее значимых потерь (автопроисшествие). Они активно пытаются уберечь себя от последних, например покупкой страховок, и проявляют безразличие к первым. Психологи объясняют этот феномен тем, что люди в практической деятельности стараются абстрагироваться от маловероятных событий.

Многообразие факторов возникновения ошибок операторов в КВС и большая значимость их последствий определили потребность в объединении усилий мирового сообщества в решении практических задач оценки численных характеристик работы различных операторов в сложных технических системах.

Первый и очень важный шаг в этом направлении состоит в создании базы данных по интенсивностям (или вероятностям) ошибок операторов при выполнении ими различных видов действий. Такая работа проводилась с начала 60-х годов прошлого века, однако по вполне объективным причинам не выполнена в полном объеме. Эти причины состоят в следующем:

– многие специалисты не в полной мере осознавали ранее важность этой проблемы;

– многие крупные организации не были готовы направить на сбор данных необходимые ресурсы;

– в условиях низких значений вероятностей ошибочных действий операторов для обеспечения достоверных статистических значений этих вероятностей необходимо собирать большие массивы экспериментальных данных, что опять – таки связано с большими расходами средств.

В настоящее время эта работа активизировались в ряде стран, получили «второе дыхание» и широкое практическое применение разработанные в прошлые годы методы оценки вероятностей ошибок операторов при выполнении ими предусмотренных групп действий.

В п. II.1.4 было отмечено, что из пяти характерных классов операторов, работающих в сложных технических системах, применительно к критически важным системам достаточно ограничиться следующими классами операторов:

– оператор-технолог;

– оператор-наблюдатель, контролер;

– оператор-исследователь,

а также в отдельных случаях нужно учитывать ошибки оператора-руководителя, хотя в сути своей они не вписываются в рамки функциональной надежности информационных систем, а относятся к области ошибок в принятии решений.

В дальнейшем ограничимся рассмотрением ошибок первых трех указанных классов операторов.

Большое влияние на безошибочность работы *операторов-технологов* оказывают степень обученности и скорость поступления информации. Процесс обучения достаточно хорошо аппроксимируется экспоненциальной функцией [9]

$$q = q_C + (q_0 - q_C)\exp(-n/N),$$

где q_C – установившееся (стационарное) значение частоты промахов, q_0 – начальное значение частоты промахов (до обучения), n – накопленная сумма сигналов, предъявленных оператору в предыдущих опытах, и половина числа сигналов, предъявленных в текущем опыте, N – «постоянная обучения», характеризующая объем обучения оператора. При $n=N$ разность $q_0 - q_C$ уменьшается на 63%. Считается, что установившееся значение q_C достигается через (4-5) «постоянных обучений» N.

Вероятность ошибки оператора также существенно зависит от скорости поступления информации. В [10] приведена экспериментальная зависимость доли q ошибочных ответов (нарушение последовательности, пропуск, включение лишнего, нарушение правил) от скорости C поступления информации (в бит/с). В результате обобщения результатов экспериментов в пяти публикациях различных авторов предложены аппроксимирующие формулы, которые сводятся к следующему выражению:

$$q = 9.7 \cdot 10^{-4} C^{1.77}.$$

Часто средняя вероятность ошибки на символ, допускаемая таким оператором, составляет примерно 5.10^{-4} при скорости

ввода информации не более 2 симв/с. Программные средства контроля правильности ввода информации позволяют более чем на порядок уменьшить вероятность ошибки.

Ошибки *оператора – наблюдателя, контролера* в определяющей степени зависят от объема сохраненной в его памяти информации. Установлено, что сбережение воспринятого материала как функции от времени, остающегося в памяти оператора материала обучения *L(t),* можно аппроксимировать или экспонентой или зависимостью вида

$$L(t) = k \big/ \log t,$$

где t – время, прошедшее с момента запоминания информации, k – коэффициент пропорциональности.

Ошибки этого класса операторов, также как и ошибки операторов – технологов, наиболее представительны в информационных системах и совместно составляют от 15% до 30% от общего числа ошибок функционирования систем. Вероятности ошибок в выполнении операций операторами – технологами и операторами – наблюдателями приведены в табл. II.5.2. Данные, приведенные в этой таблице, сформированы на основании материалов, приведенных в работах [12, 64 и 67], а также с учетом ряда опытных исследований и экспертных оценок.

Таблица II.5.2. Вероятности ошибок человека – оператора

Класс операторов	**Вид ошибки**	**Вероятность ошибки**
Оператор-технолог	***Ошибки считывания информации***	
	– Одинарного алфавитно-цифрового знака	$2 \cdot 10^{-4}$

Класс операторов	Вид ошибки	Вероятность ошибки
	– 5-буквенного слова при хорошем различении	$3 \cdot 10^{-4}$
	– Проверочного списка или цифрового показания	$1 \cdot 10^{-3}$
	– Аналогового индикатора	$5 \cdot 10^{-3}$
	– 10-значного числа	$6 \cdot 10^{-3}$
	Неисполнение отдельного требования при наличии памятки (инструкции) на рабочем месте	10^{-3}
	То же при отсутствии памятки и содержании в инструкции – до 10 требований – более 10 требований	 $3 \cdot 10^{-3}$ 10^{-2}
	при стрессовой ситуации	$5 \cdot 10^{-3}$
	Невыполнение отдельного пункта задания.	
	Количество пунктов в задании: 1 2 3 4 5	 10^{-3} $4 \cdot 10^{-3}$ $2 \cdot 10^{-2}$ $4 \cdot 10^{-2}$ $2 \cdot 10^{-1}$
	Ошибки выбора прибора (дисплея) при считывании информации.	
	Прибор выделен от подобных ему специальной маркировкой	$5 \cdot 10^{-4}$
	Прибор не выделен, но находится в группе подобных по назначению	10^{-3}
	Прибор не выделен ничем, кроме обозначения (бирки)	$3 \cdot 10^{-3}$

Класс операторов	**Вид ошибки**	**Вероятность ошибки**
	Ошибки при считывании показаний с несигнализирующих приборов (дисплеев), осматриваемых периодически (на пульте)	
	Стрелочный прибор	$3 \cdot 10^{-3}$
	Цифровой самописец (до 4 каналов)	10^{-3}
	Цифровой самописец (более 4 каналов)	$5 \cdot 10^{-2}$
	Диаграммный самописец	$6 \cdot 10^{-3}$
	Графопостроитель	10^{-2}
	Индикаторные цифровые лампы	10^{-3}
	Выявление отказавшего прибора (при отсутствии других приборов, облегчающих диагностику)	0,1
	Записи числовой информации (более 3 цифр)	10^{-3} (на одну цифру)
	Ошибки при считывании информации с приборов, по которым ведется работа (при измерениях)	
	Цифровой индикатор	10^{-3}
	Аналоговый измеритель – с меткой (чертой) – без метки	 10^{-3} $3 \cdot 10^{-3}$
	Аналоговый самописец – с меткой – без метки	 $2 \cdot 10^{-3}$ $6 \cdot 10^{-3}$
	Индикационная лампа-табло, выделенная маркировкой, находящаяся в функциональном блоке	$3 \cdot 10^{-3}$
	выделенная мнемолинией (по назначению) на мнемосхеме	10^{-3}
	Отсутствие реакции на сигнализацию	
	При срабатывании одного сигнализатора	$5 \cdot 10^{-4}$

Класс операторов	Вид ошибки	Вероятность ошибки
	При срабатывании дублирующего сигнализатора	10^{-4}
	То же при срабатывании сигнализатора и отсутствии инструкций по действиям персонала, включая отклик в течение минуты после отключения сигнала и уяснения задачи	10^{-3}
Оператор наблюдатель-контролер	*Ошибки при составлении инструкций*	
	Пропуск отдельного важного для безопасности требования	10^{-3}
	Некорректная запись требования в инструкции (несоответствие маркировке оборудования и т.п.)	$3 \cdot 10^{-3}$
	Необнаружение ошибок при контроле работы персонала	
	Контроль выполнения заданий (по графику)	$3 \cdot 10^{-3}$
	Ежесменный контроль выполнения письменных инструкций	10^{-2}
	Проверка исполнения инструкций – при нормальных условиях – при отклонениях от нормальных условий	 10^{-3} 10^{-2}
	Проверка выполнения профилактических работ	$5 \cdot 10^{-2}$
	Невыявление ошибок при оперативном контроле	
	Проверка подготовительных работ при наличии конкретных требований	10^{-2}

Класс операторов	**Вид ошибки**	**Вероятность ошибки**
	То же, но без инструкций, программ	0,1
	Целевые проверки выполнения задания	0,2
	Проверки с участием в работах	$5 \cdot 10^{-2}$
	Проверки при наличии угрозы личной безопасности	10^{-2}
	Проверка неисправного стрелочного прибора	10^{-3}
	Ошибка обнаружения неправильного состояния при беглом осмотре	$3 \cdot 10^{-3}$
	Ошибки при работе с органами управления	
	Ошибка выбора органа среди подобных	0,1
	если он выделен только маркировкой	$3 \cdot 10^{-3}$
	если он находится в группе подобных органов	10^{-3}
	если выделен специальной разметкой	$5 \cdot 10^{-4}$
	или находится в функциональном блоке	10^{-3}
	Ошибка выбора положения многопозиционного переключателя	$3 \cdot 10^{-3}$
	Ошибка выбора направления переключения тумблера	10^{-3}
	без нарушения стереотипов	10^{-3}
	при высоком уровне стресса	10^{-2}
	Ошибки расчетов	
	Простая арифметическая ошибка	10^{-2}
	Простая алгебраическая ошибка	10^{-2}
	Ошибочная запись информации или считывание графа	0.1
	Ошибочный ввод 10 цифр в калькулятор	$5 \cdot 10^{-2}$
	Ошибочный набор 10 цифр	$6 \cdot 10^{-2}$
	Наличие зависимости исполнителей работ (условные вероятности отказов)	

Класс операторов	Вид ошибки	Вероятность ошибки
	При среднем уровне зависимости (диспетчер и начальник смены, участка) в случае выполнения работ на диспетчерском пульте	0,15
	При низком уровне зависимости (диспетчер и инспектор пожарной охраны и т.п.)	$5 \cdot 10^{-2}$

Ошибки оператора – исследователя могут проявляться при решении различных задач анализа, синтеза, которые возникают в процессе разработки и проектирования информационных систем. Согласно рекомендациям знаменитого математика и педагога Пойа можно сформулировать следующие основные этапы решения задачи:

• *Поймите задачу* (Изучите данные. Изучите неизвестные. Достаточно ли данных для решения? Не противоречивы ли данные?)

• *Составьте план* (Чего вы должны добиваться? Какие методы проектирования будут использоваться? Встречалась ли вам уже такая задача? Не знаете ли вы близкой задачи? Можете ли вы воспользоваться ее результатом? Можете ли вы решить более специализированную или аналогичную задачу? Можете ли вы решить часть задачи?)

• *Выполните план* (Следуйте своему плану решения задачи. Проверяйте правильность каждого шага.)

• *Проанализируйте решение* (Все ли данные вы использовали? Проверьте правильность решения. Можете ли воспользоваться полученным результатом или примененным методом при решении других задач?)

Действия оператора – исследователя во многом имеют аналитический характер, а это означает, что прогнозирование вероятнос-

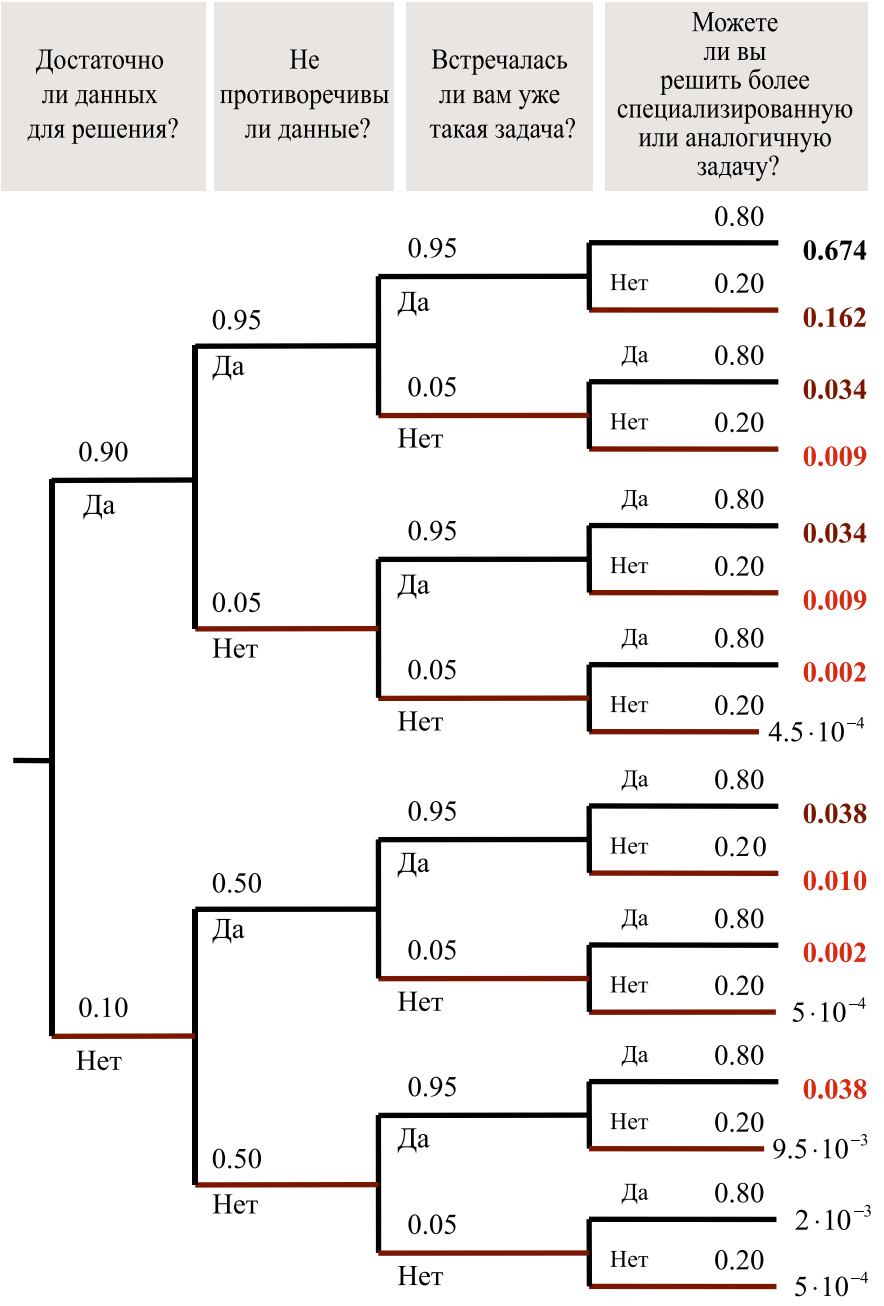


Рис. II.5.3. Дерево событий для оценки вероятностей ошибок оператора-исследователя

тей ошибок этого класса операторов на основании статистических данных бессмысленно – определяющее влияние на уровень ошибок оказывает интеллект, а затем и подготовка оператора. Вместе с тем, с помощью известных методов анализа, например, дерева событий (ETA), а также с привлечением экспертов можно оценить вероятности ошибок операторов – исследователей при выполнении ими нескольких этапов решения задачи. На рис. II.5.3 приведен пример оценок вероятностей одиночных, двойных и т.д. ошибок оператора при решении фрагмента задачи. Возможные ошибки оператора перечислены над деревом событий. Вероятности возникновения каждой ошибки оцениваются экспертами либо принимаются аналогичными известным значениям при выполнении подобных действий. Максимально возможное количество ошибок в данном примере – четыре (см. рис. II.5.3). Темно-красным цветом помечены ветви дерева событий с ошибками оператора. Этим же цветом помечены численные значения вероятностей одиночных ошибок. Оранжевым цветом помечены значения вероятностей двойных ошибок. Приведенным примером подчеркивается то обстоятельство, что оценка вероятностей ошибок в интеллектуальной деятельности оператора не может основываться на табличных данных – при решении каждой задачи целесообразен индивидуальный подход.

II.5.3.2. Прогнозирование функциональной надежности операторов

В качестве базового показателя функциональной надежности оператора целесообразно принять вероятность отсутствия ошибки при выполнении им предусмотренных действий в рамках каждой конкретной задачи (информационного процесса). Прогнозирующие расчеты этого показателя возможны с помощью известных методов. К ним относятся: Метод индекса успешного правдоподобия – SLIM (Success Likelihood Index Method); Ме-

тод диаграмм влияния – IDA (The Influence Diagram Approach); Метод корреляции надежности человека в познавательной деятельности – HCR (Human Cognitive Reliability Correlation); Метод прогнозирования интенсивности ошибок человека – THERP (Technique for Human Error Rate Prediction) и др. Каждый из этих методов ориентирован на определенные виды деятельности операторов. Для прогнозирующих расчетов функциональной надежности операторов критически важных информационных систем удобно применять Эмпирический способ оценки ошибок операторов – TESEO (Empirical Technique to Estimate Operator Errors) и/или Способ оценки и снижения ошибок человека – HEART (Human Error Assessment and Reduction Technique) [65].

Метод TESEO позволяет учесть характер и степень напряженности работы оператора, уровень подготовки, режим работы и эргономику рабочего места оператора. Эти сведения косвенно присутствуют в известных базах данных по вероятностям ошибок операторов. Поэтому, если можно найти вероятности $q_i, i = 1, 2, ..., z$ ошибок операторов при выполнении им предусмотренных z действий в рамках каждой конкретной задачи (информационного процесса), то базовый показатель функциональной надежности оператора определяется без применения метода TESEO в виде:

$$P_{\text{ОП}} = P_1, \text{ где } P_1 = \prod_{i=1}^{z} (1 - q_i)$$

– вероятность безошибочного выполнения оператором предусмотренных действий по табличным данным. Если нет достаточных данных для определения вероятности P_1, то производится оптимистичный прогноз надежности оператора в виде $P_{\text{ОП}} \le P_2 = 1 - Q$. Здесь Q – вероятность ошибки оператора, определенная по методу TESEO, и учитывающая условия работы и подготовки оператора.

Суть метода TESEO состоит в следующем. Для каждой задачи, выполняемой оператором, определяется пять актуальных факторов из общего перечня факторов, приведенного в табл. II.5.3. Затем перемножаются предложенные данным методом численные значения выбранных факторов. Полученный результат расценивается как вероятность ошибки оператора.

Таблица II.5.3. Исходные данные в методе TESEO

Факторы	**Численные значения**
Деятельность	
Простая	0.001
Требующая внимания	0.01
Не рутинная	0.1
Временная напряженность (в единицах времени)	
2Р ИЛИ 3НР (2 единицы рутинная или 3 единицы не рутинная)	10
10Р ИЛИ 30НР	1
20Р	0.5
45НР	0.3
60НР	0.1
Оператор	
Квалифицированный специалист	0.5
Средней квалификации	1
Слабо подготовленный	3
Трудности	
Аварии	3
Возможность аварии	2
Нормальный режим	1
Эргономика (условия взаимодействия оператора с системой)	
Отличная	0.7
Хорошая	1
Удовлетворительная	3-7
Очень плохая	10

Пример II.5.2. Оператор-технолог участвует в выполнении в системе информационного процесса. Работа требует внимания, рутинная. Временная напряженность – 20 с. Оператор средней квалификации. Режим работы нормальный, эргономика удовлетворительная. Требуется определить вероятность безошибочной работы оператора.

Решение.

По табл. II.5.3 находят численные значения актуальных факторов:

Длительность (требует внимания) – 0.01;

Временная напряженность (20Р) – 0.5;

Оператор (средняя квалификация) – 1;

Трудности (нормальный режим) – 1;

Эргономика (удовлетворительная) – 5.

Перемножая приведенные численные значения актуальных факторов, находим прогноз ошибки оператора Q=0.025 и вероятность его однократной безошибочной работы

$$P_{\text{ОП}} \le P_2 = 1 - Q = 0.975.$$

Достоинство метода TESEO в его простоте. Однако применение этого метода не гарантирует приемлемую точность прогноза.

Более строгий и точный метод прогноза – метод HEART. Он предусматривает выбор значения вероятности ошибки человека из таблицы вероятностей ошибок операторов и затем изменение этого значения в результате умножения на поправочные коэффициенты, которые находятся из таблицы условий, влияющих на ошибки. В табл. II.5.4 приведены значения максимальных множителей наиболее существенных условий, влияющих на ошибки.

Таблица II.5.4. Максимальные множители наиболее существенных условий, влияющих на ошибки операторов (фрагмент таблицы поправочных коэффициентов)

Условия, влияющие на ошибки (УВО)	**Максимальный множитель**
Незнакомая важная редкая ситуация	× .17
Нехватка времени для определения / исправления ошибки	× .11
Легко закрыть информацию	× .9
Нет доступных средств передачи информации	× .8
Нет доступных средств защиты от непреднамеренных действий	× .8
Необходимость осваивать новые идеи	× .6
Необходимость передачи знаний между задачами	× .5.5
Неясность в требованиях к результатам выполнения процессов	× .5
Несоответствие между ожидаемым и реальным риском	× .4
Слабая / неясная обратная связь системы	× .4
Несоответствие действий системы	× .4
Неподготовленный оператор	× .3
Отсутствие независимой проверки	× .3
Несоответствие уровня образования задаче	× .2
Отсутствие умственной / физической тренировки	× .1.8
Неясная ответственность	× .1.6
Задача не имеет внутреннего смысла	× .1.4
Несоответствующие показания	× .1.2

Порядок применения метода HEART следующий.

1. Выбирается значение вероятности ошибки человека, в частности из табл. II.5.2. Например, оператор наблюдатель – конт-

ролер при составлении инструкции пропустил важное для безопасности требование. Вероятность этой ошибки $3 \cdot 10^{-3}$.

2. Определяются условия, влияющие на ошибку. Обозначим каждое условие, как $A_i, i = 1, 2, ..., m$ (здесь m – число условий влияния) и из таблицы вероятностей находятся соответствующие им максимальные множители. Например, условия, влияющие на указанную выше ошибку оператора – наблюдателя, следующие (см. табл. II.5.4):

- Необходимость осваивать новые идеи – 6;
- Несоответствие между ожидаемым и реальным риском – 4;
- Несоответствие уровня образования задаче – 2;
- Неясная ответственность – 1.6;
- Задача не имеет внутреннего смысла – 1.4.

3. Для каждого условия эксперт назначает по своему мнению «долю влияния» $\omega_i, i = 1, 2, ..., m$ каждого условия в диапазоне от 0 до 1 на результирующее значение вероятности ошибки оператора.

4. Строится расчетная таблица с использованием следующего соотношения

$$Q = g \prod_{i=1}^{m} (A_i - 1) \cdot (\omega_i + 1),$$

где Q – откорректированное значение вероятности ошибки оператора, g – табличное значение вероятности ошибки.

Табл. II.5.5. Результаты расчетов в соответствии с приведенным примером

i	Фактор			
1	Необходимость осваивать новые идеи	6	0.8	9.0
2	Несоответствие между ожидаемым и реальным риском	4	0.9	5.7
3	Несоответствие уровня образования данной задаче	2	0.5	1.5
4	Неясная ответственность	1.6	0.5	0.9
5	Задача не имеет внутреннего смысла	1.4	0.7	0.7

Результаты расчетов в соответствии с приведенным примером показаны в табл. II.5.5. Окончательное значение вероятности ошибки оператора равно произведению рассчитанных в таблице величин на исходное значение 0.003. Таким образом, вероятность ошибки оператора – наблюдателя равна

$$Q=0.003\times 9.0\times 5.7\times 1.5\times 0.9\times 0.7 =0.145,$$

а вероятность безошибочной работы оператора 1-0.145 = 0.855.

II.5.4. Опасные отказы в функционировании критически важных информационных систем

II.5.4.1. Введение

Функциональная надежность критически важной информационной системы определяется ее способностью предотвращать или минимизировать последствия нарушения процесса нормального функционирования из-за деструктивных воздействий с целью исключения недопустимого ущерба внешней среде. Поддержание надежных условий функционирования обеспечивается содержащимися в КВС механизмами парирования опасных отказов системы, что соответствует требованию обеспечения приемлемого уровня риска.

Под *опасным отказом* КВС принято считать не обнаруженный в течение допустимого времени по требованиям безопасности скрытый отказ. Опасный отказ может быть вызван как функциональным отказом, так и структурным отказом. Вследствие функционального отказа может возникнуть и существовать в течение времени большего допустимого ошибочная команда управления, которая вызывает *риск безопасности* социальной, экологической, экономической и т.д. Отказ технических средств (структурный

отказ) также в отдельных случаях может привести к опасному отказу. Например, вследствие короткого замыкания в цепи управления может быть сформирован ложный сигнал разрешения движения поезда на препятствие или несанкционированной команды запуска ракеты или ложный сигнал диспетчера и др.

Под риском понимают сочетание вероятности возникновения ущерба и тяжести этого ущерба [60]. Существуют другие формы определения риска. Например, риск – сочетание вероятности опасного события и величины ущерба, или *риск – это произведение частоты или интенсивности или вероятности возникновения опасного события на величину ущерба*. Все эти формулировки не противоречат друг другу – для оценки риска следует рассчитать или каким – либо образом обосновать вероятность наступления опасного события и на основе имеющихся данных относительно этого опасного события оценить тяжесть ожидаемого ущерба.

Существует допустимый уровень риска, при котором или ниже которого риск приемлем. Если риск превышает допустимый уровень, то нарушаются требования безопасности. Если риск безопасности приемлем, то проявившийся отказ устройства устраняется в результате восстановления устройства. Если риск безопасности превышает допустимый уровень, то устраняется полученный ущерб. В процессе этого устройство модифицируется, после чего переводится в исходное состояние (передается в исправном состоянии эксплуатирующей организации).

Следует руководствоваться основными постулатами безопасности:

- Абсолютной безопасности не существует;
- Остаточный риск всегда остается после принятия защитных мер;
- Безопасность достигается путем уменьшения риска до допустимого уровня или ниже его;

• Остаточный риск не должен превышать допустимый уровень риска безопасности устройства на всех этапах жизненного цикла.

В соответствии со стандартом [59] в качестве основного принципа определения допустимого риска целесообразно применить **принцип ALARP,** практикуемый в Великобритании. Суть его в следующем: «Риск настолько низок, насколько это достижимо на практике».

В простейшем случае величина риска определяется как произведение вероятности возникновения опасного отказа в течение единицы времени $Q_{ОП}(1) = \lambda$ на величину ущерба от этого отказа Y, т.е.

$$R = Q_{ОП}(1)Y = \lambda Y .$$

Оценка величины нанесенного ущерба в результате опасного отказа на практике обычно затруднена. Это обусловлено тем, что ущерб может носить одновременно и экономический и социальный характер, может быть связан с нанесением вреда жизни и здоровью людей и т.д. По этой причине в стандарте [67] сгруппированы частоты (интенсивности) опасных отказов, а также сгруппированы уровни тяжестей их последствий.

II.5.4.2. Базовая модель оценки опасных отказов объекта КВС

Построим базовую модель оценки опасных отказов и связанных с ними рисков в критически важных информационных системах. Термин «базовая» использован нами с единственной целью подчеркнуть, что в модели рассматриваемой системы (объекте) учитываются только минимально необходимые для данных целей исследования параметры. Предполагается, что в исследуемом объекте какая-либо естественная или искусст-

венная избыточность, в том числе структурное резервирование, отсутствует. Объект выполняет только один информационный процесс. Заявки на выполнение этого процесса представляют собой простейший поток. При работе объекта возможно нарушение его работоспособности вследствие отказа оборудования, а также вследствие проявления ошибки в его программе или вследствие сбойной ошибки, или ошибки оператора, взаимодействующего с устройством, или ошибки во входных данных. Функциональная ошибка по любой из перечисленных причин может трансформироваться в функциональный отказ объекта вследствие несоответствия результата выполнения процесса одному из разрешенных кодовых слов или вследствие того, что погрешность результата выше допустимой, или вследствие того, что время задержки результата больше допустимой. Любое нарушение работоспособности объекта как по причине функционального отказа, так и по причине отказа оборудования расценивается как его отказ. Отказ, не обнаруженный в течение допустимого времени по требованиям безопасности, расценивается как опасный отказ.

Объект охвачен одноуровневым контролем. Под одноуровневым контролем понимается введение в состав устройства встроенных аппаратных, программных, программно – аппаратных средств контроля и диагностирования, а также применение функциональных контролей, которые в совокупности правильно обнаруживают отказы устройства. Средства одноуровневого контроля в дальнейшем будем называть *штатными средствами* контроля. Эффективность контроля оценивается *вероятностью правильного обнаружения отказов* α – это комплексный показатель эффективности средств контроля, учитывающий полноту охвата устройства контролем и диагностикой, непрерывность, своевременность и достоверность результатов контроля и диагностирования, возможности применяемых функциональных контролей.

Вероятность правильного обнаружения отказа объекта α есть вероятность суммы конечного числа событий (см. I.3.4.1):

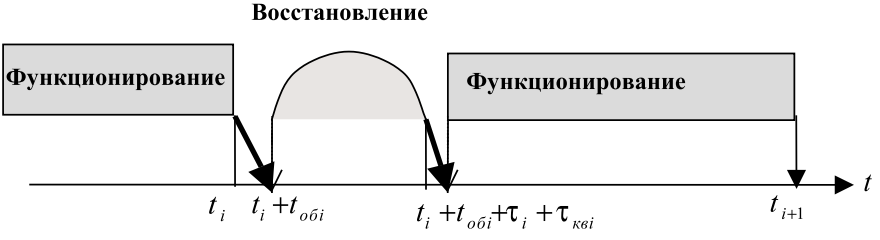
$$\alpha = P\{A\} = P\{\sum_{i=1}^{N} A_i\} = 1 - P\{\prod_{i=1}^{N} \overline{A_i}\} = 1 - \prod_{i=1}^{N} \overline{\alpha}_i \ .$$

$$\text{В свою очередь, } \overline{\alpha} = 1 - \alpha = \prod_{i=1}^{N} \overline{\alpha_i} \ .$$

Средства встроенного контроля и диагностирования, а также функционального контроля не контролируются (второй уровень контроля отсутствует).

Предполагаем, что штатные средства контроля идеально надежны.

В качестве показателя эффективности обнаружения отказа *нештатными средствами* целесообразно применять вероятность β того, что время существования скрытого отказа до его обнаружения не превышает заданное по требованиям функциональной безопасности допустимое время.



t_i *и* τ_i – моменты возникновения отказа и восстановления готовности КВС к функционированию соответственно

$t_{обi}$ – момент обнаружения скрытого отказа

$\tau_{квi}$ – длительность контроля КВС после восстановления

Рис. II.5.4. Временные задержки в восстановлении процесса функционирования КВС

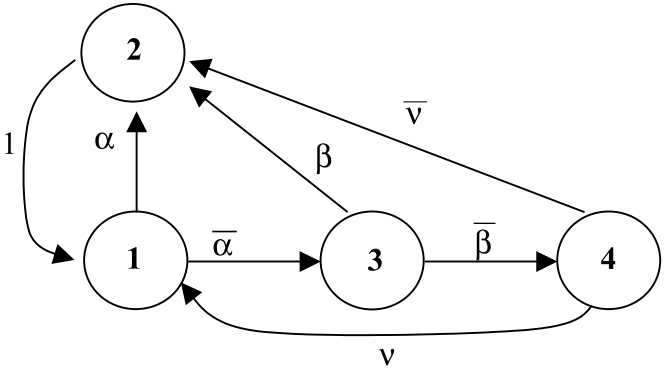


Рис. II.5.5. Граф исследования опасных отказов в КВС

В данной модели принято предположение, что в случае возникновения приемлемого риска объект эксплуатируется без модификаций, а его очередное состояние – восстановление. В противном случае он модифицируется и затем вводится в работу в исходном состоянии. Исследование функциональной надежности объекта КВС с учетом возможности возникновения опасных отказов удобно выполнять с помощью графа, показанного на рис. II.5.5.

На графовой модели рис. II.5.5 вершинами показаны следующие состояния:

1 – объект исправен;

2 – состояние восстановления работоспособности объекта;

3 – состояние обнаружения отказа нештатными средствами в случае пропуска его штатными средствами;

4 – состояние устранения ущерба и модификации объекта.

Дугам графовой модели приписаны следующие вероятности:

$p_{12} = \alpha$. Это вероятность того, что в случае обнаруженного отказа штатными средствами объект переходит из состояния 1 в состояние 2 восстановления;

$p_{13} = \overline{\alpha}$ – вероятность возникновения скрытого отказа объекта вследствие его пропуска штатными средствами обнаруже-

ния. Это вероятность перехода объекта в состояние обнаружения скрытого отказа нештатными средствами;

$p_{21} = 1$ – вероятность того, что после восстановления объект всегда переходит в исходное состояние (это принятое в модели допущение справедливо для внезапных отказов объекта);

$p_{32} = \beta$ – вероятность своевременного (в пределах допустимого времени) обнаружения нештатными средствами скрытого одиночного отказа;

$p_{34} = \overline{\beta}$ – вероятность перехода объекта в опасное состояние;

$p_{42} = \overline{\nu}$ – вероятность того, что риск нарушения функциональной надежности объекта меньше допустимого;

$p_{44} = \nu$ – вероятность возникновения недопустимого риска нарушения функциональной надежности объекта КВС.

Чтобы определить вероятность своевременного обнаружения скрытого отказа нештатными средствами нужно знать временные составляющие. Они показаны на рис. II.5.4. Если заведомо известно время, затрачиваемое штатным средством контроля и/или диагностики на обнаружение соответствующего типа отказа устройства, то интегральное время обнаружения находится усреднением этих известных времен относительно типов отказов. Однако применительно к нештатным средствам обнаружения отказа, который, как отмечалось выше, явился скрытым для штатных средств, задача определения времени обнаружения $\tau_{ОБН}$ становится неоднозначной. Причина в следующем. Пусть с помощью нештатных средств обнаруживается отказ, пропущенный штатными средствами. В этом случае для определения времени обнаружения $\tau_{ОБН}$ регистрируется момент обнаружения (обозначим t_i^*), затем с помощью экспертов оценивается момент его возникновения (обозначим $\hat{t}_i$). Безусловно, что однозначное определение момента возникновения отказа невозможно, поскольку по существу этот отказ оказался скрытым

для штатных средств. При оценке экспертами не исключено как необоснованное приближение, так и удаление предполагаемого момента возникновения отказа относительно его обнаружения. Однако по мере накопления экспериментальных данных такая оценка становится все более корректной.

Предполагается, что в течение ряда сеансов наблюдений за работой устройств данного типа получены статистические данные о моментах обнаружения их отказов, произведена предварительная статистическая обработка и показана с помощью известного критерия согласованности принадлежность этой выборки данных к одной генеральной совокупности. Итак, зарегистрирована выборка в количестве M значений t_i^* $(i=1...M)$. Среди этих событий есть некоторая часть $M_{ОП}$ опасных событий, которые расцениваются как опасные отказы, поскольку часть из них (M_P) уже перевели устройство в состояние риска, а для других из этих событий ($M_{ОП} - M_P$) отсутствует возможность устранения или блокирования обнаруженного отказа в течение остаточного времени до наступления опасного события.

На основании приведенной исходной информации оценивается безусловная вероятность своевременного обнаружения нештатными средствами скрытого отказа объекта при известном (заданном) допустимом значении $\tau_{ДОП}$ существования скрытого отказа:

$$\hat{\beta} = \frac{M - M_{ОП}}{M} P\{\hat{\tau}_{обн} \le \tau_{ДОП}\}, \text{ где } \hat{\tau}_{обн} = \frac{\sum_{i=1}^{M-M_P} (t_i^* - \hat{t}_i)}{M - M_P}. \quad (5.2)$$

Вероятность возникновения риска больше допустимого уровня $R_{ДОП}$ в общем виде определяется следующим выражением

$$\nu = P\{R > R_{ДОП}\}.$$

где R – остаточный уровень риска.

Таким образом, вероятность ν есть вероятность того, что опасный отказ объекта, во-первых, попадет в соответствующую по частоте возникновения категорию матрицы рисков и, во-вторых, попадет по уровню тяжести последствия опасного события в клетку матрицы с недопустимым или нежелательным и неприемлемым значением риска.

Вероятность возникновения приемлемого риска есть

$$\bar{\nu} = P\{R \le R_{ДОП}\} = 1 - \nu.$$

В соответствии с принятыми предпосылками в случае возникновения с вероятностью $\bar{\nu}$ приемлемого риска объект переводится в состояние 2 восстановления проявившегося отказа.

Законы распределения отказов, восстановлений, длительностей обнаружения отказов нештатными средствами в данной задаче примем на основании следующих соображений. Отказ в функционировании объекта КВС может произойти, как отмечалось выше, вследствие причин функционального и/или структурного характера. Закон распределения отказов цифровых устройств, как правило, экспоненциальный с интенсивностью отказов λ. Вероятность отсутствия ошибки при однократном выполнении процесса равна P (см. главу II.2). Вероятность отсутствия ошибки в выполнении информационного процесса в течение заданного времен при условии простейшего потока заявок равна

$$P(t) = \exp(-\Omega \cdot G \cdot g_{ФТ} \cdot t),$$

где Ω – интенсивность потока заявок, $G = 1 - P$, $g_{ФТ}$ – вероятность трансформации ошибки в функциональный отказ.

Следовательно, вероятность возникновения отказа в функционировании объекта может быть оценена как

$$F(t) = 1 - \exp(-(\Omega \cdot G \cdot g_{\Phi T} t) \cdot \exp(-\lambda \cdot t) = 1 - \exp[-(\Omega \cdot G \cdot g_{\Phi T} + \lambda) \cdot t]$$

Случайное время устранения отказа в функционировании объекта КВС можно во многих случаях принять распределенным по закону Эрланга порядка 2 с интенсивностью восстановления функционирования объекта μ и функцией распределения

$$B_1(t) = 1 - (1 + \mu \cdot t) \exp[-\mu \cdot t]$$

Время обнаружения отказа в функционировании объекта нештатными средствами и время устранения ущерба (модификации объекта в случае неприемлемости ущерба) – это случайные величины. Относительно них можно с уверенностью полагать, что параметры распределений этих случайных величин не могут быть постоянными. Вследствие неопределенности в прогнозировании времени обнаружения отказа в функционировании объекта нештатными средствами и времени устранения ущерба (модификации объекта в случае неприемлемости ущерба) примем как случайные времена, распределенные по закону Эрланга порядка 2 (по закону Релея) с интенсивностью γ и ϑ соответственно:

$$B_2(t) = 1 - (1 + \gamma \cdot t) \exp[-\gamma \cdot t]$$

$$W(t) = 1 - (1 + \vartheta \cdot t) \exp(-\vartheta \cdot t).$$

Определим математические ожидания времени пребывания устройства в каждом состоянии графа рис. II.5.5.

$$T_1 = \int_0^{\infty} [1 - F(t)]dt = \int_0^{\infty} \exp[-(\Omega \cdot G \cdot g_{ФТ} + \lambda)t]dt = \frac{1}{\Omega \cdot G \cdot g_{ФТ} + \lambda}.$$

$$T_2 = \int_0^{\infty} [1 - B_1(t)]dt = \int_0^{\infty} (1 + \mu \cdot t) \exp(-\mu \cdot t) \cdot dt = \frac{2}{\mu}$$

Математическое ожидание времени пребывания устройства в состоянии «3» обнаружения отказа нештатными средствами ограничено допустимым временем обнаружения отказа $\tau_{ДОП}$. Следовательно,

$$T_3 = \int_0^{\tau_{ДОП}} (1 + \gamma \cdot t) \exp(-\gamma \cdot t)dt = \frac{2 - \exp(-\gamma \cdot \tau_{ДОП}) \cdot (2 + \gamma \cdot \tau_{ДОП})}{\gamma}. \quad (5.3)$$

В работе [66] В. Швиром для систем, связанных с безопасностью (критически важных систем), установлено условие $\lambda\tau_{ДОП} \le 10^{-3}$, позволяющее обосновать требование к допустимому времени существования скрытого отказа, т.е. к параметру $\tau_{ДОП}$, если известна интенсивность отказов устройства. При реальных значениях параметра $\lambda \le 10^{-2} \frac{1}{ч}$ значение допустимого времени обнаружения отказа находится на уровне 0.1ч. Отсюда следует, что в выражении (5.3) выполняется условие

$$\gamma \cdot \tau_{ДОП} << 1.$$

Данное условие позволяет разложить экспоненту в ряд и с погрешностью, не превышающей первого порядка малости, ограничиться первыми двумя ее членами, т.е.

$$\exp(-\gamma \cdot \tau_{ДОП}) \approx 1 - \gamma \cdot \tau_{ДОП}.$$

Следовательно, выражение (5.3) преобразуется к виду

$$T_З = \frac{2 - \exp(-\gamma \cdot \tau_{ДОП})(2 + \gamma \cdot \tau_{ДОП})}{\gamma} \approx \frac{2(1 - 1 + \gamma \cdot \tau_{ДОП})}{\gamma} = 2\tau_{ДОП}.$$

Представленные выше сведения позволяют конкретизировать формульное выражение вероятности своевременного обнаружения скрытого отказа нештатными средствами (5.2). В этой формуле интенсивность обнаружения отказа может быть получена путем статистической оценки

$$\hat{\gamma} \approx \frac{2}{\hat{\tau}_{обн}} = \frac{2(M - M_P)}{\sum_{i=1}^{M-M_P} (t_i^* - \hat{t}_i)},$$

а условная вероятность своевременного обнаружения скрытого отказа нештатными средствами может быть определена по формуле

$$P\{\hat{\tau}_{обн} \le \tau_{ДОП}\} = \int_0^{\tau_{ДОП}} \hat{\gamma}^2 t \exp(-\hat{\gamma} \cdot t) dt = 1 - \exp(-\hat{\gamma} \cdot \tau_{ДОП}) \cdot (1 + \hat{\gamma} \cdot \tau_{ДОП}).$$

Ранее уже была показана возможность и допустимость применения приближения $\exp(-\hat{\gamma} \cdot \tau_{ДОП}) \approx 1 - \hat{\gamma} \cdot \tau_{ДОП}$. Это позволяет привести формулу оценки рассматриваемой вероятности к виду

$$P\{\hat{\tau}_{обн} \le \tau_{ДОП}\} = 1 - \exp(-\hat{\gamma} \cdot \tau_{ДОП}) \cdot (1 + \hat{\gamma} \cdot \tau_{ДОП}) \approx \hat{\gamma} \cdot \tau_{ДОП} \cdot (2 - \hat{\gamma} \cdot \tau_{ДОП})$$

Следовательно, оценка безусловной вероятности своевременного обнаружения скрытых отказов нештатными средствами вычисляется по формуле

$$\hat{\beta} = \frac{(M - M_{ОП})}{M} \cdot \hat{\gamma} \cdot \tau_{ДОП} \cdot (2 - \hat{\gamma} \cdot \tau_{ДОП})$$

Математическое ожидание времени устранения ущерба T_4 может быть определено из следующих соображений. Во – первых, если ущерб меньше допустимого уровня, то после его проявления объект сразу передается на восстановление в состояние «2». Во – вторых, чем меньше вероятность ν того, что ущерб превысил допустимый уровень, тем меньше уровень тяжести последствий от опасного отказа и тем меньше длительность устранения ущерба, а, следовательно, модификации объекта, т.е. может быть принято, что

$$T_4 = \nu \int_0^{\infty} (1 + \vartheta t) \exp(-\vartheta \cdot t) dt = \frac{2\nu}{\vartheta}, \text{ где } 0 \le \nu \le 1.$$

Итак, математические ожидания времени пребывания объекта КВС в соответствующих состояниях рассчитываются по формулам:

$$T_1 = \frac{1}{\Omega G + \lambda}; T_2 = \frac{2}{\mu}; T_3 = 2\tau_{ДОП}; T_4 = \frac{2\nu}{\vartheta}.$$

Наличие усечений времени пребывания устройства в состоянии «3» и линейный характер изменения параметра времени в состоянии «4» исключают возможность применения аппарата

Марковских случайных процессов при исследовании устройства, связанного с безопасностью. Вместе с тем, из графовой модели рис. II.5.5 следует, что модификации отдельных параметров исходной математической модели не повлияли на вложенную в нее Марковскую цепь. Все это позволяет применить для математического описания безопасности устройства аппарат полумарковских случайных процессов и воспользоваться графовым методом, изложенным в п. I.4.4.2.

Критерии отказов

Все множество состояний объекта в соответствии с рис. II.5.5 обозначается как $S = \{1, 2, 3, 4\}$. Работоспособно только состояние «1», т.е. множество работоспособных состояний $S_P = \{1\}$, а множество неработоспособных состояний $\overline{S_P} = \{2, 3, 4\}$, где $S = S_P \bigcup \overline{S}_P$ и $S_P \bigcap \overline{S}_P = 0$.

Опасное событие объекта будет в том случае, если возникший отказ не обнаружен как штатными, так и нештатными средствами. А это имеет место при переходе устройства в состояние «4». Тогда множество неопасных состояний объекта $\overline{S}_{ОП} = \{1, 2, 3\}$, а множество опасных – $S_{ОП} = \{4\}$, где $S = S_{ОП} \bigcup \overline{S}_{ОП}$ и $S_{ОП} \bigcap \overline{S}_{ОП} = 0$.

В целях применения топологического метода п.I.4.4.2 для решения поставленной задачи приведем ряд понятий и формул.

Топологические понятия:

– ***путь*** – это цепь последовательно соединенных однонаправленных дуг с началом в вершине «*i*» и окончанием в вершине «*j*», вес пути $l_k^{ij} = \prod_{i,r,j \in S} p_{ir} p_{rj}$. В данной задаче существуют следующие пути от вершины «1» (рис. II.5.5):

• от вершины «1» к вершине «2»:

1-2 или 1-3-2 или 1-3-4-2. Их веса: $l_1^{12} = p_{12} = \alpha$;

$$l_2^{12} = \prod_{1,3,2 \in S} p_{13} p_{32} = \overline{\alpha}\beta \; ; \; l_3^{12} = \prod_{1,3,4,2 \in S} p_{13} p_{34} p_{42} = \overline{\alpha}\overline{\beta}\overline{\nu} \; ;$$

- от вершины «1» к вершине «3»: 1-3, $l_1^{13} = p_{13} = \overline{\alpha}$;

- от вершины «1» к вершине «4»: 1-3-4, $l_1^{14} = \prod_{1,3,4 \in S} p_{13} p_{34} = \overline{\alpha}\overline{\beta}$.

– ***замкнутый контур*** – цепь последовательно соединенных однонаправленных дуг, причем выход конечной вершины цепи соединен последней дугой с начальной вершиной цепи, вес контура $C_j = \prod_{i,j \in S} p_{ij} p_{ji}$. В данной задаче в множестве неопасных состояний $\overline{S}_{ОП} = \{1,2,3\}$ есть два контура:

- 1-2-1, $C_1 = \prod_{1,2 \in S} p_{12} p_{21} = \alpha \cdot 1 = \alpha$, поскольку $p_{21} = 1$;

- 1-3-2-1, $C_2 = \prod_{1,2,3 \in S} p_{13} p_{32} p_{21} = \overline{\alpha}\overline{\beta} \cdot 1 = \overline{\alpha}\overline{\beta}$.

- 1-3-4-1, $C_3 = \prod_{1,2,3 \in S} p_{13} p_{34} p_{41} = \overline{\alpha} \cdot \overline{\beta} \cdot \nu$

– ***разложение графа*** – это часть графа, не содержащая выделенных вершин и связанных с ними дуг, вес разложения

$$\Delta G = 1 - \sum_{(j)} C_j + \sum_{(rj)} C_r C_j - \ldots ,$$

где в произведениях весов учитываются только те контуры, которые не имеют общих вершин.

Вес разложения графа рис. II.5.5 в области неопасных состояний $\overline{S}_{ОП} = \{1,2,3\}$ равен

$$\Delta G_{\overline{S}_{ОП}} = 1 - C_1 - C_2 = 1 - \alpha - \overline{\alpha}\beta = \overline{\alpha} - \overline{\alpha}\beta = \overline{\alpha}\overline{\beta}.$$

Вес разложения графа ΔG^1 без вершины «1» в данной задаче определяется как $\Delta G^1 = 1 - 0 - 0 = 1$, поскольку при отсутствии вершины «1» отсутствуют также и пути 1-2 и 2-1 (следовательно, их веса равны 0). Вес разложения графа ΔG^2

$$\Delta G^2 = 1 - C_3 = 1 - \overline{\alpha} \cdot \overline{\beta} \cdot \nu .$$

Вес разложения графа ΔG^3

$$\Delta G^3 = 1 - C_1 = 1 - \alpha = \overline{\alpha} .$$

Вес разложения графа ΔG^4

$$\Delta G^4 = 1 - C_1 - C_2 = 1 - \alpha - \overline{\alpha}\beta = \overline{\alpha} - \overline{\alpha}\beta = \overline{\alpha}\overline{\beta}.$$

В *соответствии с международным стандартом* [59] *основными показателями безопасности* объекта КВС являются:

– *Среднее время между опасными отказами* (определяется по топологической формуле (4.21) раздела I)

$$MTBF(H) = T_{\text{СОП}} = \frac{\sum\limits_{i \in S_P} \Delta G^i \cdot T_i}{\sum\limits_{i \in S_+} \Delta G^i \sum\limits_{j \in \overline{S_P}} p_{ij}} = \frac{T_1 \cdot \Delta G^1 + T_2 \cdot \Delta G^2 + T_3 \cdot \Delta G^3}{\Delta G^3 \cdot p_{34}}$$

где S_+ – подмножество граничных неопасных состояний $(S_+ \in \overline{S}_{ОП})$, S_- – подмножество граничных опасных состояний $(S_- \in S_{ОП})$.

В данной задаче $S_+ = \{3\}$, $S_- = \{4\}$.

$$T_{СОП} = \frac{\dfrac{1}{\Omega \cdot G \cdot g_{ФТ} + \lambda} + \dfrac{2}{\mu}(1 - \overline{\alpha} \cdot \overline{\beta} \cdot \nu) + 2\tau_{ДОП} \cdot \overline{\alpha}}{\overline{\alpha} \cdot \overline{\beta}} =$$

$$= \frac{\mu + 2(\Omega \cdot G \cdot g_{ФТ} + \lambda) \cdot (1 - \nu\overline{\alpha} \cdot \overline{\beta} + \tau_{ДОП} \cdot \mu \cdot \overline{\alpha})}{\overline{\alpha} \cdot \overline{\beta} \cdot \mu \cdot (\Omega \cdot G \cdot g_{ФТ} + \lambda)} \approx \qquad (5.4)$$

$$\approx \frac{\mu + 2(\Omega \cdot G \cdot g_{ФТ} + \lambda) \cdot (1 + \tau_{ДОП} \cdot \mu \cdot \overline{\alpha})}{\overline{\alpha} \cdot \overline{\beta} \cdot \mu \cdot (\Omega \cdot G \cdot g_{ФТ} + \lambda)}$$

поскольку $\nu \cdot \overline{\alpha} \cdot \overline{\beta} << 1$.

Формула (5.4) может быть существенно упрощена, если принять во внимание, что

$$2(\Omega \cdot G \cdot g_{ФТ} + \lambda) \cdot (1 + \tau_{ДОП} \cdot \mu \cdot \overline{\alpha}) << \mu .$$

С погрешностью, не превышающей первого порядка малости, справедливо следующее выражение

$$T_{СОП} = \frac{1}{\overline{\alpha} \cdot \overline{\beta} \cdot (\Omega \cdot G \cdot g_{ФТ} + \lambda)} .$$

Из этой формулы следует очевидный вывод: чем меньше интенсивности функциональных и структурных отказов объектов КВС и чем выше эффективность обнаружения отказов штатными и нештатными средствами, тем реже будут возникать опасные отказы КВС.

– *Время возврата к безопасному состоянию* (определяется по топологической формуле работы (4.22) раздела I.)

$$TTRS = \tau_{БОП} = \frac{\sum\limits_{i \in \overline{S_P}} \Delta G^i \cdot T_i}{\sum\limits_{i \in S_-} \Delta G^i \sum\limits_{j \in S_P} p_{ij}} = \frac{\Delta G^4 \cdot T_4}{\Delta G^4 (p_{41} + p_{42})}$$

Применительно к данному показателю $S_+ = \{1,2\}, S_- = \{4\}$

$$\tau_{БОП} = \frac{\frac{2\nu}{\vartheta}}{\overline{(\nu + \nu)}} = \frac{2\nu}{\vartheta}. \qquad (5.5)$$

– *Вероятность опасных отказов*. Поток опасных отказов – это поток редких событий. Практически это разреженный случайным образом поток отказов объекта КВС. В соответствии с теоремой И.Б. Погожева [68] разреженный случайным образом произвольный поток событий становится простейшим потоком. При этом случайное время до опасного отказа может быть представлено моделью экспоненциального распределения

$$F_S(t) = 1 - \exp(-\frac{t}{T_{ОП}}),$$

где $F_S(t)$ – вероятность отказов, связанных с безопасностью (мы их полагаем опасными), $T_{ОП}$ – среднее время до опасного отказа.

В соответствии с формулой (4.20) части I при начальном состоянии *i*=1

$$T_{ОП} = \frac{T_1 \Delta G^1_{S_{ОП}} + \sum\limits_{j \in \overline{S}_{ОП}} \sum\limits_{(k)} l_k^{1j} \Delta G_k^j T_j}{\Delta G_{S_{ОП}}},$$

где $\Delta G^1_{S_{ОП}}$ – вес разложения графа без множества опасных состояний и вершины «1», $\Delta G_{S_{ОП}}$ – вес разложения графа без множества опасных состояний.

В данной задаче

$$\Delta G^1_{S_{ОП}} = 1;\ \Delta G_{S_{ОП}} = 1 - C_1 - C_2 = 1 - \alpha - \overline{\alpha}\beta = \overline{\alpha\beta}\,.$$

$$T_{ОП} = \frac{\mu + 2(\Omega \cdot G \cdot g_{ФТ} + \lambda)(\alpha + \overline{\alpha} \cdot \tau_{ДОП} \cdot \mu)}{\overline{\alpha} \cdot \overline{\beta} \cdot \mu \cdot (\Omega \cdot G \cdot g_{ФТ} + \lambda)} = \tag{5.6}$$

$$= \frac{1}{\overline{\beta}} \cdot [\frac{1}{\overline{\alpha}(\Omega \cdot G \cdot g_{ФТ} + \lambda)} + \frac{2\alpha}{\overline{\alpha}\mu} + 2\tau_{ДОП}]$$

Заметим, что в данном выражении $\frac{1}{\overline{\alpha}(\Omega \cdot G \cdot g_{ФТ} + \lambda)} + \frac{1}{\overline{\alpha}\mu} >> \tau_{ДОП}$. Это позволяет с погрешностью, не превышающей первого порядка малости, принять, что

$$T_{ОП} \approx \frac{\mu + 2\alpha(\Omega \cdot G \cdot g_{ФТ} + \lambda)}{\overline{\alpha}\overline{\beta}}.$$

Таким образом, вероятность опасных отказов может рассчитываться с допустимой погрешностью по следующей формуле:

$$F_S(t) \approx 1 - \exp[-\frac{t\overline{\alpha}\overline{\beta}}{\mu + 2\alpha(\Omega \cdot G \cdot g_{ФТ} + \lambda)}] \approx \frac{t\overline{\alpha}\overline{\beta}}{\mu + 2\alpha(\Omega \cdot G \cdot g_{ФТ} + \lambda)}. \tag{5.7}$$

Возможность ограничения первыми двумя членами разложения экспоненциальной функции в ряд обусловлена тем, что степень экспоненциальной функции много меньше единицы.

Вероятность безопасного функционирования:

$$S_S(t) = 1 - F_S(t) \approx 1 - \frac{t\overline{\alpha}\overline{\beta}}{\mu + 2\alpha(\Omega \cdot G \cdot g_{\Phi T} + \lambda)}. \qquad (5.8)$$

– Дополнительно к рекомендованным в стандарте [60] показателям безопасности в практике российских железных дорог применяется *коэффициент безопасности*. В целях его определения в данной задаче найдем с помощью топологической формулы (4.20) [1] стационарные вероятности π_i пребывания в состояниях графа, показанного на рис. II.5.5.

$$\pi_i = \frac{\Delta G^i T_i}{\sum_{i \in S} \Delta G^i T_i},$$

Таким образом,

$$\pi_1 = \frac{T_1}{T_1 + (1 - \overline{\alpha} \cdot \overline{\beta} \cdot \nu) T_2 + \overline{\alpha} \cdot T_3 + \overline{\alpha} \cdot \overline{\beta} \cdot T_4},$$

$$\pi_2 = \frac{(1 - \overline{\alpha} \cdot \overline{\beta} \cdot \nu) T_2}{T_1 + (1 - \overline{\alpha} \cdot \overline{\beta} \cdot \nu) T_2 + \overline{\alpha} \cdot T_3 + \overline{\alpha} \cdot \overline{\beta} \cdot T_4},$$

$$\pi_3 = \frac{\overline{\alpha} \cdot T_3}{T_1 + (1 - \overline{\alpha} \cdot \overline{\beta} \cdot \nu) T_2 + \overline{\alpha} \cdot T_3 + \overline{\alpha} \cdot \overline{\beta} \cdot T_4},$$

$$\pi_4 = \frac{\overline{\alpha} \cdot \overline{\beta} \cdot T_4}{T_1 + (1 - \overline{\alpha} \cdot \overline{\beta} \cdot \nu) T_2 + \overline{\alpha} \cdot T_3 + \overline{\alpha} \cdot \overline{\beta} \cdot T_4}, .$$

Коэффициент безопасности определяется в виде суммы стационарных вероятностей пребывания устройства в множестве неопасных состояний $\overline{S}_{ОП} = \{1,2,3\}$.

$$K_{Б} = \frac{T_1 + (1-\overline{\alpha}\cdot\overline{\beta}\cdot\nu)T_2 + \overline{\alpha}T_3}{T_1 + (1-\overline{\alpha}\cdot\overline{\beta}\cdot\nu)T_2 + \overline{\alpha}T_3 + \overline{\alpha}\overline{\beta}T_4} = \tag{5.9}$$

$$= \frac{\dfrac{1}{\Omega G\cdot g_{ФТ} + \lambda} + (1-\overline{\alpha}\cdot\overline{\beta}\cdot\nu)\dfrac{2}{\mu} + \overline{\alpha}\cdot 2\tau_{ДОП}}{\dfrac{1}{\Omega G\cdot g_{ФТ} + \lambda} + (1-\overline{\alpha}\overline{\beta}\nu)\dfrac{2}{\mu} + \overline{\alpha}\cdot 2\tau_{ДОП} + \overline{\alpha}\cdot\overline{\beta}\cdot\dfrac{2\nu}{\vartheta}}$$

II.5.4.3. Анализ основных результатов.

Проанализируем полученные формульные выражения (5.1)-(5.9).

Среднее время между опасными отказами ($MTBF(H) = T_{СОП}$), как уже отмечалось, зависит главным образом от эффективности как штатных, так и нештатных средств контроля. Рассмотрим граничные условия:

– пусть эффективность контроля близка к идеальной ($\overline{\alpha} \to 0$, или $\overline{\beta} \to 0$). В этом случае $MTBF(H) = T_{СОП} \to \infty$. Этот результат означает, что для коренного повышения безопасности объекта или системы в целом следует сосредоточить усилия на своевременном и эффективном обнаружении отказов. При этом следует учитывать, что данный вывод справедлив только тогда, когда применительно к данному объекту КВС созданы все условия и имеются достаточные ресурсы для оперативного устранения или блокирования обнаруженного опасного события. Это условие в данной работе предполагается, а определение необходимых ресурсов и правил устранения или блокирования

обнаруженного опасного события является предметом самостоятельного исследования;

– пусть эффективность средств обнаружения отказов низка ($\bar{\alpha}, \bar{\beta} \to 1$). В этом случае формула (5.4) преобразуется к виду $T_{соп} \approx \frac{1}{\Omega \cdot G \cdot g_{фт} + \lambda}$. Этот результат означает, что при низкой эффективности средств обнаружения отказов уровень безопасности устройства приближается к уровню его безотказности как в части функциональных, так и в отношении структурных отказов. Этот отрицательный эффект не устраняется и при вводе в устройство достаточных ресурсов для оперативного устранения или блокирования опасных событий. Основные усилия нужно сосредоточить на повышении эффективности штатных средств обнаружения отказов и повышении оперативности использования имеющейся информации от нештатных средств обнаружения скрытых отказов.

Время возврата к безопасному состоянию равно времени устранения ущерба (времени модификации устройства). Чем меньше это время, тем быстрее устройство возвратится в исходное безопасное состояние. Это справедливо также и тогда, когда уменьшается вероятность ν того, что величина риска превышает допустимый уровень.

Вероятность отказов, связанных с безопасностью, практически линейно зависит от времени работы устройства, вероятности необнаружения отказа всеми доступными средствами контроля и обратно пропорциональна сумме показателей безотказности и ремонтопригодности. Причем, если предположить, что время восстановления устройства сколь угодно велико, то вероятность опасных отказов стремится к нулю. Здесь нет ничего удивительного – если устройство не функционирует (выключено, находится на техническом обслуживании или в ремонте), то в нем не возникают не только критические отказы, но и ка-

кие-либо отказы вообще. Эти соображения имеют зеркальный характер относительно *вероятности безопасного функционирования* – чем больше среднее время между отказами, чем длительнее восстановление, тем меньше вероятность безопасного функционирования.

II.5.5. Требования к функциональной надежности критически важных информационных систем (общие положения)

Высокий уровень функциональной надежности критически важных информационных систем (КВС) может быть достигнут на основе применения широкого спектра мер, которые охватывают как введение в системы различных видов избыточности, так и применения прогрессивных технологий организации работ по обеспечению функциональной надежности на различных этапах жизненного цикла КВС. Поэтому к функциональной надежности КВС должны предъявляться две группы требований – количественные (нормативные) и качественные (технологические). Количественные требования предназначены для оценки надежности выполнения предусмотренных в КВС информационных процессов, а также для оценки эффективности мер по повышению функциональной надежности на основе примененных видов функциональной, алгоритмической, информационной или временной избыточности. Количественные требования предъявляются на основе оценки рисков. Процедуры управления рисками в КВС показаны на рис. II.5.6. Предварительно устанавливается область применения КВС и в зависимости от этого задается критерий риска. Он представляет тот допустимый порог риска, в пределах которого функционирование критически важной информационной системы с определенными условиями приемлемо. В противном случае

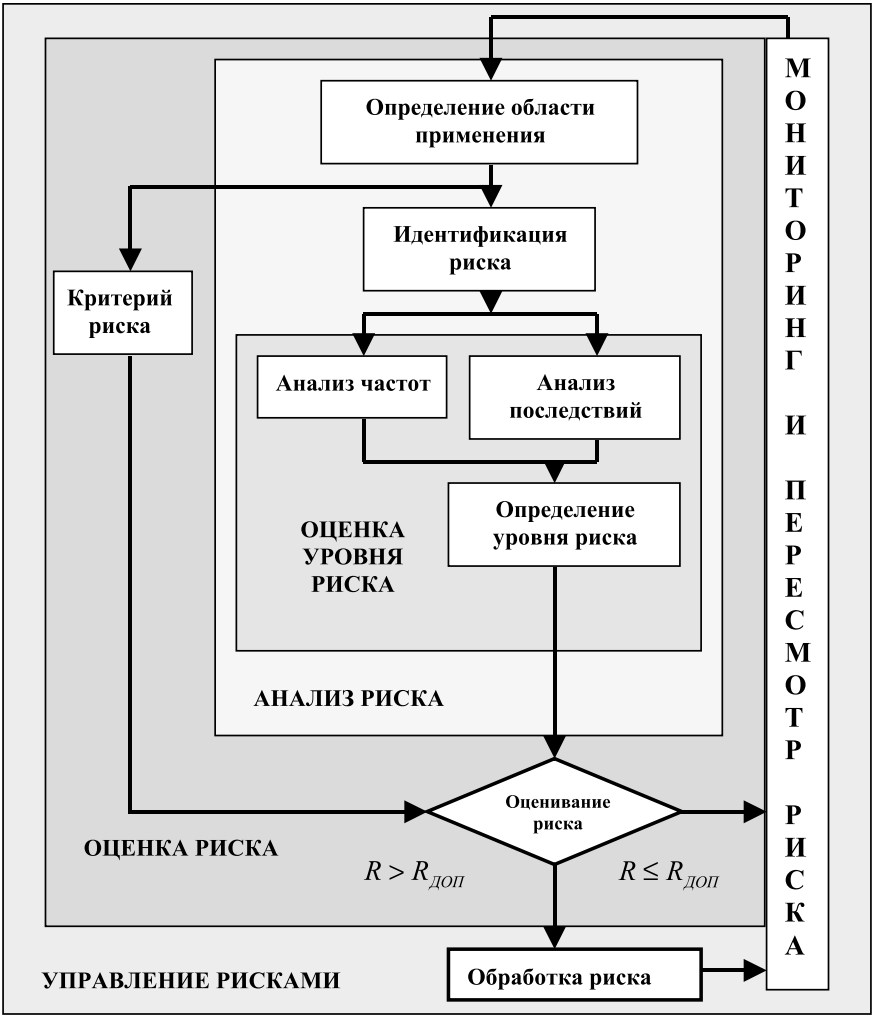


Рис. II.5.6. Управление рисками в критически важной информационной системе

уровень риска ***недопустим.*** Немедленно должны быть предприняты краткосрочные действия по снижению риска. Указанные условия приемлемости риска есть не что иное, как меры отклонений от величины допустимого риска $R_{ДОП}$. Так, при условии $0.1 \cdot R_{ДОП} \le R < R_{ДОП}$, уровень риска хотя и допустим,

но ***нежелателен.*** В этом случае риск должен быть уменьшен. Краткосрочные действия по уменьшению риска должны быть предприняты сразу же, как только это практически возможно. На железнодорожном транспорте относительно КВС приняты следующие условия**:** если мера отклонения риска от допустимого уровня составляет $0.01 \cdot R_{ДОП} \le R < 0.1 \cdot R_{ДОП}$, то принимается решение о ***допустимости*** риска. Для уменьшения риска могут быть предприняты соответствующие действия. При условии $R < 0.01 \cdot R_{ДОП}$ риск может ***не приниматься в расчет***. В этом случае риск можно считать приемлемым; никакие дополнительные действия управления риском не требуются.

Оценка величины риска – определение меры риска анализируемой опасности для здоровья человека, окружающей среды и материальных ценностей в ситуациях, связанных с реализацией опасности. Оценка величины риска является обязательной частью анализа риска и включает анализ частоты, анализ последствий и их сочетаний, а также определение уровня риска.

Цель анализа частот состоит в определении частоты возникновения каждого из нежелательных событий или сценариев аварий, выявленных на стадии идентификации риска.

Для определения частоты возникновения события обычно используются следующие основные подходы:

– прямая статистическая обработка данных и прогнозирование на этой основе частоты, с которой данное событие будет возникать в будущем;

– косвенное использование обычно неполных статистических данных или данных экспертной оценки в сочетании с применением известных методов прогнозирования частот возникновения событий таких как: метод анализа видов и последствий отказов (FMEA), метод анализа видов, последствий и критичности отказов (FMECA), метод анализа дерева неисправностей (FTA), метод анализа дерева событий (ETA), булевы методы,

Марковские методы. Эффективно применение в этих целях полумарковских, логико-вероятностных методов, имитационного моделирования, сетей Петри.

Полученные оценки частот возникновения событий соотносят с заданными уровнями частот. Количество типовых уровней частот и их характеристики представлены в таблице II.5.6.

Таблица II.5.6. Уровни частот возникновения событий

Уровень частоты	**Описание**
Частое	Вероятность частого возникновения. Постоянное наличие опасности.
Вероятное	Неоднократное возникновение. Ожидается частое возникновение опасного события.
Случайное	Вероятность неоднократного возникновения. Ожидается неоднократное возникновение опасного события.
Редкое	Вероятность того, что событие будет иногда возникать на протяжении жизненного цикла объекта. Обоснованное ожидание возникновения опасного события.
Крайне редкое	Вероятность возникновения маловероятна, но возможна. Можно предположить, что опасное событие может возникнуть в исключительном случае.
Маловероятное	Вероятность возникновения крайне маловероятна. Можно предположить, что опасное событие не возникнет.

Анализ последствий (ущербов) предусматривает оценку результатов воздействия нежелательного события в работе объекта управления на людей, имущество и окружающую среду. Анализ последствий может быть выполнен как в виде простого описания результатов, так и в виде детального количественного модели-

рования. Последствия возникновения опасных отказов в работе КВС соотносят с заданными уровнями тяжести последствий.

Центральная задача управления рисками в работе критически важных информационных систем – это оценивание и обработка риска.

В простейшем случае величина риска определяется как произведение величины нежелательного события (величины ущерба) на возможность (вероятность, частоту) его наступления:

$$R = p \cdot A$$

где: R – риск; A – величина нежелательного события (ущерба); p – вероятность его наступления.

Диапазон A достаточно широк – от экономических до этических последствий. Риск может быть явно связан с факторами, не поддающимися учету.

Оценка риска должна исходить из того, что необходимо установить экономический эквивалент угрозы. Он соответствует затратам, которые при данных условиях можно позволить, чтобы предотвратить или уменьшить угрозу.

Типы задач с учетом риска:

– задачи с минимизацией риска;

– задачи с оптимизацией риска.

Пусть $S = \{S_1, S_2, ..., S_n, K_1, K_2, ..., K_m\}$ – множество всех неблагоприятных событий и их сочетаний K. Вероятность возникновения опасного функционального отказа КВС, который приводит к *j*-му неблагоприятному событию или их сочетанию ($j = 1, 2, ..., n, ..., n+m$) в работе объекта управления, обозначим как p_j. Каждое сочетание неблагоприятных событий есть функция как от вероятности возникновения опасного отказа p_j, так и от варианта решения по парированию неблагоприятных событий E_i Тогда *i-й* риск есть средневзвешенная по вероятнос-

ти опасного функционального отказа КВС ожидаемая величина ущерба при условии принятия решения E_i

$$R_i = \sum_j p_j A_{ij}$$

Вариант решения E_i без учета возможности неблагоприятных последствий будет иметь полезность e_i. Тогда соответствующая варианту величина G_i представляет собой суммарный эффект решения:

$$G_i = e_i - R_i$$

Вектор рациональных вариантов решения:

$$\vec{E} = \{E_i : G_i > 0 \ \}$$

Вариант решения E_i^* – оптимальный, если суммарный эффект при данном варианте решения максимальный

$$G_i^* = \max_{E_i} G_i .$$

При решении конкретной задачи множество допустимых вариантов решения может быть дополнительно ограничено пределами риска.

Выбор решений в условиях неопределенности включает:

– построение матрицы эффектов и ущерба (как инструмент предварительной оценки риска) и матрицы риска (как инструмент окончательной оценки риска);

– количественную оценку вариантов.

Каждая строка матрицы эффектов и ущерба соответствует одному из вариантов намеченных альтернативных реше-

ний E_i, а каждый столбец – одной из возможных ситуаций S_j, которые могут возникнуть при разных значениях отсутствующей информации об условиях решения проблемы или об ожидаемых результатах. Для каждой пары (B_i, S_j) можно определить соответствующие значения целевой функции φ_{ij}. В общем случае эти значения могут быть как положительными, так и отрицательными, т.е. количественно оценивать эффект или ущерб при сочетании i-го варианта решения и j-ой ситуации.

В нижнюю строку таблицы вынесены наибольшие для каждого столбца (т.е. для S_j) эффекты $(\varphi_j)_{max}$.

Количественной оценкой риска для каждого *i*-го решения при *j*-ой ситуации принято считать разницу между максимально возможным для этой ситуации эффектом и фактическим:

$$R_{ij} = (\varphi_i)_{\max} - \varphi_{ij}.$$

Рассмотрим стратегию наибольшего гарантированного эффекта. В каждой строке матрицы эффектов выбирается минимальный эффект $(\varphi_i)_{\min}$. Лучшим считается вариант решения, для которого минимальный (гарантированный) выигрыш окажется наибольшим. Критерий, реализующий такой выбор, называется критерием максимального эффекта (выигрыша) или критерием Вальда:

$$R_{ij} = \max_i \min_j \varphi_{ij}$$

Стратегия наименьшего возможного риска также ориентируется на худшую ситуацию, но за такую считает не ту, которая дает наименьший эффект, а ту, которая сопряжена с наибольшим риском. В таких случаях по каждой строке матрицы риска выбирается $(R_i)_{\max}$, а лучшим считается вариант, при котором

этот максимальный риск оказывается наименьшим. Критерий, реализующий такой выбор, называется критерием минимального риска или критерием Сэвиджа.

$$R_S = \max_i \min_j R_{ij}.$$

Смешанная стратегия предусматривает сочетание пессимизма (осторожности) и оптимизма (склонности к значительному риску), в определенно заданной пропорции. Эту стратегию реализует критерий Гурвица:

$$R_\Gamma = \max_i [\alpha \min_j \varphi_{ij} + (1-\alpha) \max_j \varphi_{ij}]$$

При этом критерии приемлемого риска могут:

– задаваться нормативно-правовой документацией;

– определяться при планировании анализа риска;

– определяться в процессе получения результатов анализа риска.

Основными требованиями к выбору критерия приемлемого риска при проведении анализа риска являются его обоснованность и определенность. В общем случае приемлемый риск сочетает в себе технические, экологические, социальные аспекты и представляет некоторый компромисс между приемлемым уровнем безопасности и экономическими возможностями его достижения. Следовательно при снижении индивидуального, технического или экологического риска следует учитывать значительность затрат на реализацию этого снижения и сопутствующее возрастание социального риска.

Представление результатов оценивания риска может быть осуществлено с помощью матрицы рисков. Матрица рисков – инструмент, позволяющий ранжировать и отражать риски путем определения уровней частот и тяжести последствий.

Таблица II.5.7 – Типичный пример оценки и приемлемости риска

Частота возникновения опасного события*	**Оценка степени риска**			
Частое возникновение	Нежелательный			Недопустимый
Возможное возникновение	Допустимый	Нежелательный	Недопустимый	Недопустимый
Редкое возникновение	Допустимый	Нежелательный	Нежелательный	Недопустимый
Маловероятное возникновение	**Не принимаемый в расчет**	Допустимый	Нежелательный	Нежелательный
Невозможное возникновение	**Не принимаемый в расчет**	**Не принимаемый в расчет**	Допустимый	Допустимый
Невероятное возникновение	**Не принимаемый в расчет**	**Не принимаемый в расчет**	**Не принимаемый в расчет**	**Не принимаемый в расчет**
	Незначительный	**Значительный**	**Критический**	**Катастрофический**
	Уровни тяжести последствия опасного события			
* Масштабирование частоты возникновения опасных событий зависит от рассматриваемого применения				

Оценка степени риска	Уменьшение/контроль риска
Недопустимый	Должен исключаться
Нежелательный	Может быть приемлемым, когда уменьшение риска невыполнимо и при согласии эксплуатирующей организации
Допустимый	Приемлем при соответствующем контроле и при согласии эксплуатирующей организации
Не принимаемый в расчет	Приемлем без согласия эксплуатирующей организации

Матрица рисков строится следующим образом:

– по вертикальной оси отсчитываются вероятности (частоты) возникновения события, представленные в виде шкалы в соответствии с принятыми уровнями частот;

– по горизонтальной оси отсчитываются размеры последствий возникновения события, представленные в виде шкалы в соответствии с принятыми уровнями тяжести последствий.

Типичная форма матрицы рисков, содержащей шесть уровней частот и 4 уровня тяжести последствий, в которой уровень риска ранжируется по четырем категориям, представлена в табл. II.5.7.

По результатам оценивания риска принимается решение о необходимости обработки риска или ее отсутствии, а также о приоритетности обработки риска. Частоте возникновения опасного события поставим в соответствие вероятность возникновения функционального отказа КВС (табл. II.5.6), руководствуясь следующими соображениями. Минимальная вероятность функционального отказа системы (отказа в выполнении заданной функции), приводящего к любому виду опасных последствий, в соответствии со стандартом по функциональной безопасности [15] не превышает значения 10^{-8} 1/ч. Достичь более высокого уровня функциональной безопасности практически невозможно. Следовательно, этот уровень вероятности возникновения функционального отказа можно сопоставить с градацией частоты «***невероятного возникновения***» опасного события (см. табл. II.5.6).

С другой стороны, в соответствии с оценками погрешностей расчетов надежности, приведенных в гл. I.5 [1], каждый последующий уровень градации частоты отказов может быть оценен как отличающийся от предыдущего на один порядок. При этом градация «частого возникновения» опасных событий соответствует вероятности 10^{-3} 1/ч или средней наработки до функционального отказа 1000 часов, что минимально для информационных систем общего назначения.

Продолжим наши рассуждения и сопоставим каждому значению вероятности, приведенной в табл. II.5.7, допустимый уровень тяжести последствия опасного события согласно табл. II.5.8. Результаты сопоставления приведены в табл. II.5.8. Они позволяют определить требуемый количественный уровень функциональной надежности информационной системы. При этом будем руководствоваться следующими соображениями. При «частом возникновении» опасных событий (вероятность 10^{-3} 1/ч) не допустим никакой уровень тяжести последствия опасного события. Этот уровень требований неприемлем. С другой стороны, при «невероятном возникновении» (вероятность $\leq 10^{-8}$ 1/ч) допустим любой уровень тяжести последствия опасного события.

Таблица II.5.8. Определение допустимого уровня риска для ответственных и критически важных информационных систем

Частота возникновения опасного события	**Вероятность возникновения функционального отказа ИС (в течение часа работы)**	**Допустимый уровень тяжести последствия опасного события для ответственных информационных систем**	**Допустимый уровень тяжести последствия опасного события для критически важных информационных систем**
Частое возникновение	10^{-3}	**Нежелательный**	**Недопустимый**
Возможное возникновение	10^{-4}	**Допустимый**	**Недопустимый**
Редкое возникновение	10^{-5}	**Допустимый**	**Нежелательный**
Маловероятное возникновение	10^{-6}	**Не принимаемый в расчет**	**Нежелательный / Допустимый**
Невозможное возникновение	10^{-7}	Не принимаемый в расчет	**Допустимый**
Невероятное возникновение	10^{-8}	**Не принимаемый в расчет**	Не принимаемый в расчет
Уровни тяжести последствий		**Незначительный**	Значительный / Критический

Следовательно, этот уровень требований чрезмерен. Остальные градации требований к функциональной надежности систем можно разделить на две группы. *Первая группа* – это информационные системы, при отказе которых допустим только незначительный уровень тяжести последствий опасного события. При значительном уровне тяжести и, тем более, критическом уровне тяжести последствий опасного события оценка степени риска относится к категориям «нежелательно» или «недопустимо». При этом градация частоты возникновения опасного события находится на уровне «возможное возникновение» или «редкое возникновение» с высокой вероятностью возникновения функционального отказа $10^{-4} – 10^{-5}$. Требование дальнейшего понижения вероятности опасного события для таких систем допустимо, однако подобное требование чрезмерно и вызовет неоправданные экономические затраты. К первой группе следует отнести требования к информационным системам, которые выполняют ответственные задачи, но их функциональные отказы не должны оказывать существенного влияния на внешнюю среду.

Ко второй группе относятся критически важные информационные системы. Даже при редких опасных отказах возможен значительный и даже критический уровень тяжести последствия опасного события, что соответствует «невозможной» или даже «маловероятной» частоте опасного события и, следовательно, вероятностям возникновения функционального отказа КВС на уровнях $10^{-6} – 10^{-7}$ в течение часа работы.

Эти вероятности можно расценивать как базовые количественные требования к функциональной надежности КВС. Предъявлять более высокие требования чрезмерно и экономически не оправдано, а менее высокие требования недопустимо или нежелательно. В отдельных случаях по решению эксплуатирующей организации возможно предъявление требования к КВС на

уровне 10^{-5} 1/ч даже при условии нежелательного уровня тяжести последствия опасного события.

Следует отметить, что оценка величины ущерба находится в прямой зависимости от принятого принципа обеспечения функциональной надежности и функциональной безопасности КВС. Так, из применяемого в Великобритании и в России принципа ALARP (настолько низкий уровень остаточного риска, насколько это в разумной мере исполнимо) [60] следует, что существует некоторая допустимая область границ риска, в пределах которой риски приемлемы, а затрачиваемые на это усилия вполне оправданы. Этот принцип ориентирован, главным образом, на экономические риски. Если приоритетными являются социальные риски, то для этих целей удобно применить широко используемый в Германии MEM – принцип [60]. Практические рекомендации, которые следуют из этого принципа и которые приняты в развитых странах, например, для транспортных систем, состоят в том, что допустимый ущерб от функционального отказа КВС не должен превышать 10^{-5} случаев гибели человека в год или 10^{-4} случаев тяжелых травм человека в течение года или 10^{-3} случаев легких травм человека в течение года.

Вместе с тем, нет общего рецепта для задания количественных требований к функциональной надежности КВС. Приведенные выше материалы могут быть использованы как методические соображения для задания требований к конкретным системам.

II.5.6. Требования к функциональной надежности и архитектуре программного обеспечения критически важных систем

II.5.6.1. Полный набор требований к функциональной надежности программного обеспечения (спецификация требований)

Спецификация требований к функциональной надежности ПО является всеобъемлющим документом для каждого разработчика программного обеспечения и обеспечивает его всеми необходимыми сведениями для того, чтобы у него не возникло необходимости искать в других документах требования к программному обеспечению. Спецификация требований должна выражать требуемые свойства разрабатываемого программного обеспечения, но не процедуры его разработки. Она должна быть выражена и организована таким способом, чтобы быть: полной, ясной, точной, недвусмысленной, поддающейся верификации, испытанию, пригодной для корректирования и выполнимой; отслеживаемой в обратном направлении ко всем входным и выходным документам, где *входными документами* должны быть [61]:

– спецификация требований к КВС;

– спецификация требований к функциональной надежности КВС;

– описание архитектуры КВС;

– план обеспечения качества программного обеспечения системы,

а *выходными документами*:

– спецификация требований к функциональной надежности ПО КВС;

– спецификация требований для испытаний надежности ПО КВС.

Спецификация требований к программному обеспечению должна включать в себя способы выражения и описания, которые являются понятными персоналу, привлеченному на период всего жизненного цикла системы.

Требования к программному обеспечению должны устанавливать и документировать все интерфейсы со всеми взаимодействующими системами. В спецификации требований должны быть подробно описаны все предусмотренные режимы работы ПО в системе, а также все существенные режимы поведения программируемых электронных устройств, в частности, поведение при отказах. Должны быть обозначены и документированы любые взаимные ограничения между программным и аппаратным обеспечением. Спецификация требований должна показывать степень самопроверки ПО и заданную степень программной проверки аппаратного обеспечения. Самопроверка ПО состоит из обнаружения и сообщений программным обеспечением об его собственных отказах и ошибках. В спецификацию требований должны также включаться требования для периодических испытаний функций.

Спецификация требований к испытаниям программного обеспечения должна быть разработана на основе спецификации требований к программному обеспечению. Эта спецификация предназначена для валидации всех требований к программному обеспечению, также как описание испытаний, которые должны быть выполнены для окончательно разработанного программного обеспечения. Спецификация требований для испытаний программного обеспечения должна идентифицировать для каждой необходимой функции контрольные примеры, которые включают в себя:

– требуемые входные сигналы с их последовательностями и их значениями;

– ожидаемые выходные сигналы с их последовательностями и их значениями;

– критерии приемки, включая аспекты производительности и качества.

Наряду с качественными требованиями к функциональной надежности программного обеспечения должны быть предъявлены количественные требования. При этом следует исходить из очевидной посылки: уровень функциональной надежности ПО не может быть равным или ниже уровня функциональной надежности системы. Так, если к вероятности функционального отказа $Q_{КВС}$ критически важной системы предъявлены требования в интервале значений

$$10^{-7}1/ч < Q_{КВС} < 10^{-6}1/ч,$$

то к вероятности функционального отказа этой системы за счет программного обеспечения должны быть предъявлены требования на уровне

$$Q_{КВС}^{ПО} \le 10^{-7}\ 1/ч.$$

II.5.6.2. Требования к архитектуре функционально надежного программного обеспечения

Требования к архитектуре ПО системы предназначены для того, чтобы:

- архитектура позволяла удовлетворить заданные требования к функциональной надежности программного обеспечения;
- идентифицировать и оценить значимость для функциональной надежности организации взаимодействия между программным и аппаратным обеспечением системы;
- проанализировать требования, возлагаемые архитектурой системы на программное обеспечение.

Для формирования спецификации архитектуры ПО (единственный выходной документ) должны быть применены следующие входные документы:

– спецификация требований к программному обеспечению;

– спецификация требований к функциональной надежности системы;

– описание архитектуры системы;

– план обеспечения качества программного обеспечения.

В спецификация архитектуры ПО должна быть показана осуществимость достижения соответствия спецификации требований к ПО в целом со спецификацией требований к функциональной надежности. Спецификация архитектуры ПО должна определять, оценивать и подробно описывать смысл всех взаимодействий между аппаратным и программным обеспечением. Спецификация архитектуры ПО должна определять все программные компоненты и идентифицировать для этих компонентов следующие очень важные для построения функционально надежного ПО факторы:

– являются ли эти компоненты новыми, существующими или принадлежащими конкретной компании или частному лицу;

– были ли эти компоненты ранее аттестованы, и, если да, то условия аттестации;

– является ли конкретный компонент связанным с функциональным отказом КВС или нет.

При использовании стандартного программного обеспечения в КВС должны быть соблюдены следующие меры предосторожности:

• стандартное программное обеспечение должно быть подвергнуто испытаниям на предмет аттестации;

• должен быть выполнен анализ возможных ошибок стандартного программного обеспечения;

• должна быть определена стратегия обнаружения ошибок стандартного программного обеспечения и защиты системы от этих ошибок;

• стратегия защиты должна быть предметом сертификационных испытаний;

• должен быть журнал ошибок и они должны быть проанализированы;

• насколько это возможно, должны быть использованы только самые простые функции стандартного программного обеспечения.

Если ранее разработанное ПО используется как часть проекта, то оно должно быть ясно идентифицировано и документировано. Спецификация архитектуры программного обеспечения должна обосновывать пригодность ПО для удовлетворения спецификации требований к нему и соответствия заданному уровню функциональной надежности ПО. Последствия любых изменений в программном обеспечении для остальной системы должны быть тщательно рассмотрены и проанализированы для того, чтобы установить, требуют ли они проведения повторных сертификационных испытаний. Должны быть доказательства, что соблюдаются спецификации интерфейсов с другими модулями, которые не проверяются, не аттестуются и не оцениваются. Всякий раз, когда возможно, в проекте должны быть использованы существующие верифицированные программные модули, разработанные в соответствии с этим стандартом.

Архитектура ПО должна минимизировать часть программного обеспечения, которая наиболее влияет на функциональную надежность. В случае, если ПО состоит из компонентов с различными уровнями функциональной надежности, то все программное обеспечение должно рассматриваться как ПО с самым высоким среди компонентов уровнем функциональной надежности. Исключение составляет то обстоятельство, когда имеется доказательство независимости между компонентами с

более высоким уровнем надежности ПО и компонентами с более низким уровнем надежности ПО. Это доказательство должно быть записано в спецификации архитектуры программного обеспечения.

Спецификация архитектуры ПО должна быть сформирована таким образом, чтобы выбранные технические приемы и меры удовлетворяли спецификации требований к программному обеспечению в соответствии с заданным уровнем его функциональной надежности.

Контрольные вопросы

1. Поясните содержание понятия «критически важная система (КВС)»
2. Какая связь имеет место между информационной безопасностью и функциональной надежностью КВС?
3. Какая аргументация может быть применена при задании допустимого уровня тяжести последствия опасного события при функционировании КВС?
4. Перечислите требования к функциональной надежности программного обеспечения (спецификация требований).
5. Перечислите требования к архитектуре функционально надежного программного обеспечения.
6. Опишите порядок оценки сбойных ошибок в критически важных системах.
7. Какие уровни сбойных ошибок имеют место при работе больших и сверхбольших интегральных схем?
8. Какие количественные требования могут предъявляться к функциональной надежности критически важных информационных систем?
9. Раскройте сущность методов HEART и TESEO прогнозирования ошибок операторов.

ЗАКЛЮЧЕНИЕ

Информационные системы применяются для решения широкого спектра научных и производственных задач – от традиционных задач сбора, обработки, накопления и хранения информации, от решения задач искусственного интеллекта до управления ответственными объектами в реальном масштабе времени. Эти задачи имеют актуальное значение в жизни современного общества. Отсюда высокий уровень требований, предъявляемых к надежности информационных систем. При этом следует различать два класса задач обеспечения их надежности.

Первый класс – это задачи структурной надежности. Их содержание, методы анализа структурной надежности сложных восстанавливаемых систем, способы оценки показателей надежности при ведены нами в работе [73].

Второй класс – это задачи функциональной надежности информационных систем. Само понятие функциональной надежности трактуется различными специалистами неоднозначно. Некоторые полагают, что если в системе в течение времени выполнения конкретной задачи не возникли отказы аппаратуры, то считается, что система надежно функционирует. Распространено мнение, что для многофункциональной информационно – управляющей системы следует рассчитывать структурную надежность системы относительно каждой функции. Известны и другие подходы и попытки оценить функциональную надежность дискретных систем. Однако вследствие того, что все они сводятся к применению методологии структурной надежности к решению задач функциональной надежности, желаемого ре-

зультата оценки функциональной надежности информационных систем до настоящего времени не удается достичь. Так, прошло уже около полувека с того времени, когда были выявлены и математически описаны сбойные и программные ошибки, ошибки операторов, однако до настоящего времени не удалось создать какие – либо научно аргументированные методы расчета надежности информационных систем с учетом влияния указанных факторов. И это при том, что именно указанные факторы оказывают определяющее влияние на надежность функционирования информационных систем.

Следует обратить внимание читателя на очень важное обстоятельство – теория структурной надежности исследует процессы отказов и восстановлений объектов (элементов и структур в целом). Вопросы надежности (безошибочности) выполнения информационных процессов оказались за пределами задач, решаемых в рамках классической (структурной) надежности. Поэтому необходимо было разработать теоретические основы функциональной надежности информационных систем, которые, по нашему мнению, должны включать в себя:

– определение функционального отказа и определение на этой основе функциональной надежности системы;

– определение угроз функциональной надежности информационных систем, включая информационные атаки;

– формализация требований к показателям, создание системы показателей функциональной надежности информационных систем, разработка новых и адаптация существующих методов расчета этих показателей;

– разработка и исследование математических моделей сбоев функционального характера и сбойных ошибок при выполнении функциональных задач;

– систематизация понятий в области качества и надежности программных средств;

– разработка показателей функциональной надежности программных средств на основе разработанной системы показателей для информационной системы в целом;

– обобщение практических сведений об ошибках операторов и прогнозирование их функциональной надежности;

– задание требований к функциональной надежности критически важных информационных систем.

В монографии приведены разработанные в основном автором перечисленные выше разделы теории функциональной надежности. Они ранее публиковались в виде отдельных статей или в постановочном плане. В связи с этим можно утверждать, что впервые изложены в систематизированном виде основные положения теории функциональной надежности информационных систем как составной части общей теории надежности.

Особое место в проблеме функциональной надежности занимают задачи анализа надежности критически важных информационных систем (КВС). Функциональная надежность КВС определяется способностью системы предотвращать или минимизировать последствия нарушения процесса нормального функционирования из-за деструктивных воздействий с целью исключения недопустимого ущерба внешней среде. Поддержание надежных условий функционирования обеспечивается содержащимися в КВС механизмами парирования опасных отказов системы, что соответствует требованию обеспечения приемлемого уровня риска. Опасные отказы КВС могут привести к большому ущербу. Даже при редких опасных отказах возможен значительный и даже критический уровень тяжести последствия опасного события, что соответствует «невозможной» или даже «маловероятной» частоте опасного события и, следовательно, вероятностям возникновения функционального отказа КВС на уровнях в течение часа работы. Эти вероятности можно расце-

нивать как базовые количественные требования к функциональной надежности КВС. Предъявлять более высокие требования чрезмерно и экономически не оправдано, а менее высокие требования недопустимо или нежелательно. В отдельных случаях по решению эксплуатирующей организации возможно предъявление требования к КВС на уровне в течение часа работы даже при условии нежелательного уровня тяжести последствия опасного события. Вместе с тем, нет общего рецепта для задания количественных требований к функциональной надежности КВС. Приведенные выше материалы могут быть использованы как методические соображения для задания требований к конкретным системам.

Количественные требования есть составная часть общих требований к функциональной надежности КВС. Большое значение имеют качественные требования, которые в зависимости от предъявляемых количественных норм определяют организационный и технологический создания функционально надежной КВС. В первую очередь, это относится к программному обеспечению системы. Требования к программному обеспечению должны устанавливать и документировать интерфейсы с взаимодействующими системами. В спецификации требований должны быть описаны все предусмотренные режимы работы программного обеспечения в системе, а также режимы поведения программируемых электронных устройств, в частности, поведение при сбоях и отказах. Должны быть обозначены и документированы любые взаимные ограничения между программным и аппаратным обеспечением. В зависимости от уровня предъявляемых к КВС количественных требований спецификация качественных требований должна показывать степень самопроверки программного обеспечения и заданную степень программной проверки аппаратного обеспечения.

Требования к архитектуре программного обеспечения предназначены для того, чтобы архитектура позволяла удовлетворить заданные количественные требования к функциональной надежности КВС и в частности программного обеспечения, а также для того, чтобы установить уровень организации взаимодействия между программным и аппаратным обеспечением системы.

Литература

1. Шубинский И.Б. Структурная надежность информационных систем – М.: «Журнал Надежность», 2012, 216 с.

2. Технический отчет ISO/IEC TR 19760. Первое издание 2003-11-15 Проектирование систем – Руководство по применению ISO/IEC 15288 (Процессы жизненного цикла системы).

3. ГОСТ 24.701-86. Надежность автоматизированных систем управления. Основные положения.

4. Шубинский И.Б. Расчет надежности цифровых устройств. М.: Знание, 1984. – 48с.

5. Наумов Ю.Е., Аваев Н.А., Бедрековский М.А. Помехоустойчивость устройств на интегральных логических схемах. М.: Сов. радио, 1975. -216 с.

6. Майерс Г. Надежность программного обеспечения: Пер.с англ. – М.: Мир, 1980, 360 с.

7. Липаев В. В. Надежность программного обеспечения АСУ. – М.: Энергоиздат, 1989. 240 с.

8. Казарин О.В. Методология защиты программного обеспечения. – М.: МЦНМО (МГУ), 2009, 464 с.

9. О преднамеренных и непреднамеренных ошибках человека-оператора. М. А. Котик // Психологический журнал, №5, 1993.

10. Dubrovsky V. A taxonomy of human errors based upon the structure of an action. In Proceedings of 1985 International Conference on Systems, Man, and Cybernetics (pp. 903 907). Tucson, Arizona: IEEE

11. Профессиональный отбор горочного операторского звена с точки зрения теории информационного метаболизма / Ти-

хонов А.П., Павлухина М.О. // Информационно-управляющие системы на железнодорожном транспорте. – 1997, – №№ 5-6 .

12. Дружинин Г.В. Учет свойств человека в моделях технологий. – М.: МАИК «Наука/Интерпериодика», 2000, 327 с.

13. Дружинин Г.В., Сергеева И.В. Качество информации. – М.: Радио и связь, 1990, 172 с.

14. CENELEC EN 50159. Применения на железнодорожном транспорте – Системы связи, сигнализации и обработки данных. Часть 2: Обеспечение безопасности при связи по открытым системам передачи. – 2000.

15. ГОСТ Р/МЭК 61508. Функциональная безопасность электрических/ электронных/ программируемых электронных систем безопасности.-2008.

16. Гнеденко Б.В., Беляев Ю.К., Соловьев А.Д. Математические методы в теории надежности. – М.: Наука, 1965. – 524с.

17. Кемени Дж., Снелл Дж. Кибернетическое моделирование. Некоторые приложения / Пер. с англ. под ред. И.Б. Гутчина. М.: Сов.радио, 1972. – 192 с.

18. Шубинский И.Б., Куликов В.А. Методика расчета надежности выполнения алгоритмов. – Надежность и контроль качества, 1977, №8, с. 39-45.

19. Кокс Д.Р., Смит В.Л. Теория восстановления / Пер. с англ. под. ред. Ю.К.Беляева. М.: Сов. радио, 1967. – 236 с.

20. Гавриков В. О., Григорьев В.В., Платунов А.Е., Шубинский И.Б. Микропроцессорная система диспетчерской централизации “Тракт” нового поколения / Вторая международная научно-практическая конференция “Информационные технологии на железнодорожном транспорте”, С-Пб., Репино, 1997, 10 с.

21. Рейнгольд Э., Нивельт Ю., Део Н. Комбинаторные алгоритмы. Теория и практика (Пер. с англ. под ред. В.Б.Алексеева). М.: Мир, 1980. – 476 с.

22. Ссылки: http://www.giac.unibel.by/docs/pdf/1-2005/s10-1-2005.pdf; http://www.cs.umd.edu/~basili/papers.html;http://msquaredtechnologies.com/m2rsm http://www.aivosto.com/project/help/pm-loc.html; http://kapustin-andrey.boom.ru/Materials/Metrics2.htm; http://kapustin- Materials andrey.boom.ru/ /Metrics1.htm; http://www.met-rix.narod.ru/index.htm; http://www.maxkir.com/sd/newmethRUS.html;\ http://www.optim.su/cs/2003/4/AOP2/AOP.asp;;http://www.mstandard.ru/certification/03/;http://www.virtualmachinery.com/jhawkmetrics.htm\;http://msquaredtechnologies.com/m2rsm/docs/cyclocomplex/cyclo_1.htm

23. IEEE Std 1058-1998, Standard for Software Project Management Plans.

24. ГОСТ Р ИСО/МЭК 9126-93. Информационная технология. Оценка программного продукта. Характеристики качества и руководство по их применению.

25. ГОСТ Р ИСО/МЭК 12119-2000. Информационная технология. Пакеты программ. Требования к качеству и тестирование.

26. ГОСТ 28195-89. Оценка качества программных средств. Общие положения.

27. Холстед М. Начала науки о программах. – М.: Финансы и статистика, 1981.

28. IEEE Std 1045-1992, Standard for Software Productivity Metrics.

29. IEEE Std 1062-1998, Recommended Practice for Software Acquisition.

30. ГОСТ Р ИСО/МЭК ТО 16326-2002. Программная инженерия. Руководство по применению ГОСТ Р ИСО/МЭК 12207 при управлении проектом.

31. Половко А.М., Гиндин С.И., Новоселов А.И. Надежность программного обеспечения цифровых вычислительных комплексов. Л.: ЦНИИ «Румб», 1988.

32. Тейер Т., Липов М., Нельсон Э. Надежность программного обеспечения. М.: Мир,1981.

33. Муса Дж. Д. Измерение и обеспечение надежности программных средств//ТИИЭР. 1980. Т.68, №9 С.113-128.

34. ГОСТ 27.002 – 89. Надежность в технике. Основные понятия. Термины и определения.

35. Изосимов А.В., Рыжко А.Л. Метрическая оценка качества программ. – М.: МАИ, 1989.

36. ITC.UA, Вячеслав Колдовский, 12.04. 2007 г. Разработка ПО: метрики программных проектов.

37. Oviedo E.J. Control flow, data flow and program complexity. – Proc. IEEE COMPSAC. – Chicago, IL, Nov. 1980. – pp. 146-152.

38. Новичков Александр, Шамрай Александр , Черников Алексей. Метрики кода и их практическая реализация в Subversion (http://cmcons.com/articles/CC_CQ/dev_metrics/)

39. В.А. Петрухин, Е.М. Лаврищева. Методы и средства инженерии программного обеспечения (http://www.intuit.ru/department/se/swebok/).

40. Карповский Е.Я. Надежность специального математического обеспечения. – Киев; Одесса: Вища школа. Головное издательство, 1982.-152 с.

41. Пальчун Б.П., Юсупов Р.М. Оценка надежности программного обеспечения. – С.-Пб.:Наука, 1994.-84 с.

42. Уткин Л.В., Шубинский И.Б. Нетрадиционные методы оценки надежности информационных систем/ Под ред. проф. И.Б. Шубинского – С.Пб.: «Любавич», 2000-173 с.

43. Шаракшанэ А.С., Шахин В.П., Халецкий А. К. Испытания программ сложных автоматизированных систем/ Под ред. проф. А.С.Шаракшанэ – М.: Высш. Школа, 1982-192 с.

44. BS EN 50126: 1999. Railway applications – the specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS).

45. ГОСТ Р 50922-96. ЗИ. Основные термины и определения. 1996.

46. Защита от НСД к информации. Термины и определения, Москва, 1998 г.

47. Концепция защиты СВТ и АС от НСД к информации, Москва, 1998 г.

48. АС. Защита от НСД к информации. Классификация АС и требования по защите информации, Москва, 1998 г.

49. СВТ. Защита от НСД к информации. Показатели защищенности от НСД к информации, Москва, 1998 г.

50. ГОСТ Р 51275-99. ЗИ. Объект информатизации. Факторы, воздействующие на информацию. Общие положения. 1999.

51. Система сертификации средств защиты информации по требованиям безопасности информации № РОСС RU.0001.01БИ00. «Положение о сертификации средств защиты информации по требованиям безопасности информации».

52. ISO 15408 -1-3. (ГОСТ Р – 2002). Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. 2003.

53. ISO 13335 – 1-5. ИТ. ТО. Руководство по управлению безопасностью. 1996-1998 .

54. ISO 10181: 1-7. ВОС. Структура работ по безопасности в открытых системах. 1996-1998.

55. ГОСТ Р 50739-95. СВТ. Защита от несанкционированного доступа к информации. 1995.

56. Защита от несанкционированного доступа к информации. Часть 1. Программное обеспечение средств защиты информации. Классификация по уровню контроля отсутствия недекларированных возможностей, Москва, 1998.

57. ГОСТ Р 51583-00. ЗИ. Порядок создания автоматизированных систем в защищенном исполнении. Общие положения. 2000.

58. ГОСТ Р 51624-00. ЗИ. Автоматизированные системы в защищенном исполнении. Общие требования. 2000.

59. Специальные требования и рекомендации по технической защите конфиденциальной информации. (СТР-К), Москва, 2001.

60. IEC 62278: 2002. Railway applications – the specification and demonstration of reliability, availability, maintainability and safety (RAMS).

61. IEC 62279: 2002. Railway applications – Communication, signaling and processing systems – Software for railway control and protection systems

62. Kumamoto H., Tanaka T., Inoue K. A new Monte Carlo method for evaluating system-failure probability // IEEE Trans. Reliab., vol. R-36, N1, 1987, pp. 63-69.

63. Смит Д.Дж. Безотказность, ремонтопригодность и риск. Практические методы для инженеров, включая вопросы оптимизации надежности и систем, связанных с безопасностью / Д.Дж. Смит; [пер.с англ. Хвилевицкого Л.О.] – М.:ООО «Группа ИДТ», 2007. – 432 с.

64. ГОСТ Р 51901.15 – 2005 (МЭК 61165:1995).

65. Военная инженерная психология / под ред. Б.Ф. Ломова, А.А.Васильева, В.В.Офицерова, В.Ф.Рубахина.М.: Воениздат, 1970, – 400 с.

66. Коваленко О. В., Петрин С. В. Вероятностный анализ безопасности сложных систем человек-машина. Сарово: РФЯЦ-ВНИИЭФ, 2010, – 114с.

67. Швир В. Надежность электронных схем в устройствах СЦБ // Железные дороги мира, № 1, 1986, с. 59-67.

68. ГОСТ Р 54505-2011. Безопасность функциональная. Управление рисками на железнодорожном транспорте.

69. Шубинский И. Б. Основы анализа сложных систем. – Ленинград, Пушкин, МО, 1988.-206 с.

70. Надежность технических систем и техногенный риск. Акимов В.А. и др., под ред. проф. Фалеева М.И. М.: МЧС, 2002.

71. Avizienis A., Laprie J-C. and Randell B. Dependability of computer systems/ Fundamental concepts, terminology and examples. Technical report, LAAS – CNRS, October, 2000

72. Rus I., Komi-Sirvio S., Costa P. Computer program with insurance of high reliability. Technical report, IFIP WG-10.4, March, 2008.

СОДЕРЖАНИЕ