Лекция 2. Функциональная надежность программного обеспечения в информационных системах

Цель лекции: курса.

План лекции:
Введение.
1 Функциональная надежность программного обеспечения в информационных системах
Заключение
Контрольные вопросы

Ключевые слова: [выбрать самостоятельно].

Содержание лекции:

Введение

Понятие «функциональная надежность» до настоящего времени трактуется различными исследователями неоднозначно.

1 Функциональная надежность программного обеспечения в информационных системах

В [1, с. 13] под функциональной надежностью понимается по существу готовность системы к выполнению предусмотренных задач. Эта позиция сформулирована следующим образом: «Характеристики функциональной надежности программного обеспечения включают в себя готовность и либо присущие ей, либо внешние влияющие факторы, такие как надежность и доступность (включая отказоустойчивость и восстанавливаемость), безопасность (включая обеспечение конфиденциальности и целостности), пригодность для обслуживания, долговечность и техническую поддержку».

Другой, более распространенный, подход, который закреплен в стандарте [2], состоит в том, что для многофункциональной ИУС (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, программного 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 (рис. 2.1). 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 параметр потока отказов. В качестве комплексных показателей надежности используют коэффициенты готовности, технического использования и сохранения эффективности каждой i-й ФП.

На рис. 2.1 в каждой функциональной подсистеме квадратами обозначены объекты, участвующие в выполнении функции этой подсистемы. Например, в выполнении функции

---

[1] Федеральный закон РФ № 184-ФЗ «О техническом регулировании»

[2] ГОСТ 19.201—78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению.

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.

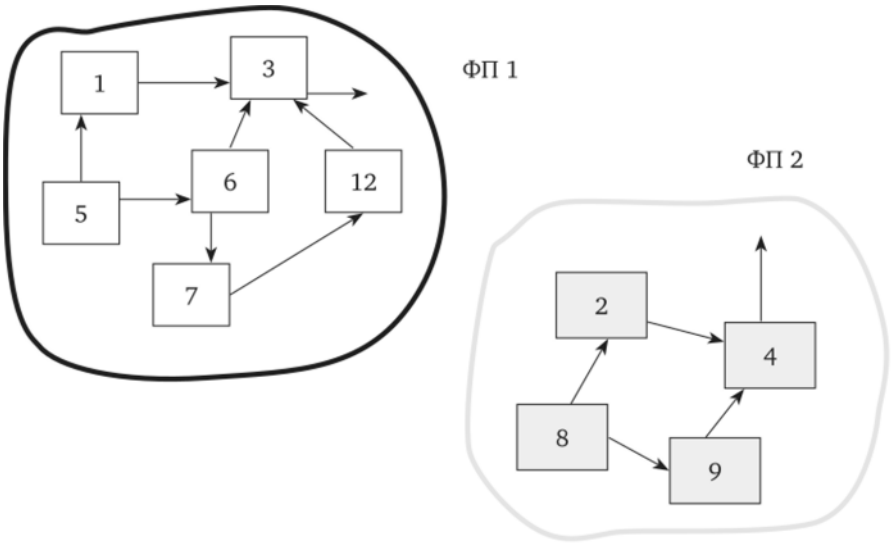


Рис. 2.1. Распределение технических средств информационной системы по функциональным подсистемам

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. И тому есть две причины.

1. Разделение многофункциональной информационной системы на ряд функциональных подсистем 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 (Н-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: они 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 принять вероятность успешного выполнения заданной процедуры при поступлении запроса. Возникают вопросы. Во-первых, процедура — это функция или часть ее? Во-вторых, что означает «успешное выполнение»: во время ее выполнения не было отказов, или процедура выполнена

[3] ГОСТ 19.201—78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению.

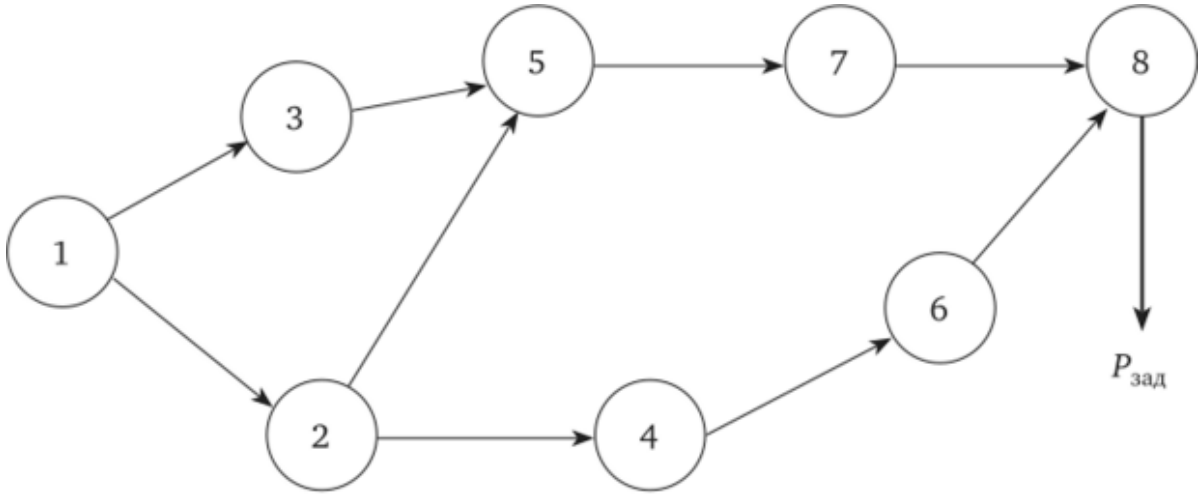
качественно, или что-либо еще? Ответы на поставленные вопросы авторами не предусмотрены.

Таким образом, в теории надежности вопрос о том, как рассчитывать надежность системы с динамично изменяющейся структурой, остается открытым. Однако сегодня можно утверждать, что эта задача не относится к возможностям теории и практики структурной 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. 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. 2.2 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 выполнения 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 выполнении информационных процессов.



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

Рис. 2.2. Фрагмент графа алгоритма задачи. Зависимость вероятности безошибочного выполнения подзадачи от функциональной надежности выполнения составных процедур

Рассмотрим причинно-следственные связи, приводящие к нарушению информационного процесса при выполнении задачи в целом. Дело в том, что информационный процесс состоит из совокупности иерархически упорядоченных информационных процессов и реализуется от низшего уровня иерархии (этот уровень иерархии процесса обозначим как imax) до процесса высшего уровня иерархии (этот уровень иерархии принято обозначать как i0 => 0). Ошибки, возникшие на любом уровне иерархии, распространяются до высшего уровня и могут привести к функциональному отказу информационной системы.

Причинно-следственная цепочка ошибок применительно к информационному процессу показана на рис. 2.3.

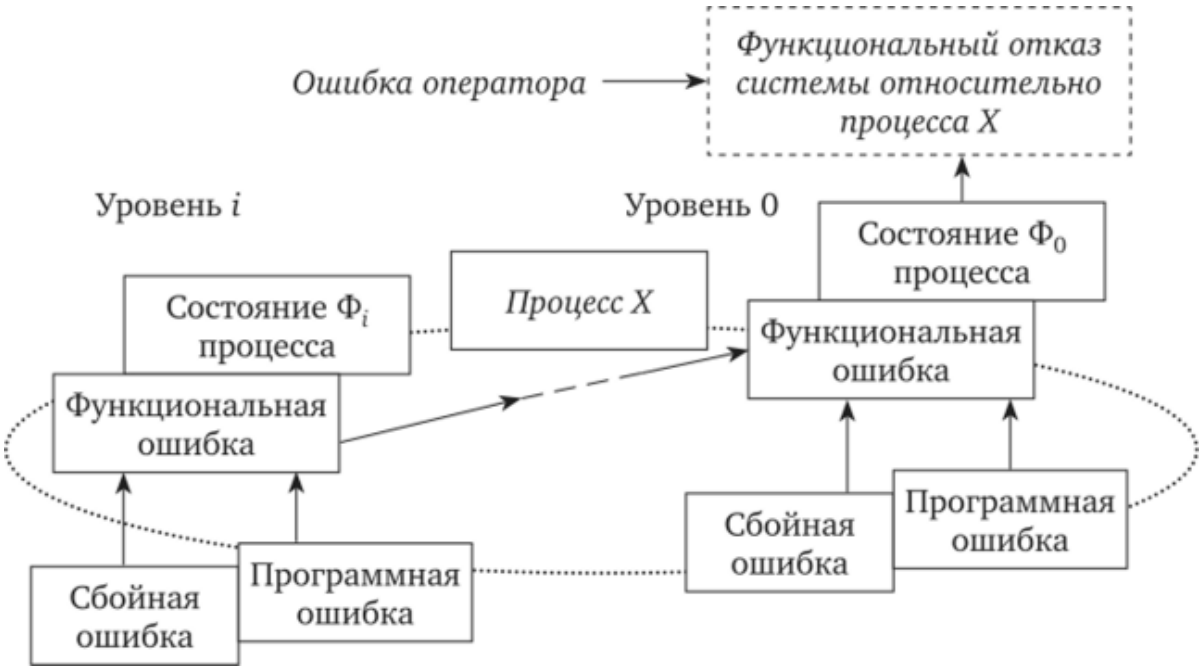


Рис. 2.3. Концептуальное представление взаимосвязи функциональной ошибки и функционального отказа системы относительно данного информационного процесса

В результате возникновения сбойной ошибки или проявления программной ошибки при выполнении процесса на ?-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. Функциональный отказ — это событие невыполнения функциональной задачи вследствие нарушения информационного процесса.

Не исключено также и то, что распространяемая по процессам функциональная ошибка окажется недостаточно существенной в соответствии с 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.

В настоящее время не существует общепринятого строгого определения понятия «надежность ПО» и, следовательно, понятия «функциональная надежность ПО». По сути своей оба эти понятия относятся к одному и тому же предмету — определению одной из наиболее важных характеристик качества функционирования ПО. В различных

публикациях предлагаемые 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 некоторых работах 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» [4, с. 4]. В данном определении есть сомнительные условия, которые не определяют надежность ПО. Прежде всего, учет заданного периода времени 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.

Другое сомнительное условие в приведенном выше и аналогичных ему 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 [5, с. 162]: 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 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).

---

[4] ГОСТ Р МЭК 62443-2-1—2015. Сети коммуникационные промышленные. Защищенность (кибербезопасность) сети и системы. Часть 2-1. Составление программы обеспечения защищенности (кибербезопасности) системы управления и промышленной автоматики.

[5] ГОСТ 19.202—78. Единая система программной документации. Спецификация. Требования к содержанию и оформлению.

Определение функциональной надежности ПО было бы неполным без учета условий возможной трансформации ошибки в выходных результатах в функциональный отказ информационной 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:

а) при наличии в составе ПО качественных средств оперативного обнаружения ошибок;

б) при наличии в составе ПО средств обеспечения устойчивости к ошибкам.

Заключение

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, повысить безошибочность процесса преобразования исходных данных в результаты.

Контрольные вопросы

Смотри руководство по организации самостоятельной работы магистрантов.