Глава 5

РЕКОМЕНДАЦИИ ПО ПРИМЕНЕНИЮ МОДЕЛЕЙ НАДЁЖНОСТИ

5.1. КЛАССИФИКАЦИЯ МОДЕЛЕЙ НАДЁЖНОСТИ

Анализ существующих методов оценивания надёжности ПС позволяет предложить несколько подходов к классификации моделей в зависимости от критерия, положенного в основу классификации.

С точки зрения структуры исследуемого ПС модели можно разделить на рассматривающие ПС в виде «чёрного ящика» и на учитывающие структурные взаимосвязи между составными элементами. Появление первых связано с изучением специфики поведения программ и свойств статистических данных, полученных в процессе испытаний. События появления ошибок ПС при проведении испытаний представляют собой случайный процесс, который может быть описан при помощи вероятностных характеристик, оцениваемых методами математической статистики. Вторые способны учитывать особенности структурных и функциональных взаимосвязей между составными частями ПС, а также и результаты их автономных испытаний.

Другой подход основывается на учёте характера получаемых оценок, количественных или качественных. В соответствии с этим классификация может предусматривать модели с расчётом числа ошибок и без расчёта числа ошибок. С числом ошибок, оставшихся в ПС, связаны оценки стоимости обслуживания ПС. Модели без вычисления количества ошибок позволяют получать только надёжностные характеристики ПС.

Для персонала, занимающегося разработками и испытаниями ПС, наиболее удобной представляется классификация, основанная на учёте стадий жизненного цикла разрабатываемых ПС. Таким образом, модели можно различать по эффективности их применения на различных стадиях жизненного цикла ПС.

Для стадий написания спецификаций, проектирования и кодирования П наилучшие оценки позволяют получить модели, основанные на методике, предложенной Холстедом. В её основу положено свойство человеческого мозга, что из определённого числа мысленных различений, необходимых для написания П, часть является ошибочной. При этом в качестве основных метрических характеристик используются:

– количество различных операторов, используемых в П;
– количество различных операндов, используемых в П;
– количество всех операторов, используемых в П;
– количество всех операндов, используемых в П.

Операторами считаются те инструкции П, которые связаны с реализацией алгоритма, а операндами – переменные и константы П.

Для стадии отладки П широкое распространение получил байесовский подход. Его преимуществом является возможность объединения в процессе оценивания имеющихся априорных сведений, полученных на ранних этапах жизненного цикла программ, с данными их испытаний. Это объединение реализуется в виде пересчёта априорного распределения оцениваемого показателя в апостериорное и позволяет существенно снижать объём статистических испытаний при отладке. Достоверность оценок, получаемых при этом, не снижается.

На стадиях отладки и эксплуатации ПС оценки их характеристик надёжности хорошо описываются детерминистическими моделями, наиболее совершенным представителем которых может выступать модель Дж. Мусы. Она предполагает, что оценки искомых показателей являются детерминированными известными величинами, определяемыми по результатам испытаний. Модель позволяет распознавать ситуацию, когда надёжность не возрастает или не убывает. Однако дальнейшее поведение характеристик надёжности не прогнозируется. Вместе с тем модель позволяет определять время функционирования, необходимое для достижения заданного показателя надёжности. Присутствие в модели

компонента календарного времени позволяет связать процесс достижения заданной надёжности с календарными датами. Кроме того, модель в принципе позволяет также учитывать и уровень профессиональной подготовленности программистов.

В качестве достаточно универсальной модели, применимой как на стадиях разработки, так и при эксплуатации ПС, может быть использована модель Э. Нельсона. Существенным достоинством её, в отличие от прочих моделей, является то, что она позволяет учитывать «мощность» тестов, используемых для отладки ПС, то есть предполагает различную степень доверия к различным тестам. Это, в свою очередь, позволяет искать пути повышения характеристик надёжности П путём формирования тестов, обнаруживающих ошибки с высокой вероятностью.

Окончательный выбор модели для последующего оценивания характеристик надёжности исследуемых П производится как с учётом особенностей конкретной стадии их жизненного цикла, так и с учётом вычислительных, экономических, временных, квалификационных и других возможностей пользователя, «мощности» модели оценивания, типа и объёма оцениваемого ПС и качества ожидаемых им оценок.

Для упрощения выбора модели оценивания можно рекомендовать таблицу 5.1, пользуясь при этом следующим критерием:

$$KM = \max \sum_{i=1}^{n} \alpha_i b_i,$$

где KM – критерий выбора модели оценивания; α_i – весовой коэффициент i-го свойства рассматриваемой модели, выбираемой пользователем; b_i – коэффициент значимости i-го свойства рассматриваемой модели: $b_i = 0$, если i-м свойством модель не обладает, $b_i = 1$, если этим свойством модель обладает.

Таблица 5.1

Свойства моделей	Модели										
	Миллса	Двухгрупповая	Шумана	Седякина-Джелински-Моранды	Шика-Уолвертона	Мусы	Литлвуда-Веррйля	Дифференциальная Литтлвуда	Марковская	Нельсона	Холстеда
Машино-независимость	1	1	0	0	0	0	0	0	0	1	1
Возможность априорной оценки	0	0	0	0	0	0	0	0	0	0	1
Точность оценки	0	0	1	1	1	1	1	1	1	1	0
Возможность прогнозирования параметра	0	0	1	1	1	1	1	1	1	1	0
Точность прогнозирования «на хвосте»	–	–	0	0	0	1	1	1	0	0	–
Опасность искажения прогноза	–	–	–	0	0	–	–	–	1	0	–
Учёт состоятельности тестов	0	0	0	0	0	0	0	0	0	1	–
Устойчивость к порядку следования тестов	1	1	0	0	0	0	0	0	1	1	–
Учёт «волнового эффекта»	0	0	0	0	0	1	1	1	0	0	0
Учёт категорий ошибок	0	0	0	0	0	0	0	1	0	0	0
Инвариантность к потоку проявления ошибок	1	1	0	0	0	0	0	0	0	1	1
Инвариантность к потоку исправления ошибок	0	0	0	0	0	0	0	0	0	0	1
Независимость от предыдущего состояния	1	1	0	0	0	0	0	0	0	1	1
Отсутствие в модели случайных величин разного порядка случайности	0	1	1	0	0	0	0	0	1	0	1
Инвариантность к обнаруженным и устранённым ошибкам	0	0	0	0	0	0	0	0	0	1	1
Инвариантность к типу программного обеспечения	0	1	0	1	1	1	1	1	1	0	0
Инвариантность к уровню подготовки программиста	0	0	1	1	1	1	1	1	1	0	0

5.2. МЕТОДОЛОГИЧЕСКИЕ ПРИНЦИПЫ ОЦЕНИВАНИЯ ПОКАЗАТЕЛЕЙ НАДЁЖНОСТИ

Все методы оценивания показателей надёжности ПС условно можно подразделить на аналитические, имитационные и экспериментальные. Однако результаты исследований надёжности ПС аналитическими методами, как правило, не используются при

имитационном моделировании и статистической обработке экспериментальных данных, накопленных в процессе отладки, испытаний, фондирования и сопровождения ПС. При этом усложняется задача статистической обработки, так как, во-первых, значительное число ошибок устраняется в процессе отладки ПС, во-вторых, интенсивность эксплуатации и время выполнения одного и того же ПИ, поставляемого пользователям для работы в различных вычислительных системах, характеризуется достаточно значительным разбросом. Фондируемое ПС почти всегда работоспособно в основных режимах функционирования, что часто приводит к необходимости вычисления эксплуатационных показателей надёжности при малом и даже нулевом числе ошибок ПС.

Учитывая эти особенности расчёта показателей надёжности ПС можно предложить следующие специфические методологические принципы.

Принцип последовательного накопления информации об оценивании показателей надёжности на всех стадиях жизненного цикла ПС. Он позволяет создать «информационный шлейф» в виде автоматизированного банка данных о показателях надёжности и дать предпосылки к сокращению объёмов приёмочных испытаний и экспериментальных исследований при сопровождении ПИ.

Принцип применения информационно-ориентированных моделей и методов расчёта показателей надёжности. Принцип предполагает, что на каждой стадии жизненного цикла можно использовать информацию, полученную на данной или предыдущих стадиях. Например, если на стадиях разработки созданы только тексты ПС, то модель должна быть ориентирована на исходные данные, которые можно будет получить из текстов ПС. Когда производится отладка ПС, используемая модель должна учитывать статистику отладки, характеристики текстов П и т.д.

Принцип объединения оценок показателей надёжности, получаемых в процессе жизненного цикла ПС. Он позволяет повышать статистическую достоверность и информативность этих оценок. Например, при экспоненциальном законе распределения интервалов между ошибками в П $a(t) = \lambda \exp(-\lambda t)$ и равновероятном распределении значений параметра λ с плотностью вероят-

ности $\varphi(\lambda) = \dfrac{1}{\lambda_в - \lambda_н}$, где $\lambda_в$, $\lambda_н$ – верхняя и нижняя априорные оценки параметра ошибок, оценка информативности априорных данных показывает, что использование априорной информации эквивалентно увеличению объёма приёмочных испытаний ПС в $12\lambda_в/(\lambda_в - \lambda_н)$ раз [24].

Принцип методического единства моделей. Предполагает, что минимальный объём исходной информации, обеспечивающий получение оценок показателей надёжности, – это текст П на алгоритмическом языке и (или) её структурная схема. Следовательно, принципа приводит к необходимости ориентироваться на модели, построение которых принципиально возможно по тексту (структурной схеме) П. К подобным моделям можно отнести ГМП, ОГМП и ПРМ. Поэтому на основе рассмотренных принципов можно предложить следующую методологию определения показателей надёжности ПС, представленную в виде табл. 5.2.

Таблица 5.2

Обеспечение расчётов		Фазы и стадии ЖЦ								
		Разработка			Производство			Эксплуатация		
		Проектирование	Программирование	Испытания	Фондирование	Тиражирование	Поставка	Ввод в эксплуатацию	Сопровождение	Снятие с эксплуатации
Данные	Тексты программ		+	+	+	+	+	+	+	+
	Данные отладки		+	+						
	Данные испытаний			+	+	+	+	+		
	Эксплуатационные данные	+					+		+	
Модели	ГМП	+	+	+	+				+	
	ОГМП				+				+	
	ПРМ		+	+	+	+	+		+	
Методы	Аналитические	+	+		+			+	+	+
	Имитационные			+	+			+		
	Экспериментальные			+	+	+	+	+	+	+

Появление стадии фондирования, характерной для ПИ и ПП, привело к смещению акцентов применимости различных мо-

делей и методов расчёта показателей. Именно эта стадия сегодня оказалась слабо информационно обеспеченной, так как, как правило, известны только тексты программ, контрольные примеры и результаты приёмочных испытаний. В то же время указанные методологические принципы требуют, чтобы на стадии фондирования, когда принимается решение о переходе ПС в ПИ, был использован весь возможный арсенал моделей и методов. За счёт резкого увеличения ущерба от тиражирования возможных ошибок при производстве ПП ответственность стадии фондирования очень высока. Поэтому необходимо уделять особое внимание специфике применения моделей и методов именно на этой стадии жизненного цикла ПО.

5.3. ХАРАКТЕРИСТИКА МОДЕЛЕЙ ОТНОСИТЕЛЬНО ФАЗ ЖИЗНЕННОГО ЦИКЛА

Рассмотренные в главе 4 модели позволяют оценивать различные свойства ПО. Следует отметить, что существует несколько различных подходов и способов классификации моделей ПО [37, 76]. Приведённая в 5.1 классификация моделей не позволяет ответить на многие вопросы, относящиеся к их выбору на фазах ЖЦ. В данном разделе используем подход к группированию моделей, базирующийся на рассмотрении ПС, как объекта промышленного производства. Как и любой подобный объект, ПС имеет, по крайней мере, три основных фазы ЖЦ, отмеченных в главе 1: 1) разработка (проектирование), 2) производство, 3) эксплуатация.

Цели, математический аппарат и анализируемые характеристики при оценивании и моделировании могут существенно различаться для фаз ЖЦ. Каждая фаза характеризуется не только своими научно-техническими целями и задачами, но, в подавляющем большинстве случаев, имеет и организационные особенности. Это обуславливается отношениями заказчика и подрядчика, необходимостью отчуждения ПС от непосредственных его создателей и другими причинами. Поэтому вся совокупность рассмотренных моделей может быть условно разделена на три большие группы: 1) модели ПС для фазы разработки; 2) модели ПС для фазы производства; 3) модели ПС для фазы эксплуатации.

Внутри каждой группы можно при необходимости выделить подгруппы моделей, которые различаются по дополнительным признакам. При использовании моделей первой группы в настоящее время удаётся сделать выводы о предполагаемой сложности создаваемого ПС, возможном его объёме, сроках разработки, потенциальном количестве дефектов, в зависимости от выбираемой системы программирования и квалификации программистов. Используя получаемые оценки, заказчик может обоснованно определить и предполагаемый перечень организаций-подрядчиков для разработки ПС. Модели, по взаимному согласованию, разрабатываются либо заказчиком, либо подрядчиком, либо совместно.

Моделирование в рамках второй группы позволяет провести анализ качества ПС при выбранной конкретной системе программирования, известной квалификации и составе коллектива программистов, оптимизировать создаваемое ПС по тому или иному критерию, найти и устранить «слабые» звенья средства. Один из перспективных подходов к решению последней из перечисленных задач будет приведён в главе 8. Характерной особенностью является то, что на данной фазе моделирование и исследование свойств и характеристик ПС осуществляется специалистами, наиболее хорошо представляющими себе логику функционирования, структуру и взаимодействие как составных частей, так и всего ПС в целом. Следует, однако, заметить, что в отдельных случаях реальные результаты исследования могут быть скрыты от заказчика. Поэтому в соответствующих договорах можно предусмотреть контроль заказчика за процессом разработки ПС, предполагаемый совместный анализ качества ПС. Если же речь идёт о приобретении уже созданного средства, то приходится полагаться на гарантии, даваемые поставщиком, а также на эффективность приёмо-сдаточных испытаний и проверок.

Модели третьей группы создаются для проверки требований, предъявляемых к ПС, выявления особенностей функционирования АСУ в реальных условиях эксплуатации. При этом анализ качества ПС в большей степени осуществляется специалистами – представителями заказчика. В подавляющем большинстве случаев провести в сжатые сроки летальный анализ логики функционирования крупного программного комплекса, его струк-

туры не представляется возможным. Кроме этого, разработчиком часто применяются специальные меры, затрудняющие проведение подобного анализа. На данной фазе будущих пользователей ПС в первую очередь интересует ответ на вопрос, насколько пригодно ПС для решения целевых задач с требуемыми показателями точности, достоверности, оперативности и т.д.

Таким образом, цели и задачи моделирования ПС на каждой фазе ЖЦ различны. Но они в значительной мере дополняют друг друга. Использование результатов моделирования предыдущей фазы может привести к уточнению моделей текущей фазы ЖЦ. Наилучший эффект, как представляется, даёт совместное использование моделей всех трёх групп.

Программные средства АСУ имеют свои особенности, позволяющие выделить их в отдельный класс, как объекты исследования и моделирования. Приведём основные особенности.

1. Уникальность или, в редких случаях, малосерийность ПС. Это обусловлено высокой логической сложностью ПС. Крупные программные комплексы, на создание которых тратятся большие материальные и людские ресурсы, для одного типа АСУ не разрабатываются в двух и большем количестве вариантов. По этой же причине не представляется возможным объединять для анализа результаты исследования различных ПС, даже имеющих примерно одинаковую размерность и разработанных одними и теми же коллективами или организациями.

2. Процесс проявления ошибок ПС, приводящих к отказам АСУ имеет в подавляющем большинстве случаев нестационарный и случайный характер. Это обусловлено двумя причинами. Первая – физическая природа отказов АСУ по вине ПС. Она принципиально отличается от природы отказов технических средств. Аппаратура отказывает из-за физического разрушения электронных или механических элементов. Программы же физически не стареют и не изнашиваются. Поэтому устранение выявленной ошибки ПС приводит к отсутствию в дальнейшей эксплуатации отказов АСУ данного типа. Следовательно, меняются вероятностные закономерности потока отказов АСУ. Это приводит к его нестационарности.

Случайность процесса возникновения отказов АСУ из-за ошибок ПС обусловлена его высокой логической сложностью.

Число возможных комбинаций исходных данных при функционировании АСУ чрезвычайно велико.

Проверить все варианты исходных данных за приемлемое для практики время испытаний не представляется возможным. Поэтому проявление дефектов и ошибок ПС при функционировании АСУ имеет случайный характер.

Таким образом, использование классических методов математической статистики для исследования свойств ПС АСУ часто затруднено.

Рассмотрим общие принципы моделирования ПС АСУ.

5.3.1. Модели программных средств для фаз разработки и производства

Разработка моделей ПС данных фаз потребовала привлечения методов, используемых в различных отраслях знаний, в том числе в медицине и психологии. Методология моделирования базируется на интуитивном предположении о том, что чем сложнее ПС, тем труднее его спроектировать и создать. Возможное количество дефектов ПС и ряд других параметров ставится в прямую зависимость от логической сложности будущего ПС. В свою очередь, оценка сложности заказываемого ПС может быть уже сделана на начальной стадии ЖЦ. Исходными данными для расчётов могут служить требования, содержащиеся в техническом задании на разработку ПС. Модели рассматриваемой группы базируются на метрической теории программ М.Х. Холстеда [68].

Идея Холстеда основывается на учёте особенностей мыслительной деятельности программистов при разработке программ. К настоящему времени в рамках этой теории разработано большое количество метрик, позволяющих оценить ряд свойств ПС АСУ.

Предполагается возможным для любой П определить и практически оценить величину следующих метрик:

– число различных операций, осуществляемых в данном ПС;

– число различных операндов ПС.

Основываясь на исходных метриках, можно получить ряд производных метрик от них. Так, Холстед предложил методику расчёта многих метрик П. К основным относятся следующие:

1) объём будущей П в битах;

2) потенциальный объём П, представляющей собой потенциальный минимальный возможный объём ПС с заданным интерфейсом, который может быть реализован программистом очень высокой квалификации на гипотетическом языке программирования, близкого к разговорному;

3) работа по программированию, которая требуется, как суммарное число элементарных мысленных различений, необходимых для генерации П;

4) время, необходимое для разработки П;

5) ожидаемое число дефектов ПС, которое оно может содержать до начала эксплуатации.

Исходными данными для расчётов являются параметры будущего ПС, которые можно задать заранее. К ним относятся число входных и выходных переменных, число различных конструкций языка программирования, число требуемых констант и некоторые другие величины.

Учёт квалификации программистов и способности человека совершать ошибки при мыслительной деятельности осуществляется путём использования в расчётах психофизиологических параметров. К ним относятся, например, параметр Страуда, психофизиологическая гипотеза Джорджа Миллера «7 + 2» и некоторые другие. Параметр Страуда характеризует время, необходимое человеческому мозгу для выполнения элементарной мыслительной операции («различения» по Холстеду). Численные значения этой константы находятся в пределах от 5 до 20 различений в секунду. Гипотеза Миллера утверждает, что мозг человека может обрабатывать в своей «сверхбыстрой» памяти одновременно и безошибочно лишь 7 + 2 объекта. Холстедом показано с использованием данной гипотезы, что в среднем после обработки 24 битов абстрактной информации на языке высокого уровня (близкого к разговорному) человек совершает ошибку (появляется дефект в программе). Точность прогноза, получаемого с помощью моделей, базирующихся на теории Холстеда, находится в пределах 10…12%. Для программного проектирования эту величину можно считать удовлетворительной. Необходимо отметить, что к настоящему времени метрические модели – единственный класс моделей, с помощью которых оцениваются свойства ПС на фазе разработки ЖЦ ПС.

Метрики П широко применяются и на фазе производства. В ходе разработки накапливается информация, в частности, о конкретном языке и используемых стандартах программирования, квалификации программистов, структуре ПО и другие данные. Поэтому появляется возможность более глубокого анализа ПС.

Плодотворно развиваются направления по разработке моделей ПС, базирующиеся на анализе текстов П, их структуры, взаимодействия с техническими средствами и т.д. Однако недостатком этих моделей является отсутствие достоверных данных о поведении ПС и АСУ в целом при их функционировании в реальных условиях эксплуатации. Поэтому значительный объём научно-технических публикаций посвящён рассмотрению проблемы моделирования ПС в фазе эксплуатации.

5.3.2. Модели программных средств для фазы эксплуатации

Модели для фаз разработки и производства имеют важное значение. С их помощью можно оценивать ряд свойств ПС в условиях значительной неопределённости и ограниченности информации. Однако при функционировании АСУ поступает достоверная информация о проявлении дефектов ПС. Поэтому значительные усилия исследователей направлены на математическое описание потоков отказов АСУ, происходящих по вине ПС.

Реализации этих потоков могут характеризовать важнейшее свойство АСУ для фазы эксплуатации.

Поток ошибок ПС с учётом его особенностей, как объекта исследования, имеет следующие свойства:

1) поток ошибок ПС имеет только одну реализацию;

2) поток ошибок является случайным нестационарным потоком;

3) поток ошибок ПС имеет тенденцию к редению.

Исходя из этих свойств, можно сделать вывод о том, что ПС относятся к классу объектов, улучшающих с течением времени своё качество, как объектов с самообучением. Некоторые положения теории систем с самообучением используются для разработки моделей надёжности ПС на этапе эксплуатации. Идея моделирования состоит в следующем.

Пусть необходимо описать закономерность появления событий потока, удовлетворяющего перечисленным свойствам. Предварительно выдвигается гипотеза о виде закона распределения времени между смежными событиями. Полагается чаще всего, что закон распределения один и тот же для всех интервалов времени, но параметры этого закона изменяются с увеличением номера события таким образом, чтобы выполнялось третье свойство данного потока. Далее, при совместном рассмотрении интервалов между смежными событиями потока находятся оценки величин, входящих в выражения для параметров распределения. Такой подход оказался достаточно плодотворным. При этом модели различаются видом закона распределения интервалов между смежными событиями, а также различными допущениями при моделировании.

Рассмотрим основные допущения, которые используются при моделировании ПС на фазе эксплуатации:

1) каждая ошибка ПС обусловлена только одним его дефектом;

2) отыскание дефекта, вызвавшего ошибку, и его устранение осуществляются мгновенно;

3) при устранении дефекта не может появиться новой ошибки им обусловленной;

4) распределения длительностей интервалов времени между смежными ошибками известны;

5) параметры закона (законов) распределения изменяются при каждом выявлении дефекта (ошибки) таким образом, чтобы имело место улучшение качества ПС.

Можно отметить, что некоторые допущения при моделировании могут быть ослаблены.

Однако указанные допущения в целом следует рассматривать как весьма ограничительные. В будущем желательно постепенно снижать высокую жёсткость данных ограничений. При этом мы рекомендуем больше обращаться к характеристикам различных потоков событий, рассмотренных нами в главе 2, прежде чем получать значения показателей надёжности каких-то конкретных ПС.

Глава 6

ПРОСТЕЙШИЕ МЕТОДЫ РАСЧЁТА НАДЁЖНОСТИ

6.1. АПРИОРНОЕ ОЦЕНИВАНИЕ НАДЁЖНОСТИ ПРОГРАММНОГО МОДУЛЯ

Метод применим на начальных стадиях производства ПП, требует минимума априорной информации и работает при отсутствии статистических данных. Он позволяет, используя лексикографический подход, основанный Холстедом [68], осуществлять предварительное оценивание надёжности ПМ при проведении статистического анализа его текста.

Описание информационной структуры ПМ в виде взаимосвязанных множеств исходных данных, преобразований и допустимых результатов вычислений показывает, что в зависимости от типа дефекта и его расположения могут в значительной степени изменяться величина поражения элементарных ПИ (например, оператор или их последовательность) и в целом свойства всей совокупности элементарных ПИ (весь ПМ), а значит, и истинное значение показателя надёжности ПМ.

Для оценивания надёжности ПМ необходимо определить количество дефектов δ, внесённых в текст ПМ на стадии программирования, рассчитать среднее число поражаемых элементарных (ЭПИ) при различном расположении дефектов в тексте ПМ и соотнести их к общему числу возможных ЭПИ.

Предполагается, что разработка ПМ производится по готовому алгоритму с использованием некоторого языка программирования. Реализация алгоритма состоит в выборе операторов и операндов данного языка. Формируются четыре основные метрические характеристики ПМ, которые могут быть найдены подсчётом из текста программы.

В качестве таких метрических характеристик используются:

– количество различных операторов в П ε_1;

– количество различных операндов в П ε_2;

– количество всех операторов в П N_1;

– количество всех операндов в П N_2.

Для нахождения метрических характеристик могут использоваться также автоматизированные системы анализа П [64].

Определяется ёмкость словаря $\varepsilon = \varepsilon_1 + \varepsilon_2$ и длина П $N = N_1 + N_2$. Процесс выбора элементов из словаря и их упорядоченное расположение в соответствии с заданным алгоритмом расчленяется на определённое число мысленных различений, некоторая часть которых является ошибочной и приводит к образованию дефектов в ПМ. Для оценивания начального количества дефектов δ предлагается модель, основанная на анализе текста ПМ и уровня используемого языка программирования Λ.

Модель предполагает, что мозг человека может производить в среднем E_0 элементарных мысленных различений до возникновения ошибки. В этом случае число внесённых дефектов в П определяется как $\delta = \Lambda \dfrac{E}{E_0}$, где Λ – уровень языка программирования, являющийся мерой его избыточности; E – общее число элементарных мысленных различений, необходимых для написания П. В простейшем случае уравнения для указанных величин имеют вид [68]:

$$\begin{aligned} \delta &= E^{2/3} / 3000, \\ E &= \varepsilon_1 N_2 N \log_2 \varepsilon_1 / 2\varepsilon_2. \end{aligned} \qquad (6.1)$$

Внесённые в текст ПМ дефекты при функционировании П могут приводить к нарушению условий реализации маршрутов обработки данных, то есть к ошибкам. Если за время функционирования ПМ будут реализованы все маршруты обработки данных, то значение показателя в виде вероятности безошибочной реализации ПМ определится так:

$$P_1 = 1 - \frac{L_n}{L}, \qquad (6.2)$$

где L_n – число поражённых дефектами маршрутов обработки данных, L – общее число маршрутов.

В общем случае число маршрутов обработки данных может быть большим, определяемом, в основном, диапазоном изменения входных переменных ПМ и длиной разрядной сетки СВТ. К тому же анализ возможных типов вносимых дефектов и степень их влияния на работоспособность П [5] говорит о том, что наиболее часто встречающиеся и наименее поддающимися отладке являются логические дефекты. Поэтому целесообразно ограничиться рассмотрением логической структуры ПМ, то есть маршрутами принятия логических решений и преобразования логических переменных.

Поскольку точное расположение дефектов в логической структуре П остаётся в значительной степени неизвестным, то информация о количестве поражённых дефектами маршрутов принятия логических решений характеризуется высокой степенью неопределённости. Поэтому уместно предположить, что дефекты равновероятно могут поражать каждый оператор ветвления (условие-предикат), определяющий тот или иной маршрут. Тогда в качестве оценки показателя надёжности ПМ может быть использована зависимость

$$P_2 = 1 - \frac{L^{(\delta)}_{n_{\text{ср}}}}{L}, \qquad (6.3)$$

где $L^{(\delta)}_{n_{\text{ср}}}$ – среднее значение числа поражённых маршрутов при условии наличия в ПМ δ дефектов.

Логическую структуру ПМ можно отобразить управляющим графом, вершинами которого являются операторы ветвления, а дугами – последовательности вычислительных операторов:

$$\Gamma = (M, n_{\text{вх}\,m}, n_{\text{вых}\,m}), \quad m = \overline{1, M}, \qquad (6.4)$$

где M – общее число вершин графа; $n_{\text{вх}\,m}$ – число дуг, входящих в m-ю вершину; $n_{\text{вых}\,m}$ – число дуг, исходящих из m-й вершины.

Управляющий граф может быть описан матрицей переходов, в которой на пересечении i-й строки и j-го столбца располагается единица, если из вершины i можно перейти к вершине j за один шаг, и нуль – в противном случае. Если задана матрица переходов $C = \|c_{ij}\|$, $i, j = \overline{1, M}$, то количество входных и выходных дуг для m-й вершины может быть определено как $n_{\text{вх}\,m} = \sum_{i=1}^{M} c_{im}$, $n_{\text{вых}\,m} = \sum_{j=1}^{M} c_{mj}$.

Общее число маршрутов принятия логических решений ПМ является функционалом следующего вида:

$$L = \Phi_1(\Gamma, n_{\text{вх } m}, n_{\text{вых } m}), \quad m = \overline{1, M} . \qquad (6.5)$$

Для расчёта общего числа маршрутов воспользуемся алгоритмом [5], а исходными данными являются: матрица переходов $C = \|c_{ij}\|$, $i, j = \overline{1, M}$; вектор входов ПМ $c_{\text{вх}}(j)$, $j = \overline{1, M}$, определяющий номер вершин управляющего графа П, на которые поступают входные данные от внешних абонентов; вектор выходов ПМ $c_{\text{вх}}(j)$, $j = \overline{1, M}$, определяющий номера вершин управляющего графа, с которых выдаются результаты обработки.

Алгоритм основан на преобразовании матрицы переходов $C = \|c_{ij}\|$ в матрицу реализации $S = \|s_{ij}\|$, $i, j = \overline{1, M}$ каждый элемент которой представляет собой число маршрутов обработки, проходящих через *j*-ю вершину графа в *i*-м направлении. Содержание алгоритма:

Определяется число дуг, входящих в произвольную *j*-ю вершину $n_{\text{вх } j}$.

Формируется матрица реализаций $S = C_{n_{\text{вх } j}}$.

Операции 1 и 2 повторяются *M* раз, то есть для каждой вершины графа.

Определяется общее число маршрутов принятия логических решений $L = \sum_{i \in C_{\text{вх}}(j)} \sum_{j=1}^{M} S$, в котором S – есть функция от j.

Если в структуре ПМ имеется один дефект, который приводит к некорректному исполнению одного j-го оператора ветвления, месторасположение которого в структуре ПМ случайно, это приведёт к различному числу поражённых дефектом маршрутов, которое в общем случае определится из выражения:

$$L_{nj}^{(1)} = \Phi_2(\Gamma, n_{\text{вх } m}, n_{\text{вых } m}), \quad m = \overline{1, M} , \qquad (6.6)$$

то есть является числом маршрутов, выходящих из j-й вершины. Следовательно, для расчёта числа поражённых дефектом маршрутов может быть использован вышеприведённый алгоритм расчёта общего числа маршрутов, но в этом случае в качестве входной рассматривается поражённая дефектом *j*-я вершина графа.

Среднее число маршрутов, поражённых дефектом, для данного управляющего графа П может быть найдено суммированием всех поражённых маршрутов по каждой вершине графа и осреднением по общему числу вершин:

$$L_{\Pi_{\text{ср}}}^{(1)} = \frac{1}{M} \sum_{j=1}^{M} L_{nj}^{(1)} . \qquad (6.7)$$

Снятие ограничения на число дефектов в структуре ПМ приводит к тому, что каждый вновь рассматриваемый дефект может поражать часть маршрутов, которые уже были уже поражены ранее рассмотренным дефектом. Это ведёт к частичному «поглощению» дефектов, то есть к снижению роста числа поражаемых маршрутов с ростом числа дефектов. Эффект «поглощения» дефектов проявляется тем сильнее, чем выше связность графа П. Неучёт эффекта «поглощения» ведёт к методической ошибке, выражающейся в значительном занижении оценки показателя надёжности ПМ.

Точное значение количества маршрутов, поражённых δ дефектами, получить достаточно сложно, так как требует рассмотрения всех возможных сочетаний дефектосодержащих вершин графа. Приведём приближённое решение данной задачи, обладающее, тем не менее, достаточной точностью для получения статистической оценки показателя надёжности ПМ и сравнительно просто реализуемое.

Решение основывается на предположении, что снижение роста области поражения маршрутов каждым последующим дефектом пропорционально относительному сокращению области непоражённых маршрутов. Поэтому среднее число маршрутов, поражаемых вторым дефектом, определится из выражения:

$$L_{\Pi_{\text{ср}}}^{(2)} = L_{\Pi_{\text{ср}}}^{(1)} \frac{L - L_{\Pi_{\text{ср}}}^{(1)}}{L} , \qquad (6.8)$$

а среднее число поражённых маршрутов при δ дефектосодержащих вершинах графа определится как:

$$L_{\Pi_{\text{ср}}}^{(\delta)} = L_{\Pi_{\text{ср}}}^{(1)} \sum_{i=1}^{\delta} \left(\frac{L - L_{\Pi_{\text{ср}}}^{(1)}}{L} \right)^{i-1} . \qquad (6.9)$$

Тогда окончательное выражение для статистической оценки показателя надёжности ПМ имеет вид:

$$P_3 = 1 - \frac{L^{(\delta)}_{\Pi_{ср}}}{L}. \qquad (6.10)$$

Метод априорного оценивания надёжности ПМ построен на достаточно общих допущениях о характере распределения дефектов в структуре ПМ и их влиянии на надёжность его функционирования, что предполагает приближённые оценки показателей надёжности. Однако метод может быть использован для анализа правильности выбранной структуры ПМ до начала работы по кодированию и отладке. Кроме того, полученные данным методом результаты могут быть использованы в качестве априорной информации при проведении оценивания надёжности ПМ в ходе отладки, испытаний или эксплуатации. Это говорит о повышении степени достоверности получаемых на более поздних стадиях жизненного цикла оценок надёжности ПМ за счёт учёта информации об особенностях структурной организации ПМ.

6.2. ОЦЕНИВАНИЕ НАДЁЖНОСТИ ПРОГРАММНОГО МОДУЛЯ НА ОСНОВЕ ОТЛАДКИ

Сложное ПО любого объекта управления представляет собой комплекс взаимосвязанных между собой ПМ. Каждый их ПМ выполняет некоторую частную задачу, входящую функционально в общую задачу управления.

Естественно, с точки зрения надёжности функционирования всего комплекса, отдельный ПМ следует рассматривать как элемент расчёта надёжности. Зная показатели надёжности всех ПМ, входящих в комплекс, можно аналитическим или экспериментальным путём найти оценку надёжности комплекса. Принятие ПМ в качестве элемента расчёта надёжности, конечно, носит условный характер. Во-первых, само понятие ПМ условно. Как правило, это часть сложной П, имеющей функциональную законченность, обособленность. Во-вторых, если ПО объекта является весьма сложным, то в качестве элемента расчёта надёжности следует выбирать не один ПМ, а их некоторую совокупность, удобную для производства расчёта. В-третьих, если сам ПМ является объектом расчёта надёжности, например, более углублённого, то в качестве элемента расчёта может быть выбран отдельный оператор модуля, его часть и т.д. Таким образом, элемент

расчёта надёжности ПО выбирается исследователем, исходя из удобства оценивания надёжности и в общем носит условный характер. В дальнейшем будем отождествлять понятие элемента расчёта с ПМ.

Поставим следующую задачу. Требуется определить вероятностные показатели безошибочного функционирования некоторого ПМ по результатам его отладки, проводимой до использования модуля по его целевому назначению на объекте [3]. Отладка выполняется следующим образом. Случайно выбираются значения исходных данных ПМ, при которых производятся вычисления на СВТ. Если до окончания вычислений ошибки при функционировании ПМ не происходит, что может определяться с помощью системы контроля, то случайным образом снова выбираются значения входных переменных ПМ, вновь производятся вычисления, определяется достоверность результата вычислений и процесс повторятся далее. Если в процессе вычислений до истечения их окончания фиксируется ошибка, то запоминается величина промежутка времени от начала вычислений до момента возникновения ошибки. Причина ошибки определяется, сама ошибка устраняется (в дальнейшем предполагается мгновенно), а новые ошибки не вносятся в ПМ. Затем вновь случайным образом выбираются значения входных переменных ПМ и процесс повторяется до тех пор, пока не будет получена достаточно представительная совокупность результатов испытаний для оценивания надёжности ПМ.

Введём следующие обозначения: t_u – длительность одного успешного прогона П модуля на СВТ при фиксированных исходных данных; t_i – промежуток времени от начала i-го прогона до установления события ошибки в данном прогоне П; n – общее число прогонов П; r – число прогонов из n, в которых имели место ошибки.

Тогда, в простейшем случае, предполагая, что время до ошибки распределено по экспоненциальному закону с помощью метода максимального правдоподобия получим для среднего времени безошибочного функционирования ПМ следующую оценку:

$$\widetilde{T} = \frac{\sum_{i=1}^{r} t_i + (n-r)t_u}{r}. \qquad (6.11)$$

Если длительности успешных прогонов П будут различны, то в (6.11) вместо $(n-r)t_u$ следует записать их сумму.

Оценка интенсивности возникновения ошибки при прогоне П определяется как обратная величина для $\widetilde{T}$:

$$\widetilde{\lambda} = 1/\widetilde{T}. \qquad (6.12)$$

Располагая (6.11), (6.12), можно построить зависимости среднего времени безошибочного функционирования и интенсивности ошибки ПМ от времени отладки $t_o \approx nt_u$ или числа n прогонов П модуля. Экстраполируя эти зависимости, можно предсказать значения данных показателей для более длительных периодов отладки. Однако точность подобного прогноза будет невелика. Кроме того, эти показатели надёжности являются усреднёнными, носящими вспомогательный характер и поэтому, как известно из теории надёжности, могут выполнять вспомогательную роль при оценивании надёжности.

Обычно, в качестве основного показателя используется вероятность безошибочной работы ПМ за требуемое время его функционирования при управлении объектом.

В моделях надёжности ПО часто предполагают, что количество оставшихся в ПМ ошибок убывает монотонно с ростом времени отладки, а скорость их выявления пропорциональна числу оставшихся ошибок. При этих предположениях зависимость числа появления ошибок от времени отладки, как часто полагают, является экспоненциальной, хотя в общем случае это не имеет никакого основания. Для подтверждения допущения об «экспоненциальности» числа оставшихся в П ошибок ссылаются на испытания ПО различной структуры и назначения. Однако, например, для ПО операционных систем наблюдается аномальное отклонение от экспоненциального закона при малом времени отладки. Обычно это объясняют растягиванием сроков ввода ПО у ряда пользователей. Можно указать и на другие примеры уклонений, но мы этого делать не будем, оставив для анализа самую простейшую модель.

При условии, что все обнаруженные ошибки исправляются мгновенно и при исправлении не вносятся новые, принимают общее начальное количество ошибок в ПМ постоянным. Обозначим его E_o. Пусть ПМ содержит I команд. Тогда количество остающихся в ПМ ошибок можно представить так:

$$\varepsilon_o(t_o) = \frac{E_o}{I} - \varepsilon_u(t_o), \qquad (6.13)$$

где $\varepsilon_u(t_o)$ – число исправленных ошибок за время отладки. Обычно полагают, что $\varepsilon_u(t_o) < E_o / I$, то есть $\varepsilon_r(t_o) > 0$ всегда. Однако, поскольку фактически большинство ПМ достигает вполне отлаженного состояния, можно предположить, что для больших значений t_o величина $\varepsilon_r(t_o)$ остаётся малой. Примерный вид зависимости (6.13) показан на рис. 6.1. На рис. 6.2 показаны графики $P(t)$.

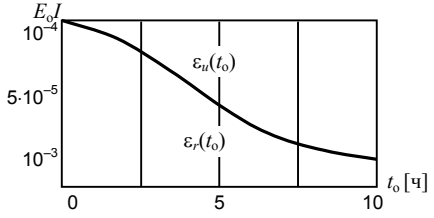

Рис. 6.1

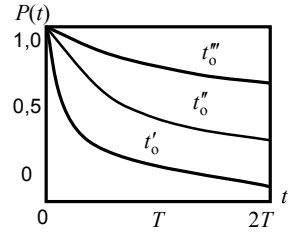

Рис. 6.2

Пусть необходимое время функционирования ПМ равно t. Тогда условная вероятность ошибки для ПМ в интервале [t, $t + \Delta t$), найденная при условии, что до момента t ошибки не было, определится по формуле:

$$P(t, \Delta t) = \lambda(t)\Delta t \equiv k\varepsilon_r(t_o)\Delta t, \qquad (6.14)$$

где k – некоторая постоянная, $\varepsilon_r(t_o)$ – количество оставшихся ошибок после отладки ПМ в течение времени t_o. При принятых допущениях об экспоненциальном распределении времени до ошибки вероятность её отсутствия равна:

$$P(t) = e^{-\lambda t} = e^{-k\varepsilon_r(t_o)t}. \qquad (6.15)$$

На рис. 6.2. показаны графики этой вероятности при трёх различных значениях времени отладки ПМ, принято, что $t'''_o > t''_o > t'_o$.

Среднее время безошибочного функционирования ПМ равно:

$$T(t_o) = \frac{I}{k[E_o - I\varepsilon_u(t_o)]}. \qquad (6.16)$$

Выражения (6.15), (6.16) оказываются наиболее удачными для использования на практике при оценивании надёжности ПМ. Они просты по своему смыслу и, кроме того, учитывают влияние времени отладки ПМ на показатели надёжности его функционирования.

Определение длительности отладки ПМ, достаточной для достижения заданного среднего времени его безошибочной работы (или заданного значения вероятности безошибочной работы ПМ для времени его функционирования t), возможно после определения неизвестных величин k и E_o. Находятся значения этих величин следующим образом. Пусть для двух различных периодов отладки t'_o и t''_o определены соответствующие значения числа исправленных ошибок в ПМ, равные $\varepsilon_u(t'_o)$ и $\varepsilon_u(t''_o)$. Тогда можно вычислить интенсивности ошибки для каждого периода отладки:

$$\lambda_1 = \frac{k[E_o - I\varepsilon_u(t'_o)]}{I}, \quad \lambda_2 = \frac{k[E_o - I\varepsilon_u(t''_o)]}{I}. \qquad (6.17)$$

Решая (6.17), найдём:

$$E_o = \frac{I[\eta\varepsilon_u(t'_o) - \varepsilon_u(t''_o)]}{\eta - 1}, \qquad (6.18)$$

где $\eta = \lambda_1 / \lambda_2$. Значение k может быть найдено подстановкой (6.18) в одну из формул (6.17). Получим

$$k = \lambda_1 / [E_o / I - \varepsilon_u(t'_o)] = \lambda_2 / [E_o / I - \varepsilon_u(t''_o)]. \qquad (6.19)$$

Если $\varepsilon_u(t'_o) = \varepsilon_u(t''_o)$, то в выражении (6.18) будем иметь неопределённость 0/0, которую необходимо раскрыть. Но практически этот случай не имеет смысла.

В данном методе предполагалось, что число ошибок в ПМ неслучайное. На самом деле любому типовому ПМ соответствует своё распределение вероятностей случайной величины – числа ошибок в модуле. Поэтому можно говорить об определённом числе ошибок только в соответствии с определённой вероятностью. Для этого случая изложенный метод оценивания надёжности нуждается в дальнейшем развитии.

Замечание. Если прогоны ПМ статистически независимы, что довольно часто наблюдается на практике, то предположение об экспоненциальном распределении времени до ошибки может быть снято. Допустим, что сделано предположение в пользу нормального распределения времени до ошибки. В этом случае нужно найти методом максимального правдоподобия оценки для среднего времени и среднеквадратического отклонения от него и поступать таким же образом, как было изложено в данном разделе. Эту задачу мы оставляем для рассмотрения читателю.

Глава 7

ПЛАНИРОВАНИЕ ОТЛАДОЧНЫХ ИСПЫТАНИЙ НА ОСНОВЕ ОЦЕНИВАНИЯ НАДЁЖНОСТИ

Отладочные испытания преследуют цель – достигнуть заданный уровень надёжности ПО.

Рассмотрим применение одной из моделей роста надёжности – модель Л.И. Волкова. Пусть $R(j)$ – вероятность успешного однократного прогона П после устранения в ней j ошибок. Она равна (4.17):

$$R(j) = R_\infty - (R_\infty - R_0)\left(1 - \frac{\alpha}{R_\infty}\right)^j. \qquad (7.1)$$

Требуется достигнуть для П заданный уровень вероятности её безошибочного функционирования $R_{\text{Тр}}$. Для этого необходимо определить объём испытаний в прогонах, который позволил бы вывести испытуемую П по вероятности безошибочного функционирования на уровень $R_{\text{Тр}}$. Из формулы (7.1) получим:

$$\frac{R_\infty - R(j)}{R_\infty - R_0} = \left(1 - \frac{a}{R_\infty}\right)^j, \; j = \frac{\ln(R_\infty - R(j)) - \ln(R_\infty - R_0)}{\ln(1 - a/R_\infty)}. \qquad (7.2)$$

Номер ошибки j соответствует числу выполненных прогонов П h_j после устранения ошибки с номером $j-1$. Тогда число испытаний П в зависимости от номеров устранённых ошибок можно представить в виде рис. 7.1.

На рисунке сплошная кривая находится по методу наименьших квадратов с аппроксимацией кривой параболической функцией $N(k) = g + fk^2$, где g – усреднённое значение числа

испытаний в группе, а f – усреднённая скорость роста числа испытаний в группе.

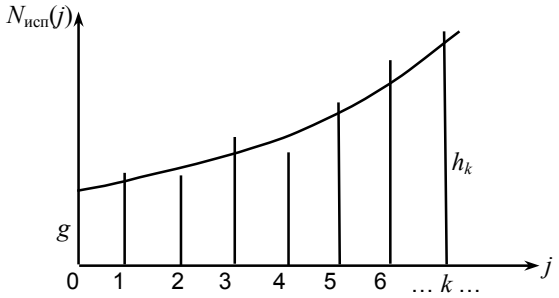

Рис. 7.1

Далее составляется сумма квадратов отклонений числа испытаний от наблюдённых значений $Q(j)=\sum_{k=1}^{j}(h_k-g-fk^2)^2$. Для отыскания оценок $\hat{g}$ и $\hat{f}$ нужно взять частные производные $\frac{\partial Q}{\partial g}$, $\frac{\partial Q}{\partial f}$, приравнять их к нулю и решить полученную систему уравнений:

$$\begin{cases}\sum_{k=1}^{j}(h_k-\hat{g}-\hat{f}k^2)=0,\\ \sum_{k=1}^{j}k^2(h_k-\hat{g}-\hat{f}k^2)=0.\end{cases} \qquad (7.3)$$

Из первого уравнения (7.3) найдём $\hat{g}=\frac{1}{j}\sum_{k=1}^{j}(h_k-\hat{f}k^2)$, а после подстановки его во второе уравнение и выполнения окончательного решения получим:

$$\hat{g}=\frac{C_2C_3-C_1}{C_1^2+jC_3},\quad \hat{f}=\frac{C_1C_2-jC_4}{C_1^2+jC_3},\quad C_1=\sum_{k=1}^{j}k=\frac{j(j+1)}{2},$$

$$C_2=\sum_{k=1}^{j}h_k,\quad C_3=\sum_{k=1}^{j}k^3,\quad C_4=\sum_{k=1}^{j}kh_k.$$

Необходимое число приращений количества испытаний ΔN П, связанное с числом дополнительно устранённых ошибок в ней Δj, определяется как $\Delta N=N(j+\Delta j)-N(j)$. Поэтому

$$\Delta N_{\text{Тр}}=N_{\text{Тр}}-N(j)=\sum_{k=1}^{j+\Delta j}(\hat{g}+\hat{f}k^2)=\sum_{k=1}^{\Delta j}(\hat{g}+\hat{f}(j+k)^2\ . \qquad (7.4)$$

Таким образом, чтобы достичь требуемую вероятность безошибочного функционирования П, равную $R_{\text{Тр}}$, необходимо в среднем, отлаживать её до тех пор, пока дополнительно в ней не будет устранено ещё Δj ошибок (рис. 7.2).

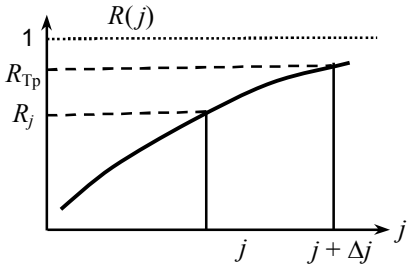

Рис. 7.2

Представляет интерес решение распределительной задачи для многомодульного программного комплекса. Пусть требуется достичь заданной вероятности его функционирования. Как распределить между модулями требуемые вероятности их функционирования? Каким образом обеспечить отладку каждого модуля, чтобы достичь для него заданную вероятность? Эту задачу надо решать как для независимых модулей, таки при наличии зависимости между модулями.