

Исследование методов тестирования программных модулей обработки полетной информации

Программное обеспечение контроля полетов используется для решения задач контроля при эксплуатации ВС, а также при расследовании летных происшествий. Результаты решения этих задач непосредственно связаны с обеспечением безопасности полетов и поэтому ПО КП должно удовлетворять требованиям полноты решения задач воспроизведения и контроля. Требования к ПО КП изложены в нормативных документах [1], [2] и могут быть представлены в виде следующих групп:

1. решение перечня обязательных задач;
2. состав и форма представления входной и выходной информации;
3. точность представления информации и достоверность результатов контроля.

Основным средством проверки соответствия ПО КП вышеприведенным требованиям является тестирование. Тестирование позволяет обнаружить ошибки в программах и программных комплексах, а в случае отсутствия ошибок тестируемое ПО признается работоспособным. ПО КП будем считать надежным, если оно удовлетворяет требованиям, изложенным в документах [1], [2], [3].

Программные комплексы контроля полетов достаточно сложны, поэтому целесообразно проводить их тестирование поэтапно: модуль → программа → подсистема → система → комплекс. В данной работе рассмотрены вопросы тестирования модулей, входящих в состав основных программных комплексов контроля полетов.

1. Программные комплексы контроля полетов

Программное обеспечение контроля полетов состоит из трех основных программных комплексов [3]:

- комплекс программ воспроизведения полетной информации;
- комплекс программ допускового контроля полета;
- комплекс программ контроля качества выполнения полета.

Комплекс программ воспроизведения ПИ решает задачу обработки копии полета, которая заключается в графическом и табличном представлении изменения аналоговых параметров и разовых команд во времени с возможностью анализа как всей копии, так и ее фрагментов ([2], [3]). Соответствующий комплекс программ включается в любую АСКП и состоит из ряда программ и подпрограмм.

Комплекс программ допускового контроля обеспечивает реализацию всех алгоритмов контроля, определенных в эксплуатационной документации на ВС. Результаты решения задач контроля используются для принятия заключения о нахождении объекта контроля "ВС-экипаж-среда" в том или ином состоянии. Программы комплекса предназначены для выявления опасных отклонений в работе систем ВС и ошибок пилотирования, которые могут влиять на безопасность полета [4].

Комплекс программ контроля качества полета является программной системой, которая служит для вычисления показателей качества как каждого элемента полета, так и всего полета в целом. Для решения задач контроля качества выполнения полета программное обеспечение комплекса должно обладать следующими функциями:

- формирование информационного "портрета" этапов полета;
- определение значений показателей качества пилотирования.

2. Тестирование модулей обработки полетной информации

В данном разделе приводится краткий обзор методов тестирования и даются рекомендации по использованию каждого из рассмотренных методов при тестировании программных модулей, входящих в состав различных комплексов контроля полетов.

Как правило, методы тестирования программ, как отдельных модулей, относят либо к стратегии "черного ящика", либо к стратегии "белого ящика"[6].

При использовании стратегии "черного ящика" программа рассматривается как "черный ящик". Есть еще два синонима, обозначающих эту стратегию: тестирование с управлением по данным и тестирование с управлением по входу – выходу.

При тестировании программы по методу "черного ящика", на ее вход подаются различные тестовые наборы, а на выходе производится проверка: не произошло ли отклонений по выходным данным программы от описанных в ее спецификациях.

Напротив, при тестировании программ по стратегии "белого ящика" программа рассматривается как "белый ящик", а стратегия имеет второе название – "тестирование, управляемое логикой программы". Для проведения тестирования по стратегии "белого ящика", необходимо иметь тексты программы на языке высокого уровня, а тестовые данные получают путем анализа логической организации программы.

К стратегии "черного ящика" относятся следующие методы:

- эквивалентное разбиение;
- анализ граничных значений;
- метод функциональных диаграмм;
- метод предположения об ошибке;
- динамическое тестирование;
- стохастическое тестирование.

К стратегии "белого ящика" принадлежат такие методы:

- покрытие операторов;
- покрытие решений/условий;
- комбинаторное покрытие условий;
- статическое тестирование;
- формальная верификация;
- доказательство правильности программ.

При использовании метода эквивалентного разбиения проводится разбиение входной области данных программы на некоторое конечное число классов эквивалентности таким образом, что если один тест класса обнаруживает ошибку, то и другие члены класса должны обнаруживать эту же ошибку (классы эквивалентности не пересекаются).

Метод анализа граничных условий является логическим развитием метода эквивалентных разбиений и характеризуется следующими положениями:

1. элементы класса эквивалентности выбираются таким образом, чтобы можно было протестировать границы этого класса;
2. при разработке тестов рассматриваются как входные условия, так и множество результатов (выходные классы эквивалентности).

Метод функциональных диаграмм характеризуется набором алгоритмических правил для выбора значений операндов в операторах программы с определением ожидаемых выходных значений для каждого применяемого теста и является довольно громоздким в реализации.

При применении метода предположения об ошибке составляется список возможных ошибок, а также особых ситуаций, в которых предположительно могут появиться ошибки. Затем, используя этот список, составляются соответствующие ТНД. В определении возможных ошибок может послужить углубленный анализ спецификаций на программу (определяются моменты, не описанные в спецификациях [7]).

Основным содержанием методов динамического тестирования является выполнение тестов программами, приведенными к виду загрузочных модулей. Среди этих методов особенно сложным представляется метод детерминированного динамического тестирования, при котором контролируются все возможные комбинации исходных данных, что позволяет выявить отклонения в результатах работы программы от эталонных значений, обозначенных в спецификациях [5]. Но на практике, уже в программах средней сложности перебрать все комбинации входных данных оказывается невозможным. В связи с этим широкое распространение получило стохастическое динамическое тестирование, при котором тестовые данные генерируются в виде множеств случайных величин с заданными распределениями, что дает возможность варьировать исходными данными. Недостаток этого метода заключается в том, что отдельные ошибки могут быть не обнаружены, если они не выходят за рамки средних статистических оценок.

Тестирование программ по стратегии "белого ящика" ведется с использованием методик, которые покрывают логику (текст) программы, причем чем выше степень покрытия логики программы, тем более всесторонними считаются ТНД [6].

Наиболее формализованным методом проверки программного обеспечения является статическое тестирование. При использовании этого метода в качестве эталона берутся правила структурного построения программных модулей ([8], [9]), а проверка выполнения этих правил проводится при помощи анализа текста программ на языке программирования. Отсюда данный метод получил свое второе название – символическое тестирование. Метод символического тестирования может быть углублен вплоть до уровня формальной верификации и доказательства правильности программ. Такой подход к тестированию программных модулей обеспечивает всестороннюю проверку ПО, но чрезвычайно трудоемок и требует больших материальных затрат.

Результаты сравнительного анализа методов из стратегий "белого" и "черного" ящиков с целью определения предпочтительных методов при тестировании программных модулей контроля полета приведены в таблице 1.

Таблица 1

Методы тестирования	Модули комплекса воспроизведения	Модули комплекса допускового контроля	Модули комплекса контроля качества полета
Эквивалентное разбиение	+	+	+
Анализ граничных значений		+	+
Функциональные диаграммы			
Предположение об ошибке	+		+
Динамическое тестирование	+	+	+
Стохастическое тестирование		+	+
Покрытие операторов	+	+	+
Покрытие решений/условий		+	+
Комбинаторное покрытие условий	+	+	+
Статическое тестирование	+	+	+
Доказательство правильности программ	+	+	+

Поскольку модули воспроизведения в основном занимаются визуализацией ПИ, при их тестировании в меньшей степени применимы методы анализа граничных значений и стохастического тестирования, которые хорошо "работают" для модулей из комплексов допускового контроля и контроля качества полета, так как эти комплексы должны качественно определять события контроля. По причине того, что модули допускового контроля реализуют алгоритмы контроля пилотирования и работоспособности ВС, при их тестировании мало что может дать метод предположения об ошибке, который может быть удачно применен при тестировании модулей из комплекса воспроизведения.

3. Комплексная стратегия проектирования тестов для программных модулей обработки полетной информации

Поскольку применение методов из стратегий "белого" и "черного" ящиков обеспечивает создание набора тестов, покрывающих каждый свою область входных и выходных значений и логики модуля, предлагается определить комплексную стратегию, которая будет состоять из следующих правил:

- 1) Используя метод эквивалентного разбиения, выделить правильные и неправильные классы эквивалентности по входным и выходным данным. Метод эквивалентного разбиения учитывает

синтаксические и семантические правила построения программ, и способы задания данных, и поэтому он эффективен для тестирования любых модулей и программ, входящих в состав АСКП.

- 2) Обязательно использовать метод граничных значений, включая анализ граничных значений для входных и выходных переменных программы. Анализ граничных значений может дать наборы дополнительных тестовых условий, поэтому необходимо рассмотреть тесты, построенные на предшествующих шагах и дополнить их.
- 3) Логика программы проверяется согласно критериев покрытия решений/условий или комбинаторного покрытия условий, из которых последний является наиболее полным. Рекомендуется дополнять набор тестов вновь построенными тестами, удовлетворяющими одному из приведенных критериев покрытия.
- 4) Дополнительные тестовые наборы данных можно получить, используя метод предположения об ошибке.

Предложенная комплексная стратегия тестирования даст возможность провести более полное и эффективное тестирование программных продуктов контроля полетов и обеспечить выявление максимального количества ошибок в тестируемых модулях по сравнению с использованием каждого из методов в отдельности.

4. Тестирование функциональных групп модулей

Прежде чем тестировать комплекс программ КП целиком, необходимо протестировать составляющие его компоненты – модули, содержащие, как правило, небольшое количество операторов, а затем провести тестирование группы модулей, составляющих программу, подсистему или систему.

При анализе подходов к тестированию модулей возникает дилемма: тестировать ли каждый модуль в отдельности, а затем собирать программу и тестировать ее в целом, или модули для прохождения тестирования последовательно подключать к группе ранее протестированных модулей. Первый подход носит название монолитного метода тестирования, а второй – пошагового метода тестирования и сборки программ. Сравнительный анализ этих методов позволяет сделать следующие выводы:

- 1) При пошаговом тестировании можно эффективнее протестировать межмодульные интерфейсы и выявить ошибки в них на ранней стадии тестирования ввиду того, что сборка программы начинается с подключения к группе тестируемых модулей второго модуля. В противоположность пошаговому методу, при монолитном тестировании модули собираются в единую программу на последнем этапе тестирования, и поэтому отладка межмодульных интерфейсов перегружена, т.к. приходится тестировать сразу все возможные связи между модулями.
- 2) При применении монолитного метода тестирования неизбежны большие затраты труда, чем при применении пошагового метода.
- 3) Результаты пошагового тестирования более качественны, поскольку тестирование модулей происходит комплексно при пошаговом наращивании количества модулей. При монолитном тестировании такие возможности отсутствуют.
- 4) При применении пошагового метода раньше определяются ошибки в интерфейсах между модулями, поэтому отладка программ более эффективна.
- 5) Использование вычислительной техники при монолитном тестировании эффективнее в силу того, что тестируемые цепочки модулей длиннее при использовании пошагового метода и короче при монолитном методе.
- 6) Применение монолитного тестирования предоставляет большие возможности для распараллеливания работ на первом этапе.

Преимущества пошагового подхода демонстрируются в пунктах 1), 2), 3), 4). Поскольку последствия необнаруженных ошибок в АСКП могут быть очень серьезны, эти преимущества перекрывают недостатки, отраженные в пунктах 5) и 6).

Проанализированы практические приемы проведения тестовых испытаний функциональных групп модулей. Проведенный анализ позволяет сделать заключение: применение пошагового метода тестирования при тестировании ПО КП предпочтительнее метода монолитного тестирования, поскольку результаты пошагового тестирования более качественны.

Пошаговый метод тестирования модулей состоит из нисходящей и восходящей стратегии. Какая из этих двух стратегий может обеспечить более эффективное тестирование программ в программных комплексах контроля полетов?

Рассмотрение этих стратегий позволило определить следующие преимущества восходящей стратегии тестирования модулей:

- 1) возможность ранней локализации ошибок, если ошибки находятся в основном в модулях нижнего уровня;
- 2) проще процесс создания тестовых условий и оценка результатов тестирования.

Недостатки восходящей стратегии состоят в следующем:

- 1) программа как целое не существует, пока к группе тестируемых модулей не будет добавлен последний модуль;
- 2) необходима разработка программ-драйверов.

В свою очередь преимущества нисходящего тестирования модулей выражаются в следующем:

- 1) возможность обнаружения ошибок на ранней стадии тестирования, если ошибки находятся в основном в верхних модулях программы;
- 2) структура программы формируется на ранних этапах тестирования.

К недостаткам нисходящего тестирования можно отнести следующие:

- 1) необходимо разрабатывать модули-заглушки, что является непростой задачей;
- 2) сложно обеспечивать тестовыми данными тестируемые модули посредством подчитывания их через модули-заглушки;
- 3) трудно создавать тестовые условия и сложно оценивать результаты тестирования, поскольку работу модулей нижних уровней должны имитировать заглушки;
- 4) стимулируется незавершение тестирования ввиду сложности построения различных версий корректных заглушек, обеспечивающих поступление ТНД.

С учетом пунктов 2) и 3) из недостатков нисходящего тестирования и пункта 2) из преимуществ восходящего тестирования предпочтение отдается стратегии восходящего тестирования.

По результатам оценки методов тестирования функциональных групп модулей сделаны выводы о возможности применения этих методов при тестировании ПО КП, которые приведены в таблице 2.

Таблица 2

Методы тестирования	Программы комплекса воспроизведения	Программы комплекса допускового контроля	Программы комплекса контроля качества полета
Пошаговый метод	+	+	+
Метод монолитного тестирования			
Нисходящее тестирование			
Восходящее тестирование	+	+	+

Приведенные методы можно эффективно применять для тестирования следующих компонент:

- подпрограмм, входящих в состав программ обработки ПИ;
- программ, входящих в состав систем или комплексов КП.

Использованные сокращения

АСКП	автоматизированная система контроля полетов
ВС	воздушное судно
КП	контроль полетов
ПИ	полетная информация
ПО	программное обеспечение
ПО КП	программное обеспечение контроля полетов
ТНД	тестовые наборы данных

Л и т е р а т у р а

1. Порядок збирання та практичного використання інформації бортових систем реєстрації на підприємствах цивільної авіації України. АПУ-3. Експлуатація повітряних суден. Київ-96.
2. Единые требования МАК к системам обработки и анализа полетной информации. Москва. МАК. 1993.
3. Програма перевірки відповідності програмного забезпечення контролю польоту вимогам ДСТУ 3275-95 і “Порядку збирання та практичного використання інформації бортових систем реєстрації на підприємствах цивільної авіації України”. Київ. 1997.
4. Н.А.Яцков. Основы построения автоматизированных систем контроля полетов воздушных судов. Киев, КИИГА, 1989. – 344с.
5. В.В.Липаев. Проектирование программных средств. Москва, “Высшая школа”, 1990. – 303с.
6. Г.Майерс. Искусство тестирования программ. / Пер. с англ. Москва, “Мир”, 1982. – 176с.
7. Р.Гантер. Методы управления проектированием программного обеспечения. / Пер. с англ. Москва, “Мир”, 1981. –392с.
8. Э.Йодан. Структурное проектирование и конструирование программ. / Пер. с англ. Москва, “Мир”, 1979. – 416с.
9. Дж.Хьюз, Дж.Мичтом. Структурный подход к программированию. / Пер. с англ., Москва, “Мир”, 1980. – 278с.