УДК 656.7.052.002.5:681.3.06.019. 3 (045)

И.Э. Райчев, А.Г.Харченко, Н.А.Яцков

ИССЛЕДОВАНИЕ МЕТОДОВ ТЕСТИРОВАНИЯ ПРОГРАММНЫХ МОДУЛЕЙ ОБРАБОТКИ ПОЛЕТНОЙ ИНФОРМАЦИИ

Рассмотрены вопросы тестирования модулей, входящих в состав комплексов контроля полетов, обоснован выбор методов, которые могут быть применены для их тестирования. Даны рекомендации по использованию методов при тестировании программных модулей и предложена комплексная стратегия проектирования тестов для модулей обработки полетной информации.

Программное обеспечение контроля полетов используется для решения задач контроля при эксплуатации воздушных судов, а также при расследовании летных происшествий. Результаты решения этих задач непосредственно связаны с обеспечением безопасности полетов. Требования к программному обеспечению контроля полетов изложены в стандарте [1] и нормативном документе о порядке сбора и практического использования информации бортовых систем регистрации на предприятиях гражданской авиации Украины. Эти требования можно представить в виде следующих групп:

- перечень обязательных задач;
- состав и форма представления входной и выходной информации;
- точность представления информации и достоверность результатов контроля.

Основным средством проверки соответствия программного обеспечения контроля полетов приведенным требованиям является тестирование. Тестирование позволяет обнаружить ошибки в программах и программных комплексах, а в случае отсутствия ошибок тестируемое программное обеспечение признается работоспособным. Программное обеспечение контроля полетов считают надежным, если оно удовлетворяет требованиям стандарта [1] и единым требованиям МАК к системам обработки и анализа полетной информации

Программные комплексы контроля полетов достаточно сложны, поэтому целесообразно проводить их тестирование поэтапно: модуль → программа → подсистема → система → комплекс.

Программное обеспечение контроля полетов состоит из трех основных программных комплексов:

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

Комплекс программ воспроизведения полетной информации 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. Соответствующий комплекс программ включается в любую автоматизированную систему контроля полетов и состоит из ряда программ и подпрограмм.

Комплекс программ допускового контроля полета обеспечивает реализацию всех алгоритмов контроля, определенных в эксплуатационной документации на воздушное судно. Результаты решения задач контроля полетов используются для принятия заключения о нахождении объекта контроля "воздушное судно-экипаж-среда" в том или ином состоянии. Программы комплекса предназначены для выявления опасных отклонений в работе систем воздушного судна и ошибок пилотирования, которые могут влиять на безопасность полета [2].

Комплекс программ контроля качества выполнения 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:

- формированием информационного «портрета» этапов полета;

- определением значений показателей качества пилотирования.

Как правило, методы тестирования программ как отдельных модулей относят либо к стратегии "черного ящика", либо к стратегии "белого ящика"[3].

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 "белыйм

Обязательное использование метода граничных значений включает в себя 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.

Логика программы проверяется согласно критериям покрытия решений/условий или комбинаторного покрытия условий, из которых последний является наиболее полным. Набор тестов рекомендуется дополнять вновь построенными тестами, удовлетворяющими одному из приведенных критериев покрытия.

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.

Прежде чем тестировать комплекс программ контроля полетов 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.

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 инетерфейсов перегружена, так как приходится тестировать сразу все возможные связи между модулями.

При применении монолитного метода тестирования неизбежны большие затраты труда, чем при применении пошагового метода.

Результаты пошагового тестирования более качественны, поскольку тестирование модулей происходит комплексно при пошаговом наращивании количества модулей. При монолитном тестировании такие возможности отсутствуют.

При применении пошагового метода раньше определяются ошибки в интерфейсах между модулями, поэтому отладка программ более эффективна.

Применение вычислительной техники при монолитном тестировании эффективнее в силу того, что тестируемые цепочки модулей длиннее при использовании пошагового метода и короче при монолитном методе.

Применение монолитного тестирования предоставляет большие возможности для распараллеливания работ на первом этапе.

Таким образом, преимущество пошагового метода тестирования перед монолитным методом очевидно, поскольку последствия необнаруженных ошибок в автоматизированной системе контроля полетов могут быть очень серьезны.

Проведенный анализ практических приемов тестовых испытаний функциональных групп модулей позволяет сделать заключение: применение пошагового метода тестирования при тестировании программного обеспечения контроля полетов предпочтительнее метода

монолитного тестирования, поскольку результаты пошагового тестирования более качественны.

Пошаговый метод тестирования модулей состоит из нисходящей и восходящей стратегии. Анализ этих стратегий позволил определить следующие достоинства восходящей стратегии тестирования модулей:

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

Недостатки восходящей стратегии состоят в следующем:

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

В свою очередь, к достоинствам нисходящего тестирования модулей относятся:

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

К недостаткам нисходящего тестирования можно отнести следующие:

- необходимость разработки модулей-заглушек, что является непростой задачей;
- сложное обеспечение тестовыми данными тестируемых модулей посредством подчитывания их через модули-заглушки;
- трудность создания тестовых условий и сложность оценки результатов тестирования, поскольку работу модулей нижних уровней должны имитировать заглушки;
- стимулирование незавершения тестирования ввиду сложности построения различных версий корректных заглушек, обеспечивающих поступление тестовых наборов данных.

Предпочтение отдается стратегии восходящего тестирования.

По результатам оценки методов тестирования функциональных групп модулей сделаны выводы о возможности применения этих методов при тестировании программного обеспечения контроля полетов, которые приведены в табл. 2.

Таблица 2

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

Таким образом, приведенные методы можно эффективно применять для тестирования подпрограмм, входящих в состав программ обработки полетной информации, и программ, входящих в состав систем или комплексов контроля полетов.

Список литературы

1. *ДСТУ* 3275-95. Системи автоматизованого оброблення польотної інформації. Наземні, загальні вимоги. - К.: Держстандатр України, 1996. - 13 с.

2. *Яцков Н.А.* Основы построения автоматизированных систем контроля полетов воздушных судов. - К.: КИИГА, 1989. – 344 с.

3 *Майерс. Г.* Искусство тестирования программ / Пер. с англ. - М.: Мир, 1982. – 176 с.

4. *Гантер Р.* Методы управления проектированием программного обеспечения / Пер. с англ. - М.: Мир, 1981. –392 с.

5. *Липаев В.В.* Проектирование программных средств. - М.: Высш. шк. - 1990. – 303 с.

6. *Йодан Э.* Структурное проектирование и конструирование программ / Пер. с англ. - М.: Мир, 1979. – 416 с.

7. *Хьюз Дж., Мичтом Дж.* Структурный подход к программированию / Пер. с англ. - М.: Мир, 1980. – 278 с.

Стаття надійшла до редакції 2 грудня 1999 року.