

УДК 004

ТЕСТИРОВАНИЕ КАК КЛЮЧЕВОЙ ЭЛЕМЕНТ СИСТЕМЫ ОБЕСПЕЧЕНИЯ КАЧЕСТВА ИНФОРМАЦИОННЫХ СИСТЕМ

Воробьев Александр Федорович

магистрант

Дальневосточный федеральный университет, Владивосток

author@apriori-journal.ru

Аннотация. В статье раскрывается понятие тестирования информационной системы как надежного способа оценки ее качества. Рассматриваются наиболее распространенные виды и критерии тестирования, которые применяют разработчики и эксплуатирующие службы. По мнению автора, тестирование нельзя позиционировать в качестве единственного метода обеспечения качества информационной системы, однако тестирование является ключевым элементом системы, предназначенной помочь найти и исправить ошибки на самой ранней стадии проектирования.

Ключевые слова: тестирование; виды тестирования; критерии тестирования; информационная система; обеспечение качества информационных систем.

TESTING AS KEY ELEMENT OF SYSTEM OF ENSURING QUALITY OF INFORMATION SYSTEMS

Vorob'ev Aleksandr Fedorovich

undergraduate
Far eastern federal university, Vladivostok

Abstract. In article the concept of testing of information system as reliable way of an assessment of her quality reveals. The most widespread types and criteria of testing which are applied by developers and the operating services are considered. According to the author, testing can't be positioned as the only method of ensuring quality of information system, however testing is a key element of the system intended to help to find and correct errors at the earliest design stage.

Key words: testing; types of testing; criteria of testing; information system; ensuring quality of information systems.

При использовании информационных систем возникает острая необходимость оценивать качество их функционирования. Категория качества работы информационной системы характеризуется набором свойств, отражающих, насколько она удовлетворяет требованиям всех заинтересованных сторон, таких как заказчик продукта, спонсор, конечный пользователь, разработчики и прочие.

Можно выделить один из наиболее часто используемых методов оценки качества информационной системы, который входит в набор эффективных средств современной системы обеспечения качества программного продукта – это его тестирование. Тестирование нельзя пози-

ционировать в качестве единственного метода обеспечения качества. Оно является только частью общей системы обеспечения качества информационной системы, элементы для которой выбираются по критерию максимальной эффективности применения в конкретном проекте.

Тестирование – одна из самых популярных методик повышения качества, подкрепленная богатым опытом и исследованиями разработки информационных систем. Выделяют несколько видов тестирования (таблица 1).

Таблица 1

Виды тестирования информационных систем

№ п/п	Вид тестирования	Особенность применения данного вида тестирования на практике
1.	Блочное тестирование	тестирование полного класса, метода или небольшого приложения, выполняемое отдельно от прочих частей системы
2.	Тестирование компонента	тестирование элемента системы, выполняемое в изоляции от остальных частей системы
3.	Интеграционное тестирование	тестирование совместного выполнения нескольких подсистем
4.	Регрессивное тестирование	повторное выполнение тестов, направленное на обнаружение дефектов в уже прошедшей этот набор тестов программе
5.	Тестирование системы	проверка окончательной конфигурации, интегрированной с другими программными и аппаратными системами

Существует четыре класса критериев для тестирования информационных систем (рисунок 1).

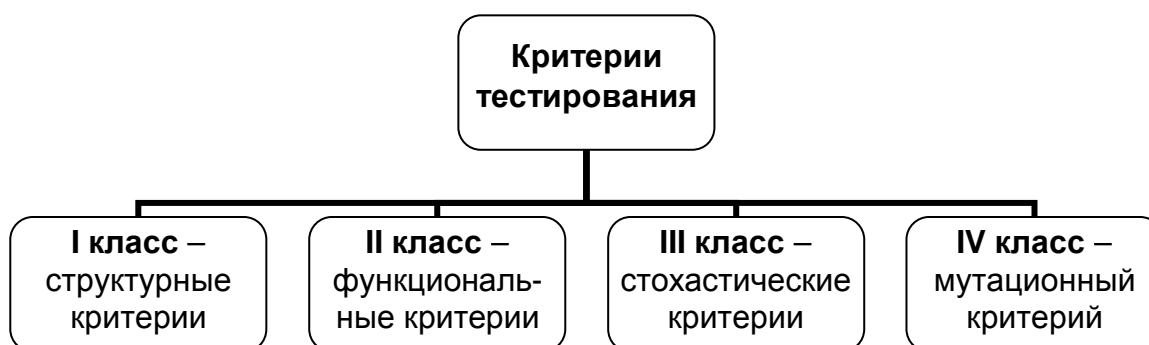


Рис. 1. Классы критериев тестирования информационных систем

Использование тех или иных критериев обосновывается конкретным видом тестирования и поставленной задачей.

В структурных критериях используют модель программы в виде «белого ящика», что предполагает наличие исходного текста программы или спецификации программы представленных в виде потокового графа управления. Вся структурная информация доступна и понятна разработчикам приложения, поэтому I класс критериев обычно используется на этапах модульного и интеграционного тестирования.

К I классу структурных критериев относят:

- Условие критерия тестирования команд (C0 критерий) – совокупность тестов должна обеспечить прохождение каждой отдельной команды не менее одного раза. Этот критерий, как правило, используется в больших программных системах, где другие критерии невозможно применить.
- Условие критерия тестирования ветвей (C1 критерий) – совокупность тестов должна обеспечить прохождение каждой отдельной ветви не менее одного раза. Это достаточно мощный и при этом экономичный критерий, поскольку множество ветвей в тестируемом

приложении не так уж велико. Данный критерий довольно часто используется при автоматизации тестирования.

- Условие критерия тестирования путей (C2 критерий) – совокупность тестов должна обеспечить прохождение каждого пути не менее одного раза. Если программа содержит в себе цикл, то число итераций ограничивается константой.

Функциональные критерии (II класс) являются важнейшими критериями тестирования для индустрии программного обеспечения. Прежде всего, они обеспечивают контроль степени выполнения выставляемых требований заказчика в программном продукте. Поскольку требования к продукту формулируются в целом, они отображают взаимодействие тестируемого продукта с окружением.

При функциональном тестировании используется преимущественно модель «черного ящика». Проблема такого вида тестирования – это трудоемкость: так как, документы, фиксирующие требования к программному обеспечению, достаточно объемны, то соответствующая проверка должна быть всеобъемлющей.

К самым распространённым видам функциональных критериев относят:

- Тестирование пунктов спецификации – совокупность тестов должна обеспечить проверку каждого тестируемого пункта и обязательно не менее одного раза.
- Тестирование классов входных данных – совокупность тестов должна обеспечить проверку представителя каждого класса входных данных и обязательно не менее одного раза.
- Тестирование правил – набор тестов в совокупности должен обеспечить проверку каждого правила, если входные и выходные значения описываются набором правил некоторой грамматики.
- Тестирование классов выходных данных – совокупность тестов должна обеспечить проверку представителя каждого выходного

класса, с обязательным условием, что выходные результаты заранее расклассифицированы, причем, что отдельные классы результатов учитывают ограничения на ресурсы и на время.

- Тестирование функций – совокупность тестов должна обеспечить проверку каждого действия, реализуемого тестируемым модулем, и обязательно не менее одного раза.
- Комбинированные критерии для программ и спецификаций – совокупность тестов должна обеспечить проверку всех комбинаций непротиворечивых условий и обязательно не менее одного раза.

При обследовании сложных программных комплексов, когда набор детерминированных тестов имеет громадную мощность, используют стохастическое тестирование, к основным критериям (III класс) которого относят:

- Статистические методы окончания тестирования – стохастические методы принятия решений о совпадении гипотез о распределении случайных величин. К ним относятся такие хорошо известные методы: Стьюдента, Хи-квадрат и другие.
- Метод оценки скорости выявления ошибок – основан на модели скорости выявления ошибок, в соответствии с которой тестирование прекращается, если интервал времени между текущей ошибкой и следующей слишком велик для фазы тестирования приложения.

Мутационный критерий (IV класс) используется при оценке программ, созданных высококлассными программистами. Считается, что профессиональные программисты создают сразу почти правильные программы, отличающиеся от готовых программ мелкими ошибками или описками типа: перестановка местами максимальных значений индексов, ошибки в знаках операций, занижение или завышение границы цикла и прочими, так называемыми, мутациями. Набор тестов, используемых для оценки качества программного продукта, считается соответствующим му-

тационному критерию, когда выявляются все ошибки программ-мутантов и основной программы, признающей в итоге правильной.

Процедура тестирования состоит из ряда этапов, каждый из которых по своему важен и необходим для достижения конечной цели – обеспечения надлежащего качества информационной системы.

1. Необходимо заранее разработать планы для тестирования. Начинать выбирать способ тестирования непосредственно перед запуском теста – это ошибка многих проектов. План тестирования является неотъемлемой частью проекта. Опережающее планирование позволяет организации начать тестирование вовремя и не выходить за рамки плана и бюджета.
2. Определение стадий разработки. Тщательно выбранный процесс разработки дает возможность описать все детали процесса тестирования. Ясное понимание этапов работы на каждой стадии составляет основу организации разработки и процесса тестирования.
3. Определение критериев входа и выхода. Критерии входа и выхода выявляются для тех моментов, когда стадия начинается и заканчивается. Разработчики должны определить эти стадии на предпроектном этапе.
4. Определить условия теста необходимо как можно раньше. Необходимо еще во время разработки спецификации идентифицировать условия теста, а не непосредственно перед запуском теста. Описание условий тестирования на протяжении всего процесса разработки обеспечивает завершенность и возможность проверки спецификаций.
5. Управление метриками тестирования. Разнообразие метрик тестирования позволяет руководителям проекта достоверную информацию о ходе работ. Метрики тестирования отслеживают стадию ошибок, на которой они обнаружены, количество, и меры для их устранения.
6. Наличие менеджера по тестам и независимая тестовая команда. Менеджер, ответственный за тестирование и независимые специа-

листы отделяют процесс обнаружения ошибок от процесса разработки приложения.

7. Вовлечение заказчика в процесс разработки. Вовлечение заказчиков в определение условий тестирования позволяет разработчикам рассматривать выдвигаемые заказчиком критерии как условия тестов, а не как субъективные пожелания.
8. Организация рабочих бригад. Рабочая бригада является формой организации коллектива, апробированной в производственной практике многих предприятий и отраслей. Бригады устанавливают изнутри проекта точные границы ответственности каждого сотрудника.
9. Определение архитектуры тестирования. Наличие архитектуры тестирования упрощает выполнение тестов и позволяет больше времени уделить результатам тестов, а также внесению необходимых изменений.
10. Эффективно использовать инструменты тестирования. Тестирование весьма трудоемкий процесс. Инструменты тестирования автоматизируют некоторые стороны этого процесса и помогают сэкономить время и затраты.

Тестирование как ключевой элемент системы обеспечения качества информационных систем несет в себе важнейшую функцию. Оно предназначено помочь найти и исправить ошибки на самой ранней стадии проектирования. Это позволяет позиционировать тестирование как постоянной и построенной на единых критериях деятельности на протяжении всего этапа разработки информационной системы.

Список использованных источников

1. Винниченко И.В. Автоматизация процессов тестирования. СПб.: Питер, 2005. 203 с.
2. Петров В.Н. Информационные системы. СПб.: Питер, 2003. 688 с.
3. Котляров В.П., Коликова Т.В. Основы тестирования программного обеспечения. М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2006. 285 с.
4. Грекул В.И., Денищенко Г.Н., Коровкина Н.Л. Проектирование информационных систем. М.: ИНТУИТ, 2005. 240 с.
5. Смирнова Г.Н., Сорокин А.А., Тельнов Ю.Ф. Проектирование экономических информационных систем. М.: Финансы и статистика, 2002. 512 с.
6. Орлов С. Технологии разработки программного обеспечения. СПб.: Питер, 2002. 464 с.