УДК 621.389

А.А. Пивень, Ю.И. Скорин

Харьковский национальный экономический университет, Харьков.

ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Проводится анализ, и рассматриваются виды тестирования программного обеспечения, назначение тестирования программного обеспечения, краткое описание развития тестирования программного обеспечения с 1990-х годов, а также рассмотрена классификация тестирования программного обеспечения по объекту тестирования. Проанализированы следующие виды тестирования: функциональное, регрессионное, стресс-тестирование, нагрузочное тестирование, тестирование интерфейса пользователя, тестирование безопастности, тестирование удобства использования, модульное тестирование и системное тестирование программного обеспечения.

Ключевые слова: *программное обеспечение, надёжность, практичность, эффективность, мобильность, функциональность, верификация, качество, производительность, доступность программного обеспечения, интеграционное и системное тестирование, тестирование стабильности и надежности.*

Вступление

Существующие на сегодняшний день методы тестирования программного обеспечения (ПО) не позволяют однозначно и полностью выявить все дефекты и установить корректность функционирования анализируемой программы, поэтому все существующие методы тестирования действуют в рамках формального процесса проверки исследуемого или разрабатываемого ПО.

Наиболее распространёнными проблемами, возникающими в процессе разработки программного обеспечения, считают недостаточную надежность. Самый сложный процесс — поиск и исправление ошибок в программных продуктах.

Существует множество подходов к решению задачи тестирования и верификации программного обеспечения, но эффективное тестирование сложных программных продуктов — это процесс в творческий, не сводящийся к следованию строгим и чётким процедурам или созданию таковых.

С точки зрения ISO 9126, качество программных средств можно определить как совокупную характеристику исследуемого программного обеспечения с учётом следующих составляющих: надёжность, сопровождаемость, практичность, эффективность, мобильность, функциональность.

В начале 1990-х в понятие «тестирование» стали включать планирование, проектирование, создание, поддержку и выполнение тестов и тестовых окружений, и это означало переход от тестирования к обеспечению качества, охватывающего весь цикл разработки программного обеспечения.

В это время начинают появляться различные программные инструменты для поддержки процесса тестирования:

более продвинутые среды для автоматизации с возможностью создания скриптов и генерации отчетов, системы управления тестами, программное обеспечение для проведения нагрузочного тестирования.

В середине 1990-х с развитием Интернета и разработкой большого количества веб-приложений особую популярность стало получать «гибкое тестирование» (по аналогии с гибкими методологиями программирования).

В 2000-х появилось еще более широкое определение тестирования, когда в него было добавлено понятие «оптимизация бизнес-технологий».

Оптимизация бизнес технологий направляет развитие информационных технологий в соответствии с целями бизнеса.

Основной подход заключается в оценке и максимизации значимости всех этапов жизненного цикла разработки программного обеспечения для достижения необходимого уровня качества, производительности, доступности [1].

Изложение основного материала

На данный момент существуют различные виды тестирования, а именно:

Регрессионное тестирование, которое проводится с целью проверить, не влияют ли новые функции, улучшения и исправленные дефекты на существующую функциональность продукта.

Функциональное тестирование, т.е. тестирование функций приложения на соответствие требованиям. Оценка производится в соответствии с ожидаемыми и полученными результатами, при условии, что функции отрабатывали на различных значениях [2].

Тестирование графического интерфейса пользователя предполагает проверку соответствия приложения требованиям к графическому интерфейсу.

Тестирование безопасности – оценка уязвимости программного обеспечения к различным атакам.

Компьютерные системы очень часто являются мишенью незаконного проникновения.

Под проникновением понимается широкий диапазон действий: попытки хакеров проникнуть в систему из спортивного интереса, месть рассерженных служащих, взлом мошенниками для незаконной наживы.

Тестирование безопасности проверяет фактическую реакцию защитных механизмов, встроенных в систему, на проникновение. В ходе тестирования безопасности испытатель играет роль взломщика.

Ему разрешено все:

попытки узнать пароль с помощью внешних средств;

атака системы с помощью специальных утилит, анализирующих защиты;

подавление, ошеломление системы (в надежде, что она откажется обслуживать других клиентов);

целенаправленное введение ошибок в надежде проникнуть в систему в ходе восстановления;

просмотр несекретных данных в надежде найти ключ для входа в систему.

Тестирование удобства пользования приложением определяет, соответствует ли приложение потребностям целевой аудитории и отвечает ли оно требованиям пользователя.

При тестировании во внимание принимаются следующие аспекты: однородность, логика и структура, навигация.

Если создаваемая нагрузка на систему превышает нормальные сценарии её использования, с целью протестировать время отклика системы на высоких или пиковых нагрузках, такое тестирование называется стресс-тестированием.

В этом случае нагрузка обычно столь высока, что предсказуема ошибочная работа системы, однако не существует четкой границы между тем, когда тестирование является нагрузочным и тем, когда оно становится стресс-тестированием [3].

Стресс-тестирование – один из видов тестирования программного обеспечения, которое оценивает надежность и устойчивость системы в условиях превышения пределов нормального функционирования.

Стресс-тестирование особенно необходи-мо для "критически важного" программного обеспечения, однако также используется и для остального ПО.

Обычно стресс-тестирование лучше обнаруживает устойчивость, доступность и обработку исключений системой под большой нагрузкой, чем то, что считается корректным поведением в нормальных условиях.

В общем случае методология стресс-тестирования основана на снятии и анализе показателей производительности приложения при нагрузках, значительно превышающих ожидаемые на стадии сопровождения и несёт в себе цель определить выносливость или устойчивость приложения на случай всплеска активности по его использованию.

Необходимость стресс-тестирования диктуется следующими факторами: большая часть всех систем разрабатываются с допущением о функционировании в нормальном режиме и даже в случае, когда допускается возможность увеличения нагрузки, реальные объёмы её увеличения не принимаются во внимание.

В случае SLA-контракта (соглашения об уровне услуг) стоимость отказа системы в экстремальных условиях может быть очень велика.

Обнаружение некоторых ошибок или дефектов в функционировании системы не всегда возможно с использованием других типов тестирования.

Тестирования, проведенного разработ-чиком, может быть недостаточно для эмуляции условий, при которых происходит отказ системы. Предпочтительнее быть готовым к обработке экстремальных условий системы, чем ожидать её отказа [4]. Тестирование стабильности или надежности – один из видов тестирования программного обеспечения, целью которого является проверка работоспособности приложения при длительном тестировании со средним (ожидаемым) уровнем нагрузки.

Перед тем как подвергать программное обеспечение экстремальным нагрузкам стоит провести проверку стабильности в предполагаемых условиях работы, то есть погрузить продукт в полную рабочую атмосферу.

При тестировании, длительность его проведения не имеет первостепенного значения, основная задача - наблюдая за потреблением ресурсов, выявить утечки памяти и проследить чтобы скорость обработки данных и/или время отклика приложения в начале теста и с течением времени не уменьшалась.

В противном случае вероятны сбои в работе продукта и перезагрузки системы.

Часто в "домашних" условиях тестирование стабильности совмещают со стресс-тестированием, то есть проверяют не только стабильность, но и способность приложения переносить жесткие условия и сильные нагрузки длительное время.

Тестирование совместимости — метод, основной целью которого является обеспечение качественной работы конечного продукта с другими программами, операционными системами,. аппаратными средствами и т.д.

Модульное тестирование, или юнит-тестирование — процесс, позволяющий проверить на корректность отдельные модули исходного кода программы.

Идея состоит в том, чтобы писать тесты для каждой нетривиальной функции или метода. Это

позволяет достаточно быстро проверить, не привело ли очередное изменение кода к регрессии, то есть к появлению ошибок в уже оттестированных местах программы, а также облегчает обнаружение и устранение таких ошибок.

Интеграционное тестирование — одна из фаз тестирования программного обеспечения, при котором отдельные программные модули объединяются и тестируются в группе.

Обычно интеграционное тестирование проводится после модульного тестирования и предшествует системному тестированию.

Интеграционное тестирование в качестве входных данных использует модули, над которыми было проведено модульное тестирование, группирует их в более крупные множества, выполняет тесты, определённые в плане тестирования для этих множеств, и представляет их в качестве выходных данных и входных для последующего системного тестирования.

Целью интеграционного тестирования является проверка соответствия проектируемых единиц функциональным, приёмным и требованиям надежности.

Системное тестирование программного обеспечения — это тестирование программного обеспечения, выполняемое на полной, интегрированной системе, с целью проверки соответствия системы исходным требованиям.

Системное тестирование относится к методам тестирования чёрного ящика, и, тем самым, не требует знаний о внутреннем устройстве системы.

Основной задачей системного тестирования является проверка как функциональных, так и не функциональных требований к системе в целом. При этом выявляются дефекты, такие как неверное использование ресурсов системы, непредусмотренные комбинации данных пользовательского уровня, несовместимость с окружением, непредусмотренные сценарии использования, отсутствующая или неверная функциональность, неудобство использования и т.д.

Для минимизации рисков, связанных с особенностями поведения системы в той или иной среде, во время тестирования рекомендуется использовать окружение максимально приближенное к тому, на которое будет установлен продукт после выдачи.

Можно выделить два подхода к системному тестированию: на базе требований, когда для каждого требования пишутся тестовые случаи (test cases), проверяющие выполнение данного требования; на базе случаев использования (use case based) [5].

Выводы

С ростом важности информационных технологий для жизни общества возрастает цена ошибок в программах.

В этих условиях на первый план выходят технологии и методы тестирования программного обеспечения, которые позволяют своевременно выявить и исправить эти ошибки. Качественное ПО - это репутация фирмы, поэтому необходимо серьёзно относиться к вопросам тестирования, это поможет не только отстоять свои позиции на рынке но и завоевать новые.

Список литературы

1. http://en.wikipedia.org/wiki/Software_testing.

2 Синицын С. В., Налютин Н. Ю. Верификация программного обеспечения — М.: БИНОМ, 2008. — 368 с.

3. Калбертсон Роберт, Браун Крис, Кобб Гэри Быстрое тестирование. — М.: «Вильямс», 2002. — 374 с.

4. Биндер Роберт В.. Testing Object-Oriented Systems: Objects, Patterns, and Tools. Addison-Wesley Professional. стр. 45

5. Лайза Криспин, Джанет Грегори Гибкое тестирование: практическое руководство для тестировщиков ПО и гибких команд = Agile Testing: A Practical Guide for Testers and Agile Teams — М.: «Вильямс», 2010. — 464 с.

Надійшла до редколегії 23.03.2012

Рецензент: д-р техн. наук, доц. К.О. Метєшкін, Харківська національна академія міського господарства, Харків.

ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

А.О. Півень, Ю.І. Скорін

Проводиться аналіз та розглядаються види тестування програмного забезпечення, призначення тестування програмного забезпечення, стислий опис розвитку тестування програмного забезпечення з 1990х років, а також розгляне на класифікація тестування програмного забезпечення по об'єкту тестування . Проаналізовані наступні види тестування: функціональне, регресивне тестування, стрес-тестування, навантажувальне тестування, тестування інтерфейсу користувача, тестування безпеки, тестування зручності користування, модульне тестування та системне тестування програмного забезпечення.

Ключові слова: *програмне забезпечення, надійність, практичність, ефективність, мобільність, функціональність, верифікація, якість, продуктивність, інтеграційне і системне тестування, тестування стабільності і надійності.*

SOFTWARE TESTING

A.A. Pivem, Yu.I. Skorin

Analyzes, and discusses types of software testing, the purpose of software testing, a brief description of of software testing with the 1990s, as well as the classification of software testing facility testing. Analyzed the following types of testing: functional, regression, stress testing, stress testing, user interface, testing, safety, usability testing, unit testing and system testing software.

Keywords: *software, reliability, practicality, efficiency, mobility, functionality, quality, productivity, availability of software, integration and system testing, testing of stability and reliability.*