

Тестирование программного обеспечения

Выполнила:

преподаватель спецдисциплин

Н.В. Полякова

• **Тестирование ПО**—процесс исследования ПО с целью получения информации о качестве продукта (ISO 9126) с учетом следующих составляющих:

–Функциональность

–Надёжность

–Удобство использования (практичность)

–Производительность (эффективность)

–Удобство сопровождения

–Переносимость (мобильность)

• Существует множество подходов к решению задачи тестирования ПО.

- **Основная цель тестирования** – доказательство того, что результат разработки соответствует предъявленным к нему требованиям.

- **Процесс верификации** проводится сверху вниз:

от общих требований, заданных в техническом задании и/или спецификации на всю информационную систему до детальных требований на программные модули и их взаимодействие.

Различие тестирования и отладки

- Тестирование – поиск ошибок в ПО
- Отладка – поиск местоположения ошибок в программном коде и их исправление.

Терминология

• **Валидация**—доказательство того, что в результате разработки ПО достигнуты те цели, которые планировали достичь благодаря ее использованию.

• **Верификация**—достижение гарантии того, что верифицируемый объект (требования или программный код)

—соответствует требованиям,

—реализован без непредусмотренных функций и

—удовлетворяет проектным спецификациям и стандартам.

• **Тестирование**—это обнаружение (доказательство наличия) ошибок (несоответствий, неполноты, двусмысленностей и т.д.) в текущем состоянии •

Валидация программной системы - проверка соответствия системы ожиданиям заказчика

- Цель - доказательство того, что в результате разработки ПО достигнуты те цели, которые планировали достичь благодаря ее использованию.

Верификация программного обеспечения

- **Процесс верификации** включает в себя
 - инспекции,
 - тестирование кода,
 - анализ результатов тестирования,
 - формирование и анализ отчетов о проблемах.
- **Цель** - достижение гарантии того, что *верифицируемый* объект (требования или программный код)
 - соответствует требованиям,
 - реализован без непредусмотренных функций и
 - удовлетворяет проектным спецификациям и стандартам.

Участники тестирования

- **Группа обеспечения качества ПО (Quality Assurance Team, QA team).**
- **Специалист по контролю качества** - член проектной группы
 - осуществляет взаимодействие с разработчиками, менеджером программы и специалистами по безопасности и сертификации
 - отслеживает общее качество ПО,
 - соответствие ПО стандартам и спецификациям.
- **Специалисты по тестированию** – участник команды разработчиков;
 - определяет стратегию тестирования, тест-требования и тест-планы для каждого этапа проекта;
 - выполняет тестирование системы, собирает и анализирует отчеты о прохождении тестирования.
- **Разработчики (программисты)** – выполняют юнит - тестирование.

Классификация видов тестирования

—по объекту тестирования

—по знанию системы

—по степени автоматизации

—по степени изолированности

КОМПОНЕНТОВ

—по времени проведения тестирования

Классификация по объекту тестирования

- Функциональное тестирование (functional testing)
- Тестирование производительности (performance/stress testing)
 - Нагрузочное тестирование (load testing)
 - Стресс тестирование (stress testing)
 - Тестирование стабильности (stability/load testing)
- Тестирование удобства использования (usability testing)
- Тестирование интерфейса пользователя (UI testing)
- Тестирование безопасности (security testing)
- Тестирование локализации (localization testing)
- Тестирование совместимости (compatibility testing)

Функциональное тестирование — это тестирование ПО в целях проверки реализуемости функциональных требований, то есть способности ПО в определённых условиях решать задачи, нужные пользователям. Функциональные требования определяют, что именно делает ПО, какие задачи оно решает.

Функциональные требования включают в себя:

- Функциональная пригодность (*suitability*).
- Точность (*accuracy*).
- Способность к взаимодействию (*interoperability*).
- Соответствие стандартам и правилам (*compliance*).
- Защищённость (*security*).

Направления тестирования производительности

В тестировании производительности различают следующие направления:

- нагрузочное (load)
- стресс (stress)
- тестирование стабильности (endurance or soak or stability)
- конфигурационное (configuration)

Возможны два подхода к тестированию производительности программного обеспечения:

1) в терминах рабочей нагрузки: ПО подвергается тестированию в ситуациях, соответствующих

различным сценариям использования;

2) в рамках бета-тестирования, когда система

испытывается реальными конечными пользователями.

Нагрузочное тестирование — это простейшая форма тестирования производительности. Обычно проводится для того, чтобы оценить поведение приложения под заданной ожидаемой нагрузкой. Этой нагрузкой может быть, например, ожидаемое количество одновременно работающих пользователей приложения, совершающих заданное число транзакций за интервал времени.

Такой тип тестирования обычно позволяет получить время отклика всех самых важных бизнес-транзакций. В случае наблюдения за базой данных, сервером приложений, сетью и т. д., этот тип тестирования может также идентифицировать некоторые узкие места приложения.

Стресс-тестирование обычно используется для понимания пределов пропускной способности приложения.

Этот тип тестирования проводится для определения надёжности системы во время экстремальных или диспропорциональных нагрузок и отвечает на вопросы о достаточной производительности системы в случае, если текущая нагрузка сильно превысит ожидаемый максимум



Тестирование стабильности проводится с целью убедиться в том, что приложение выдерживает ожидаемую нагрузку в течение длительного времени.

При проведении этого вида тестирования осуществляется наблюдение за потреблением приложением памяти, чтобы выявить потенциальные утечки.

Также тестирование выявляет деградацию производительности, выражающуюся в снижении скорости обработки информации и/или увеличении времени ответа приложения после продолжительной работы по сравнению с началом теста.



•
Конфигурационное тестирование — ещё один из видов традиционного тестирования производительности.

В этом случае вместо того, чтобы тестировать производительность системы с точки зрения подаваемой нагрузки, тестируется эффект влияния на производительность изменений в конфигурации. Пример конфигурационного - эксперименты с различными методами балансировки нагрузки.

Конфигурационное тестирование также может быть совмещено с нагрузочным, стресс или тестированием стабильности.

•

Тестирование удобства использования (*проверка эргономичности*) — исследование, выполняемое с целью определения, удобен ли некоторый искусственный объект (такой как веб-страница, пользовательский интерфейс или устройство) для его предполагаемого применения.

Проверка эргономичности измеряет эргономичность объекта или системы и сосредоточена на определённом объекте или небольшом наборе объектов, в то время как исследования взаимодействия человек-компьютер в целом — формулируют универсальные принципы.

Проверка эргономичности — метод оценки удобства продукта в использовании, основанный на привлечении пользователей в качестве тестировщиков, испытателей и суммировании полученных от них выводов.

Тестирование безопасности — оценка уязвимости программного обеспечения к различным атакам.

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

При неограниченном времени и ресурсах хорошее тестирование безопасности взломает любую систему. Задача проектировщика системы — сделать цену проникновения более высокой, чем цена получаемой в результате информации.

Классификация видов тестирования по знанию системы

- Тестирование чёрного ящика (blackbox)
- Тестирование белого ящика (whitebox)
- Тестирование серого ящика (graybox)

Тестирование чёрного ящика или **поведенческое тестирование** — стратегия (метод) тестирования функционального поведения объекта (программы, системы) с точки зрения внешнего мира, при котором не используется знание о внутреннем устройстве тестируемого объекта.

Под стратегией понимаются систематические методы отбора и создания тестов для тестового набора.

Стратегия поведенческого теста исходит из технических требований и их спецификаций.

Тестирование по стратегии белого ящика — тестирование кода на предмет логики работы программы и корректности её работы с точки зрения компилятора того языка, на котором она писалась.

Тестирование по стратегии белого ящика, также называемое **техникой тестирования, управляемой логикой программы**, позволяет проверить внутреннюю структуру программы. Исходя из этой стратегии, тестировщик получает тестовые данные путём анализа логики работы программы.

Техника Белого ящика включает в себя следующие методы тестирования:

покрытие решений

покрытие условий

покрытие решений и условий

комбинаторное покрытие условий

Классификация видов тестирования по степени автоматизированности

- Ручное тестирование (manualtesting)
- Автоматизированное тестирование (automatedtesting)
- Полуавтоматизированное тестирование (semiautomatedtesting)

Ручное тестирование — часть процесса тестирования на этапе контроля качества в процессе разработки программного обеспечения.

Оно производится тестировщиком без использования программных средств, для проверки программы или сайта путём моделирования действий пользователя. В роли тестировщиков могут выступать и обычные пользователи, сообщая разработчикам о найденных ошибках.

Автоматизированное тестирование программного обеспечения — часть процесса тестирования на этапе контроля качества в процессе разработки программного обеспечения.

Оно использует программные средства для выполнения тестов и проверки результатов выполнения, что помогает сократить время тестирования и упростить его процесс.

Классификация по степени изолированности компонентов

- **Модульное тестирование (юнит-тестирование)**— тестируется минимально возможный для тестирования компонент, например, отдельный класс или функция.
– Часто модульное тестирование осуществляется разработчиками ПО.
- **Интеграционное тестирование**— тестируются интерфейсы между компонентами, подсистемами.
– При наличии резерва времени на данной стадии тестирование ведётся итерационно, с постепенным подключением последующих подсистем.
- **Системное тестирование**— тестируется интегрированная система на её соответствие требованиям.
- **Приемочное тестирование**

Модульное тестирование (юнит-тестирование)

- Позволяет проверить на корректность отдельные модули исходного кода программы.
- Цель юнит-тестирования — изолировать отдельные части программы и показать, что по отдельности эти части работоспособны

Этот тип тестирования обычно выполняется программистами.

Интеграционное тестирование

- Тестируются интерфейсы между компонентами, подсистемами или системами. При наличии резерва времени на данной стадии тестирование ведётся итерационно, с постепенным подключением последующих подсистем.

Цель интеграционного тестирования - проверка соответствия проектируемых единиц функциональным, приемным и требованиям надежности.

Системное тестирование(приемочное тестирование)

- Тестирование программного обеспечения (ПО), выполняемое на полной, интегрированной системе, с целью проверки соответствия системы исходным требованиям.

- Основа для тестов - общие требования к программе:

- корректность реализации функций,

- производительность,

- время отклика,

- устойчивость к сбоям, атакам, ошибкам пользователя и т.д.

- Для системного и компонентного тестирования используются специфические виды критериев тестового покрытия:

- покрыты ли все типовые сценарии работы,

- покрыты ли все сценарии с нестандартными ситуациями, попарные композиции сценариев

- и проч.

Классификация тестирования по времени проведения тестирования

- Альфа - тестирование (alphatesting)
 - Тестирование при приёмке (smoketesting)
 - Тестирование новых функций (newfeaturetesting)
 - Регрессионное тестирование (regressiontesting)
 - Тестирование при сдаче (acceptancetesting)
- Бета - тестирование (betatesting)

Альфа-тестирование и бета-тестирование

являются подкатегориями системного тестирования.

"Альфа-" и "бета-тестирование" относятся к стадиям до выпуска продукта.

Уровни тестирования

