

Лабораторная работа №1

Виды тестирования. Планирование тестирования

Цель: изучить классификацию видов тестирования, разработать проверки для различных видов тестирования, научиться планировать тестовые активности в зависимости от особенностей поставляемой на тестирование функциональности.

План занятия:

1. Изучить теоретические сведения.
2. Выполнить практическое задание по лабораторной работе.
3. Оформить отчет и ответить на контрольные вопросы.

Теоретические сведения

Тестирование (Testing) – процесс анализа программного средства и сопутствующей документации с целью выявления дефектов и повышения качества продукта [1].

Конечной целью тестирования является предоставление пользователю качественного программного обеспечения (ПО) [2].

Качество (Quality) – степень, с которой компонент, система или процесс соответствует зафиксированным требованиям и/или ожиданиям и нуждам пользователя или заказчика [3].

Дефект (defect, bug, ошибка) – ключевой термин тестирования, означающий отклонение фактического результата от ожидаемого. Для обнаружения дефекта необходимо выполнить три условия: знать фактический результат, знать ожидаемый результат, зафиксировать факт разницы между фактическим и ожидаемым результатом.

Процесс тестирования как процесс поиска дефектов сводится к следующей последовательности действий:

1. Узнаем ожидаемый результат.
2. Узнаем фактический результат.
3. Сравниваем ожидаемый и фактический результаты.

Источником ожидаемого результата является спецификация – детальное описание того, как должно работать ПО.

В общем случае любой дефект представляет собой отклонение от спецификации. Важно обнаружить эти дефекты до того, как их найдут конечные пользователи.

Тестирование можно классифицировать по очень большому количеству признаков. Далее приведен обобщенный список видов тестирования по различным основаниям.

1. Виды тестирования в зависимости от объекта тестирования: функциональные, пограничные, нефункциональные (рисунок 1).



Рисунок 1 – Классификация видов тестирования в зависимости от объекта

Рассмотрим функциональные виды тестирования.

Функциональное тестирование (Functional Testing) – тестирование, основанное на сравнительном анализе спецификации и функциональности компонента или системы.

Тестирование безопасности (Safety Testing) – тестирование программного продукта с целью определить его способность при использовании оговоренным образом оставаться в рамках приемлемого риска причинения вреда здоровью, бизнесу, программам, собственности или окружающей среде.

Тестирование защищенности (Security Testing) – тестирование с целью оценить защищенность программного продукта от внешних воздействий (от проникновений). На практике зачастую под термином тестирование безопасности понимают в том числе и тестирование защищенности.

Рассмотрим пограничные виды тестирования.

Тестирование совместимости (Compatibility Testing) – проверка работоспособности приложения в различных средах (браузеры и их версии, операционные системы, их типы, версии и разрядность). Виды тестирования совместимости: кроссбраузерное тестирование (различные браузеры или версии браузеров), кроссплатформенное тестирование (различные операционные системы или версии операционных систем).

Рассмотрим нефункциональные виды тестирования, направленные на проверку характеристик или свойств программы (внешний вид, удобство использования, скорость работы и т. п.).

Тестирование требований (Requirements Testing) – проверка требований на соответствие основным атрибутам качества.

Тестирование прототипа (Prototype Testing) – метод выявления структурных, логических ошибок и ошибок проектирования на ранней стадии развития продукта до начала фактической разработки.

Тестирование пользовательского интерфейса (GUI Testing) – тестирование, выполняемое путем взаимодействия с системой через графический интерфейс пользователя (правописание выводимой информации; расположение и выравнивание элементов GUI; соответствие названий форм/элементов GUI их назначению; унификация стиля, цвета, шрифта; окна сообщений; изменение размеров окна, поведение курсора и горячие клавиши).

Тестирование удобства использования (Usability Testing) – тестирование с целью определения степени понятности, легкости в изучении и использовании, привлекательности программного продукта для пользователя при условии использования в заданных условиях эксплуатации (на этом уровне обращают внимание на визуальное оформление, навигацию, логичность, наличие обратной связи и др.).

Тестирование доступности (Accessibility Testing) – тестирование, которое определяет степень легкости, с которой пользователи с ограниченными способностями могут использовать систему или ее компоненты.

Тестирование интернационализации (Internationalization Testing) – тестирование адаптации продукта к языковым и культурным особенностям целого ряда регионов, в которых потенциально может использоваться продукт.

Тестирование локализации (Localization Testing) – тестирование адаптации продукта к языковым и культурным особенностям конкретного региона, отличного от того, в котором разрабатывался продукт.

Тестирование производительности (Performance Testing) – процесс тестирования с целью определения производительности программного продукта. В рамках тестирования производительности выделяют нагрузочное тестирование, объемное тестирование, тестирование стабильности и надежности, стрессовое тестирование.

Нагрузочное тестирование (Performance and Load Testing) – вид тестирования производительности, проводимый с целью оценки поведения компонента или системы при возрастающей нагрузке, например количестве параллельных пользователей и/или операций, а также определения, какую нагрузку может выдержать компонент или система.

Объемное тестирование (Volume Testing) – позволяет получить оценку производительности при увеличении объемов данных в базе данных приложения.

Тестирование стабильности и надежности (Stability/Reliability Testing) – позволяет проверять работоспособность приложения при длительном (многочасовом) тестировании со средним уровнем нагрузки.

Стрессовое тестирование (Stress Testing) – вид тестирования производительности, оценивающий систему или компонент на граничных значениях рабочих нагрузок, или за их пределами, или же в состоянии ограниченных ресурсов, таких как память или доступ к серверу.

Тестирование на отказ и восстановление (Failover and Recovery Testing) – тестирование при помощи эмуляции отказов системы или реально вызываемых отказов в управляемом окружении.

Тестирование установки (Installability Testing) и лицензирования – процесс тестирования установки программного продукта. Включает формальный тест программы установки приложения (проверка пользовательского интерфейса, навигации, удобства использования, соответствия общепринятым стандартам оформления); функциональный тест программы установки; тестирование механизма лицензирования и функций защиты от пиратства; проверку стабильности приложения после установки.

2. Виды тестирования в зависимости от знания кода: белый ящик, серый ящик, черный ящик.

Белый ящик (White Box Testing) – тестирование, основанное на анализе внутренней структуры компонентов или системы (у тестировщика есть доступ к внутренней структуре и коду приложения).

Черный ящик (Black Box Testing) – тестирование системы без знания внутренней структуры и компонентов системы (у тестировщика нет доступа к внутренней структуре и коду приложения либо в процессе тестирования он не обращается к ним).

Серый ящик (Grey Box Testing) – комбинация методов белого и черного ящика, состоящая в том, что у тестировщика есть доступ только к некоторой части внутренней структуры и кода приложения.

3. Виды тестирования в зависимости от степени автоматизации: ручное, автоматизированное тестирование.

Ручное тестирование – такое тестирование, в котором тест-кейсы выполняются тестировщиком вручную без использования средств автоматизации.

Автоматизированное тестирование (Automated Testing) – набор техник, подходов и инструментальных средств, позволяющий исключить человека из выполнения некоторых задач в процессе тестирования. Тест-кейсы частично или полностью выполняет специальное инструментальное средство.

4. Виды тестирования в зависимости от степени изолированности тестируемых компонентов: модульное, интеграционное, системное тестирование.

Модульное тестирование (Unit/Component Testing) – тестируются отдельные части (модули) системы.

Интеграционное тестирование (Integration Testing) – тестируется взаимодействие между отдельными модулями.

Системное тестирование (System Testing) – тестируется работоспособность системы в целом.

5. Виды тестирования в зависимости от подготовленности: интуитивное тестирование, исследовательское тестирование, тестирование по документации.

Интуитивное тестирование выполняется без подготовки к тестам, без определения ожидаемых результатов, проектирования тестовых сценариев.

Исследовательское тестирование – метод проектирования тестовых сценариев во время выполнения этих сценариев.

Тестирование по документации – тестирование по подготовленным тестовым сценариям, руководству по осуществлению тестов.

6. Виды тестирования в зависимости от места и времени проведения тестирования: приемочное тестирование, альфа-тестирование, бета-тестирование.

Приемочное тестирование (User Acceptance Testing, UAT) – формальное тестирование по отношению к потребностям, требованиям и бизнес-процессам пользователя, проводимое с целью определения соответствия системы критериям приемки

и представления возможности пользователям, заказчикам или иным авторизованным лицам решать, принимать систему или нет.

Альфа-тестирование (Alpha Testing) – моделируемое или действительное функциональное тестирование, выполняется в организации, разрабатывающей продукт, но не проектной командой (это может быть независимая команда тестировщиков, потенциальные пользователи, заказчики). Альфа-тестирование часто применяется к коробочному программному обеспечению в качестве внутреннего приемочного тестирования.

Бета-тестирование (Beta Testing) – эксплуатационное тестирование потенциальными или существующими клиентами/заказчиками на внешней стороне (в среде, где продукт будет использоваться), никак не связанными с разработчиками, с целью определения действительно ли компонент или система удовлетворяет требованиям клиента/заказчика и вписывается в бизнес-процессы. Бета-тестирование часто проводится как форма внешнего приемочного тестирования готового программного обеспечения для того, чтобы получить отзывы рынка.

7. Виды тестирования в зависимости от глубины тестового покрытия: Smoke, MAT, AT.

Тестовое покрытие – одна из метрик оценки качества тестирования, представляющая из себя плотность покрытия тестами требований либо исполняемого кода.

Smoke Test – поверхностное тестирование для определения пригодности сборки для дальнейшего тестирования, должно покрывать базовые функции программного обеспечения; уровень качества: Acceptable/Unacceptable.

Minimal Acceptance Test (MAT, Positive Test) – тестирование системы или ее части только на корректных данных/сценариях; уровень качества: High/Medium/Low.

Acceptance Test (AT) – полное тестирование системы или ее части как на корректных (Positive Test), так и на некорректных данных/сценариях (Negative Test); уровень качества: High/Medium / Low. Тест на этом уровне покрывает все возможные сценарии тестирования: проверку работоспособности модулей при вводе корректных значений; проверку при вводе некорректных значений; использование форматов данных, отличных от тех, которые указаны в требованиях; проверку исключительных ситуаций, сообщений об ошибках; тестирование на различных комбинациях входных параметров; проверку всех классов эквивалентности; тестирование граничных значений интервалов; сценарии, не предусмотренные спецификацией и т. д.

8. Виды тестирования в зависимости от тестовых активностей: NFT, RT, DV. Данная классификация тестирования иначе называется видами тестирования в зависимости от ширины тестового покрытия.

Тестирование новых функциональностей (New Feature Test, NFT) – определение качества поставленной на тестирование новой функциональности, которая ранее не тестировалась. Данный тип тестирования включает в себя: проведение полного теста (АТ) непосредственно новой функциональности; тестирование новой функциональности на соответствие документации; проверку всевозможных взаимодействий ранее реализованной функциональности с новыми модулями и функциями.

Регрессионное тестирование (Regression Testing, RT) проводится с целью оценки качества ранее реализованной функциональности. Включает в себя проверку стабильности ранее реализованной функциональности после внесения изменений, например добавления новой функциональности, исправление дефектов, оптимизация кода, разворачивание приложения на новом окружении. Регрессионное тестирование выполняется на уровнях МАТ, АТ.

Валидация дефектов (Defect Validation, DV) – проверка результатов исправления дефектов; может включать элементы регрессионного тестирования; уровень проверки не определяется.

Процесс тестирования программного продукта включает следующие этапы:

1. Планирование тестирования.
2. Разработка рабочей тестовой документации (тестов).
3. Исполнение тестирования.

Планирование тестирования включает изучение и анализ предмета тестирования, составление тест-плана. На стадии планирования тестирования перед тестирующим стоит задача поиска компромисса между объемом тестирования, который возможен в теории, и объемом тестирования, который возможен на практике. Результирующий тест-план представляет собой документ, описывающий и регламентирующий перечень работ по тестированию, а также соответствующие техники и подходы, стратегию, области ответственности, ресурсы, расписание и ключевые даты.

Разработка тестов, как правило, выполняется до реализации программного обеспечения, основываясь исключительно на спецификации к программному продукту.

Выполнение тестирования начинается после поставки первой версии программного продукта и представляет собой практический поиск дефектов с использованием тестовой документации, составленной ранее, а также контроль исправления обнаруженных дефектов.

Для всех программных продуктов выполняют следующие типы тестов и их композиции.

Для первой поставки программного обеспечения рекомендуется проводить Smoke + NFT_{АТ} готовой функциональности: поверхностное тестирование (Smoke Test) выполняется для определения пригодности сборки для дальнейшего тестирования; полное тестирование системы или ее части как на корректных, так и на некорректных данных/сценариях (Acceptance Test, АТ) позволяет обнаружить дефекты и внести запись о них в багтрекинг-систему.

Для последующих поставок программного обеспечения композиции тестов могут быть следующими.

Если не была добавлена новая функциональность, то DV + RT_{МАТ}. То есть, выполняется проверка исправления дефектов программистом (Defect Validation, DV), а также проверка работоспособности остальной функциональности после исправления дефектов на позитивных сценариях (Minimal Acceptance Test, МАТ).

Если была добавлена новая функциональность, то Smoke + DV + NFT_{АТ} + RT_{МАТ}. В частности, выполняется поверхностное тестирование (Smoke Test), проверка исправления дефектов программистом (Defect Validation, DV), тестирование новых функциональностей (New Feature Testing, NFT), проверка старых функциональностей, т. е. регрессионное тестирование (Regression Test).

Если была добавлена новая функциональность, то возможен также вариант DV + NFT_{АТ} + RT_{МАТ}, т. е. без выполнения Smoke Test.

Таким образом, для второй и последующих поставок обобщенная схема композиции тестов выглядит следующим образом:

Smoke + DV + (NFT_{АТ}) + RT_{МАТ}.

В зависимости от типа и специфики приложения (web, desktop, mobile) выполняют специализированные тесты (например, кроссбраузерное или кроссплатформенное тестирование, тестирование локализации и интернационализации и др.).

Практическое задание:

1. Выбрать объект реального мира (например, карандаш, стол, чашка, клавиатура, сумка и др.) с целью последующей разработки тестовых проверок для него.
2. Разработать различные проверки в соответствии с классификацией видов тестирования для выбранного объекта реального мира. Результаты внести в таблицу 1.1.

Таблица 1.1 – Тестовые проверки для различных видов тестирования

Объект тестирования: указать		
Вид тестирования	Краткое определение	Тестовые проверки

	вида тестирования	
Functional Testing		
Safety Testing		
Security Testing		
Compatibility Testing		
GUI Testing		
Usability Testing		
Accessibility Testing		
Internationalization Testing		
Performance Testing		
Stress Testing		
Negative Testing		
Black Box Testing		
Automated Testing		
Unit/Component Testing		
Integration Testing		

3. Разработать композицию тестов для первой поставки программного обеспечения (build 1), состоящей из трех модулей (модуль 1, модуль 2, модуль 3).

4. Разработать композицию тестов для второй поставки программного обеспечения (build 2): исправлены заведенные дефекты, доставлена новая функциональность – модуль 4.

5. Разработать композицию тестов для третьей поставки программного обеспечения (build 3): заказчик решил расширять рынки сбыта и просит осуществить поддержку программного обеспечения на английском языке.

6. Разработать композицию тестов для четвертой поставки программного обеспечения (build 4): заказчик хочет убедиться, что программное обеспечение выдержит нагрузку в 2000 пользователей.

7. Оформить отчет и защитить лабораторную работу.

Содержание отчета:

1. Цель работы.
2. Разработанные проверки выбранного объекта реального мира для различных видов тестирования.

3. Тестовые активности для сформулированных задач.

4. Выводы по работе.

Контрольные вопросы:

1. Что такое тестирование?
2. Что такое качество программного обеспечения?

3. Что такое дефект?
4. Назовите три условия обнаружения дефекта.
5. Какие существуют виды тестирования в зависимости от объекта тестирования? Дайте характеристику каждому.
6. Какие существуют виды функционального тестирования? Дайте характеристику каждому.
7. Какие существуют виды нефункционального тестирования? Дайте характеристику каждому.
8. Какие существуют виды тестирования в зависимости от глубины покрытия? Дайте характеристику каждому.
9. Какие существуют тестовые активности? Дайте характеристику каждому.
10. Какие существуют виды тестирования в зависимости от знания кода? Дайте характеристику каждому.
11. Какие существуют виды тестирования в зависимости от степени автоматизации? Дайте характеристику каждому.
12. Какие существуют виды тестирования в зависимости от изолированности компонентов? Дайте характеристику каждому.
13. Какие существуют виды тестирования в зависимости от подготовленности? Дайте характеристику каждому.
14. Какие существуют виды тестирования в зависимости от места и времени проведения? Дайте характеристику каждому.
15. Какие этапы составляют процесс тестирования?
16. Какая композиция тестов выполняется для первой поставки программного продукта?
17. Какая композиция тестов выполняется для последующих поставок программного продукта?