

Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

Дисциплина: Тестирование, оценка программного обеспечения

Отчет
по Лабораторной работе №1
на тему: Виды тестирования. Планирование тестирования

Студент
гр. 910902

Ромашко И.А.

Проверил

Кабариха В.А.

Минск 2022

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

1. Разработать различные проверки в соответствии с классификацией видов тестирования для выбранного объекта реального мира. Результаты внести в таблицу.

Объект тестирования: сумка		
Вид тестирования	Краткое определение вида тестирования	Тестовые проверки
Functional Testing	Тестирование, основанное на анализе спецификации и функциональности компонента или системы	Можно носить вещи? Закрывается?
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	Замок прищемляет палец? Ручка натирает плечо?
Security Testing	Тестирования, оценивающее защищенность программного продукта от внешних воздействий	Испортится на морозе? Если поставить на пол, испачкается?
Compatibility Testing	Проверка работоспособности приложения в различных средах	Место использование: помещение, улица Место хранение: шкаф, полка
GUI Testing	Тестирование, путем взаимодействия с системой через графический интерфейс	Какого цвета? Какая форма? Надписи есть?
Usability Testing	Тестирования, определяющее понятность, простоту в изучении и использовании программного продукта для пользователя	Удобно носить? Держится на плече?
Accessibility Testing	Тестирование, которое определяет степень легкости, с которой	Положить вещи в сумку закрытыми глазами.

	пользователи с ограниченными способностями могут использовать систему или ее компоненты	Застегнуть сумку закрытыми глазами
Internationalization Testing	Тестирование адаптации продукта к языковым и культурным особенностям целого ряда регионов	В сумку складывают вещи, которыми пользуются в различных регионах
Performance Testing	Процесс тестирования с целью определения производительности программного продукта	Сколько можно носить?
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	Уронить Дернуть за ручки
Negative Testing	Тестирование некорректных сценариев/данных	Порезать Оторвать ручки
Black Box Testing	Тестирование системы без знания внутренней структуры и компонентов системы	Сложить вещи в сумку и пройти с ней
Automated Testing	Набор техник, подходов и инструментальных средств, позволяющий исключить человека из выполнения некоторых задач в процессе тестирования	Автоматически оторвать ручки, порезать
Unit/Component Testing	Тестируются отдельные части (модули) системы	Отдельно закрываем/открываем молнию
Integration Testing	Тестируется взаимодействие между отдельными модулями	Ручки не отрываются? Замок не отрывается?

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

Модуль 1:

Smoke Test - поверхностное тестирование для определения пригодности сборки для дальнейшего тестирования, должно покрывать базовые функции

программного обеспечения.

Модуль 2:

New Feature Test - 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.

Модуль 3:

Acceptance Test - полное тестирование системы или ее части как на корректных, так и на некорректных данных/сценариях. 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.

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

Модуль 1: Smoke Test

Модуль 2:

Defect Validation - проверка результатов исправления дефектов.

Модуль 3: NFT_{AT}

Модуль 4:

Regression Testing - 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 как правило выполняется на уровне МАТ.

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

Т.к. никаких изменений не было внесено, то тестировать будем по схеме $DV+RT_{МАТ}$, т.е. выполняется проверка исправления дефектов программистом DV, а также проверка работоспособности остальной функциональности после исправления дефектов на позитивных сценариях МАТ.

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

Поскольку никаких изменений не было внесено, то тестировать будем по схеме

$DV+RT_{MAT}$

Для проверки выдержит ли ПО нагрузку в 2000 пользователей, используем *Performance Test*

Вывод: в данной лабораторной работе были изучены виды тестирования, разработаны проверки для выбранного предмета. Были получены знания о планировании тестовых активностей.