

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования
Кафедра инженерной психологии и эргономики

ОТЧЕТ
к лабораторной работе
на тему

Виды тестирования. Планирование тестирования

Студент

Г.Г.Митенёв

Руководитель

В.А. Кабариха

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.

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: зажигалка Zippo		
Вид тестирования	Краткое определение вида тестирования	Тестовые проверки
Functional Testing	– тестирование, основанное на сравнительном анализе спецификации и функциональности компонента или системы.	<ol style="list-style-type: none"> 1. Зажигалка имеет бензобак, в который заливается воспламеняющаяся жидкость 2. Зажигалка имеет кремниевый бегунок для высекаания искр 3. Зажигалка имеет подпружиненную крышку, для открывания и закрывания зажигалки
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.	<ol style="list-style-type: none"> 1. Проверить, что на зажигалке нет острых, царапающих или режущих выступов 2. Проверить, чтобы пустая зажигалка не издавала резких запахов и загрязняла окружающую среду

Security Testing	– тестирование с целью оценить защищенность программного продукта от внешних воздействий (от проникновений).	<ol style="list-style-type: none"> 1. Проверить герметичность бензобака зажигалки 2. Проверить возможность самопроизвольного открывания подпружиненной крышки
------------------	--	---

Compatibility Testing	–проверка работоспособности приложения в различных средах (браузеры и их версии, операционные системы, их типы, версии и разрядность)	<ol style="list-style-type: none"> 1. Проверить работоспособность зажигалки в различных погодных условиях (например, в условиях экстремально жаркой или холодной погоды, большой влажности и т.д.)
GUI Testing	–тестирование, выполняемое путем взаимодействия с системой через графический интерфейс пользователя.	<ol style="list-style-type: none"> 1. Проверить на наличие на донышке зажигалки наличие надписи Zippo 2. Проверка правильности дополнительных надписей и товарных знаков на зажигалке
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.	<ol style="list-style-type: none"> 1. Проверка зажигалки на прокручиваемость кремниевого колёсика 2. Проверка зажигалки на заполняемость бака

Accessibility Testing	– тестирование, которое определяет степень легкости, с которой пользователи с ограниченными способностями могут использовать систему или ее компоненты.	<ol style="list-style-type: none"> 1. Проверить зажигалку на возможность использования при болезни Паркинсона или других дисфункциях опорно-двигательного аппарата 2. Проверить наличие выступов, удобных для использования слабовидящими или шрифта Брайля
Internationalization Testing	– тестирование адаптации продукта к языковым и культурным особенностям целого ряда регионов, в которых потенциально может использоваться продукт.	<ol style="list-style-type: none"> 1. Проверить наличие товарных и иных знаков на иностранных языках 2. Проверить отсутствие оскорбительных надписей на иностранных языках

Performance Testing	– процесс тестирования с целью определения производительности программного продукта.	<ol style="list-style-type: none"> 1. Проверить герметичность бензобака при его заполнении 2. Проверить работу крышки после её раскрытия и закрытия 3. Проверить исправность колёсика при его прокручивании
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	<ol style="list-style-type: none"> 1. Проверить целостность зажигалки после нескольких ударов 2. Проверить работу крышки после многократного её раскрытия и закрытия 3. Проверить кремниевое покрытие после многократного использования

Negative Testing	- тестирование, в рамках которого применяются сценарии, которые соответствуют внештатному поведению тестируемой системы.	<ol style="list-style-type: none"> 1. Определить работоспособность зажигалки после контакта с водой 2. Проверка зажигаемости фитиля после его расслоения
Black Box Testing	– тестирование системы без знания внутренней структуры и компонентов системы	<ol style="list-style-type: none"> 1. Проверка работы зажигалки и её использования без прямого доступа к её бензобаку
Automated Testing	– набор техник, подходов и инструментальных средств, позволяющий исключить человека из выполнения некоторых задач в процессе тестирования.	<ol style="list-style-type: none"> 1. Проверить высекаемость искр при прокручивании кремниевого колёсика роботом 2. Проверить возможность автоматического заполнения бака

Unit/Component Testing	– тестируются отдельные части (модули) системы.	<ol style="list-style-type: none"> 1. Проверка крышки/бензобака рюкзака 2. Проверка фитиля/колёсика
Integration Testing	–тестируется взаимодействие между отдельными модулями.	<ol style="list-style-type: none"> 1. Проверка передачи искры от колёсика к фитилю 2. Проверка намокаемости части фитиля жидкостью из бака

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

build №1: Smoke + NFTAT

Smoke – проверка первых трех модулей и определения возможности дальнейшего тестирования. Если дальнейшее тестирование возможно, надо проверить каждый модуль до уровня АТ.

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

Build №2: Smoke + DV + NFTAT + RTMAT.

Smoke тестирование проверяет работу основных функций, после необходимо провести верификацию исправленных багов, сделать регрессионное тестирование старых функциональностей до уровня МАТ и проверить новый модуль до уровня АТ.

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.

Build№3: Smoke и тестирование локализации с поддержкой английского языка. Проверка всех старых функциональностей, регрессионное тестирование по уровню МАТ и проверка исправленности багов.

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

Build№4: Smoke + DV и тестирование производительности в 2000+ пользователей.

Выводы:

При выборе вида тестирования необходимо выбирать те, которые обеспечивают проверку различных аспектов продукта (нефункциональных и функциональных). Планирование тестирования зависит от входных данных при поставке модулей продукта.

