Министерство образования и науки Международного Университета Кыргызской Республики

International University Of Kyrgyzstan

Факультет: Программная Инженерия

СРС

Тема: Классификация тестирования (Белый ящик)

Группа: Пи1-21

Проверила: Досболова Фарида

Выполнил: А. у Кубанычбек

Бишкек 2022

**Глава 1**

**Классификация тестирования**

**1. Цель**

Каждый программный продукт должен выполнять одну или несколько ключевых задач. От приложения с гео-картами мы ожидаем точной ориентации в пространстве, от сайта интернет-магазина ― корректного поиска товаров по заданным параметрам и т. д. Но те же программные продукты мы можем протестировать и с точки зрения дизайна.

Таким образом, анализ ПО с позиции его ключевых или вспомогательных функций определяет тип тестирования:

* **Функциональное**
* **Нефункциональное**

Функциональное тестирование направлено на проверку того, какие функции ПО реализованы, и того, насколько верно они реализованы.

Нефункциональное – проверка корректности работы нефункциональных требований. Оценивается, КАК программный продукт работает. Эта проверка включает в себя следующие виды:

* **Тестирование производительности** – работа ПО под определённой нагрузкой.
* **Тестирование пользовательского интерфейса** – удобство пользователя при взаимодействии с разными параметрами интерфейса (кнопки, цвета, выравнивание и т. д.).
* **Тестирование UX** – правильность логики использования программного продукта.
* **Тестирование защищенности** – определение безопасности ПО: защищено ли оно от атак хакеров, несанкционированного доступа к данным и т. д.
* **Инсталляционное тестирование** – оценка вероятности возникновения проблем при установке, удалении, а также обновлении ПО.
* **Тестирование совместимости** – тестирование работы программного продукта в определённом окружении.
* **Тестирование надежности** – работа программы при длительной средней ожидаемой нагрузке.
* **Тестирование локализации** –оценка правильности версии программного продукта (языковой и культурный аспекты).

1

**2. Степень автоматизации**

В зависимости от того, используют ли тестировщики дополнительные программные средства для тестирования приложений или программ, тестирование бывает:

* **Мануальное (ручное)** – без использования дополнительных программных средств, т. е. «вручную».
* **Автоматизированное** – с использованием программных средств (более детально в описании курса по автоматизации тестирования ПО).

Каждый из подходов имеет свои преимущества и недостатки. Ручное тестирование проще освоить, оно широко применяется на проектах всех типов, но мануальные проверки отличаются монотонностью. А вот написание тестов даёт больше возможностей для творческой реализации, но автоматизация требует базовых навыков программирования.

**2**

**3. Позитивность сценария**

Этот подход определяет поведение системы в привычных и экстремальных условиях.

* **Позитивная проверка** – оценка ожидаемого поведения. Это тестирование проводится в первую очередь, ведь позволяет определить корректность работы программы.
* **Негативная** – определение устойчивости системы в нестандартной ситуации. Например, неожиданный сценарий взаимодействия пользователя с интерфейсом.

Эти типы тестирования нередко проводятся параллельно. Ведь работая над некоторой функциональностью, тестировщику проще оценить её поведение и в стандартных, и в нестандартных условиях.

**3**

**4. Доступ к коду программного продукта**

В процессе тестирования инженер может работать с ПО, не обращаясь к его коду, а может определить правильность работы, взглянув на код. По доступу к коду программного продукта тестирование делится на:

* **Тестирование «белого ящика»** – с доступом к коду.
* **Тестирование «черного ящика»** – без доступа к коду продукта.
* **Тестирование «серого ящика»** – на основе ограниченного знания внутренней структуры ПО. Часто говорят, что это смесь тестирования «белого ящика» и «чёрного ящика», но это в корне неверно. В данном случае тестировщик не работает с кодом программного продукта, но он
* знаком с внутренней структурой программы и взаимодействием между компонентами.

## 4

## 5. Уровень

Этот пункт определяет объект тестирования.

* **Модульное / юнит-тестирование** – проверка корректной работы отдельных единиц ПО, модулей. Этот вид тестирования могут выполнять сами разработчики.
* **Интеграционное тестирование** – проверка взаимодействия между несколькими единицами ПО.
* **Системное** – проверка работы приложения целиком.
* **Приёмочное** – оценка соответствия заявленным требованиям к программному продукту.

Переход на каждую новую ступень – движение от микроуровня к макро. Это важный этап тестирования, ведь безошибочно написанные модули могут просто не работать вместе.

5

**6. Исполнитель**

От объекта тестирования движемся к его субъекту. Вы могли слышать об альфа- и бета-тестировании. А поучаствовать в одном из них можно, даже не будучи тестировщиком. Итак, по исполнителю тестирование делится на:

* **Альфа-тестирование** – проверка программного продукта на поздней стадии разработки. Проводится разработчиками или тестировщиками.

Тестирование. По на ранних этапах разработки. Проводится силами команды разработчиков продукта. На этом этапе выполняются много ошибок и реализации ПО и проверяются некоторые архитектурные решения.

* **Бета-тестирование** – оценка ПО перед выходом на рынок в фокус-группе или добровольцами. Отзывы собираются, анализируются и учитываются при внесении правок.

**Отличие:**

В отличие от альфа-тестирования, проводимого силами штатных разработчиков или тестировщиков, бета-тестирование предполагает привлечение добровольцев из числа обычных будущих пользователей продукта, которым доступна предварительная версия продукта (так называемая бета-версия).

**Как проводится альфа тестирование?**

Обычно альфа тестирование заключается в систематической проверке всех функций программы с использованием техник тестирования «белого ящика» и «черного ящика» (подробнее о них тут). Альфа-тестирование является методологией оценки качества и стабильности тестируемого продукта в тестовой среде.

**Что лучше альфа или бета?**

Альфа-версия — вроде всё сделали, протестировали самое основное. Бета-версия — оттестировали большую часть, ловим тараканов при поддержке небольшого круга доверенных людей. Релиз-кандидат — почти готовая к выпуску программа.

6

**7. Формальность**

Определяет подготовленность тестировщика перед началом проверки программы.

* **Тестирование по тестам** – использование заранее написанных тест-кейсов.
* **Исследовательское тестирование** – одновременная разработка тестов и их использование.
* **Свободное тестирование** – проверка качества без разработки тестов и написания документации. Основывается на интуиции и опыте тестировщика.

Начинающие тестировщики редко работают на свободном уровне. А вот опытные QA-специалисты могут позволить себе проверку без дополнительной подготовки. Мастерство растёт со временем, как и оплата труда тестировщика.

7

**Глава2**

**White box**

Тестирование методом белого ящика (также: прозрачного, открытого, стеклянного ящика; основанное на коде или структурное тестирование) – метод тестирования программного обеспечения, который предполагает, что внутренняя структура/устройство/реализация системы известны тестировщику. Мы выбираем входные значения, основываясь на знании кода, который будет их обрабатывать. Точно так же мы знаем, каким должен быть результат этой обработки. Знание всех особенностей тестируемой программы и ее реализации – обязательны для этой техники. Тестирование белого ящика – углубление во внутренне устройство системы, за пределы ее внешних интерфейсов.

Согласно ISTQB:

**тестирование белого ящика – это:**

– тестирование, основанное на анализе внутренней структуры компонента или системы.

– тест-дизайн, основанный на технике белого ящика – процедура написания или выбора тест-кейсов на основе анализа внутреннего устройства системы или компонента.

Почему «белый ящик»? Тестируемая программа для тестировщика – прозрачный ящик, содержимое которого он прекрасно видит.

Тестировщик, который, как правило, является программистом, изучает реализацию кода поля ввода на веб-странице, определяет все предусмотренные (как правильные, так и неправильные) и не предусмотренные пользовательские вводы, и сравнивает фактический результат выполнения программы с ожидаемым. При этом ожидаемый результат определяется именно тем, как должен работать код программы.

Тестирование методом белого ящика похоже на работу механика, который изучает двигатель машины, чтобы понять, почему она не заводится.

Техника белого ящика применима на разных уровнях тестирования – от модульного до системного, но главным образом применяется именно для реализации модульного тестирования компонента его автором.

**Преимущества:**

– тестирование может производиться на ранних этапах: нет необходимости ждать создания пользовательского интерфейса;

– можно провести более тщательное тестирование, с покрытием большого количества путей выполнения программы.

**Недостатки:**

– для выполнения тестирования белого ящика необходимо большое количество специальных знаний

– при использовании автоматизации тестирования на этом уровне, поддержка тестовых скриптов может оказаться достаточно накладной, если программа часто изменяется.

**8**

**Сравнение Black Box и White Box**



9

Источники:

https://qa-academy.by/qaacademy/news/klassifikaciya-vidov-testirovaniya/

10

Содержание

**Глава11**

Классификация тестирования1

1. Степень автоматизации2

2. Позитивность сценария3

2.1 Позитивная проверка5

3.1Негативная5

3. Доступ к коду программного продукта4

3.1 Тестирование белого ящика 5

3.2 Тестирование черного ящика 5

3.3 Тестирование серого ящика 5

4. Уровень5

4.1 Модульное / юнит-тестирование 5

4.2 Интеграционное5

4.3 Системное 5

4.4 Приемочное5

5. Исполнитель6

5.1 Альфа тестирование6

5.2 Бета тестирование6

6 Формальность7

6.1 Тестирование по тестам7

6.2 Исследовательское тестирование7

6.3 Свободное тестирование7

**Глава28**

White Box8

Сравнение Black Box и White Box9

Источники10

Содержание11

11