**ТРПО: ТЕСТИРОВАНИЕ И ОТЛАДКА ПО**

**Таблица 1 – Уровни тестирования**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Название уровня** | **Сущность (характеристика уровня)** | **Когда осуществляется** | **Кто осуществляет** |
| **1** | **Модульное (компонентное)** | Проверка отдельных модулей или компонентов программы (функций, методов, классов) на корректность и правильность их работы. | Обычно проводится в начале разработки после написания каждого модуля. | Разработчики или тестировщики. |
| **2** | **Интеграционное** | Проверка взаимодействия между разными модулями или компонентами программы, а также их взаимосвязей и взаимодействия с внешними системами. | Проводится после завершения модульного тестирования и интеграции компонентов. | Тестировщики, иногда с участием разработчиков. |
| **3** | **Системное** | Проверка всей системы как единого целого, включая взаимодействие всех ее компонентов и соответствие функциональным и нефункциональным требованиям. | Проводится после интеграционного тестирования и завершения разработки системы. | Специализированные тестировщики или QA-инженеры.м |
| **4** | **Приемочное** | Проверка системы на соответствие конечным пользователям, клиентам и их требованиям. Пользователи выполняют тестирование с точки зрения реального использования продукта. | Проводится перед выпуском продукта в эксплуатацию или внедрением. | Конечные пользователи, клиенты, бизнес-аналитики.м |
| **5** | **Выходное** | Повторное тестирование продукта после внесения изменений или исправлений для убедительности в отсутствии новых ошибок и сохранении работоспособности старых функций. | Проводится на разных этапах разработки и внедрения после каждого изменения в продукте. | Тестировщики, иногда с участием разработчиков. |

**Таблица 2 – Стратегии тестирования (По доступу к коду и архитектуре приложения)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Название стратегии** | **Сущность (характеристика стратегии)** | **Преимущества** | **Недостатки** | **Код осуществляет** |
| **1** | **Метод белого (прозрачного) ящика** | Метод белого ящика проверяет внутреннюю структуру и логику программы. Он основан на знании внутренних деталей реализации и кода программы. | Позволяет проверить все пути выполнения программы.  Эффективно для выявления ошибок внутри кода, включая ошибки логики. | Требует знания внутренней структуры программы.  Тестирование всех путей может быть трудоемким. | Тесты, основанные на анализе кода (например, юнит-тесты с использованием библиотеки unittest в Python). |
| **2** | **Метод черного ящика** | Метод черного ящика проверяет программу без знания внутренних деталей реализации. Тестировщик рассматривает программу как "черный ящик" и проверяет ее на основе входных данных и ожидаемых результатов. | Не требует знания внутренней реализации программы.  Основан на функциональных и поведенческих требованиях. | Не всегда способен выявить сложные ошибки внутри кода. | Тесты, основанные на функциональных требованиях и спецификациях, а также сценариях использования. |
| **3** | **Метод серого ящика** | Метод серого ящика сочетает элементы метода белого ящика и метода черного ящика. Тестировщик имеет ограниченное знание внутренней структуры программы и использует это знание для проведения тестов. | Объединяет преимущества обоих методов.  Эффективен для выявления ошибок и аномалий как внутри кода, так и на уровне интерфейса. | Может потребовать больше усилий, чем метод черного ящика. | Тесты, которые учитывают как внутреннюю структуру программы, так и внешнее поведение. |

**Таблица 3 – Отладка, программные ошибки**

|  |  |  |
| --- | --- | --- |
| **№** | **Понятие** | **Характеристика** |
| **1** | **Отладка** | Отладка - это процесс поиска и устранения ошибок (багов) в программном коде с целью обеспечения правильной работы программы. Включает в себя исследование и анализ кода, выявление и исправление ошибок, а также проверку кода на предмет правильности выполнения. |
| **2** | **Инструменты отладчика** | Инструменты отладчика - это программы или функциональные компоненты, предназначенные для облегчения процесса отладки. Они предоставляют возможности для пошагового выполнения кода, установки точек останова, просмотра значений переменных, анализа стека вызовов и других инструментов для выявления и устранения ошибок. |
| **3** | **Программная ошибка** | Программная ошибка (или баг) - это неправильное или нежелательное поведение программы, которое приводит к некорректным результатам или сбоям в работе. Программные ошибки могут возникать из-за различных причин, включая ошибки в коде, неправильную логику, некорректные данные и другие факторы. |
| **4** | **Виды программных ошибок (перечислить)** | Синтаксические ошибки: Ошибки, связанные с неверным синтаксисом программного кода.  Логические ошибки: Ошибки, когда код компилируется без ошибок, но не выполняет желаемую функцию из-за некорректной логики.  Работающие ошибки: Ошибки, которые проявляются только в определенных сценариях или условиях.  Динамические ошибки: Ошибки, которые проявляются во время выполнения программы.  Неявные ошибки: Ошибки, которые сложно обнаружить и которые не вызывают явных ошибок или сбоев. |
| **5** | **Тест-кейс** | Тест-кейс - это документированное описание теста, который проверяет определенный аспект функциональности программы или системы. Он включает в себя входные данные, ожидаемый результат и шаги, которые необходимо выполнить для проведения теста. Тест-кейсы используются для систематического тестирования программы с целью выявления ошибок. |
| **6** | **Отчёт о результатах тестирования** | Отчёт о результатах тестирования - это документ, который содержит информацию о результатах проведенных тестов. Он включает в себя информацию о проведенных тест-кейсах, выявленных ошибках, статусе тестирования и другие данные. Отчёт о результатах тестирования помогает команде разработки и тестирования принимать решения о качестве продукта и необходимости исправлений. |

Таблица 6 – Отчёт результатах тестирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Тест | Ожидаемый результат | Физический результат | Результат  тестирования |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Проверка полей ввода экрана «Вход» | Заполнение данных | Заполнение данных | Выполнено |
| 2 | Проверка кнопки «Регистрация» экрана «Вход» | Переход на экран «Вход» | Переход на экран «Вход» | Выполнено |
| 3 | Проверка полей ввода экрана «Регистрация» | Заполнение данных | Заполнение данных | Выполнено |
| 4 | Проверка кнопки «Вход» экрана «Регистрация» | Переход на экран «Регистрация» | Переход на экран «Регистрация» | Выполнено |
| 5 | Проверка кнопки «Зарегестрироваться» | Переход домашний экран | Переход домашний экран | Выполнено |
| 6 | Проверка кнопки «Меню» домашнего экрана | Открытие меню | Открытие меню | Выполнено |
| 7 | Проверка сворачиваемости меню | Закрытие меню | Закрытие меню | Выполнено |