

План тестирования

Содержание

1. Введение
2. Объект тестирования
 - 2.1 Описание проекта
 - 2.2 Атрибуты качества
3. Риски
4. Аспекты тестирования
 - 4.1 Обоснованность применения
 - 4.2 Правильность, точность
 - 4.3 Мультиплатформенность
 - 4.4 Уровень завершенности
 - 4.5 Устойчивость к дефектам
 - 4.6 Понятность
 - 4.7 Привлекательность
 - 4.8 Время работы
 - 4.9 Исползования ресурсов
 - 4.10 Стабильность
 - 4.11 Сосуществование
5. Подходы к тестированию
6. Представления результатов
7. Заключение

1. Введение

План тестирования (Test Plan) - это документ, описывающий весь объем работ по тестированию, начиная с описания объекта, стратегии, критериев начала и окончания тестирования, до необходимого в процессе работы инструментария, а также оценки рисков с вариантами их разрешения. План тестирования данного проекта разработан для выше описанных целей, достижение которых поможет продукту повысить свою стабильность, устойчивость к ошибкам и исправить все неверные последовательности взаимодействий, выявленные в процессе тестирования.

2. Объект тестирования

2.1 Описание проекта

Проект под названием "Easy HTML" разработан для круга пользователей, тесно работающих или каким-то образом связанных с программированием на языке разметки. Программа призвана упростить добавление целого ряда тегов, тем самым упрощая задачу создания сайта.

Пользователю предоставляется ряд дополнительных возможностей:

- 1) Открытие файла
- 2) Сохранение файла
- 3) Предпросмотр написанного кода в виде веб-страницы.

2.2 Атрибуты качества

Модель качества, установленная в первой части стандарта ISO 9162-1, классифицирует качество ПО в 6-ти структурных наборах характеристик, которые в свою очередь детализированы субхарактеристиками, такими как:

- Функциональность
- Надежность
- Практичность
- Эффективность
- Сопровождаемость
- Мобильность

В контексте разрабатываемого обеспечения можно выделить следующие атрибуты качества в соответствии с вышеописанным стандартом:

- Функциональность
 - а) Обоснованность применения
 - б) Правильность, точность
 - в) Мультиплатформенность
- Надежность
 - а) Уровень завершенности
 - б) Устойчивость к дефектам
- Практичность
 - а) Понятность
 - б) Привлекательность
- Эффективность

- а) Время работы
- б) Использование ресурсов
- Сопровождаемость
 - а) Стабильность
- Мобильность
 - а) Сосуществование

3. Риски

Внесение изменений в старые стандарты HTML или создание новых влекут за собой отсутствие возможности добавления новых тегов в код, в данном ПО. Однако все теги, необходимые для написания любого сайта имеются в наличии у разработчика.

4. Аспекты тестирования

4.1 Обоснованность применения

Требуется проверить, насколько требования, описанные в соответствующей документации к данному проекту, соблюдены в приложении с точки зрения комплексной оценки использования программы.

4.2 Правильность, точность

Тестирование данного аспекта позволит установить – соблюдены ли функциональные требования, реализация которых, определяет основной функционал ПО. Также в данном аспекте тестирования пользователь может удостовериться в соответствии стандарта, реализованного в приложении, стандарту HTML5.

4.3 Мультиплатформенность

По результатам тестирования данного пункта, установим – работоспособна ли программа на различных платформах.

4.4 Уровень завершенности

Проверка завершенности всех функциональных модулей приложения. Все ли функции, которые предоставляет интерфейс, оно выполняет в полной мере.

4.5 Устойчивость к дефектам

Корректность выполнения всех функций, описанных в предыдущем подпункте.

4.6 Понятность

Доступность, простота и легкость в использовании ПО. Тестирующий должен понимать, что ему нужно сделать для достижения той или иной цели. В этом случае аспект тестирования будет подтвержден успешно.

4.7 Привлекательность

Приложение должно быть способным заинтересовать своим внешним видом и привлечь внимание пользователя. Данный аспект является скорее опциональным, ввиду наличия разрозненности мнений касательно внешнего вида ПО.

4.8 Время работы

Эффективность, как один из наиболее важных факторов работы

приложения зависит от аппаратных данных вычислительной системы и платформы, однако пользователь, при тестировании все равно должен убедиться в приемлемом быстродействии. На выполнение любой операции, должно уходить адекватное время работы приложения.

4.9 Использование ресурсов

При тестировании данного аспекта требуется убедиться, что приложение корректно использует сторонние ресурсы: браузер по умолчанию, текстовый редактор, файловая система.

4.10 Стабильность

В ходе тестирования, пользователь должен установить отсутствие случайных и непреднамеренных ошибок путем использования программы в течение некоторого промежутка времени или используя определенные последовательности действий для удовлетворения собственных интересов.

4.11 Сосуществование

Требуется установить, что использование разработанного продукта в системе не препятствует выполнению других процессов или программ. Также данный аспект тестирования подразумевает собой использование стороннего ПО для корректного выполнения собственных функций.

5. Подходы к тестированию

Все аспекты тестирования (п.п. 4.1 – 4.11) следует протестировать с помощью пользовательских тестов. Некоторые из описанных ранее аспектов требуют дополнительных условий. Так для проверки аспекта Мультиплатформенности, тестирующий должен иметь дополнительную систему для проверки разрабатываемого приложения в полной мере.

6. Представления результатов

Каждый из тестовых сценариев подразумевает собой пошаговое выполнение вариантов использования. При тестировании следует учитывать альтернативные потоки выполнения. Описанные в них обходы ошибок следует применять и при тестировании. Таким образом гарантируется правильность работы всей программы.

Не все аспекты тестирования могут быть проверены с помощью тестовых сценариев, представленных в виде вариантов использования. Таким образом, п.п. 4.1, 4.3, 4.6, 4.11 следует тестировать используя либо другие методы, либо другой инструментарий.

Обоснованность применения (п. 4.1), сосуществование (п. 4.11), а также понятность (п. 4.6) может быть проверен лишь путем комплексного использования ПО пользователем в течение конкретного промежутка времени.

Мультиплатформенность требует наличия дополнительной платформы в том или ином виде. Достаточно протестировать запуск приложения хотя бы на двух (например – Windows и UNIX) для определения результата проверки данного аспекта.

7. Заключение

При выполнении всех вариантов использования (тестовых сценариев), а также других методов, описанных в п.6, тестируемое приложение подвергается полной проверке каждого аспекта тестирования. Из результатов данного тестирования пользователь, он же разработчик, сможет определить – является ли продукт пригодным к использованию, и насколько заявленные требования совпадают с реальной реализацией приложения.