

# 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
- Эффективность
  - а) Стабильность

- 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 Стабильность(эффективность)**

При тестировании требуется установить, что при многократном использовании функции, приложение исполняет его без изменений.

#### **4.8 Использование ресурсов**

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

#### **4.9 Стабильность(сопровожаемость)**

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.

#### ***4.10 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***

#### ***6.1 Тестовый сценарий обоснованности применения***

##### ***Идентификатор***

Обоснованность применения.

##### ***Назначение***

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

##### ***Сценарий***

1. Нажать кнопку “Open”.
2. Выбрать .html или текстовый файл для открытия.
3. Установить курсор поля отображения кода в случайную позицию.
4. Нажать на любую кнопку тега (например, “ol”).
5. Заполнить поля атрибутов случайной информацией.
6. Нажать кнопку “Add”.
7. Нажать кнопку “Save”.
8. Ввести имя файла результата и выбрать его местоположение.
9. Нажать кнопку “Preview”.
10. Нажать кнопку “Exit”.

##### ***Ожидаемый результат***

Все функции, описанные в требованиях присутствуют в приложении.

#### ***6.2 Тестовый сценарий правильности***

##### ***Идентификатор***

Правильность, точность.

##### ***Назначение***

Тестирование проводится с целью установить, соблюдены ли функциональные требования, реализация которых, определяет основной функционал ПО. Также в данном аспекте тестирования можно

удостовериться в соответствии стандарта, реализованного в приложении, стандарту HTML5.

#### *Сценарий*

1. Найти заведомо правильный html-код страницы.
2. С помощью программных средств воспроизвести найденный код.
3. Нажать кнопку “Preview”.
4. Сохранить полученный файл.
5. Сравнить оригинал и результат работы приложения.

#### *Ожидаемый результат*

Оригинал страницы и результат работы приложения совпадают.

### **6.3 Тестовый сценарий мультиплатформенности**

#### *Идентификатор*

Мультиплатформенность.

#### *Назначение*

По результатам тестирования данного пункта устанавливается работоспособна ли программа на различных платформах.

#### *Сценарий*

1. Запустить приложение в ОС Windows.
2. Пройти тестовый сценарий уровня завершённости.
3. Запустить приложение в Unix-системе.
4. Пройти тестовый сценарий уровня завершённости.

#### *Ожидаемый результат*

Приложение запускается под обе ОС, тестовые сценарии уровня завершённости пройдены.

### **6.4 Тестовый сценарий уровня завершённости**

#### *Идентификатор*

Завершённость.

#### *Назначение*

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

#### *Сценарий*

1. Нажать кнопку “Open”.
2. Выбрать .html или текстовый файл для открытия.
3. Установить курсор поля отображения кода в случайную позицию.
4. Нажать на любую кнопку тега (например, “head”).
5. Заполнить поля атрибутов случайной информацией.
6. Нажать кнопку “Add”.
7. Нажать кнопку “Save”.
8. Ввести имя файла результата и выбрать его местоположение.
9. Нажать кнопку “Preview”.
10. Нажать кнопку “Exit”.

#### *Ожидаемый результат*

Все функции, которые описаны в требованиях, выполняются.

### **6.5 Тестовый сценарий устойчивости к дефектам**

#### *Идентификатор*

Устойчивость к дефектам.

### *Назначение*

Проверка корректности выполнения всех функций, описанных в требованиях.

### *Сценарий*

1. Нажать кнопку “Open”.
2. Выбрать .html или текстовый файл для открытия.
3. Установить курсор поля отображения кода в случайную позицию.
4. Нажать на любую кнопку тега (например, “h1”).
5. Заполнить поля атрибутов случайной информацией.
6. Нажать кнопку “Add”.
7. Нажать кнопку “Save”.
8. Ввести имя файла результата и выбрать его местоположение.
9. Нажать кнопку “Preview”.
10. Нажать кнопку “Exit”.

### *Ожидаемый результат*

Все функции приложения выполняются корректно, согласно требованиям.

## *6.6 Тестовый сценарий понятности*

### *Идентификатор*

Понятность.

### *Назначение*

Тестирование доступности, простоты и легкости в использовании ПО.

### *Сценарий*

1. Открыть любой html-файл.
2. Добавить после тега <body> в файл тег “table” с атрибутом class=”table1”.
3. Сохранить файл.
4. Посмотреть полученный результат.

### *Ожидаемый результат*

Сценарий тестирования пройден с использованием средств интерфейса без предварительной подготовки.

## *6.7 Тестовый сценарий стабильности(эффективность)*

### *Идентификатор*

Стабильность(эффективность).

### *Назначение*

При тестировании требуется установить, что при многократном использовании функции, приложение исполняет его без изменений.

### *Сценарий*

1. 5 раз нажать кнопку “Open” и выбрать один и тот же html-файл.
2. 5 раз нажать кнопку “ul” и заполнить поля атрибутов одинаковыми данными.
3. 5 раз нажать кнопку “Save” и выбрать одно и то же имя и местоположение.
4. 5 раз нажать кнопку “Preview”.

### *Ожидаемый результат*

При каждом использовании одной и той же функции приложение

реагирует одинаково.

#### **6.8 Тестовый сценарий использования ресурсов**

*Идентификатор*

Использование ресурсов

*Назначение*

Проверка корректного использования сторонних ресурсов, таких как файловая система, браузер по умолчанию.

*Сценарий*

1. Нажать кнопку “Open”.
2. Выбрать .html или текстовый файл.
3. Нажать кнопку “Save”.
4. Выбрать имя и местоположение файла.
5. Нажать кнопку “Preview”.

*Ожидаемый результат*

Выбирается и открывается нужный файл. При сохранении соблюдаются введенные имя и местоположение. При просмотре открывается web-браузер по умолчанию.

#### **6.9 Тестовый сценарий стабильности(сопровождаемость)**

*Идентификатор*

Стабильность(сопровождаемость).

*Назначение*

Проверка отсутствия случайных и непреднамеренных ошибок.

*Сценарий*

1. Пройти тестовые сценарии 6.1 – 6.8.

*Ожидаемый результат*

Отсутствие случайных и непреднамеренных ошибок во время тестирования.

#### **6.10 Тестовый сценарий сосуществования**

*Идентификатор*

Сосуществование.

*Назначение*

Проверка отсутствия конфликтов приложения с другими процессами или программами.

*Сценарий*

1. Запустить приложение.
2. Использовать сторонние приложения.

*Ожидаемый результат*

Приложение работает без сбоев и не является причиной ошибок других программ.

### **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.