**Лабораторная работа**

**Создание верификационных и валидационных испытаний для проекта веб-сайта**

**Цель работы:** разработать испытания для контроля качества проекта и функционального тестирования программы.

**Отчет по лабораторной работе:**

* титульный лист
* ссылка на проект (\*при наличии) и изображение макета тестируемого проекта
* таблица требований, метрик, оценочных элементов по качеству (за основу стоит взять оформление в практической работе “Определение показателей качества”)
* сценарий для популярной функции
* \*результаты прохождения сценария
* \*код теста проверки сценария
* таблица анализа функциональных требований
* спецификация для тест дизайна
* набор позитивных и негативных тестовых сценариев
* матрица соответствия функциональных требований
* список ТК, скриншот двух тест-кейсов из Test-Rail
* вывод с процентом тестового покрытия
* ПРИЛОЖИТЬ К ОТЧЕТУ Приложение и Отчет (выгруженный) о тестовом прогоне

**План работы** предлагается следующий:

1. Сформулировать **требования к оценке качества проекта** (сайта) - к каждой [характеристике](https://drive.google.com/file/d/1pDVZRM9xaakabWXfTR4pP6l3FNzsDYiU/view?usp=drive_link)  по [ISO 9126](https://drive.google.com/file/d/1-FWv26UaJbca0dEh0WcpG2hQiViu32Rh/view?usp=drive_link).

1.1 Создать оценочные элементы для проверки качества. Учесть не менее одного оценочного элемента к каждой характеристике. Предусмотреть оценочный элемент для прохода сценария работы с самой популярной функцией (см.принцип простоты в [Лабораторной GUI](https://docs.google.com/document/d/1ACrPLcweX-ieHOSh_rriDYDO_34D_EuX/edit)).

1.2 \*Записать сценарий для проверки популярной функции в программе Selenium IDE.

1.3 \*Сценарий запустить, проверить корректность выполнения.

1.4 \*Скриншот с результатом прохождения и код теста поместить в отчет с соответствующими подписями.

1. Сформулировать **требования к функционалу одной заполняемой формы** (если такого макета не было - перед этим создать).
2. Проанализировать требования к форме и определить набор тестовых данных на основании EP, BVA, EG (составление спецификации требований).
3. Разработать шаблон теста на основании CE (в отчет можно не включать).
4. Написать тест-кейсы на основании первоначальных требований, тестовых данных и шагов теста (не меньше 3х позитивных, 3х негативных).
5. Разработать и заполнить **матрицу соответствия требований** (Traceability matrix), на основании сделанных тест-кейсов.

*Так как проект находится в данный момент на этапе Test Design, а не Test Execution, матрица будет монохромная: “Требование проверяется/требование не проверяется”.*

1. Рассчитать покрытие требований к функционалу тест-кейсами.
2. Зарегистрироваться на сайте TestRail или Теst IT. Создать проект, перенести в него написанные тест-кейсы.
3. Заполнить тестовое окружение в [Приложении](https://docs.google.com/document/d/1Zy7EvR9mKi7TCyAAsZEwv_Dy55X0AakR/edit?usp=drive_link&ouid=108020577343891577171&rtpof=true&sd=true) с помощью Dxdiag.
4. В отчет поместить скриншот позитивного и негативного ТК из TestRail и журнал (например, скриншот) со списком созданных тест-кейсов. Провести тестовый запуск, отметить различные результаты тестирования для каждого теста. Выгрузить отчет о тестовом запуске в графическом формате.
5. Написать вывод по проделанной работе, включив туда расчет тестового покрытия и результаты прохождения испытаний

**Теоретическая часть**

**Тестовый случай (Test Case)**

Под тест кейсом понимается структура вида:

Action > Expected Result > Test Result

Пример:

| Action | Expected Result | Test Result |
| --- | --- | --- |
|  |  | (passed/failed/blocked) |
| Open page "login" | Login page is opened | Passed |

**Пример тест кейса:**

do A1, verify B1

do A2, verify B2

do A3, verify B3

В приведенном примере конечная проверка - В3. Это значит, что именно она является ключевой. Значит, A1 и А2 - это действия приводящие систему в тестопригодное состояние. А В1 и В2 - условия того, что система находится в состоянии пригодном для тестирования. Таким образом имеем:

| Action | Expected Result | Test Result  (passed/failed/blocked) |
| --- | --- | --- |
| **PreConditions** | | |
| do A1 | verify B1 |  |
| do A2 | verify B2 |  |
| **Test Case Description** | | |
| do A3 | verify B3 |  |
| **PostConditions** | | |

PostConditions в данном примере не были описаны, но по логике вещей надо выполнить шаги, которые бы вернули систему в первоначальное состояние. (например, удалили созданную запись, или отменили бы изменения сделанные в документе).

Теперь ответим на вопрос: "Почему данное разбиение удобно использовать?"

Ответ: конечная проверка одна, т.е. в случае если тест провален (test failed) будет сразу ясно из-за чего. Т.к. если провальными окажутся проверки В1 и/или В2, то тест кейс будет заблокирован (test blocked), из-за того, что функцию невозможно привести в тестопригодное состояние (состояние пригодное для проведения тестирования), но это не значит, что тестируемая функция не работает.

| Action | Expected Result | Test Result (passed/failed/blocked) |
| --- | --- | --- |
| **PreConditions** | | |
| do A1 | verify B1 | passed |
| do A2 | verify B2 | failed |
| **Test Case Description** | | |
| do A3 | verify B3 | blocked |
| **PostConditions** | | |

**Детализация описания тест кейсов (Test Case Specification)**

Уровень детализации тест кейсов должен быть таков, чтобы обеспечивать разумное соотношение времени прохождения к тестовому покрытию. Т.е. до тех пор, пока покрытие тестами определенного функционала не меняется, можно уменьшать детализацию тест кейсов.

*Пример тест кейса 1:*

Проверка отображения страницы

| Действие | Ожидаемый результат | Результат теста |
| --- | --- | --- |
| Открыть страницу "Вход в систему" | - Окно "Вход в систему" открыто  - Название окна - Вход в систему  - Логотип компании отображается в правом верхнем углу  - На форме 2 поля - Имя и Пароль  - Кнопка Вход доступна  - Ссылка "забыл пароль" - доступна | ... |

*Пример тест кейса 2:*

| **Название**: Проверка отображения страницы |
| --- |
| **Действие**: Открыть страницу "Вход в систему" |
| **Проверка**: Проверьте, что отображаемая страница соответствует странице на картинке 1 (и прилагаем изображение страницы "Вход в систему") |

В примере 1 и 2 покрытие будет одинаковым, но вот время, которое потребуется для прохождения, будет разным. Мне кажется, что второй пример будет даже нагляднее.

**Практическое применение техник тест дизайна при разработке тест кейсов**

Вашему вниманию предложен последовательный подход к разработке тестовых случаев (тест кейсов), использующий самые простейшие техники тест дизайна:

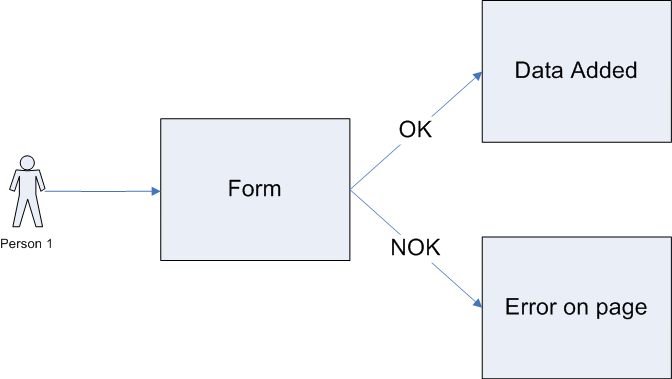
* Эквивалентное Разделение (Equivalence Partitioning), далее в тексте - EP
* Анализ Граничных Значений (Boundary Value Analysis), далее в тексте - BVA
* Предугадывание ошибки (Error Guessing), далее в тексте - EG
* Причина / Следствие (Cause/Effect), далее в тексте - CE

Далее на примере, рассмотрим предложенный подход.

*Пример*: Протестировать функциональность формы приема заявок, требования к которой представлены в следующей таблице:

| Элемент | Тип элемента | Требования |
| --- | --- | --- |
| Тип обращения | combobox | Набор данных:   1. Консультация 2. Проведение тестирования 3. Размещение рекламы 4. Ошибка на сайте   \* - на процесс выполнения операции приема заявок не влияет. |
| Контактное лицо | editbox | 1. Обязательное для заполнения  2. Максимально 25 символов  3. Использование цифр и спец символов не допускается |
| Контактный телефон | editbox | 1. Обязательное для заполнения 2. Допустимые символы "+" и цифры 3. "+" можно использовать только в начале номера 4. Допустимые форматы:  * начинается с плюса - 11-15 цифр * +31612361264 * +375291438884 * без плюса - 5-10 цифр, например: * 0613261264 * 2925167 |
| Сообщение | text area | 1. Обязательное для заполнения  2. Максимальная длина 1024 символа |
| Отправить | button | Состояние:  1. По умолчанию - не активна (Disabled)  2. После заполнения обязательных полей становится активна (Enabled)  Действия после нажатия  1. Если введенные данные корректны - отправка сообщения  2. Если введенные данные НЕ корректны - валидационное сообщение |

Вариант использования (иногда его может и не быть):



**1. Анализ требований**

Читаем, анализируем требования и выделяем для себя следующие нюансы:

* какие из полей обязательные для заполнения?
* имеют ли поля ограничения по длине или по размерности (границы)?
* какие из полей имеют специальные форматы?

**2. Определение набора тестовых данных**

Отталкиваясь от требований к полям, используя техники тест дизайна начинаем определение набора тестовых данных:

* в зависимости от того обязательное поле или нет, определим, какие поля необходимо проверить на пустое значение, так как оно может вызывать ошибку (В результирующей таблице оранжевый цвет)
* т.к. исчерпывающее тестирование не представляется возможным из-за огромного числа всевозможных комбинаций значений, в первую очередь необходимо определить минимальный набор данных. Это можно сделать, используя такие техники, как EP и BVA. (В результирующей таблице голубой цвет)
* На форме присутствует поле, имеющее составной тип (цифры используются совместно с символами), обладает специальным форматом данных и поэтому выделение тестовых данных для него - это достаточно трудоемкая задача. В пределах данной работы ограничимся только простой проверкой форматов и основных требований, описанных в форме приема заявок.
* По завершению генерации данных используя стандартные техники, можно добавить некоторое количество значений на основании личного опыта (техника EG) - это будет использование спец. символов, очень длинных строк, разных форматов данных, регистров в строках (Upper, Lowwer, Mixed cases), отрицательные и нулевые значения, кейворды Null - NaN - Infinity и т.д. Сюда можно включить все, что вы полагаете может вывести приложение из строя (в результирующей таблице фиолетовый цвет)

**2.1 Выбор тестовых данных для каждого отдельно взятого поля**

* Поле **Тип обращения**. Так как все данные входят в 1 класс эквивалентности, то есть не изменяют сам процесс выполнения приема заявки, берем любую (1-ю) позицию в листе с ожидаемым результатом ОК. Но т.к. реализовано поле как лист, имеет также смысл рассмотреть и граничные условия (техника BVA), т.е. берем первый и последний элементы. Итого: 1-я и последняя позиции в листе. Ожидаемый результат при использовании - ОК.
* Поле **Контактное лицо**. Это обязательное поле размером от 1 до 25 символов (включая границы). Проверка на обязательность добавляет к тестовым данным пустое значение. Проведем анализ граничных условий (BVA), получим набор: 0, 1, 2, 24, 25 и 26 символов. Пустое значение (0 символов) уже было добавлено при анализе обязательности поля для ввода, поэтому при BVA мы не будем добавлять его еще раз. (если его добавить второй раз, произойдет дублирование тестовых данных, которое не приведет к нахождению новых дефектов, а значит повторное добавление в домен не имеет смысла). В связи с тем, что значения 2 и 24 символа являются, с нашей точки зрения, не критичными, их можно не добавлять. В итоге получаем, что минимальный набор данных для тестирования поля - это строки 1 и 25 - ОК, и 0 (пустое значение), 26 символов - NOK.
* поле **Контактный телефон** состоит из нескольких частей: код страны, код оператора, номер телефон (который может быть составной и разделенный дефисами). Для определения правильного набора тестовых данных необходимо рассматривать каждую составную часть по отдельности. Применяя BVA и EP, получим:
* для номеров с плюсом:
* По BVA получим номера с 10, 11, 12 и 14, 15, 16 цифрами, где 10 и 16 - NOK, а 11, 12, 14, 15 - OK
* Рассматривая полученные данные с позиции EP выделим, что 11, 12, 14, 15 входят в один класс эквивалентности. Поэтому при тестировании мы можем использовать любое из них, но так как 11 и 15 - это границы интервала, то на наш взгляд их пропускать нельзя. Следовательно мы можем уменьшить набор значений до двух, исключив 12 и 14, а оставив 11 и 15 для проверки граничных условий.
* Итого имеем:
* 11 и 15 цифр - OK, (+12345678901, +123456789012345)
* 10 и 16 цифр - NOK; (+1234567890, +1234567890123456)
* для номеров без плюса:
* По BVA получим номера с 4, 5, 6 и 9, 10, 11 цифрами.
* Действуя аналогично примеру для номеров телефонов с плюсом, исключим значения 6 и 9, оставив 5 и 10.
* Итого имеем:
* 5 и 10 цифр - OK, (12345, 1234567890)
* 4 и 11 цифр - NOK; (1234, 12345678901)
* поле Сообщение. подбор данных проводим по аналогии с полем Контактное лицо. На выходе получаем значения: строки 1 и 1024 - ОК, и 1025 символов - NOK.

**Результирующая таблица данных, для использования при последующем составлении тест кейсов**

| Поле | OK/NOK | Значение | Комментарий |
| --- | --- | --- | --- |
| Тип обращения | OK  NOK | Консультация  Ошибка на сайте | первый в списке  последний в списке |
| Контактное лицо | OK  NOK | Йцукенгшщзйцукенгшщзйцуке  a  ЙЦУКЕНГШЩЗФЫВАПРОЛДЖЯЧСМИ  ЙЦУКЕНГШЩЗфывапролджЯЧСМИ  Йцукенгшщзйцукенгшщзйцукей  @#$%^&;.?,>|\/№"!()\_{}[<  1234567890123456789012345  adsadasdasdas dasdasd asasdsads(...)sas | 25 символов нижний регистр  1 символ  25 символов ВЕРХНИЙ регистр  25 символов СМеШаННыЙ регистр  пустое значение  длина больше максимальной(26 символов  спец. символы (ASCII)  только цифры  очень длинная строка (1Mb) |
| Контактный телефон | OK  NOK | +12345678901  +123456789012345  12345  1234567890  +1234567890  +1234567890123456  1234  12345678901  +YYYXXXyyyxxzz    Yyyxxxxzz  +###-$$$-%^-&^-&!  1232312323123213231232(...)99 | с плюсом - минимальная длина  с плюсом - максимальная длина  без плюса - минимальная длина  без плюса - максимальная длина  пустое значение  с плюсом - < минимальной длины  с плюсом - > максимальной длины  без плюса - < минимальной длины  без плюса - > максимальной длины  с плюсом - буквы вместо цифр  без плюса - буквы вместо цифр  спец. символы (ASCII)  очень длинная строка (1Mb) |
| Сообщение | OK  NOK | йццуйцуйц(...)йцу  йццуйцуйц(...)йцуц  adsadasdasdas dasdasd asasdsads(...)sas  @##$$$%^&^& | максимальная длина (1024 символа)  пустое значение  длина больше максимальной (1025 символов)  очень длинная строка (1Mb)  только спец. символы (ASCII) |

**3. Разрабатываем шаблон теста**

На основании **техники CE** и, по возможности, имеющихся вариантов использования (Use case) создадим шаблон планируемого теста. Данный документ будет представлять собой шаги и ожидаемые результаты теста, но без конкретных данных, которые подставляются на следующем этапе разработки тест кейсов. Шаблон откорректировать, с учетом знаний о правилах заполнения тест-кейсов:

Пример шаблона тест кейса

| Действие | Ожидаемый результат |
| --- | --- |
| 1. Открываем форму отправки сообщения | * Форма открыта * Все поля по умолчанию пусты * Обязательные поля помечены - \* * Кнопка "Отправить" не активна |
| 2. Заполняем поля формы:   * Тип обращения * Контактное лицо * Контактный телефон * Сообщение | * Поля заполнены * Кнопка "Отправить" - активна (Enabled) |
| 3. Нажимаем кнопку "Отправить" | * Если введенные данные корректны - * Сообщение "Заявка отправлена" выведено на экран. * Новая заявка появилась в списке на странице "Заявки". * Если введенные данные НЕ корректны; * Валидационное сообщение со всеми ошибками выведено на экран. * Заявка НЕ появилась в списке на странице "Заявки". |

**4. Написание тест кейсов на основании первоначальных требований, тестовых данных и шаблона теста**

Пример **позитивного** тест кейса (все поля OK) - форму откорректировать, с учетом знаний о правилах заполнения тест-кейсов:

| Действие | Ожидаемый результат |
| --- | --- |
| 1. Открываем форму отправки сообщения | * Форма открыта * Все поля по умолчанию пусты * Обязательные поля помечены - \* * Кнопка "Отправить" не активна |
| 2. Заполняем поля формы:   * Тип обращения = Консультация * Контактное лицо = йцукенгшщзйцукенгшщзйцуке * Контактный телефон = +7-916-111-11-11 * Сообщение | * Поля заполнены * Кнопка "Отправить" - активна (Enabled) |
| 3. Нажимаем кнопку "Отправить" | * Сообщение "Заявка отправлена"выведено на экран. * Новая заявка появилась в списке на странице "Заявки". |

Пример негативного тест кейса (поле Контактное лицо - NOK) - форму откорректировать, с учетом знаний о правилах заполнения тест-кейсов:

| Действие | Ожидаемый результат |
| --- | --- |
| 1. Открываем форму отправки сообщения | * Форма открыта * Все поля по умолчанию пусты * Обязательные поля помечены - \* * Кнопка "Отправить" не активна |
| 2. Заполняем поля формы:   * Тип обращения = Консультация * Контактное лицо = @#$%^&;.?,>|\/№"!()\_{}[< * Контактный телефон = (916)333-33-33 * Сообщение = йццуйцуйц(...)йцу - 1024 символа | * Поля заполнены * Кнопка "Отправить" - активна (Enabled) |
| 3. Нажимаем кнопку "Отправить" | * Валидационное сообщение со всеми ошибками выведено на экран: * "В поле "Контактное лицо" запрещено использование цифр и спец. символов." * Заявка НЕ появилась в списке на странице "Заявки". |