**Практическая работа**

**ТЕМА: РАЗРАБОТКА РАБОЧЕЙ ТЕСТОВОЙ ДОКУМЕНТАЦИИ**

**1.Тестовая документация**

**Тестовая документация** (тестовая модель) - документ, отражающий модули тестируемого продукта и их текущее состояние (дефекты), а также исторические данные по предыдущим раундам тестирования.

Уровень детализации тестовой модели приложения может меняться в зависимости от сложности модуля (чем сложнее модуль, тем больше деталей, вплоть до тест-кейсов). То есть в одной тестовой модели могут быть совмещены несколько уровней детализации"

Тестовая модель не статична на протяжении работы над проектом. Она дополняется по мере выполнения тестов (например, могут добавляться новые проверки), а также по мере добавления функциональности тестовая модель расширяется.

Таким образом, тестовая модель должна отображать текущее состояние (поведение) тестируемого приложения на текущий момент.

**Модель, отражающая поведение нашего приложения**

**Примеры.**

**1.Acceptance Sheet** (иногда также называется Check list) - документ, который содержит подробный перечень всех модулей и функций приложения, а также результаты всех тестов данных функций. Как правило, Acceptance Sheet содержит статистику по наиболее важным показателям каждой сборки (англ. - build), определяющим ее качество.

**Чек-лист (check list)** - это документ, описывающий, что должно быть протестировано. При этом чек-лист может быть абсолютно разного уровня детализации. Насколько детальным будет чек-лист, зависит от требований к отчетности, уровня знания продукта сотрудниками и сложности продукта. Как правило, чек-лист содержит только действия (шаги) без ожидаемого результата. Чек-лист менее формализован, чем тестовый сценарий. Его уместно использовать тогда, когда тестовые сценарии будут избыточны. Также чек-лист ассоциируется с гибкими подходами в тестировании.

**Чек-листы - один из фундаментальных инструментов тестирования.** Они позволяют не забывать о важных тестах, фиксировать результаты своей работы, отслеживать статистику о статусе программного продукта и делить задачи по уровню детализации.

Иногда чек-листами называют подробные инструкции о тестируемом продукте, содержащие последовательность действий, множество деталей и т.д. Это не так. Главный принцип чек-листов заключается в том, что каждый тестировщик по-своему проходит их, расширяя тестовый набор своей экспертизой. Какие **преимущества чек-листов по сравнению с тест-кейсами**:

1) расширение тестового покрытия за счет вариативности при выполнении тестов;

2) сокращение затрат на содержание и поддержку тестов;

3) отсутствие рутины, которую так не любят квалифицированные тестировщики;

4) возможность проходить и комбинировать тесты по-разному в зависимости от предпочтений сотрудников.

При этом **чек-листы сохраняют множество плюсов**, за которые так популярны детальные тест-кейсы:

1) статистика: кто, когда, что проходил (с детализацией по сборке продукта и окружению, на котором проводилось тестирование);

2) памятка, которая помогает не забыть важные тесты;

3) возможность оценить состояние продукта, его готовность к выпуску.

**Минусы чек-листов:**

1) начинающие тестировщики не всегда эффективно проводят тесты без достаточно подробной документации; .

2) чек-листы невозможно использовать для обучения начинающих сотрудников, так как в них недостаточно подробной информации;

3) заказчику или руководству может быть недостаточно того уровня детализации, который предлагают чек-листы.

Итого: если в компании высокая текучесть кадров, низкоквалифицированные сотрудники или этого требует руководство, выбора нет и придется создавать и поддерживать подробные детальные тест-кейсы. Но если в команде квалифицированные сотрудники, то чек­листы значительно удобнее и помогут получить максимум пользы от тестовой документации.

**Содержимое чек-листа:**

1)список проверок (с требуемой степенью детализации);

2)статус проверок:

•сборка, на которой проводилось тестирование;

•тестовое окружение (если применимо);

•тестировщик;

3)результат проверки.

Инструменты ведения чек-листов:

1)таблицы Excel/OpenOffice для самостоятельной работы;

2)таблицы GoogleDocs для распределения в команде;

3)онлайн-сервис для ведения чек-листов Sitechco для командной работы и контроля версий.

2.**Test Survey** - документ, который содержит подробный перечень всех проверок для отдельных функциональностей, а также ожидаемый результат проверок.

3.**Test Cases** - набор входных значений, предусловий выполнения, ожидаемых результатов и постусловий выполнения, разработанный для определенной цели или тестового условия, таких как выполнение определенного пути программы, или же для проверки соответствия определенному требованию.

Заголовок: проверка возможности входа в систему с валидными данными.

Шаги:

1)открыть браузер;

2)зайти на главную страницу веб-приложения;

3)в форме логина ввести:

а) usemame: Кirill;

6) password: 123456;

4)нажать «Enter».

Ожидаемый результат: пользователю открывается страница личного кабинета.

**2.Сравнительный анализ разных видов документаций (табл. 1)**

Таблица 1

**Сравнительный анализ тестовых документаций**

|  |  |  |  |
| --- | --- | --- | --- |
| Тип докvментации | Что описываем? | Когда используем? | Пример |
| Acceptance Sheet | Перечень функцио- нальностей, которые нужно протестиро- вать | На кратко- срочных про- ектах  с простой биз-  нес-логикой | Форма ,входа на сайт |
| Test Survey | Конкретные про- верки в рамках от- дельных функцио- нальностей. Может содержать ожида- емый результат | На проектах средней про- должительно- сти с понят- ной бизнес- логикой | Форма входа на сайт: ввод корректных данных;  ввод некорректного имени;  ввод некорректного пароля и т.д. |
| Test Cases | Порядок вьшолне- ния и результат тестов | На крупных долгосрочных проектах,тре- бующих от те- стировщиков глубоких зна- ний предмет- ной области | Форма входа на сайт: откройте форму входа;  введите имя польза- вателя testl;  введите пароль testl; нажмите кнопку  «ВОЙТИ».  Ожидаемый резуль- тат: пользователь  попадает на домаш нюю страницу |

**3.Процесс создания чек-листа для тестирования сайта**

Необходимо протестировать ключевые страницы сайта: главную страницу; различные страницы регистрации и ввода данных; внутренние страницы с различным содержанием; поиск; прочие страницы с нестандартным функционалом. Также необходимо изучить ключевую аудиторию. Это необходимо для того, чтобы знать, с каких устройств будут заходить на сайт, какие операционные системы и браузеры будут у большинства пользователей, какая скорость соединения ожидается (может, ЗG и приоритет с мобильным трафиком), на­ сколько технически грамотной будет аудитория.

При создании чек-листа, например, в Ехсеl-таблице необходимо будет добавлять в таблицу дефекты, проблемы и задачи в упорядоченный список и помечать приоритет, критичность и пр. Также не­ обходимо будет назначать выполнение для членов команды или себе с датой завершения, комментировать ошибки, проблемы и задачи, прикреплять скриншоты, видео и документы, связанные с ошибками, отмечать решенные проблемы и делать отчеты с найденными ошибками.

Также необходимо будет составить список, где перечислить все операционные системы и устройства, на которых будет тестироваться сайт. Пример представлен ниже.

**Объем работы:**

Полная поддержка:

• Windows 8: IElO+, Chrome (latest), Firefox (latest), Safari (latest);

• Мае OS Х Mavericks: Chrome (latest), Firefox (latest), Safari (latest);

• iPhone 4/5, iPad 2, Asus VivoTab Smart.

Частичная поддержка:

• Windows 7: IE9+, Chrome (latest), Firefox (latest), Safari (latest);

• Windows ХР: IE8, Chrome (latest), Firefox (latest), Safari (latest).

Полная поддержка означает, что весь контент читабелен, все функции работают, отклонения от утвержденного дизайна минимальны.

Частичная поддержка обозначает, что весь контент читабелен, навигация работает, прочая функциональность не мешает нормальной работе, при ухудшении дизайна содержание не страдает.

Далее нужно записать в плане тестирования все необходимые проверки, которые будут проводиться. Пример описан далее.

На главной странице обычно находится много разных функциональных элементов: меню, поиск, слайдеры и пр. И все их стоит занести в план тестирования.

**Поиск:** нажатие или касание на форму поиска, заполнение ее и нажатие на иконку поиска отправляют данные в форму.

**Навигация:** наведение курсора на пункт меню подсвечивается белым цветом; нажатие на пункт меню переводит на соответствующую пункту страницу.

**Карусель:** нажатие на стрелки влево/вправо переводит на следующий/предыдущий слайд; нажатие на элементы пагинации (в веб­ дизайне и дизайне программного обеспечения под пагинацией понимают постраничный вывод информации, т.е. показ ограниченной части информации на одной (веб)-странице; пагинация повсеместно используется в веб-приложениях для разбиения большого массива данных на странице и включает в себя навигационный блок для перехода на другие страницы) переводит на соответствующий элементу слайдер; проверка слайдов на мобильных устройствах.

**Тексты:** нажатие на кнопку «Далее» переводит на новую страницу с соответствующим текстом.

Таким же образом описываются все остальные элементы функциональности на главной странице.

**Как оценить степень регрессии дизайна от браузера к браузеру?** Существуют тесты для выявления визуальных различий между двумя версиями, которые подчеркивают потенциальные проблемы. Например, такие ресурсы, как Wraith, PhantomCSS, Huxley, CSS Regression Testing with ResemЫe.js, CSSte.st.

**4.Разработка чек-листа для мобильного приложения.**

**Планирование тестовых активностей**

Рекомендуется до начала тестирования ознакомиться с аналогичными приложениями. Это поможет получить основное представление о том, как должны функционировать приложения данного типа.

Создание документации значительно улучшает качество продукта за счет уточнения деталей и списка проверок. После завершения тестирования наличие тестовой документации позволяет оценить, насколько успешно были проведены все этапы тестирования.

Вне зависимости от типа документации она должна содержать все необходимые проверки для всех сценариев использования приложения. Но также не стоит забывать о проверках, диктуемых особенностями мобильных устройств.

При тестировании приложений рекомендуется проверять следующие нижеперечисленные объекты:

1) установка, запуск, удаление, обновление приложения;

2) изменение ориентации экрана;

3) различные виды соединений и переключения между ними;

4) прерывание работы приложения;

5) графический интерфейс пользователя и навигация;

6) уведомления в приложении;

7) работа с использованием кнопок управления мобильным устройством;

8) контроль данных: сохранение, использование, удаление;

9) работа приложения в режиме энергосбережения. Необходимо пояснить, для чего необходимы эти проверки. Например, при тестировании приложения с использованием различных типов интернет-соединения (2G/ЗG/4G/Wi-Fi) можно обнаружить неверную обработку потери соединения и неполную загрузку форм при медленном соединении.

Другой пример - проверка реакции приложения на различные прерывания. Иногда это приводит к аварийному завершению работы и потере введенных данных.

**5.Пример чек-листа (табл. 2)**

Таблица 2

**Пример чек-листа**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| -  Проверка | Результат | | | |
| build 4 | build 3 | build 2 | build 1 |
| Операции с файлами | ok | ok | ok | ok |
| Создание файла | ok | ok | ok | ok |
| Открытие файла | ok | ok | ok | ok |
| Сохранение документа | ok | ok | ok | ok |
| Печать | ok | ok | ok | ok |
| Редактирование файлов | bugs | bugs | bugs | bugs |
| Отмена | ok | ok | ok | ok |
| Копирование | ok | ok | ok | ok |
| Вырезание | bug #146 | bug #146 | ok | ·ok |
| Вставка | ok | ok | ok | ok |
| Удаление | ok | ok | ok | ok |
| Поиск | bug #133 | bug #133 | ok | bug #123 |
| Поиск с заменой | bug #126 | ok | bug #126 | bug #126 |
| Вставка даты | ok | ok | ok | ok |
| Форматирование | bugs | bugs | bugs | ok |
| Перенос строки | bug #129 | bug#l29 | bug #129 | ok |
| Изменение шрифта | bug#l58 | ok | ok | ok |
| Справка | ok | ok | ok | ok |

**6.Чек-лист.**

**Список базовых проверок wеЬ-приложения**

**Спеллинr** (англ. - spelling) - это правописание, орфография.

*Таблица 3.*

**Чек-лист веб-тестирования**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название элемента | | Functional Test | GUI Test | |
| Поле | | Обязательность ввода. Обработка только пробелов.  Использование пробелов  в тексте (перед текстом, после него и внутри; пробелы в на- чале **и** в конце строки должны отсекаться при сохранении). Минимально/максимально допустимое количество сим- волов (осуществлять проверку на ввод большого текста без пробелов не нужно).  Формат данных (исходя из его логического назначения и требований приложения).  Формат числовых данных (если допускаются): негатив- ные, дробные с точкой **и** запя- той, с точкой **и** запятой 123.123.123,00).  Ввод тегов и скриптов (про- верка должна осуществляться только *в* пользовательской части. Введенные теги должны отобразиться в том же виде,  в котором они были введены). | Название поля (спеллинг, соответствие с открытым мо- дулем или страницей).  Выравнивание названий по- лей (выравнивание по левому краю или правому краю (в за- висимости от требований приложения, отступы, иден- тичность расстояний'между названием и полем).  Корректное расположение текста внутри текста, длин- ный текст не выходит за гра- ницы поля при вводе.  Унификация дизайна (цвет, шрифт, размер (высота/ши- рина), выравнивание полей). Расположение вводимого текста внутри поля (унифика- ция, выравнивание по ниж- нему краю (для textarea -  по верхнему), если иное не определено специфич-  ными требованиями прило- жения) | |
|  | Использование специальных символов (введенные сим- волы должны отобразиться  в том же виде, в котором они были введены, если только ввод спец. символов не запре- щен требованиями приложе- ния).  Возможность редактирования введенных значений.  Корректное распределение текста по строкам (переход на новую строку автомати- чески).  Уникальные данные (напри- мер, уникальность логина,  e-mail пользователя). Автоматическая постановка курсора в первое поле для ввода при открытии формы | |  |
| Кнопка | Отсутствие вызова одного  и того же действия повторно при нажатии на кнопку не- сколько раз.  Недоступные кнопки  не скрыты, а заблокированы (если кнопка недоступна  в данный момент, то она  не должна пропадать с формы, пользователь должен знать о ее существовании; но кнопка должна быть недоступна - не- кликабельна и отображаться более светлой).  Нажатие на пространство между близко расположен- ньrми кнопками не должно приводить к действию.  Нажатие на пространство возле кнопки не должно при-  водить к действию. | | Название кнопки (спеллинг, соответствии с действием). Эффект «нажатия» (вид кнопки должен изменяться при нажатии, если это не про- тиворечит возможностям бра- узера).  Название всплывающей под- сказки (соответствие с назва- нием кнопки, спеллинг).  Унификация дизайна (цвет, шрифт, размер (высота/ши- рина), цвет подсветки, вырав нивание).  Проверка осуществляется как для кнопки как элемента, так и для названия кнопки |
|  | Должно работать нажатие на всю площадь кнопки,  а не только по ее названию | |  |
| Радио баттон | Функциональность (включе- ние/выключение).  Не может быть меньше 2 ра- диокнопок.  По умолчанию одна радио- кнопка должна быть включена. Не может быть включено более 1 радиокнопки.  При переходе на следующую страницу и возвращении назад выбранная радио кнопка  не должна сбрасываться | | Унификация дизайна для всего приложения.  Выравнивание расположения радиобаттона с соответству- ющим названием.  Выравнивание расположений радиобаттонов (по краю) |
| Чек-бокс | Функциональность (включе- ние/выключение).  Обязательность выбора хотя бы одного чек-бокса.  Наличие дополнительного чек- бокса, выставляющего/снимающего все чек-боксы при наличии больше 10 чек-боксов. При переходе на следующую страницу и возвращении назад выбранный чек-бокс не дол- жен сбрасываться.  Отсутствие функции сорти-  ровки таблицы по полю, кото- рое содержит только чек-боксы | | Унификация дизайна для всего приложения.  Выравнивание расположения чек-бокса с соответствующим названием.  Корректность отображения недоступного чек-бокса |
| Поле со списком/ выпада- ющий спи- сок | Сортировка по алфавиту или по смыслу.  В случае если значения выходят за границы списка и нет возможности увеличе  ния размера списка, то необ- ходимо отображение хинтов (всплывающих подсказок).  Выбор пункта списка  по нажатию соответству-  ющей первой буквы на кла- виатуре. | | Спеллинг значений. Подсветка при выборе каж- дого из значений, при выборе нескольких значений одно- временно.  Унификация дизайна (цвет, шрифт, размер (высота/ши- рина), цвет подсветки, вырав- нивание). Проверка осуще- ствляется как для поля как  элемента, так и для значений и их названий |
| Название элемента | Functional Test | | GUI Test |
|  | Возможность выбора несколь ких значений для поля  со списком.  Возможность введения значе ний вручную (если это позво- ляет приложение).  Возможность выбора значе- ния из списка указателем  мыши либо стрелками с кла- виатvры | |  |
| Меню | Осуществление соответствущего перехода при выборе пункта меню.  Визуальное различие в момент работы на определенной вкладке (подсветка, подчеркивание) | | Подсветка таба при наведении курсора.  Изменение курсора при наве- дении.  Эффект «нажатия», если это не противоречит возможно- стям браузера.  Если работа в данный момент в выбранной вкладке,  то в меню она отличается ви- зуально, при этом является некликабельной.  Совпадение названий в слу- чае, если меню дублируется  в нескольких местах |
| Окно | Возможность изменения раз- мера окна браузера. Не- сколько способов сменить разрешение окна браузера при тестировании:  сменить разрешение самого экрана;  воспользоваться виртуальной машиной;  воспользоваться расширени- ями для браузеров.  Возможность изменения мас- штаба страницы | | Появление скролла при уменьшении (изменении) размера окна браузера.  Сохранение расположения элементов при уменьшении (изменении) окна браузера, при изменении масштаба. Если дефект воспроизводится только при смененном мае- штабе, то в описании дефекта одним из шагов можно указать «увеличить/уменьшить масштаб до ...%».  Соответствие названия окна в зависимости от назначения страницы (например, назва-  ние окна должно быть |
|  |  | | Proflle, если пользователь находится на странице про- филя).  Спеллинг, синтаксис назва- ний.  Унификация названий |
| Полосы прокрутки | Отсутствие полос прокрутки  в случае, если текст вмещается на странице без прокрутки.  Соответствующие изменения текста при использовании полос прокрутки.  Возможность изменения по- ложения полос прокрутки при помощи мыши, кнопок Page  up/down, Home/End | | Унификация видов и типов полос прокрутки на всех стра- ницах (если есть полоса про- крутки, используемая  по умолчанию, то она должна быть применена на всех иден- тичных формах) |
| Ссылка | Функционирование ссылки (должен осуществиться пере- ход на соответствующую стра ницу).  При наведении указателя мыши отображается подсказка (желательно).  Форматы ссылок и префиксов | | Унификация стилей (в соот- ветствии с дизайном сайта). Расположение ссылок (в со- ответствии с дизайном сайта). Например, расположение всех ссылок слева и справа  от элементов.  Названия (унификация, идентичность названий ссы- лок одинакового назначения, спеллинг, соответствие с от- крытым модулем или страни- цей, вместимость названия ссылки в отведенном блоке). Изменение вида курсора при наведении на ссылку.  Изменение вида ссылки при наведении курсора (подчер- кивание) |
| Таблица | При появлении нескольких страниц есть кнопки «Впе- ред», «Назад», «На первую»,  «На последнюю страницу»,  (пагинация). | | Унификация дизайна для всего приложения  (цвет, шрифт, размер (вы-  сота/ширина), выравнива- ние). |
|  | Проверка сортировок (и про- верка сортировки по умолча- нию).  Проверка фильтрации (если есть возможность).  Обновление значений таб- лицы после добавления/изме- нения/удаления данных.  Единичное/множественное выделение нескольких значе- ний | | Название (соответствие с те- кущим модулем, спеллинг). Выравнивание иконок сорти- ровки в названии колонок.  Выравнивание названий ко- лонок, значений внутри таб-  ЛИЦЫ.  Корректное отображение длинных названий (соответ- ствующие переходы на новые строки, сокращение названий (появление ..., либо сокраще- ние по слову).  Корректное отображение данных после использования сортировки (размеры колонок и столбцов фиксированы, текст не разбивает структуру  таблицы) |
| Поп-ап (всплыва- ющее окно) | Должно быть модальным. Корректное выделение фоно- вой страницы (если фон стра- ницы изменяется при появле- нии всплывающего окна,  то при изменении масштаба страницы фон должен заполнять всю страницу - размер измененного фона соответствует размеру страницы).  Фиксированное положение всплывающего окна (динамическое изменение положения) в случае использования полос прокрутки | | Спеллинг, синтаксис текста, расположенного на всплыва- ющем окне.  Отображение всплывающего окна по центру страницы, окна, формы.  Выравнивание текста, пред- ставленного на всплывающем окне.  Корректное расположение текста на всплывающем окне: текст должен быть в рамках всплывающего окна; длинное название должно распола- гаться на новой строке, если иное не определено специ- фичными требованиями при-  ложения |
| Календарь | При возможности ввода даты вручную необходимо проверить разные форматы. | | Унификация дизайна для всего приложения (цвет, шрифт, размер (высота/ши-  рина), выравнивание). |
|  | Проверки логичности ввода (даты в будущем, валидации возраста и т.д.) | | Отображение календаря ря- дом с полем.  Корректное выравнивание всех элементов и ссылок в ка- лендаре |
| Поле для загрузки файлов | Обязательность выбора фай- лов.  Форматы.  Ограничения на размер. | | Унификация дизайна для всего приложения (цвет,  шрифт, размер (высота/ши- рина), выравнивание). |
|  | При отсутствии изображений | | Выравнивание названий за- |
|  | должна быть соответствующая | | груженных файлов, самих |
|  | миниатюра, либо картинка совсем не должна отобра- | | миниатюр файлов |
|  | жаться. | |  |
|  | Контроль за размером (высота/ | |  |
|  | ширина), должно быть до- | |  |
|  | ступно изменение размера картинки (снимка, видеока- | |  |
|  | дра). С сохранением пропор- ций либо с исправлением | |  |
|  | оных.  Загрузка исполняемых файлов (ЕХЕ, **РНР,** JSP etc.). Пере- | |  |
|  | именованный ЕХЕ | |  |
| Сообщения | Пользователь должен быть | | Сообщение о том, что нет |
|  | информирован о действиях, | | соответствующих элементов |
|  | происходящих в системе, по- | | (в таблицах, при поиске, при |
|  | средством сообщений об | | переходе на страницы). |
|  | успешном завершении опера- | | Спеллинг, синтаксис сообще- |
|  | ции. | | ний. |
|  | На необратимые действия, | | Соответствие сообщений |
|  | такие как удаление, должны | | по смыслу в зависимости |
|  | быть подтверждающие сооб- | | от выполняемого действия. |
|  | щения. | | Соответствие названия требу- |
|  | Введенные в форму данные | | емого действия в сообщении |
|  | не должны сбрасываться | | об ошибке действию, которое |
|  | после появления сообщения | | пользователь должен выпол- |
|  | об ошибке | | нить. Например, если необхо- |
|  |  | | димо выбрать значение |
|  |  | | из списка, в сообщении |
|  |  | | об ошибке должно быть |
|  |  | | указано «Please select», а не «Please enter».  Унификация стилей (цвет, размер) для всего приложения.  Соответствие цветов типу сообщений (красный для со- общений об ошибках, зеленые цвета не противоречат специфичным требованиям приложения.  Соответствие названий полей в сообщениях об ошибках  и сообщениях об успешном завершении операции назва- ниям полей, форм, таблиц, кнопок и т.д.  Соответствие порядка выве- дения сообщений об ошиб- ках в соответствии с поряд- ком расположения полей,  в которых были найдены ошибки.  Поле, в котором содержится ошибка, должно (желательно) выделяться цветом.  Невалидное значение  не должно отображаться в со- общении об ошибке. Пример, как быть не должно:  [«Email2309234@@mail.ru](mailto:Email2309234@@mail.ru)  не соответствует допустимому формату».  Согласование числительного и связанного с ним существи- тельного должно соблю- даться. Например, «1 день»,  «2 дня», «5 дней»> |
| Общие проверки | Перед тестированием каждой новой сборки необходимо осуществить очистку кэша  и файлов куки. Для полной очистки кэша и файлов куки можно воспользоваться при- ложением Ccleaner.  404 Error (переход по некор- ректной ссылке должен вести на страницу с ошибкой 404, а не просто на страницу Page cannot Ье found. Страница  ошибки 404 должна быть реа лизована в общем дизайне тестируемого приложения). ТаЬ order (сверху вниз, слева направо). Поля, доступные для прочтения и недоступные для редактирования, должны пропускаться при переключе- нии кнопкой ТаЬ.  Значок логотипа должен быть ссылкой на главную страницу. Фокус на кнопке для исполнения действий (ввод данных нажатие Enter действие осуществилось).  Проверка Breadcrumbs («Хлеб- ные крошки» - элемент нави- гации, являющийся призна- ком удобства пользования сайтом в целом и перемеще- нием по его структуре).  Отображение flаsh-элементов при отключенном/неустанов- ленном в браузере flash-плeepe (пользователю должно быть предложено скачать и установить последнюю версию flash- плеера; на месте flаsh-объекта должно отображаться альтерна-  тивное изображение). | | Уменьшение/увеличение мае- штаба страницы (элементы должны соответственно пере- распределиться с сохране- нием пропорций).  Уменьшение/увеличение раз- мера окна браузера (элементы должны соответственно пере- распределиться с сохране нием пропорций, взаимора- сположение должно сохра- няться) |
|  | Проверка работоспособности отправки е-mаil-уведомлений (как администратору, так  и пользователю), если только отсутствие писем не является спецификой проекта.  Проверка работоспособности приложения при отключен- ном JavaScript. Основная функциональность и навига- ция должна работать.  Тестирование ссылок и прав доступа.  Всегда проверять взаимосвязь компонентов. Обращать внимание на то, как ведет себя один компонент при измене- нии/удалении другого. На- пример, при удалении катего- рии товара не должны уда-  ляться все товары в этой кате- гории | |  |

**7. Документирование результатов тестирования.**

**Создание понятных отчетов о тестировании**

Отчет о тестировании (test-report) - документ, содержащий ин­ формацию о выполненных тестовых действиях, результатах проведенной работы. Обычно он включает в себя таблицы, графики, списки, просто описывающую информацию в виде текста. Их пропорция и содержание определяют пользу и понятность отчета.

Важно понять, для кого, для чего и в каких условиях создается отчет и насколько это улучшит восприятие излагаемой информации. Надо помнить, что каждое действие преследует определенную цель. В случае отчета важно понять, для кого, для чего и в каких условиях это делается. Рассмотрим схему, отражающую различные виды от четности.



Для кого?

Метрики о ситуации

Бизнес-пользователи

Аналитические

разрезы

Вид компании

Аутсорсинг

Компания-разработчик

Временной отрезок

Итоговый

**Аналитические разрезы** - это и есть наш отчет. В нем мы даем анализ нашей работе и оценку тестируемому продукту. Вид компании, в идеальной ситуации, не должен влиять на качество и смысловую емкость отчетности.

Саму отчетность можно разделить на финальную и регулярную - дневную, недельную, месячную, версионную (для каждой версии продукта) и т.п. Различия заключаются в глубине временной выборки. Итак, перед написанием отчета сначала надо определиться, для кого он пишется.

В ходе проекта, информация должна консолидироваться по тому направлению, которое необходимо отразить. Можно выделить три группы целевых аудиторий.

1) **технические пользователи** - Test-manager. Для них приоритетно понимание хода тестирования, какие возникают проблемы, как они решаются, построение самого процесса тестирования, описание применяемых методов и технологий;

2) **Product Manager**, они же менеджеры продукта. Их фокус сконцентрирован на сроках выполнения, выжимках из результатов тестирования без излишних технических подробностей и на общей статистике (цифровые и сравнительные метрики);

3) **бизнес-пользователи.** Обычно это и есть те люди, которые принимают решения о завершении тестирования. Они же определяют качество проделанной работы. Для них в первую очередь важен конечный результат в максимально кратком и ясном формате (да/нет), наглядное представление информации (графики, диаграммы), экспертное мнение о возможности выпуска продукта в промышленную среду и т.п. без углубления в детали.

Отчеты могут делиться на два вида относительно времени.

1. **(Недельный, дневной, месячный)/ промежуточный.** В общем это практически тот же финальный отчет, но с измененными приоритетами фокуса и уменьшенной глубиной временной выборки. В нем обязательно должны содержаться две главные метрики:

• оценка степени готовности продукта;

• оценка проведенных работ по тестированию за время между отчетностями (прогресс).

Этот отчет должен показать, какова динамика вашей работы. Важно помнить, что **прогресс** - величина не постоянная, а динамическая, она определяется за счет сравнения состояния проекта на прошлой неделе и настоящей. Соответственно, прогресс - это совокупность метрик, позволяющих понять, в каком состоянии находится проект. Они создаются для каждого проекта индивидуально, основываясь на целях, которые ставятся для успешного проведения тестирования. Метрики ставятся при создании тест-кейсов, их прохождении (провален/пройден), обнаружении дефектов (критичность). Они позволяют доступно и достаточно быстро составить общую сравнительную картину по проекту. Есть еще один важный и часто используемый тип временного отчета - **версионный** (отчет по итерации). Он схож с итоговым отчетом. В нем описываются те задачи, которые были выполнены командой тестирования для конкретной версии продукта.

2. **Конечный /финальный.** В финальном отчете важно показать общий взгляд на проделанную работу (в контексте установленных метрик) и эволюцию продукта. Также надо дать исчерпывающую

информацию о статусе продукта в данный момент (количество оставшихся неисправленных ошибок, полностью ли протестирован продукт или требуется дополнительный цикл тестирования, оценка возможности выпуска продукта во «внешний мир» и т.д).

Какие приемы представления информации и данных используются в отчетности?

Для бизнес-пользователей зачастую используют представление информации в виде графиков. Они наглядно показывают, насколько продукт готов к выпуску в промышленную среду, на сколько процентов проект выполнен. Это может быть, к примеру, график пройденных тест-кейсов по модулям. Он наглядно покажет, какой объем работы в каждом модуле уже проделан, и поможет вычленить проблемы. Также очень полезным может быть график отношения обнаруженных и исправленных дефектов. В случае продуктивной работы программистов над исправлением дефектов и написанием качественного кода кривая критических ошибок с выходом нового релиза стремится книзу, при этом приоритет и важность ошибок тоже уменьшаются. Но если разработчиками или тестировщиками уделяется мало внимания существующим дефектам, то кривая закрытых багов растет медленнее, чем незакрытых. В идеальном случае кривая неза­ крытых багов (найденных, но неисправленных) должна сойтись с кривой исправленных. Другими словами, к финальному релизу необходимо, чтобы все дефекты были устранены. Если это не так, то руководство может принять решение о продлении разработки и тестирования с целью устранения всех дефектов. Также необходимо оформить сводную таблицу. График строится на основании этих данных. Пример таблицы, на основании которой был построен график пройденных тест-кейсов по модулям, представлен на рис. 2.

Может показаться, что отчеты разных типов сильно отличаются. Тем не менее в них есть схожие черты и данные, которые стоит указывать всегда:

|  |  |  |  |
| --- | --- | --- | --- |
|  | 121 | Прохождение | |
|  |  |
| Модуль | Общее количе- ство ТК по модулю | passed | not passed yet |
|  |  |  |  |
| 1 | 6 | 1 | 5 |
| 2 | 8 | 6 | 2 |
| 3 | 1 | о | 1 |
| 4 | 37 | 17 | 20 |
| 5 | 28 | 2 | 26 |
| 6 | 28 | 23 | 5 |
| 7 | 13 | 8 | 5 |

**Рис. 2.** **Сводная таблица для отчетности**

1) состав команды;

2) сроки выполнения, за которые составляется отчет;

3) описание процессов тестирования;

4) изменения тестовой модели, дополнение тест-кейсами;

5) процент пройденных тест-кейсов;

6) критичные и блокирующие проблемы и принятые меры по их устранению;

7) результаты регресса (плюс акцент на сохранившихся проблемах);

8) план на следующую итерацию/неделю/месяц.

**8.Пример отчета о результатах тестирования**

Для того чтобы заполнить некоторые части отчета, придется сделать допущения о текущем моменте развития проекта и сложившейся ситуации с качеством.

**Краткое описание.** За период 26-28 мая было выпущено четыре сборки, на последней из которых успешно прошло 100% тест-кейсов Smоkе-тестирования и 76% тест-кейсов тестирования критического пути. 98% требований высокой важности реализовано корректно. Метрики качества находятся в зеленой зоне, потому есть все основания рассчитывать на завершение проекта в срок (на текущий момент реальный прогресс в точности соответствует плану). На следующую итерацию (29 мая) запланировано выполнение оставшихся низкоприоритетных тест-кейсов.

**Команда тестировщиков.**

|  |  |  |
| --- | --- | --- |
| Имя. | Должность | Роль |
| Джо Блэк | Тестировщик | Ответственный за обеспечение каче- ства |
| Джим Уайт | Старший разработчик | Ответственный за парное тестиро- вание и аудит кода |

**Описание процесса тестирования.** Каждая из четырех выпущенных за подотчетный период сборок (3-6) была протестирована под ОС Windows 7 Ent х64 и ОС Linux Ubuntu 14 LTS х64 в среде исполнения РНР 5.6.0. Smоkе-тестирование выполнялось с использованием автоматизации на основе командных файлов (см. \ \PROJECTS\FC\ Testing\Aut\Scripts). Тестирование критического пути выполнялось вручную. Регрессионное тестирование показало высокую стабильность функциональности (обнаружен только один дефект с важностью «средняя»), а повторное тестирование показало ощутимый прирост качества (исправлено 83% обнаруженных ранее дефектов).

**Расписание.**

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Дата | Деятельность 1i1 Продолжи- тельность, ч | |
| Джо Блэк | 27.05.2015 | Разработка тест-кейсов | 2 |
| Джо Блэк | 27.05.2015 | Парное тестирование | 2 |
| Джо Блэк | 27.05.2015 | Автоматизация Smоkе-тестирования | 1 |
| Джо Блэк | 27.05.2015 | Написание отчетов о дефектах | 2 |
| Джим Уайт | 27.05.2015 | Аудит кода | 1 |
| Джим Уайт | 27.05.2015 | Парное тестирование | 2 |
| Джо Блэк | 28.05.2015 | Разработка тест-кейсов | 3 |
| Джо Блэк | 28.05.2015 | Парное тестирование | 1 |
| Джо Блэк | 28.05.2015 | Написание отчетов о дефектах | 2 |
| Джо Блэк | 28.05.2015 | Написание отчета о результатах тестирования | 1 |
| Джим Уайт | 28.05.2015 | Аудит кода | 1 |
| Джим Уайт | 28.05.2015 | Парное тестирование | 1 |

**Статистика по новым дефектам.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Статус | Количество | Важность | | | |
| низкая | средняя | высокая | критическая |
| Найдено | 23 | 2 | 12 | 7 | 2 |
| Исправлено | 17 | о | 9 | 6 | 2 |
| Проверено | 13 | о | 5 | 6 | 2 |
| Открыто заново | 1 | о | о | 1 | о |
| Отклонено | 3 | о | 2 | 1 | о |

**Список новых дефектов.**

|  |  |  |
| --- | --- | --- |
| Идентификатор | Важность | Описание |
| BR21 | Высокая | Приложение не различает файлы и символические ссылки на файлы |
| BR22 | Критическая | Приложение игнорирует файлы.md во входном каталоге |
| Итак далее - описание всех 23 найденных дефектов | | |

**Статистика по всем дефектам.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Статус | | Количество | | Важность | | | | | | | | | |
| низкая | | | средняя | | высокая | | критическая | | |
| Проверено | | 17 | | о | | | 7 | | 7 | | 3 | | |
| Открыто заново | | 1 | | о | | | о | | 1 | | о | | |
| Отклонено | | 4 | | о | | | 3 | | 1 | | о | | |
| Найдено | | 1 34 | | 1 | | 5 | 1 | | 18 | 8 | 1 | | 3 |
| Исправлено | | 1 25 | | 1 | | 3 | 1 | | 12 | 7 | 1 | | 3 |

**Рекомендации.** В настоящий момент никаких изменений не требуется.

**Контрольные вопросы**

**(ответить на 10 вопросов на выбор)**

1. Какие существуют разновидности рабочей тестовой документации?

2. Check List: что описывают и когда используют?

3. Acceptance Sheet: что описывают и когда используют?

4. Test Survey: что описывают и когда используют?

5. Test Cases: что описывают и когда используют?

6. От чего зависит степень детализации каждой функциональной проверки?

7. Какие общие функциональные проверки выполняют для всего приложения?

8. Какие общие функциональные проверки выполняют для wеb-приложения?

9. Какая структура итогового отчета о результатах тестирования?

10. Перечислите базовые проверки для поля ввода данных.

11. Перечислите базовые проверки для поля загрузки файлов.

12. Перечислите базовые проверки для ввода даты.

13. Перечислите базовые проверки для поля со списком.

14. Перечислите базовые проверки для поля переключателя.

15. Перечислите базовые проверки для чек-бокса.

16. Перечислите базовые проверки для меню.

17. Перечислите базовые проверки для таблиц.

18. Перечислите базовые проверки для ссылок.

19. Перечислите базовые проверки для сообщений.

**Практическое задание**

1. Получить у преподавателя адрес сайта (или взять сайт выбранный на предыдущей практической работе).

2. В зависимости от сложности бизнес-логики wеb-приложения выбрать наиболее подходящий вид рабочей тестовой документации (Acceptance Sheet, Test Survey, Check list).

3. Анализируемое wеb-приложение разбить на модули и подмодули (3-5 модуля).

4. Разработать рабочую тестовую документацию, указанную в п.2, для всех модулей и подмодулей wеb-приложения.