**Лабораторная работа № 12 (2021)**

**Юзабилити-тестирование (ч.2)**

**1. Цель лабораторной работы:**

1.1. Приобретение умений по проведению юзабилити-тестирования.

1.2. Приобретение практических навыков по проведению экспертной оценки функциональности и управления вашего интерфейса.

**2. Краткие теоретические сведения**

**Методики тестирования**

Для выявления проблем удобства использования приложения, в том числе на ранних этапах планирования и разработки ПО, используется методику двойной проверки:

• Изучение опыта взаимодействия пользователя с приложением **через имитацию поведения пользователей**;

• Проверка соответствия принципам обеспечения удобства пользования и корректного визуального представления в контексте функциональных требований посредством **экспертной оценки**.

**Экспертная оценка** приложения осуществляется в соответствии с целями проекта, функциональными и нефункциональными требованиями к ПО.

Процедуры экспертной оценки включают в себя:

* Выявление и изучение возможных сценариев использования и путей пользователя (User Journeys) в контексте бизнес-целей и функционала приложения.
* Анализ информационной архитектуры приложения.
* Анализ интерфейса и элементов интерфейса.
* Анализ функционального соответствия.

Крупный недостаток юзабилити-тестирования – высокая стоимость. Более быстрым и дешевым способом проверки качества интерфейса является экспертная оценка. Она позволяет обнаружить порядка 80% проблемных мест.

Эксперт (или несколько) проводят аудит системы.

Виды экспертной оценки:

• проверка по контрольному списку

• эвристическая оценка

• мысленная прогонка по интерфейсу.

*Проверка по контрольному списку* ближе всего к формальному тестированию качества.

Составляется список произвольных требований т. н. чек-листы (см. Прил. 1), после чего интерфейс проверяется на соответствие этим требованиям.

Четкий контрольный список может использоваться кем угодно, что дает возможность вынести проверку интерфейса из деятельности юзабилити-специалиста, передав ее отделу контроля качества.

*Эвристическая оценка* была разработана Якобом Нильсеном и Рольфом Моличем (Прил. 2), которые надеялись с ее помощью сократить продолжительность проведения проверки по контрольному списку. При эвристической оценке вместо десятков и сотен конкретных требований интерфейс проверяется на соответствие всего нескольким общим принципам.

*Мысленная прогонка* формализует метод, по которому интерфейс оценивается. Если исходить из того, что интерфейс предназначен для использования функций, можно проверить, как эти функции вызываются и используются. Если просто проговорить словами, как работают интерфейсы всех функций, становится понятно, какие из них неоправданно подавлены, а какие работают недостаточно хорошо. Конечно, для этого тоже необходим опыт эксперта.

**3. Задание на лабораторную работу**

3.1. Описать возможную методику Ю-тестирования.

3.2. Разработать свой собственный контрольный список (на основе Приложения 1), в который включить проверку тех элементов интерфейса, которые у Вас присутствуют в приложении (обязательные – **Кнопки**, **Поля ввода, Списки, Формы, Меню**). Для этих элементов выбрать наиболее важные для вашего интерфейса характеристики (**не менее 3-х**) и протестировать интерфейс (**наличие скриншотов**, подтверждающих ваши утверждения – **обязательно**).

3.3. Провести эвристическую оценку по Якобу Нильсену и Рольфу Моличу (Прил. 2) (**наличие скриншотов**, подтверждающих ваши утверждения – **обязательно**).

3.8. Проанализировать результаты тестирования и составить отчет.

**4. Требования к оформлению отчета**

Отчет должен содержать:

1. Титульный лист;
2. Название и цели работы;
3. Описание методик эксперимента;
4. Результаты тестирования по контрольному списку с предоставлением скриншотов;
5. Результаты тестирования по Якобу Нильсену и Рольфу Моличу с предоставлением скриншотов
6. Основные (интерфейсные проблемы, проявляющиеся по всему интерфейсу) и частные проблемы (проблемы, проявляющиеся на отдельных экранах) выявленные в интерфейсе;
7. Предложения по модификации интерфейса.

**Приложение 1**

**Контрольный список интерфейса**

**Требования к конкретным элементам управления**

**Кнопки**

* Все кнопки, запускающие действия, имеют текст в инфинитивной форме глагола (пример: искать), а не другую часть речи либо форму глагола (пример: готово). Давать кнопке текст «ОК» можно, только если какой-либо глагол не вмещается.
* Кликабельный размер кнопок совпадает с их видимым или логическим размером.
* Между кнопками, стоящими рядом, должно быть пустое пространство, щелчок по которому не отрабатывается.
* Нет разных состояний кнопок, которые выглядят одинаково.
* Недоступные команды не исчезают с экрана, а становятся заблокированными.
* Частотные кнопки снабжены не только текстом, но и пиктограммами; редко используемые кнопки - только текстовыми подписями.
* В модальных диалоговых окнах нет кнопок Применить.

**Поля ввода**

* В полях ввода уже стоят наиболее вероятные значения.
* Если в поле вводится численное значение, границы диапазона выводятся во всплывающей подсказке.
* Если в поле вводится численное значение из ограниченного диапазона, поле снабжено крутилкой (Spinner).
* Длина полей не меньше, и, по возможности, не больше, длины вводимых в них данных.
* Если поле предназначено для ввода заметного количества текста, оно многострочное.
* Многострочные поля имеют максимально возможную высоту; нет резервов для их увеличения.

**Списки**

* В списках уже стоят наиболее вероятные значения.
* Если список содержит более 50 элементов, используется фильтр или режим поиска.
* Нет часто используемых коротких списков (менее пяти элементов); такие списки представлены как группы радиокнопок или чекбоксов.
* Ширина списков не меньше ширины входящих в них элементов.
* Элементы списка отсортированы; либо структурно, т.е. по общим признакам, либо по алфавиту, либо по частотности (только списки меньше 7 элементов).
* Если в списке более 50 отсортированных по алфавиту элементов, первыми тремя элементами являются наиболее частотные элементы. Они также повторяются на своих алфавитных местах.
* Многострочные списки множественного выбора снабжены чекбоксами возле каждого элемента (списки старого стиля отсутствуют).
* Многострочные списки имеют высоту не менее 4 строк.
* Если есть свободное место, используются расширенные комбобоксы, а не однострочные.

**Чекбоксы и радиокнопки**

* Если чекбоксов в группе больше 10, вводится дополнительный, выставляющий/снимающий все чекбоксы.
* Внутри группы радиокнопок одна обязательно установлена по умолчанию.
* Чекбоксы и радиокнопки внутри своих групп расставлены по вертикали.
* Если в окне, помимо терминационных кнопок, есть только набор радиокнопок, двойной щелчок по радиокнопке устанавливает ее и закрывает окно.

**Взаимодействие**

* Система, завершив длительную операцию (больше минуты работы), пищит через встроенный динамик компьютера.
* Если в интерфейсе не используется непосредственного манипулирования, система не имеет своих курсоров. Если непосредственное манипулирование применяется, свои курсоры применяются только если аналогов из ОС не существует.

**Системные сообщения и отработка ошибок**

* В формах ввода проверка корректности вводимых значений выполняется прямо во время ввода; если вводятся некорректные данные, система сразу сообщает об этом пользователю, не дожидаясь момента, когда пользователь завершит ввод данных во всей форме.
* Сообщения о некорректности введенных данных показываются рядом с элементом управления, данные в котором некорректны.
* Текст сообщений о некорректности введенных данных не говорит, что, дескать, совершена ошибка, напротив, он только информирует пользователя, данные какого типа и формата приемлемы.
* Если окно с текстом о некорректности введенных данных всплывающее — оно не должно заслонять собой поле ввода, где содержится текст с ошибкой.
* Текст сообщений о проблемах состоит из трех частей: в первой кратко описывается проблема, во второй части - как ее решить, в третьей - описывается, как не допускать возникновения этой проблемы в дальнейшем.
* Статусные сообщения («Синхронизация успешно завершена») выводятся только в строке статуса.

**Клавиатура**

* В формах ввода нажатие табуляции ведет к правильной последовательности перемещения по форме.
* Обработка формы запускается не только по нажатию на терминационую кнопку, но и по нажатию клавиши Enter на последнем поле этой формы.
* Для наиболее частотных элементов управления (включая меню) установлены клавиши быстрого вызова.
* Каждому пункту меню назначены ALT-комбинации (выделены подчеркиванием).
* ALT-комбинации и горячие клавиши стандартные.
* Если горячих клавиш больше 40, в интерфейсе есть способ их изменить.
* По нажатию клавиши Tab переход от элемента к элементу внутри формы осуществляется сверху вниз слева направо.

**Визуал**

* Направление теней во всех элементах управления должно быть одинаковым: снизу справа.
* Индикация
* Индикация цветом не является единственной; если она используется, система снабжена и другой индикацией.

**Пиктограммы**

* В группах пиктограмм нет пиктограмм, по цвету и форме сходных между собой.
* Нет пиктограмм со стандартными значениями, но нестандартными сюжетами.
* В пиктограммах нет текста.
* В наборах пиктограмм пиктограммы одного значения, но разных размеров используют одни и те же особенности и/или сюжет.

**Окна**

* На растягивающихся окнах есть индикатор растягиваемости.
* Заголовки окон соответствуют названиям элементов, при помощи которых окна были вызваны. Если окно вызывается элементом, не имеющим явного названия, в заголовке окна отражается название экранной формы.
* Тип окна (модальное, немодальное, возможность минимизации/максимизации) был выбран осознанно, в соответствии с задачами пользователей.
* В диалоговых окнах отсутствуют меню или инструментальные панели.
* Кнопки Применить используются только в окнах-палитрах (вместо кнопок ОК).

**Строка статуса**

* В строке статуса выводится только информация о текущем состоянии системы и кнопки (не выглядящие кнопками) для функций, предназначенных только опытным пользователям.
* Индикаторы выполнения выводятся в строке статуса. Исключение: окна-маcтера, в них индикаторы выполнения можно выводить внутри самих окон.

**Меню**

* Первая буква в названии пунктов меню - заглавная.
* Все пункты меню первого уровня активизируют раскрывающиеся меню.
* Используются не более двух подуровней меню.
* Если в меню есть пиктограммы, ими снабжены только самые частотные элементы.
* Элементы, открывающие вложенные меню, выглядят иначе, чем терминальные элементы.

**Контекстные меню**

* На всех объектах, видимых в интерфейсе, есть специфичное для каждого объекта контекстное меню.
* В контекстных меню не более 10 элементов.
* В контекстных меню элементы отсортированы по убыванию частоты их использования.
* Все элементы контекстных меню присутствуют и в других фрагментах интерфейса; нет команд, вызываемых только из контекстных меню.

**Структура интерфейсных форм**

* В группах интерактивных элементов (поля форм, элементы меню и т. п.) этих элементов не больше семи.
* Кнопка «Отмена» всегда самая правая.
* Многостраничные формы имеют указание на то, что они многостраничные; пользователь всегда видит количество оставшихся экранов (пример: «Экран x из y»).
* Если в форме есть несколько кнопок, одна является кнопкой по умолчанию. Если кнопка в форме только одна, она не может быть кнопкой по умолчанию. Опасные для пользователя кнопки не являются кнопками по умолчанию.
* Если в окне есть свободное место, наиболее частотная терминационная кнопка больше остальных.
* Кнопки находятся в секции, на которую они оказывают непосредственное воздействие.
* Терминационные кнопки (управляющие окном) расположены либо снизу в ряд, либо справа в колонку.
* Кнопки, относящиеся ко всему блоку вкладок, расположены за пределами блока.
* Если окно или вкладка имеет автоматически пополняемое содержимое, например, в нем перечислены приходящие сообщения, в названии элемента интерфейса, который открывает окно или вкладку, выводится число объектов в этом окне и отдельно число новых объектов. Пример: Документы (8/3).
* Пункты меню и кнопки, инициирующие другие действия пользователя, обозначены в конце многоточием (…). Примеры: элемент «Сохранить как…» требует многоточия, т.к. пользователь должен выбрать название файла, а элемент «О программе» многоточия не требует, т.к. на открывающемся окне нет самостоятельных интерфейсных элементов.
* Подписи к интерфейсным элементам размещены единообразно.
* Недоступные в данный момент интерфейсные элементы заблокированы, а не скрыты.
* В интерфейсе присутствуют сообщения о выполнении того или иного действия. Например, сообщение о том, что данные успешно сохранены или что-то удалено и т. д.

**Формы ввода**

* У формы ввода есть название.
* Во всех формах, служащих для сбора информации, есть пункты «Другое» и «Не применимо» или подобный.
* Все поля, обязательные для заполнения, помечены, и есть соответствующее пояснение.
* Поле «Отчество» никогда не является обязательным для заполнения.
* Многостраничные формы ввода имеют кнопки «Назад» и «Далее».
* Во всех формах, служащих для сбора информации, есть описание целей сбора данных, объясняется, что с этими данными будет сделано и что не будет.

**Текст**

* На все главные интерфейсные элементы повешены всплывающие подсказки, текст которых отражает результат использования этих элементов.
* В интерфейсе отсутствуют жаргонизмы.
* В интерфейсе отсутствуют отрицательные формулировки (например, чекбокс «Не показывать примечания» неприемлем, взамен него нужно выводить чекбокс «Показывать примечания»).
* Ни один элемент не называется по-разному в разных местах (интерфейсный глоссарий не просто сделан в явной форме, но и выверен).
* В тексте всех подтверждений дается наименование объекта, над которым совершается подтверждаемое действие.
* Для улучшения удобочитаемости длинные числа разбиваются неразрывным пробелом по три цифры: 1 234 567.
* Каждый элемент списка содержит на конце точку или начинается с прописной буквы по след. правилу: «Текст всех элементов начинается со строчной буквы. Все элементы оканчиваются по последней букве слова без каких-либо знаков препинания, кроме последнего, который оканчивается точкой. Исключение: если хоть один элемент списка содержит более одного предложения, все элементы начинаются с заглавной буквы и заканчиваются точкой.»
* Любому списку предшествует, по меньшей мере, один абзац текста.
* В таблицах все столбцы с цифрами выравниваются по правому краю.
* Точка в конце фразы отсутствует в заголовке (если он отделен от текста), в конце подписи под рисунком и в таблице.
* Подписи к интерфейсным элементам начинаются с прописной буквы и заканчиваются двоеточием.

**Приложение 2**

**Эвристическая оценка Якоба Нильсена и РольфаМолича:**

1. В любой момент времени система показывает, что с ней происходит.
2. Система использует термины, понятия и метафоры, присутствующие в реальном мире, а не обусловленные компьютером.
3. В любой момент пользователь контролирует систему, а не наоборот. Любую команду можно отменить или повторить.
4. В любой момент времени система выглядит и функционирует единообразным и стандартным способом.
5. Интерфейс системы препятствует появлению человеческих ошибок.
6. В любой момент времени интерфейс показывает объекты и команды сам, не требуя от пользователя вспоминать их.
7. В интерфейсе есть методы ускорения работы, предназначенные для опытных пользователей и не мешающие пользователям неопытным.
8. Интерфейс эстетичен и в любой момент времени не содержит ненужной сейчас информации.
9. Интерфейс помогает пользователям обнаруживать и исправлять проблемы, включая человеческие ошибки.
10. Справка доступна в любой момент времени. Она достаточна, но не избыточна; к ней легко обращаться; она не абстрактна, а нацелена на решение конкретных задач пользователя; в ней описываются конкретные шаги по решению проблем.