

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Кафедра технологий программирования

Лабораторная работа №5

По курсу “Проектирование человеко-машинных интерфейсов”

# **Анализ и оценка проектного решения**

Методические указания по выполнению лабораторной работы

Подготовила:

Давидовская М.И.,

Ст. преподаватель кафедры ТП

Минск, 2022 г.

## Содержание

Введение.....	3
Рекомендации по выполнению лабораторной работы.....	3
Методы быстрой оценки концепций.....	3
Обратная карточная сортировка.....	3
Метод "Тестирование ожиданий".....	5
Метод "Оценка восприятия дизайна".....	5
Метод "Оценка по контрольным спискам (чеклистам)".....	6
Примеры чеклистов для проверки приложения.....	10
Применение количественных методов для анализа элементов интерфейса.....	11
<i>Модель KLM для мобильных приложений</i> .....	15
Usability-тестирование.....	15
Показатели производительности (Performance Metrics).....	16
Показатели, основанные на аспектах для обсуждения usability.....	17
<i>Общие</i> показатели (Self-reported metrics).....	18
Поведенческие и физиологические показатели (Behavioral and Physiological Metrics).....	20
Задание на лабораторную работу.....	24
Комментарии к выполнению usability-тестирования.....	25
Выполнение обратной карточной сортировки: шаги.....	27
Тестирование информационной архитектуры с помощью treejack.....	28
Выполнение тестирования ожиданий: шаги.....	28
Исследование с помощью контрольного списка.....	29
Анализ интерфейса по модели GOMS и KLM.....	29
Рекомендуемые инструменты для обработки результата.....	29
Рекомендуемая структура отчёта по работе.....	30
Рекомендуемая литература.....	31

## Введение

**Цель работы** — изучить методы быстрой оценки концепций применительно к разработке пользовательского интерфейса приложения, провести оценку разработанного интерактивного прототипа и получить практические навыки по проведению usability-тестирования пользовательского интерфейса.

## Рекомендации по выполнению лабораторной работы

### Методы быстрой оценки концепций

К методам быстрой оценки концепций относят:

- методы с участием пользователя:
- обратную карточную сортировку;
- тестирование ожиданий;
- оценку восприятия дизайна;
- методы без участия пользователя:
- посценарная оценка;
- проверка по контрольным спискам (чеклистам);
- оценка по эвристикам;
- оценка на соответствие стандартам; • количественные методы оценки интерфейсов;
- другие методы.

По определению Алана Купера [1, с. 106] карточная сортировка — это один из методов экспериментальных оценок организации пользователем идей и информации, сводящийся к сортировке пользователем колоды карт, каждая из которых описывает определённую функциональность продукта или web-сайта либо содержит связанный с ним фрагмент информации. При этом карты готовятся на основе способов выполнения задач, решать которые должен проектируемый продукт.

### Обратная карточная сортировка

Обратная карточная сортировка необходима для проверки того, что спроектированная структура навигации понятна пользователю.

По оценке Дмитрия Сатина [2], существуют следующие виды обратной карточной сортировки:

- тестирование дерева (списком либо карточками);
- тестирование экранов.

Самым распространённым считается тестирование дерева, поэтому зачастую термины «тестирование дерева» и «обратная карточная сортировка» рассматривают как синонимы, смотри, в частности, статью [3].

Основная идея метода базируется на том, что крупный web-ресурс, как правило, организован иерархически — в виде дерева тем и подтем. Тестирование дерева фактически есть оценка того, как пользователи могут найти нужные разделы в этой иерархии. Вопреки представлениям о традиционном usability testing, тестирование дерева производится не на самом web-сайте, а на упрощённой текстовой версии структуры сайта.

Суть процедуры:

1. Будущему пользователю даётся задание что-либо найти, например, товар определённой категории (например, наручные часы) из заданного ценового диапазона (например, от 50 руб.).
2. Каждому участнику тестирования ("будущему пользователю") показывается текстовый список тем "верхнего уровня".
3. Участники теста выполняют "спуск по дереву" с целью поиска того раздела (= темы), который соответствует заданию.
4. Процедура повторяется для каждого участника по разным предметам поиска (например, для разных категорий товаров, поставщиков, сетей магазинов и т.п.).

Анализ результатов такого теста призван ответить на вопросы:

- Можно ли (в принципе) найти нужную информацию на web-сайте, моделью которого является это дерево?
- Можно ли найти нужную информацию, не выполняя "поиск с возвратом"?
- Где имеет место наибольшая вероятность "заблудиться"?
- Насколько существенны интеллектуальные усилия будущего пользователя для перехода от одной темы к другой?

- "Успешные" и "провальные" фрагменты дерева будущего сайта – выделить контрасты.

В качестве инструментов здесь используются лист бумаги, ручка и карандаш, либо специализированное программное обеспечение (Visio, Dia, Cacao.com и т.п.).

Для анализа информационной архитектуры обратной карточкой сортировкой можно использовать сервис Treejack от OptimalWorkshop (<https://support.optimalworkshop.com/hc/en-us/categories/200144524-Treejack>).

## **Метод "Тестирование ожиданий"**

Другой известный метод быстрой оценки ранних концепций – тестирование ожиданий.

По оценке Д. Сатина [2], тестирование ожиданий призвано ответить на вопрос: "Соответствуют ли ожидания пользователей запланированному поведению системы?".

Суть: тестирование ожиданий есть тест на понимание пользователями процессов, происходящих в системе. Проводится либо как самостоятельное исследование, либо в составе Usability Testing (UT).

При проведении этого вида тестов пользователям задаются вопросы об ожидаемом поведении того или иного элемента интерфейса будущего приложения. Типичные вопросы:

- "Что вы ожидаете увидеть, когда нажмёте на эту кнопку (щёлкните эту ссылку и т. п.)?";
- "Что произойдёт после нажатия этой кнопки?".

Далее ожидания пользователей сравниваются с тем, что либо уже реализовано, либо планировалось реализовать. В результате теста должны быть выявлены те элементы интерфейса, поведение которых пользователю неочевидно.

## **Метод "Оценка восприятия дизайна"**

Ещё один метод оценки ранних концепций – оценка восприятия дизайна. Используется для оценки нескольких вариантов дизайна пользовательского интерфейса с целью выбрать тот, который лучше других будет вызывать у пользователей "целевые" эмоции.

Последовательность шагов:

1. Задать список прилагательных, с которыми должен ассоциироваться дизайн и их антонимов (пример: современный – классический, интересный – скучный и т. п.).
2. Нарисовать несколько макетов.
3. Распечатать макеты в цвете.
4. На основании прилагательных и их антонимов сформировать шкалы.
5. Провести тест с пользователями (~10), которые по шкалам оценивают дизайн.
6. Обработать результат.

Обработка результатов:

1. Посчитать средние (или суммы) по каждой паре прилагательных.
2. Построить диаграмму-профиль для каждого варианта дизайна.
3. Выбрать вариант дизайна, лучше соответствующий целевым эмоциям.

## **Метод "Оценка по контрольным спискам (чеклистам)"**

На основании экспертного мнения, стандартов, примеров готовых списков составляется контрольный список проверки, т. е. формируется список утверждений, которые можно проверить на сайте/ при использовании мобильного приложения. В основе контрольного списка:

- Стандарты платформы
- Паттерны поведения
- Сложившиеся привычки
- Регламенты
- Другое

Применяется в тех случаях, когда анализ необходимо провести быстро и когда соблюдаются условия:

1. Есть устоявшиеся проверенные предположения относительно работы продукта.
2. Есть серия характеристик, которыми должен обладать продукт от версии к версии.
3. Необходимо сравнить два и более продукта одного класса.

Что может быть в основе контрольного списка:

- Стандарты платформы;
- Брендбук;
- Существующие привычки;
- Best practices;
- Существующие чеклисты;
- Userfocus (в переводе UsabilityLab);
- Обобщение эвристик Нильсена-Шнейдермана (см. пример контрольного листа);
- «usability checklist» (найти примеры по запросу в поисковой системе).

Чтобы составить контрольный список необходимо выбрать ключевые параметры и характеристики, которыми должен обладать интерфейс:

- Последовательности
- Времена отклика и времена выполнения задачи
- Типовое поведение
- Принципы восприятия
- Сформулировать тезисы на основе характеристик.
- Проверить по тезисам исходный интерфейс.

### ***Пример контрольного списка***

#### **Архитектура и навигация**

1. Соответствует ли структура сайта целями, для достижения которых он предназначен?
2. Понятна ли схема навигации?
3. Можно ли определить в каком месте сайта вы находитесь?
4. Как вы находите на сайте то, что вам нужно?
5. Является ли разумным количество элементов в навигационных панелях?
6. Логично ли отсортированы элементы навигационных панелей?
7. Названия гиперссылок соответствуют названиям страницы?
8. Гиперссылки выделены отчетливо?
9. Существует ли отчетливо выделенная ссылка на главную страницу?

10. Существует ли возможность поиска информации на сайте?
11. Существует ли карта сайта?
12. Каждая ли страница позволяет понять, на каком сайте вы находитесь?
13. Может ли пользователь управлять навигацией по сайту?

### **Планировка и дизайн**

14. Размер страницы превышает размер окна?
15. Схема планировка повторяется на всех страницах?
16. Существует ли отчетливый фокус на каждой странице?
17. Планировка визуально видна?
18. Эффективно ли используется выравнивание?
19. Эффективно ли используется группировка?
20. Есть ли хороший контраст?
21. Не громоздкая ли планировка?
22. Нравится ли сайт эстетически?

### **Содержание**

23. Тексты на сайте понятны и лаконичны?
24. Организован ли текст в виде небольших блоков?
25. Встречаются ли в тексте грамматические и орфографические ошибки и опечатки?
26. Содержат ли страницы вводный текст?
27. Поддерживают ли мультимедийные компоненты задачи пользователя?
28. Являются ли единицы измерения, используемые на сайт понятными и не вызовут ли они трудностей при использовании их иностранцами?
29. Представлены ли на сайте время и дата создания страниц?
30. Представлены ли на сайте номера контактных телефонов?
31. Представлены ли на сайте адреса с почтовыми индексами?

### **Формы и взаимодействие**

32. Соответствуют ли формы задачам пользователя?
33. Обладают ли диалоги логичной последовательностью шагов?



- 34. Обладают ли диалоги понятной кнопкой или ссылкой для перехода к следующему шагу?
- 35. Являются ли диалоги последовательными и лаконичными?
- 36. Все ли элементы форм используются по назначению?
- 37. Сгруппированы ли элементы формы по своей сути?
- 38. Понятно ли выглядит кнопка отправки формы?

#### **Графика**

- 39. Является ли качество используемой графики приемлемым?
- 40. Все ли графические элементы имеют альтернативные текстовые надписи?
- 41. Содержат ли графические элементы информацию о размере файла?
- 42. Оптимизированы ли графические элементы для передачи по Интернету?
- 43. Реагируют ли графические элементы на движения мышки? Это полезно?
- 44. Используется ли анимация? Её не слишком много? Объем файлов приемлемый?

#### **Цвета и оформление текста**

- 45. Подходящий ли выбор цветов для сайта?
- 46. Используется ли слишком много цветов?
- 47. Цвета используются логично и последовательно?
- 48. Адекватно ли различаются используемые цвета в черно-белом режиме?
- 49. Являются ли тексты понятными?
- 50. Размер шрифта достаточно большой?
- 51. Цвет шрифта подходящий и достаточно контрастный?
- 52. Отформатирован ли текст так, чтобы в строке было от 10 до 12 слов?
- 53. Достаточной ли ширины поля вокруг текста?
- 54. Гарнитура шрифта используется надлежащим образом и последовательно?

#### **Устойчивость к ошибкам**

- 55. Должен ли пользователь что-нибудь запоминать, переходя между страницами?
- 56. Возникает ли предупреждение при попытке совершения необратимых или дорогостоящих действий?

57. Можно ли отменить рискованные или дорогостоящие действия?
58. Перехватываются ли возникающие ошибки локально, без обращения к серверу?
59. Содержат ли страницы с сообщением о возникших ошибках полезную информацию?
60. Содержат ли страницы с пустыми результатами поиска советы по расширению условий поиска?
61. Существует ли система помощи (справки)?
62. Структурирована ли помощь по задачам пользователя? Объясняет ли она пользователю, как совершить то, или иное действие?
63. Система помощи контекстно-зависимая? **Платформа**
- и особенности реализации**
64. Загрузка страниц происходит достаточно быстро? Занимает ли она от 3 до 15 секунд?
65. Все ли гиперссылки работают правильно?
66. Существуют ли поврежденные графические элементы?
67. Написан ли текст страниц так, чтобы их могли найти поисковые системы?
68. Работает ли сайт с браузером пользователя?
69. Работает ли сайт с оборудованием, которое использует пользователь?
70. Работает ли сайт на мониторах высокого и низкого разрешения?
71. Используются ли нестандартные plug-in'ы? Являются ли они необходимыми и полезными?

### **Примеры чеклистов для проверки приложения**

- Чеклист для оценки юзабилити приложения — <https://www.cossa.ru/trends/178643/>
- Обязательный чек-лист разработки UX-дизайна мобильного приложения — <https://h.amazingsoftworks.com/ru/company/skillbox/blog/418287/>
- Чеклист UX/UI дизайнера — <https://medium.com/ux-draft/3a-что-мы-любим-чек-лист-и-почему-расшарили-его-на-команду-a0b6cd9abb13>
- Полный чек-лист тестирования веб-приложения — <https://www.software-testing.ru/images/stories/library/complete-web-application-testing-checklis.pdf>

- Usability UX Checklist — <https://edufpmi.bsu.by/mod/resource/view.php?id=8245>
- Пример чеклиста для формы регистрации и логина — <https://www.nngroup.com/articles/checklist-registration-login/>.
- Чеклист для аудита юзабилити — <https://vc.ru/marketing/152547-polnyy-chek-list-dlya-samostoyatel'nogo-audita-yuzabiliti-sayta>
- Чек-лист Usability сайта 2020 — <https://zen.yandex.ru/media/id/5c1286654d027e00a966e775/cheklist-usability-saita-2020-203-punkta-5ecf4e439c18f0062c0d6aaa>
- Usability чеклист — <https://habr.com/ru/post/178867/>
- Чек-лист по юзабилити: 200+ пунктов на проверку — <https://texterra.ru/blog/chek-list-po-yuzabiliti-200-punktov-na-proverku.html>
- Чек-лист по юзабилити сайта — <https://web-optimizator.com/checklist-usability/>

## Применение количественных методов для анализа элементов интерфейса

Существует множество методов количественного анализа элементов интерфейса. Одним из лучших подходов к количественному анализу моделей интерфейсов является классическая модель GOMS — «правила для целей, объектом, методов и выделения» (the model of goals, objects, method and selection rules). Моделирование GOMS позволяет предсказать, сколько времени потребуется опытному пользователю на выполнение конкретной операции при использовании данной модели интерфейса.

Разработчики этой модели обнаружили, что для решения поставленной задачи при работе с компьютером пользователь совершает определенный набор жестов. Таким образом, время выполнения задачи складывается из времени выполнения этих жестов и из промежутков между жестами, которые требуются для обдумывания.

В результате тщательного лабораторного исследования удалось получить среднее время выполнения различных жестов:

- $K = 0.2$  с - время, необходимое для **нажатия клавиши клавиатуры**

- $P = 1.1$  с - время, необходимое для **перемещения указателя мыши** к определенной позиции на мониторе
- $H = 0.4$  - время, необходимое для **перемещения руки пользователя** с клавиатуры на мышшь
- $M = 1.35$  - ментальная пауза, т.е. время, необходимое пользователю на **обдумывание следующего шага**.

Конечно, временные показатели различных жестов могут сильно отличаться, ведь все люди разные. Но с помощью этих временных значений мы можем сравнивать интерфейсы между собой.

### Правила расстановки ментальных операций

<b>Правило 0</b>	Операторы М следует устанавливать:
Начальная расстановка операторов М	<ul style="list-style-type: none"> <li>☒ перед всеми операторами К;</li> <li>☒ перед всеми операторами Р, предназначенными для выбора команд.</li> </ul> Перед операторами Р, предназначенными для указания на аргументы выбранных команд, ставить оператор М не следует.
<b>Правило 1</b> Удаление ожидаемых операторов М	Из последовательности «Оп1 М Оп2» оператор М может быть удален, если оператор Оп2 является полностью ожидаемым с точки зрения оператора Оп1. Например, курсор мыши перемещается к нужному объекту с целью нажатия кнопки по его достижении (последовательность Р К). Согласно правилу 0 эта последовательность преобразуется в Р М К, затем, согласно данному правилу, в Р К.
<b>Правило 2</b> Удаление операторов М  внутри когнитивных единиц	Когнитивной единицей является непрерывная последовательность вводимых символов, которые образуют название команды или аргумент (например, имена файлов при сохранении, значения временных интервалов как параметры автосохранения и т.д.). Если строка вида М К М К М К ... принадлежит когнитивной единице, то следует удалить все операторы М, кроме первого.
<b>Правило 3</b> Удаление операторов М перед последовательными разделителями и	Если оператор К означает лишний разделитель, стоящий в конце когнитивной единицы, то следует удалить находящийся перед ним оператор М.

<b>Правило 4</b> Удаление операторов М, которые являются прерывателями команд	Если оператор К является разделителем, стоящим после постоянной строки, то стоящий перед ним оператор М удаляется. Постоянными строками являются, например, имена команд или ответы Y/N на запрос. Добавление пробела перед аргументами команды или нажатие клавиши Enter при ответе в этих случаях в силу привычности не требует ментального оператора. Если оператор К является разделителем для строки аргументов или любой другой изменяемой строки, то оператор М перед ним сохраняется.
<b>Правило 5</b> Удаление перекрывающихся операторов М	Не следует учитывать любую часть оператора М, перекрывающую оператор R. Так, последовательность R М или М R означает, что ментальная операция может быть совмещена с ожиданием ответа компьютера. Если время ожидания превышает время ментальной операции, то последняя не учитывается полностью.

Goms - модель, основанная на оценке скорости печати. Разработчики модели GOMS во время ее создания заметили, что время, требующееся для выполнения какой-то задачи системой "пользователь-компьютер", является суммой всех временных интервалов, которые потребовались системе на выполнение последовательности элементарных жестов, составляющих данную задачу.

#### **Пример 1. Форма авторизации**

Пусть интерфейсы одинаковы с точки зрения внешнего вида. Каждый из них содержит 2 поля ввода - Логин и Пароль, и 2 кнопки - ОК и Отмена. Предположим, что пользователю нужно ввести логин из 5 символов, пароль из 5 символов и нажать на кнопку ОК. Причем в первом интерфейсе переход между полями ввода можно делать Tab-ом, а для нажатия на клавишу ОК можно просто нажать Enter. Во втором интерфейсе для перехода между полями и для нажатия на кнопку ОК нужно использовать мышь. Кажется все и так очевидно, но все-таки подсчитаем, какой интерфейс лучше с точки зрения модели GOMS:

- На выполнение задачи в первом интерфейсе пользователь затратит KKKKK (5 нажатий клавиши клавиатуры для ввода логина) + К (переход к полю Пароль с помощью клавишу Tab) + KKKKK (ввод пароля) + К (нажатие Enter, чтобы подтвердить ввод) = 2.4 секунды;
- На выполнения задачи во втором интерфейсе пользователь затратит KKKKK (ввод логина) + Н (перемещение руки на мышь) + Р (указание на поле Пароль) + клик мышью + Н (перемещение руки на клавиатуру) + KKKKK (ввод пароля) + Н (перемещение руки на мышь) + Р (указание на кнопку ОК) + клик мышью = 5.6 секунд (без учета времени клика мышью)

Вывод: первый вариант интерфейса значительно быстрее второго, с помощью первого задача выполняется за 2.4 секунды, с помощью второго - более чем за 5.6 секунд. Результат на лицо)

Таким способом можно сравнивать любые интерфейсы и выносить заключение о том, какой интерфейс более быстрый, а какой будет сильно тормозить работу.

### Пример 2. Приложение перевода температур из одной шкалы в другую

Необходимо спроектировать интерфейс приложения, переводящего температурные показания из шкалы Фаренгейта в шкалу Цельсия или наоборот.

Значение температуры можно ввести с помощью клавиатуры или мыши. Это значение представляется в среднем четырьмя знаками.

Результат перевода должен отображаться на экране монитора.

Критерий эффективности интерфейса: минимальное время перевода из одной шкалы в другую и максимальная скорость и точность операций (собственно время пересчета температуры не учитывается; речь идет только об операциях с интерфейсом).

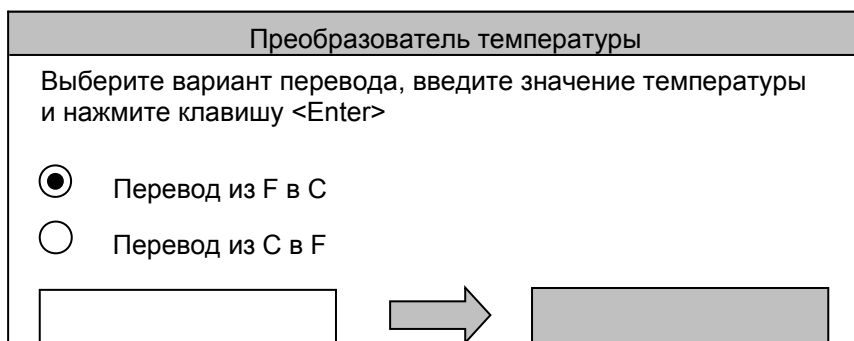


Рис. 1

#### Оценка по модели GOMS

Действия пользователя

Очередное действие пользователя	Формируемая последовательность операций
Перемещение руки к мыши	Н
Перемещение курсора к необходимому переключателю в группе	Н Р
Нажатие на необходимый переключатель	Н Р К
Перемещение руки к клавиатуре	Н Р К Н
Ввод четырех символов	Н Р К Н К К К К

Нажатие клавиши <Enter>	Н Р К Н К К К К К
-------------------------	-------------------

*Расстановка ментальных операций*

Начальная расстановка операторов М согласно табл. 1 (аргументов команд в данном случае нет)	Н М Р М К Н М К М К М К М К М К
Удаление ожидаемых операторов М по правилу 1 (Р М К преобразуется в Р К)	Н М Р Н М К М К М К М К М К К
Удаление операторов М внутри когнитивных единиц по правилу 2. Перед конечным К оператор М остается согласно правилу 4	Н М Р К Н М К К К К К М
Замена символов операторов соответствующими интервалами согласно табл. 2 и подсчет общего времени работы	$H + M + \dots = 0.4 + 1.35 + \dots = 7.15$ с

Если переключатель уже установлен в нужное положение (перевод из F в C), метод действий становится следующим: М К К К К М К

Соответствующее суммарное время равно 3.7 с.

Случаи перевода температуры из одной шкалы в другую равновероятны. Тогда среднее время перевода равно  $(7.15 + 3.7) / 2 = 5.4$  с.

## **Модель KLM для мобильных приложений**

Приложения можно оценивать по времени работы пользователя без запуска прототипа на телефоне. Для этого используется модель KLM, для которой определяются базовые элементы взаимодействия для мобильных телефонов и приложений, далее выполняется оценка эффективности пользователей, полученная на основе нескольких пользовательских тестов.

Подробнее о модели KLM для мобильных приложений:

- <https://sites.google.com/site/436goms/gomsmodel/klm-for-phones>
- <https://norwegianrockcat.com/assets/iris31.pdf>

## **Usability-тестирование**

Usability-тестирование определено как набор методов, позволяющих измерить характеристики взаимодействия пользователя с продуктом с целью оценки уровня пользовательских свойств продукта.

Детально различные методы проведения usability-тестирования описаны в книге «Tullis T., Albert B. Measuring the User Experience: Collecting, Analyzing, and

Presenting Usability Metrics» (<https://edufpmi.bsu.by/mod/url/view.php?id=8247>). Ниже представлено краткое изложение методов, описанных в данной книге.

Применительно к программному обеспечению в целом наиболее известны следующие виды показателей usability:

- показатели производительности (Performance Metrics);
- показатели, основанные на аспектах для обсуждения usability (т.н. IssuesBased Metrics);
- самоговорящие показатели (Self-reported metrics);
- поведенческие и физиологические показатели (Behavioral and Physiological Metrics);
- комбинированные и сравнительные показатели.

### ***Показатели производительности (Performance Metrics)***

К числу показателей производительности относят:

- успех задания (task success);
- время выполнения задания (time-on-task);
- ошибки (errors);
- эффективность (efficiency);
- обучаемость (learnability).

Успех задания показывает, сколько пользователей без излишних физических и умственных затрат достигнут своей цели, решая типичные для продукта задачи. Известны два подвида этого показателя: «двоичный» успех (задание либо выполнено, либо не выполнено) и градация успешности задания.

Для оценки времени выполнения задания вводятся дискретные временные интервалы и каждому из интервалов ставится в соответствие число участников тестирования, уложившееся в него. Далее анализируется разброс времён выполнения задания участниками и пороговые значения: сколько участников теста уложились в приемлемое время.

Под эффективностью (efficiency) понимается объём усилий (когнитивных и физических) в совокупности с анализом числа действий или шагов для выполнения задания.

Обучаемость определяется временем и усилиями, потраченными на освоение чего-либо. Часто меру эффективности рекомендуют представлять как функцию от



опыта пользователя. При сборе данных для измерения обучаемости требуется наблюдать за тем, как различные меры эффективности изменяются со временем: время на задание, количество ошибок, число шагов, показатель успешности заданий в минуту.

## ***Показатели, основанные на аспектах для обсуждения usability***

Юзабилити (от англ. usability — «удобство и простота использования, степень удобства использования», также удобство использования, пригодность использования, эргономичность — способность продукта быть понимаемым, изучаемым, используемым и привлекательным для пользователя в заданных условиях (ISO/IEC 25010)[2]; свойство системы, продукта или услуги, при наличии которого конкретный пользователь может эксплуатировать систему в определенных условиях для достижения установленных целей с необходимой результативностью, эффективностью и удовлетворённостью (ISO 9241-210).

Удобство (пригодность) использования системы не сводится только к тому, насколько её легко эксплуатировать. В соответствии со стандартами серии ISO 9241 эту характеристику следует понимать более широко, учитывая личные цели пользователя, его эмоции и ощущения, связанные с восприятием системы, а также удовлетворённость работой. Свойства, необходимые для обеспечения пригодности использования, зависят также от задачи и окружающей среды. Пригодность использования — не абсолютное понятие, оно может различным образом проявляться в определённых условиях эксплуатации.

Существует два основных способа оценки удобства ( пригодности ) использования продукта:

- прямая оценка на основе анализа результативности, эффективности и удовлетворённости, достигнутых в результате эксплуатации продукта в реальных условиях: если в указанных условиях одна система более эргономична, чем другая, то оценка должна это выявлять;
- косвенная оценка на основе анализа отдельных подхарактеристик, отражающих определённые свойства системы в установленных условиях эксплуатации.

Показатели, основанные на аспектах для обсуждения usability, предполагают выявление этих аспектов (в лаборатории, путём online-тестирования, либо путём

экспертной оценки пользовательского интерфейса) с целью определения того, какие из них – истинные, а какие – ложные. Истинные аспекты характеризуются тем, что либо с ними сталкивается большинство участников тестирования, либо минимум один участник теста, но выполнен анализ его поведения, последовательность действий участника логически обоснована и подтверждает наличие аспекта для обсуждения.

Примеры отрицательных аспектов для обсуждения usability:

- всё, что мешает успешному завершению задания;
- всё, что сбивает пользователя с пути решения задачи;
- всё, что вводит пользователя в замешательство.

«Полезные» отрицательные аспекты указывают на возможные улучшения продукта, положительные (пример – ясное отображение сложной информации) перечисляются, чтобы не быть потерянными в дальнейших итерациях проектирования.

Один из распространённых критериев оценки аспектов для обсуждения usability – уровень серьезности (по степени влияния на опыт пользователя, а также по комбинации факторов частоты использования и влияния на бизнесцели).

Факторы, влияющие на выявление usability-вопросов:

- разнородность состава участников (по мотивации, уровню знаний и опыту работы в заданной предметной области и др. );
- типизация заданий, а также их формулировка;
- метод исследования;
- форма продукта («бумажный» прототип, полнофункциональный прототип и готовый продукт);
- среда, в которой проводится тестирование (к её характеристикам относят прямое или косвенное взаимодействие между испытуемыми, освещение, расположение рабочих мест, присутствие наблюдателей, видеозапись процесса тестирования);
- состав модераторов – лиц, ответственных за проведение тестирования.

### **Общие показатели (Self-reported metrics)**

Общие показатели (self-reported metrics) – показатели usability, которые в общих чертах можно охарактеризовать как дающие наиболее общую информацию о пользовательском восприятии программного продукта и взаимодействия с ним. Такие метрики вводятся в рассмотрение, когда требуется выяснить:

- общий уровень удовлетворённости;

- простоту использования;
- эффективность навигации;
- ясность терминологии;
- осведомлённость о конкретных функциях;
- визуальную привлекательность, то есть некий неформальный обобщённый показатель на основе субъективных данных.

В качестве подходов к сбору данных наиболее известны: шкалы оценки (рейтинги), списки атрибутов, открытые вопросы (вопросы типа «перечислите три особенности продукта, которые Вам понравились больше всего»). Из шкал используются шкала Ликерта и семантически дифференцируемая шкала. Первая используется для случаев нечётного числа степеней согласия в некотором вербально выражаемом значении от полюса к полюсу. Например, «полностью отрицаю», «не согласен», «затрудняюсь ответить», «согласен», «полностью разделяю». Для второй шкалы характерно наличие пары полярных по значению прилагательных (например, «сильный» – «слабый») и нечётное число позиций между ними, соответствующих близости тому или другому полюсу, что составляет SUS – System Usability Scale. Для сравнения следует упомянуть CSUQ – Computer System Usability Questionnaire как своеобразный симбиоз шкал первого («согласен» – «не согласен») и второго типа (позиции, определяющие близость к тому и другому полюсу).

Дополнительно о SUS:

1. <https://measuringu.com/sus/>
2. <https://usabilitygeek.com/how-to-use-the-system-usability-scale-sus-to-evaluate-the-usability-of-your-website/>

При использовании указанных шкал анализ результата состоит в том, чтобы назначить каждой позиции рейтинга числовое значение, которое усредняется по участникам. Близкой этой идеи является анкета вида QUIS – Questionnaire for User Interaction Satisfaction. Другой подход – рассматривать только два крайних значения и узнать процент пользователей, которые попали в тот и в другой «полюс».

Подход на основе сеансов обратной связи с пользователями приемлем для технической аудитории, которая способна «домысливать» тот графический материал, который представляет интерфейс программного продукта (веб-страницы). Может быть альтернативой usability-тесту при наличии ограничений во времени, когда невозможно провести «классический» usability-тест. Как правило, обратная связь задействуется на момент завершения примерно половины работ

над проектом. Цель её введения – выявление тех мест, где пользовательский интерфейс нуждается в улучшении.

Варианты организации обратной связи для веб-приложений:

- всплывающий (то есть по аналогии с всплывающими подсказками) опрос случайного пользователя при выходе из системы;
- сбор обратной связи в различных местах Internet-страницы от всех желающих.

Известные реализации обратной связи:

- Website Analysis and Measurement Inventory (WAMMI), разработчик – Human Factor Research Group of University College Cork in Ireland: [www.wammi.com](http://www.wammi.com);
- American Customer Satisfaction Index (ACSI), разработчик – Business School of the University of Michigan, адрес в Internet: [www.TheACSI.org](http://www.TheACSI.org). Этот вид обратной связи используется на [www.ForeSeeResults.com](http://www.ForeSeeResults.com);
- постраничный аналог тестирования «после каждой задачи» – обратная связь от OptionLab, адрес в Internet: [www.OptionLab.com](http://www.OptionLab.com).

На что следует обратить внимание при использовании online-средств:

- число задаваемых пользователю вопросов;
- добровольность обратной связи;
- дублирование данных (от одного пользователя).

### ***Поведенческие и физиологические показатели (Behavioral and Physiological Metrics)***

К поведенческим и физиологическим показателям usability относят признаки поведения пользователя, куда входят явные признаки поведения и признаки, регистрируемые с применением специального оборудования. Явные вербальные признаки поведения включают:

- соотношение позитивных и негативных комментариев пользователя;
- если категорий комментариев более трёх, то в роли таких признаков будет выступать доля комментариев, попавших в каждую группу.

Анализируемые невербальные признаки включают выражение лица и язык тела (расположено ли лицо пользователя близко к экрану, характер перемещения «мыши» и др.).

Ряд признаков поведения регистрируются с применением специального оборудования: выражение лица, реакция зрачка, электропроводность кожи, сердечный ритм и др.

Комбинированные и сравнительные usability-показатели призваны ответить на вопрос: «Какова общая картина usability продукта?» Сюда входят:

- комбинированные показатели (статистика);
- графические оценочные сводки (usability scorecards);
- сравнительные usability-показатели (сравнение с эталоном).

К первой группе показателей относят:

- вычисление относительно конечных целей;
- вычисление процентных соотношений;
- вычисление z-значений;
- единый usability-показатель от J. Sauro и др. (SUM – Single Usability Metric).

Последовательность шагов процедуры вычисления комбинированных показателей относительно конечных целей:

- сформулировать конечные цели, например: «Каждый участник тестирования должен выполнить минимум 80% заданий, тратя на каждое в среднем не более 70 секунд»;
- для каждого участника выставить двоичную оценку достижения цели (1 – достигнута/0 – не достигнута);
- вычислить процент участников, достигших этих целей в ходе исследований.

Для комбинированных показателей, основанных на процентных соотношениях, процедура вычисления будет следующей:

1. преобразовать значение каждого отдельного показателя (A) в процентное соотношение (P, %):
  - без использования весовых коэффициентов;
  - с использованием весовых коэффициентов;
2. усреднить процентные соотношения разных показателей для каждого участника;
3. усреднить эти «средние» по всем участникам.

Правила преобразования (или правила выбора минимальных или максимальных значений) будут следующими.

1. если по определению значения показателя – от 0 до 100, то  $P = A\%$ .  
*Пример: SUS – System Usability Scale (см. выше);*

2. если минимум равен 0, а максимум известен (обозначим его далее как Max), то  $P = (A / \text{Max}) * 100\%$ . *Пример: число успешно выполненных заданий, удовлетворённость по оценочной шкале;*

3. если минимум равен 0, а максимум неизвестен, то Max = <наибольшее значение, выявленное в ходе исследования>. Когда цель исследования – минимизация значения A, тогда выбирают  $P = (1 - (A / \text{Max})) * 100\%$ . *Пример: количество ошибок;*

4. если минимум и максимум неизвестны, то Max = <наибольшее значение, выявленное в ходе исследования>, Min = <наименьшее значение, выявленное в ходе исследования>. Когда цель исследования аналогична правилу 3, тогда выбирают  $P = (\text{Min} / A) * 100\%$ . *Пример: время выполнения задания.*

Для комбинированных показателей, которые основаны на вычислении z-значений (z-scores), базовым является предположение о нормальном распределении самих показателей.

Пусть  $x$  – значение показателя, подлежащее преобразованию,  $\mu$  – (теоретическое) среднее распределения (оценка математического ожидания),  $\sigma$  – стандартное отклонение распределения показателя (оценка среднеквадратического отклонения величины  $x$  относительно несмещённой оценки её дисперсии). Каждое множество показателей нормализуется ( $\mu=0, \sigma=1$ ):  $z = (x - \mu) / \sigma$ .

При этом для усреднения шкалы всех показателей должны иметь одинаковое направление. При усреднении z-значения всех показателей делают одинаковый вклад в среднее z-значение. Недостаток – нельзя использовать обобщённое среднее z-значение как обобщённый показатель usability, так как по определению он равен 0.

Область применения комбинированных показателей, основанных на вычислении z-значений:

- итеративное исследование разных прототипов (данные из разных usability исследований);
- сравнение данных от разных групп пользователей (в одном usability исследовании);
- сравнение разных условий или дизайна (в одном usability исследовании).

Например, если цель эксперимента поставлена как изучение влияния возраста пользователей Internet-ресурса на производительность последнего, то в качестве z-значений могут быть комбинации показателей времени на задачу и успешности заданий.

Единый usability-показатель (Single Usability Metric, SUM) характеризуется тем, что в равной мере учитываются несколько метрик:

- показатель успешности выполнения заданий;
- время на задание;
- ошибки при выполнении задания;
- позадачная оценка удовлетворённости (близкая к After-Scenario Questionnaire (ASQ)), включающая три пятибалльных шкалы: простота использования, удовлетворённость и воспринимаемое время.

Рекомендуемая Excel-таблица для вычисления SUM представлена по адресу: <http://www.measuringusability.com/sum>. Здесь Completion – процентный показатель завершённости выполнения задания, Satisfaction – удовлетворённости, Time – процент участников, уложившихся в приемлемое время, Errors – процент ошибок.

Графические оценочные сводки (usability scorecards) имеют целью представить данные usability-тестирования таким образом, чтобы общие тенденции и наиболее важные аспекты стали наглядными – на сводных оценочных диаграммах. При этом используются два распространённых подвида диаграмм: для двух показателей и для трёх и более показателей. Пример сопоставления двух показателей – уровень (завершённость) выполнения задания и субъективная оценка выполнения заданий (удовлетворённость).

Графические оценочные сводки для трёх и более показателей, как правило, имеют вид лепестковых диаграмм и представляют обобщённые значения показателей – итоги usability-исследования. Пример: сопоставление уровня выполнения заданий, частоты посещения Internet-страницы, процента ошибок, оценок удовлетворённости и полезности.

Показатели, основанные на сравнении с гипотетическим эталоном, требуют численно измеряемых целей, которые формулируются до начала usability-тестирования. В основе таких целей могут быть показатели типа рассмотренных выше Completion, Satisfaction, Errors, а также удовлетворённости. Далее требуется найти процент участников, которые достигли этих целей. Причём цели могут различаться для различных заданий.

Показатели, основанные на сравнении с эталонами от экспертов, предполагают два одинаковых usability-исследования: первое – с пользователями, второе – с экспертами (специалистами в предметной области продукта, знакомыми досконально как с заданиями, так и с самим продуктом). Далее требуется

проанализировать, насколько результаты двух usability-исследований близки, имеют общие тенденции. Наиболее распространённый аспект для привлечения экспертов – время выполнения заданий, для многих остальных аспектов результат работы экспертов предполагается очевидным.

## **Задание на лабораторную работу**

Изучить методы оценки ранних концепций применительно к разработке пользовательского интерфейса приложения для предметной области, соответствующей варианту задания на лабораторную работу №1.

Выполнение лабораторной работы состоит из следующих шагов:

1. определить задачи тестирования, зафиксировать численные оценки качества разрабатываемого интерфейса;
2. провести тестирование прототипа с использованием различных методов тестирования;
3. проанализировать результаты тестирования, соотнести их с численными оценками качества продукта;

Выполнить следующие виды тестирований прототипа:

1. провести обратную карточную сортировку, оценив несколько экранов оффлайн;
2. провести обратную карточную сортировку с использованием TreeJack.
3. провести тестирование ожиданий;
4. исследовать прототип на соответствие контрольному списку (чеклисту) для мобильного приложений и по отдельному контрольному списку для веб-приложений, выбрав чеклист из примеров Примеры чеклистов для проверки приложения или разработав собственный на основе нескольких примеров чеклистов;
5. проанализировать интерфейс веб-приложения по модели GOMS;
6. проанализировать интерфейс мобильного приложения по модели KLM;
7. проанализировать интерфейс по показателям производительности «успех задания (task success)» и «время выполнения задания (time-on-task)»;
8. Выполнить исследование интерфейса с помощью анкеты SUS – System Usability Scale (<https://measuringu.com/sus/>, <https://usabilitygeek.com/how-to-use-the-system-usability-scale-sus-to-evaluate-the-usability-of-your-website/>);



9. сформировать рекомендации по дальнейшей модификации интерфейса.

Каждый из шагов необходимо выполнить для веб-приложения и мобильного приложения.

### ***Комментарии к выполнению usability-тестирования***

Начальный этап тестирования связан с разработкой прототипа интерфейса.

На этом этапе проектировщик использует имеющиеся результаты проектирования: общую схему приложения, планы отдельных экранных форм, глоссарий. Эти результаты сводятся воедино в общую схему, которую необходимо проверить по сформулированным ранее сценариям. Целью такой проверки является выявление несоответствие последовательности действий, описанной в сценарии, и структуры полной схемы. Обнаруженные несоответствия должны быть устранены за счет модификации экранных форм и/или корректировки общей схемы приложения.

Тестовые задания, которые в ходе проведения представляют собой задачи для пользователей, формируют исходя из указанных задач тестирования. Основой формулирования тестовых заданий являются пользовательские сценарии. Тестовое задание включает последовательность действий записанных в сценарии, но в отличие от него содержит конкретные значения данных, с которыми оперирует пользователь.

#### **Рассмотрим пример пользовательского сценария.**

Анна Сергеевна (смотри лабораторную работу No1) общаясь с клиентами по телефону, создает новые заказы. При формировании нового заказа, она выбирает клиента из списка, если его там нет, то вводит клиента в список клиентов. Затем добавляет в заказ необходимые товары, используя сложный (то есть составной) поиск. Она распечатывает информацию заказа, после этого сохраняет её.

В соответствии со сценарием необходимо имитировать ввод данных клиента при оформлении нового заказа. Имитировать ввод с клавиатуры для прототипа в виде презентации невозможно. Если же данные клиента вводятся из списка, то это достаточно просто. От слайда с формой нового (пустого) заказа организуют переход к слайду со списком клиентов. Перемещение по списку записей с данными клиентов имитируют последовательным переходом к слайдам с изображением списка со смещенными записями. После указания нужной записи организуют переход к слайду с формой нового заказа с заполненными данными клиента.

Задание атрибутов товара при организации поиска имитируют переходом к слайдам с изображениями списков значений атрибутов. Затем выполняют переход к слайду, отображающему результаты поиска. Если результат одиночный, то от этого слайда организуют возврат к форме с текущим (новым) заказом, если результаты множественные выбор нужной записи имитируют аналогично рассмотренному выше выбору записи из списка.

Для добавления следующего товара используются либо слайды с множественными результатами последнего поиска, либо слайды с изображением атрибутов товара для имитации нового поиска. Когда заданные товары будут добавлены в заказ, используют слайды с изображением выбора команд меню печати и сохранения заказа.

### **Тестовое задание может быть сформулировано в следующем виде.**

Создать новый заказ для клиента ООО «Регионторг». Ввести в заказ данные клиента из списка клиентов. Затем организовать поиск требуемого клиенту товара, определяя атрибуты товара, например: категория – выключатель, цвет – белый, страна-производитель – Польша. Используя результаты поиска, добавить в заказ сначала товар VIKO, затем товар VIKO-2. Организовать новый поиск товара с атрибутами: категория – распределительный щиток, цвет – белый, странапроизводитель – Германия. Добавить в заказ товар SIMENS. Распечатать заказ и сохранить его.

Проверка соответствия тестового задания и последовательности перехода между слайдами выполняется на готовом ролике. Выявленные несоответствия могут потребовать изменения навигационной системы приложения.

Когда сформулированы необходимые тестовые задания, выполнены проверки и требуемые коррективы, приступают к тестированию с привлечением пользователей из целевой аудитории.

В начале этого этапа формулируют задачи тестирования. Например, оценить производительность действий при использовании продукта. Тестирование проводится путем наблюдения за пользователем с фиксированием длительности выполнения действий. Критерий оценки можно сформулировать как выполнение контрольного тестового задания в течение 3 минут 75% тестируемых пользователей после тренинга-выполнения пяти различных сценариев. Тестирование проводится на представителях пользовательской аудитории, ранее не знакомых с

разрабатываемым продуктом. Уровень опытности тестируемых пользователей должен соответствовать уровню, определенному в профилях конечных пользователей. Считается, что тестирование на одном пользователе позволяет выявить примерно 60% ошибок. Поэтому число тестируемых пользователей, необходимых для проведения одного сеанса зависит от сложности и объема проектируемого продукта. Для «средних» приложений достаточно 4–8 человек. В ходе тестирования:

- категорически не рекомендуется прерывать или смущать пользователя;
- нельзя внушать тестируемому, что тестируют именно его;
- желательно присутствие разработчиков приложения (программистов), но их роль в тестировании исключительно пассивная.

Все результаты тестирования обобщаются с целью сформулировать рекомендации относительно модификации прототипа интерфейса. Модификации могут быть связаны с изменениями содержимого экранных форм, элементов навигационной системы, терминологии и даже функциональности тех или иных элементов интерфейса.

Рекомендуемые инструменты:

- калькулятор для вычисления доверительного интервала применительно к двоичному успеху, <http://www.measuringusability.com/wald.htm>;
- online-анкеты/опросы от GARY PERLMAN (QUIS, ASQ, CSUQ и др.), <http://www.acm.org/perlman/question.html>;
- Excel-таблица для вычисления Единого юзабилити-показателя (Single Usability Metric, SUM), <http://www.measuringusability.com/sum>.

### **Выполнение обратной карточной сортировки: шаги**

Процедуру обратной карточной сортировки следует выполнять в два этапа.

На первом этапе обратной карточной сортировки используется один из следующих трёх способов:

1. **Первый** – пользователю показывают меню верхнего уровня и просят показать, где он найдет тот или иной объект. Пользователь выбирает пункт меню, а ему в свою очередь показывают, что в нем содержалось, то есть раскрывают второй уровень. Пользователь ищет на втором уровне тот пункт, куда бы он пошел, а ему в ответ показывается третий уровень. Все это можно повторить при наличии всего лишь нескольких строк текста.

2. **Второй** способ – это карточный или графический. Вы распечатываете карту своего сайта или продукта, кладете её перед пользователем, даете ему стопку карточек с какими-то предметами и объектами. Далее пользователь размещает эту карточку на карте в том месте, где (по его мнению) он сможет найти объект, написанный на карточке.
3. **Третий** способ – это тестирование экранов. Пользователю показывается экран, где опции пронумерованы так, чтобы на вопросы пользователь однозначно называл номер, и не возникало путаницы. Вопросы могут быть разного характера, но в них есть одно общее: пользователя просят, чтобы он назвал номер раздела, в котором (по его мнению) находится требуемый элемент. Например, «Вам нужно узнать погоду на завтра. Где вы будете её искать?», «В каком разделе находятся неподписанные документы?».

На втором этапе обратной карточной сортировки проводится оценка результатов. Если сортировка проводилась по тексту или карточками графически – определяется процент попаданий. Чем больше пользователь «попал» в категории, тем лучше. Здесь нужно составить список категорий, которые оказались неуспешными. Выясняется причина неуспеха (например, неудачная терминология). Если сортировка проводилась на экранах, то оценивается количество успешных попыток.

## ***Тестирование информационной архитектуры с помощью treejack***

1. Изучите документацию инструмента Treejack от OptimalSort (<https://www.optimalworkshop.com/treejack>), доступную по адресу — <https://support.optimalworkshop.com/hc/en-us/categories/200144524-Treejack>.
2. Выполните тестирование информационной архитектуры приложения с помощью инструмента Treejack

## ***Выполнение тестирования ожиданий: шаги***

После обратной карточной сортировки проводится тестирование ожиданий. Содержательно здесь проводится тест на понимание пользователями процессов, которые происходят в системе. Этот тест выполняется в два этапа.

**Первый** этап – нужно собрать целевых пользователей и задать им вопросы относительно их представления о поведении элементов, то есть что они думают, о том, как будет себя вести то или иной элемент. Здесь совершенно неважно то, хорошо ли прорисован элемент графически или нет. Таким способом можно:

- тестировать названия кнопок – дать список глаголов и спросить: "Что Вы ожидаете?", "Что эта кнопка делает?" и т.п.;
- задавать вопросы пользователю об ожидаемом поведении того или иного элемента: "Что вы ожидаете увидеть, когда нажмете на кнопку/пункт меню?", "Что произойдет после нажатия на эту кнопку?" и т.п.

**Второй** этап тестирования ожиданий предполагает сравнение ожиданий пользователей с тем, что планировалось реализовать или уже реализовано. Здесь нужно выработать рекомендации, например, по названию кнопки. В крайнем случае – составить список "обманутых ожиданий".

### ***Исследование с помощью контрольного списка***

Используя пример списка в лабораторной работе, стандарты в области разработки веб-приложений и мобильных приложений, составить контрольные списки для быстрой оценки веб-приложения и мобильного приложения и провести оценку веб-приложения и мобильного приложения.

### ***Анализ интерфейса по модели GOMS и KLM***

Выполнить анализ по модели GOMS не менее 3 форм для различных методов ввода, когда навигация осуществляется по полям только с помощью мыши, с помощью клавиатуры и мыши (например, с использованием кнопки Tab). Желательно выполнить оценку для двух вариантов каждой формы, чтобы выбрать наиболее эффективный вариант. В качестве результата предоставить модели расчета для каждого варианта формы и метода ввода. В итоге должно быть предложено 12 моделей расчетов (3 формы \* 2 варианта каждой формы \* 2 методов ввода данных).

Выполнить анализ по модели KLM не менее 3 форм для мобильного приложения.

### ***Рекомендуемые инструменты для обработки результата***

Название	Описание	Адрес
Microsoft Visio (бесплатно по MSDNAA)	Инструмент для разработки различных схем и диаграмм, в том числе – макетов пользовательского интерфейса	

Сервис деловых диаграмм Cacroo.com	Сервис для разработки различных схем и диаграмм, в том числе – макетов пользовательского интерфейса	<a href="http://www.cacroo.com">http://www.cacroo.com</a>
Карточная сортировка Websort.net (до 10 респондентов - бесплатно)	Инструмент для проведения удалённой карточной сортировки	<a href="http://websort.net/">http://websort.net/</a>
Microsoft Expression Blend (бесплатно по MSDNAA)	Разработка интерфейса взаимодействия с пользователем для платформ MS .NET и Silverlight. Входит в состав пакета графических программ Microsoft Expression Studio.	
Axure	Средство проектирования прототипов пользовательского интерфейса	<a href="http://www.axure.com">http://www.axure.com</a>

### **Рекомендуемая структура отчёта по работе**

Отчёт должен содержать:

1. название, цель и задачи работы;
2. тестовые задания, сформированные в соответствии с пользовательскими сценариями;
3. показатели производительности;
4. показатели, основанные на аспектах для обсуждения usability;
5. анализ результатов обратной карточной сортировки;
6. анализ результатов исследования treejack  
(<https://www.optimalworkshop.com/treejack>)
3. результаты тестирования ожиданий;
4. контрольные списки и анализ результатов;
5. результаты анализа форм веб-приложения по модели GOMS;
6. Результаты анализа форм мобильного приложения по модели KLM;

7. результаты тестирования с выводом о соответствии критерию оценки качества;
8. общие рекомендации по модификации интерфейса;
9. общие выводы, сделанные в процессе выполнения лабораторной работы.

## **Рекомендуемая литература**

1. Купер А. Алан Купер об интерфейсе. Проектирование взаимодействия: пер. с англ. / А. Купер, Р. Рейман, Д. Кронин. СПб: Символ-Плюс, 2010. 688с., ил.
2. Сатин Д. Быстрые и дешёвые методы оценки пользовательских интерфейсов [Электронный ресурс] / Д. Сатин // 1С – Битрикс – CMS. Режим доступа: [www.1c-bitrix.ru/download/ppt/seminars/101209/satin\\_bitrix.pptx](http://www.1c-bitrix.ru/download/ppt/seminars/101209/satin_bitrix.pptx) (дата обращения: 25.10.2013).
3. Сатин Д. Быстрые методы оценки [Электронный ресурс] /Д. Сатин // [http://di.by/upload/iblock/563/fast\\_methods.pptx](http://di.by/upload/iblock/563/fast_methods.pptx) (дата обращения: 25.10.2013).
4. Tree testing [Электронный ресурс]. Режим доступа: [http://en.wikipedia.org/wiki/Tree\\_testing](http://en.wikipedia.org/wiki/Tree_testing) (дата обращения: 25.10.2013).
5. Раскин Д. Интерфейс: новые направления в проектировании компьютерных систем. - Пер. с англ. - СПб: Символ-Плюс, 2015. - 272 с., ил.
6. Tullis T. Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics / T. Tullis, A. William. Morgan Kaufmann Publishers, 2008. 336 p.
7. Сергеев С.Ф. Введение в проектирование интеллектуальных интерфейсов / С.Ф. Сергеев, П.И. Падерно, Н.А. Назаренко. СПб: СПбГУ ИТМО, 2011. 108 с.
8. Questionnaire for User Interaction Satisfaction [Электронный ресурс]. Режим доступа: <http://www.lap.umd.edu/QUIS/index.html> (дата обращения: 15.10.2016).
9. Computer System Usability Questionnaire [Электронный ресурс]. Режим доступа: <http://hcibib.org/perlman/question.cgi> (дата обращения: 15.10.2016).
10. Measuring Usability: Quantitative Usability, Statistics & Six Sigma by Jeff Sauro [Электронный ресурс]. Режим доступа: <http://www.measuringusability.com/SUM/index.htm> (дата обращения: 15.10.2016).
11. After-Scenario Questionnaire [Электронный ресурс]. Режим доступа: <http://hcibib.org/perlman/question.cgi?form=ASQ> (дата обращения: 15.10.2016).