

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий

(наименование института полностью)

Кафедра Прикладная математика и информатика

(наименование)

02.03.03 Математическое обеспечение и администрирование информационных систем

(код и наименование направления подготовки, специальности)

WEB-дизайн и мультимедиа

(направленность (профиль) / специализация)

## **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)**

на тему «Разработка алгоритма для оценки usability пользовательских интерфейсов»

Студент

М. Тургунбаев

(И.О. Фамилия)

(личная подпись)

Руководитель

канд. пед. наук, доцент, О.М. Гущина

(ученая степень, звание, И.О. Фамилия)

Консультант

М. В. Дайнеко

(ученая степень, звание, И.О. Фамилия)

Тольятти 2021

## Аннотация

Тема бакалаврской работы – «Разработка алгоритма для оценки usability пользовательских интерфейсов».

Целью бакалаврской работы является разработка алгоритма для оценки юзабилити пользовательских интерфейсов.

Объектом бакалаврской работы являются способы оценивания юзабилити пользовательских интерфейсов.

Предметом бакалаврской работы является алгоритм для оценки юзабилити пользовательских интерфейсов.

Актуальность бакалаврской работы заключается в том, что в связи с ростом пользовательских интерфейсов для компьютеров и смартфонов, возрастает проблема оценивания юзабилити пользовательских интерфейсов.

В первом разделе приводятся основные понятия юзабилити и юзабилити-тестирования, а также сравнительный анализ методов юзабилити-тестирования.

Во втором разделе описывается процесс разработки алгоритма для тестирования юзабилити пользовательских интерфейсов.

В третьем разделе описывается процесс тестирования разработанного алгоритма на различных пользовательских интерфейсах и оценка его эффективности.

В заключении описывается вывод по эффективности разработанного алгоритма, а также его применимости в различных условиях.

Бакалаврская работа состоит из пояснительной записки на 51 страниц, введения в две страницы, включая 30 рисунков, 4 таблиц, список из 30 источников

## **Abstract**

The title of the graduation work is *Development of an algorithm for evaluating the usability of user interfaces*.

The graduation work consists of an introduction, 30 figures, 4 tables, the list of 30 references including 7 foreign sources.

The aim of the research is to implement an algorithm for evaluating the usability of user interfaces.

The object of the research is the methods for evaluating the usability of user interfaces.

The subject of this investigation is the algorithm for evaluating the usability of user interfaces.

The relevance of the present graduation work is that, due to the growth of user interfaces for computers and smartphones, the problem of evaluating the usability of user interfaces has become urgent and pressing.

The first part of the graduation work gives some information about the basic concepts and notions of usability and usability testing, as well as carries out a comparative analysis of the usability testing methods.

The second part of the research deals with the process of developing an algorithm for testing and assessing the user interfaces usability.

The third part of the graduation work is devoted to testing the developed algorithm on various user interfaces and evaluating its effectiveness.

In conclusion, we present the obtained results related to assessing the effectiveness of the developed algorithm, as well as discuss its applicability under various condition

## Содержание

Введение.....	5
1 Современные подходы для оценивания юзабилити пользовательских интерфейсов .....	7
1.1 Определение понятия «юзабилити».....	7
1.2 Определение юзабилити-тестирования .....	8
1.3 Анализ методов юзабилити-тестирования .....	12
2 Обзор процесса разработки алгоритма для оценки юзабилити интерфейсов .....	23
2.1 Выбор архитектурного решения.....	23
2.2 Реализация диаграммы последовательности алгоритма.....	24
2.3 Реализация диаграммы деятельности алгоритма.....	25
2.4 Разработка алгоритма юзабилити.....	27
3 Тестирование разработанного алгоритма юзабилити .....	38
3.1 Тестирование разработанного алгоритма юзабилити на пользовательских интерфейсах .....	38
3.2 Оценка алгоритма юзабилити-тестирования .....	44
Заключение .....	47
Список используемой литературы .....	49

## Введение

С ростом использования компьютеров и смартфонов все больше внимания уделяется удобству использования или же «юзабилити». Сегодня практически все сайты и приложения используют интерфейс в виде графической оболочки (GUI) в качестве основного способа взаимодействия с пользователем. В связи с этим возникает проблема проектирования удобной и понятной графической оболочки для широкого использования различными группами людей. Для решения проблем проектирования GUI используют различные алгоритмы и методики тестирования интерфейсов, а сам способ тестирования юзабилити интерфейсов называется юзабилити-тестированием.

Актуальность бакалаврской работы заключается в том, что в связи с ростом пользовательских интерфейсов для компьютеров и смартфонов, возрастает проблема оценивания юзабилити пользовательских интерфейсов.

Объектом бакалаврской работы являются методы оценивания юзабилити пользовательских интерфейсов.

Предметом бакалаврской работы является алгоритм для оценки юзабилити пользовательских интерфейсов.

Целью бакалаврской работы является разработка алгоритма для оценки юзабилити пользовательских интерфейсов.

Для достижения цели необходимо выполнить следующие задачи:

- провести анализ методов юзабилити-тестирования и выбрать наиболее эффективный;
- разработать алгоритм тестирования юзабилити пользовательских интерфейсов;
- провести тестирования разработанного алгоритма на различных пользовательских интерфейсах;
- оценить эффективность реализованного алгоритма.

Данная бакалаврская работа предлагает решение проблем проектирования GUI и их оценки посредством создания алгоритма юзабилити-

тестирования, которая включает в себя тестирование веб и мобильных интерфейсов, а также их оценки посредством различных показателей и формирование отчета итогов тестирования.

Бакалаврская работа состоит из 3 разделов.

В первом разделе приводятся основные понятия юзабилити и юзабилити-тестирования. Кроме того, также проводится сравнительный анализ методов юзабилити-тестирования.

Во втором разделе описывается архитектура алгоритма тестирования юзабилити, разрабатывается диаграмма деятельности, диаграмма последовательности алгоритма и программная реализация самого алгоритма для тестирования юзабилити пользовательских интерфейсов.

В третьем разделе описывается процесс тестирования разработанного алгоритма на различных пользовательских интерфейсах, а также оценка его эффективности.

Бакалаврская работка состоит из пояснительной записки на 50 страниц, введения в две страницы, включая 30 рисунков, 4 таблиц, список из 30 источников.

# **1 Современные подходы для оценивания юзабилити пользовательских интерфейсов**

## **1.1 Определение понятия «юзабилити»**

С ростом и развитием компьютерных технологий стали широко распространяться интерфейсы, которые позволяют пользователю взаимодействовать с разными программами и приложениями для решения различных задач. В связи с этим появилась дисциплина под названием «юзабилити». Юзабилити означает, что та или иная вещь хорошо работает, если человек с небольшим опытом может ее использовать по назначению в независимости от того, является ли она веб-приложением, самолетом или дверью [26]. Это определение опирается на 4 основных положения [29]:

- юзабилити призывает сфокусироваться на пользователях;
- человек использует продукт для того, чтобы быть более производительным;
- пользователи — занятые люди, выполняющие определенные задачи;
- пользователи решают, когда изделие удобно, а когда нет.

Юзабилити — по большей части измеримая характеристика, которая присутствует в большей или меньшей степени практически во всех продуктах и описывает насколько эффективно в реальных обстоятельствах пользователь сможет взаимодействовать с продуктом, изделием или интерфейсом. Существует 5 признаков, по которым оценивается юзабилити [27]:

- обучаемость рассматривает, насколько пользователю легко обучиться взаимодействовать с продуктом, изделием или интерфейсом;
- эффективность показывает, сколько времени и усилий затратил пользователь;
- запоминаемость рассматривает, насколько легко пользователю будет взаимодействовать после продолжительного отсутствия;
- ошибки. Количество ошибок, которые совершили пользователи;

- удовлетворенность. Эмоциональное восприятие пользователя.

Если внедрить юзабилити в процесс создания продукта, то можно получить следующие выгоды:

- увеличивается производительность персонала и пользователей;
- увеличивается количество продаж и соответственно рост дохода;
- уменьшаются время и расходы на обучение персонала и пользователей;
- уменьшаются расходы на обслуживание;
- повышается удовлетворенность клиента.

Так как результаты часто измеримы и могут быть использованы в количественном анализе, то юзабилити обладает высокой ценностью, если сравнивать конкретные варианты макетов или дизайна, чтобы в дальнейшем выбрать наиболее эффективное решение.

Таким образом, можно сделать вывод что юзабилити необходимо для того, чтобы пользователь смог эффективно взаимодействовать с продуктом или интерфейсом для выполнения своих целей. Для оценки юзабилити используется понятие «юзабилити-тестирование».

## **1.2 Определение юзабилити-тестирования**

Юзабилити-тестирование — это исследование, которое выполняется с целью определить, насколько удобен некоторый объект, такой как продукт, изделие или интерфейс [30].

Существуют 4 аспекта продукта, для которых применение юзабилити-тестирования будет особенно полезно:

- наименования. Насколько понятны названия разделов, каталогов, заголовков и текст на кнопках;
- информационная архитектура. Насколько грамотно информация разделена на категории. Воспринимают ли пользователи информационные элементы так, как они были задуманы;



- первое взаимодействие и доступность. Насколько легко новые пользователи выполняют свои целевые действия. Нужны ли им инструкции;
- эффективность. Эффективно ли пользователи решают свои задачи, допускают ли ошибки и в каких количествах.

В процессе юзабилити-тестирования рассматривается, насколько хорошо пользователи выполняют целевые задачи и с какими проблемами при этом сталкиваются. Чтобы провести юзабилити-тестирование, необходимо, чтобы продукт, изделие или интерфейс были более-менее сформированными и имели внутреннюю согласованность. Смысл тестирования состоит в том, чтобы проверить качество в независимости от того, тестируется ли уже готовое программное обеспечение, минимальная версия продукта или же бумажный прототип. Исходя из этого можно сделать вывод, что юзабилити-тестирование должно проводиться на поздних стадиях проектирования, когда уже сформирована концепция продукта и есть возможность создать макет или прототип [11].

Юзабилити-тестирование наиболее эффективно, если применять его для тех пользователей, у которых еще не было опыта взаимодействия с продуктом, изделием или интерфейсом. Чем больше пользователь имеет опыт взаимодействия с продуктом, тем менее эффективно будет применение юзабилити-тестирования.

При проведении юзабилити-тестирования, также необходимо учесть, что тестируется то, что можно будет измерить количественно; что тестирование построено корректно; что итоговые результаты тестирования окажутся полезными для дальнейшего проектирования, и что есть необходимые ресурсы для исправления проблем, которые будут выявлены в ходе юзабилити-тестирования.

Важно учитывать, что юзабилити-тестирование по сути своей является средством для анализа, нежели синтеза. Оно не является заменой проектированию взаимодействия и источником идей для создания нового привлекательного продукта. Юзабилити-тестирование — метод оценки,

который предназначен для оценивания эффективности существующих идей и их последующего совершенствования [2].

Следует различать понятия «изучение пользовательской аудитории» и «юзабилити-тестирование», поскольку первое применяется до процесса генерации идей, а второе после.

Якоб Нильсен, доктор физических наук, основатель компании «Nielsen Norman Group» делит тестирование на 2 вида: полное и промежуточное.

Полное тестирование применяют тогда, когда необходимо сравнить 2 версии продукта, выявить проблемы перед этапом перепроектирования, изучить причины возвратов продукта или же когда необходимо определить причины запросов на обучение и поддержку [3]. Данный вид тестирования обычно проводится независимыми профессионалами и подробно документируется. Тестирование строится таким образом, чтобы была возможность получить количественные данные, которые затем можно будет проверить на статистическую значимость. Однако у полного тестирования есть свой недостаток: оно проводится в завершающей стадии разработки. На этом этапе уже поздно вносить значительные изменения в проект. Оценку пользовательского интерфейса гораздо лучше проводить до того, как начнется написание кода, чтобы потом было время внести изменения.

Промежуточное тестирование проводится на этапе проектирования. Данный вид тестирования позволяет проектировщикам увидеть, как целевая аудитория реагирует на информацию и инструменты, которые им предоставили для решения поставленных задач [14]. Это тестирование дает возможность проводить проверки юзабилити в процессе разработки, но у данного тестирования также есть свои недостатки: увеличение бюджета и времени, спад боевого настроения команды.

Есть много разных мнений относительно того, как проводить юзабилити-тестирование и интерпретировать результаты тестирования. Большинство подходов пытаются заменить методы инженерного проектирования, или же имеют перевес в сторону количественных данных.

Кэролин Снайдер в своей работе «Paper Prototyping» (Snyder, 2003) дает рекомендации по успешному проведению промежуточного тестирования [5]:

- проводить тестирование следует тогда, когда уже существует проектное решение, чтобы была возможность внести правки;
- тестировать только те задачи и опыт пользователя, которые связаны с конкретным продуктом;
- для набора участников из целевой группы использовать метод персонажей в качестве фильтра;
- ставить перед участниками ясные и понятные задачи, и просить их размышлять вслух;
- предоставлять участникам возможность напрямую взаимодействовать с прототипом (исключение составляет специализированное оборудование);
- контролировать процесс тестирования, чтобы была возможность выявить проблемы и причины их появления;
- не относиться предвзято, по возможности следует пригласить модератора, который ранее не принимал участия в проекте;
- сфокусироваться на поведении участников;
- в завершении тестирования проанализировать результаты, которые были получены в ходе тестирования;
- по возможности вовлекать в процесс тестирования проектировщиков.

Проектировщики — это те люди, которые принимают решения относительно этапа проектирования. Не каждый проектировщик способен модерировать процесс тестирования, сохраняя при этом нейтралитет и непредвзятость. Включение проектировщиков в этап планирования и наблюдения за сеансами тестирования может дать дополнительное преимущество в анализе и решении проблем [4].

Проектировщиков следует привлекать к следующим видам деятельности:

- планирование тестирования и формулирование вопросов для тестирования;
- разработка критериев отбора участников с использованием метода персонажей;
- использовать сценарии, чтобы выбрать тестовые задачи;
- наблюдение за процессом тестирования;
- анализ результатов тестирования.

Важно учитывать, что с помощью юзабилити-тестирования реально выявить только наиболее слабые места интерфейса, но почти невозможно обнаружить сильные, поскольку пользователи их не замечают и не смогут сказать [6, 7], какие улучшения необходимо предпринять.

При анализе данных, которые были получены в ходе тестирования, необходимо искать наиболее крупные и важные проблемы, от которых зависит исход успешного взаимодействия пользователя и продукта.

Юзабилити-тестирование проводится на всем этапе цикла разработки продукта. На ранних этапах разработки тестируются предыдущие версии продукта, а также продукты конкурентов [8]. В середине разработки продукта юзабилити-тестированием дается обратная связь, в каких местах необходимы улучшения. На финальных этапах тестирования проверяется, соответствует ли продукт тем требованиям, для которых он был спроектирован.

Юзабилити-тестирование включает в себя ряд различных юзабилити-методов, которые помогают оценить эффективность продукта или интерфейса.

### **1.3 Анализ методов юзабилити-тестирования**

Существуют большое количество методов юзабилити-тестирования, которые применяются на разных этапах проектирования и исследования интерфейсов. Весь процесс создания продукта можно поделить на 6 этапов. Для каждого этапа используются свои методы, но есть также методы, которые

могут использоваться на нескольких этапах. Использование методов юзабилити-тестирования на разных этапах показано в таблице 1.

Таблица 1 – Использование методов юзабилити-тестирования на разных этапах

Этап	Используемые методы
1. Планирование и оценивание	<ul style="list-style-type: none"> <li>– отбор участников для тестирования;</li> <li>– анализ контекста использования;</li> <li>– персонажи;</li> <li>– анализ конкурентов.</li> </ul>
2. Составление требований к проекту	<ul style="list-style-type: none"> <li>– опрос пользователей;</li> <li>– персонажи;</li> <li>– интервью с пользователями;</li> <li>– наблюдение за пользователями;</li> <li>– контекст использования.</li> </ul>
3. Дизайн и проектирование	<ul style="list-style-type: none"> <li>– руководство для создания дизайна;</li> <li>– оценка производительности;</li> <li>– бумажные прототипы;</li> <li>– экспертная оценка;</li> <li>– раскадровка;</li> <li>– шаблоны;</li> <li>– фокус-группы;</li> <li>– карточная сортировка;</li> <li>– оценивание прототипы.</li> </ul>
4. Реализация и программирование	<ul style="list-style-type: none"> <li>– правила по стилю;</li> <li>– быстрые прототипы.</li> </ul>
5. Тестирование и оценка	<ul style="list-style-type: none"> <li>– тестирование производительности;</li> <li>– эвристическая оценка;</li> <li>– анализ критических ситуаций.</li> </ul>
6. Этап выпуска	<ul style="list-style-type: none"> <li>– опросы пользователей;</li> <li>– тестирование производительности.</li> </ul>

Не все методы применяются с одинаковой частотой. Есть несколько методов, которые доказали свою эффективность и используется большинством проектировщиков. Далее они будут рассмотрены более подробно.

Анализ рабочих заданий — это метод юзабилити-тестирования, который включает в себя анкетирование или интервью для того, чтобы сформировать, как люди выполняют конкретные задания и с какими проблемами

сталкиваются [9, 10]. Вопросы для разных ситуаций могут различаться, но есть несколько аспектов, для которых требуется найти ответы в процессе анкетирования или интервью:

- какая цель стоит за заданием;
- частота и важность выполнения задания;
- сигналы или поводы для выполнения задания;
- действия, которые необходимо выполнить;
- решения, которые необходимо предпринять;
- информация, нужная для задания;
- ошибки;
- способы исправления ошибок.

После заполнения анкет и проведения интервью, анализируются данные и проводится декомпозиция. Это делается при помощи диаграммы потоков.

Сегментация пользовательской аудитории. Данный метод юзабилити-тестирования применяется для разбивки пользователей по схожим характеристикам. Делается это при помощи нахождения похожих характеристик и параметров для представителя каждой группы [11, 12]. Чтобы пользователи выглядели более конкретными, требуется прописать характеристики представителя каждой группы целевой аудитории, чтобы в дальнейшем сформировалась репрезентативная выборка, которая будет отражать свойства генеральной совокупности.

Персонажи — вымышленные модели пользователей, которые включают в себя подробное описание и характеристики выбранной сегментированной аудитории. Персонажи помогают проектировщикам и разработчикам не забывать в ходе реализации продукта, для кого они создают данный продукт. Для создания персонажей используются данные, которые были собраны в ходе исследования потребностей пользователей [15]. Для одного продукта, если он планируется для широкого круга аудитории, создаются несколько персонажей, которые могут быть совершенно не похожи друг на друга. Пример реализации персонажа показан на рисунке 1.

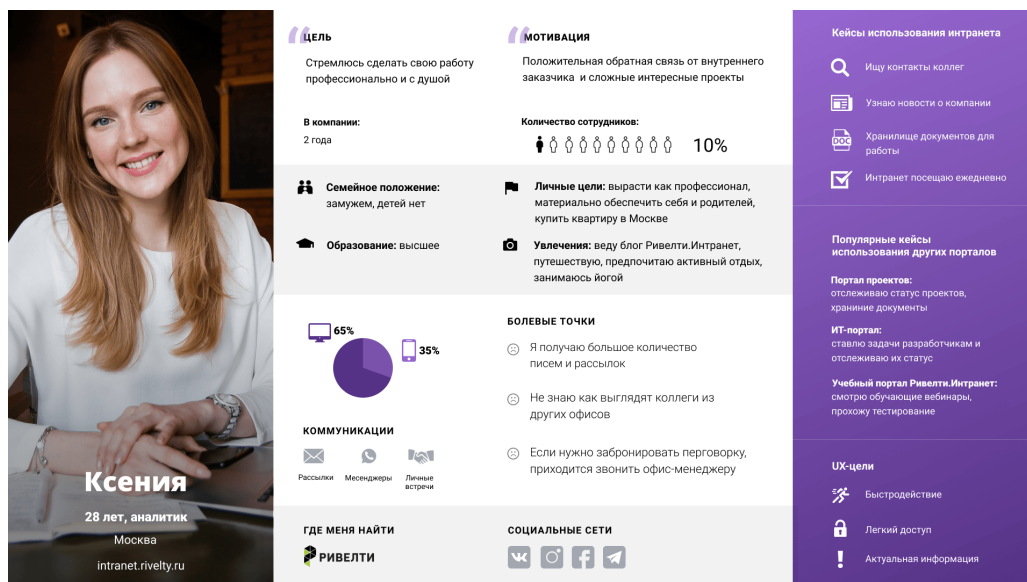


Рисунок 1 – Пример персонажа, созданный компанией Skillbox

Метод карточной сортировки. Данный метод юзабилити-тестирования используется для продумывания структуры и организации информации на сайте. Карточная сортировка помогает при составлении навигации, категорий или меню сайта [18]. Основной сложностью данного метода является анализ результатов, так как зачастую не всегда ясно почему те или иные элементы сгруппированы в одну категорию, поэтому иногда проводятся собеседования с пользователями, и выявляются принципы группировки карточек у пользователей [15]. Пример реализации карточной сортировки показан на рисунке 2.

Экспертная оценка. В данном методе юзабилити-тестирования экспертами и профессиональными дизайнерами исследуются продукт и выявляются в ходе тестирования проблемы, которые имеются у продукта. Экспертная оценка не может полностью заменить полноценное юзабилити-тестирование [13, 20]. Существует разные вариации для реализации данного метода, но наибольшую популярность обрел метод эвристической оценки, который будет рассматриваться более подробно.

Какие торты вы любите?

Добавьте описание

Карточки	Категории
Чизкейки <span style="float: right;">🗑</span>	Праздничные <span style="float: right;">🗑</span>
Песочные <span style="float: right;">🗑</span>	Будничные <span style="float: right;">🗑</span>
Слоеные <span style="float: right;">🗑</span>	Добавить категорию
Пироги <span style="float: right;">🗑</span>	
Добавить карточку	

Сохранить изменения
Отменить
Предпросмотр

**Смена типа вопроса**  
Карт. сортировка

**Индекс вопроса**  
1

**Размер изображения**  
Обычный

**Текст кнопки "Далее"**  
Далее

**Параметры**

Обязательный

Показ медиа

Рисунок 2 – Пример реализации карточной сортировки компанией  
Mail.ru

Эвристическая оценка. В данном методе юзабилити-тестирования специалистами исследуется продукт, базируясь на списке определенных юзабилити-принципов. В ходе тестирования выявляются недостатки и затем они классифицируются как нарушение одного или нескольких юзабилити-принципов [16]. Список эвристик специалистами определяется заранее. Наиболее распространенным является набор из 10 правил, которые были предложены Якобом Нильсеном:

- видимость статуса системы. Пользователю всегда необходимо знать, что происходит в данный момент времени, получая обратную связь;
- соответствие между системой и реальным миром. Система должна говорить на языке, понятном для пользователя, используя знакомые ему термины и концепции, а не «системно-ориентированный» язык;
- управляемость и свобода для пользователя. Пользователи часто совершают действия по ошибке и им нужен ясно видимый «аварийный выход»



из не желаемого состояния системы, без необходимости проходить длительный процесс;

- согласованность и стандарты. Пользователи не должны задаваться вопросом, означают ли одно и то же разные слова, ситуации или операции. Необходимо следовать соглашениям, принятым для конкретно-выбранной платформы;

- предотвращение ошибок. Продуманный дизайн, который не позволяет какой-то проблеме даже возникнуть, лучше, чем самые хорошие сообщения об ошибках. Следует устранять сами условия возникновения ошибок, либо выявлять их и предупреждать пользователя о предстоящей проблеме;

- распознавать лучше, чем вспоминать. Минимизируйте нагрузку на память пользователя, явно показывая ему объекты, действия и варианты выбора. Пользователь не должен в одной части диалога запоминать информацию, которая потребуется ему в другой. Инструкции по использованию системы должны быть видимы или легко получаемы везде, где возможно;

- гибкость и эффективность использования. Акселераторы (средства быстрого выполнения команд), которые новичок даже не видит, для опытного пользователя часто могут ускорить взаимодействие. Поэтому система должна удовлетворять как неопытных, так и опытных пользователей. Следует давать возможность настраивать под себя часто используемые операции;

- эстетический и минималистический дизайн. В интерфейсе не должно быть информации, которая не нужна пользователю или которая может понадобиться ему в редких случаях. Каждый избыточный элемент диалога отнимает внимание от нужных элементов;

- помочь пользователю понять и исправить ошибку. Сообщения об ошибках следует писать простым языком, без кодов, четко формулируя проблему и предлагая конструктивное решение.

– справка и документация. Хотя было бы лучше, если бы система была пригодна к использованию без документации, всё же необходимо предоставлять справку и документацию. Информация должна быть простой в поиске, соответствовать задаче пользователя, описывать конкретную последовательность действий, и не должна быть слишком большой.

Наибольший эффект будет достигнут, если оцениваться продукт будет не одним специалистом, а как минимум тремя специалистами.

Прототипирование. Данный метод юзабилити-тестирования используется для минимизации рисков на ранних этапах создания продукта [18]. Бывают 2 основных вида прототипа: горизонтальные и вертикальные. В вертикальном прототипе сокращается количество функций до необходимого минимума, и они полностью работоспособны. В горизонтальном прототипе обозначаются все функции продукта, но при этом они не реализуются, тестируется только дизайн интерфейса.

Также прототипы классифицируются по степени детализации: прототипы высокой точности (high-fidelity), показана на рисунке 3 и низкой (low-fidelity), показан на рисунке 4.

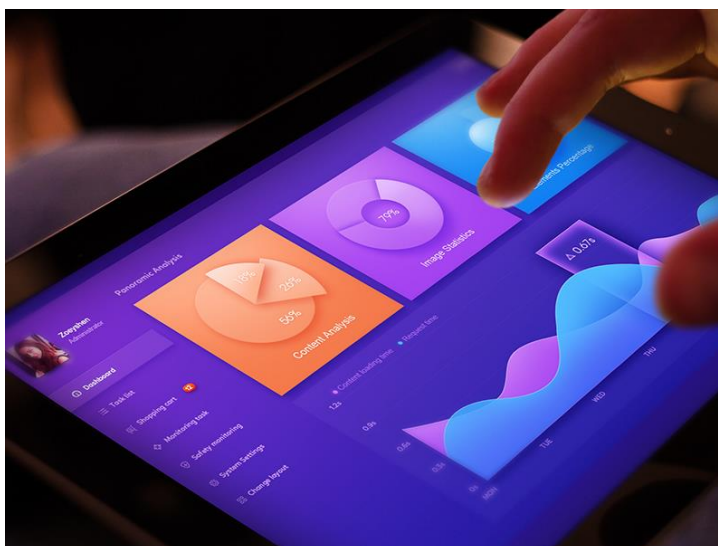


Рисунок 3 – Прототип высокой точности (high-fidelity)

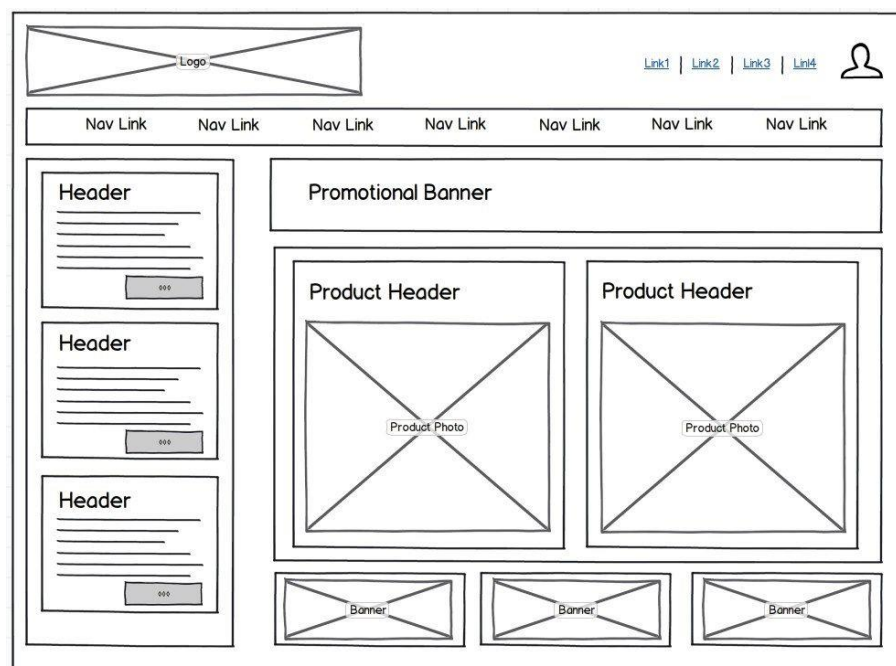


Рисунок 4 – Прототип низкой точности (low-fidelity)

Прототипы высокой точности выглядят и ведут себя точно так же, как и конечный продукт, но при этом программно они никак не реализовываются. Прототипы низкой точности лишь схематично отображают интерфейс продукта, но при этом их изменение делается гораздо проще и быстрее, чем у прототипов высокой точности [22].

Окулография (eye-tracking). В данном методе юзабилити-тестирования измеряется положения и анализируются движения глаз. Для проведения данного тестирования, требуется специальное оборудование: айтрекер. С помощью него анализируются точки взгляда с применением датчика изображения и рассчитывается точка взгляда по специальному алгоритму, и после выводится на экран тепловая карта с отображением областей, на которые смотрел пользователь [24]. Окулография применяется, когда необходимо найти проблемные места интерфейса, изучить паттерны поведения пользователей и оценить привлекательность разных дизайнов интерфейса. Из недостатков можно отметить, что стоимость специализированного айтрекера достаточно высока и анализ, и интерпретация

результатов требует большого количество времени. Также необходима высокая квалификация специалиста для того, чтобы он смог качественно и точно интерпретировать полученные данные. Пример реализации окулографии показан на рисунке 5.



Рисунок 5 – Пример реализации окулографии (айтрекинга)

Оценка производительности. В данном методе юзабилити-тестирования пользователями выполняется определенный набор заданий [1]. В это время, пока проходит тестирование, замеряется время, которое потребовалось пользователям для решения, количество ошибок, которые были совершены, частота обращения, число посещенных страниц и т.д. При проведении данного юзабилити-тестирования специалист объясняет, что необходимо сделать респонденту и после этого наблюдает за действиями респондента [4]. По итогу тестирования накапливается много количественных данных, которые анализируются методами статистического анализа и выявляются проблемы. Из недостатков следует отметить, что количественные данные не дадут ответа, в чем именно заключается проблема, поэтому иногда в качестве дополнения проводятся серии интервью для сбора качественных данных. Пример

реализации юзабилити-метода оценки производительности показан на рисунке 6.

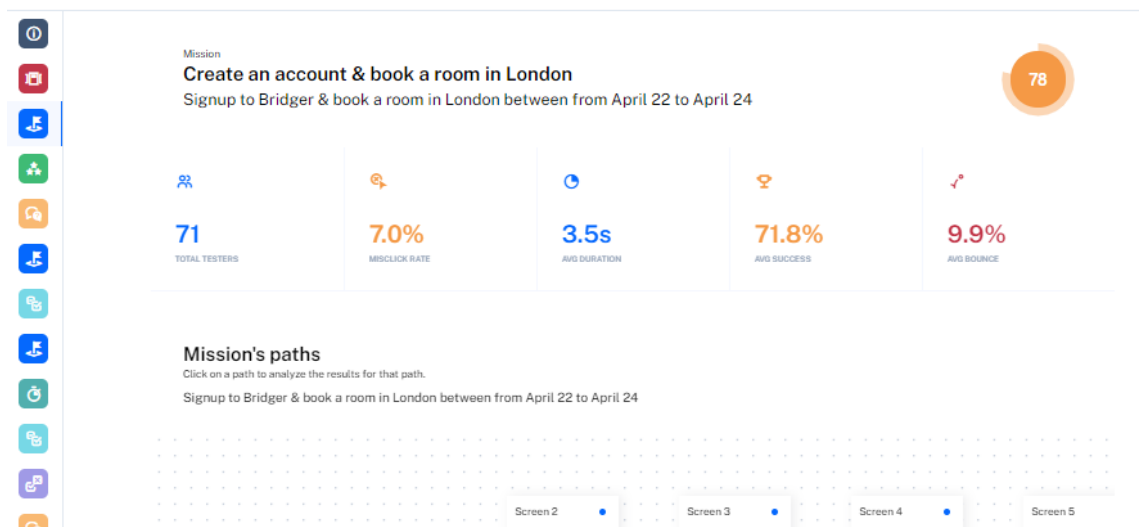


Рисунок 6 – Реализация метода оценки производительности компанией Maze

Для сравнения методов юзабилити-тестирования была построена таблица 2.

Проанализировав наиболее популярные методы юзабилити-тестирования, можно сделать вывод, что не все методы можно реализовать через программный код и автоматизировать анализ результатов [25]. Кроме того, некоторые методы имеют специфическое применение, такие как карточная сортировка, так как данный метод подходит для проектирования иерархии навигаций и категорий, и не все интерфейсы содержат в себе большое количество навигаций и категорий.

Исходя из этого, можно сказать, что метод оценки производительности смотрится наиболее выигрышным, так как способен собрать большой объем количественных данных, его можно реализовать через программный код и при этом автоматизировать анализ данных.

Таблица 2 – Анализ методов юзабилити тестирования

Метод	Преимущества	Недостатки	Программная реализация	Автоматизация анализа результатов
Персоны	Помогают сфокусироваться на проблемах пользователя	Если персона составлена неверно, то цена ошибки продукта высока	Частично	Нет
Оценка производительности	Большой объем количественных данных, которые могут улучшить продукт	Нехватка качественных данных, которые показывают причины проблем	Да	Легко. Есть возможность автоматизации анализа
Системы ай-трекинга	Большой объем количественных данных и их визуализация	Для расшифровки данных требуется экспертная оценка, чтобы получить качественные данные	Да	Сложно. Частичная автоматизация анализа
Прототипирование	Возможность на ранних стадиях выявить ошибки	Риск потери времени на создание прототипа	Да	Сложно. Нет автоматизации анализа
Экспертная оценка	Качественный анализ и выявление проблем	Весь результат зависит от профессионализма специалиста	Нет	Зависит от квалификации специалиста. Нет автоматизации анализа
Карточная сортировка	Организация информации в категории, меню	Не подходит для продуктов, где нет большого количества информации	Да	Сложно. Нет автоматизации анализа
Интервью	Качественные данные, которые помогают выявить глубинные причины	Отсутствие количественных данных, которые могут подтвердить достоверность	Нет	Сложно. Нет автоматизации анализа

В данном разделе были рассмотрены определения юзабилити и юзабилити-тестирования и также анализ методов юзабилити-тестирования. В следующем разделе данной работы будет рассмотрен процесс создания метода оценки производительности в формате веб-приложения.

## 2 Обзор процесса разработки алгоритма для оценки юзабилити интерфейсов

### 2.1 Выбор архитектурного решения

Для того, чтобы применение метода производительности было наиболее эффективным, его следует разрабатывать в виде веб-приложения, где будет возможность размещать и выполнять различные задания.

В качестве архитектурного решения можно рассматривать несколько паттернов проектирования [23], которые можно использовать для построения веб-приложения.

Под паттерном проектирования понимается не какой-то конкретный код, а концепция решения различных проблем. Наиболее распространенные виды паттернов [11], это:

- model-view-controller (MVC);
- model-view-presenter (MVP);
- model-view-view model (MVVP).

Шаблон проектирования MVC состоит из 3 частей: модель, представление и контроллер. Модель (Model) дает необходимые данные и реагирует на команды контроллера, изменяя при этом своё состояние. Представление (View) отображает данные модели пользователю и реагирует на изменения модели. Контроллер (Controller) интерпретирует действия пользователя, и оповещает модель, если необходимы изменения. Шаблон проектирования MVC для отделения бизнес-логики от визуализации, за счет этого повышается возможность повторно использовать код. На рисунке 7 показана схема шаблона проектирования MVC,

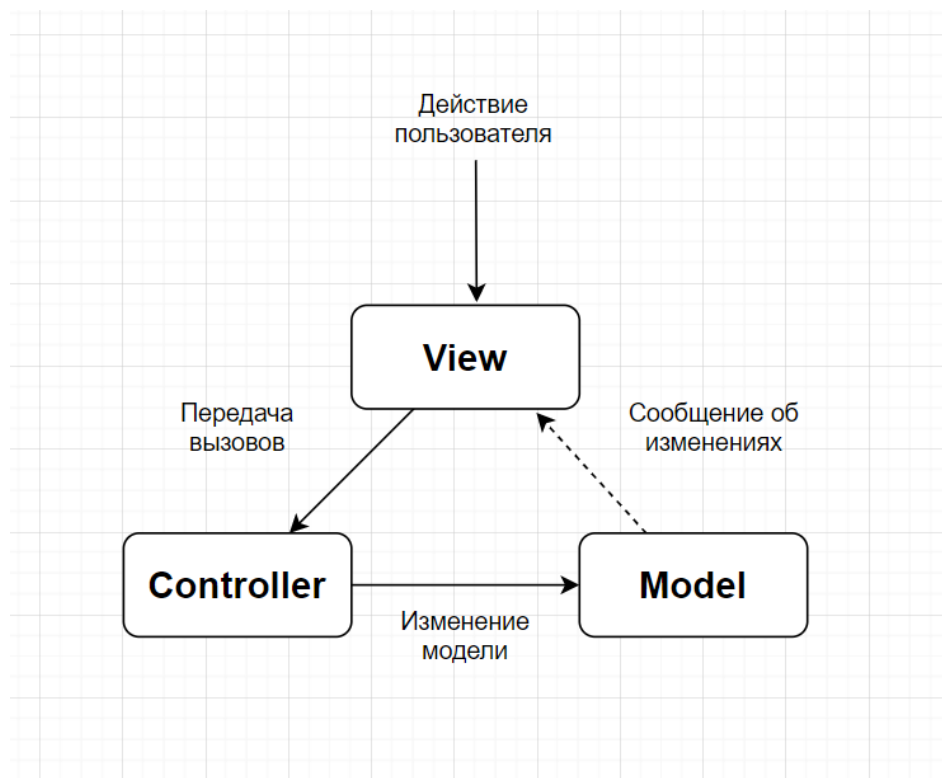


Рисунок 7 – Схема шаблона проектирования MVC

На данный момент шаблон проектирования MVC является наиболее популярным и используемым и включает в себя принципы других архитектурных решений, он и будет взят за основу для создания веб-приложения.

## 2.2 Реализация диаграммы последовательности алгоритма

Для понимания жизненного цикла алгоритма юзабилити-тестирования и взаимодействия юзабилити-специалистов и пользователей в течении времени требуется визуализация данного процесса. Наиболее лучшим решением является отображение данного процесса в виде диаграммы последовательности.

Диаграмма последовательности представляет из себя UML диаграмму [17], на которой для нескольких объектов на временной вертикальной оси показывается жизненный цикл объекта, а также взаимодействие актеров. На рисунке 8 показана реализация диаграммы последовательности.





Рисунок 8 – Диаграмма последовательности

Юзабилити-специалист создает проект с названием, описанием, и целями, загружает прототип интерфейса и отправляет его на сервер. Пользователь видит опубликованный проект, заходит в него и начинает выполнять задания, после чего результаты отправляются на сервер. В свою очередь сервер предоставляет проект с результатами тестирования юзабилити-специалисту.

### 2.3 Реализация диаграммы деятельности алгоритма

Для понимания последовательности действий юзабилити-специалиста, пользователя, а также роль работы сервера, требуется визуализировать данный процесс в виде диаграммы. Наиболее оптимальным видом диаграммы для данного процесса является диаграмма деятельности.

Диаграмма деятельности представляет из себя UML диаграмму, которая используется для моделирования бизнес-процессов, технологических процессов, последовательных и параллельных вычислений. На рисунке 9 показана реализация диаграммы деятельности.

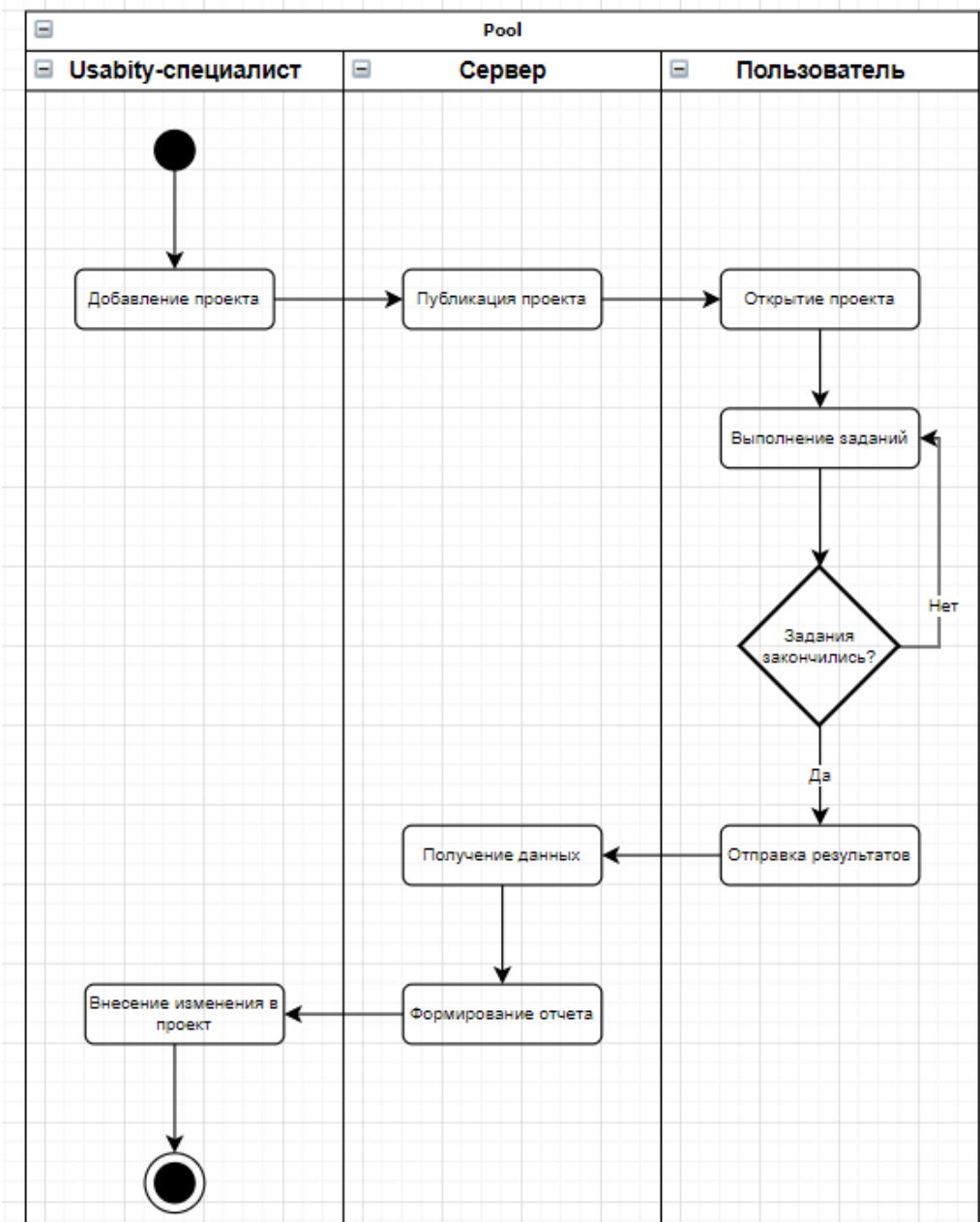


Рисунок 9 – Диаграмма деятельности

Юзабилити-специалист добавляет проект с описанием, целями и прототипом и публикует его на сервере. Пользователь видит опубликованный проект, открывает его и начинает выполнять различные задания, если они закончились, то результаты тестирования отправляются на сервер, где происходит получение данных и формирование отчета [28]. После этого юзабилити-специалист вносит изменения в проект согласно результатам тестирования.

## 2.4 Разработка алгоритма юзабилити

Для разработки алгоритма юзабилити будет использоваться язык программирования JavaScript, а для работы с бэкендом язык программирования PHP. Это позволит реализовать алгоритм юзабилити в виде веб-приложения [21], что упростит взаимодействие юзабилити-специалистов и пользователей, которые будут тестировать интерфейсы.

Весь процесс разработки можно разбить на несколько этапов:

- процесс загрузки прототипа;
- процесс добавления текстового описания;
- процесс выполнения заданий;
- процесс формирования результатов тестирования.

Для того чтобы данный метод работал эффективно, необходимо иметь уже готовый прототип низкой или высокой детализации. Формирование прототипа может быть реализовано разными путями:

- ручное создание прототипа;
- создание прототипа в специализированной программе.

Ручное создание прототипа предполагает, что имеются готовые изображения интерфейса. Они загружаются на веб-сервис для создания прототипов, например Invision, и после этого происходит ручная связка всех изображений с целью формирования прототипа. Данный подход отнимает много времени и трудозатрат, что является не самым эффективным решением.

Создание прототипа в специализированной программе, таких как Figma, Sketch, Adobe XD предполагает, что исходные макеты интерфейсов связываются в прототип посредством их линкования и обладают возможностью имитации работы программы [19]. Данный подход обрел широкую популярность благодаря низким трудозатратам и времени.

Исходя из этого следует, что использование готовых прототипов, созданных в специализированных программах гораздо выгоднее, чем создание прототипов вручную.

Для того, чтобы была возможность использовать готовые прототипы, созданных в специализированных программах существует технология API (application programming interface), которая позволяет взаимодействовать одной программе с другой.

В настоящее время Figma обладает наибольшей популярностью использования среди проектировщиков и дизайнеров. Преимуществом данной специализированной программы является то, что она работает в формате веб-приложения, что значительно упрощает работу с ней [25]. Кроме того, API Figma обладает широкими возможностями и удобством для использования в своих проектах. Поэтому API данной программы будет выбран в качестве предоставления готовых прототипов.

API Figma базируется на REST (Representational State Transfer) архитектуре. REST представляет из себя согласованный набор определенных ограничений, которые учитываются при проектировании распределенной гипермедиа-системы. Одной из главных особенностей REST является то, что данная архитектура позволяет наилучшим образом использовать протокол HTTP [18]. Данный протокол используется для того, чтобы определить структуру запросов и ответов браузера.

Кроме того, API Figma поддерживает аутентификацию с помощью токенов доступа и OAuth2. Запросы выполняются через HTTP Endpoints, которые позволяют запрашивать файлы, изображения, версии файлов и комментариев. В данной работе будет рассмотрена аутентификация с помощью OAuth2.

OAuth2 — это протокол веб-безопасности, который позволяет предоставлять третьей стороне ограниченный доступ к защищённым ресурсам пользователя без необходимости передавать ей (третьей стороне) логин и пароль [19].

Для реализации подключения API используются GET и POST запросы. Реализация GET запроса показана на рисунке 10

```

class Api extends Model
{
    public function getApi($prototip) {
        $opts = array(
            'http' => array(
                'method' => "GET",
                'header' => "Accept-language: en\r\n" .
                    "Cookie: foo=bar\r\n" .
                    "X-Figma-Token: 191099-10e30a83-fcfd-4e69-8b33-37c9dca5b849\r\n"//токен
            )
        )
    }
}

```

Рисунок 10 – Реализация GET запроса

Функция `getApi` состоит из HTTP заголовка, который содержит метод GET, заголовка, где содержится уникальный access token (токен доступа).

На рисунке 11 в переменной `context` открывается файл с помощью установленных выше HTTP-заголовков, переменная `auth` содержит формат для принятия ссылки прототипа, и далее результат возвращается во `View`.

```

    $context = stream_context_create($opts);
    $auth = file_get_contents('https://api.figma.com/v1/files/'.$prototip.', false, $context);
    $result = json_decode($auth);
    return $result;
}
}

```

Рисунок 11 – Открытие файла и возвращение во View

Далее идет реализация POST запроса, который показан на рисунке 12. Функция `prototip` содержит условие получения id прототипа из POST запроса, для дефолтного значения присваивается id тестового прототипа, после этого идет реализация получения результата и отображения его во `View`.

В проекте, для того чтобы получить доступ к тестированию и созданию проектов, необходимо будет иметь собственный логин и пароль. На рисунке 13 показана реализация системы авторизации.

```

use Illuminate\Http\Request;
use App\Api;
class IndexController extends Controller
{
    public function prototip(Request $request)
    {
        if(!empty($request->prototip)) {
            $prototip = $request->prototip;
        }
        else {
            $prototip = 'VSghq0IDXknItxXggawX8K';
        }
        $model = new Api();
        $result = $model->getApi($prototip);
        return view('index', ['result' => $result]);
    }
}

```

Рисунок 12 – Реализация POST запроса

```

if (config.auth) {
    var username = config.auth.username || '';
    var password = config.auth.password || '';
    requestHeaders.Authorization = 'Basic ' + btoa(username + ':' + password);
}

```

Рисунок 13 – Реализация системы авторизации

После того, как система аутентификации и логика подключения прототипов была настроена, необходимо спроектировать пользовательский интерфейс для того, чтобы возможность создавать проекты и тестировать их.

Чтобы создать проект, необходимо в строку ввода ввести ссылку на прототип, который планируется протестировать. Данный процесс изображен на рисунке 14.



## Создание проекта

Вставьте ссылку на прототип

Ссылка на прототип Figma

<https://www.figma.com/proto2fffg222>

Создать

Рисунок 14 – Создание проекта

Далее необходимо описать детали проекта: название проекта, цель проекта, контекст использования, рисунок 15.



## Опишите детали проекта

Название проекта

Интернет-магазин покупки сумок

Цель проекта

Найти и купить сумку Berkely

Например: Купить автомобиль Mazda

Контекст использования

У вас потерялась сумка, нужно купить новую сумку

Поможет пользователям лучше понять для чего нужен ваш продукт

Продолжить

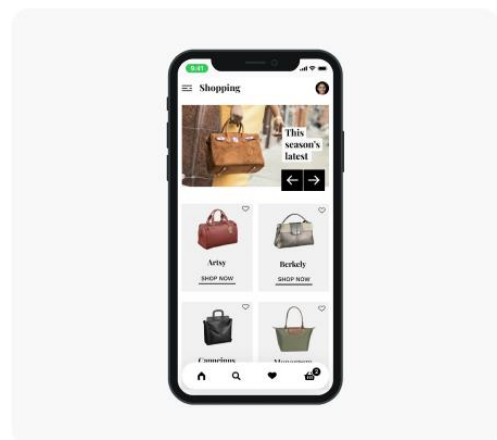


Рисунок 15 – Описание деталей проекта

Также можно добавить свои дополнительные вопросы, чтобы собрать больше количественных данных, которые помогут принять более

качественные решения. Процесс создания пользовательских вопросов и ответов показан на рисунке 16.

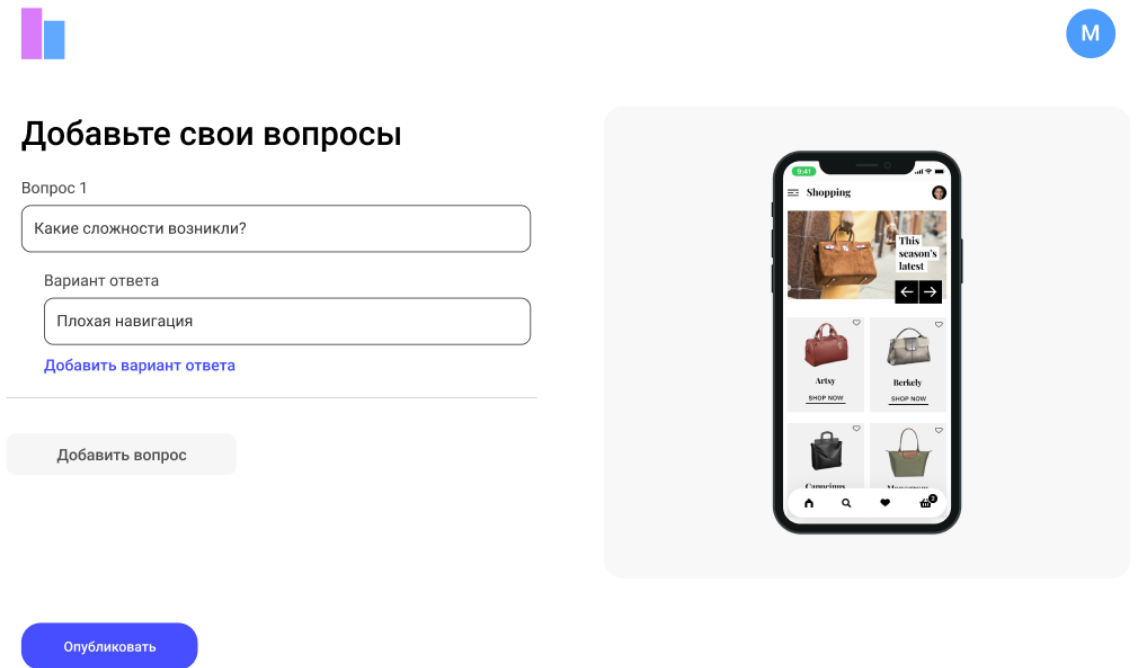


Рисунок 16 – Создание пользовательских вопросов и ответов

После того, как проект будет полностью сформирован и описан, его можно опубликовать, для этого необходимо нажать кнопку «Опубликовать», как показано на рисунке 17.



### Проект опубликован!

Ваш проект увидят пользователи и выполнят задания,  
проект доступен в каталоге проектов

Все проекты

Рисунок 17 – Публикация проекта

Алгоритм юзабилити-тестирования состоит из 3 этапов:

- чтение задания и взаимодействие с прототипом Figma;



- оценивание по 5 бальной шкале различных аспектов;
- выбор ответов на определенные вопросы.

На каждом этапе алгоритма производится ряд определенных операций, который в конечном итоге формирует отчет о результате тестирования. На рисунке 18 представлен алгоритм юзабилити-тестирования.



Рисунок 18 – Алгоритм юзабилити-тестирования

Рассмотрим каждый этап более подробно. На первом этапе происходит сбор следующих данных:

- количество пользователей;
- процент успешно-завершенных попыток;
- среднее время, затраченное на одного пользователя;
- процент ошибочных нажатий.

Количество пользователей. Данный показатель показывает, сколько всего пользователей зашли на данный проект.

Процент успешно-завершенных попыток. Этот показатель рассчитывается из соотношения пользователей, которые выполнили все задания к количеству пользователей, которые зашли на данный проект.

Формула (1) для расчета представлена ниже:

$$\frac{AmountofSuccesUser}{AmountofProjectUser} * 100\% \quad (1)$$

Среднее время, затраченное на одного пользователя. Данный показатель сформировывается, исходя из соотношения общего затраченного времени всех пользователей на количество пользователей. Формула (2) для вычисления базируется на формуле среднего арифметического и представлена ниже.

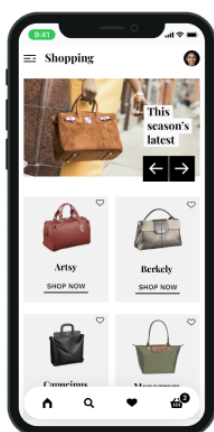
$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad (2)$$

где  $n$  — общее количество значений  $x$ .

Процент ошибочных нажатий или мискликов формируется из соотношения ошибочных нажатий всех пользователей на количество нажатий всех пользователей определенного проекта и умноженное на 100%. Формула (3) для расчета представлена ниже.

$$\frac{TotalMisclicks}{TotalClicks} * 100\% \quad (3)$$

Реализация 1 этапа тестирования со стороны пользователя показана на рисунке 19.



#### Цель

Найти и купить сумку Berkely

#### Контекст

У вас потерялась сумка, нужно купить новую сумку

Рисунок 19 – Реализация 1 этапа тестирования

Данные 4 показателя основываются на данных, полученных в процессе взаимодействия пользователя с прототипом. Но этих данных недостаточно, чтобы сделать выводы о дальнейшем улучшении юзабилити, поэтому необходимо также собрать и дополнительные данные, к которым относятся:

- удобство использования;
- типографика;
- цвета;
- понятность текста.

Удобство использования показывает, насколько удобно было выполнять задания и взаимодействовать с прототипом. Типографика показывает, насколько удачно подобраны шрифтовые гарнитуры. Цвета показывают, насколько правильно подобраны цвета для данного проекта. Понятность

текста означает, поняли ли пользователи текста интерфейсов и насколько хорошо.

Все эти 4 показателя рассчитываются по формуле среднего арифметического в диапазоне от 1 до 5 баллов, как показано на рисунке 20.

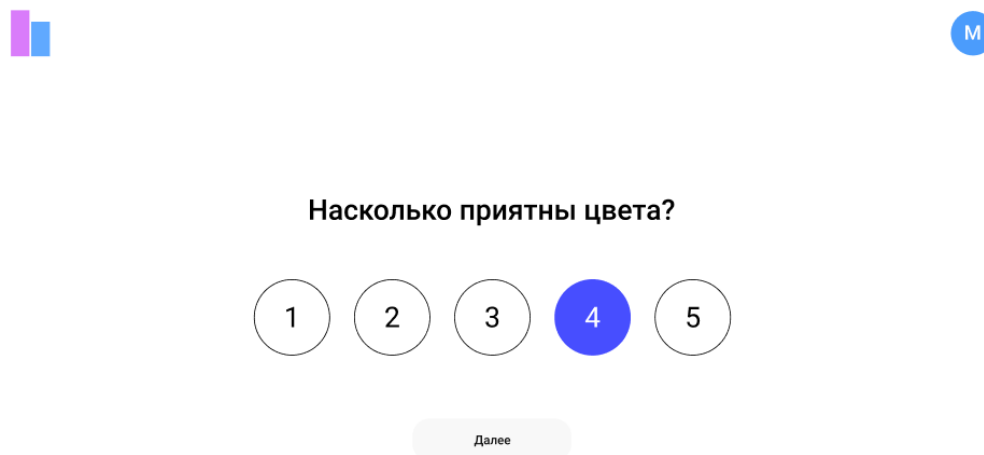


Рисунок 20 – Реализация 2 этапа тестирования

3 Этап состоит из произвольного выбора ответов на вопросы. Пользователям предлагается выбрать 1 или несколько ответов на определенный вопрос. После этого формируется гистограмма, которая визуализирует результаты 3 этапа, которая показана на рисунке 21.

### Какие сложности возникли?

12 из 20 ответили



Рисунок 21 – Реализация 3 этапа тестирования

Гистограмма показывает, насколько часто встречаются те или иные значения. Каждый столбец гистограммы показывает частоту попадания значения выборки в интервал значений, то есть чем выше столбец, тем гораздо вероятнее действительное значение показателя.

Частота попадания в каждый интервал считается по следующей формуле:

$$p_j = \frac{\Delta n_j}{N} \quad (4)$$

где  $\Delta n_j$  — количество попаданий в  $j$  интервал,

$N$  — объем выборки.

После того, как была найдена частота попадания в каждый интервал, можно найти высоту столбца гистограммы, которая определяется по следующей формуле:

$$h_i = \frac{P_i}{\Delta x_i} \quad (5)$$

где  $\Delta x_i$  — длина одного шага.

Все 3 этапа формируют единый алгоритм для оценки юзабилити пользовательских интерфейсов, что дает возможность качественно провести тестирование и собрать данные для дальнейшего улучшения интерфейса.

В данном разделе была рассмотрена реализация алгоритма юзабилити, а именно создание метода оценки производительности. В следующем разделе будет рассмотрено тестирование данного алгоритма на нескольких интерфейсах и анализ результатов тестирования.

### 3 Тестирование разработанного алгоритма юзабилити

#### 3.1 Тестирование разработанного алгоритма юзабилити на пользовательских интерфейсах

Для того, чтобы оценить эффективность данного алгоритма, необходимо протестировать его на различных пользовательских интерфейсах. Для этого будут взяты 2 разных интерфейса: один интерфейс будет представлять из себя часть функционала мобильного приложения, а другой интерфейс — часть функционала веб-сайта.

Сначала будет взят интерфейс мобильного приложения для редактирования фото и видео в виде прототипа. Интерфейс данного мобильного приложения (рисунок 22) состоит из 11 различных экранов, которые отражают различные состояния мобильного приложения.

Синие линии между экранами означают взаимосвязь, то есть при нажатии на определенный элемент откроется тот экран, с которым была соединена синяя линия.

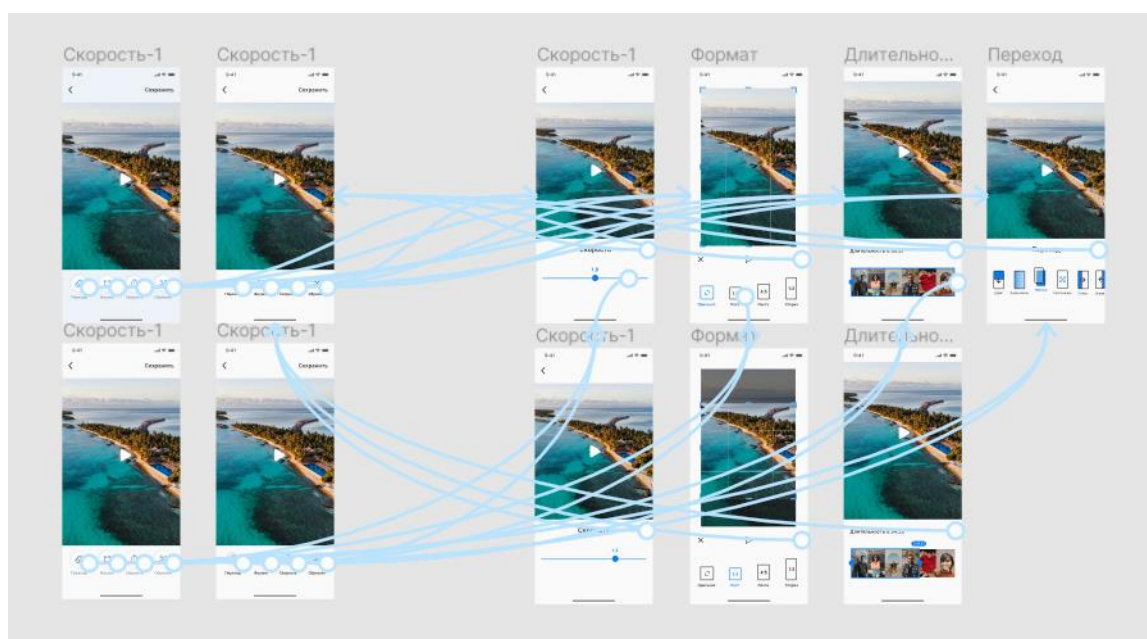


Рисунок 22 – Прототип мобильного приложения

После того как был выбран прототип, необходимо скопировать URL данного прототипа, сделать это можно прямо в самой программе Figma, нажав на кнопку Copy link, после чего ссылка будет скопирована в буфер обмена. Данный процесс показан на рисунке 23.

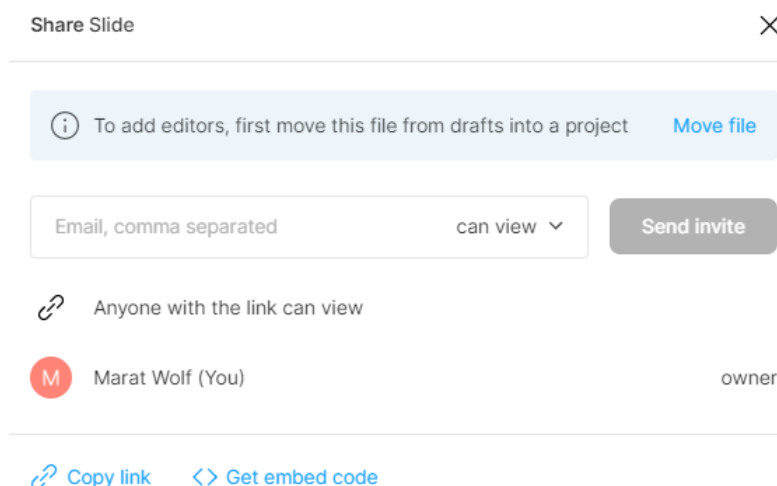


Рисунок 23 – Процесс копирования URL прототипа

После того, как ссылка на прототип мобильного приложения была скопирована, ее необходимо вставить в строку ввода для создания проекта, рисунок 24.

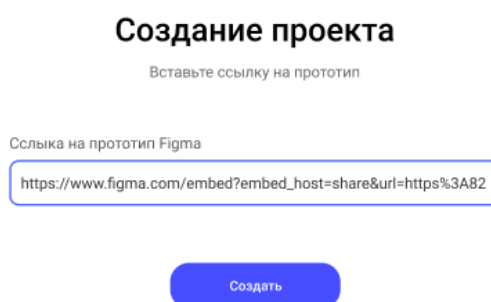


Рисунок 24 – Процесс создания проекта на основе прототипа мобильного приложения

Далее следует описать детали проекта. Для данного проекта описание будет выглядеть следующим образом, как показано на рисунке 25.

### Опишите детали проекта

Название проекта

Цель проекта

Например: Купить автомобиль Mazda

Контекст использования

Вы хотите отправить видео в социальную сеть, но для этого необходимо изменить формат видео.

Поможет пользователям лучше понять для чего нужен ваш продукт

[Продолжить](#)

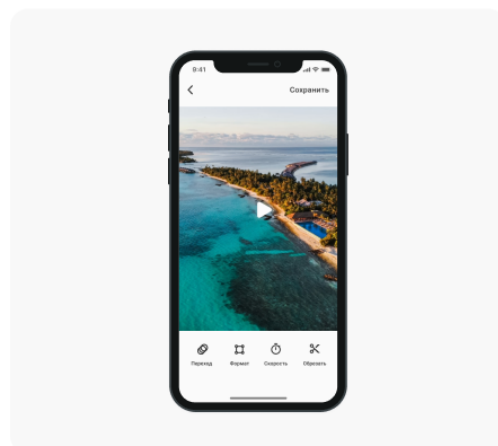


Рисунок 25 – Описание деталей проекта

Далее необходимо составить вопросы, на которые необходимо узнать ответы. Для данного приложения вопросы и варианты ответов будут выглядеть, как показано в таблице 3.

Таблица 3 – Список вопросов и варианты ответов

Вопрос	Варианты ответов
Легко ли было найти экран изменения формата видео?	Да, легко
	Нет, не легко
Понятно ли, что обозначают иконки?	Да, понятно
	Нет, не понятно
Легко ли ориентироваться в приложении?	Да, легко
	Немного сложно
	Трудно



После того проект был опубликован. Через некоторый промежуток времени пользователи выполнили задания, и в данном проекте появились результаты тестирования, которые изображены на рисунке 26.



Рисунок 26 – Результаты 1 и 2 этапа тестирования

Всего участие в тестировании приняло 15 человек, процент успешных попыток равняется 85%, а среднее затраченное время равняется 24 секундам, также процент ошибочных нажатий равняется 20%, что является достаточно хорошим результатом.

Кроме того видно, что приложением достаточно удобно пользоваться и типографика находится на хорошем уровне, но цвета и понятность текста находятся на низком уровне. Внизу показателей есть рекомендации, что необходимо предпринять для того, чтобы повысить данные показатели.

Также была составлена таблица с процентной долей ответов на эти или иные вопросы, результаты приведены в таблице 4.

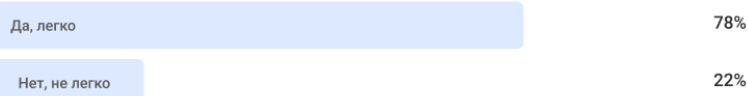
Для того, чтобы анализировать данные результаты было проще, составляется гистограмма, которая наглядно отображает результаты в графическом виде, как показано на рисунке 27.

Таблица 4 – Результаты ответов на вопросы тестирования

Легко ли было найти экран изменения формата видео?	
Да, легко	78%
Нет, не легко	22%
Понятно ли, что обозначают иконки?	
Да, понятно	66%
Нет, не понятно	44%
Легко ли ориентироваться в приложении?	
Да, легко	75%
Немного сложно	17%
Трудно	8%

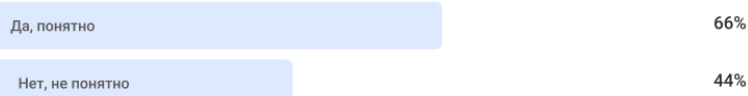
**Легко ли было найти экран изменения формата видео?**

15 из 15 ответили



**Понятно ли, что обозначают иконки?**

15 из 15 ответили



**Легко ли ориентироваться в приложении?**

15 из 15 ответили



Рисунок 27 – Результаты 3 этапа тестирования

Исходя из ответов видно, что находить функции приложения достаточно удобно. Иконки были понятны лишь 66% пользователей, что является не самым лучшим результатом, поэтому следует их доработать. Навигация и

ориентирование в приложении является достаточно удобным благодаря показателю в 75%.

Исходя из результатов тестирования сформировывается следующий вывод, что необходимо изменить цвета данного приложения, переписать текст и сделать его более понятным, а также упростить и изменить иконки приложения.

Для второго тестирования был взят интерфейс веб-сайта (рисунок 28). Прототип состоит из 4 экранов интерфейса и представляет собой логику покупки товара в интернет-магазине.

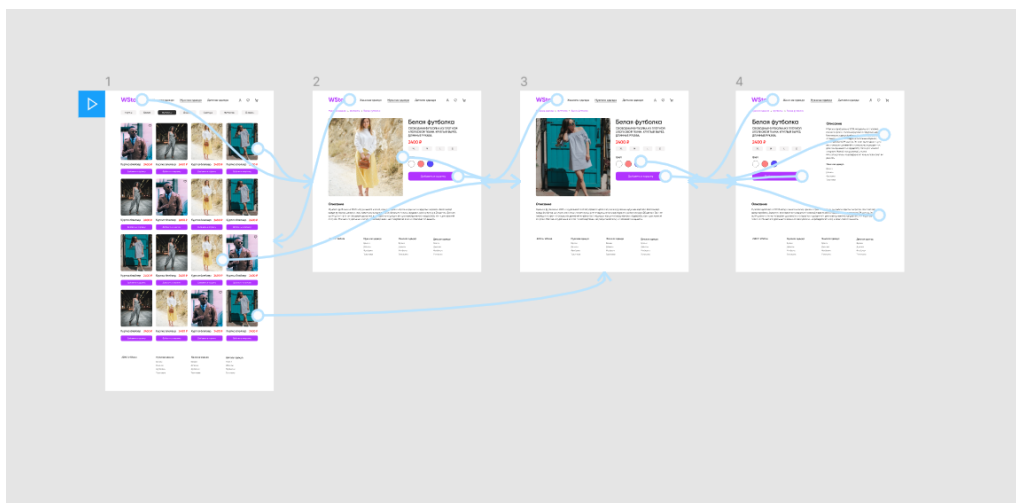


Рисунок 28 – Прототип интернет-магазина

Процесс создания и описания проекта аналогичен процессу создания и описания проекта мобильного приложения, который был рассмотрен ранее.

В юзабилити-тестировании функционала интернет-магазина приняло участие 12 человек. Результаты тестирования показаны на рисунке 29.

Исходя из результатов тестирования видно, что у большей половины пользователей возникли проблемы в взаимодействии с данным интерфейсом, потому что процент ошибочных нажатий и незавершенных сессий составляет свыше 50%. Также пользователи отметили, что интерфейсом недостаточно удобно пользоваться и текста недостаточно понятны. Из положительных

моментов было отмечено, что типографика и цветовая палитра достаточно хорошо подобраны. Соответственно исходя из результатов тестирования и анализа полученных данных, следует изменить логику и информационную архитектуру интернет-магазина, а также переписать и улучшить текста на сайте.

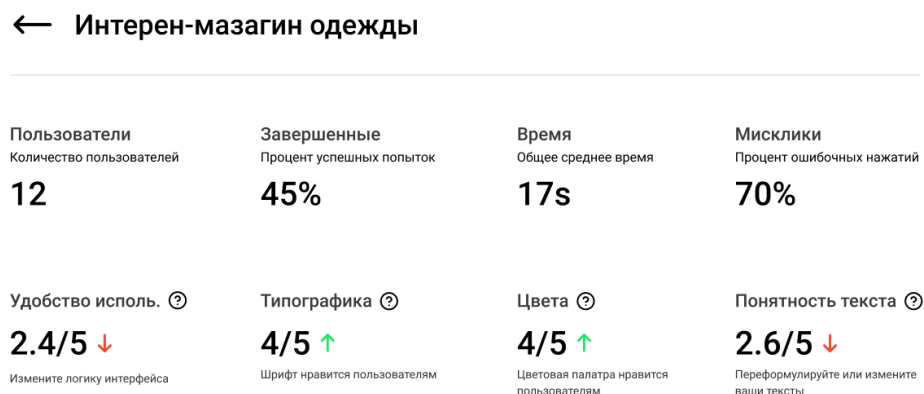


Рисунок 29 – Результаты тестирования

### 3.2 Оценка алгоритма юзабилити-тестирования

Данный алгоритм юзабилити-тестирования для оценки юзабилити пользовательских интерфейсов опирался на метод оценки производительности. Исходя из тестирований данного алгоритма на различных пользовательских интерфейсах, было видно, что данный алгоритм показывает свою эффективность для решения задач, связанных с элементами интерфейсов и логики взаимодействия. На рисунке 30 показано наглядное сравнение результатов тестирования прототипов мобильного приложения для редактирования фото и видео и интернет-магазина.

Такие параметры, как процент успешных завершённых попыток показывает, насколько прост или сложен интерфейс. Результат в 85% у прототипа мобильного приложения показал, что им достаточно легко

пользоваться, а результат в 45% у прототипа интернет-магазина показал, что его интерфейс недостаточно понятен пользователям.

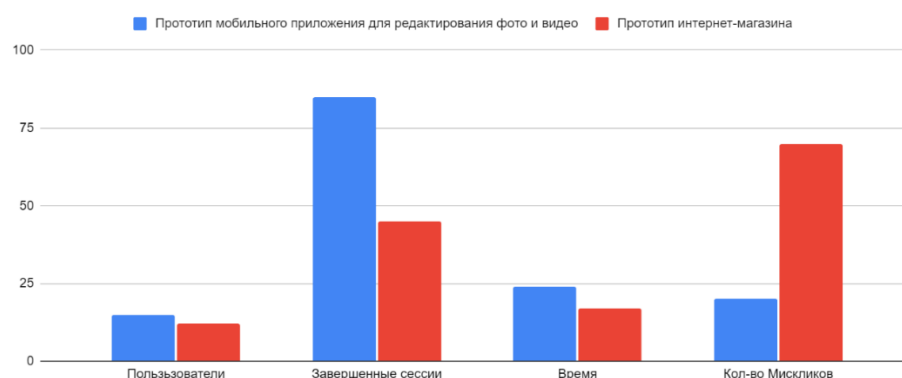


Рисунок 30 – Столбчатая диаграмма сравнения результатов тестирования

В качестве дополнительных критериев подтверждения приводятся такие показатели как процент ошибочных нажатий. Результат в 70% у прототипа интернет-магазина говорит о проблемах интерфейса, а результат в 20% мобильного приложения говорит о достаточно понятном интерфейсе. Кроме того, также имеются показатели относительно восприятия пользователями интерфейса, который оценивается по 5 бальной шкале и ряд дополнительных вопросов в виде опроса, которые помогают сделать более качественное юзабилити-тестирование.

Применение данного алгоритма показывает свою наибольшую ценность, когда тестируются прототипы ранней стадии проектирования, то есть до их программной реализации, что дает возможность выявить проблемы на ранней стадии и своевременно их предотвратить.

Преимуществом реализации данного алгоритма является то, что он тесно взаимодействует со специализированными программами для проектирования и прототипирования, а именно со специализированной программой Figma. Это стало возможным благодаря использованию API

Figma, что экономит время и трудозатраты на создание прототипов вручную, кроме того, это позволяет собрать много важных данных, таких как среднее время использования, долю успешных попыток и процент ошибочных нажатий.

В отчете о проведенном тестировании также наглядно показывается вся информация в количественном виде. Но часто просто предоставление информации недостаточно, поэтому к каждому результату, где это возможно предоставляется рекомендация по устранению проблем и повышению показателей, например изменение цветовой палитры интерфейса или изменение текста на более понятный для пользователей.

В данном разделе было рассмотрено тестирование алгоритма юзабилити на различных пользовательских интерфейсах для их оценки, и оценивание эффективности данного алгоритма юзабилити-тестирования.

## Заключение

В процессе выполнения бакалаврской работы были проанализированы различные методы для оценивания юзабилити и был выбран метод оценки производительности, так как данный метод юзабилити-тестирования обладает высокой эффективностью с проведением различных тестирований, а также сбором и анализом результатов тестирования.

Также был разработан алгоритм для оценки юзабилити тестирования в формате веб-приложения, что способствовало эффективному взаимодействию юзабилити-специалистов и пользователей. В качестве исходных данных для тестирования использовались прототипы интерфейсов.

Для реализации подключения уже готовых прототипов был использован API Figma, что позволило быстро и эффективно тестировать интерфейсы.

В алгоритме были реализованы функции оценивания юзабилити по различным параметрам и вывод результатов в виде наглядного отчета для его дальнейшего ознакомления юзабилити-специалистом.

Для оценивания эффективности разработанного алгоритма для оценки юзабилити интерфейсов было проведено тестирование на различных пользовательских интерфейсах. Тестирование показало слабые и сильные места интерфейсов исходя из различных показателей, таких как процент ошибочных нажатий, затраченного времени, ответов на вопросы и т.д.

Из преимуществ созданного алгоритма следует отметить, что для юзабилити-специалистов формируется готовый отчет с полным результатом тестирования, а также рекомендации по улучшению отдельных аспектов интерфейса.

Применение разработанного алгоритма юзабилити-тестирования показывает свою эффективность на стадии проектирования продукта, до его программной реализации, потому что стоимость внесения изменений на стадии проектирования продукта относительно невысока по сравнению со стадией программной реализации продукта.

Применение на поздней стадии этапа разработки продукта будет показывать меньшую эффективность, так как затраты на изменения сильно возрастают.

Эффективность разработанного алгоритма может быть в дальнейшем улучшена через добавление дополнительной функциональности, которая повысит информативность результатов тестирования. В качестве такой функциональности может выступать реализация записи сессий тестирования, а также реализации карты кликов и их последующей визуализации.

Записи сессий тестирования помогут отслеживать поведение взаимодействия с интерфейсом каждого пользователя, который пройдет тестирование.

Визуализация карты кликов покажет не только количество кликов, но и их наглядное нажатие в виде точек на интерфейсе.

Кроме этого, применение разработанного алгоритма может быть расширена через тестирование интерфейсов умных часов, фитнес-браслетов, смарт-телевизоров и др.



## Список используемой литературы

1. Архитектура Rest [Электронный ресурс]. URL: <https://habr.com/ru/post/483202/> (дата обращения: 15.03.2021).
2. Базовое понимание OAuth 2.0 [Электронный ресурс]. URL: <https://stfalcon.com/ru/blog/post/oauth-2.0> (дата обращения: 22.03.2021).
3. Беленькая, М. Н. Администрирование в информационных системах [Текст] : учеб. пособие для студентов вузов, обучающихся по направлению 230100 "Информатика и вычисл. техника" / М. Н. Беленькая, С. Т. Малиновский, Н. В. Яковенко. - Москва : Горячая линия - Телеком, 2011. - 399 с. 15экз.
4. Берд, Дж. Веб-дизайн. Руководство разработчика. / Дж. Берд. - СПб.: Питер, 2012. - 224 с
5. Буренин, С. Н. Web-программирование и базы данных [Электронный ресурс] : учеб. практикум / С. Н. Буренин. - Москва : Моск. гуманитар. ун-т, 2014. - 120 с
6. Вотролл, Э. Изучаем веб-дизайн / Э. Вотролл, Дж. Сьярто. - М.: Эксмо, 2010. - 496 с.
7. Гагарина, Л. Г. Технология разработки программного обеспечения [Электронный ресурс] : учебное пособие для студентов вузов, обучающихся по направлениям подготовки 09.04.01 и 09.03.03 "Информатика и вычислительная техника" / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. СидороваВиснадул ; под ред. Л. Г. Гагариной. - Москва : ФОРУМ: ИНФРА-М, 2017. - 400 с. <http://znanium.com/go.php?id=768473>
8. Гвоздева, Т.В. Проектирование информационных систем: технология автоматизированного проектирования. Лабораторный практикум. Учебно-справочное пособие / Т.В. Гвоздева, Б.А. Баллод. - СПб.: Лань, 2018. - 156 с
9. Гмурман В. Е. Теория вероятностей и математическая статистика. Учебник. М.: Юрайт, 2018. 480 с.

10. Гордеев, С. И. Организация баз данных в 2 ч. Часть 2 : учебник для вузов / С. И. Гордеев, В. Н. Волошина. — 2-е изд., испр. и доп. — М. : Издательство Юрайт, 2019. — 501 с.
11. ГОСТ 19.701 – 90. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения (ИСО 5807–85) [Текст]. Введен 1992–01–01. – М.: Изд-во стандартов, 1992. – 14 с. – (Единая система программной документации)
12. Дакетт, Дж. HTML и CSS. Разработка и дизайн веб-сайтов / Дж. Дакетт – М.: ЭКСМО, 2013. – 480 с.
13. Закас, Н. JavaScript. Оптимизация производительности/ Н. Закас – СПб.: Символ, 2012. – 256 с.
14. Клейменов, С. А. Администрирование в информационных системах [Текст] : учеб. пособие для студентов вузов, обучающихся по специальности "Информ. системы и технологии" / С. А. Клейменов, В. П. Мельников, А. М. Петраков ; под ред. В. П. Мельникова. - Москва : Академия, 2008. - 271 с.
15. Коцюба, И.Ю. Основы проектирования информационных систем. Учебное пособие / И.Ю. Коцюба, А.В. Чунаев, А.Н. Шиков. – СПб: Университет ИТМО, 2015. – 206 с.
16. Мэтью, Д. HTML5. Разработка веб-приложений / Д. Мэтью. - М.: Рид Групп, 2012. - 320 с
17. Стоян, С. React.js. Быстрый / С. Стоян. — Питер, 2017. — 304 с.
18. Сысолетин, Е. Г. Разработка интернет-приложений : учеб. пособие для СПО / Е. Г. Сысолетин, С. Д. Ростунцев. — М. : Издательство Юрайт, 2019. — 90 с.
19. Федотова, Е. Л. Информационные технологии и системы [Электронный ресурс] : учебное пособие для студентов вузов, обучающихся по специальности 080801 "Прикладная информатика" и другим экономическим специальностям / Е. Л. Федотова. - Москва : ФОРУМ: ИНФРА-М, 2013. - 352 с. URL: <http://znanium.com/go.php?id=374014> (дата обращения 23.10.2018).

20. Фельке-Моррис, Терри Большая книга веб-дизайна (+ CD-ROM) / Терри Фельке-Моррис. - М.: Эксмо, 2012. - 608 с.
21. Черткова, Е. А. Статистика. Автоматизация обработки информации : учеб. пособие для вузов / Е. А. Черткова ; под общ. ред. Е. А. Чертковой. — 2-е изд., испр. и доп. — М. : Издательство Юрайт, 2017. — 195 с.
22. Чистов, Д.В. Проектирование информационных систем : учебник и практикум для академического бакалавриата / Д. В. Чистов. — М. : Юрайт, 2016. — 258 с.
23. Albert, W. Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics / W. Albert. – Morgan Kaufmann, – 2013. – 320 p.
24. API Figma [Электронный ресурс] – URL: <https://www.figma.com/developers/api> (дата обращения: 18.03.2021).
25. Barnum, C. Usability Testing Essentials: Ready, Set...Test! / C. Barnum. – Morgan Kaufmann, – 2010. – 408 p.
26. Krug, S. Don't Make Me Think: A Common Sense Approach to Web Usability / S. Krug. – New Riders Pub, – 2018. – 201 p.
27. Nielsen, J. Usability Engineering / J. Nielsen. – Morgan Kaufmann, – 1993. – 376 p.
28. Performance Measurement [Электронный ресурс] – URL: <http://www.usabilityhome.com/PerfMeas.htm#:~:text=Usability%20Testing%3A%20Performance%20Measurement&text=This%20technique%20is%20to%20used,t he%20tasks%20during%20usability%20test.> (дата обращения: 12.02.2021).
29. Redish, J. A Practical Guide to Usability Testing / J. Redish, J. Dumas. – Intellect Ltd, – 1999. – 404 p.
30. Usability Testing [Электронный ресурс]. URL: <https://www.usability.gov/how-to-and-tools/methods/usability-testing.html> (дата обращения: 04.03.2021).