

Школа Инженерная школа информационных технологий и робототехники
 Направление подготовки 09.03.01 «Информатика и вычислительная техника»
 Отделение школы (НОЦ) Отделение информационных технологий

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Применение инструментов для автоматизации тестирования web-приложений
УДК 004.415.53:004.774

Студент

Группа	ФИО	Подпись	Дата
8В6Б	Богер Виктор Владимирович		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Хамухин Александр Анатольевич	к.т.н.		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН	Конотопский Владимир Юрьевич	к.э.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ООД	Матвиенко Владимир Владиславович			

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Погребной Андрей Владимирович	к.т.н.		

Планируемые результаты обучения по ООП

Код результатов	Результат обучения (выпускник должен быть готов)	Требования ФГОС, критерии АИОР
P1	Применять базовые и специальные естественнонаучные и математические знания в области информатики и вычислительной техники, достаточные для комплексной инженерной деятельности.	Требования ФГОС (ОК-1, 10, ПК-4, 5, 6), критерий 5 АИОР (п. 1.1)
P2	Применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач.	Требования ФГОС (ОК-11, 12, 13, ПК-1, 2, 11), критерий 5 АИОР (п.1.1, 1.2)
P3	Ставить и решать задачи комплексного анализа, связанные с созданием аппаратно-программных средств информационных и автоматизированных систем, с использованием базовых и специальных знаний, современных аналитических методов и моделей.	Требования ФГОС (ОК-1, 8, ПК-2, 4, 6), критерий 5 АИОР (п. 1.2)
P4	Разрабатывать программные и аппаратные средства (системы, устройства, блоки, программы, базы данных и т. п.) в соответствии с техническим заданием и с использованием средств автоматизации проектирования.	Требования ФГОС (ОК-2, 3, ПК-3, 4, 5), критерий 5 АИОР (п. 1.3)
P5	Проводить теоретические и экспериментальные исследования, включающие поиск и изучение необходимой научно-технической информации, математическое моделирование, проведение эксперимента, анализ и интерпретация полученных данных, в области создания аппаратных и программных средств информационных и автоматизированных систем.	Требования ФГОС (ОК-6, ПК-6, 7), критерий 5 АИОР (п.1.4)
P6	Внедрять, эксплуатировать и обслуживать современные программно-аппаратные комплексы, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды.	Требования ФГОС (ОК-4, 15, 16, ПК-9, 10, 11), критерий 5 АИОР (п. 1.5)
	Универсальные компетенции	
P7	Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности.	Требования ФГОС (ОК-1, 4, ПК-1, 6, 7), критерий 5 АИОР (п. 2.1)
P8	Владеть иностранным языком на уровне, позволяющем работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты комплексной инженерной деятельности.	Требования ФГОС (ОК-14, ПК-7), критерий 5 АИОР (п. 2.2)
P9	Эффективно работать индивидуально и в качестве члена группы, состоящей из специалистов различных направлений и квалификаций, демонстрировать ответственность за результаты работы и готовность следовать корпоративной культуре организации.	Требования ФГОС (ОК-2, 3, 4), критерий 5 АИОР (п. 2.3, 2.4)
P10	Демонстрировать знания правовых, социальных, экономических и культурных аспектов комплексной инженерной деятельности.	Требования ФГОС (ОК-1, 5, 9), критерий 5 АИОР (п. 2.5)
P11	Демонстрировать способность к самостоятельной к самостоятельному обучению в течение всей жизни и непрерывному самосовершенствованию в инженерной профессии.	Требования ФГОС (ОК-6, 7), критерий 5 АИОР (п. 2.6)

Школа Инженерная школа информационных технологий и робототехники
 Направление подготовки 09.03.01 «Информатика и вычислительная техника»
 Отделение школы (НОЦ) Отделение информационных технологий

УТВЕРЖДАЮ:
 Руководитель ООП
 _____ Погребной А.В.
 (Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ на выполнение выпускной квалификационной работы

В форме:

бакалаврской работы

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8В6Б	Богеру Виктору Владимировичу

Тема работы:

Применение инструментов для автоматизации тестирования web-приложений	
Утверждена приказом директора (дата, номер)	28.02.2020, №59-50/с

Срок сдачи студентом выполненной работы:	
--	--

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	<ul style="list-style-type: none"> - научная литература по тематике исследования - программное обеспечение, позволяющее создавать автоматизированные тесты - печатный вариант учебного пособия
Перечень подлежащих исследованию, проектированию и разработке вопросов	<ul style="list-style-type: none"> - обзор научной литературы - сбор и анализ имеющихся средств, позволяющее создавать автоматизированные тесты – проектирование программного обеспечения – анализ результатов работы – финансовый менеджмент, ресурсоэффективность и ресурсосбережение – социальная ответственность – заключение
Перечень графического материала	<ul style="list-style-type: none"> - презентация работы
Консультанты по разделам выпускной квалификационной работы	
Раздел	Консультант

Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Конотопский Владимир Юрьевич
Социальная ответственность	Матвиенко Владимир Владиславович

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	
--	--

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Хамухин Александр Анатольевич	к.т.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8В6Б	Богер Виктор Владимирович		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
8В6Б	Богеру Виктору Владимировичу

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	бакалавриат	Направление/специальность	09.03.01 Информатика и вычислительная техника

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Использовать действующие ценники и договорные цены на потребленные материальные и информационные ресурсы, а также указанную в МУ величину тарифа на эл. энергию
2. Нормы и нормативы расходования ресурсов	—
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	Действующие ставки единого социального налога и НДС (см. МУ, ставка дисконтирования $i=0.1$)

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения	Оценка готовности полученного результата к выводу на целевые рынки, краткая характеристика этих рынков
2. Планирование и формирование бюджета научных исследований	Построение плана-графика выполнения ВКР, составление соответствующей сметы затрат, расчет величины НДС и цены результата ВКР
3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	Качественная и количественная характеристика экономического и др. видов эффекта от внедрения результата, определение эффективности внедрения

Перечень графического материала (с точным указанием обязательных чертежей):

1. Оценка конкурентоспособности технических решений
2. Матрица SWOT
3. Альтернативы проведения НИ
4. График проведения и бюджет НИ – <u>выполнить</u>
5. Оценка ресурсной, финансовой и экономической эффективности НИ – <u>выполнить</u>

Дата выдачи задания для раздела по линейному графику

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ОСГН	Конотопский Владимир Юрьевич	К.Э.Н.		26.02.2020 г.

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8В6Б	Богер Виктор Владимирович		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8В6Б	Богеру Виктору Владимировичу

Школа	ИШИТР	Отделение (НОЦ)	ОИТ
Уровень образования	бакалавриат	Направление/специальность	09.03.01 Информатика и вычислительная техника

Тема ВКР:

Применение инструментов для автоматизации тестирования web-приложений	
Исходные данные к разделу «Социальная ответственность»:	
1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	–Проект по созданию автоматизированных тестов web-приложений; –Кибернетический центр, 407 аудитория
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности: <ul style="list-style-type: none"> – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны. 	– "Трудовой кодекс Российской Федерации" от 30.12.2001 N 197-ФЗ (ред. от 24.04.2020); – ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя; – ГОСТ Р 50923-96. Дисплеи. Рабочее место оператора. Общие эргономические требования и требования к производственной среде. Методы измерения; – ГОСТ Р ИСО 14738-2007. Безопасность машин. Антропометрические требования при проектировании рабочих мест машин; – ГОСТ Р ИСО 6385-2016. Эргономика. Применение эргономических принципов при проектировании производственных систем; – СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы.
2. Производственная безопасность: 2.1. Анализ выявленных вредных и опасных факторов 2.2. Обоснование мероприятий по снижению воздействия	– отклонение показателей микроклимата; – отсутствие или недостаток естественного света; – опасность поражения током; – монотонность труда.
3. Экологическая безопасность:	– отсутствие прямого воздействия на окружающую среду.

	– средства, необходимые для разработки и эксплуатации программного комплекса могут наносить вред окружающей среде. Вследствие этого требуется организация утилизации отходов оргтехники.
4. Безопасность в чрезвычайных ситуациях:	Во время разработки программного продукта и его дальнейшего использования типичной ЧС является: – пожар;

Дата выдачи задания для раздела по линейному графику	
--	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
ассистент ООД	Матвиенко Владимир Владиславович			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8В6Б	Богер Виктор Владимирович		

Реферат

Ключевые слова: тестирование, автоматизация, тестировщик, план тестирования.

Объектом исследования является процесс тестирования web - приложения в процессе разработки и эксплуатации проекта.

Цель работы – автоматизация процесса тестирования web - приложения.

В рамках работы ВКР был произведен анализ самых известных продуктов, созданных для автоматизации процесса тестирования. На основе анализа был выбран наиболее подходящий продукт.

В результате исследования был создан план тестирования, на основе которого разработан набор автоматических тестов. Система была настроена согласно спроектированному алгоритму.

Степень внедрения: автоматизация процесса тестирования web – приложений – сложный и многогранный процесс, внедрение которого требует большого количества времени и доработки вместе с развитием проекта.

Область применения: автоматизация процесса тестирования web – приложений представляет интерес для компаний, занимающихся разработкой web – приложений.

Экономическая значимость разработки состоит в значительном сокращении труда тестировщиков, а также выгодой финансовых вложений в долгосрочной перспективе.

В будущем планируется доработка процесса автоматизации тестирования и его внедрение в другие проекты.

Определения, обозначения, сокращения, нормативные ссылки

Тестировщик - специалист, занимающийся тестированием.

Автоматизированное тестирование – процесс тестирования, при котором основные функции и шаги теста производятся автоматически с помощью.

Тестовый скрипт – это набор инструкций, для автоматической проверки определенной части приложения.

Тестовый сценарий – набор входных значений, предусловий выполнения, ожидаемых результатов и постусловий выполнения, разработанный для определенной цели или тестового условия, таких как выполнение определенного пути программы или же для проверки соответствия определенному требованию.

Список сокращений:

АТ – Автоматизация Тестирования;

ПО – Программное Обеспечение.

Оглавление

Введение	13
1 Аналитический обзор области тестирования	15
1.1 Методики тестирования	16
1.1.1 Различие ручного и автоматизированного тестирования	16
1.2 Уровни тестирования программного обеспечения	18
1.2.1 Компонентное (модульное) тестирование.....	18
1.2.2 Интеграционное тестирование.....	19
1.2.3 Системное тестирование	19
1.2.4 Приемочное тестирование.....	20
2 Используемые средства разработки	20
2.1 Обзор и сравнение инструментов автоматизации тестирования.....	20
2.2 Язык тестовых скриптов	24
2.3 Распределенная система контроля версий	27
3 Процесс автоматизации тестирования интерфейса пользователя web-приложения	28
3.1 Составление плана автоматизации процесса тестирования.....	28
3.2 Выбор стратегии автоматизации.....	29
3.3 Разработка тестовых сценариев	29
3.4 Разработка тестовых скриптов	30
3.5 Проектирование алгоритма автоматизации процесса тестирования web-приложения	33
3.6 Реализация автоматизации тестирования web-приложения	34
4 Результаты проведенного исследования.....	36
4.1 Оценка эффективности внедрения автоматизации тестирования web-приложения	36
5 Социальная ответственность.....	39
5.1 Введение	39
5.2 Правовые и организационные вопросы обеспечения безопасности ...	39
5.2.1 Правовые нормы трудового законодательства для рабочей зоны оператора ЭВМ.....	39
5.2.2 Организационные мероприятия при компоновке рабочей зоны....	41
5.3 Производственная безопасность	41
5.3.1 Анализ опасных и вредных производственных факторов	42
5.4 Экологическая безопасность	47
5.4.1 Анализ влияния процесса исследования на окружающую среду ..	47
5.4.2 Обоснование мероприятий по защите окружающей среды.....	47
5.5 Безопасность в чрезвычайных ситуациях	48

5.5.1	Анализ вероятных ЧС, которые могут возникнуть на рабочем месте при проведении исследований	48
5.5.2	Обоснование мероприятий по предотвращению ЧС и разработка порядка в случае возникновения ЧС	48
5.6	Выводы по разделу	50
6	Финансовый менеджмент, ресурсоэффективность и ресурсопотребление	50
6.1	Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения.....	50
6.1.1	Потенциальные потребители результатов исследования	50
6.1.2	Анализ конкурентных технических решений	51
6.1.3	SWOT-анализ	52
6.2	Планирование научно-исследовательских работ	54
6.2.1	Структура работ в рамках научного исследования	54
6.2.2	Определение трудоемкости выполнения работ	54
6.2.3	Расчет материальных затрат научно-технического исследования.	59
6.2.4	Расчет заработной платы	59
6.2.5	Расчет зарплат на социальный налог	60
6.2.6	Расчет затрат на электроэнергию	60
6.2.7	Расчет амортизационных расходов	61
6.2.8	Расчет расходов, учитываемых непосредственно на основе платежных документов	63
6.2.9	Расчет прочих расходов.....	63
6.2.10	Расчет общей себестоимости разработки	64
6.2.11	Расчет прибыли	64
6.2.12	Расчет НДС	64
6.2.13	Расчет НИР	65
6.2.14	Оценка экономической эффективности проекта	65
6.2.15	Определение срока окупаемости инвестиций	66
6.3	Вывод по разделу	66
7	Заключение.....	67
8	Список использованной литературы	68
	Приложение А	70
	Приложение Б	73

Введение

В современном мире существует уже больше 2,5 миллиардов сайтов. Большая часть из них рассчитана на получение прибыли и конкурентность в этой сфере крайне высокая, поэтому цена ошибки, в процессе разработки могут привести к потере выручки. Для этого необходимо качественное тестирование веб – приложений. Этот процесс подразумевает под собой массу задач из разных областей, а именно: строение архитектуры, сетевая безопасность, функционал, работа с огромным объемом данных, дизайн, нагрузка и так далее.

К основным видам тестирования относят: ручное и автоматизированное тестирование. Под первым видом тестирования понимается то, что инженер самостоятельно выполняет действия, которые доступны пользователю. Тестировщик проверяет сайт на предмет соответствия требованиям, ищет ошибки и прочее.

Автоматизированное тестирование заключается в создании программ, задача которых сводится к исполнению действий, указанных в тестовых сценариях. Поэтому тестировщик может получить результат выполнения этих сценариев без своего непосредственного участия: различные ошибки на разных стадиях разработки, или же успешное прохождение теста.

Исходя из этого, стоит отметить, что автоматизированное тестирование приводит к полной замене человеческого труда, поскольку созданная программа нацелена на многократное использования и нуждается в доработке только при изменении самого проекта. Однако к минусу такого подхода можно отнести тот факт, что для того, чтобы внедрять автоматизированные тесты, нужно убедиться в том, что проект будет использоваться в долгосрочной перспективе, поэтому нет смысла тратить ресурсы на разработку для малых проектов.

Автоматизированное тестирование используется вместе с написанием unit-тестов, которые созданы для проверки функционала на первых этапах проекта. После этого, тесты используются в конце итераций для охвата функционала в целом.

На данный момент в мире создано большое количество фреймворков для автоматизации тестирования. Основная проблема заключается в том, что все это количество инструментов тестирования имеют свои плюсы и недостатки, а также заточены под определенные ситуации. Ведь суть автоматизации заключается в

экономии времени и снижении трудозатратности проекта, и, если не подобрать правильный стек технологий для той или иной цели, то это лишь усложнит процесс разработки. В связи с этим необходимо было максимально универсализировать каждый шаг жизненного цикла ПО. Именно поэтому для написания автоматических тестов было принято решение выбрать фреймворк PyTest для языка Python. Такое решение было принято в связи с тем, чтобы лучше понять суть разработки систем тестирования. Плюсом данного языка является то, что в нем отлично развиты методологии объектно-ориентированного программирования, различные фикстуры и параметризации исполняемых тестов.

1 Аналитический обзор области тестирования

Тестирование является неотъемлемой частью процесса разработки программного обеспечения. Основной целью тестирования является обнаружение и устранение ошибок до того момента, пока продукт не попадет в эксплуатацию. Помимо этого, сюда можно отнести такие факторы как заявленные требования и характеристики, безопасность продукта, требования по эргономике, удобство использования, поведение продукта при нагрузках и так далее. К недостаткам данного процесса можно отнести то, что все программы претерпевают изменения в связи с расширением и добавлением функционала, вследствие чего будет расширяться набор тестов. Зачастую, жизненный цикл тестирования документируется в виде тест плана, тест-кейсов и чек листов. В тестовом плане описывается стратегия тестирования, методы и средства тестирования, порядок тестирования и другие его особенности.

Каждая программа – это механизм по переработке информации. Она так или иначе преобразовывает информацию на выходе. Как раз для выходной информации создаются различные интерфейсы [1]:

- Пользовательский интерфейс (UI);
- Программный интерфейс (API);
- Сетевой протокол;
- Файловая система;
- Состояние окружения;
- События.

Наиболее распространенные интерфейсы это:

- Пользовательский;
- Графический;
- Текстовый;
- Консольный;
- Речевой.

Вышеперечисленные интерфейсы нуждаются в тестировании. Для этого создаются тестовые наборы для того, чтобы проверить, как ведет себя программа в тех или иных ситуациях.

1.1 Методики тестирования

Многие компании сталкиваются с вопросом о выборе между ручным и автоматизированным тестированием, поскольку обеспечение качества создаваемого программного обеспечения – трудоемкий и сложный процесс, включающий в себя различные параметры качества, модель и стратегия тестирования, всевозможные проверки программного обеспечения и, что самое важное, выявление ошибок и дефектов и их дальнейшее решение и оптимизация.

И перед тем, как начать работу, необходимо определиться, есть ли необходимость внедрения автоматизированного тестирования или стоит остановиться на ручном. Очевидно, что каждый из этих подходов обладают преимуществами и недостатками, о которых необходимо знать для принятия осознанного решения, которое приведет к достижению цели.

1.1.1 Различие ручного и автоматизированного тестирования

В случае с ручным подходом, все тестовые случаи (test cases), выполняются вручную, без использования дополнительных программных средств. В случае с автоматизированным подходом все иначе – тестовые случаи выполняются с помощью разработанных скриптов.

Для любого проекта важны три фактора: время, стоимость и качество. Чтобы успешно создать проект необходимо сократить как стоимость, так и время, при этом суметь сохранить качество. И если речь заходит о тестировании, то один из подходов с поставленной задачей может справиться куда лучше второго.

Тестирование программного обеспечения может выполняться следующими методами: методом черного или белого ящика, проводиться по строго заготовленным сценариям или же интуитивно.

Так же стоит учитывать и тот факт, что тестирование может быть различных уровней: модульным, системным, интеграционным или приемочным. Кроме того, оно

может быть основано на проверке различных аспектов качества: производительности, безопасности и так далее.

Чтобы решить, когда стоит инвестировать в автоматизацию, а когда выгоднее проводить проверку вручную, поможет таблица ниже.

Таблица 1 – Сравнение ручного и автоматизированного тестирования.

Характеристика	Ручное тестирование	Автоматизированное тестирование
Надежность	Низкая надежность. Человеческий фактор	Высокая надежность. Исключен человеческий фактор
Скорость выполнения	Низкая	Быстрая
Стоимость	Включает зарплату инженера	Включает зарплату инженера и стоимость инструментов автоматизации
Регулярность использования	Небольшое количество	Используется многократно

Таким образом можно отметить, что ручные проверки нужны в следующих видах тестирования:

- Исследовательское тестирование. Создание тестовых сценариев основано на анализе выполняемого проекта и опыта тестировщика. Используется в случае, когда тестирование критичных модулей нужно выполнить в срочном порядке;
- Тестирование юзабилити (тестирование удобства использования). Проводится при необходимости получения информации об удобстве использования продукта;
- Интуитивное тестирование (ad-hoc testing). Такой вид тестирования без создания тестового плана и тестовых сценариев. Тестирование основано на опыте тестировщика.

Для автоматизации присущи следующие виды тестирования:

- Регрессионное тестирование. Данный вид тестирования стоит на первом месте для автоматизации, так как необходим регулярный запуск тестов. кандидат на автоматизацию из-за регулярного запуска тестов;
- Нагрузочное тестирование. Автоматизация нагрузочного тестирования позволяет быстрее получать результаты, экономить на мощностях и стоимости инструментов;
- Тестирование локализации. Необходимо в случае, если продукт попадает на мировой рынок.

Важно отметить, что при регулярном обновлении продукта, добавляется новая функциональность, значит тестирования локализации выгоднее автоматизировать.

Исходя из общемировой практики, самыми выигрышными вариантами будет сочетание двух подходов. Доля таких подходов будет различаться в зависимости от требований проекта, сроков, финансов и так далее.

1.2 Уровни тестирования программного обеспечения

К основным уровням тестирования принято относить [2]:

- Компонентное или Модульное тестирование (Component Testing or Unit Testing);
- Интеграционное тестирование (Integration Testing);
- Системное тестирование (System Testing);
- Приемочное тестирование (Acceptance Testing).

1.2.1 Компонентное (модульное) тестирование

Компонентное (модульное) тестирование нацелено на проверку функциональности и поиск дефектов в модулях (частях) приложения. Это могут быть как сами модули, так и функции, классы, объекты. Стоит отметить, что найденные ошибки, зачастую исправляются в коде без их внесения в системы менеджмента багов (Bug Tracking System).

Одним из самых популярных подходов к данному тестированию является разработка автоматизированных тестов до начала основного кодирования

(разработки) программного обеспечения. Это называется разработка от тестирования (test-driven development) или подход тестирования вначале (test first approach). При этом подходе создаются и интегрируются небольшие куски кода, напротив которых запускаются тесты, написанные до начала кодирования. Разработка ведется до тех пор, пока все тесты не будут успешно пройдены.

По сути, компонентный и модульный уровень тестирования представляет одно и то же, разница лишь в том, что в компонентном тестировании в качестве параметров функций используют реальные объекты и драйверы, а в модульном тестировании - конкретные значения [2].

1.2.2 Интеграционное тестирование

Интеграционное тестирование расширяет охватываемую область тестирования и состоит из двух или более модулей. Так, проверяется взаимодействие между различными частями системы.

Выделяют два уровня интеграционного тестирования [3]:

- Компонентный интеграционный уровень (Component Integration testing);
- Системный интеграционный уровень (System Integration Testing).

В первом случае происходит проверка взаимодействия между компонентами системы после проведения компонентного тестирования. Во втором же случае - взаимодействие между разными системами после проведения системного тестирования.

Также несколько подходов к интеграционному тестированию [3]:

- Снизу вверх (Bottom Up Integration);
- Сверху вниз (Top Down Integration);
- Большой взрыв ("Big Bang" Integration).

1.2.3 Системное тестирование

Задача системного тестирования сводится к проверке и функциональных, и нефункциональных требований и выявлении дефектов в целом [3]:

- Неверное использование ресурсов системы;
- Непредусмотренные комбинации данных пользовательского уровня;
- Несовместимость с окружением;

- Непредусмотренные сценарии использования;
- Отсутствующая или неверная функциональность;
- Неудобство использования.

Для минимизации рисков, связанных с особенностями поведения в системы в той или иной среде, во время тестирования рекомендуется использовать окружение максимально приближенное к тому, на которое будет установлен продукт после выдачи.

1.2.4 Приемочное тестирование

Приемочное тестирования включает в себе формальную проверку (тестирование) соответствия системы требований и проводится с целью[3]:

- Определения удовлетворяет ли система приемочным критериям;
- Вынесения решения заказчиком или другим уполномоченным лицом принимается приложение или нет.

Основу такого тестирования составляют наборы тестовых случаев (test cases) и сценариев, которые разработаны по требованиям к приложению.

Используется в случаях, когда продукт достиг необходимого уровня качества.

2 Используемые средства разработки

В этом разделе будет обзор, а также сравнение средств разработки для автоматизации тестирования. Помимо этого, рассмотрим наиболее подходящий набор технологий для успешного создания проекта.

2.1 Обзор и сравнение инструментов автоматизации тестирования

На данный момент было создано большое количество систем для автоматизированного тестирования веб-приложений. К сожалению цена таких продуктов может оказаться крайне высока даже несмотря на ограниченный функционал и низкую скорость работы. Также существуют и бесплатные системы, но несравнимо большая часть из них работают по методу record and play, и вдобавок, тоже могут быть сильно ограничены в функционале. Ниже мы рассмотрим самые популярные инструменты для автоматизации тестирования.

Начнем с Selenium. Selenium – это проект, в рамках которого разрабатывается серия программных продуктов с открытым исходным кодом (open source):

- Selenium WebDriver;
- Selenium RC;
- Selenium Server;
- Selenium Grid;
- Selenium IDE.

Считается, что Selenium является самым популярным фреймворком с открытым исходным кодом, предназначенным для автоматизации тестирования веб-приложений. Он успел завоевать сердца многих тестировщиков, особенно тех, у кого в распоряжении имеются продвинутое навыки программирования и опыт написания скриптов. Selenium можно рассматривать в качестве родоначальника некоторых современных инструментов автоматизации тестирования с открытым исходным кодом, например: Katalon Studio, Watir, Protractor и Robot Framework.

Фреймворк Selenium поддерживается несколькими ОС (Windows, Mac, Linux), а также многими браузерами (Chrome, Firefox, IE, и браузерами Headless). Скрипты для данного фрейма можно написать на большинстве популярных сегодня языков программирования: Java, Groovy, Python, C#, PHP, Ruby и Perl [5].

Однако стоит отметить, что данный фреймворк имеет как плюсы, так и минусы. К преимуществам можно отнести гибкость, а также возможность написания сложных и эффективных скриптов для тестирования разрабатываемых приложений. С другой стороны, для того чтобы начать работать с Selenium, тестировщик должен обладать незаурядными знаниями в программировании и быть готовым уделять некоторое количество своего времени и энергии для написания специальных фреймов и библиотек, обеспечивающих выполнение определенных функций в процессе тестирования.

Katalon Studio — это эффективный инструмент для автоматизации процесса тестирования веб-приложений, мобильных приложений и веб-сервисов. Katalon Studio является потомком таких фреймворков, как Selenium и Appium. Он перенял у последних множество преимуществ, связанных с интегрированной автоматизацией тестирования ПО [5].

Для начала работы с данным инструментом вы можете как обладать начальными знаниями в тестировании ПО, так и быть настоящим гуру своего дела. Люди, далекие от программирования, могут с легкостью запустить свой проект по автоматизации тестирования (например, запустив функцию Object Spy для записи тестовых скриптов), а для программистов и опытных тестировщиков Katalon Studio окажется полезным с точки зрения экономии времени при написании новых библиотек и поддержке существующих скриптов.

Katalon Studio может быть интегрирован в CI/CD, он прекрасно работает в связке с популярными инструментами во время тестирования ПО: qTest, JIRA, Jenkins и Git. Для него предусмотрена приятная функция — Katalon Analytics, благодаря которой пользователи получают полное представление о процессе тестирования. Для этого предусмотрены специальные отчеты, которые выводятся на экран пользователей в виде метрики, диаграмм и графиков [5].

Watir — это инструмент с открытым исходным кодом для автоматизации тестирования веб-приложений, использующий библиотеки Ruby. Для Watir предусмотрена возможность кросс-браузерного тестирования в большинстве существующих браузеров: Firefox, Opera, headless-браузеров и IE. Он также поддерживает управляемое данными тестирование и интегрирован с инструментами BDD, такими как: RSpec, Cucumber и Test / Unit [5].

Ниже указан полноценный анализ инструментов и их различия.

Продукт	Selenium	Katalon Studio	Unified functional testing	TestComplete	Watir
Доступен для использования	2004	2015	1998	1999	2008
Приложения для тестирования	Web-приложения	Web (UI, API), Мобильные приложения	Web (UI, API), Мобильные, десктопные, упакованные приложения	Web (UI, API), Мобильные, десктопные приложения	Web-приложения
Цена	Бесплатно	Бесплатно	В зависимости от комплектации >2300 USD	В зависимости от комплектации >1100 USD	Бесплатно
Платформы	Windows, Linux, OS X	Windows, Linux, OS X	Windows	Windows	Windows, Linux, OS X
ЯП	Java, C#, Perl, Python, JavaScript, Ruby, PHP	Java/Groovy	VBScript	JavaScript, Python, VBScript, Jscript, Delphi, C++, C#	Ruby

Рисунок 1 - Сравнение инструментов для автоматизации тестирования

Навыки программирования	Продвинутые навыки необходимы для внедрения различных инструментов	Не требует. Не рекомендуется для расширенных тестовых сценариев	Не требует. Не рекомендуется для расширенных тестовых сценариев	Не требует. Не рекомендуется для расширенных тестовых сценариев	Продвинутые навыки необходимы для внедрения различных инструментов
Легкость установки и использования	Необходимы продвинутые навыки для установки и использования	Легок в установке и использовании	Сложен в установке. Нужны навыки в использовании	Легок в установке. Нужны навыки для использования инструментов	Необходимы продвинутые навыки для внедрения различных инструментов

Рисунок 2 - Сравнение инструментов для автоматизации тестирования (продолжение)

Так как нас не устраивает подход Record Playback, который хоть и является самым популярным благодаря простоте и автоматическому созданию скриптов, было

принято решение придерживаться подхода Scripting. В данном подходе разработкой скриптов занимается программист высокого уровня, который работает отдельно от тестировщиков, непосредственно запускающих тесты. Данный метод исключает хрупкость воспроизведения и перевозпроизведения тестов, а также позволяет совершать куда более сложные задачи в сравнении с первым методом. Также немало важна цена продукта, поддерживаемые операционные системы и браузеры, поэтому выбор пал на вариант Selenium. Помимо этого, была важна возможность разработки скриптов на языке Python.

2.2 Язык тестовых скриптов

Python – изумительный язык программирования. Как сказал Дэн Каллаэн (Dan Callahan) в своем докладе на PyCon 2018, "Пайтон занимает второе место в списке лучших языков, подходящих для чего угодно, и это чудесно". Однако я убежден, что если рассматривать тест-автоматизацию, то Python – один из наилучших языков для нее. Ниже – десять причин, почему это так:

1. Дзен Python, зафиксированный в PEP20 – идеальное руководство для тест-автоматизации. Код тестов должен быть естественным мостиком между шагами тестов, написанными на обычном языке, и программными вызовами, автоматизирующими их. Тесты должны быть читабельны и наглядны, потому что они описывают функциональность, которая тестируется. Они должны внятно сообщать, что именно они покрывают. Простые шаги лучше сложных. Код тестов должен как можно меньше увеличивать энтропию тестов. Выразительная элегантность Python делает его мощным мостом от тест-кейсов к тест-коду.
2. Pytest – один из лучших тест-фреймворков, доступных сейчас на любом языке, а не только на Python. Он может справиться с любым функциональным тестом, будь то юнит, интеграция или end-to-end. Тест-кейсы пишутся просто как функции (никаких побочных эффектов, если вы избегаете глобальных переменных), и могут принимать параметризованные входные данные. Общие наборы объектов – это обобщенный, переиспользуемый способ работы с настройкой и

очисткой. Базовые утверждения-ассерты имеют автоматический самоанализ, и сообщения о падении выдают значимую информацию. Тесты можно фильтровать при выполнении. Плагины расширяют pytest и позволяют покрывать код, запускать тесты параллельно, использовать сценарии Gherkin и интегрироваться с другими фреймворками – например, Django и Flask. Другие фреймворки Python тоже хороши, но pytest – это лучшее на данный момент.

3. Несмотря на все жалобы на CheeseShop, у Python богатая библиотека полезных пакетов для тестирования: pytest, unittest, doctest, tox, logging, paramiko, requests, Selenium WebDriver, Splinter, Hypothesis.
4. Python и объектно-ориентирован, и функционален. Он позволяет разработчикам решать, что лучше подойдет для их задач – функции или классы. Это очень полезно для тест-автоматизации, потому что, во-первых, отсутствуют побочные эффекты распределенных функций, а во-вторых, простой синтаксис этих функций делает их понятными. Pytest использует функции для кейсов вместо того, чтобы записывать их в классы, как JUnit.
5. Динамическая, сразу доступная утиная типизация Python отлично подходит для тест-автоматизации, потому что большая часть функциональных тестов выше юнит-уровня не требует пристального внимания к типам. Однако если вам требуются статические типы, то вам помогут mypy, Pyre, и MonkeyType. Типировать можно любым способом.
6. Хорошая поддержка IDE сильно упрощает использование языка и его фреймворков. Если говорить о Python -тестировании, то, JetBrains PyCharm поддерживает визуальное тестирование с pytest, unittest, и doctest прямо "из коробки", а профессиональная версия включает поддержку фреймворков BDD (например, pytest-bdd, behave, lettuce) и веб-разработки. Если требуется что-то полегче, то Visual Studio Code сейчас бешено популярен. Его Python -расширения поддерживают кучу полезного: сниппеты, линтинг, окружения, дебаг, тестирование и

терминал командной строки прямо в окне. Atom, Sublime, PyDev, и Notepad++ тоже отлично работают.

7. Python и командная строка похожи на арахисовое масло и джем – этот брак заключен на небесах. Весь процесс автоматизации тестирования может управляться через командную строку. Pipenv умеет управлять пакетами и окружениями. В любом тест-фреймворке есть запуск консоли для поиска и запуска тестов. Нет необходимости сначала создавать тестовый код, так как Python – интерпретируемый язык, и это еще больше упрощает запуск. Богатая поддержка командной строки сильно упрощает управление тестированием как вручную, так и при помощи инструментов или билд-скриптов/процессов CI.
8. Python всегда был дружелюбен к новичкам благодаря своему Дзену – неважно, делают ли они первые шаги в программировании, или уже имеют солидный опыт. Это большое преимущество Python как языка автоматизации, потому что тесты нужно создавать быстро и просто. Никто не хочет тратить время зря, когда на руках новые фишки, которые нужно проверить. К тому же многие ручные тестировщики, зачастую не имеющие опыта программирования, нынче начинают автоматизировать (добровольно или принудительно), и низкая кривая обучения Python им очень помогает.
9. Python, конечно, отлично подходит новичкам, но он не игрушка. Он подходит для задач индустриального масштаба, потому что его дизайн делает упор на один правильный способ выполнения задания. Разработка может масштабироваться благодаря значимому синтаксису, хорошей структуре, модулярности и богатой экосистеме инструментов и пакетов. Гибкость командной строки позволяет встроиться в любой инструмент или процесс. Python может быть медленнее других языков, но для функциональных тестов это не важно, так как задержки системы (например, задержки ответа веб-страниц и REST-вызовов) намного длиннее, чем провалы в производительности на уровне языка.

10. Python – один из наиболее популярных языков программирования в мире. Его систематически выводят в топ на TIOBE, Stack Overflow, и GitHub (а также GitHut). Его любят выбирать веб-разработчики, инженеры инфраструктур, дата-аналитики и тест-автоматизаторы. Сообщество Python тоже помогает продвигать язык. Разработчиков на Python очень много, как и онлайн-поддержки для него. В обозримом будущем этот язык (я про Python 3) никуда не денется.

На данный момент было создано большое количество систем для автоматизированного

2.3 Распределенная система контроля версий

Git – распределённая система контроля версий, которая даёт возможность разработчикам отслеживать изменения в файлах и работать совместно с другими разработчиками. Она была разработана в 2005 году Линусом Торвальдсом, создателем Linux, для того, чтобы другие разработчики могли вносить свой вклад в ядро Linux. Git известен своей скоростью, простым дизайном, поддержкой нелинейной разработки, полной децентрализацией и возможностью эффективно работать с большими проектами.

Git стоит отдельно от других СКВ из-за подхода к работе с данными, так как в других системах хранится информация в виде списка изменений.

К преимуществам Git относят:

- Бесплатный и open-source. Это значит, что его можно бесплатно скачать и вносить любые изменения в исходный код;
- Небольшой и быстрый. Он выполняет все операции локально, что увеличивает его скорость. Кроме того, Git локально сохраняет весь репозиторий в небольшой файл без потери качества данных;
- Резервное копирование. Git эффективен в хранении бэкапов, поэтому известно мало случаев, когда кто-то терял данные при использовании Git;

- Простое ветвление. В других СКВ создание веток— утомительная и трудоёмкая задача, так как весь код копируется в новую ветку. В Git управление ветками реализовано гораздо проще и эффективнее.

GitHub – сервис онлайн-хостинга репозитория, обладающий всеми функциями распределённого контроля версий и функциональностью управления исходным кодом – всё, что поддерживает Git и даже больше. Обычно он используется вместе с Git и даёт разработчикам возможность сохранять их код онлайн, а затем взаимодействовать с другими разработчиками в разных проектах.

3 Процесс автоматизации тестирования интерфейса пользователя web-приложения

3.1 Составление плана автоматизации процесса тестирования

Перед тем как начать автоматизацию процесса тестирования, нужно составить требования и план к проекту. Тестовые планы составляются с помощью общепринятых шаблонов.

Чаще всего время разработчика в тестировании ограничено. Для этого на ранних этапах цикла тестирования создается план автоматизации. Он должен быть простым и четко отображать выбранную стратегию.

Тестовые планы желательно составлять на основе общепринятых практик и стандартов, чем и будем руководствоваться. Также было выделено несколько этапов данного процесса:

- Решение об автоматизированном тестировании;
- Цели и примерные результаты;
- Стратегия автоматизации;
- Средства автоматизации тестирования;
- План тестирования;
- Разработка скриптов;
- Проведение тестирования и анализ результатов;
- Оценка эффективности проведения автоматизации тестирования.

План тестирования в приложении А.

3.2 Выбор стратегии автоматизации

Для первого этапа была выбрана стратегия автоматизации. Поскольку ресурсы на проект невелики, а также на проекте не применялась автоматизация тестирования, то было отдано предпочтение стратегии “Let’s try”. Ее характеризуют следующими факторами:

- Уделять большее внимание составлению тест-планов, тест-кейсов и т.д.;
- Уделять большее внимание инструментам, которые можно использовать как помощь в ручном тестировании;
- Уделять большее внимание экспериментированию с технологиями и методологиями АТ, так как нет необходимости в получении срочных результатов, поэтому можно экспериментировать;
- Уделять внимание работе с проектом, начиная с верхнего уровня.

Благодаря такой стратегии, внедрение автоматизации будет проще и нагляднее. После выполнения таких требований можно приступать к более сложной работе, так как у нас будет готовый базис для дальнейших работ.

3.3 Разработка тестовых сценариев

Исходя из общепринятых практик, каждый тестовый случай должен иметь три части.

PreConditions - список действий, которые позволят системе успешно подлежать основной проверке. Или же это могут быть условия, выполнение которых скажут о том, что система готова для основной проверки.

Test Case Description - список действий, который переводит систему из состояния в состояние, который нацелен на получение результата, на основании которого можно говорить о том, удовлетворяет ли поставленным требованиям реализация.

PostConditions - список действий, который переводит систему в изначальное состояние. Такое состояние называют initial state.

Также виды тестовых случаев делят на два типа:

- Позитивный тест кейс. Использует только корректные (валидные) данные и проверяет, что приложение правильно выполнило вызываемую функцию.
- Негативный тест кейс. Здесь проверяются и валидные, и не валидные данные. Цель заключается в проверке исключительных ситуаций, таких как срабатывание валидаторов.

Стоит сказать, что осуществить покрытия тестами в 100% невозможно, так как это обусловлено сложностью современных web-приложений. Поэтому будут выделены сегменты, подлежащие автоматизированному тестированию.

Пример тестовых сценариев в приложении Б.

3.4 Разработка тестовых скриптов

Ранее упоминалось, что было принято решение не использовать инструменты по автоматическому созданию тестовых скриптов. Все тестовые сценарии будут реализованы вручную с помощью фреймворка `pytest` и вот почему использован именно он:

- Независимость от API (no boilerplate);
- Подробный отчет;
- Удобный `assert` (стандартный из Python);
- Динамические фикстуры всех уровней, которые могут вызываться как автоматически, так и для конкретных тестов;
- Дополнительные возможности фикстур (возвращаемое значение, финализаторы, область видимости, объект `request`, автоиспользование, вложенные фикстуры);
- Параметризация тестов, то есть запуск одного и того же теста с разными наборами параметров;
- Метки (marks), позволяющие пропустить любой тест, пометить тест, как падающий;
- Плагины;
- Возможность запуска тестов, написанных на `unittest` и `nose`, то есть полная обратная совместимость с ними;

Так как будет увеличиваться количество пакетов в проекте, то необходимо зафиксировать используемые пакеты. Это делается для того, чтобы можно было быстро переключаться в свежее виртуальное окружение, а также работать нескольким людям над одним проектом, получая одинаковые результаты. Пакеты будут отображены в requirements.

В тестовых скриптах будут использоваться assert-методы. Они проверяют любую конструкцию и возвращают true или false.

```
assert user_is_authorized(), "User is guest"
```

Рисунок 3 – метод assert

Один из важных приемов в правильном написании скриптов, являются фикстуры. Но так как наш проект будет расти, будут использоваться фикстуры продвинутым методом. Фикстуры будут задаваться глобально и передаваться в методы как параметры. Создадим метод browser и укажем, что он является фикстурой с помощью декоратора @pytest.fixture. После этого мы можем вызывать эту фикстуру в тестах, передав ее как параметр. Кроме того, фикстурам нужно задать область покрытия. Для этого используется scope. Чтобы запускать фикстуру без явного вызова, в параметрах используют autouse.

```
@pytest.fixture(scope="class")  
def browser():
```

Рисунок 4 – Фикстуры и декораторы

Также, чтобы оперативная память не заканчивалась при большом количестве тестов, нужно использовать финализаторы yield. После того, как завершится тест, в котором вызывались фикстуры, выполнение фикстуры продолжится со строки, после yield.

Чтобы запускать тесты выборочно были применены метки, они указываются в декораторах. Используются, к примеру, если тесты разделяются на regression и smoke. Существует базовый список меток:

- @pytest.mark.parametrize — для параметризации тестов;
- @pytest.mark.xfail – помечает, что тест должен не проходить и PyTest будет воспринимать это, как ожидаемое поведение;

- @pytest.mark.skipif – позволяет задать условие при выполнении которого тест будет пропущен;
- @pytest.mark.usefixtures – позволяет перечислить все фикстуры, вызываемые для теста.

Существуют плагины перезапусков. Они нужны для перезапуска упавших тестов, чтобы убедиться, что он действительно был найден баг, а не было случайного падения теста. Для этих целей был выбран плагин pytest-rerunfailures.

Очень важно настроить локализацию, так как тестирование должно проводиться на нескольких языках. Нужно сделать так, чтобы сервер сам решал, какой язык интерфейса нужно отобразить, основываясь на данных браузера. Браузер передает данные о языке пользователя через запросы к серверу, указывая в Headers (заголовке запроса) параметр accept-language. Если сервер получит запрос с заголовком {accept-language: ru, en}, то он отобразит пользователю русскоязычный интерфейс сайта. Для указания языка браузера с помощью WebDriver используется класс Options и метод add_experimental_option.

```
from selenium.webdriver.chrome.options import Options

options = Options()
options.add_experimental_option('prefs', {'intl.accept_languages': user_language})
browser = webdriver.Chrome(options=options)
```

Рисунок 5 – Локализация в Chrome

```
fp = webdriver.FirefoxProfile()
fp.set_preference("intl.accept_languages", user_language)
browser = webdriver.Firefox(firefox_profile=fp)
```

Рисунок 6 – Локализация в Firefox

Еще одним немаловажным решением в тестирования является кроссбраузерность. В тесты будет включена возможность тестирования нескольких браузеров, но в некоторых случаях будет нужно переработать код программы для конкретных браузеров.

Page Object Model или кратко Page Object - это паттерн программирования, который очень популярен в автоматизации тестирования и является одним из стандартов при автоматизации тестирования веб-продуктов. Это также один из удобных способов структурировать свой код таким образом, чтобы его было удобно

поддерживать, менять и работать с ним. Для этого нам понадобятся навыки работы ООП.

Основная идея состоит в том, что каждую страницу веб-приложения можно описать в виде объекта класса. Способы взаимодействия пользователя со страницей можно описать с помощью методов класса. В идеале тест, который будет использовать Page Object, должен описывать бизнес-логику тестового сценария и скрывать Selenium-методы взаимодействия с браузером и страницей. При изменениях в верстке страницы не придется исправлять тесты, связанные с этой страницей. Вместо этого нужно будет поправить только класс, описывающий страницу. Для этого создадим папку и файлы типа `base_page`, `main_page` и т.д., в которых будут наследоваться все остальные классы, а так как в класс нужно будет добавлять методы, существуют конструкторы. Конструктор – метод, который вызывается, когда создается объект. Его объявляют, как `__init__`.

```
class BasePage():
    def __init__(self, browser, url):
        self.browser = browser
        self.url = url

    def open(self):
        self.browser.get(self.url)
```

Рисунок 7 – `base_page`

Мы стремимся к тому, что все проверки тоже становятся отдельными методами. Остается только описание шагов.

```
def test_add_to_cart(browser):
    page = ProductPage(url="", browser) # инициализируем объект Page Object
    page.open() # открываем страницу в браузере
    page.should_be_add_to_cart_button() # проверяем что есть кнопка добавления в корзину
    page.add_product_to_cart() # жмем кнопку добавить в корзину
    page.should_be_success_message() # проверяем что есть сообщение с нужным текстом
```

Рисунок 8 – Реализация Page Object

3.5 Проектирование алгоритма автоматизации процесса тестирования web-приложения

На данном этапе необходимо проанализировать процесс тестирования, после чего создаем алгоритм автоматизации тестирования. Так как проект не создается в крупной компании, то процесс тестирования немного проще.

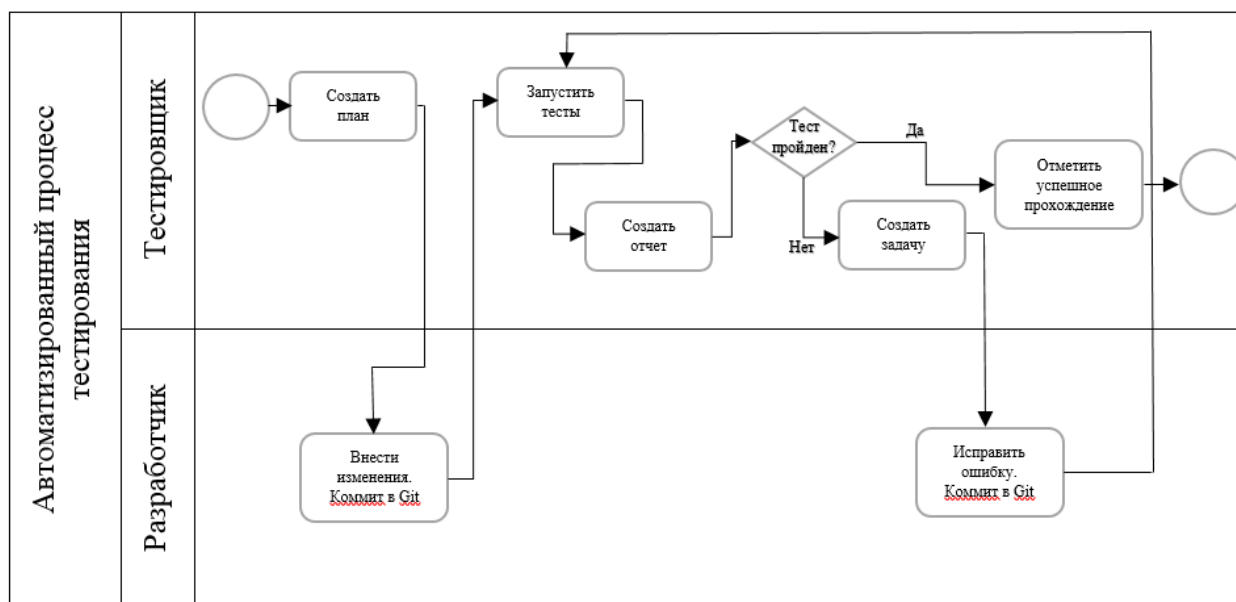


Рисунок 9 – Диаграмма автоматизированного процесса тестирования web-приложения

Чтобы еще больше снизить нагрузку для тестировщика, существуют серверы непрерывной интеграции и другие инструменты для экономии времени. Но, как упоминалось выше, проект не был создан в крупной компании, поэтому экономия небольшого времени не критична.

3.6 Реализация автоматизации тестирования web-приложения

Часть сборки уже упоминалась выше. Осталось упомянуть, что необходимо добавить в проект файл `conftest.py`. Он нужен для хранения фикстур и хранения глобальных настроек.

В файле `pytest` хранятся параметры конфигураций.

Используемые пакеты проекта хранятся в `requirements.txt`.

Для работы относительных импортов в проект добавлен файл `__init__.py`.

Также стоит скачать файлы `gitignore`, для того чтобы в репозиторий не грузились файлы кэша.

Остальные файлы типа `test_main_page.py`, `main_page.py` содержат в себе основную архитектуру тестовых скриптов.

Выполнение тестов основано на составленных тестовых сценариях, указанных в приложении Б.

Отчет о проведении тестов указан ниже.

```

===== test session starts =====
platform win32 -- Python 3.8.2, pytest-5.1.1, py-1.8.1, pluggy-0.13.1 -- c:\user
s\viktor\environments\selenium_env\scripts\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\Viktor\environments\finalproject, inifile: pytest.ini
plugins: rerunfailures-7.0
collected 5 items

test_main_page.py::test_guest_can_go_to_login_page PASSED [ 20%]
test_main_page.py::test_guest_can_write_email PASSED [ 40%]
test_main_page.py::test_guest_can_write_password PASSED [ 60%]
test_main_page.py::test_guest_can_login PASSED [ 80%]
test_main_page.py::test_guest_can_logout PASSED [100%]

===== 5 passed in 18.91s =====

```

Рисунок 10 – Результаты тестирования логирования часть 1

```

===== test session starts =====
platform win32 -- Python 3.8.2, pytest-5.1.1, py-1.8.1, pluggy-0.13.1 -- c:\user
s\viktor\environments\selenium_env\scripts\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\Viktor\environments\finalproject, inifile: pytest.ini
plugins: rerunfailures-7.0
collected 6 items

test_main_page.py::test_guest_can_go_to_login_page PASSED [ 16%]
test_main_page.py::test_guest_can_write_email PASSED [ 33%]
test_main_page.py::test_guest_can_write_password PASSED [ 50%]
test_main_page.py::test_guest_can_rewrite_password PASSED [ 66%]
test_main_page.py::test_guest_can_signin PASSED [ 83%]
test_main_page.py::test_guest_can_logout PASSED [100%]

===== 6 passed in 21.61s =====

```

Рисунок 11 – Результаты тестирования логирования часть 2

```

===== test session starts =====
platform win32 -- Python 3.8.2, pytest-5.1.1, py-1.8.1, pluggy-0.13.1 -- c:\user
s\viktor\environments\selenium_env\scripts\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\Viktor\environments\finalproject, inifile: pytest.ini
plugins: rerunfailures-7.0
collected 5 items

test_main_page.py::test_guest_can_go_to_login_page PASSED [ 20%]
test_main_page.py::test_guest_can_press_allproducts_btn PASSED [ 40%]
test_main_page.py::test_guest_can_press_addproduct_btn PASSED [ 60%]
test_main_page.py::test_guest_can_press_basket_btn PASSED [ 80%]
test_main_page.py::test_guest_can_press_checkout_btn PASSED [100%]

===== 5 passed in 21.14s =====

```

Рисунок 12 – Результаты тестирования корзины покупок

```

===== test session starts =====
platform win32 -- Python 3.8.2, pytest-5.1.1, py-1.8.1, pluggy-0.13.1 -- c:\user
s\viktor\environments\selenium_env\scripts\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\Viktor\environments\finalproject, inifile: pytest.ini
plugins: rerunfailures-7.0
collected 5 items

test_main_page.py::test_guest_can_go_to_login_page PASSED [ 20%]
test_main_page.py::test_guest_can_press_allproducts_btn PASSED [ 40%]
test_main_page.py::test_guest_can_press_addproduct_btn PASSED [ 60%]
test_main_page.py::test_guest_can_press_basket_btn PASSED [ 80%]
test_main_page.py::test_guest_can_press_checkout_btn FAILED [100%]

===== FAILURES =====
C:\Users\Viktor\environments\finalproject\test_main_page.py:26: TypeError: __ini
t__(<) missing 1 required positional argument: 'url'
===== 1 failed, 4 passed in 20.78s =====

```

Рисунок 13 – Пример ошибки в тесте

По результатам тестирования видно, что все тестовые случаи пройдены успешно. Также указано время выполнения каждого сегмента проверок.

4 Результаты проведенного исследования

Реализован алгоритм автоматизации тестирования web-приложения. Такой вид тестирования экономит время тестировщиков. Помимо этого, время проведения тестов можно ускорить, если есть мощное оборудование и высокоскоростной интернет.

В результате работы не было выявлено ошибок выполнения тестирования, что говорит о работоспособности сегментов приложения, которые были включены в тестовые случаи.

4.1 Оценка эффективности внедрения автоматизации тестирования web-приложения

В данном разделе показан расчет выгоды и эффективности внедрения автоматизации тестирования.

Формула расчета выгоды:

$$K = \frac{N * T * P}{L + P' * T''} \quad (1)$$

где К – коэффициент выгоды;

Н – количество версий продукта, которое планируется выпустить в ходе реализации проекта;

Т – примерное время, затрачиваемое тестировщиком на выполнение ручных тестов для одного модуля;

Р – зарплата тестировщика;

Л – стоимость лицензии на средство автоматизации;

Т' – примерное время на разработку и выполнение автоматических тестов для одного модуля специалистом по АТ;

Р' – зарплата специалиста по АТ.

Таблица 2 – Расчет коэффициента выгоды

Показатель	Значение показателя
N	2
P, руб.	35000
T, мин.	60
L, руб.	0
T', мин.	350
P', руб.	60000
K	0,2

Невыгодность внедрения автоматизации обуславливается тем, что проект развивался недолго. В долгосрочной перспективе ручное тестирование будет уступать автоматизированному. К тому же, чем выше квалификация тестировщика, тем быстрее будут создаваться автоматизированные тесты. Более того, есть возможность создавать гибкие тесты, часть которых могут использоваться в других проектах, что значительно сокращает трудозатратность.

Формула расчета эффективности автоматизации тестирования:

$$AT_{cr} + AT_{ev} + AT_{upd} \leq MT_{val} * N, \quad (2)$$

где AT_{cr} – время создания автоматического теста;

AT_{ev} – среднее время на выяснение причины падения теста;

AT_{upd} – среднее время обновления автоматического теста;

MT_{val} – среднее время проведения ручного теста;

N – количество выполнений в течение одной итерации.

Таблица 3 - Формула расчета эффективности автоматизации тестирования

Показатель	Значение показателя
AT_{cr} , мин.	15
AT_{ev} , мин.	10
AT_{upd} , мин.	4

MT_{val} , мин.	8
N	3
$AT_{cr} + AT_{ev} + AT_{upd}$	29
$MT_{val} * N$	24

Как уже упоминалось выше, разработка автоматизированных тестов актуальна для больших проектов на большом промежутке времени. Преимущества ручного тестирования будут нивелированы после какого-то времени.

5 Социальная ответственность

5.1 Введение

Разработанный в рамках исследовательской работы проект по автоматизированным тестам web-приложений для автоматизации рабочего процесса, является одной из важных отраслей в QA. Пользователями данного программного обеспечения являются инженеры компании. Взаимодействие пользователя (тестировщика-программиста) с разработанной программой производится с помощью программных и аппаратных средств. Рабочая зона разработчика находится в учебной аудитории КЦ г. Томска.

Таблица 4 – средняя температура в Томске по временам года

	Зима	Весна	Лето	Осень
Средняя температура (°C)	-16.1	1,2	17,3	0,9

Социальная направленность данного проекта заключается в том, что минимизировано воздействие вредных факторов на человека, поскольку самому человеку в тестировании приложения участвовать не нужно.

Данный раздел посвящен анализу вредных и опасных факторов производственной среды, оценке чрезвычайных ситуаций, разработки мероприятий по снижению негативных факторов для операторов ЭВМ и, в частности, для инженеров компании, которые будут использовать разработанный продукт.

5.2 Правовые и организационные вопросы обеспечения безопасности

5.2.1 Правовые нормы трудового законодательства для рабочей зоны оператора ЭВМ

Регулирование отношений между работником и работодателем, касающихся оплаты труда, трудового распорядка, особенности регулирования труда женщин, детей, людей с ограниченными способностями и проч., осуществляется законодательством РФ, а именно трудовым кодексом РФ [6].

Продолжительность рабочего дня не должна быть меньше указанного времени в договоре, но не больше 40 часов в неделю. Для работников до 16 лет – не более 24 часов в неделю, от 16 до 18 лет и инвалидов I и II группы – не более 35 часов.

Возможно установление неполного рабочего дня для беременной женщины; одного из родителей (опекуна, попечителя), имеющего ребенка в возрасте до четырнадцати лет (ребенка-инвалида в возрасте до восемнадцати лет). Оплата труда при этом производится пропорционально отработанному времени, без ограничений оплачиваемого отпуска, исчисления трудового стажа и других прав.

При работе в ночное время продолжительность рабочей смены сокращается на один час. К работе в ночную смену не допускаются беременные женщины; работники, не достигшие возраста 18 лет; женщины, имеющие детей в возрасте до трех лет, инвалиды, работники, имеющие детей-инвалидов, а также работники, осуществляющие уход за больными членами их семей в соответствии с медицинским заключением, матери и отцы-одиночки детей до пяти лет.

Организация обязана предоставлять ежегодный отпуск продолжительностью 28 календарных дней. Дополнительные отпуска предоставляются работникам, занятым на работах с вредными или опасными условиями труда, работникам имеющими особый характер работы, работникам с ненормированным рабочим днем и работающим в условиях Крайнего Севера и приравненных к нему местностях.

В течение рабочего дня работнику должен быть предоставлен перерыв для отдыха и питания продолжительностью не более двух часов и не менее 30 минут, который в рабочее время не включается. Всем работникам предоставляются выходные дни, работа в выходные дни осуществляется только с письменного согласия работника.

Организация-работодатель выплачивает заработную плату работникам. Возможно удержание заработной платы только в случаях установленных ТК РФ ст. 137. В случае задержки заработной платы более чем на 15 дней, работник имеет право приостановить работу, письменно уведомив работодателя.

Законодательством РФ запрещена дискриминация по любым признакам и принудительный труд [6].

5.2.2 Организационные мероприятия при компоновке рабочей зоны

К мероприятиям, относящимся к компоновке рабочей зоны относятся работы по организации рабочего места пользователя, позволяющие наилучшим образом организовать деятельность работника, делая его работу максимально удобной и безопасной.

Основным направлением использования разработанной программной системы является сбор данных с серверов телемеханики компании. Сам продукт не влияет на организацию рабочей зоны.

Требования к помещениям для работы с ПЭВМ регламентируются в соответствии с СанПиН 2.2.2/2.4.1340-03. В документе указаны нормы помещениям для работы с ПЭВМ, норма площади рабочего места с персональным компьютером составляет 4,5 квадратных метра .

Разработанный программный продукт не влияет на организацию рабочей зоны, однако работа с ним позволит реорганизовать работу специалистов, что в свою очередь повлияет на организацию рабочей зоны. Это может быть охарактеризовано с помощью следующих факторов:

- Сокращение времени передачи данных;
- Удобный интерфейс для работы с данными;
- Уменьшение загруженности серверных ресурсов и как следствие уменьшение количества сбоев.

Все перечисленные факторы повышают, облегчают работу и положительно сказываются на производительности труда [7].

5.3 Производственная безопасность

Для обеспечения производственной безопасности необходимо проанализировать воздействия на человека вредных и опасных производственных факторов, которые могут возникать при разработке или эксплуатации проекта.

Производственный фактор считается вредным, если воздействие этого фактора на работника может привести к его заболеванию. Производственный фактор считается опасным, если его воздействие на работника может привести к его травме.

Все производственные факторы классифицируются по группам элементов: физические, химические, биологические и психофизические. Для данной работы целесообразно рассмотреть физические и психофизические вредные и опасные факторы производства, характерные как для рабочей зоны программиста, как разработчика рассматриваемой в данной работе системы, так и для рабочей зоны пользователя готового продукта – инженера-оператора ПЭВМ [8]. Выявленные факторы представлены в таблица 5.

Таблица 5 – Вредные и опасные производственные факторы при выполнении работ за ПЭВМ

Факторы (ГОСТ 12.0.003-2015)	Этапы работ			Нормативные документы
	Разработка	Изготовление	Эксплуатация	
1.Отклонение показателей микроклимата	+	+	+	1) СанПиН 2.2.4.548-96 [8]; 2) СП 52.13330.2016 [9]; 3) ГОСТ Р 12.1.019-2009 ССБТ [11]; 4) Р 2.2.2006-05 [14]
2.Отсутствие или недостаток естественного света	+	+	+	
3. Опасность поражения электрическим током	+	+	+	
4. Монотонность рабочего процесса	+	+	-	

5.3.1 Анализ опасных и вредных производственных факторов

Отклонение показателей микроклимата

Микроклимат помещения – это комплекс физических факторов внутренней среды помещения, которые оказывают влияние на здоровье человека. Основные факторы, характеризующие микроклимат помещения, устанавливаются в соответствии с СанПиН 2.2.4.548 – 96 [9]. К ним относятся:

- температура воздуха;
- скорость движения воздуха;
- влажность;

- интенсивность теплового облучения.

Согласно вышеуказанному документу, работа разработчика-программиста относится к категории работ 1б, так как основная часть работы происходит с использованием ПЭВМ.

Показатели микроклимата разделяются на допустимые значения и оптимальные значения микроклимата. При допустимых значениях работник может ощущать небольшой дискомфорт и понижение работоспособности, при этом ухудшение состояния здоровья возникать не будет. При оптимальных значениях наблюдается высокий уровень работоспособности и обеспечивается нормальное состояние организма работника.

Микроклимат с пониженной температурой приводит к обострению язвенной болезни, радикулита, обуславливает возникновение заболеваний органов дыхания, сердечно-сосудистой системы. Охлаждение человека (как общее, так и локальное) приводит к изменению его двигательной реакции, нарушает координацию и способность выполнять точные операции, вызывает тормозные процессы в коре головного мозга, что может быть причиной возникновения различных форм травматизма. При локальном охлаждении кистей снижается точность выполнения рабочих операций.

Микроклимат с повышенной температурой вызывает нарушение состояния здоровья, снижение работоспособности и производительности труда, может привести к заболеванию общего характера, которое проявляется чаще всего в виде теплового коллапса. Он возникает вследствие расширения сосудов и уменьшения давления в них крови. Обморочному состоянию предшествует головная боль, чувство слабости, головокружение, тошнота.

Допустимые и оптимальные значения показателей микроклимата холодного и тёплого периода года для категории работ 1б представлены в **Ошибка! Источник с ссылки не найден..**

Таблица 6 – Оптимальные и допустимые значения показателей микроклимата (СанПин 2.2.4.548-96)

Тип величины	Период года	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
--------------	-------------	-------------------------	------------------------------	------------------------------------	--------------------------------

Оптимальный	Холодный	21-23	20-24	40-60	0,1
	Тёплый	22-24	21-25		
Допустимый	Холодный	19-24	18-25	15-75	0,1-0,2
	Тёплый	20-28	19-29		0,1-0,3

В рассматриваемой аудитории (г. Томск, КЦ, 407) для обеспечения оптимальной температуры помещения используется регулируемое отопление на основе регулируемой системы водяного отопления. Так же в аудитории установлен кондиционер LG S12SWT класса «зима-лето», который можно использовать в холодный период для обогрева помещения, а в тёплый для его охлаждения. Помимо этого, во время перерывов осуществляется естественная вентиляция воздуха, то есть проветривание помещения. Всё это обеспечивает высокую производительность труда, а также нормальное состояние здоровья работников в аудитории.

Отсутствие или недостаток естественного света

Рабочее (общее) освещение – это основное освещение, которое обеспечивает нормальные условия для нахождения человека в помещении.

Большую роль в создании благоприятных условий для работоспособности на предприятии играет освещение. Отсутствие хорошего освещения может привести к профессиональным заболеваниям, а также ухудшению концентрации работников. Работа разработчика-программиста в основном проводится за дисплеем персонального компьютера. В результате недостаточной освещённости рабочего места у работника может ухудшиться зрение, а также возникнуть переутомление. Тоже самое происходит и при избыточном освещении помещения.

Поэтому необходимо создать систему освещения рабочего места для комфортной работы, отвечающую нормам, указанным в соответствии со СП 52.13330.2016 [10]. Показатель, отвечающий за качество освещения, называется освещённостью и обозначается буквой Е. Согласно вышеуказанному документу, освещённость рабочего места должна быть равна 200 лк, так как работа программиста за компьютером относится к 3 категории зрительной работы.

Опасность поражения электрическим током

Поражение электрическим током является опасным производственным фактором и, поскольку оператор ПЭВМ имеет дело с электрооборудованием, то

вопросам электробезопасности на его рабочем месте должно уделяться много внимания. Нормы электробезопасности на рабочем месте регламентируются ГОСТ Р 12.1.019-2009 ССБТ [6], вопросы требований к защите от поражения электрическим током освещены в ГОСТ Р 12.1.019-2009 ССБТ.

Электробезопасность – система организационных и технических мероприятий и средств, обеспечивающих защиту людей от вредного и опасного воздействия электрического тока, электрической дуги, электромагнитного поля и статического электричества.

Опасность поражения электрическим током усугубляется тем, что человек не в состоянии без специальных приборов обнаружить напряжение дистанционно.

Помещение, где расположено рабочее место оператора ПЭВМ, относится к помещениям без повышенной опасности ввиду отсутствия следующих факторов: сырость, токопроводящая пыль, токопроводящие полы, высокая температура, возможность одновременного прикосновения человека к имеющим соединение с землей металлоконструкциям зданий, технологическим аппаратам, механизмам и металлическим корпусам электрооборудования.

Для оператора ПЭВМ при работе с электрическим оборудованием обязательны следующие меры предосторожности:

- Перед началом работы нужно убедиться, что выключатели и розетка закреплены и не имеют оголённых токоведущих частей;
- При обнаружении неисправности оборудования и приборов необходимо, не делая никаких самостоятельных исправлений, сообщить человеку, ответственному за оборудование.

Опасность поражения электрическим током

Монотонность – однообразное повторение рабочих операций. Р 2.2.2006-05 Гигиена труда [14]. Опасность монотонности заключается в снижении внимания к процессу производства, быстрой утомляемости и снижении интереса к трудовому процессу, что влияет на безопасность труда в целом. Одной из форм, предрасполагающей к формированию монотонности, является автоматизм – деятельность, осуществляемая без непосредственного участия сознания. Он

формируется в результате нескольких факторов: многолетнего опыта деятельности, рутинности работы, отсутствия вовлеченности в трудовой процесс, воображения и творческого подхода, физических перегрузок. Особое значение это имеет на сложных производствах или производствах с вредными условиями труда, где аккуратность и внимание имеют решающее значение. Монотонность сопровождается скукой, апатией к выполнению трудовой деятельности. Но нельзя точно определить, что выполнение именно этих действий является монотонным и скучным занятием. Каждый человек сам для себя определяет род своей деятельности и дает ей свою объективную оценку. К примеру, один сотрудник, работающий на конвейере, считает свою работу монотонной и скучной, а другой, наоборот, считает ее очень интересной. Многие люди, занимающиеся динамичной, активной работой, которую никак нельзя назвать монотонной, считают ее скучной, неинтересной.

Основными отрицательными последствиями монотонного труда являются:

- снижение работоспособности и производительности труда;
- ухудшение функционального состояния организма работающих;
- повышенная заболеваемость;
- снижение творческой инициативы;
- производственный травматизм.

Необходимо разрабатывать мероприятия, направленные на уменьшение отрицательных последствий монотонии. К ним относятся:

- автоматизация однообразного ручного труда;
- оптимизация содержания труда, темпа и ритма работы;
- совмещение профессий и чередование операций;
- внедрение рациональных режимов труда и отдыха с введением 5-минутных регламентированных перерывов через каждый час работы с целью снятия фактора М.;
- рациональная организация рабочего места;

- введение в режим рабочего дня комплексов производственной гимнастики, функциональной музыки и др.

5.4 Экологическая безопасность

5.4.1 Анализ влияния процесса исследования на окружающую среду

В данном разделе рассматривается воздействие на окружающую среду деятельности по разработке проекта, а также самого продукта в результате его реализации на производстве.

Разработка программного обеспечения и работа за ПЭВМ не являются экологически опасными работами, потому объект, на котором производилась разработка продукта, а также объекты, на которых будет производиться его использование операторами ПЭВМ относятся к предприятиям пятого класса, размер селитебной зоны для которых равен 50 м [12].

Программный продукт, разработанный непосредственно в ходе выполнения бакалаврской работы, не наносит вреда окружающей среде.

Средства, необходимые для разработки и эксплуатации программного комплекса могут наносить вред окружающей среде.

5.4.2 Обоснование мероприятий по защите окружающей среды

Современные ПЭВМ производят практически без использования вредных веществ, опасных для человека и окружающей среды. Исключением являются аккумуляторные батареи компьютеров и мобильных устройств. В аккумуляторах содержатся тяжелые металлы, кислоты и щелочи, которые могут наносить ущерб окружающей среде, попадая в гидросферу и литосферу, если они были неправильно утилизированы. Для утилизации аккумуляторов необходимо обращаться в специальные организации, специализировано занимающиеся приемом, утилизацией и переработкой аккумуляторных батарей [12].

Люминесцентные лампы, применяющиеся для искусственного освещения рабочих мест, также требуют особой утилизации, т.к. в них присутствует от 10 до 70 мг ртути, которая относится к чрезвычайно-опасным химическим веществам и может стать причиной отравления живых существ, а также загрязнения атмосферы, гидросферы и литосферы. Сроки службы таких ламп составляют около 5-ти лет, после

чего их необходимо сдавать на переработку в специальных пунктах приема. Юридические лица обязаны сдавать лампы на переработку и вести паспорт для данного вида отходов [12].

5.5 Безопасность в чрезвычайных ситуациях

5.5.1 Анализ вероятных ЧС, которые могут возникнуть на рабочем месте при проведении исследований

В рабочей среде оператора ПЭВМ возможно возникновение следующих чрезвычайных ситуаций техногенного характера:

- Пожары и взрывы в зданиях и на коммуникациях;
- Внезапное обрушение зданий.

Среди возможных стихийных бедствий можно выделить метеорологические (ураганы, ливни, заморозки), гидрологические (наводнения, паводки, подтопления), природные пожары.

К чрезвычайным ситуациям биолого-социального характера можно отнести эпидемии, эпизоотии, эпифитотии.

Экологические чрезвычайные ситуации могут быть вызваны изменениями состояния, литосферы, гидросферы, атмосферы и биосферы в результате деятельности человека [13].

Наиболее характерной для объекта, где размещаются рабочие помещения, оборудованные ПЭВМ, чрезвычайной ситуацией является пожар.

Помещение для работы операторов ПЭВМ по системе классификации категорий помещений по взрывопожарной и пожарной опасности относится к категории Д (из 5-ти категорий А, Б, В1-В4, Г, Д), т.к. относится к помещениям с негорючими веществами и материалами в холодном состоянии [13].

5.5.2 Обоснование мероприятий по предотвращению ЧС и разработка порядка в случае возникновения ЧС

Каждый сотрудник организации должен быть ознакомлен с инструкцией по пожарной безопасности, пройти инструктаж по технике безопасности и строго соблюдать его.

Запрещается использовать электроприборы в условиях, не соответствующих требованиям инструкций изготовителей, или имеющие неисправности, которые в соответствии с инструкцией по эксплуатации могут привести к пожару, а также эксплуатировать электропровода и кабели с поврежденной или потерявшей защитные свойства изоляцией. Электроустановки и бытовые электроприборы в помещениях по окончании рабочего времени должны быть обесточены (вилки должны быть вынуты из розеток). Под напряжением должны оставаться дежурное освещение и пожарная сигнализация. Недопустимо хранение легковоспламеняющихся, горючих и взрывчатых веществ, использование открытого огня в помещениях офиса.

Перед уходом из служебного помещения работник обязан провести его осмотр, закрыть окна, и убедиться в том, что в помещении отсутствуют источники возможного возгорания, все электроприборы отключены и выключено освещение. С периодичностью не реже одного раза в три года необходимо проводить замеры сопротивления изоляции токоведущих частей силового и осветительного оборудования.

Повышение устойчивости достигается за счет проведения соответствующих организационно-технических мероприятий, подготовки персонала к работе в ЧС [13].

Работник при обнаружении пожара или признаков горения (задымление, запах гари, повышение температуры и т.п.) должен:

- Немедленно прекратить работу и вызвать пожарную охрану по телефону «01», сообщив при этом адрес, место возникновения пожара и свою фамилию;
- Принять по возможности меры по эвакуации людей и материальных ценностей;
- Отключить от сети закрепленное за ним электрооборудование;
- Приступить к тушению пожара имеющимися средствами пожаротушения;
- Сообщить непосредственному или вышестоящему начальнику и оповестить окружающих сотрудников;
- При общем сигнале опасности покинуть здание согласно «Плану эвакуации людей при пожаре и других ЧС».

Для тушения пожара применять ручные углекислотные огнетушители (типа ОУ-2, ОУ-5), находящиеся в помещениях офиса, и пожарный кран внутреннего противопожарного водопровода. Они предназначены для тушения начальных возгораний различных веществ и материалов, за исключением веществ, горение которых происходит без доступа воздуха. Огнетушители должны постоянно содержаться в исправном состоянии и быть готовыми к действию. Категорически запрещается тушить возгорания в помещениях офиса при помощи химических пенных огнетушителей (типа ОХП-10) [13].

5.6 Выводы по разделу

В ходе выполнения данного раздела были рассмотрены и проанализированы различные факторы, негативно влияющие на рабочие процессы, рассмотрены их основные источники возникновения и выдвинуты меры по их снижению и предотвращению. Также проведен анализ с точки зрения правовой организации процесса исследования и эксплуатации программного продукта. В заключении был проведен анализ экологической безопасности исследования и меры предохранения от наиболее значимых чрезвычайных ситуаций.

6 Финансовый менеджмент, ресурсоэффективность и ресурсопотребление

6.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения

6.1.1 Потенциальные потребители результатов исследования

Исследуемый проект состоит в написании автотестов – программ, заложенных в них изначально тесткейсов.

Разработка сайта подразумевает под собой решение множества задач из совершенно разных областей, таких как: работа с большими объемами данных, вычислительные системы, сетевая безопасность, дизайн и прочее. Тестирование производится на различных этапах разработки, но в этой работе будет затронута только проблема тестирования интерфейса интернет-сервиса.

Создание собственных сайтов играет существенную роль в большинстве сфер бизнеса; более того, монетизационная модель множества компаний, таких как

Яндекс, Google, заключается исключительно в получении прибыли от интернет-сервисов за счет рекламы, платного контента, пожертвований пользователей и т. п. Интернет-сервисы могут приносить огромную прибыль, так что конкуренция в этой сфере очень высока. Даже небольшие ошибки, допущенные при разработке, могут повлиять на выбор пользователя относительно того, какой из конкурирующих сервисов ему предпочесть. Поэтому, как и во многих других областях, тестирование играет очень важную роль и в процессе создания веб-сайтов и разработке интернет-сервисов. Стоит отметить и то, что автоматизация тестирования актуальна не только для крупных компаний, но и для меньших компаний, чьи проекты нацелены на долгосрочную перспективу.

6.1.2 Анализ конкурентных технических решений

В качестве конкурентных решений можно выделить:

- процесс ручного тестирования пользовательского интерфейса, когда программист самостоятельно выполняет действия, заранее перечисленные в плане тестирования;
- процесс тестирования, являющейся формой программирования на языках, специально разработанных для АТ ПО. Разработкой занимаются программисты высокого уровня, которые работают отдельно от тестировщиков, непосредственно запускающих тесты.

Анализ проведен с использованием оценочной карты. Результаты проведения анализа представлены в таблице 1.

Анализ конкурентных технических решений определяется по формуле:

$$K = \sum B_i \cdot B_i, \quad (3)$$

где K – конкурентоспособность научной разработки или конкурента;

B_i – вес показателя (в долях единицы);

B_i – балл i -го показателя

Таблица 7 - Перечень работ и продолжительность их выполнения

Критерии оценки		Баллы	Конкуренто-способность
-----------------	--	-------	------------------------

	Вес критерия	B_{Φ}	$B_{\kappa 1}$	$B_{\kappa 2}$	K_{Φ}	$K_{\kappa 1}$	$K_{\kappa 2}$
1	2	3	4	5	6	7	8
Технические критерии оценки ресурсоэффективности							
1. Повышение производительности труда пользователя	0.26	5	3	4	1.3	0.78	1.04
2. Удобство в эксплуатации (соответствует требованиям потребителей)	0.135	5	4	4	0.675	0.54	0.54
3. Открытость исходного кода тестируемой программы	0.02	4	3	5	0.08	0.06	0.1
4. Кроссплатформенность	0.05	3	5	3	0.15	0.25	0.15
5. Потребность в ресурсах памяти	0.045	4	3	4	0.18	0.135	0.18
6. Простота эксплуатации	0.12	5	5	4	0.6	0.6	0.48
7. Скорость работы	0.13	4	4	4	0.52	0.52	0.52
Экономические критерии оценки эффективности							
1. Конкурентоспособность продукта	0.13	3	2	5	0.39	0.26	0.65
2. Уровень проникновения на рынок	0.02	1	3	4	0.02	0.06	0.08
3. Цена	0.06	4	5	4	0.24	0.3	0.24
4. Предполагаемый срок эксплуатации	0.015	4	1	4	0.06	0.015	0.06
5. Финансирование научной разработки	0.015	2	1	5	0.03	0.015	0.075
Итого	1				4.25	3.52	4.16

Исходя из полученных данных, можно сделать вывод о том, что технологии автоматизации имеют некоторые преимущества. Это обусловлено повышенной производительностью труда, удобством и простотой в эксплуатации. Однако технология уступает в стоимости средств разработки и низкой надежностью.

6.1.3 SWOT-анализ

Основываясь на проведенных ранее анализах рынка и конкурентных

технических решений произведем SWOT-анализа для выявления угроз и возможностей, сильных и слабых сторон проекта, который показан в таблице ниже (Ошибка! Источник ссылки не найден.).

Таблица 8 – SWOT-анализ

Внешние факторы		Внутренние факторы	
		Сильные стороны проекта: А1. Снижение трудозатратности тестировщиков; А2. Быстрое внедрение проекта; А3. Проект предполагает качественную реализацию;	Слабые стороны проекта: Д1. Высокая стоимость разработки; Д2. Отсутствие квалифицированного персонала; Д3. Сложность внедрения; Д4. Сложность сопровождения проекта;
	Возможности: О1. Автоматизация процесса тестирования пользовательского интерфейса; О2. Интеграция с системами управления проектами; О3. Появление дополнительного спроса на продукт;	Направления развития: Интеграция с системами управления проектами для расширения функциональных возможностей; Повышение надежности тестов; Написание статей, описывающих процесса автоматизации тестирования интерфейса пользователя;	Сдерживающие факторы: Не каждая компания готова внедрять АТ в проекты; Необходимость изменять тесты при изменении версии программы;
	Угрозы: У1. Возникновение ошибок при изменении версии программы У2. Предвзятость потребителей о пользе внедрения технологии; У3. Прекращение поддержки тестировщиком; У4. Непопулярность на рынке;	Неосведомленность потенциальных потребителей о пользе технологии; Прекращение развития проекта из-за особенностей внедрения технологии;	Отсутствие популярности из-за специфики внедрения; Невысокая степень надежности тестов;

Исходя из результатов проведения SWOT-анализа, можно сказать, что проект выгоден, однако нужно аккуратно обходиться с конкурентами, т.к. риск потери клиентов достаточно высок.

В процессе проведения SWOT-анализа выявлены слабые и сильные стороны разрабатываемой работы, так же определены внешние угрозы и возможности конкурентов в данной отрасли. Можно сказать, что проект выгоден, однако не каждая компания будет готова внедрять его. На основе анализа, определены различные мероприятия, которые позволят преодолеть возможные угрозы и трудности или улучшить текущее состояние.

6.2 Планирование научно-исследовательских работ

6.2.1 Структура работ в рамках научного исследования

План работ содержит сформированный перечень работ для выполнения исследования. Исполнителями являются студент Богер В.В. и научный руководитель Хамухин А. А.

Таблица 9 – Перечень работ и распределение исполнителей

Этапы работы	Исполнители	Загрузка исполнителей
Постановка целей и задач, получение исходных данных	НР	НР – 100%
Составление и утверждение ТЗ	НР, И	НР – 60% И – 40%
Подбор и изучение материалов по тематике	НР, И	НР – 20% И – 100%
Разработка календарного плана	НР, И	НР – 50% И – 50%
Обсуждение литературы	НР, И	НР – 30% И – 100%
Выбор структурной схемы проекта	НР, И	НР – 40% ИП – 80%
Проектирование архитектуры	И	И – 100%
Оформление расчетно-пояснительной записки	НР, И	НР – 30% ИП – 100%
Подведение итогов	НР, И	НР – 60% И – 100%

Сформированный план работ отражает жизненный цикл исследуемой работы.

6.2.2 Определение трудоемкости выполнения работ

Для определения трудоемкости выполнения работ, необходимо на основе экспертной оценки ожидаемой трудоемкости выполнения каждой работы рассчитать

длительность работ в рабочих и календарных днях для каждого из вариантов исполнения работ по следующим формулам:

$$t_{ожi} = \frac{3t_{мини} + 2t_{маxi}}{5}, \quad (4)$$

где $t_{ожi}$ – ожидаемая трудоемкость выполнения i -ой работы чел.-дн.;

$t_{мини}$ – минимально возможная трудоемкость выполнения заданной i -ой работы (оптимистическая оценка: в предположении наиболее благоприятного стечения обстоятельств), чел.-дн.;

$t_{маxi}$ – максимально возможная трудоемкость выполнения заданной i -ой работы (пессимистическая оценка: в предположении наиболее неблагоприятного стечения обстоятельств), чел.-дн.

Расчет продолжительности выполнения каждого этапа в рабочих днях (ТРД) ведется по формуле:

$$T_{РД} = \frac{t_{ож}}{K_{ВН}} \cdot K_{Д}, \quad (5)$$

где $t_{ож}$ – продолжительность работы, дн.;

$K_{ВН}$ – коэффициент выполнения работ, учитывающий влияние внешних факторов на соблюдение предварительно определенных длительностей, в частности, возможно $K_{ВН} = 1$;

$K_{Д}$ – коэффициент, учитывающий дополнительное время на компенсацию непредвиденных задержек и согласование работ ($K_{Д} = 1-1,2$; в этих границах конкретное значение принимает сам исполнитель).

Расчет продолжительности этапа в календарных днях ведется по формуле:

$$T_{КД} = T_{РД} \cdot T_{К}, \quad (6)$$

где $T_{КД}$ – продолжительность выполнения этапа в календарных днях;

T_K – коэффициент календарности, позволяющий перейти от длительности работ в рабочих днях к их аналогам в календарных днях.

$$T_K = \frac{T_{\text{кал}}}{T_{\text{кал}} - T_{\text{вых}} - T_{\text{пр}}}, \quad (7)$$

где T_K – коэффициент календарности;

$T_{\text{кал}}$ – количество календарных дней в году;

$T_{\text{вых}}$ – количество выходных дней в году;

$T_{\text{пр}}$ – количество праздничных дней в году.

В расчетах учитывается, что календарных дней в 2020 году 366, а сумма выходных и праздничных дней составляет 119 дней. Исходя из этих данных, можно рассчитать, что коэффициент календарности равен 1,48.

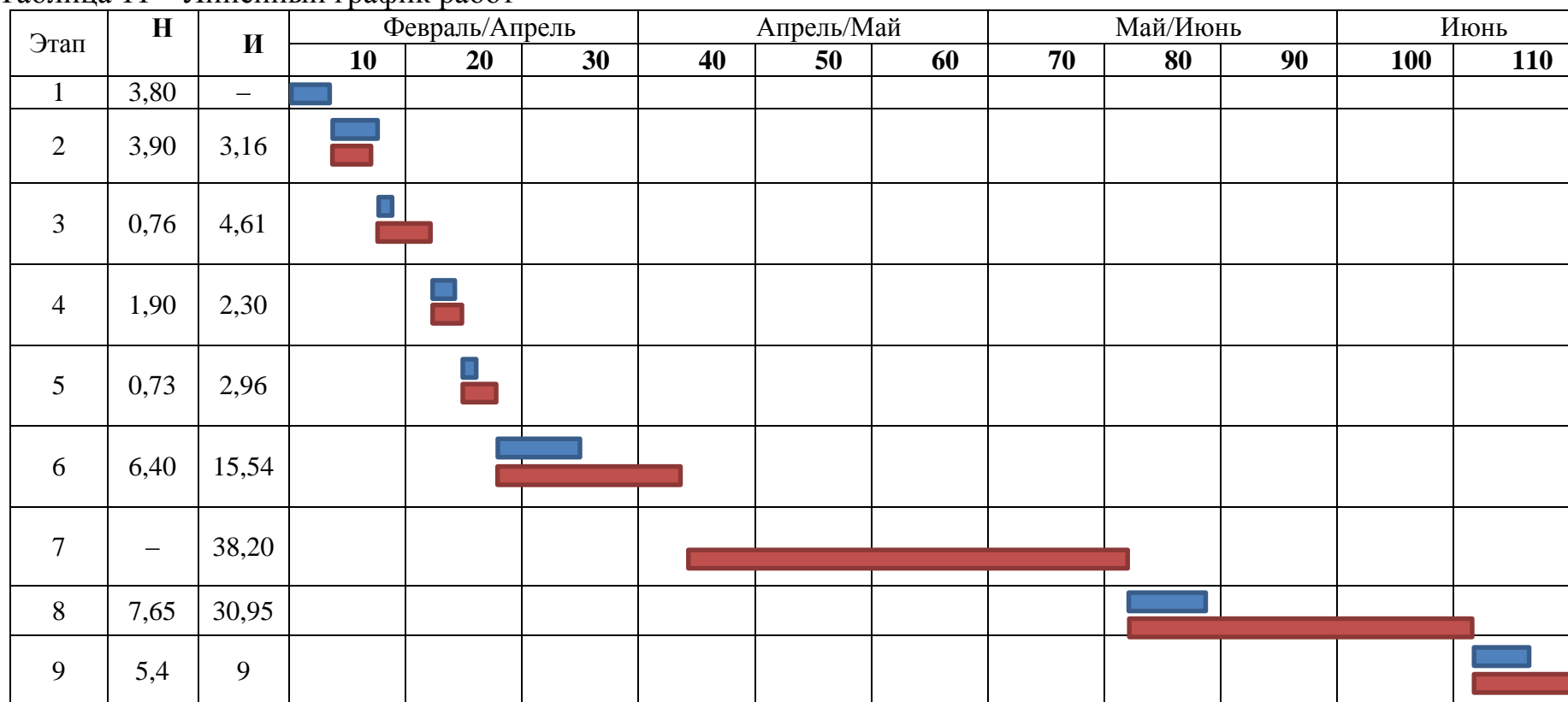
$$T_K = \frac{366}{366 - 105 - 14} = 1,48$$

В таблице 4 приведен пример определения продолжительности этапов работ и их трудоемкости по исполнителям, занятым на каждом этапе. Две последних величины далее будут использованы для определения затрат на оплату труда участников и прочие затраты. Величины трудоемкости этапов по исполнителям $T_{\text{кд}}$ (данные столбцов 8 и 9 кроме итогов) позволяют построить линейный график осуществления проекта – таблица 10.

Таблица 10 – Трудозатраты на выполнение проекта

Этап	Исполнители	Продолжительность работ, дни			Трудоемкость работ по исполнителям чел.- дн.			
					$T_{РД}$		$T_{КД}$	
		t_{min}	t_{max}	$t_{ож}$	НР	И	НР	И
1	2	3	4	5	6	7	8	9
Постановка задачи	НР	2	4	2,8	3,11	–	3,80	–
Разработка и утверждение технического задания (ТЗ)	НР, И	4	6	4,8	3,20	2,13	3,90	3,16
Подбор и изучение материалов по тематике	НР, И	2	4	2,8	0,62	3,11	0,76	4,61
Разработка календарного плана	НР, И	2	4	2,8	1,56	1,56	1,90	2,30
Обсуждение литературы	НР, И	1	3	1,80	0,60	2,00	0,73	2,96
Выбор структурной схемы проекта	НР, И	11	13	11,80	5,24	10,49	6,40	15,54
Проектирование архитектуры	И	22	25	23,20	–	25,78	–	38,20
Оформление расчетно- пояснительной записки	НР, И	18	20	18,80	6,27	20,89	7,65	30,95
Подведение итогов	НР, И	5	8	6,2	4,46	7,44	5,4	9
Итого:				75	25,06	73,4	30,54	97,72

Таблица 11 – Линейный график работ



НР И

6.2.3 Расчет материальных затрат научно-технического исследования

Во время проведения исследования использовались разные канцелярские принадлежности, производилось распечатывание материалов и документов по исследованию (**Ошибка! Источник ссылки не найден.**).

Таблица 12 – Расчет материальных затрат

Наименование материала	Единицы измерения	Количество	Цена за 1 ед., руб.	Затраты, руб.
Канцелярские принадлежности	Шт.	1	1500	1500
Итого				1500

Допустим, что ТЗР составляют 5 % от отпускной цены материалов, тогда расходы на материалы с учетом ТЗР равны $C_{\text{мат}} = 1500 * 1,05 = 1575$ руб..

6.2.4 Расчет заработной платы

Среднедневная тарифная заработная плата (ЗП_{дн-т}) рассчитывается по формуле:

$$\text{ЗП}_{\text{дн-т}} = \text{МО} / 20,58 \quad (8)$$

учитывающей, что в году 247 рабочих дней и, следовательно, в месяце в среднем 20,58 рабочих дня (при шестидневной рабочей неделе).

Пример расчета затрат на полную заработную плату приведены в таблице 13. Затраты времени по каждому исполнителю в рабочих днях с округлением до целого взяты из таблицы 4. Для учета в ее составе премий, дополнительной зарплаты и районной надбавки используется следующий ряд коэффициентов: $K_{\text{ПР}} = 1,1$; $K_{\text{доп.ЗП}} = 1,188$; $K_{\text{р}} = 1,3$. Таким образом, для перехода от тарифной (базовой) суммы заработка исполнителя, связанной с участием в проекте, к соответствующему полному заработку (зарплатной части сметы) необходимо первую умножить на интегральный коэффициент $K_{\text{и}} = 1,1 * 1,188 * 1,3 = 1,699$. Вышеуказанное значение $K_{\text{доп.ЗП}}$ применяется при

шестидневной рабочей неделе, при пятидневной оно равно 1,113, соответственно в этом случае $K_{и} = 1,62$.

Таблица 13 - Затраты на заработную плату

Исполнитель	Оклад, руб./мес.	Среднедневная ставка, руб./раб. день	Затраты времени, раб. дни	Коэффициент	Фонд з/платы, руб.
НР	33 664	1342,09	25	1,699	93 488,65
И	93 77, 78	455,64	73	1,62	53 888,05
Итого:					147 376,70

6.2.5 Расчет зарплат на социальный налог

Затраты на единый социальный налог (ЕСН), включающий в себя отчисления в пенсионный фонд, на социальное и медицинское страхование, составляют 30 % от полной заработной платы по проекту, т.е. $C_{соц} = C_{зп} \cdot 0,3$.
Итак, в нашем случае $C_{соц} = 147376,70 \cdot 0,3 = 44\,213,01$ руб.

6.2.6 Расчет затрат на электроэнергию

Данный вид расходов включает в себя затраты на электроэнергию, потраченную в ходе выполнения проекта на работу используемого оборудования, рассчитываемые по формуле:

$$C_{эл.об.} = P_{об} \cdot t_{об} \cdot Ц_{э} \quad (9)$$

где $P_{об}$ – мощность, потребляемая оборудованием, кВт;

$Ц_{э}$ – тариф на 1 кВт·час;

$t_{об}$ – время работы оборудования, час.

Для ТПУ $Ц_{э} = 6,59$ руб./кВт·час (с НДС).

Время работы оборудования вычисляется на основе итоговых данных таблицы 4 для инженера (ТРД) из расчета, что продолжительность рабочего дня равна 8 часов.

$$t_{об} = T_{рД} * K_t, \quad (10)$$

где $K_t \leq 1$ – коэффициент использования оборудования по времени, равный отношению времени его работы в процессе выполнения проекта к ТРД, определяется исполнителем самостоятельно. В ряде случаев возможно определение $t_{об}$ путем прямого учета, особенно при ограниченном использовании соответствующего оборудования.

Мощность, потребляемая оборудованием, определяется по формуле:

$$P_{об} = P_{ном.} * K_C, \quad (11)$$

где $P_{ном.}$ – номинальная мощность оборудования, кВт;

$K_C \leq 1$ – коэффициент загрузки, зависящий от средней степени использования номинальной мощности. Для технологического оборудования малой мощности $K_C = 1$.

Пример расчета затраты на электроэнергию для технологических целей приведен в таблице 14.

Таблица 14 - Затраты на электроэнергию технологическую

Наименование оборудования	Время работы оборудования $t_{об}$, час	Потребляемая мощность $P_{об}$, кВт	Затраты $Э_{об}$, руб.
Персональный компьютер	584*1	0,3	1154,56
Итого:			1151,56

6.2.7 Расчет амортизационных расходов

В статье «Амортизационные отчисления» рассчитывается амортизация используемого оборудования за время выполнения проекта.

Используется формула

$$C_{AM} = \frac{H_A * C_{OB} * t_{pf} * n}{F_D}, \quad (12)$$

где H_A – годовая норма амортизации единицы оборудования;

C_{OB} – балансовая стоимость единицы оборудования с учетом ТЗР. При невозможности получить соответствующие данные из бухгалтерии она может быть заменена действующей ценой, содержащейся в ценниках, прейскурантах и т.п.;

F_D – действительный годовой фонд времени работы соответствующего оборудования, берется из специальных справочников или фактического режима его использования в текущем календарном году. При этом второй вариант позволяет получить более объективную оценку C_{AM} . Например, для ПК в 2020 г. (298 рабочих дней при шестидневной рабочей неделе) можно принять $F_D = 247 * 8 = 1976$ часа;

t_{pf} – фактическое время работы оборудования в ходе выполнения проекта, учитывается исполнителем проекта;

n – число задействованных однотипных единиц оборудования.

При использовании нескольких типов оборудования расчет по формуле делается соответствующее число раз, затем результаты суммируются.

Для определения H_A следует обратиться к приложению 1, содержащему фрагменты из постановления правительства РФ «О классификации основных средств, включенных в амортизационные группы». Оно позволяет получить рамочные значения сроков амортизации (полезного использования) оборудования $\equiv C_A$. Например, для ПК это $2 \div 3$ года. Необходимо задать конкретное значение C_A из указанного интервала, например, 2,5 года. Далее определяется H_A как величина обратная C_A , в данном случае это $1 : 2,5 = 0,4$.

Стоимость ПК 40000 руб, время использования 504 часа, тогда для него $C_{AM}(ПК) = (0,4 \cdot 40000 \cdot 584 \cdot 1) / 2408 = 3880,39$ руб. Итого начислено амортизации 3880,39 руб.

6.2.8 Расчет расходов, учитываемых непосредственно на основе платежных документов

Сюда относятся:

- командировочные расходы, в т.ч. расходы по оплате суточных, транспортные расходы, компенсация стоимости жилья;
- арендная плата за пользование имуществом;
- оплата услуг связи;
- услуги сторонних организаций.

Норма оплаты суточных – **100 руб./день**.

Оплата проживания в общежитии 50 руб./день*110 дней = 5500 руб. $C_{нр} = 50 \cdot 110 = 5500$ руб.

6.2.9 Расчет прочих расходов

В статье «Прочие расходы» отражены расходы на выполнение проекта, которые не учтены в предыдущих статьях, их следует принять равными 10% от суммы всех предыдущих расходов, т.е.

$$C_{\text{проч.}} = (C_{\text{мат}} + C_{\text{зп}} + C_{\text{соц}} + C_{\text{эл.об.}} + C_{\text{ам}} + C_{\text{нп}}) \cdot 0,1$$

Для нашего примера это

$C_{\text{проч.}} = (1575 + 147376,70 + 44\,213,01 + 1154,56 + 3880,39 + 5500) \cdot 0,1 = 20323,92$ руб.

6.2.10 Расчет общей себестоимости разработки

Проведя расчет по всем статьям сметы затрат на разработку, можно определить общую себестоимость проекта «Макет демонстрационной модели принципов КТ».

Таблица 15 – Смета затрат на разработку проекта

Статья затрат	Условное обозначение	Сумма, руб.
Материалы и покупные изделия	$C_{\text{мат}}$	1575
Основная заработная плата	$C_{\text{зп}}$	147376,70
Отчисления в социальные фонды	$C_{\text{соц}}$	44213,01
Расходы на электроэнергию	$C_{\text{эл.}}$	1154,56
Амортизационные отчисления	$C_{\text{ам}}$	3880,39
Непосредственно учитываемые расходы	$C_{\text{нр}}$	5500
Прочие расходы	$C_{\text{проч}}$	20306,65
Итого:		222 739,29

Таким образом, затраты на разработку составили $C = 222\,739,29$ руб.

6.2.11 Расчет прибыли

Прибыль от реализации проекта в зависимости от конкретной ситуации (масштаб и характер получаемого результата, степень его определенности и коммерциализации, специфика целевого сегмента рынка и т.д.) может определяться различными способами. Если исполнитель работы не располагает данными для применения «сложных» методов, то прибыль следует принять в размере $5 \div 20\%$ от полной себестоимости проекта. В нашем примере она составляет 44547,85 руб. (20 %) от расходов на разработку проекта.

6.2.12 Расчет НДС

НДС составляет 20% от суммы затрат на разработку и прибыли. В нашем случае это $(222\,739,29 + 44547,85) \cdot 0,2 = 53457,429$ руб.

6.2.13 Расчет НИР

Цена равна сумме полной себестоимости, прибыли и НДС, в нашем случае

$$Ц_{\text{НИР(КР)}} = 222\,739,29 + 44547,85 + 53457,429 = 320\,744,56 \text{ руб.}$$

6.2.14 Оценка экономической эффективности проекта

На данном этапе разработки проекта крайне сложно рассчитать экономическую эффективность, однако есть факторы, которые необходимо упомянуть.

Наиболее распространенный способ оценки эффективности (прежде всего экономической) является расчет **возврата инвестиций (ROI)**. Вычисляется он достаточно просто, являясь отношением прибыли к затратам. Как только значение ROI переходит единицу – решение возвращает вложенные в него средства и начинает приносить новые. Проблема заключается в том, что проект данной категории оценивать крайне трудно, поскольку данные проекты разнятся настолько, что усреднить затраты на него не удастся.

В случае автоматизации экономический эффект проявляется в **экономии затрат на ручное тестирование**. Кроме того, эффект в данном случае может быть и не явным – например, результаты нахождения дефектов в процессе ad-hoc тестирования инженерами, время которых высвободилось за счет автоматизации. Такой эффект достаточно сложно рассчитать.

Однако не всегда экономия является целью внедрения автоматизации. Один из примеров – **скорость выполнения тестирования** (как по скорости выполнения одного теста, так и по частоте проведения тестирования). По ряду причин скорость тестирования может быть критической для бизнеса – если вложения в автоматизацию окупаются полученной прибылью.

Другой пример – **исключение “человеческого фактора”** из процесса тестирования систем. Это важно, когда точность и корректность выполнения

операций является критической для бизнеса. Цена такой ошибки может быть значительно выше стоимости разработки и поддержки автотеста.

6.2.15 Определение срока окупаемости инвестиций

Оценка таких проектов индивидуальна и предвидеть окупаемость продукта крайне сложно, однако факторы, которые упоминались выше, способствуют развитию и окупаемости проекта.

6.3 Вывод по разделу

На основе изученного материала, а также проведенного тщательного исследования можно сказать, что проект имеет перспективную модель развития.

7 Заключение

В результате проведения исследовательской работы был пройден весь жизненный цикл автоматизированного тестирования. Были разработаны план тестирования, тестовые сценарии, тестовые скрипты. А поскольку, крупнейшие компании типа Yandex постепенно отказываются от тестирования по принципу record and playback, то разработка тестовых скриптов вручную представляет наибольшую актуальность. Сборка всего проекта с внедрением Git позволяет быть конкурентоспособной на рынке.

8 Список использованной литературы

1. Основные положения тестирования // Хабр. URL: <https://habr.com/ru/post/110307/> (дата обращения 10.04.2020);
2. Уровни тестирования ПО // Studfiles. URL: <https://studfile.net/preview/4532339/> (дата обращения 11.04.2020);
3. <http://www.protesting.ru/testing/levels/integration.html>
4. Тестирование // ПроТестинг. URL: <http://www.protesting.ru/testing/levels/integration.html> (дата обращения 14.04.2020);
5. Топ 10 инструментов для автоматизации // Хабр. URL: <https://habr.com/ru/post/342234/> (дата обращения 18.04.2020);
6. Трудовой кодекс Российской Федерации: // Гарант. URL: <http://base.garant.ru/12125268> (дата обращения: 22.05.2020);
7. ГОСТ 12.2.032-78. ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования. – М.: Информационно-издательский центр Минздрава России, 1979. – 10 с.
8. ГОСТ 12.0.003-2015. Система стандартов безопасности труда (ССБТ). Опасные и вредные производственные факторы. Классификация. – М.: Стандартинформ, 2016.
9. СанПиН 2.2.4.548-96 Гигиенические требования к микроклимату производственных помещений. – М.: Информационно-издательский центр Минздрава России, 1997. – 11 с.
10. СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95*. – М.: Стандартинформ, 2017.
11. Об утверждении Правил по охране труда при эксплуатации электроустановок (с изменениями на 15 ноября 2018 года). – М.: АО "Кодекс", 2018;
12. ГОСТ Р ИСО 1410-2010. Экологический менеджмент. Оценка жизненного Цикла. Принципы и структура.
13. ГОСТ Р 22.0.01-2016. Безопасность в ЧС. Основные положения.

14.Р 2.2.2006-05 Гигиена труда. Руководство по гигиенической оценке факторов рабочей среды и трудового процесса. Критерии и классификация условий труда: // ТехЭксперт. URL: <http://docs.cntd.ru/document/1200040973> (дата обращения: 22.05.2020);

Приложение А

План тестирования

Введение

1. Назначение

План тестирования создан для того, чтобы:

- Собрать информацию об объекте тестирования;
- Определить стратегию, для реализации процесса тестирования;
- Определить требования и необходимые ресурсы;
- Определить список тестов для покрытия web-приложения.

2. Исходные данные

Тестирование будет проводиться на сайте `selenium1py.pythonanywhere.com`. Однако созданные тестовые скрипты очень гибки в использовании, так как создавались вручную с направленностью на их внедрение в другие web-приложения.

3. Область применения

Объектом тестирования для автоматизации тестирования является интерфейс пользователя web-приложения.

4. Тестовые требования

Ниже перечислены требования к пользовательским сценариям использования:

- ошибки в функциональности интерфейса;
- ошибки, не подходящие в разряд требований;
- необработанные исключения при взаимодействии с интерфейсом web-приложения.

5. Стратегия тестирования

Для тестирования созданы списки тестовых сценариев и тестовые скрипты. Тестирование производится методом “черного ящика”. Завершение тестирования сопровождается успешным выполнением всех тестовых сценариев.

6. Ресурсы

Так как проект разрабатывался в рамках ВКР без участия той или иной компании, то роль специалиста по автоматизации и программиста принял на себя один человек. Поэтому в ответственностях будут перечислены пункты для одного человека:

- Создание плана тестирования;
- Оценка эффективности автоматизации тестирования;
- Создание тестовых сценариев;
- Создание тестовых скриптов;
- Запуск тестовых скриптов;
- Отчет о результатах;
- Исправление ошибок.

Системные ресурсы (пк): ОС Win 8.1 и выше, Intel(R) Core(TM) i5-4200 CPU @ 1.60GHz, 4Гб ОЗУ, 64 разрядная ОС.

7. Этапы проекта

Основные задания	Задачи	Дата начала	Дата окончания
Планирование тестирования	Разработка плана тестирования	26.02.2020	16.03.2020
Проектирование тестирования	Разработка тестов	16.03.2020	10.05.2020

Выполнение тестирования	Тестирование исходя из составленного плана по требованиям	10.05.2020	18.05.2020
Оценка тестирования	Подведение итогов выполнения	18.05.2020	30.05.2020

Приложение Б

Действие: Проверка логирования	Ожидаемый результат	Фактический результат: – Пройден – Провален
Предусловие:		
В консоли перейти в папку проекта	Открыта консоль с адресом указанного каталога	пройден
Активировать окружение	Окружение отображено в строке консоли	пройден
Запустить тест	Происходит имитация действий, прописанные скриптами. По завершению получен отчет.	пройден
Шаги:		Проверка входа
1.Открыть браузер	Браузер успешно открыт	пройден
2. Перейти по ссылке	Отобразился сайт, указанный по ссылке	пройден
3.Нажать кнопку “Войти или зарегистрироваться”	Открылось окно регистрации с полями ввода	пройден
4. Заполнить поле E-mail	Введенные буквы отображаются, поле подсвечивается синим	пройден
5. Заполнить поле Password	Введенные буквы отображаются, поле подсвечивается синим	пройден
6. Нажать кнопку “Войти”	Переход на основную страницу. Вверху появился авторизованный пользователь	пройден
7. Нажать кнопку “Выйти”	Переход на основную страницу. Вверху исчез	пройден

	авторизованный пользователь	
8. Закреть браузер	Браузер успешно закрыт	пройден
Шаги:		Проверка регистрации
1.Повторить шаги 1-5	См. тест-кейсы выше	пройден
2. Заполнить поле подтверждения Password	Введенные буквы отображаются, поле подсвечивается синим	пройден
3. Нажать кнопку “Зарегистрироваться”	Переход на основную страницу. Вверху появился авторизованный пользователь	пройден
4. Нажать кнопку “Выйти”	Переход на основную страницу. Вверху исчез авторизованный пользователь	пройден
5. Закреть браузер	Браузер успешно закрыт	пройден

Действие: Проверка корзины товаров	Ожидаемый результат	Фактический результат: – Пройден – Провален
Предусловие		
В консоли перейти в папку проекта	Открыта консоль с адресом указанного каталога	пройден
Активировать окружение	Окружение отображено в строке консоли	пройден
Запустить тест	Происходит имитация действий, прописанные скриптами. По завершению получен отчет.	пройден
Шаги:		

1.Открыть браузер	Браузер успешно открыт	пройден
2. Перейти по ссылке	Отобразился сайт, указанный по ссылке	пройден
3.Нажать кнопку “Все товары”	Переход на страницу товаров	пройден
4. Нажать кнопку “Добавить в корзину”	Переход на страницу корзины товаров	пройден
5. Нажать кнопку “Посмотреть корзину”	Переход на страницу корзины с выбранными товарами	пройден
6. Нажать кнопку “Перейти к оформлению”	Переход на страницу оформления товаров корзины	пройден
7. Закрыть браузер	Браузер успешно закрыт	пройден