

**Министерство образования и науки Российской Федерации**  
федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Институт кибернетики  
Направление подготовки 09.04.02 Информационные системы и технологии  
Кафедра Информационные системы и технологии

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

Тема работы
<b>Разработка технологии автоматизированного тестирования программного обеспечения семейства информационных систем</b>

УДК \_\_\_\_\_

Студент

Группа	ФИО	Подпись	Дата
8ИМ5А	Борисова Анастасия Владимировна		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент кафедры ВТ	Кудинов Антон Викторович	К.т.н., доцент		

**КОНСУЛЬТАНТЫ:**

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Попова Светлана Николаевна			

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент кафедры экологии и безопасности жизнедеятельности	Акулов Петр Анатольевич			

**ДОПУСТИТЬ К ЗАЩИТЕ:**

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
Информационные системы и технологии	Мальчуков Андрей Николаевич	К.т.н.		

Томск – 2017 г.

## Планируемые результаты обучения

Код результат ов	Результат обучения (выпускник должен быть готов)
<b>Общепрофессиональные компетенции</b>	
P1	Воспринимать и самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте.
P2	Владеть и применять методы и средства получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе в глобальных компьютерных сетях.
P3	Демонстрировать культуру мышления, способность выстраивать логику рассуждений и высказываний, основанных на интерпретации данных, интегрированных из разных областей науки и техники, выносить суждения на основании неполных данных, анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями.
P4	Анализировать и оценивать уровни своих компетенций в сочетании со способностью и готовностью к саморегулированию дальнейшего образования и профессиональной мобильности. Владеть, по крайней мере, одним из иностранных языков на уровне социального и профессионального общения, применять специальную лексику и профессиональную терминологию языка.
<b>Профессиональные компетенции</b>	
P5	Разрабатывать стратегии и цели проектирования, критерии эффективности и ограничения применимости, новые методы, средства и технологии проектирования геоинформационных систем (ГИС) или промышленного программного обеспечения.
P6	Планировать и проводить теоретические и экспериментальные исследования в области создания интеллектуальных ГИС и ГИС технологии или промышленного программного обеспечения с использованием методов системной инженерии.
P7	Осуществлять авторское сопровождение процессов проектирования, внедрения и сопровождения ГИС и ГИС технологий или промышленного программного обеспечения с использованием методов и средств системной инженерии, осуществлять подготовку и обучение персонала.
P8	Формировать новые конкурентоспособные идеи в области теории и практики ГИС и ГИС технологий или системной инженерии программного обеспечения. Разрабатывать методы решения нестандартных задач и новые методы решения традиционных задач. Организовывать взаимодействие коллективов, принимать управленческие решения, находить компромисс между различными требованиями как при долгосрочном, так и при краткосрочном планировании.
<b>Общекультурные компетенции</b>	
P9	Использовать на практике умения и навыки в организации исследовательских, проектных работ и профессиональной эксплуатации современного оборудования и приборов, в управлении коллективом.
P10	Свободно пользоваться русским и иностранным языками как средством делового общения.
P11	Совершенствовать и развивать свой интеллектуальный и общекультурный уровень. Проявлять инициативу, в том числе в ситуациях риска, брать на себя всю полноту ответственности.
P12	Демонстрировать способность к самостоятельному обучению новым методам исследования, к изменению научного и научно-производственного профиля своей профессиональной деятельности, способность самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности, способность к педагогической деятельности.

**Министерство образования и науки Российской Федерации**  
федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Институт кибернетики  
Направление подготовки 09.04.02 Информационные системы и технологии  
Кафедра Информационные системы и технологии

УТВЕРЖДАЮ:  
Зав. кафедрой

\_\_\_\_\_  
(Подпись)      (Дата)      (Ф.И.О.)

**ЗАДАНИЕ**  
**на выполнение выпускной квалификационной работы**

В форме:

магистерской диссертации
--------------------------

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

Студенту:

Группа	ФИО
8ИМ5А	Борисова Анастасия Владимировна

Тема работы:

Разработка технологии автоматизированного тестирования программного обеспечения семейства информационных систем	
Утверждена приказом директора (дата, номер)	

Срок сдачи студентом выполненной работы:

--	--

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ:**

<b>Исходные данные к работе</b>	Задача внедрения технологии автоматизированного тестирования и применение ее на реальных продуктах в компании ООО «ТомскАСУпроект». А именно, в подготовке тест-кейсов для тестирования и тестирование 50% функционала desktop-модуля программной системы «CAD-GIS» «Автоматизированная система для создания ведомости подсчета объемов работ по автомобильным дорогам и генеральным планам» и 100% функционала web-приложения «Корпоративная геоинформационная система» ООО «РН – Пурнефтегаз»
---------------------------------	---

<b>Перечень подлежащих исследованию, проектированию и разработке вопросов</b>	Необходимо изучить методы и средства для web- и desktop-приложений, с помощью которых будет проводиться тестирование. Необходимо описать и выбрать наиболее подходящие методы и средства для написания тестовых наборов.
<b>Перечень графического материала</b>	Таблица методов тестирования. Таблица сравнительных характеристик средств для тестирования web-приложения. Таблица сравнительных характеристики средств для тестирования desktop-приложения. Таблица примеров тест-кейсов. Чек-лист покрытия тестами функциональности тестируемых приложений.
<b>Консультанты по разделам выпускной квалификационной работы</b>	
<b>Раздел</b>	<b>Консультант</b>
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Попова Светлана Николаевна
Социальная ответственность	Акулов Петр Анатольевич
Раздел на иностранном языке	Мирошниченко Евгений Александрович
<b>Названия разделов, которые должны быть написаны на русском и иностранном языках:</b>	
Описание предметной области (Description of the subject field)	

<b>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</b>	
---	--

**Задание выдал руководитель:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент кафедры ВТ	Кудинов Антон Викторович	К.т.н., доцент		

**Задание приняли к исполнению студент:**

Группа	ФИО	Подпись	Дата
8ИМ5А	Борисова Анастасия Владимировна		

**Министерство образования и науки Российской Федерации**  
федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Институт кибернетики  
Направление подготовки 09.04.02 Информационные системы и технологии  
Уровень образования – магистр  
Кафедра Информационные системы и технологии  
Период выполнения – осенний / весенний семестр 2016/2017 учебного года

Форма представления работы:

Магистерская диссертация

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН  
выполнения выпускной квалификационной работы**

Срок сдачи студентом выполненной работы:

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
02.02.2017	Глава «Описание предметной области»	20
25.03.2017	Глава «Аналитика и тестирование»	30
24.05.2017	Глава «Результаты тестирования»	30
09.06.2017	Глава «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»	10
09.06.2017	Глава «Социальная ответственность»	10

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент кафедры ВТ	Кудинов Антон Викторович	К.т.н., доцент		

**СОГЛАСОВАНО:**

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
Информационные системы и технологии	Мальчуков Андрей Николаевич	К.т.н.		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И  
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

<b>Группа</b>	<b>ФИО</b>
8ИМ5А	Борисова Анастасия Владимировна

<b>Институт</b>	<b>кибернетики</b>	<b>Кафедра</b>	<b>ИСТ</b>
<b>Уровень образования</b>	магистр	<b>Направление</b>	Информационные системы и технологии

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

1. <i>Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих</i>	<i>Работа с информацией, представленной в научных публикациях и статьях, аналитических материалах, статистических бюллетенях и изданиях, нормативно-правовых документах.</i>
2. <i>Нормы и нормативы расходования ресурсов</i>	
3. <i>Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>	

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

1. <i>Оценка коммерческого и инновационного потенциала НИИ</i>	<i>Оценка потенциальных потребителей исследования, SWOT-анализ, создание графика проведения и бюджета НИ, оценка ресурсной, финансовой и экономической эффективности НИ</i>
2. <i>Разработка устава научно-технического проекта</i>	
3. <i>Планирование процесса управления НИИ: структура и график проведения, бюджет, риски и организация закупок</i>	<i>Планирование этапов работ, определение трудоёмкости и построение календарного графика, формирование бюджета</i>
4. <i>Определение ресурсной, финансовой, экономической эффективности</i>	<i>Оценка сравнительной эффективности исследования</i>

**Перечень графического материала** (с точным указанием обязательных чертежей):

<ol style="list-style-type: none"> <li>1. <i>Сегментирование рынка</i></li> <li>2. <i>Диаграмма FAST</i></li> <li>3. <i>Матрица SWOT</i></li> <li>4. <i>График проведения и бюджет НИИ</i></li> <li>5. <i>Оценка ресурсной, финансовой и экономической эффективности НИИ</i></li> <li>6. <i>Потенциальные риски</i></li> </ol>	
--	--

<b>Дата выдачи задания для раздела по линейному графику</b>	
---	--

**Задание выдал консультант:**

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
Старший преподаватель	Попова Светлана Николаевна			

**Задание принял к исполнению студент:**

<b>Группа</b>	<b>ФИО</b>	<b>Подпись</b>	<b>Дата</b>
8ИМ5А	Борисова Анастасия Владимировна		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»**

Студенту:

<b>Группа</b>	<b>ФИО</b>
8ИМ5А	Борисова Анастасия Владимировна

<b>Институт</b>	<b>ИК</b>	<b>Кафедра</b>	<b>Информационных систем и технологий</b>
<b>Уровень образования</b>	Магистр	<b>Направление/специальность</b>	09.04.02 Информационные системы и технологии

**Исходные данные к разделу «Социальная ответственность»:**

<p><i>1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения</i></p>	<p><i>Объектом исследования являются средства тестирования Selenium IDE и TestComplete, языки программирования C# Script, язык команд Selenium; язык гипертекстовой разметки HTML; язык запросов xpath, CSS.</i></p> <p><i>Целью магистерской диссертации является изучение и разработка технологии автоматизированного тестирования программного обеспечения семейства информационных систем, что позволило бы обеспечить повышение качества разрабатываемого программного продукта и эффективности тестирования.</i></p>
--	--

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

<p><b>1 Производственная безопасность:</b></p> <p>1.1 Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения;</p> <p>1.2 Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности.</p>	<p><i>Производственная безопасность на стадии разработки тестовых наборов.</i></p> <p><i>1.1 В качестве вредных факторов выделены:</i></p> <ul style="list-style-type: none"> <li>- отклонение показателей микроклимата;</li> <li>- недостаточная освещенность рабочей зоны.</li> </ul> <p><i>1.2 В качестве опасных факторов выделены:</i></p> <ul style="list-style-type: none"> <li>- опасность возникновения пожара;</li> <li>- опасность поражения электрическим током;</li> </ul>
<p><b>2 Экологическая безопасность:</b></p> <p>2.1 Анализ воздействия объекта на окружающую среду;</p> <p>2.2 Разработать решения по обеспечению экологической безопасности со ссылками на НТД по охране окружающей среды.</p>	<p><i>2.1 Влияние объекта исследования на окружающую среду:</i></p> <ul style="list-style-type: none"> <li>- утилизация бумажных отходов и поломанной техники;</li> </ul> <p><i>2.3 Мероприятия по защите окружающей среды:</i></p> <ul style="list-style-type: none"> <li>- отказ использования бумаги в пользу ведения записей в электронном формате.</li> </ul>
<p><b>3 Безопасность в чрезвычайных ситуациях:</b></p> <p>3.1 Перечень возможных ЧС при разработке и эксплуатации проектируемого решения;</p> <p>3.2 Выбор наиболее типичной ЧС;</p> <p>3.3 Разработка действий в результате возникшей ЧС и мер по ликвидации её последствий.</p>	<p><i>3.1 Возможные чрезвычайные ситуации:</i></p> <ul style="list-style-type: none"> <li>- пожар;</li> <li>- кибертерроризм;</li> </ul> <p><i>3.2 Типичная чрезвычайная ситуация:</i></p> <ul style="list-style-type: none"> <li>- пожар;</li> <li>- кибертерроризм.</li> </ul> <p><i>3.3 Мероприятия по предотвращению наиболее типичной ЧС – пожара, согласно нормативным документам: НПБ 105-03 и ППБ 01-03.</i></p>
<p><b>4 Правовые и организационные</b></p>	<p><i>4.1 Описание правовых норм для работ,</i></p>

<p><b>вопросы обеспечения безопасности:</b></p> <p>4.1 Специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства;</p> <p>4.2 Организационные мероприятия при компоновке рабочей зоны.</p>	<p>связанных с работой за ПЭВМ согласно следующим документам:</p> <ul style="list-style-type: none"> <li>– Трудовой кодекс Российской Федерации" от 30.12.2001 N 197-ФЗ (ред. от 30.12.2015).</li> </ul> <p>4.2 Влияние разработанных тесовых наборов на ПО:</p> <ul style="list-style-type: none"> <li>– повышение качества ПО;</li> <li>– сокращение времени на тестирование в последующих версиях;</li> <li>– сокращение затрат;</li> <li>– сокращение человеческих ресурсов т.к. выполнение по большей части происходит без участия человека.</li> </ul>
---	--

<b>Дата выдачи задания для раздела по линейному графику</b>	
---	--

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент кафедры экологии и безопасности жизнедеятельности	Акулов Петр Анатольевич			

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8ИМ5А	Борисова Анастасия Владимировна		

## РЕФЕРАТ

Выпускная квалификационная работа содержит 118 с., 11 рис., 19 табл., 27 источников, 3 приложения.

Ключевые слова: автоматизированное тестирование, тестовый случай, качество программного обеспечения, методы тестирования, верификация.

Объектами исследования являются средства тестирования web- и desktop-приложений, тестируемые модуль программной системы «CAD-GIS» «Автоматизированная система для создания ведомости подсчета объемов работ по автомобильным дорогам и генеральным планам», «Корпоративная геоинформационная система» ООО «РН – Пурнефтегаз» (КГИС), и методы тестирования.

Цель работы – внедрение технологии автоматизированного тестирования и применение ее на реальных продуктах в компании ООО «ТомскАСУпроект».

В процессе исследования проводились сбор и анализ средств и методов для тестирования. Были изучены программные средства TestComplete, Selenium IDE, и трекер-система Redmine. Были получены навыки работы с языки C# Script, язык команд Selenium, язык гипертекстовой разметки HTML, язык запросов xpath, CSS. Были подготовлены тест-кейсы для тестирования и тестирование web- и desktop-приложений с помощью выбранных средств.

В результате на первых итерациях тестирования было найдено 23 ошибки в тестируемом desktop-приложении и 66 ошибок в web-приложении «КГИС» ООО «РН – Пурнефтегаз и проанализирована эффективность внедрения автоматизированного тестирования.

Область применения: результаты применимы на предприятии в ООО «ТомскАСУпроект».

Экономическая эффективность/значимость работы: данная работа сократит время и средства при регрессионном тестировании в случае улучшения и выхода новых версий тестируемых приложений.

## ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ

**БГД** – База географических данных.

**ГИС** – Географическая информационная система.

**ИТ** – информационные технологии.

**Качество программного обеспечения** – весь объем признаков и характеристик программной продукции, который относится к ее способности удовлетворять установленным или предполагаемым потребностям.

**Критерии оценки качества программного обеспечения** – набор определенных и задокументированных правил и условий, которые используются для решения о приемлемости общего качества конкретной программной продукции. Качество представляется набором установленных уровней, связанных с программной продукцией.

**Корпоративная геоинформационная система** – приложения масштаба предприятия, позволяющие рассматривать ГИС, как инструмент поддержки принятия решений. Основной признак системы – интеграция пространственных данных территории деятельности компании со всей совокупностью информационных систем предприятия.

**Тест** – набор операций, выполняемых с тестируемой программой.

**Тестовый случай (Test Case)** – это артефакт, описывающий совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или её части.

**Desktop-приложение** – клиентское программное обеспечение, реализующее Windows Forms интерфейс. Приложение устанавливается на рабочую станцию пользователя и запускается локально, или запускается удаленно.

**Web-приложение** – клиент-серверное приложение, в котором клиентом выступает браузер, а сервером — web-сервер.

**Автотесты** – автоматические тесты, созданные с помощью средств для автоматизированного тестирования. Оно использует программные средства для

выполнения тестов и проверки результатов выполнения, что помогает сократить время тестирования и упростить его процесс.

**Программное обеспечение (ПО)** – программы, процедуры, правила и любая соответствующая документация, относящиеся к работе вычислительной системы.

**Метрика** – это количественный масштаб и метод, который может использоваться для измерения.

**Верификация (verification)** – это процесс оценки системы или её компонентов с целью определения удовлетворяют ли результаты текущего этапа разработки условиям, сформированным в начале этого этапа. Т.е. выполняются ли цели, сроки, задачи по разработке проекта, определенные в начале текущей фазы.

**Валидация (validation)** – это определение соответствия разрабатываемого ПО ожиданиям и потребностям пользователя, требованиям к системе.

**Оценка** – действие по применению конкретного теста к конкретному программному модулю.

**САПР** – система автоматизированного проектирования.

**СГИО** – система геоинформационного обеспечения.

**AutoCAD Civil 3D** – система автоматизированного проектирования.

**Реестр работ** – содержит все добавленные в БГД работы, используемые в В1.

**Реестр элементов дороги** – содержит все добавленные в БГД элементы дороги (используются в В2-В5).

**В1** – Ведомость объемов работ.

**В2** – Покилометровая ведомость объемов земляных масс.

**В3** – Ведомость уширений проезжей части и земляного полотна.

**В4** – Ведомость устройства дорожной одежды.

**В5** – Ведомость укрепления откосов земляного полотна.

## СОДЕРЖАНИЕ

Реферат .....	9
Определения, обозначения, сокращения .....	10
Введение.....	13
1. Описание предметной области.....	15
1.1. Классификация методов тестирования .....	15
1.2. Деятельность компании ООО «ТомскАСУпроект» .....	29
1.3. Тестируемые приложения .....	30
2. Аналитика и тестирование.....	35
2.1. Выбор метода тестирования.....	35
2.2. Аналитический обзор программных средств тестирования web- и desktop-приложений.....	38
2.3. Тестирование.....	48
3. Результаты тестирования .....	55
3.1. Результаты автоматизации .....	55
3.2. Эффективность автоматизированного тестирования .....	61
4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	66
5. Социальная ответственность .....	82
Заключение .....	97
Список используемых источников.....	99
ПРИЛОЖЕНИЕ А .....	102
ПРИЛОЖЕНИЕ Б.....	115
ПРИЛОЖЕНИЕ В .....	117

## ВВЕДЕНИЕ

Число основанных на web- и desktop-технологиях программных продуктов с каждым днем увеличивается. При этом растет уровень сложности приложений и для их разработки требуются все более длительные сроки и широкий спектр операций по тестированию, в том числе при их модернизации и переходе от одной версии к другой. Качество разработки программных продуктов во многом зависит от эффективности информационных систем и любая ошибка может стать причиной значительных финансовых потерь для организации. Поэтому на сегодняшний день тестирование стало обязательной частью процесса производства ПО. Следствием такой деятельности является повышение качества продукта по всем его характеристикам.

Цель работы состоит во внедрении технологии автоматизированного тестирования и применение ее на реальных продуктах в компании ООО «ТомскАСУпроект». А именно, в проведении сравнительного анализа существующих средств для автоматизации тестирования web- и desktop-приложений, в выборе и изучении подходящих продуктов и методов для тестирования. А также в подготовке тест-кейсов для тестирования и тестирование 50% функционала desktop-модуля программной системы «CAD-GIS» «Автоматизированная система для создания ведомости подсчета объемов работ по автомобильным дорогам и генеральным планам» и 100% функционала web-приложения «Корпоративная геоинформационная система» ООО «РН – Пурнефтегаз».

Объектами исследования являются выбранные средства тестирования в результате анализа конкурирующих инструментов, используемых для разработки тестовых сценариев для web- и desktop-приложений – Selenium IDE, и TestComplete, тестируемые приложения и методы тестирования.

Предмет исследования – подготовка тест-кейсов для тестирования и тестирование «Корпоративной геоинформационной системы» ООО «РН – Пурнефтегаз» и модуля программной системы «CAD-GIS»

«Автоматизированная система для создания ведомости подсчета объемов работ по автомобильным дорогам и генеральным планам».

Практическая значимость результатов заключается в том, что в данной работе классифицированы и описаны все современные способы тестирования. Такая работа позволяет систематизировать и расширить знания и значительно ускорить процессы планирования тестирования и разработки тест-кейсов, а также оптимизировать трудозатраты. Разработанные тестовые наборы позволят сократить компании время, ресурсы и количество тестировщиков, а так же достигнуть высокого качества рассмотренных продуктов в последующих их версиях.

В результате были созданы тестовые наборы, покрывающие 50% функций desktop-приложения и содержащие 63 теста и 100% функций web-приложения, содержащие 248 тестов. На первых итерациях тестирования было найдено 23 ошибки в тестируемом desktop-приложении и 66 ошибок в web-приложении «КГИС» ООО «РН – Пурнефтегаз и проанализирована эффективность внедрения автоматизированного тестирования.

В ходе исследования была посчитана выгода компании от автоматизации тестирования и составила для модуля «Ведомости» -113 838 рублей, для модуля «КГИС» + 2 282 880 рублей. Был сделан вывод, что для маленькой компании и, особенно, для маленьких проектов автоматизированное тестирование дорогими программными средствами не выгодно. Но при большом объеме задач и наличии простого средства автоматизации, данный вид тестирования весьма эффективен.

# 1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Тестирование - это процесс обнаружения ошибок в ПО путем исполнения его выходного кода на тестовых данных, сбора рабочих характеристик в динамике выполнения в конкретной операционной среде, выявления различных ошибок, дефектов, отказов и изъянов, вызванных нерегулярными и аномальными ситуациями или аварийным прекращением работы ПО. Важное место в проведении верификации и тестирования занимают организационные аспекты - деятельность группы специалистов, осуществляющих планирование этих процессов, подготовку тестовых данных и наблюдение за тестированием [3].

## 1.1. Классификация методов тестирования

Тестирование можно классифицировать по очень большому количеству признаков. В данной работе собраны и описаны все способы классификации тестирования. Такая работа позволяет систематизировать и расширить знания и значительно ускорить процессы планирования тестирования и разработки тест-кейсов, а также оптимизировать трудозатраты. Но следует помнить, что общепринятого решения, которое в равной степени подходило бы для любой методологии управления проектами, любого отдельного проекта тестирования и любой его стадии, просто не существует.

### **Классификация по запуску кода на исполнение**

- Статическое тестирование – без запуска кода на исполнение. В рамках этого подхода тестированию могут подвергаться:
  - Документы (требования, тест-кейсы, и т.д.).
  - Графические прототипы (эскизы пользовательского интерфейса).
  - Код приложения
  - Параметры (настройки) среды исполнения приложения.
  - Подготовленные тестовые данные.
- Динамическое тестирование – с запуском кода на исполнение. Запускаться на исполнение может как код всего приложения целиком), так и

код нескольких взаимосвязанных частей (интеграционное тестирование), отдельных частей (модульное или компонентное тестирование) и даже отдельные участки кода. Основная идея этого вида тестирования состоит в том, что проверяется реальное поведение (части) приложения [12].

### **Классификация по доступу к коду и архитектуре приложения**

- Метод белого ящика – у тестировщика есть доступ к внутренней структуре и коду приложения, а также есть достаточно знаний для понимания увиденного.

- Метод чёрного ящика – у тестировщика либо нет доступа к внутренней структуре и коду приложения, либо недостаточно знаний для их понимания, либо он сознательно не обращается к ним в процессе тестирования. При этом абсолютное большинство видов тестирования работают по методу чёрного ящика. В рамках тестирования по методу чёрного ящика основной информацией для создания тест-кейсов выступает документация (особенно – требования) и общий здравый смысл (для случаев, когда поведение приложения в некоторой ситуации не регламентировано явно; иногда это называют «тестированием на основе неявных требований»).

- Метод серого ящика – комбинация методов белого ящика и чёрного ящика, состоящая в том, что к части кода и архитектуры у тестировщика доступ есть, а к части – нет [2].

### **Классификация по степени автоматизации**

- Ручное тестирование – тест-кейсы выполняются человеком вручную без использования средств автоматизации.

- Автоматизированное тестирование – набор техник, подходов и инструментальных средств, позволяющий исключить человека из выполнения некоторых задач в процессе тестирования. Разработка тест-кейсов, подготовка данных, оценка результатов выполнения, написания отчётов об обнаруженных дефектах – всё это и многое другое по-прежнему делает человек.

## **Классификация по уровню детализации приложения (по уровню тестирования)**

- Модульное (компонентное) тестирование (unit testing) направлено на проверку отдельных небольших частей приложения, которые (как правило) можно исследовать изолированно от других подобных частей. При выполнении данного тестирования могут проверяться отдельные функции или методы классов, сами классы, взаимодействие классов, небольшие библиотеки, отдельные части приложения. Часто данный вид тестирования реализуется с использованием специальных технологий и инструментальных средств автоматизации тестирования, значительно упрощающих и ускоряющих разработку соответствующих тест-кейсов.

- Интеграционное тестирование направлено на проверку взаимодействия между несколькими частями приложения (каждая из которых, в свою очередь, проверена отдельно на стадии модульного тестирования).

- Системное тестирование направлено на проверку всего приложения как единого целого, собранного из частей, проверенных на двух предыдущих стадиях. Здесь не только выявляются дефекты «на стыках» компонентов, но и появляется возможность полноценно взаимодействовать с приложением с точки зрения конечного пользователя, применяя множество других перечисленных в данной главе видов тестирования [3].

## **Классификация по (убыванию) степени важности тестируемых функций (по уровню функционального тестирования)**

- Дымовое тестирование (smoke test) направлено на проверку самой главной, самой важной, самой ключевой функциональности, неработоспособность которой делает бессмысленной саму идею использования приложения (или иного объекта, подвергаемого дымовому тестированию). Дымовое тестирование проводится после выхода нового билда, чтобы определить общий уровень качества приложения и принять решение о (не)целесообразности выполнения тестирования критического пути и расширенного тестирования.

- Тестирование критического пути направлено на исследование функциональности, используемой типичными пользователями в типичной повседневной деятельности. Сама идея позаимствована из управления проектами и трансформирована в контексте тестирования в следующую: существует большинство пользователей, которые чаще всего используют некое подмножество функций приложения.

- Расширенное тестирование направлено на исследование всей заявленной в требованиях функциональности – даже той, которая низко проранжирована по степени важности. При этом здесь также учитывается, какая функциональность является более важной, а какая – менее важной. Но при наличии достаточного количества времени и иных ресурсов тест-кейсы этого уровня могут затронуть даже самые низкоприоритетные требования [12].

#### **Классификация по принципам работы с приложением**

- Позитивное тестирование направлено на исследование приложения в ситуации, когда все действия выполняются строго по инструкции, без каких бы то ни было ошибок, отклонений, ввода неверных данных и т.д.

- Негативное тестирование – направлено на исследование работы приложения в ситуациях, когда с ним выполняются (некорректные) операции и/или используются данные, потенциально приводящие к ошибкам (деление на ноль) [11].

#### **Классификация по природе приложения**

- Тестирование web-приложений сопряжено с интенсивной деятельностью в области тестирования совместимости (в особенности – кросс-браузерного тестирования), тестирования производительности, автоматизации тестирования с использованием широкого спектра инструментальных средств.

- Тестирование мобильных приложений
- Тестирование настольных приложений

## **Классификация по фокусировке на уровне архитектуры приложения**

- Тестирование уровня представления сконцентрировано на той части приложения, которая отвечает за взаимодействие с «внешним миром» (как пользователями, так и другими приложениями). Здесь исследуются вопросы удобства использования, скорости отклика интерфейса, совместимости с браузерами, корректности работы интерфейсов.

- Тестирование уровня бизнес–логики отвечает за проверку основного набора функций приложения и строится на базе ключевых требований к приложению, бизнес–правил и общей проверки функциональности.

- Тестирование уровня данных сконцентрировано на той части приложения, которая отвечает за хранение и некоторую обработку данных (чаще всего – в базе данных или ином хранилище). Здесь особый интерес представляет тестирование данных, проверка соблюдения бизнес–правил, тестирование производительности [14].

## **Классификация по привлечению конечных пользователей (операционное тестирование)**

- Альфа–тестирование выполняется внутри организации–разработчика с возможным частичным привлечением конечных пользователей. Может являться формой внутреннего приёмочного тестирования. Суть этого вида вкратце: продукт уже можно периодически показывать внешним пользователям, но он ещё достаточно «сырой», потому основное тестирование выполняется организацией–разработчиком.

- Бета–тестирование выполняется вне организации–разработчика с активным привлечением конечных пользователей/заказчиков. Может являться формой внешнего приёмочного тестирования. Суть этого вида вкратце: продукт уже можно открыто показывать внешним пользователям, он уже достаточно стабилен, но проблемы всё ещё могут быть, и для их выявления нужна обратная связь от реальных пользователей.

- Гамма–тестирование – финальная стадия тестирования перед выпуском продукта, направленная на исправление незначительных дефектов, обнаруженных в бета–тестировании. Как правило, также выполняется с максимальным привлечением конечных пользователей/заказчиков. Может являться формой внешнего приёмочного тестирования. Суть этого вида вкратце: продукт уже почти готов, и сейчас обратная связь от реальных пользователей используется для устранения последних недоработок [4].

### **Классификация по степени формализации**

- Тестирование на основе тест-кейсов – формализованный подход, в котором тестирование производится на основе заранее подготовленных тест-кейсов, наборов тест-кейсов и иной документации.

- Исследовательское тестирование – частично формализованный подход, в рамках которого тестировщик выполняет работу с приложением по выбранному сценарию, который, в свою очередь, дорабатывается в процессе выполнения с целью более полного исследования приложения. Ключевым фактором успеха при выполнении исследовательского тестирования является именно работа по сценарию, а не выполнение разрозненных бездумных операций. Существует даже специальный сценарный подход, называемый сессионным тестированием. В качестве альтернативы сценариям при выборе действий с приложением иногда могут использоваться чек–листы, и тогда этот вид тестирования называют тестированием на основе чек–листов.

- Свободное (интуитивное) тестирование – полностью неформализованный подход, в котором не предполагается использования ни тест-кейсов, ни чек–листов, ни сценариев – тестировщик полностью опирается на свой профессионализм и интуицию для спонтанного выполнения с приложением действий, которые, как он считает, могут обнаружить ошибку.

### **Классификация по целям и задачам**

- Позитивное тестирование (рассмотрено ранее).
- Негативное тестирование (рассмотрено ранее).

- Функциональное тестирование – направлено на проверку корректности работы функциональности приложения (корректность реализации функциональных требований).

- Нефункциональное тестирование – направлено на проверку нефункциональных особенностей приложения (корректность реализации нефункциональных требований), таких как удобство использования, совместимость, производительность, безопасность и т.д.

- Инсталляционное тестирование – направлено на выявление дефектов, влияющих на протекание стадии инсталляции (установки) приложения. В общем случае такое тестирование проверяет множество сценариев и аспектов работы инсталлятора в таких ситуациях, как:

- новая среда исполнения
- обновление существующей версии;
- изменение текущей версии на более старую;
- повторная установка приложения;
- повторный запуск инсталляции;
- удаление приложения;
- установка нового приложения из семейства приложений;
- автоматическая инсталляция без участия пользователя.

- Регрессионное тестирование – тестирование, направленное на проверку того факта, что в ранее работоспособной функциональности не появились ошибки, вызванные изменениями в приложении или среде его функционирования.

- Повторное тестирование – выполнение тест-кейсов, которые ранее обнаружили дефекты, с целью подтверждения устранения дефектов.

- Приёмочное тестирование – формализованное тестирование, направленное на проверку приложения с точки зрения конечного пользователя/заказчика и вынесения решения о том, принимает ли заказчик работу у исполнителя (проектной команды).

- Операционное тестирование – тестирование, проводимое в реальной или приближенной к реальной операционной среде, включающей операционную систему, системы управления базами данных, серверы приложений, web-серверы, аппаратное обеспечение и т.д.
- Тестирование удобства использования – направлено на исследование того, насколько конечному пользователю понятно, как работать с продуктом, а также на то, насколько ему нравится использовать продукт.
- Тестирование доступности – направлено на исследование пригодности продукта к использованию людьми с ограниченными возможностями (слабым зрением и т.д.)
- Тестирование интерфейса – направлено на проверку интерфейсов приложения или его компонентов. По определению ISTQB–гlossария этот вид тестирования относится к интеграционному тестированию.
- Тестирование безопасности – направлено на проверку способности приложения противостоять злонамеренным попыткам получения доступа к данным или функциям, права на доступ к которым у злоумышленника нет.
- Тестирование интернационализации – направлено на проверку готовности продукта к работе с использованием различных языков и с учётом различных национальных и культурных особенностей.
- Тестирование локализации – направлено на проверку корректности и качества адаптации продукта к использованию на том или ином языке с учётом национальных и культурных особенностей. Проверяет корректность перевода и адаптации продукта, а не готовность продукта к таким действиям.
- Тестирование совместимости – направлено на проверку способности приложения работать в указанном окружении. Здесь, например, может проверяться:
  - Совместимость с аппаратной платформой, операционной системой и сетевой инфраструктурой.
  - Совместимость с браузерами и их версиями
  - Совместимость с мобильными устройствами.

- И так далее.
- Тестирование данных и баз данных – два близких по смыслу вида тестирования, направленных на исследование таких характеристик данных, как полнота, непротиворечивость, целостность, структурированность и т.д. В контексте баз данных исследованию может подвергаться адекватность модели предметной области, способность модели обеспечивать целостность и консистентность данных, корректность работы триггеров, хранимых процедур и т.д.
- Тестирование использования ресурсов – совокупность видов тестирования, проверяющих эффективность использования приложением доступных ему ресурсов и зависимость результатов работы приложения от количества доступных ему ресурсов.
- Сравнительное тестирование – направлено на сравнительный анализ преимуществ и недостатков разрабатываемого продукта по отношению к его основным конкурентам.
- Демонстрационное тестирование – формальный процесс демонстрации заказчику продукта с целью подтверждения, что продукт соответствует всем заявленным требованиям. В отличие от приёмочного тестирования этот процесс более строгий и всеобъемлющий, но может проводиться и на промежуточных стадиях разработки продукта.
- Избыточное тестирование – тестирование приложения со всеми возможными комбинациями всех возможных входных данных во всех возможных условиях выполнения. Для сколь бы то ни было сложной системы нереализуемо, но может применяться для проверки отдельных крайне простых компонентов [16].
- Тестирование надёжности – тестирование способности приложения выполнять свои функции в заданных условиях на протяжении заданного времени или заданного количества операций.
- Тестирование восстанавливаемости – тестирование способности приложения восстанавливать свои функции и заданный уровень

производительности, а также восстанавливать данные в случае возникновения критической ситуации, приводящей к временной (частичной) утрате работоспособности приложения.

- Тестирование отказоустойчивости – тестирование, заключающееся в эмуляции или реальном создании критических ситуаций с целью проверки способности приложения задействовать соответствующие механизмы, предотвращающие нарушение работоспособности, производительности и повреждения данных.

- Тестирование производительности – исследование показателей скорости реакции приложения на внешние воздействия при различной по характеру и интенсивности нагрузке. В рамках тестирования производительности выделяют следующие подвиды:

- Нагрузочное тестирование – исследование способности приложения сохранять заданные показатели качества при нагрузке в допустимых пределах и некотором превышении этих пределов (определение «запаса прочности»).

- Тестирование масштабируемости – исследование способности приложения увеличивать показатели производительности в соответствии с увеличением количества доступных приложению ресурсов.

- Объёмное тестирование – исследование производительности приложения при обработке различных (как правило, больших) объёмов данных.

- Стрессовое тестирование – исследование поведения приложения при нештатных изменениях нагрузки, значительно превышающих расчётный уровень, или в ситуациях недоступности значительной части необходимых приложению ресурсов. Стрессовое тестирование может выполняться и вне контекста нагрузочного тестирования: тогда оно, как правило, называется «тестированием на разрушение» и представляет собой крайнюю форму негативного тестирования.

- Конкурентное тестирование – исследование поведения приложения в ситуации, когда ему приходится обрабатывать большое количество одновременно поступающих запросов, что вызывает конкуренцию между

запросами за ресурсы. Иногда под конкурентным тестированием понимают также исследование работы многопоточных приложений и корректность синхронизации действий, производимых в разных потоках.

### **Классификация по техникам и подходам**

- Позитивное тестирование (рассмотрено ранее).
- Негативное тестирование (рассмотрено ранее).
- На основе опыта тестировщика, сценариев, чек-листов:
  - Исследовательское тестирование (рассмотрено ранее).
  - Свободное (интуитивное) тестирование (рассмотрено ранее).
- По степени вмешательства в работу приложения:
  - Инвазивное тестирование – тестирование, выполнение которого может повлиять на функционирование приложения в силу работы инструментов тестирования или в силу вмешательства в сам код приложения. Некоторые источники рассматривают инвазивное тестирование как форму негативного или даже стрессового тестирования.
    - Неинвазивное тестирование – тестирование, выполнение которого незаметно для приложения и не влияет на процесс его обычной работы.
  - По техникам автоматизации:
    - Тестирование под управлением данными – способ разработки автоматизированных тест-кейсов, в котором входные данные и ожидаемые результаты выносятся за пределы тест-кейса и хранятся вне его – в файле, базе данных и т.д.
    - Тестирование под управлением ключевыми словами – способ разработки автоматизированных тест-кейсов, в котором за пределы тест-кейса выносятся не только набор входных данных и ожидаемых результатов, но и логика поведения тест-кейса, которая описывается ключевыми словами (командами).
    - Тестирование под управлением поведением – способ разработки автоматизированных тест-кейсов, в котором основное внимание уделяется

корректности работы бизнес–сценариев, а не отдельным деталям функционирования приложения.

- На основе (знания) источников ошибок:

- Тестирование предугадыванием ошибок – тесты разрабатываются на основе опыта тестировщика и его знаний о том, какие дефекты типичны для тех или иных компонентов или областей функциональности приложения.

- Эвристическая оценка – техника тестирования удобства использования, направленная на поиск проблем в интерфейсе пользователя, представляющих собой отклонение от общепринятых норм.

- Тестирование добавлением ошибок – в приложение специально добавляются заранее известные, специально продуманные ошибки с целью мониторинга их обнаружения и устранения и, таким образом, формирования более точной оценки показателей процесса тестирования.

- На основе выбора входных данных:

- Тестирование на основе классов эквивалентности – направлено на сокращение количества разрабатываемых и выполняемых тест-кейсов при сохранении достаточного тестового покрытия (выявление наборов эквивалентных тест-кейсов).

- Тестирование на основе граничных условий – на основе классов эквивалентности, позволяющая выявить специфические значения исследуемых параметров, относящиеся к границам классов эквивалентности.

- Доменное тестирование – на основе классов эквивалентности и граничных, позволяющая эффективно создавать тест-кейсы, затрагивающие несколько параметров одновременно.

- Парное тестирование – тест-кейсы строятся по принципу проверки пар значений параметров вместо того, чтобы пытаться проверить все возможные комбинации всех значений всех параметров.

- Тестирование на основе ортогональных массивов – инструментальная техника попарного и  $N$  – комбинационного тестирования, основанная на использовании т.н. «ортогональных массивов» (двумерных

массивов, обладающих следующим свойством: если взять две любые колонки такого массива, то получившийся «подмассив» будет содержать все возможные попарные комбинации значений, представленных в исходном массиве).

- На основе среды выполнения:
  - Тестирование в процессе разработки – тестирование, выполняемое непосредственно в процессе разработки приложения и/или в среде выполнения, отличной от среды реального использования приложения.
  - Операционное тестирование (рассмотрено ранее).
  - Тестирование на основе кода (на основе структур).
  - Тестирование по потоку управления – семейство техник тестирования, в которых тест-кейсы разрабатываются с целью активации и проверки выполнения различных последовательностей событий, которые определяются посредством анализа исходного кода приложения.
  - Тестирование по потоку данных – семейство техник тестирования, основанных на выборе отдельных путей из потока управления с целью исследования событий, связанных с изменением состояния переменных.
  - Тестирование по диаграмме или таблице состояний – тест-кейсы разрабатываются для проверки переходов приложения из одного состояния в другое.
  - Инспекция (аудит) кода – семейство техник повышения качества кода за счёт того, что в процессе создания или совершенствования кода участвуют несколько человек.
- Тестирование на основе структур кода предполагает возможность исследования логики выполнения кода в зависимости от различных ситуаций и включает в себя:
  - Тестирование на основе выражений – проверяется корректность (и сам факт) выполнения отдельных выражений в коде.
  - Тестирование на основе ветвей – проверяется выполнение отдельных ветвей кода.

- Тестирование на основе условий – проверяется выполнение отдельных условий.
- Тестирование на основе комбинаций условий – проверяется выполнение сложных (составных) условий.
- Тестирование на основе отдельных условий, порождающих ветвление («решающих условий») – проверяется выполнение таких отдельных условий в составе сложных условий, которые в одиночку определяют результат вычисления всего сложного условия.
- Тестирование на основе решений – проверяется выполнение сложных ветвлений (с двумя и более возможными вариантами). Несмотря на то что «два варианта» сюда также подходит, формально такую ситуацию стоит отнести к тестированию на основе условий.
- Тестирование на основе путей – проверяется выполнение всех или некоторых специально выбранных путей в коде приложения.
  - Тестирование на основе (моделей) поведения приложения:
    - Тестирование по таблице принятия решений – тест-кейсы разрабатываются на основе т.н. таблицы принятия решений, в которой отражены входные данные (и их комбинации) и воздействия на приложение, а также соответствующие им выходные данные и реакции приложения.
    - Тестирование по диаграмме или таблице состояний (рассмотрено ранее).
    - Тестирование по спецификациям (рассмотрено ранее).
    - Тестирование по моделям поведения приложения – исследование приложения (и разработка тест-кейсов) строится на некой модели: таблице принятия решений, таблице или диаграмме состояний, пользовательских сценариев, модели нагрузки и т.д.
    - Тестирование на основе вариантов использования – тест-кейсы разрабатываются на основе вариантов использования.

- Параллельное тестирование – поведение нового (или модифицированного) приложения сравнивается с поведением эталонного приложения.
- Тестирование на основе случайных данных – входные данные, действия или даже сами тест-кейсы выбираются на основе (псевдо) случайных значений так, чтобы соответствовать операционному профилю – подмножеству действий, соответствующих некоей ситуации или сценарию работы с приложением.
- А/В–тестирование – техника тестирования, в которой исследуется влияние на результат выполнения операции изменения одного из входных параметров [5].

## **1.2. Деятельность компании ООО «ТомскАСУпроект»**

Данная работа проводилась в компании ООО «ТомскАСУпроект», которая была основана в мае 2010 года. Компания занимается разработкой и адаптацией программных продуктов, автоматизирующих работу проектных, изыскательских и ресурсодобывающих организаций в области геоинформационных систем и систем автоматизированного проектирования.

Также оказывает услуги по подготовке картографических материалов и наполнению ГИС, конвертации геоданных, классификации топографических планов согласно отраслевым стандартам, камеральной обработке данных, полученных в ходе геологических и геодезических изысканий.

Осуществляет поставку готовых решений САПР и ГИС от ведущих мировых производителей: Autodesk, Bentley, ESRI, PitneyBowes.

Основными заказчиками являются: ПАО «НК «Роснефть», ОАО «ТомскНИПИнефть», ОАО «Томскнефть» ВНК, ООО «РН-СахалинНИПИморнефть», ООО «РН-Юганскнефтегаз», ПАО «ВСНК», ФГАОУ ВО НИ ТПУ и другие.

Использует следующие технологии:

- web-разработка: .NET, ASP.NET MVC 5, WebApi 2, JSON, HTML, CSS, JS, NodeJS, jQuery, ORM (NHibernate, Entity Framework);
- СУБД: MS SQL, ORACLE, PostgreSQL и др;
- desktop-разработка: .NET, WPF, WCF, AutoCAD .NET.

Собственные программные продукты:

- серия TapCAD: Topos, Design;
- серия TapCAD/Roads: Дороги, Осадка Болота, Осадка Грунт, Экспликация, Ген. план;
- серия TapGEO: ZIS, Portal, GDB.

В штате компании находится около 30 специалистов: руководители проектов, программисты, системные аналитики, тестировщики/технические писатели, специалисты по обработке данных. Из них около 15 разработчиков и три тестировщика, использующих технологию ручного тестирования.

### **1.3. Тестируемые приложения**

В данной работе проводится тестирование следующих программных продуктов:

1. desktop-модуль программной системы «CAD-GIS» «Автоматизированная система для создания ведомости подсчета объемов работ по автомобильным дорогам и генеральным планам» использующий стандартную функциональность базовой САПР AutoCAD Civil 3D;

2. web-приложение «Корпоративная геоинформационная система» ООО «РН – Пурнефтегаз».

Проблема решения, когда нужно закончить тестирования является одной из самых сложных. Исчерпывающее тестирование (т. е. испытание всех входных значений) невозможно. Таким образом, возникает экономическая проблема выбора конечного числа тестов, дающих максимальную вероятность обнаружения ошибок для данных затрат.

Возникшие в ходе проведения испытаний замечания фиксировались в трекер-систему Redmine (PM), которую использует данная компания. Там

формируется отчет с перечнем замечаний, требующих устранения. При исправлении замечаний приложение отправляется на повторное тестирование.

Условием полного завершения тестирования принято считать устранение разработчиками всех найденных ошибок, и успешное завершение всех написанных тестовых сценариев после этого.

### **1.3.1. «Автоматизированная система для создания ведомости подсчета объемов работ по автомобильным дорогам и генеральным планам»**

Данная программа обеспечивает оперативное ведение реестра работ, реестра спецификаций работниками СГИО и отдела автомобильных дорог ОАО «ТомскНИПИнефть», используя стандартную функциональность базовой САПР AutoCAD Civil 3D. Пользователь может добавлять элементы дороги в реестр, редактировать и удалять из реестра.

Модуль имеет интерфейс, представленный на рисунке 1.1.

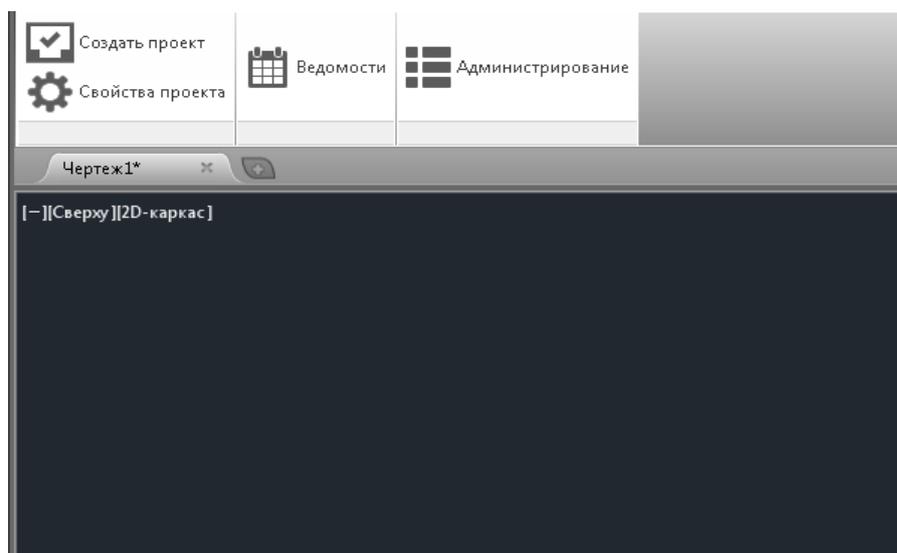


Рисунок 1.1 – Интерфейс модуля Полотно. Ведомости

Тестирование проводилось совместными усилиями, поэтому в рамках данной работы описаны результаты испытаний только около 50% функциональности модуля.

В Таблице №1.1 представлен перечень функций и испытаний:

Таблица 1.1 – Перечень функций и испытаний

№	Объект испытаний/ компонент объекта испытаний	Описание
1	Создать проект	Создает новый проект.
2	Свойства проекта	Позволяет редактировать свойства текущего проекта.
3	Создание ведомостей	Создание и редактирование ведомости В3, В2.
Работа с ведомостями		
1	Добавить сооружение	Добавляет сооружение.
2	Удалить сооружение	Удаляет выбранное сооружение
3	Экспорт в MS Word	Экспортирует выбранные ведомости в MS Word.
4	Добавить элемент дороги из реестра	Добавляет элемент дороги из реестра.
5	Добавить пустую строку	Добавляет пустую строку.
6	Удалить строку	Удаляет строку из текущей ведомости.
7	Добавить ссылку в ячейку	Добавляет ссылку в ячейку.
8	Удалить ссылку из ячейки	Удаляет ссылку из ячейки.
9	Назад	Отмена последнего действия.
10	Вперед	Переход на одно действие вперед.
12	Очистить расчет	Очищает окно расчета.

### 1.3.2. «Корпоративная геоинформационная система»

Web-приложение «Корпоративная геоинформационная система» ООО «РН – Пурнефтегаз», предназначено для обслуживания задач централизованного сбора, учета, хранения и обновления пространственных данных и для обеспечения доступа к электронным картам с использованием web-технологий широкому кругу сотрудников [1].

Приложение имеет интерфейс, представленный на рисунке 1.2.

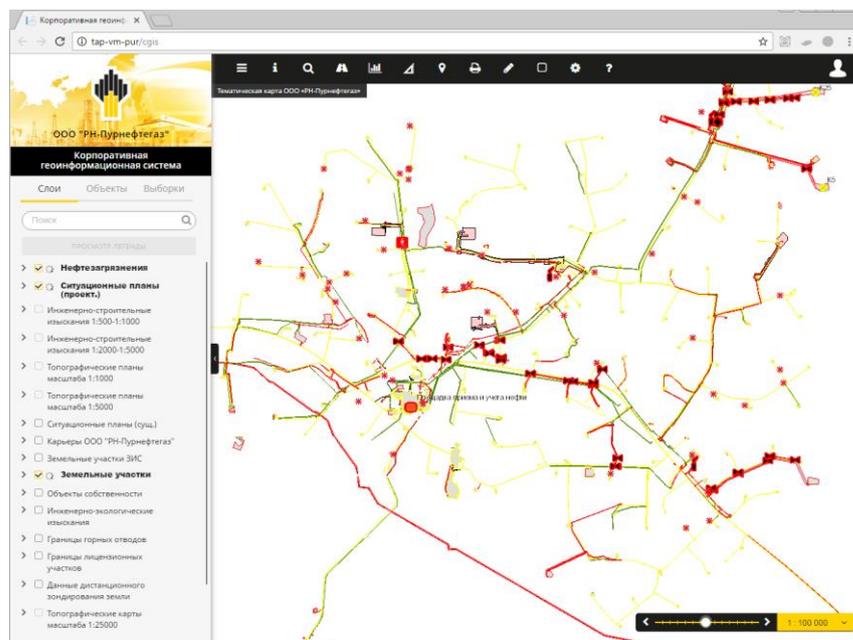


Рисунок 1.2 – Интерфейс приложение «КГИС»

В Таблице №1.2 представлен перечень функций и испытаний:

Таблица 1.2 – Перечень функций и испытаний

№	ОБЪЕКТ ИСПЫТАНИЙ/ КОМПОНЕНТ ОБЪЕКТА ИСПЫТАНИЙ	ОПИСАНИЕ	НАИМЕНОВАНИЕ МЕТОДИКИ/ИСПЫТАНИЯ
1	Работа с картой	Возможность работы с картой, с возможностью масштабирования.	Работа с картой
2	Дерево слоев и объектов	Возможность работы со слоями и объектами карты	Дерево слоев и объектов
3	Работа с выборками	Возможность работы с выборками, объектами выборок, пользователями, которые могут просматривать выборку.	Работа с выборками
4	Кнопка «Меню»	Инструмент «Меню», позволяющий сохранять карты, открывать новые и загружать обзорные карты.	Кнопка «Меню»
5	Просмотр информации об объектах	Возможность просмотра информации об объекте, осуществления операций над объектом (добавление в выборку, как пункта маршрута, создание объекта собственности)	Инструмент «Информация об объекте»
6	Поиск объектов	Возможность поиска объектов на карте.	Инструмент «Поиск объектов»
7	Поиск маршрута	Инструмент поиска маршрута между объектами и построения графа дорог.	Поиск маршрута
8	Отображение статистики	Инструмент отображения статистики по использованию модуля КГИС.	Инструмент «Статистика»

9	Определение расстояния и просмотр координат точки	Инструмент для определения координат точки в заданной СК, расстояния от точки до точки, а также площадь и периметр объекта.	Инструмент «Измерение»
10	Переход по координатам	Инструмент для перехода по координатам	Инструмент «Переход по координатам»
11	Экспорт карт	Инструмент экспорта карты.	Экспорт карт
12	Моделирование затоплений	Инструмент моделирования затоплений	Инструмент «Моделирование затоплений»
13	Инструменты рисования	Инструменты рисования на карте.	Инструмент «Рисование»
14	Помощь	Инструмент помощь, открывающий руководство пользователя	Инструмент «Помощь»
15	Администрирование	Инструмент администрирования	Раздел «Администрирование»

## 2. АНАЛИТИКА И ТЕСТИРОВАНИЕ

### 2.1. Выбор метода тестирования

Существует множество видов тестирования, которые имеют различную классификацию. И одна из корневых – классификация по степени автоматизации, делящаяся на ручное и автоматизированное тестирование.

Ручное тестирование – тестирование, в котором тест-кейсы выполняются человеком вручную без использования средств автоматизации. От тестировщика в те или иные моменты времени требуются такие качества, как терпеливость, наблюдательность, креативность, умение ставить нестандартные эксперименты, а также умение видеть и понимать, что происходит «внутри системы», т.е. как внешние воздействия на приложение трансформируются в его внутренние процессы.

Автоматизированное тестирование – набор техник, подходов и инструментальных средств, позволяющий исключить человека из выполнения некоторых задач в процессе тестирования. Тест-кейсы частично или полностью выполняет специальное инструментальное средство, однако разработка тест-кейсов, подготовка данных, оценка результатов выполнения, написания отчётов об обнаруженных дефектах – всё это и многое другое по-прежнему делает человек [11].

Изучив информацию о том и другом виде тестирования, а так же информацию о существующих решениях для тестирования приложений и проанализировав ее, была составлена таблица 2.1 – с преимуществами и недостатками данных видов тестирования.

Таблица 2.1 – Преимущества и недостатки автоматизированного тестирования

Преимущества	Недостатки
1. Скорость выполнения тест-кейсов может значительно превосходить возможности человека. 2. Отсутствие влияния человеческого фактора в процессе выполнения тест-кейсов (усталости, невнимательности и т.д.).	1. Высокие затраты на сложные средства автоматизации, разработку и сопровождение кода тест-кейсов. 2. Автоматизация требует более тщательного планирования и

Преимущества	Недостатки
3. Минимизация затрат при многократном выполнении тест-кейсов (участие человека здесь требуется лишь эпизодически).	управления рисками, т.к. в противном случае проекту может быть нанесён серьёзный ущерб.
4. Способность средств автоматизации выполнить тест-кейсы, в принципе непосильные для человека в силу своей сложности, скорости или иных факторов.	3. Средств автоматизации крайне много, что усложняет проблему выбора того или иного средства и может повлечь за собой финансовые затраты (и риски), необходимость обучения персонала (или поиска специалистов).
5. Способность средств автоматизации собирать, сохранять, анализировать, агрегировать и представлять в удобной для восприятия человеком форме колоссальные объёмы данных.	4. В случае ощутимого изменения требований, смены технологического домена, переработки интерфейсов (как пользовательских, так и программных) многие тест-кейсы становятся безнадёжно устаревшими и требуют создания заново.
6. Способность средств автоматизации выполнять низкоуровневые действия с приложением, операционной системой, каналами передачи данных и т.д.	
7. Позволяет чаще проводить регрессионное тестирование	

Так как обоих приложениях имеется часто используемая функциональность, а так же рутинные операции, такие как переборы данных, валидационные сообщения и длинные end-to-end сценарии, то можно сделать вывод, что преимущества автоматизированного тестирования довольно значительные и превышают недостатки.

Существует несколько высокоуровневых технологий автоматизации тестирования, каждая из которых, в свою очередь, базируется на своём наборе технических решений (инструментальных средствах, языках программирования, способах взаимодействия с тестируемым приложением и т.д.). Изучив информацию об эволюции технологий ([5], [7], [8]), была составлена таблица 3.2 с расположением от самой «старой» технологии к «новой».

Таблица 2.2 – Эволюция высокоуровневых технологий автоматизации тестирования

№	ПОДХОД	ОПИСАНИЕ	НЕДОСТАТКИ	ПРЕИМУЩЕСТВА
1	Частные решения	Для решения каждой отдельной задачи пишется отдельная программа.	Много времени уходит на поддержку. Почти невозможно повторное использование и нет системности	Просто и быстро
2	Тестирование под управлением поведением (BDT)	Развитие идей тестирования под управлением данными и ключевыми словами. Отличие – в концентрации на бизнес-сценариях без выполнения мелких проверок	Такие тест-кейсы пропускают большое количество функциональных и нефункциональных дефектов, а потому должны быть дополнены классическими более низкоуровневыми тест-кейсами	Высокое удобство проверки высокоуровневых пользовательских сценариев
3	Тестирование под управлением ключевыми словами (KDT)	Из тест-кейса вонне выносятся описание его поведения.	Сложность выполнения низкоуровневых операций	Концентрация на высокоуровневых действиях. Данные и особенности поведения хранятся вонне и могут быть изменены без изменения кода тест-кейса
4	Тестирование под управлением данными (DDT)	Из тест-кейса вонне выносятся входные данные и ожидаемые результаты.	Логика тест-кейса по-прежнему строго определяется внутри, а потому для её изменения тест-кейс надо переписывать	Один и тот же тест-кейс можно повторять многократно с разными данными
5	Запись и воспроизведение (Record & Playback)	Средство автоматизации записывает	Крайне низкое качество, линейность, неподдерживаемость	Простота, высокая скорость создания тест-кейсов

№	ПОДХОД	ОПИСАНИЕ	НЕДОСТАТКИ	ПРЕИМУЩЕСТВА
	Playback)	действия тестировщика и может воспроизвести их, управляя тестируемым приложением	тест-кейсов. Требуется серьезная доработка полученного кода	
6	Использование фреймворков	Конструктор, позволяющий использовать остальные подходы	Относительная сложность (особенно в создании фреймворка)	Гибкость и мощность

Следует добавить, что «старые» решения по-прежнему используются.

В ходе анализа было принято решение остановиться на методе Record & Playback, так как он является самым распространенным подходом, а так же основное его достоинство – это простота освоения.

Во многих инструментах, реализующих этот подход, действия (нажатие кнопки мыши и т.д.) привязываются не к координатам текущего положения мыши, а к объектами HTML-интерфейса (полям ввода, кнопкам и т.д.), на которые происходит воздействие, и их атрибутами. Инструмент для тестирования автоматически воспроизводит записанные действия и сравнивает их результаты с эталонными, при том, что имеется в возможность настраивать точность сравнения. Также можно добавлять проверки, такие как условия на свойства объектов (цвет, размер, расположение, и т.д.) или на функциональность приложения (содержимое сообщения и т.д.).

## **2.2. Аналитический обзор программных средств тестирования web- и desktop-приложений**

Проанализировав существующие решения для тестирования, были выделены общие и основные критерии, которые влияют на выбор инструмента тестирования.

Сначала, нужно узнать, как хорошо инструмент для автоматизации будет распознавать элементы управления в приложении. При отсутствии такой функции, стоит найти плагин, или соответствующий модуль. Если такового нет

– от инструмента лучше отказаться. Так как при опознавании большего количества элементов экономится большее количество времени на написании и поддержке скриптов.

Затем, необходимо определить, сколько времени уходит на поддержание скриптов, написанных выбранным инструментом. Удобнее инструмент, позволяющий выносить название кнопки в переменную в начале скрипта и быстро заменять ее значение. В идеале – описать меню как класс.

Заем узнать, насколько инструмент удобен при написания новых скриптов. А именно, о количестве времени, возможности структурирования кода, читаемости кода, степень удобства среда разработки для рефакторинга (переработки кода) и т.п.

Так же в соответствии с выбранным методом тестирования, и учетом его недостатков, необходимыми критериями выбора программного средства являлось следующее:

1. Простое средство тестирования – бесплатное или имеющее невысокую стоимость
2. Возможность создания автоматических тестов с помощью записи действий – с целью снизить нагрузку на менеджера проектов, связанную с планированием и управлением рисков поставленной задачи.
3. Большее количество источников для изучения, что бы свести к минимуму риски, связанные с невыполнением поставленной задачи в срок.

### **2.2.1. Выбор ПО для тестирования desktop-приложения**

На сегодняшний день существует множество программных продуктов для тестирования desktop-приложений. Поэтому в процессе изучения и выбора средств для автоматизации тестирования, была создана сравнительная таблица, приведенная в таблице 2.3, в которой находятся более популярные, поддерживаемые и часто используемые продукты.

Отличительной особенностью выбора программного средства для тестирования модуля программной системы «CAD-GIS» «Автоматизированная система для создания ведомости подсчета объемов работ по автомобильным

дорогам и генеральным планам» являлось, ко всему прочему, возможность тестировать приложения, написанные для AutoCAD.

Таблица 2.3. Сравнительные характеристики программных продуктов для тестирования desktop-приложений

Характеристика/ Инструмент	TestComplete	Ranorex	Microsoft Test Manager + Coded UI	AutoIt	(Winium)Обе ртка над Selenium от 2ГИС	SilkTest	QuickTest Professional (UFT)
<b>Стоимость</b>	Платформа: 1067 евро - фиксированная лицензия 2675 - плавающая Десктоп: 1000 евро - фиксированная лицензия 2135 - плавающая	1990 евро - фиксированная лицензия 3490 евро - плавающая	от 500 долларов	Бесплатная	Бесплатная	6271 доллар	около 10000 долларов
<b>Пробная версия</b>	30-ти дневный триал	30-ти дневный триал	30-ти дневный триал	Бесплатная	Бесплатная	45-ти дневный триал	30-ти дневный триал
<b>Необходимость программирования</b>	низкая	низкая	низкая	высокая	высокая	низкая	низкая
<b>Поддерживаемые языки программирования</b>	C++, C#, DelphiScript, VBS, Jscript	C#, VB.NET	C#	язык, похожий на BASIC	Любые языки	VB.NET, C # и Java, 4Test (язык, похожий на C++)	VBScript
<b>Запись скриптов</b>	да	да	да	да	да	да	да
<b>Наличие справочной информации</b>	Обучающие видео, наличие учебника на русском языке	На английском (оф.сайт, видео). На русском только форумы	В основном только MSDN.	Видео (англ), Форумы, руководство пользователя на русском	Руководство пользователя на русском	Учебник, руководство пользователя на русском	Видео (англ), Форумы
<b>Подходит для AutoCAD</b>	Да, при условии подключения дополнительных модулей, написанных на AutoLisp и AutoCad ActiveX Automation	Да, но требует самостоятельной разработки дополнительных модулей	Да, но требует самостоятельн ой разработки дополнительны х модулей	Да, но при условии подключения дополнительны х библиотек	Нет	Да, но при условии подключения дополнительных модулей, написанных на AutoLisp и AutoCad ActiveX Automation	Да, но требует самостоятельной разработки дополнительных модулей

Характеристика/ Инструмент	TestComplete	Ranorex	Microsoft Test Manager + Coded UI	AutoIt	(Winium)Обе ртка над Selenium от 2ГИС	SilkTest	QuickTest Professional (UFT)
<b>Поддерживаемые технологии</b>	NET, Java, Visual C++, Visual Basic, Delphi, C++Builder, ActiveX web страниц и других приложений	.NET, WinForms, WPF, Win32, VB6, Java, Oracle Forms, MFC, ActiveX, Microsoft Visual FoxPro, Microsoft Office GUI, Microsoft Access, Microsoft Dynamics AX, Microsoft, Air, ComponentOne, Janus, Syncfusion, Telerik, SkinSoft	WPF, NET	Работает на уровне ОС	Windows Desktop (WPF, WinForms)	WinForms , WPF, SAP, Oracle Forms, AJAX, HTML5, DHTML, Apache Flex, Adobe Air, Silverlight, Java Applets	WPF, Web services, Macromedia Flex, Ajax, Delphi, .NET, J2EEWeb, Visual Basic, ActiveX, Java, Oracle, SAP Solution, TE, PowerBuilder, Siebel, PeopleSoft, VisualAge, Stingray
<b>Визуализация тестовых команд</b>	да	да	нет	нет	нет	да	да

Как видно из таблицы 2.3, наиболее подходящими продуктами оказались TestComplete, Ranorex и Microsoft Test Manager + Coded UI.

Не смотря на не самую низкую стоимость по сравнению с аналогичными продуктами, был выбран TestComplete, потому что он обладает рядом преимуществ:

1. Более адаптирован для работы с приложениями на AutoCAD. Хотя, в базовой версии ни один из программных продуктов не имеет инструментария для работы с чертежами AutoCAD, в TestComplete имеется возможность установить дополнительные модули для работы с чертежами, написанные на AutoLisp – графическом языке программирования и с применением технологии AutoCad ActiveX Automation – интерфейс, позволяющий выполнять адаптацию AutoCAD, использовать хранящиеся в чертежах данные совместно с другими приложениями, а также автоматизировать выполнение различных задач. В то время, как для остальных средств такие модули необходимо разрабатывать «с нуля».

2. Наличие обучающего пособия на русском языке. В следствии того, что время написание работы ограничено, изучение программного средства для тестирования должно было пройти за минимально короткое время. Только TestComplete имеет полноценный учебник на русском языке, в котором описаны основные возможности программы и приведены примеры создания автотестов.

### **2.2.2. Выбор ПО для тестирования web-приложения**

В процессе изучения и выбора средств для автоматизации тестирования web-приложения была создана сравнительная таблица, приведенная в таблице 2.4.

Таблица 2.4 – Сравнительные характеристики программных продуктов для тестирования web-приложений

Характеристика/ Инструмент	TestCafe	Nightwatch	Selenium IDE	Библиотека для интерпретатора Ruby WATIR	Canoo Webtest	HP Quicktest Professional	Ranorex
<b>Стоимость</b>	Бесплатно/ 400 долларов в год за место	Бесплатно	Бесплатно (Обучающие видео около 10 т.р.)	Бесплатно	Бесплатно	От 9000 евро за лицензию (Обучающие видео около 20 т.р.)	1990 евро – фиксированная лицензия 3490 евро – плавающая
<b>Пробная версия</b>	30–ти дневный триал	–	–	–	–	30–ти дневный триал	30–ти дневный триал
<b>Необходимость программирования</b>	Высокая/низкая	Высокая	Низкая	Высокая	Высокая	Низкая	Низкая
<b>Поддерживаемые языки программирования</b>	ES5, ES6, ES7 JavaScript	ES5 JavaScript	HTML, Java, Ruby, C#, Python	Python, C++, Java, C	Язык, основанный на XML	VBScript	Visual Basic, C#
<b>Запись скриптов</b>	Да	Да	Да	Да	Да	Да	Да
<b>Наличие справочной информации</b>	Форумы, документация на английском языке	Форумы, документация на английском языке	Обучающие видео, Форумы, руководство пользователя на русском языке	Форумы, документация на английском языке	Документация на английском языке	Обучающие видео, Форумы, руководство пользователя на английском языке	На английском (оф.сайт, видео). На русском только форумы
<b>Поддерживаемые платформы</b>	Desktop web, Mobile web	Desktop web	Desktop web	Desktop web	Desktop web	Desktop web, Mobile web	Desktop web, Mobile web

Характеристика/ Инструмент	TestCafe	Nightwatch	Selenium IDE	Библиотека для интерпретатора Ruby WATIR	Canoo Webtest	HP Quicktest Professional	Ranorex
<b>Поддерживаемые браузеры</b>	Все, поддерживающие HTML5 и ES 5+	Все, имеющие webdriver	Firefox. Запуск тестов из Selenium IDE в других браузерах с помощью интерфейса командной строки	Internet Explorer 5.5 и ниже (имеет версии для Firefox и Safari)	Имеет библиотеку, имитирующую браузер	Chrome, Firefox, Internet Explorer и Safari	Internet Explorer, Firefox, Chrome, Safari
<b>Фиксирование ошибок</b>	Уведомление/отчет	Уведомление	Отчет	Зависит от разработчика	Отчет, графики	Отчет, графики	Отчет
<b>Поддержка Record-Play</b>	Нет/да	Нет	Да	Нет	Нет	Да	Да
<b>Интеграция с системами непрерывной интеграции</b>	Да	Да	Да	Да	Нет	Рекомендовано с HP Quality Center	Да
<b>Визуализация тестовых команд</b>	Да	Да	Да	Нет	Да	Да	Да

На основании проведенного исследования, было принято решение, что наиболее выгодное средство для тестирования в данной ситуации является Selenium IDE, так как именно этот инструмент является весьма популярным, бесплатным, имеет функции записи и воспроизведения тестов, а также много источников информации об обучении. С помощью него на написание и изменение тестов уйдет относительно немного времени. Благодаря этому сократятся риски, связанные с бесполезной тратой времени в связи с отсутствием возможности в сжатые сроки переработать тестовые ситуации при изменении приложения.

### **История проекта Selenium**

Selenium – это комплект из нескольких инструментов, каждый из которых предполагает свой собственный подход к автоматизации тестирования. В совокупности набор инструментов Selenium предоставляет богатый набор возможностей, специально собранных вместе для тестирования всех типов web-приложений. Selenium предоставляет несколько вариантов для идентификации элементов интерфейса, сравнения ожидаемого и наблюдаемого поведения тестируемого приложения. Одной из ключевых особенностей Selenium является возможность запуска одних и тех же тестов в различных браузерах.

### **Набор инструментов Selenium**

Selenium состоит из нескольких инструментов, каждый из которых имеет свое назначение.

#### Selenium 2 (или Selenium WebDriver)

Selenium 2 – последнее пополнение в пакете инструментов Selenium и является основным вектором развития проекта. Это абсолютно новый инструмент автоматизации, который обеспечивает отличный набор возможностей для управления браузером, имеет более целостный и объектно-ориентированный программный интерфейс (API), а также не имеет ограничений, свойственных более ранним версиям.

Результатом слияния WebDriver и Selenium стал Selenium 2. Этот инструмент предоставляет для использования WebDriver API, а в качестве ядра

может использоваться как более новая собственная реализация WebDriver, так и реализация, которая лежала в основе Selenium 1. Кроме того, в Selenium 2 встроен интерфейс Selenium RC для обеспечения обратной совместимости. Это, в частности, позволяет без особых усилий мигрировать существующие тесты на новую версию [2].

Поддерживает следующие браузеры и операционные системы:

- Google Chrome 12.0.712.0+;
- Internet Explorer 6, 7, 8, 9, 32–битные и 64–битные версии;
- Firefox 3.0, 3.5, 3.6, 4.0, 5.0, 6, 7, 8, 9;
- Opera 11.5+;
- HtmlUnit 2.9;
- Android – 2.3+ для телефонов и планшетов;
- iOS 3+ для телефонов и 3.2+ для планшетов.

### Selenium 1 (или Selenium Remote Control)

Selenium Remote Control версия, которая включала в себя некий сервер, принимающий запросы из кода и работающий с браузером.

На протяжении длительного времени Selenium RC был основным направлением развития проекта, пока в результате слияния WebDriver и Selenium не появился Selenium 2, более новый и мощный инструмент.

Selenium 1 все еще активно поддерживается (в основном в режиме сопровождения), потому что он предоставляет некоторые возможности, которые все еще недоступны в Selenium 2, включая поддержку некоторых языков программирования (например, Perl) и поддержку всевозможных браузеров [6].

### Selenium IDE

Selenium IDE (Integrated Development Environment, встроенная среда разработки) – плагин для браузера Firefox, работающий по принципу record and play. Пользователь может записать свои действия на странице (ввод значений, клики и т.д.), и потом проигрывать эти сценарии автоматически. Основными минусами является сложность поддержки сценариев и частые проблемы со

"сложными" кейсами (например работа со всплывающими окнами). Несомненный плюс – простота использования, когда не требуется особых навыков в написании кода для написания автотестов [6].

### Selenium Grid

Selenium Grid позволяет масштабировать большие тестовые наборы, а также запускать тесты, которые необходимо выполнить в нескольких окружениях. Selenium Grid позволяет запускать тесты параллельно, то есть разные тесты могут быть запущены в одно и то же время на нескольких удаленных машинах. Это сулит два преимущества. Во-первых, если очень много тестов или время выполнения ваших тестов слишком велико, можно значительно увеличить производительность с помощью Selenium Grid, разделив тесты на несколько потоков и запуская их одновременно на нескольких серверах. Во-вторых, если тесты необходимо запускать в разных окружениях (например, в разных браузерах или операционных системах), можно настроить удаленные сервера соответствующим образом и запустить одни и те же тесты сразу в нескольких различных средах. В обоих случаях использование параллельных процессов в Selenium Grid позволяет значительно ускорить тестирование [9].

Следует добавить, что тесты, созданные в Firefox с помощью Selenium IDE, можно запустить в любом другом поддерживаемом браузере, используя специальный режим запуска Selenium RC.

### **2.3. Тестирование**

Тестирование отвечает на вопрос о соответствии разрабатываемого программного продукта требованиям, содержащих начальный замысел изделия (т.е. то, что заказал заказчик). Процесс охватывает все виды деятельности, которые необходимы для получения качественного продукта. Ошибки вносятся на каждой стадии разработки. Поэтому, каждому этапу разработки соответствует этап тестирования. Если что-то разрабатывается, то оно подвергается тестированию, а результаты тестирования определяют

соответствие разработки набору требований, предъявляемых заказчиком. При тестировании регистрируются найденные ошибки в процесс разработки. При разработке передаются новые исправленные проектные версии для тестирования.

Несмотря на то, что автоматизированное тестирование звучит очень просто, от тестировщика в те или иные моменты времени требуются такие качества, как терпеливость, наблюдательность, креативность, умение ставить нестандартные эксперименты.

Процесс автоматизации тестирования происходил на основе проверяемых требований и критерия полноты тестирования (документов [22, 23]) и делился на следующие этапы:

#### **Создание тест-кейсов.**

Тестирование на основе тест-кейсов – самый распространённый способ тестирования, который также позволяет достичь максимальной полноты исследования приложения за счёт строгой систематизации процесса, удобства применения метрик и широкого набора выработанных за десятилетия и проверенных на практике рекомендаций.

Для тестирования были созданы как позитивные тест-кейсы, так и негативные.

#### **Проведение дымового тестирования**

Данный вид тестирования проводился для определения общего уровня качества приложения и принятия решения о (не)целесообразности выполнения расширенного тестирования. Поскольку тест-кейсов на уровне дымового тестирования относительно немного, а сами они достаточно просты, но при этом очень часто повторяются, они являются хорошими кандидатами на автоматизацию. По результатам этого тестирования принимается решение о возможности и необходимости дальнейшего тестирования.

#### **Создание автоматических тестов**

**Улучшение автоматических тестов** путем их ручного редактирования и добавления различных функций.

## **Создание тестового набора**

Набором тестов является комплект тестовых сценариев. При этом пользователь имеет возможность запускать все доступные тестовые сценарии в наборе тестов как одно непрерывное пакетное задание.

## **Воспроизведение тестов**

После выполнения всех шагов запускался тестовый набор. Расхождения регистрировались, и делался вывод о том, хорошо ли функционирует приложение или регрессионное тестирование выявило проблемы.

### **2.3.1. Тестирование модуля «CAD-GIS» «Автоматизированная система для создания ведомости подсчета объемов работ по автомобильным дорогам и генеральным планам»**

Тестирование проводилось совместными усилиями, поэтому в рамках данной работы был разработан тестовый набор, покрывающий только около 50% функциональности модуля.

Частично тест-кейсы для приложения можно увидеть в приложении Б, где:

- номер – уникальный идентификатор тест-кейса. Его удобно использовать для одинакового понимания, о какой проверке идет речь;
- название – краткое описание сути проверки;
- предварительные шаги – описание действий, которые необходимо выполнить, но прямого отношения к проверке они не имеют. Если предварительных шагов нет, то секция остается не заполненной;
- шаги – описание действий, необходимых для проверки;
- ожидаемый результат – сама проверка: что ожидается получить после выполнения шагов;
- фактический результат – результат проверки: что получается после выполнения шагов.

В итоге было написано 63 тест-кейса.

На этапе дымового тестирования не было найдено серьезных ошибок, поэтому было решено перейти к следующему этапу тестирования.

Функциональные тесты в TestComplete создавались в форме тестов на основе ключевых слов (keyword tests) или в форме скриптов. Тесты обоих видов записывались и создавались с нуля с помощью встроенных в TestComplete редакторов. Интерфейс программы представлен на рисунке 2.1.

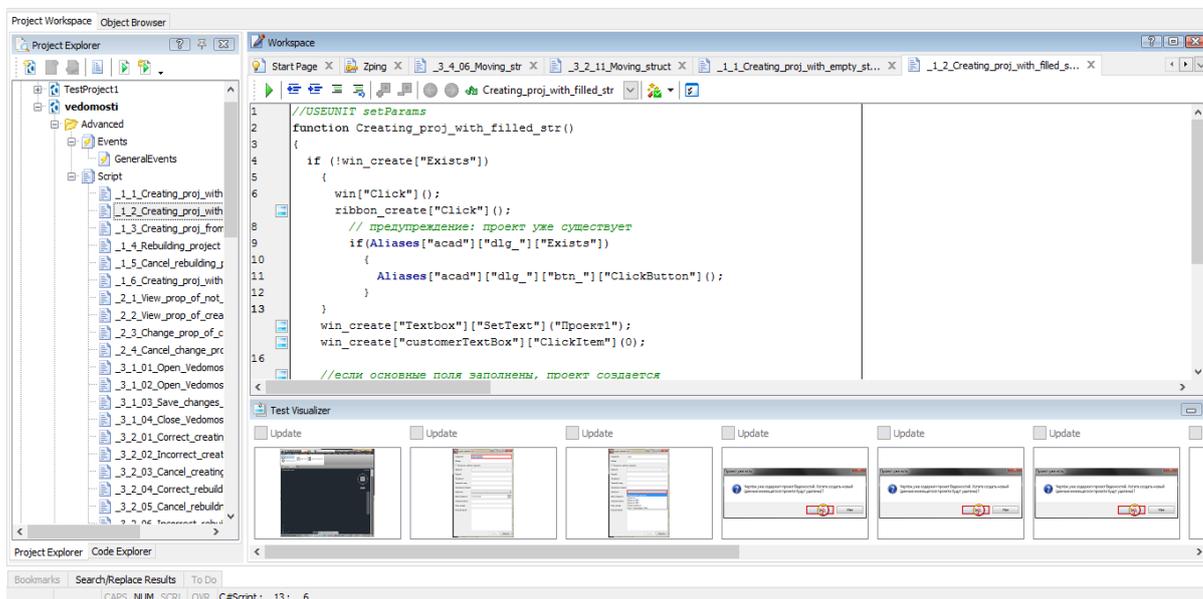


Рисунок 2.1 – Интерфейс программы TestComplete

Для написания тестов был выбран язык C#Script – язык, который имеет мало общего с языком программирования C#.

Автотесты создавались на основе ключевых слов, затем генерировались в скрипты и далее дополнялись различными проверками о том, что тестируемое окно открыто, необходимые поля заполнены и т.д.

Во время выполнения тестов, TestComplete формирует детальный отчет о всех выполненных действиях над приложением, прохождении контрольных точек, обнаруженных ошибках и т.п. Отчет содержит не только список сообщений, но и изображения экрана, что позволяет быстрее определить, что же происходило в тестируемой программе в процессе тестирования.

При написании автотестов возникали следующие проблемы:

1. При каждом запуске тестируемой программы автоматически генерируется новый код для этого диалогового приложения, поэтому использование написанных ранее автотестов было невозможным. Так как при инициализации оконных приложений Windows присваивает каждый раз коды



- ожидаемый результат – сама проверка: что ожидается получить после выполнения шагов;
- автоматизация – пометка об существовании законченного теста;
- замечания – результат проверки: что получается после выполнения шагов.

В итоге было написано 248 тест-кейсов.

На этапе дымового тестирования не было найдено серьезных ошибок, поэтому было решено перейти к следующему этапу тестирования.

Язык Selenium представляет собой набор команд, составляющих тесты. Используя его можно проверять наличие элементов интерфейса пользователя по их HTML-тегам, проверить определенный контент, работу гиперссылок, полей ввода, меню, отправляемых форм, табличных данных и прочее. Также, команды Selenium поддерживают проверки размеров окон, позиций курсора мыши, работу с диалоговыми окнами, элементами Ajax, всплывающими окнами, обработку событий и другие функции современных web-приложений.

Тестовые сценарии, решено написать в текстовом формате HTML. Они выглядят как HTML-таблица из трех столбцов. В первом столбце содержится команда Selenium, во втором ее цель, в третьем значение переменной. Второй и третий столбцы необязательны для заполнения, однако они должны находиться в таблице. Каждая строка является новой командой Selenium.

Пример созданного тестового сценария в текстовом формате HTML в виде таблицы можно увидеть на рис. 2.3:

Admin coords		
click	css=button[tooltip-title='Администрирование']	
click	//div[@id='gridAdministrativeDialogModal']/div[2]/div[2]/form/div[2]/ul/li[3]/span	
click	id=add-sr	
click	css=div#tab-spatial-references table > tbody > tr:last-child	
type	css=div#tab-spatial-references table > tbody > tr:last-child > td.administration__col-1 > input[type="text"]	Autotest
type	css=div#tab-spatial-references table > tbody > tr:last-child > td[data-field='description'] input	1234
click	css=div#tab-spatial-references table > tbody > tr:last-child > td[data-field='type'] div.jq-selectbox__select-text	
click	css=div#tab-spatial-references table > tbody > tr:last-child > td[data-field='type'] div.jq-selectbox__select-text	
click	css=div#tab-spatial-references table > tbody > tr:last-child > td[data-field='type'] li:contains('МЕТР.')	
click	css=div#tab-spatial-references table > tbody > tr:last-child > td:nth-child(4) button.sr-success	
verifyText	css=div#tab-spatial-references table > tbody > tr:last-child span.administration__value	Autotest
verifyText	css=div#tab-spatial-references table > tbody > tr:last-child td:nth-child(2) span	1234
runScript	window.SELENIUM_TEST_ENV = true;	
click	id=saveAdministrativeSettings	
waitForVisible	css=span.gritter-title:contains('Масштабы сохранены!')	
waitForVisible	css=div.gritter-without-image > p:contains('Системы координат сохранены')	

Рисунок 2.3 – Тестовый сценарий в текстовом формате HTML

В сгенерированные с помощью инструмента Record–Play тестовые сценарии, был добавлен код, для выполнения разнообразных функций. Такие изменения сценариев тестирования, как условное ветвление, рефакторинг и обработка исключительных ситуаций для проверки параметров web-страницы. Следует отметить, что недостаток данного средства для автоматизации тестирования, как и многих других, в том, что почти для всех команд Selenium не может распознавать автоматически цель, которая идентифицирует элементы на странице web-приложения. Она выглядит как определение типа локатора (метода поиска) с последующим определением искомой позиции: locatorType=location. Поэтому для поиска элементов использовались дополнительно такие языки, как XPath и CSS.

Затем был создан тестовый набор, с помощью которого пользователь имеет возможность запускать все тестовые сценарии в Selenium IDE последовательно. Тестовые наборы в Selenium IDE можно определить, также используя обычный HTML файл. Список тестовых сценариев задается таблицей, где каждая строка содержит путь к файлу тестового сценария.

В результате воспроизведения тестов было найдено 66 ошибок.

### 3. РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

В ходе проведенной работы было использовано много методов тестирования (см. пункт 1.1). Так как не было высоких навыков и знаний в языках написания программ и полного открытого исходного кода программ, использовался метод тестирования *черного ящика*. Так, как приложения ещё не полностью доработаны и основное тестирование выполняется организацией–разработчиком, следовательно, использовалось *альфа-тестирование*. Заказчиком было заявлено тестирование всей функциональности, поэтому использовалось *расширенное и функциональное тестирование*. Для нахождения наибольшего числа ошибок было использовано как *негативное*, так и *позитивное* тестирование. Так, как было необходимо проверить работу всего приложения как единого целого, собранного из частей, использовалось *системное* тестирование.

Так, как было принято решение не только о выпуске приложений, но и о дальнейшей их поддержке и усовершенствования их в новых версиях, использовалось *регрессионное* тестирование. Важным аспектом так же является и то, что по мере возрастания количества итераций тестирования, количество багов и их критичность уменьшались.

#### 3.1. Результаты автоматизации

После разработки автотестов, они загружались на Gitlab, который позволяет отслеживать версию тестов, а также оставлять комментарии тестировщикам с подробным описанием внесенных изменений.

Для того, что бы любой работник компании мог запустить данные тесты, они публиковались на CI-сервере TeamCity, который позволяет запускать тесты на специально созданной виртуальной машине для тестирования. TeamCity позволяет запускать тесты по расписанию, а также хранит историю запусков в отчеты, что позволяет оперативно реагировать на ошибки и не требуют дополнительного вмешательства тестировщика.

Так же ООО «ТомскАСУпроект» использует для работы трекер-систему Redmine (PM). Данная информационная система предназначена для фиксации различных ошибок, заданий, поручений, работ, а также осуществления контроля за ходом их исполнения ответственными исполнителями.

Под задачей в данной информационной системе подразумевается объект системы PM, отражающие суть ошибки, задачи, поручения или работы.

Каждая ошибка имеет свое описание и исполнителя, которому поручается устранить их. К каждой можно прикреплять файл со скриншотом ошибки. Так же для каждой выставляется приоритет – чем выше приоритет, тем более критична ошибка.

Для более удобной автоматизации процесса, были составлены чек-листы для каждого тестируемого ПО, которые отражали покрытие функциональности автотестами. Желтым отмечены пункты, по которым написаны автотесты.

### **3.1.1. Ошибки модуля «CAD-GIS» «Автоматизированная система для создания ведомости подсчета объемов работ по автомобильным дорогам и генеральным планам»**

Чек-лист модуля можно видеть на рис. 3.1.

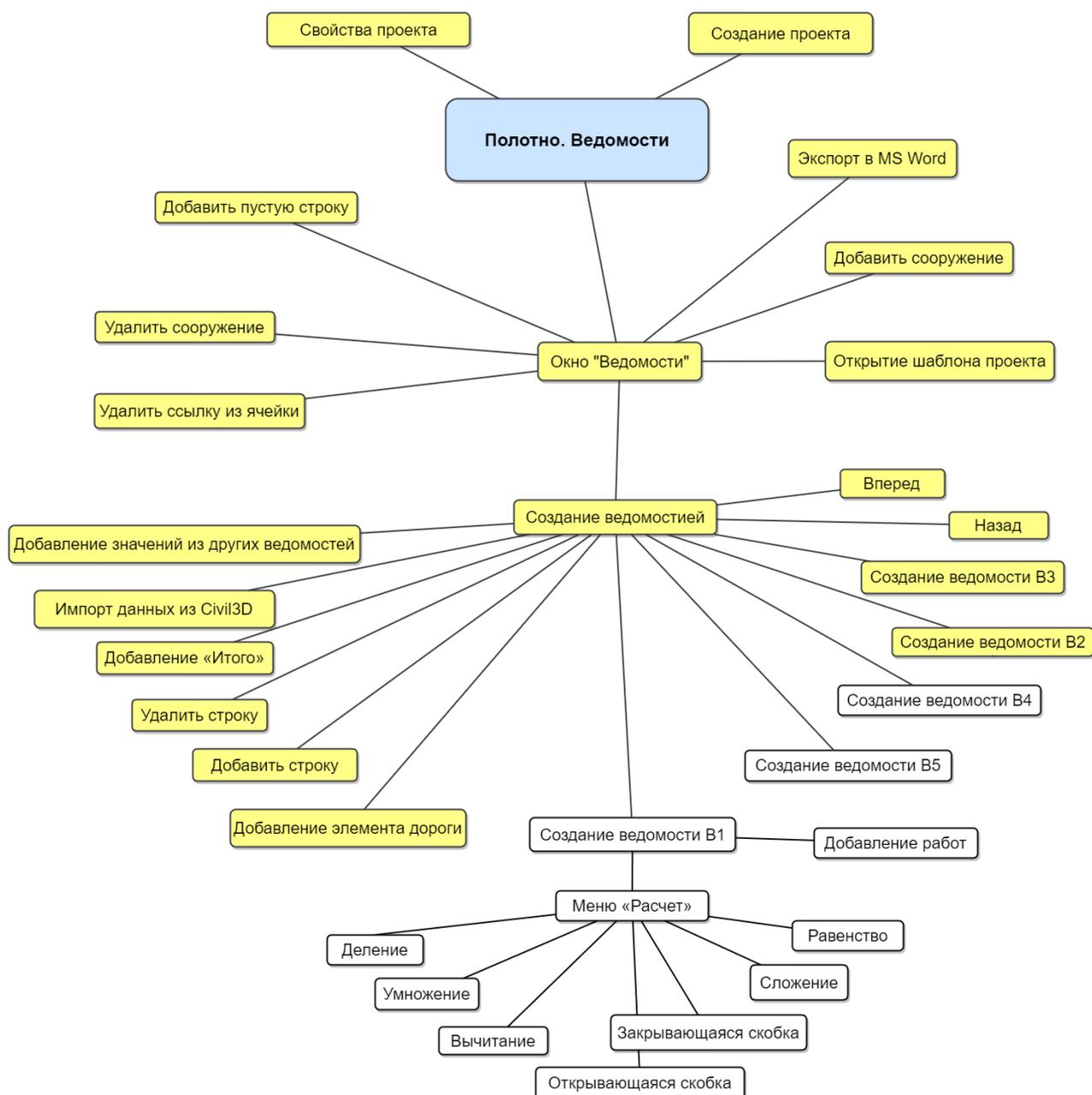


Рисунок 3.1 – Чек-лист покрытия функциональности модуля

Как видно из рисунка, написанные автотесты покрывают около 50% функциональности модуля, как и было запланировано.

Во время первого запуска тестового набора обнаружено 23 ошибки, которые были внесены в РМ. Некоторые из них можно видеть на рис. 3.2. Все ошибки можно найти в текстовом документе с таблицей тест-кейсов.

Описание				Цитировать
Ошибки, найденные при тестировании				
Подзадачи				Добавить
Ошибка #1422: Ведомости. Ошибка выбора значений при расчете B1	Новая	Онищенко Михаил		
Задание #1454: Ведомости. Ошибка удаления записей из ведомости B2	Новая	Онищенко Михаил		
Задание #1470: Ведомости. Ошибка отображения отмены действия	Новая	Онищенко Михаил		
Ошибка #1706: B3. Ошибка при изменении радиуса	Новая			
Ошибка #1707: Возможен ввод отрицательных значений длин	Новая	Онищенко Михаил		
Ошибка #1708: Ошибка при пересчете зависимых значений	Новая	Онищенко Михаил		
Ошибка #1710: Разный результат при вводе букв(и иных символов) в числовые значения	Новая	Онищенко Михаил		
Ошибка #1711: Ошибки при изменении значений в столбцах «От ПК+» «До ПК+»	Новая	Онищенко Михаил		
Ошибка #1712: Разные результаты при изменении значений в ячейках	Новая	Онищенко Михаил		
Задание #1840: Ошибка удаления сооружения	Новая	Онищенко Михаил		
Связанные задачи				Добавить

[Редактировать](#)
[Трудозатраты](#)
[Следить](#)
[Копировать](#)
[Удалить](#)  
 Экспортировать в [Atom](#) | [PDF](#)

Рисунок 3.2 – Задача в РМ. «Ошибки Ведомостей»

### 3.1.2. Ошибки «Корпоративной геоинформационной системы» ООО «РН – Пурнефтегаз»

Чек-лист «КГИС» можно видеть на рис. 3.3.

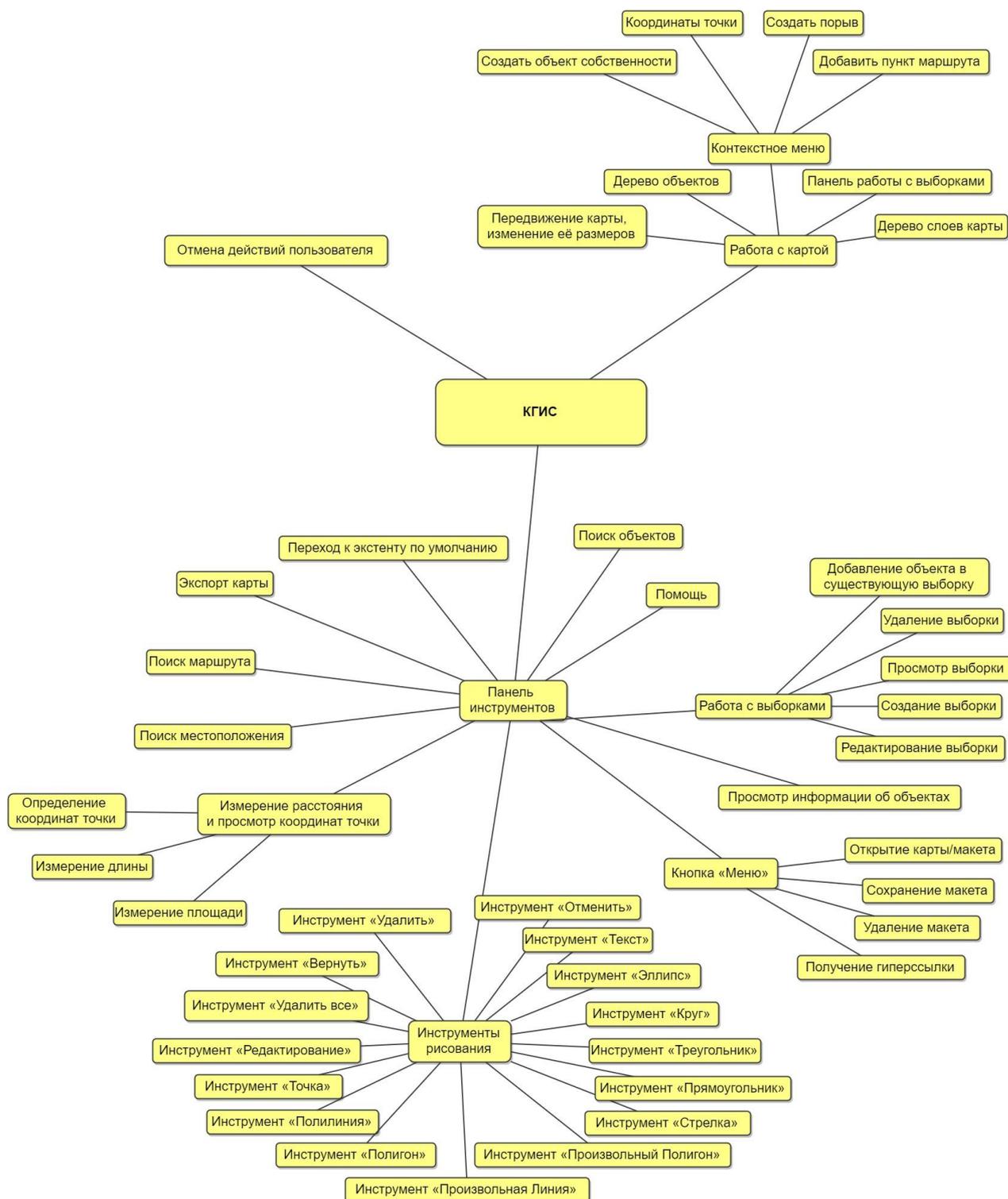


Рисунок 3.3 – Чек-лист покрытия полной функциональности «КГИС»

Как видно из рисунка, написанные автотесты покрывают 100% функциональности модуля.

При первом запуске тестового набора было обнаружено 66 ошибок, которые можно увидеть в документе с таблицей написанных тест-кейсов. Затем ошибки были занесены в РМ и отправлены на исправления программистам.

## Пример нескольких занесенных ошибок в РМ:

[ТНН] ГИСП Томскнефть ВНК 2017 год

Обзор Действия Оперативный план **Задачи** Диаграмма Ганта Календарь Документы Wiki Файлы

**Задачи** ↓ Другое

Фильтры

Статус открыто Добавить фильтр

Опции

Применить Очистить Сохранить

#	Трекер	Статус	Категория	Тема	Назначена	Дата начала	Дата завершения	Вес	Оценка трудозатрат	Трудозатраты	Готовность
4296	Ошибка	Требуется оценка	ГИСП	● Сохранение макета. Не работает получение ссылки и сохранение карты	Кардашов Сергей	14.03.2017		0		0.50	0%
4250	Ошибка	Требуется оценка	ГИСП	● Не включается видимость слоев через панель Администрирование.	Кардашов Сергей	13.03.2017		0		1.00	0%
4249	Ошибка	Требуется оценка	ГИСП	● Не видно текст кнопок при наведении мыши	Кардашов Сергей	13.03.2017		0		0.75	0%
4248	Ошибка	Требуется оценка	ГИСП	● Открытие карты/ макета приводит к ошибке	Кардашов Сергей	13.03.2017		0		0.50	0%
4241	Ошибка	Требуется оценка	ГИСП	● [Рисование] Не корректно отображается окно настройки цветов	Кардашов Сергей	10.03.2017		0		0.50	0%
4240	Ошибка	Требуется оценка	ГИСП	● Не работает переход к объектам из дерева Объектов	Кардашов Сергей	10.03.2017		0		0.50	0%

Рисунок 3.4 – Ошибки, занесенные в РМ

Пример подробного описания ошибки и ее скриншот показан на рис. 3.5,

3.6.

[ТНН] ГИСП Томскнефть ВНК 2017 год

Обзор Действия Оперативный план **Задачи** Диаграмма Ганта Календарь Документы Wiki Файлы

**Ошибка #4296** Показать все поля

Требуется уточнение  Оценить задачу  Приступить к работе

**Сохранение макета. Не работает получение ссылки и сохранение карты** « Предыдущее | 1/18

Добавил(а) Борисова Анастасия 14.03.2017 18:33.

Люди: Сроки: Прочее:

Назначена: Кардашов Сергей Трудозатраты: 0.50 час Статус: Требуется оценка

Приоритет: Нормальный

Готовность: 0%

Категория: ГИСП

Вес: 0

Возвраты: 0

Описание

Шаги:

- 1) Открыть <http://tap-vm-tn/cgis>
- 2) Нажать Меню -> Сохранение макета
- 3) Нажать Получить ссылку

Промежуточный результат: Появляется сообщение об ошибке (см.скрин)

- 4) Ввести название карты в поле Название
- 5) Нажать Сохранить

Результат: Ничего не происходит.

Файлы

 (139 КБ) Добавил(а) Борисова Анастасия 14.03.2017 18:33.

Рисунок 3.5 – Ошибка «Сохранение макета. Не работает получение ссылки и сохранение карты»

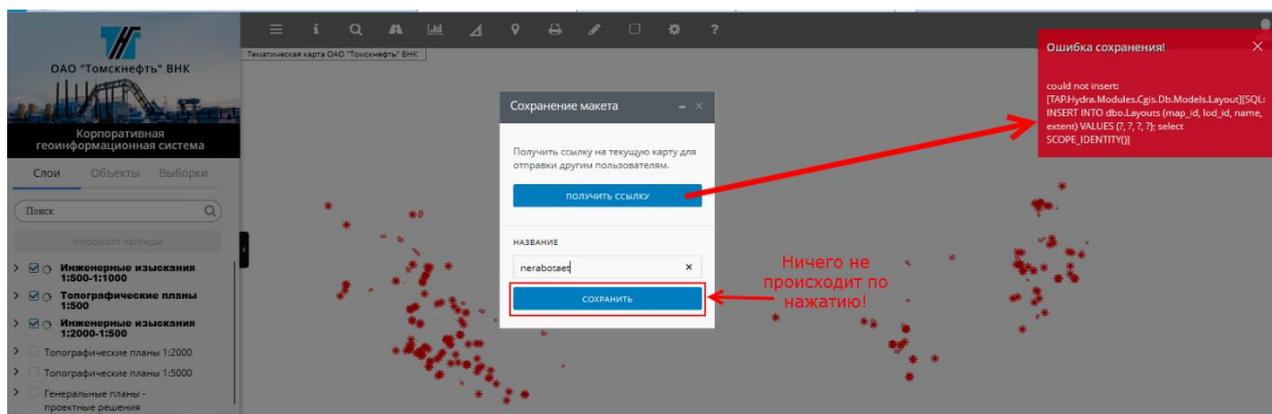


Рисунок 3.6 – Скриншот ошибки «Сохранение макета. Не работает получение ссылки и сохранение карты»

По мере исправления ошибок проводились повторные запуски тестов, во время которых находилось куда меньшее количество багов.

### 3.2. Эффективность автоматизированного тестирования

Для проверки целесообразности автоматизации процесса тестирования необходимо посчитать затраты на ручное тестирование и затраты на автоматизацию. Расчеты будут производиться исходя из данных, полученных в ходе опроса работников компании ООО «ТомскАСУпроект». Следует учесть, что компания небольшая и в отдел тестирования входит всего три человека, занимающиеся ручным тестированием. В ходе опроса были получены данные, необходимые для подсчетов.

- В данной компании не было еще тестировщика, занимающегося автоматизацией, но примерная их оплата составляла бы от 250 рублей в час, в то время как оплата ручного тестировщика составляет 200 рублей в час (суммы указаны без учета налогов на оплату труда и других издержек).
- Оба проекта рассчитаны как минимум еще на три года. Регрессионное тестирование проводится как минимум раз в неделю. При тестировании в сумме на два ПО используется 311 тестов.
- Новую версию модуля «Ведомости» планируется выпускать раз в год.
- Приложение «КГИС» будет разрабатываться в двух видах для ООО «РН-Пурнефтегаз» и ОАО «Томскнефть» с 98% одинаковых функций. Но всё-

таки при необходимо проводить тестирование обоих приложений, чтобы исключить ошибки, связанные с конфликтами модулей. Новые версии было решено выпускать раз в полгода.

- На подготовку к циклу обычно уходит порядка 45 минут (распределение задач и др. орг. задачи). Среднее время, необходимое одному тестировщику на выполнение одного тест-кейса, составляет 15 минут.

- При каждом прогоне в среднем 5% тестов имеют отрицательные результаты. На определение ошибки для каждого теста у ручного тестировщика уходит около 2 минут, в то время как при автоматизированном тестировании анализ ошибки занимает 10 минут. При ручном тестировании тестировщик сразу видит, где именно и при каких входных данных произошла ошибка, а при автоматизированном тестировании эту информацию необходимо искать в отчете программы.

- Вероятность появления необходимости изменения одного теста между циклами тестирования оценена в 3%, Среднее время, необходимое для актуализации одного ручного теста около 6 минут. Для актуализации автоматизированного теста потребуется 20 минут.

- Автоматизация одного теста оценивается в 1 час.

Оценка целесообразности автоматизации тестирования производится с помощью подсчета затрат на ручное и автоматизированное тестирование и их сравнение [17], [10]. Это, конечно, примерные расчеты, так как их параметры в процессе работы постоянно меняются.

При расчете эффективности не учитывались офисные расходы, управленческие, административные (общие налоги, маркетинг, бухгалтерия и т.п.), так как они будут уплачиваться вне зависимости от вида тестирования, выбранного компанией.

Для расчёта инвестиций, необходимых для внедрения и эксплуатации автоматизированных тестов за выделенный период ( $I_p$ ), используется следующая формула:

$$I_p = I_0 + C_0 + \sum_{n=1}^k (C_e + C_a + C_m)$$

$I_0$  - Оценка стартовых инвестиции (ПО и т.п.). В нашем случае компания потратила 10 т.р. на подробные обучающие уроки по Selenium IDE, и TestComplete за лицензию 136 т.р. в год. В данной работе использовалась бесплатная пробная версия.

$C_0$  - Оценка стоимости разработки и отладки библиотеки автоматических тестов, рассчитывающаяся как произведение среднего времени, нужного для написания одного автоматизированного теста одним разработчиком тестов (в часах), умноженное на цену его рабочего часа и на общее количество тестов, которые предстоит автоматизировать.

$C_0$  для модуля «Ведомости» =  $1 * 250 * 63 = 15750$  р.

$C_0$  для «КГИС» =  $1 * 250 * 248 = 62000$  р.

$k$  - Это количество планируемых прогонов тестов (циклов тестирования) за всё оставшееся время жизненного цикла продукта.

Для модуля «Ведомости» тестирование будет проводиться примерно раз в неделю в среднем 8 месяцев до разработки каждой новой версии и равно  $3 \text{ года} * 8 \text{ мес.} * 4 \text{ нед.} = 96$  раз.

Для «КГИС» тестирование будет проводиться примерно раз в неделю в среднем 4 месяца до разработки каждой новой версии (8 мес. в год) и будет тестироваться для двух компаний. Получается в сумме  $3 \text{ года} * 8 \text{ мес.} * 4 \text{ нед.} * 2 = 192$  раза.

$C_e$  - Оценка стоимости одного прогона всех автоматизированных тестов, которая рассчитывается как время, необходимое для подготовки к выполнению тестирования, сложенное со средним временем выполнения одного теста одним тестировщиком, умножено на цену рабочего часа и на общее количество тестов. В нашем случае эта переменная принята за 0, поскольку подготовка к циклу тестирования не требуется, а само тестирование не требует дополнительного контроля со стороны работника и происходит полностью автономно.

С<sub>а</sub> - Оценка затрат на анализ результатов одной итерации цикла автоматизированного тестирования. Вычисляется как оценка доли отрицательных тестов, умноженная на количество тестов, на среднее время, необходимое для анализа причин отрицательной оценки одного теста одним тестировщиком, и на цену одного рабочего часа тестировщика.

С<sub>а</sub> для модуля «Ведомости» = 0,05\*63\*0,16\*250= 126 р/ч.

С<sub>а</sub> для «КГИС» = 0,05\*248\*0,16\*250= 496 р/ч.

С<sub>т</sub> - Оценка стоимости поддержания автоматизированных тестов в рабочем и актуальном состоянии. Рассчитывается как вероятность появления необходимости изменения одного теста между циклами тестирования, умноженная на количество тестов, на среднее время, необходимое для актуализации одного теста и на цену одного рабочего часа тестировщика.

С<sub>т</sub> для модуля «Ведомости» = 0,03\*63\*0,30\*250= 141,75 р/ч.

С<sub>т</sub> для «КГИС» = 0,03\*248\*0,30\*250= 558 р/ч.

Таким образом, итоговая стоимость внедрения и эксплуатации системы автоматизированных тестов равна:

Для модуля «Ведомости»: 136000 \* 3 + 15750 + 96 \* (0 + 126 + 141,75) = 449 454 рублей.

Для «КГИС»: 10000 + 62000 + 192 \* (0 + 496 + 558) = 274 368 рублей.

Оценка стоимости ручного тестирования (G<sub>р</sub>) представлена в следующей формуле:

$$G_p = G_0 + \sum_{n=1}^k (G_e + G_a + G_m)$$

G<sub>0</sub> - Оценка стоимости разработки базы тест-кейсов для ручного тестирования равна нулю.

k - Это количество планируемых прогонов тестов (циклов тестирования) за всё оставшееся время жизненного цикла продукта.

G<sub>e</sub> - Оценка стоимости однократного выполнения цикла ручного тестирования, которая рассчитывается как среднее время, затрачиваемое на подготовку к тестированию плюс среднее время, нужное для выполнения

одного тест-кейса одним тестирующим, умноженное на суммарное количество кейсов и на цену одного рабочего часа тестирующего.

$$Ge \text{ для модуля «Ведомости»} = (0,75+63*0,25)*200= 3300 \text{ р/ч.}$$

$$Ge \text{ для «КГИС»} = (0,75+248*0,25)*200= 12 550 \text{ р/ч.}$$

$G_a$  - Оценка стоимости анализа результатов для одного прогона цикла ручного тестирования. Вычисляется как оценка средней доли отрицательных тестов в прогоне, умноженная на количество тестов, на среднее время, необходимое для анализа причин отрицательной оценки одного теста одним тестирующим, и на цену одного рабочего часа тестирующего;

$$G_a \text{ для модуля «Ведомости»} = 63*0,05*0,25*200= 158 \text{ р.}$$

$$G_a \text{ для «КГИС»} = 248*0,05*0,25*200= 620 \text{ р.}$$

$G_m$  - Оценка стоимости поддержания ручных тестов в актуальном состоянии. Рассчитывается как вероятность появления необходимости изменения одного теста между циклами тестирования, умноженная на количество тестов, на среднее время, необходимое для актуализации одного теста и на цену одного рабочего часа тестирующего [17].

$$G_m \text{ для модуля «Ведомости»} = 63*0,03*0,1*200=38 \text{ р.}$$

$$G_m \text{ для «КГИС»} = 248*0,03*0,1*200 = 149 \text{ р.}$$

Итоговая стоимость затрат на ручное тестирование равна:

$$\text{Для модуля «Ведомости»}: 0 + 96 * (3300 + 158 + 38) = 335 616 \text{ рублей.}$$

$$\text{Для «КГИС»}: 0 + 192 * (12550 + 620 + 149) = 2 557 248 \text{ рублей}$$

Подводя итоги, посчитаем выгоду:

$$\text{для модуля «Ведомости»}: 335 616 - 449 454 = -113 838 \text{ рублей.}$$

$$\text{для «КГИС»}: 2 557 248 - 274 368 = + 2 282 880 \text{ рублей}$$

Следовательно, можно прийти к заключению, что для маленькой компании и, особенно, для маленьких проектов автоматизированное тестирование дорогими программными средствами не выгодно. Но при большом объеме задач и имея простое средство автоматизации, автоматизированное тестирование весьма выгодно.

## 4. ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ

### 4.1. Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения

#### 4.1.1. Потенциальные потребители результатов исследования

Целевым предприятием разрабатываемых тестов, в первую очередь является компания ООО «ТомскАСУпроект», занимается разработкой и адаптацией программных продуктов, автоматизирующих работу проектных, изыскательских и ресурсодобывающих организаций в области геоинформационных систем. Потенциальным покупателем тестируемого ПО является предприятие ООО «РН-Пурнефтегаз» и ОАО «Томскнефть».

Целевым рынком для результатов выпускной квалифицированной работы является рынок по изучению, применению и реализации методов и инструментов автоматизированного тестирования ПО.

Выделены два наиболее значимых критерия сегментирования. Сегментировать рынок услуг по тестированию можно по следующим критериям: вид тестирования, размер тестируемого ПО (см. таблица 4.1).

Таблица 4.1 – Карта сегментирования рынка по наиболее важным критериям

		Вид тестирования	
		Автоматизированное	Ручное
Размер ПО	Большое		
	Среднее		
	Малое		

 - существование целесообразных решений и методов в сфере тестирования и его окупаемость.

Автоматизированное тестирование целесообразно применять для средних и крупных программных продуктов, так как его окупаемость

происходит за счет частых выполнений одинаковых действий, частого изменения функционала и модернизации, а так же выпусков новых версий.

Так, как тестируемые программы имеют достаточно большой функционал, в результате сегментирования рынка было решено выбрать автоматизированный вид тестирования.

## 4.2. Организация и планирование работ

### 4.2.1. Структура работ в рамках научного исследования

При организации работ в рамках данной работы необходимо планировать занятость каждого участника проекта в работе. На данном этапе определяется полный перечень работ, распределение времени работ между всеми участниками. В качестве структуры, показывающей необходимые данные, используется линейный график работ, представленный в таблице 4.2.

Таблица 4.2 – Перечень этапов работ и распределение исполнителей

Основные этапы	№	Содержание работ	Исполнители
Разработка задания на выполнение	1	Постановка целей и задач	НР
	2	Составление списка задач для выполнения.	НР, И
	3	Подбор и изучение материалов по теме.	НР, И
Исследование	4	Анализ и выбор методов и средств тестирования.	НР, И
Тестирование desktop-приложения	5	Ознакомление с тестируемым desktop-приложением	И
	6	Изучение выбранного средства тестирования	И
	7	Написание тест-кейсов	И
	8	Написание тестового набора	И
	9	Фиксация ошибок и их отправка разработчикам	И
	10	Повторное тестирование	И
Тестирование web-приложения	11	Ознакомление с тестируемым web-приложением	И
	12	Изучение выбранного средства тестирования	И
	13	Написание тест-кейсов	И
	14	Написание тестового набора	И
	15	Фиксация ошибок и их отправка разработчикам	И
	16	Повторное тестирование	И
	17	Анализ результатов исследований	НР, И
Анализ и оформление результатов	18	Оформление работы	И
	19	Подведение итогов	НР, И

НР – научный руководитель;

И – инженер – в его роли действуют исполнитель ВКР.

#### 4.2.2. Продолжительность этапов 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 последующим формулам:

$$t_{ожi} = \frac{3t_{мини} + 2t_{маxi}}{5} \quad (4.1)$$

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

$t_{мини}$  – минимально возможная трудоемкость, чел.-дн.;

$t_{маxi}$  – максимально возможная трудоемкость, чел.-дн.

Для построения линейного графика необходимо рассчитать длительность этапов в рабочих днях, а затем перевести ее в календарные дни. Расчет продолжительности выполнения каждого этапа в рабочих днях ведется по формуле:

$$T_{рД} = \frac{t_{ож}}{K_{ВН}} \cdot K_{Д}, \quad (4.2)$$

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

$K_{ВН}$  – коэффициент выполнения работ, учитывающий влияние внешних на соблюдение предварительно определенных длительностей, в рамках данной ВКР установим  $K_{ВН} = 1$ ;

$K_{Д}$  – коэффициент, учитывающий дополнительное время на компенсацию непредвиденных задержек и согласование работ, примем  $K_{Д} = 1,1$ .

Расчет продолжительности этапа в календарных днях ведется по формуле:

$$T_{кД} = T_{рД} \cdot T_{к}, \quad (4.3)$$

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

$T_K$  – коэффициент календарности, позволяющий перейти от длительности работ в рабочих днях к их аналогам в календарных днях. Он рассчитывается по формуле:

$$T_K = \frac{T_{КАЛ}}{T_{КАЛ} - T_{ВД} - T_{ПД}}, \quad (4.4)$$

Где  $T_{КАЛ}$  – календарные дни,  $T_{КАЛ} = 365$ ;

$T_{ВД}$  – выходные дни,  $T_{ВД} = 52$ ;

$T_{ПД}$  – праздничные дни,  $T_{ПД} = 10$ .

Подставив значения в формулу 4.4, получим следующий результат:

$$T_K = \frac{365}{365 - 52 - 10} = 1,205. \quad (4.5)$$

Все рассчитанные значения представлены в таблице 4.3.

Для наглядного отображения графика и распределения работ между участниками проекта использована диаграмма Ганта.

Для построения диаграммы календарных работ используются данные, из таблицы 4.3.

Таблица 4.3 - Временные показатели проведения научного исследования

Название работы	Трудоёмкость работ						Длительность работ в рабочих днях $T_{рд}$		Длительность работ в календарных днях $T_{кд}$	
	$t_{min}$ , чел-дни		$t_{max}$ , чел-дни		$t_{ожид}$ , чел-дни		НР	И	НР	И
	НР	И	НР	И	НР	И				
Постановка целей и задач. Получение исходных данных.	2	–	5	–	3,2	–	3,52	–	4,241	–
Составление списка задач для выполнения.	2	2	5	5	3,2	3,2	3,52	3,52	4,241	4,241
Подбор и изучение материалов по теме.	1	3	3	15	1,8	7,8	1,98	8,58	2,385	10,338
Анализ и выбор методов и средств тестирования.	–	5	–	15	–	9		9,9	0	11,92
Ознакомление с тестируемым desktop-приложением	–	2	–	6	–	3,6		3,96	0	4,771
Изучение выбранного средства тестирования	–	7	–	12	–	9		9,9	0	11,929
Написание тест-кейсов	–	5	–	14	–	8,6		9,46	0	11,399
Написание тестового набора	–	8	–	17	–	11,6		12,76	0	15,375
Фиксация ошибок и их отправка разработчикам	–	1	–	2	–	1,4		1,54	0	1,855
Повторное тестирование	–	1	–	3	–	1,8		1,98	0	2,385
Ознакомление с тестируемым web-приложением	–	2	–	6	–	3,6		3,96	0	4,771
Изучение выбранного средства тестирования	–	7	–	12	–	9		9,9	0	11,929
Написание тест-кейсов	–	5	–	14	–	8,6		9,46	0	11,399
Написание тестового набора	–	12	–	21	–	15,6		17,16	0	20,677
Фиксация ошибок и их отправка разработчикам	–	1	–	2	–	1,4		1,54	0	1,855
Повторное тестирование	–	1	–	3	–	1,8		1,98	0	2,385
Анализ результатов исследований	1	3	4	6	2,20	4,2	2,42	4,62	2,916	5,567
Оформление работы	–	3	–	6	–	4,2		4,62	0	5,567
Подведение итогов	1	1	3	5	1,8	2,6	1,98	2,86	2,385	3,446
<b>Итого:</b>	<b>7</b>	<b>69</b>	<b>20</b>	<b>164</b>	<b>12,2</b>	<b>107</b>	<b>13,42</b>	<b>117,7</b>	<b>16,17</b>	<b>141,828</b>

Таблица 4.4 – Календарный план-график проведения работ

№Э	T <sub>кд</sub> НР	T <sub>кд</sub> И	Январь		Февраль			Март			Апрель			Май		Июнь		
			2	3	1	2	3	1	2	3	1	2	3	2	3	1	2	3
1	4,241	4,241	■															
2	2,385	10,338	■	■														
3	0	11,92			■	■												
4	0	4,771				■												
5	0	11,929				■	■											
6	0	11,399					■	■										
7	0	15,375						■	■									
8	0	1,855							■									
9	0	2,385								■								
10	0	4,771									■							
11	0	11,929										■	■					
12	0	11,399											■	■				
13	0	20,677												■	■	■		
14	0	1,855													■			
15	0	2,385														■		
16	2,916	5,567															■	■
17	0	5,567																■
18	2,385	3,446																■
19	4,241	4,241																■

НР – ■ ; И – ■

### 4.2.3. Расчет накопления готовности проекта

В данном пункте описан расчет текущих состояний работы над проектом. Величина накопления готовности работы показывает, на сколько процентов по окончании текущего этапа выполнен общий объем работ по проекту в целом.

Степень готовности определяется формулой (4.5)

$$СГ_i = \frac{ТР_i^H}{ТР_{общ.}} = \frac{\sum_{k=1}^i ТР_k}{ТР_{общ.}} = \frac{\sum_{k=1}^i \sum_{j=1}^m ТР_{kj}}{\sum_{k=1}^i \sum_{j=1}^m ТР_{kj}}, \quad (4.6)$$

где  $ТР_i^H$  – накопленная трудоемкость  $i$ -го этапа проекта по его завершении;

$ТР_{общ.}$  – общая трудоемкость проекта;

$ТР_k$  – трудоемкость  $k$ -го этапа проекта,  $k = \overline{1, i}$ ;

$ТР_{kj}$  – трудоемкость работ, выполняемых  $j$ -м участником на  $k$ -м этапе, здесь  $j = \overline{1, m}$  – индекс исполнителя, в данной работе  $m = 2$ , так как в разработке проекта участвуют научный руководитель и исполнитель.

Применительно к таблице (4.3) величины  $ТР_{ij}$  ( $ТР_{kj}$ ) находятся в столбцах ( $Т_{РД(НР)}$ ,  $j = 1$ ) и ( $Т_{РД(И)}$ ,  $j = 2$ ).  $ТР_{общ.}$  равна сумме чисел из итоговых клеток этих столбцов. Пример расчета  $ТР_i$  (%) и  $СГ_i$  (%) на основе этих данных содержится в таблице (4.4).

Таблица 4.5 – Нарастание технической готовности работы и удельный вес каждого этапа

Этап	ТР <sub>i</sub> , %	СГ <sub>i</sub> , %
Постановка целей и задач. Получение исходных данных.	2,68	2,68
Составление списка задач для выполнения.	5,37	8,05
Подбор и изучение материалов по теме.	8,05	16,10
Анализ и выбор методов и средств тестирования.	7,55	23,65
Ознакомление с тестируемым desktop-приложением	3,02	26,67
Изучение выбранного средства тестирования	7,55	34,22

Этап	ТР <sub>i</sub> , %	СГ <sub>i</sub> , %
Написание тест-кейсов	7,21	41,44
Написание тестового набора	9,73	51,17
Фиксация ошибок и их отправка разработчикам	1,17	52,34
Повторное тестирование	1,51	53,85
Ознакомление с тестируемым web-приложением	3,02	56,87
Изучение выбранного средства тестирования	7,55	64,42
Написание тест-кейсов	7,21	71,64
Написание тестового набора	13,09	84,73
Фиксация ошибок и их отправка разработчикам	1,17	85,90
Повторное тестирование	1,51	87,41
Анализ результатов исследований	5,37	92,78
Оформление работы	3,52	96,30
Подведение итогов	3,69	100,00

### **4.3. Расчет сметы затрат на выполнение проекта**

В состав бюджета выполнения работ входит стоимость всех расходов, необходимых для их выполнения. При формировании бюджета используется группировка затрат по следующим статьям:

- материалы и покупные изделия;
- заработная плата;
- социальный налог;
- расходы на электроэнергию (без освещения);
- амортизационные начисления;
- оплата услуг связи;
- прочие (накладные расходы) расходы.

#### **4.3.1. Расчет накопления готовности проекта**

Данная статья включает стоимость всех материалов, используемых при разработке проекта: приобретаемые материалы, необходимые для создания ВКР. Также включаются затраты на канцелярские принадлежности и диски.

Все расчеты сведены в таблицу 4.6.

Таблица 4.6 – Расчет материальных затрат

Наименование	Единица измерения	Количество	Цена за ед., руб.	Затраты на материалы, (З <sub>м</sub> ), руб.
		И	И	И
диск CD/RW	Шт	2	30	60
бумага формата А4	Уп.	1	290	290
Тонер для принтера	Шт.	1	250	250
Итого				600

Транспортно-заготовительные расходы (ТЗР) составляют 5 % от отпускной цены материалов, тогда расходы на материалы с учетом ТЗР равны

$$С_{\text{мат}} = 600 \cdot 1,05 = 630 \text{ руб.}$$

### 4.3.3. Расчет заработной платы

Данная статья расходов включает в себя заработную плату научного руководителя и студента, а также премии и доплаты. Расчет выполняется на основе трудоемкости выполнения каждого этапа и величины месячного оклада исполнителя.

Величина месячного оклада научного руководителя (МОНР) получена из открытых данных, размещенных на официальном сайте Национального исследовательского Томского политехнического университета. Величина месячного оклада инженеров (МОИ) берется как месячный оклад инженера кафедры.

Основной расчет фонда заработной платы выполняется по формуле:

$$ЗП_{\text{дн-т}} = \text{МО}/N, \quad (4.7)$$

где МО – месячный оклад, руб.;

$N$  – количество рабочих дней в месяц, при шестидневной рабочей неделе –  $N = 24,91$ , а при пятидневной рабочей неделе –  $N = 20,58$ .

Среднедневная заработная плата научного руководителя равна:

$$ЗП_{\text{дн-т}} = \frac{26\,300}{24,91} = 1\,055,8 \frac{\text{руб.}}{\text{раб. день}}$$

А среднедневная тарифная заработная плата инженеров равна

$$ЗП_{\text{дн-т}} = \frac{7\,864,11}{20,58} = 382,12 \frac{\text{руб.}}{\text{раб. день}}$$

Затраты времени по каждому исполнителю в рабочих днях взяты из таблицы 4.3. Для перехода от тарифной суммы заработка исполнителя, связанной с участием в проекте, к соответствующему полному заработку необходимо будет тарифную сумму заработка исполнителя, связанной с участием в проекте умножить на интегральный коэффициент. Интегральный коэффициент находится по формуле:

$$K_{\text{и}} = K_{\text{пр}} \cdot K_{\text{доп.ЗП}} \cdot K_{\text{р}}, \quad (4.8)$$

где  $K_{\text{пр}}$  – коэффициент премий,  $K_{\text{пр}} = 1,1$ ;

$K_{\text{доп.ЗП}}$  – коэффициент дополнительной зарплаты, при шестидневной рабочей неделе  $K_{\text{доп.ЗП}} = 1,188$ , а при пятидневной рабочей неделе  $K_{\text{доп.ЗП}} = 1,113$ ;

$K_{\text{р}}$  – коэффициент районной надбавки,  $K_{\text{р}} = 1,3$ .

Результаты вычислений представлены в таблице 4.7.

Таблица 4.7 – Затраты на заработную плату

Исполнитель	Оклад, руб./месэ	ЗП <sub>дн-т</sub> , руб./раб.день	Затраты времени, раб.дни	Коэффициент	Фонд з/платы, руб.
НР	26 300	1 055,8	14	1,699	25113,25
И	7 864,11	382,12	118	1,59	71693,354
Итого:					96806,6044

#### 4.3.4. Отчисления на социальные нужды

Отчисления на социальные нужды установлены в размере 30,2% от заработной платы. Размер отчислений рассчитываются по формуле:

$$C_{\text{соц}} = C_{\text{ЗП}} \cdot 0,302, \quad (4.9)$$

где  $C_{\text{ЗП}}$  – размер заработной платы.

Подставив необходимые значения в формулу 4.10 получим:

$$C_{\text{соц}} = 96806,6044 \cdot 0,302 = 29235,59 \text{ руб.}$$

#### 4.3.5. Расчет затрат на электроэнергию

Затраты на электроэнергию рассчитываются по формуле:

$$C_{\text{эл.об.}} = P_{\text{об}} \cdot t_{\text{об}} \cdot Ц_{\text{э}}, \quad (4.10)$$

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

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

$\text{Ц}_э$  – тариф на 1 кВт·час. Для ТПУ  $\text{Ц}_э = 5,8$  руб./кВт·час.

Время работы оборудования вычисляется на основе итоговых данных таблицы 4.3 для инженера ( $T_{рд}$ ) из расчета, что продолжительность рабочего дня равна 8 часов.

$$t_{об} = T_{рд} \cdot K_t, \quad (4.11)$$

где  $K_t$  – коэффициент использования оборудования по времени,  $K_t = 0,9$ .

Мощность, потребляемая оборудованием, определяется по формуле:

$$P_{об} = P_{ном} \cdot K_C, \quad (4.12)$$

где  $K_C$  – коэффициент загрузки;

$P_{ном}$  – номинальная мощность оборудования, кВт. Для технологического оборудования малой мощности  $K_C = 1$ .

Таблица 4.8 – Затраты на электроэнергию технологическую

Наименование оборудования	Время работы оборудования $t_{об}$ , час	Потребляемая мощность $P_{об}$ , кВт	Затраты $\text{Э}_{об}$ , руб.
ПК инженера	847,4	0,09	442,34
<b>Итого:</b>			<b>442,34</b>

#### 4.3.6. Расчет амортизационных расходов

Для расчета амортизационных расходов используется формула:

$$C_{ам} = \frac{N_A \cdot \text{Ц}_{об} \cdot t_{рф} \cdot n}{F_D}, \quad (4.13)$$

где  $N_A$  – годовая норма амортизации единицы оборудования;

$\text{Ц}_{об}$  – балансовая стоимость единицы оборудования с учетом ТЗР, стоимость ПК инженера – 49 140 руб.;

$t_{рф}$  – фактическое время работы оборудования в ходе выполнения проекта,  
 $t_{рф} = 118 \cdot 8 = 944$  часов;

$n$  – число задействованных однотипных единиц оборудования;

$F_D$  – действительный годовой фонд времени работы соответствующего оборудования,  $F_D = 298 \cdot 8 = 2384$  часа.

$N_A$  определяется по формуле:

$$N_A = \frac{1}{CA}, \quad (4.12)$$

где CA – срок амортизации, который можно получить из постановления правительства РФ «О классификации основных средств, включенных в амортизационные группы» Для электронно-вычислительной техники CA свыше 2 лет до 3 лет включительно. В данной работе примем CA=2,5 года. Тогда

$$N_A = \frac{1}{2,5} = 0,4.$$

Таким образом,

$$C_{AM}(ПК) = \frac{0,4 \cdot 49\,140 \cdot 944 \cdot 1}{2384} = 7783,25 \text{ руб}$$

Итого начислено амортизации 7783,25 руб.

#### 4.3.7. Расчет расходов на услуги связи

Расходы на услуги связи определены наличием подключения к сети Интернет на компьютере, использованном в данной работе.

Ежемесячная оплата, согласно тарифу, составляет 350 рублей. В соответствии с таблицей 4.3, трудоемкость выполняемой задачи составляет четыре календарных месяца. Таким образом, сумма расходов на услуги связи составляет  $5 \cdot 350 = 1750$  руб. Общая сумма расходов  $C_{св} = 1750$  руб.

#### 4.3.8. Расчет прочих расходов

Прочие расходы следует принять равными 10% от суммы всех предыдущих расходов. Они находятся по формуле:

$$C_{проч} = (C_{мат} + C_{ЗП} + C_{соц} + C_{эл.об.} + C_{AM} + C_{св}) \cdot 0,1, \quad (4.13)$$

Где  $C_{мат}$  – расходы на материалы, руб.;

$C_{ЗП}$  – основная заработная плата, руб.;

$C_{соц}$  – расходы на единый социальный налог, руб.;

$C_{эл.об.}$  – расходы на электроэнергию, руб.;

$C_{AM}$  – амортизационные расходы, руб.;

$C_{св}$  – расходы на услуги связи, руб.

Подставив полученные выше результаты, получим:

$$C_{\text{проч}} = (630 + 96\,806,6044 + 29\,235,59 + 442,34 + 7\,783,25 + 1\,750) \cdot 0,1 = 13\,664,778 \text{ руб.}$$

#### 4.3.9. Расчет общей себестоимости разработки

Проведя расчет по всем статьям сметы затрат на разработку, можно определить общую себестоимость проекта.

Таблица 4.9 – Смета затрат на разработку проекта

Статья затрат	Условное обозначение	Сумма, руб.
Материалы и покупные изделия	$C_{\text{мат}}$	630
Основная заработная плата	$C_{\text{ЗП}}$	96 806,6044
Отчисления в социальные фонды	$C_{\text{соц}}$	29 235,59
Расходы на электроэнергию	$C_{\text{эл.об.}}$	442,34
Амортизационные отчисления	$C_{\text{АМ}}$	7 783,25
Расходы на услуги связи	$C_{\text{св}}$	1 750
Прочие расходы	$C_{\text{проч}}$	13 664,778
<b>Итого:</b>		<b>150 312,562</b>

Таким образом, затраты на разработку составили  $C = 150\,312,562$  руб.

#### 4.3.10. Расчет прибыли

Результатом данной работы является создание автоматических тестов для тестируемого web-приложения и desktop-приложения. Возьмем половину расходов на тестирование web-приложения и половину расходов на desktop-приложение.

Предположим, что заработная плата у тестировщиков автоматизированным способом и ручным одинакова, а так же примерно одинаково время выполнения ручных тестов и время написания автоматических тестов.

Рассмотрим прибыль с тестов на web-приложение. В данном случае приложение будет разрабатываться в двух видах для ООО «РН-Пурнефтегаз» и ОАО «Томскнефть» с 98% одинаковых функций. Но всё-таки при наличии различий необходимо проводить тестирование обоих приложений, что бы исключить ошибки, связанные с конфликтами модулей. Так же при нахождении ошибок, приложение будет отправляться на доработку, а затем снова

тестироваться. Предположим, что таких циклов будет около трех. Так же приложение решено модернизировать и выпускать новую версию с добавлением новых функций раз в пол года. Следовательно, в первый год придется тестировать web-приложение 7 раз.

При автоматизированном тестировании после первого тестирования (после написания тестового набора) на тестирование будет уходить всего несколько часов без необходимости участия специалиста по тестированию. Следовательно, за год уйдет  $150\ 312/2 \cdot 1 = 75\ 156$  руб.

При ручном тестировании все действия придется проводить каждый раз, следовательно, уйдет  $150\ 312/2 \cdot 7 = 526\ 093$  руб.

Общая прибыль за год будет равна  $526\ 093 - 75\ 156 = 450\ 930$  руб.

Рассмотрим прибыль с тестов на dsktop-приложение. В данном случае приложение сложнее по архитектуре, чем web-приложение, следовательно, циклов доработки будет больше. Возьмем по минимуму, например, 4 цикла. Принято решение, что новые версии данного приложения с новыми функциями и улучшением будут выходить раз в год. Следовательно, при автоматизированном тестировании за год уйдет  $150\ 312/2 \cdot 1 = 75\ 156$  руб.

При ручном тестировании все действия придется проводить каждый раз, следовательно, уйдет  $150\ 312/2 \cdot 5 = 375\ 780$  руб.

Общая прибыль за год будет равна  $375\ 780 - 75\ 156 = 300\ 624$  руб.

В итоге прибыль с тестирования обоих приложений будет равна  $375\ 780 + 450\ 930 = 826\ 710$ .

#### **4.3.11. Расчет НДС**

НДС составляет 18% от суммы затрат на разработку и прибыли.

В нашем случае это  $(150\ 312 + 826\ 710) \cdot 0,18 = 175\ 864$  руб.

#### **4.3.12. Цена разработки НИР**

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

$C_{\text{НИР(КР)}} = 150\ 312 + 826\ 710 + 175\ 864 = 1\ 152\ 886$  руб.

#### **4.4. Оценка экономической эффективности проекта**

В рамках данной работы оценить экономическую эффективность проекта невозможно.

##### **4.4.1. Определение срока окупаемости**

Срок окупаемости используется, как показатель эффективности проекта. Чем меньше срок окупаемости, тем эффективнее проект. Для расчета используется формула:

$$PP = \frac{C}{PP_{\text{ч}}}, \quad (4.14)$$

где  $C$  – затраты на разработку, руб.;

$PP_{\text{ч}}$  – годовая чистая прибыль, руб.

Подставив полученные выше результаты, получим:

$$PP = \frac{150\ 312}{826\ 710} = 0,18 \text{ года или 2 месяца.}$$

Исходя из низкого срока окупаемости, можно сделать вывод, что проект эффективен.

Следует отметить, что основное направление данной работы – это не получение коммерческой выгоды, а достижение социального экономического эффекта.

##### **4.4.2. Оценка научно-технического уровня НИР**

Научно-технический уровень характеризует влияние проекта на уровень и динамику обеспечения научно-технического прогресса в данной области. Для того, чтоб бы оценить научную ценность, техническую значимость и эффективность, планируемых и выполняемых НИР, был использован метод балльных оценок. Балльная оценка заключается в том, что каждому фактору по принятой шкале присваивается определенное количество баллов. Обобщенная оценка проводится по сумме баллов по всем показателям. На ее основе делается вывод о целесообразности НИР.

Сущность метода заключается в том, что на основе оценок признаков работы определяется интегральный показатель (индекс) ее научно-технического уровня по формуле:

$$I_{\text{НТУ}} = \sum_{i=1}^3 R_i \cdot n_i, \quad (4.13)$$

где  $I_{\text{НТУ}}$  – интегральный индекс научно-технического уровня;

$R_i$  – весовой коэффициент  $i$ -го признака научно-технического эффекта;

$n_i$  – количественная оценка  $i$ -го признака научно-технического эффекта, в баллах.

Частные оценки уровня  $n_i$  и их краткое обоснование даны в таблице 4.10.

Таблица 4.10 Оценки научно-технического уровня НИР

Значимость	Фактор НТУ	Уровень фактора	Выбранный балл	Обоснование выбранного балла
0,4	Уровень новизны	Внедряемая	7	Экономит время, сокращает число машин, ресурсов и тестировщиков программного обеспечения, которые используются компанией в целях достижения высокого качества продукта.
0,1	Теоретический уровень	Разработка web-сервисов	6	Изучение области методов и средств автоматизированного тестирования
0,5	Возможность реализации	В течение первых лет	10	Сложные операции и организация тестирования и фиксации ошибок на предприятии тестируемого ПО

Интегральный показатель научно-технического уровня для данного проекта составляет:

$$I_{\text{НТУ}} = 0,4 \cdot 7 + 0,1 \cdot 6 + 0,5 \cdot 10 = 8,4.$$

По полученным данным можно сделать вывод, что проект имеет высокий уровень научно-технического эффекта.

## **5. СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ**

В разделе социальная ответственность содержится информация о производственной и экологической безопасности при выполнении и оформлении данной магистерской работы.

Выполнение в рамках магистерской диссертации проекта заключалось в анализе методов и средств автоматизированного тестирования и разработкой тестовых наборов для программного обеспечения. В соответствии с этим, производилось взаимодействие студентов-дипломников с ПК.

Целью раздела является выявление и анализ вредных и опасных факторов труда пользователя персонального компьютера (ПК) и разработка мер защиты от них, оценка условий труда и микроклимата рабочей среды. Также рассматриваются вопросы создания оптимальных условий труда, техники безопасности, пожарной профилактики и охраны окружающей среды.

### **5.1. Производственная безопасность на стадии разработки тестовых наборов**

Научно-исследовательская деятельность выполнялась в помещении организации ООО «ТомскАСУпроект», которое оснащено видео-дисплейными терминалами (ВДТ), персональными электронно-вычислительными машинами (ПЭВМ), компьютерными столами, стульями, огнетушителями, кондиционером, противопожарной сигнализацией и датчиками дыма.

Производственный фактор считается вредным, если его воздействие может привести к заболеванию человека, и считается опасным, если его воздействие может привести к травме [24].

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 для рабочей зоны программиста и пользователя. Выявленные факторы представлены в таблице 5.1.

Таблица 5.1 – Вредные и опасные производственные факторы при выполнении работ за ПЭВМ

Источник фактора, наименование видов работ	Факторы (по ГОСТ 12.0.003-74)		Нормативные документы
	Вредные	Опасные	
1) Работа за ПК	1) Отклонение показателей микроклимата; 2) Недостаточная освещенность рабочей зоны.	1) Опасность возникновения пожара; 2) Опасность поражения электрическим током.	1) СН 2.2.4/2.1.8.562-96; 2) СанПиН 2.2.4.548-96; 3) СанПиН 2.2.2/2.4.1340-03; 4) СП 52.13330.2011; 5) ГОСТ Р 12.1.019-2009 ССБТ; 6) СНиП 21-01-97.

### 5.1.1. Вредные производственные факторы

#### Отклонение показателей микроклимата

Микроклиматом производственных помещений является сочетание температуры, подвижности воздуха, влажности, температуры окружающих поверхностей и их тепловым излучением. Микроклимат определяет теплообмен организма человека и оказывает существенное влияние на функциональное состояние системы самочувствие, организма, работоспособность и здоровье.

Метеорологические условия для рабочей зоны производственных помещений регламентируются ГОСТ 12.1.005-88 «Общие санитарно-гигиенические требования к воздуху рабочей зоны» и СанПиН 2.2.4.548—96 «Гигиенические требования к микроклимату производственных помещений».

Принципиальное значение в нормах имеет отдельное нормирование каждого компонента микроклимата: температуры, влажности, скорости движения воздуха. В рабочей зоне должны обеспечиваться параметры микроклимата, соответствующие оптимальным и допустимым значениям. В таблице 5.1 [21] приведены параметры микроклимата. В нашем случае работа была легкой и проводилась в теплое время года. Температура колебалась в

пределах 21-27 градусов, относительная влажность в пределах 45-58. Следовательно, рабочая зона имела допустимые нормы.

Таблица 5.2 – Оптимальные значения характеристик микроклимата

Период года	Температура воздуха, °С		Температура поверхностей, °С		Относительная влажность, %	Скорость движения воздуха, м/с
	оптимальная	допустимая	оптимальная	допустимая		
Холодный	22-24	20-22	21-25	19-23	40-60	0,1
Теплый	23-25	20-28	22-26	19-27	40-60	0,1

### Недостаточная освещенность рабочей зоны

Недостаточная освещенность рабочей зоны является вредным производственным фактором, возникающим при работе с ПЭВМ, уровни которого регламентируются СП 52.13330.2011.

Причины недостаточной освещенности: недостаточность искусственного и естественного освещения, пониженная контрастность.

Работа с компьютером подразумевает постоянный зрительный контакт с дисплеем ПЭВМ и занимает от 80 % рабочего времени. Недостаточность освещения увеличивает количество допускаемых ошибок и утомляемость, снижает производительность труда и приводит к появлению болезней зрения.

Разряд зрительных работ программиста и оператора ПЭВМ относится к разряду III и подразряду Г (работы высокой точности). В таблице 5.2 показаны нормативы искусственного освещения при работах заданной точности.

Таблица 5.3 – Требования к освещению помещений промышленных предприятий для операторов ПЭВМ [25]

Характеристика зрительной работы	Эквивалентный размер объекта	Разряд зрительной работы	Подразряд зрительной работы	Контраст объекта фоном	Характеристика фона	Искусственное освещение		
						Освещённость, лк		
						При системе комбинированного освещения		При системе общего освещения
						всего	В том числе от общего	
Высокой точности	0,264	III	Г	Средний, большой	Светлый, средний	400	200	200

При создании и поддержании благоприятных условий освещения рабочие места для операторов ПЭВМ должны соответствовать санитарно-

эпидемиологическим правилам СанПиН 2.2.2/2.4.1340-03. Рабочее помещение должно иметь естественное и искусственное освещение, которое должно соответствовать показателям, из таблицы 5.6. Что бы рассеять естественное освещение, нужно на окнах рабочих помещений использовать жалюзи. Источником искусственного освещения должны быть лампы накаливания – для местного освещения и люминесцентные лампы. [26].

### **5.1.2. Опасные производственные факторы**

#### **Опасность возникновения пожара**

Возникновение пожара – опасный производственный фактор, т.к. пожар наносит большой материальный ущерб на предприятии, и часто сопровождается травмами и несчастными случаями. Регулирование пожаробезопасности производится СНиП 21-01-97.

В помещении с ПЭВМ повышен риск возникновения пожара, из-за множества факторов: присутствие большого количества электронных схем, устройств электропитания, устройств кондиционирования воздуха; возможные неисправности электрооборудования, освещения, или неправильная их эксплуатация может послужить причиной пожара.

Виды источников воспламенения:

- искры при разряде статического электричества;
- искра от электрооборудования;
- искра от удара и трения;
- открытое пламя [18].

Для профилактики организации действий при пожаре нужно проводить комплекс организационных мер: регулярно обеспечивать проверки пожарной сигнализации, первичных средств пожаротушения; проводить инструктаж и тренировки по действиям в случае пожара; не загромождать или блокировать пожарные выходы; выполнять правила техники безопасности и технической эксплуатации электроустановок; во всех служебных помещениях устанавливать «Планы эвакуации людей при пожаре и других ЧС», которые регламентируют действия персонала при возникновении пожара.

Что бы предотвратить пожар, помещение с ПЭВМ нужно оборудовать первичными средствами пожаротушения: углекислотными огнетушителями типа ОУ-2 или ОУ-5; пожарной сигнализацией, и, при необходимости, автоматической установкой объемного газового пожаротушения [18].

### **Опасность поражения электрическим током**

Поражение электрическим током - опасный производственный фактор и, так как программист имеет дело с электрооборудованием, то электробезопасности на его рабочем месте необходимо уделять особое внимание. Нормы электробезопасности на рабочем месте регламентируются СанПиН 2.2.2/2.4.1340-03, вопросы требований к защите от поражения электрическим током освещены в ГОСТ Р 12.1.019-2009 ССБТ.

Электробезопасностью является система организационных и технических мероприятий и средств, которая обеспечивает защиту людей от вредных и опасных воздействий электрической дуги, электрического тока, электромагнитного поля и статического электричества.

При поражении электрическим током опасно еще и тем, что человек не имеет возможности 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.

Обучение и инструктаж безопасным методам труда, проверка знаний правил безопасности и инструкций в соответствии с занимаемой должностью применительно к выполняемой работе являются основными организационными мероприятиями по обеспечению безопасности.

К мероприятиям по предотвращению возможности поражения электрическим током можно отнести:

- с целью защиты от поражения электрическим током, который возникает между корпусом приборов и инструментом при пробое сетевого напряжения на корпус, корпуса приборов и инструментов должны быть заземлены;
  - при включенном сетевом напряжении работы на задней панели корпуса приборов должны быть запрещены;
  - все работы по устранению неисправностей должен производить квалифицированный персонал;
  - необходимо постоянно следить за исправностью электропроводки
- [3, 4].

## **5.2. Экологическая безопасность**

### **Влияние автоматизированного тестирования 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.

В ходе выполнения ВКР и дальнейшем использовании алгоритмов отсутствуют выбросы каких-либо вредных веществ в атмосферу и гидросферу, следовательно, загрязнение воздуха и воды не происходит.

Так же и люминесцентные лампы, которые применяются для искусственного освещения рабочих мест, требуют особой утилизации, т.к. в них присутствует от 10 до 70 мг ртути, относящейся к чрезвычайно-опасным химическим веществам и которая может стать причиной отравления живых организмов, а также загрязнения атмосферы, гидросферы и литосферы. Срок работы таких ламп около 5-ти лет, после чего их нужно сдавать на переработку в специальные пункты приема.

Во время разработки и написания ВКР образовывался мусор, такой как: канцелярские принадлежности, бумажные отходы, неисправные комплектующие персонального компьютера, люминесцентные лампы.

## **Мероприятия по защите окружающей среды**

Для уменьшения вредного влияния на литосферу необходимо производить сортировку отходов и обращаться в службы по утилизации для дальнейшей переработки или захоронения.

В основном, организации, занимающиеся приёмом и утилизацией ртути содержащих отходов, принимают люминесцентные лампы в массовых количествах. Лампа состоит из электронного блока — выгодный компонент для реставрации и утилизации; колба и цоколь также ценное сырьё. По стране утилизацией «ртутных» ламп занимаются более 50 фирм, но единственное их условие — деньги, которые вы должны заплатить за вывоз.

Такие лампы нельзя выкидывать в мусоропровод или уличные контейнеры, а нужно отнести в свой районный ДЕЗ (Дирекция единичного заказчика) или РЭУ (Ремонтно-эксплуатационное управление), где есть специальные контейнеры. Там они принимаются бесплатно, основанием должна служить утилизация в соответствии с Управлением Федеральной службы по надзору в сфере защиты прав потребителей и благополучия человека по Томской области. Пункты приёма отработавших свой срок люминесцентных ламп по городам можно найти в интернете. [25]

Переработка макулатуры представляет собой многоэтапный процесс, цель которого заключается в восстановлении бумажного волокна и, зачастую, других компонентов бумаги (таких как минеральные наполнители) и использование их в качестве сырья для производства новой бумаги.

Организации, занимающиеся покупкой сломанных компьютеров на запчасти, готовы платить за запчасти деньги, которые они сэкономят на покупке новых деталей, необходимых для ремонта. Компьютерная техника (или ее компоненты) может также заинтересовать тех, кто скупает старые платы и радиодетали для получения из них после переработки драгоценных и редких металлов. Многие сетевые гипермаркеты электронной техники периодически устраивают программу утилизации. Также можно самостоятельно отвезти

сломанный компьютер в пункт приема металлолома не составит труда. Такие точки приема есть в каждом городе.

Также следует отменить, что разрабатываемая в рамках ВКР система позволяет создавать сложные по структуре документы технических задания для компаний, разрабатывающих информационные системы, вести документооборот и переписку между участниками непосредственно в самой системе в электронном формате, что экономит бумагу.

### **5.3. Безопасность в чрезвычайных ситуациях**

#### **5.3.1. Основные чрезвычайные ситуации в офисном помещении**

Чрезвычайные ситуации бывают техногенного, природного, биологического, социального или экологического характера.

При работе в кабинете могут возникнуть следующие классификации чрезвычайных ситуаций:

- преднамеренные/непреднамеренные;
- техногенные: взрывы, пожары, обрушение помещений, аварии на системах жизнеобеспечения/природные – связанные с проявлением стихийных сил природы;
  - экологические – это аномальные изменения состояния природной среды, такие как загрязнения биосферы, разрушение озонового слоя, кислотные дожди/ антропогенные – являются следствием ошибочных действий людей;
  - биологические – различные эпидемии, эпизоотии, эпифитотии;
  - социальные – это обстановка на определенной территории, сложившаяся в результате опасного социального явления, которое повлекло в результате человеческие жертвы, ущерб здоровью, имуществу или окружающей среде;
- комбинированные.

### **5.3.2. Типичные чрезвычайные ситуации**

#### **Пожар**

Наиболее вероятной чрезвычайной ситуацией, которая возникает при работе с ПЭВМ, является пожар, так как в современных ЭВМ очень высокая плотность размещения элементов электронных схем. Соединительные провода и кабели располагаются в большой близости друг от друга, и при протекании электрического тока по ним выделяется большое количество теплоты, поэтому возможно оплавление изоляции и возникновение возгорания.

Биологические, так как программист работает в кабинете и контактирует с большим количеством людей, в том числе с другими сотрудниками, то велик риск заражения одного сотрудника от другого (чем больше людей, тем выше риск). В связи с большим скоплением народа в одном помещении появляется необходимость в непрерывном проветривании, что приводит к образованию сквозняков, что так же может сказаться на здоровье.

Возникновение других видов ЧС – маловероятно [25].

#### **Кибертерроризм**

Терроризм – это метод, посредством которого организованная группа или партия стремятся достичь провозглашенные ими цели через систематическое использование насилия.

Компьютерный терроризм (кибертерроризм) — использование компьютерных и телекоммуникационных технологий (прежде всего, интернета) в террористических целях.

В киберпространстве используются различные способы для совершения кибертеракта:

- несанкционированный доступ к государственным и военным секретам, банковской и личной информации;
- причинение ущерба отдельным физическим элементам информационного пространства, например, разрушение сетей электропитания, создание помех;

- использование специальных программ для разрушения аппаратных средств;
- кражи или уничтожения информации, программ и технических ресурсов при преодолении защитных систем, внедрения вирусов, программных закладок;
- воздействие на программное обеспечение и информацию;
- раскрытие публикации закрытой информации и угроза ее использования;
- захват каналов СМИ для распространения дезинформации, слухов и демонстрации силы террористической организации и объявления своих требований;
- уничтожение или активное подавление линий связи, неправильная адресация, перегрузка узлов коммуникации;
- проведение информационно-психологических операций.

Использование документов и данных хранящихся в системы, включая личные переписки участников, дают киберпреступникам возможность использовать их в злых умыслах.

В результате чего, физические или юридические лица, чьи данные были использованы в злых умыслах, могут быть подвержены шантажу и вымогательству со стороны киберпреступников, что может негативно сказаться на репутации и финансовому состоянию компаний разрабатывающие информационные системы. Также в создаваемых документах технических заданий могут содержаться информации о государственной или коммерческой тайне.

### **Действия в результате возникновения чрезвычайной ситуации и мер по ликвидации ее последствий**

При работе компьютерной техники выделяется много тепла, это может приводить к пожароопасной ситуации. Серьезную опасность представляют электроизоляционные материалы, которые используются для защиты от механических воздействий отдельных радиодеталей.

В связи с этим, участки, на которых используется компьютерная техника, по пожарной опасности относятся к категории пожароопасных «В».

Меры, соблюдение которых поможет исключить с большой вероятностью возможность возникновения пожара:

- что бы понизить воспламеняемость и способность распространять пламя кабели покрывают огнезащитным покрытием;
- при ремонтно-профилактических работах строго соблюдаются правила пожарной безопасности;
- помещения, в которых должны располагаться ПЭВМ проектируют I или II степени огнестойкости;
- помещения, в которых эксплуатируются устройства ПЭВМ, должно оборудоваться первичными средствами пожаротушения и быть обеспечено инструкциями по их применению. Для пожаротушения можно использовать углекислотные огнетушители типа ОУ-2, ОУ-5(описание ниже), а также порошковый тип. Пенные огнетушители не допускается применять, так как жидкость пропускает ток;
- устройства ПЭВМ нужно устанавливать в стороне от отопительных и нагревательных приборов (на расстоянии не менее 1 м и в там, где не затруднена их вентиляция и отсутствуют прямые солнечные лучи);
- разрабатываются организационные меры по обучению персонала навыкам ликвидации пожара имеющимися в наличии средствами тушения пожара до прибытия пожарного подразделения [19].

При пожаре людям необходимо покинуть помещение в течение минимального времени.

В помещениях с компьютерной техникой, нельзя применять воду и пену так как существует опасность повреждения или полного выхода из строя электронного оборудования.

При тушении пожаров необходимо применение углекислотных и порошковых огнетушителей, обладающих высокой скоростью тушения, большим временем действия, возможностью тушения электроустановок,

высокой эффективностью борьбы с огнем. Воду разрешено применять только во вспомогательных помещениях [18].

#### **5.4. Правовые и организационные вопросы обеспечения безопасности**

##### **Описание правовых норм для работ, связанных с работой за ПЭВМ**

Регулировка отношений работник – работодатель, которые касаются оплаты труда, трудового распорядка, регулировка труда женщин, детей, людей с ограниченными способностями и т.д., осуществляется законодательством РФ, а именно трудовым кодексом РФ.

Продолжительность рабочего времени не должна превышать 40 часов в неделю.

Порядок исчисления нормы рабочего времени на определенные календарные периоды (месяц, квартал, год) в зависимости от установленной продолжительности рабочего времени в неделю определяется федеральным органом исполнительной власти, осуществляющим функции по выработке государственной политики и нормативно-правовому регулированию в сфере труда.

Продолжительность ежедневной работы (смены) не может превышать:

- для работников от 15 до 16 лет – 5 часов, от 16 до 18 лет – 7 часов;
- для учащихся общеобразовательных учреждений, образовательных учреждений начального и среднего профессионального образования, совмещающих в течение учебного года учебу с работой, от 14 до 16 лет – 2,5 часа, от 16 до 18 лет – 4 часов;
- для инвалидов – в соответствии с медицинским заключением, выданным в порядке, установленном федеральными законами и иными нормативными правовыми актами российской федерации.

Для работников, занятых на работах с вредными и (или) опасными условиями труда, где установлена сокращенная продолжительность рабочего

времени, максимально допустимая продолжительность ежедневной работы (смены) не может превышать:

- при 36-часовой рабочей неделе - 8 часов;
- при 30-часовой рабочей неделе и менее - 6 часов.

Продолжительность работы (смены) в ночное время сокращается на один час без последующей отработки. К работе в ночное время не допускаются: беременные женщины; работники, не достигшие возраста 18 лет, за исключением лиц, участвующих в создании и (или) исполнении художественных произведений, и других категорий работников в соответствии с настоящим Кодексом и иными федеральными законами.

В течение рабочего дня (смены) работнику должен быть предоставлен перерыв для питания и отдыха. Это время и его продолжительность должно устанавливаться правилами внутреннего трудового распорядка или по соглашению между работником и работодателем.

Работникам должны предоставляться выходные дни (еженедельный непрерывный отдых).

Организация-работодатель должна выплачивать заработную плату работникам. Удержание заработной платы возможно только в случаях, установленных ТК РФ ст. 137. При задержке заработной платы более чем на 15 дней, может приостановить работу, письменно уведомив работодателя.

Законодательством РФ запрещена дискриминация по любым признакам и принудительный труд [20].

Основными элементами рабочего места программиста являются: рабочий стол, рабочий стул (кресло), дисплей, клавиатура, мышь; вспомогательными - пюпитр, подставка для ног.

Взаимное расположение элементов рабочего места должно обеспечивать возможность осуществления всех необходимых движений и перемещений для эксплуатации и технического обслуживания оборудования [25].

На рабочем месте, оснащённом ЭВМ, должно быть пространство для пользователя не менее 850 м. Для стоп должно быть предусмотрено

пространство по глубине и высоте не меньше 150 мм, по ширине – не менее 530 мм. Расположение ЭВМ на рабочем месте должно быть таким, чтобы поверхность экрана находилась на расстоянии 400 – 700 мм от глаз пользователя. Конструкция рабочего места и взаимное расположение всех его элементов (сиденье, органы управления, средства отображения информации и т.д.) должны соответствовать антропометрическим, физиологическим и психологическим требованиям, а также характеру работы [25].

Рабочее кресло обеспечивает поддержание рабочей позы в положении сидя, и чем длительнее это положение в течение рабочего дня, тем жестче должны быть требования к созданию удобных и правильных рабочих сидений.

Высота поверхности сиденья должна регулироваться в пределах 400 – 550 мм. Ширина и глубина его поверхности должна быть не менее 400 мм. Поверхность сиденья должна быть плоской, передний край – закругленным. Сиденье и спинка кресла должны быть полумягкими, с несскользящим, не электризующимся и воздухопроницаемым покрытием, материал которого обеспечивает возможность легкой очистки от загрязнения.

Опорная поверхность спинки стула должна иметь высоту 280 – 320 мм, ширину – не менее 380 мм и радиус кривизны горизонтальной плоскости – 400 мм. Расстояние сцинки от переднего края сиденья должно регулироваться в пределах 260 – 400 мм.

Рациональной рабочей позой может считаться такое расположение тела, при котором ступни работника расположены на плоскости пола или на подставке для ног, бедра сориентированы в горизонтальной плоскости, верхние части рук – вертикальный угол локтевого сустава колеблется в пределах 70 – 90, запястья согнуты под углом не более чем 20, наклон головы – в пределах 15 – 20, а также исключены частые ее повороты [25].

### **Влияние разработанных тесовых наборов на работу компании**

Основным направлением реализации разработанных тесовых наборов является выявление факта, соответствует ли разрабатываемый программный продукт требованиям, в которых зафиксирован первоначальный замысел

изделия (т.е. то, что заказал заказчик). Тестирование является обязательной частью процесса производства ПО.

Результатом разработанных тестов является повышение качества продукта по всем его характеристикам. Автоматизация связана с управлением, потому что это может сэкономить время, сократить число машин, ресурсов и тестировщиков программного обеспечения, которые используются компанией по тестированию программного обеспечения в целях достижения высокого качества продукта. Для многих автоматизированных проектов трудно определить, чего можно от них ожидать, так как те, кто выполняет автоматизацию, не понимают, что автоматизация является только еще одним инструментом, который может быть использован для тестирования программного обеспечения. Это не дает каких-либо гарантий и не защищает продукт от плохой производительности, в то время как тестирование программного обеспечения влияет на улучшение качества выпускаемой продукции. Недостаточное применение автоматизации в конечном счете приведет к затрате большего количества времени и усилий, чем полное отсутствие автоматизации.

## ЗАКЛЮЧЕНИЕ

В ходе проделанной работы было проведено внедрение технологии автоматизированного тестирования и применено на реальных продуктах в компании ООО «ТомскАСУпроект». Были проанализированы существующие методы и средства для автоматизированного тестирования web- и desktop-приложений, вследствие чего были выбраны и изучены такие программные средства, как TestComplete и Selenium IDE, а так же трекер-система для управления проектами и задачами Redmine (PM).

Для проведения автоматизированного тестирования в данных программах были изучены языки C# Script, язык команд Selenium, язык гипертекстовой разметки HTML, язык запросов xpath, CSS.

Были созданы тестовые наборы, покрывающие 50% функций desktop-приложения и имеющие 63 теста и 100% функций web-приложения, имеющие 248 тестов.

При тестировании использовались такие методы, как:

- Дымовое тестирование
- Тестирование на основе тест-кейсов
- Тестирование методом черного ящика
- Системное тестирование
- Функциональное тестирование
- Расширенное тестирование
- Негативное и позитивное тестирование
- Альфа тестирование
- Регрессионное тестирование

В итоге проведенного тестирования было найдено 23 ошибки в модуле программной системы «CAD-GIS» «Автоматизированная система для создания ведомости подсчета объемов работ по автомобильным дорогам и генеральным планам» и 66 ошибок в web-приложении «Корпоративная геоинформационная система» ООО «РН – Пурнефтегаз.

Проведенная работа имеет большую значимость для компании, так как в данной компании практикуется ручное тестирование, а благодаря использованию технологии автоматизированного тестирования, на реальном продукте показана эффективность этого метода в реальности. Тестируемые приложения планируют модернизировать и выпускать следующие их версии в будущем, а web-приложение так же будет использоваться в нескольких компаниях, имея в основном только различный дизайн и наполнение данными. Поэтому созданные тестовые наборы необходимы будут и в дальнейшем для проведения регрессионного тестирования

В ходе исследования выгода компании от автоматизации тестирования составила для модуля «Ведомости» -113 838 рублей, для модуля «КГИС» + 2 282 880 рублей. Был сделан вывод, что для маленькой компании и, особенно, для маленьких проектов автоматизированное тестирование дорогими программными средствами не выгодно. Но при большом объеме задач и имея простое средство автоматизации, данный вид тестирования весьма эффективен.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Технический дизайн «Корпоративная геоинформационная система» ООО «РН – Пурнефтегаз» - 2014. – 31с.
2. Бейзер Б. Тестирование чёрного ящика. Технологии функционального тестирования программного обеспечения и систем. — СПб.: Питер, 2004. — 320 с.
3. Канер Кем, Фолк Джек, Нгуен Енг Кек. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений. — Киев: ДиаСофт, 2001. — 544 с.
4. Калбертсон Роберт, Браун Крис, Кобб Гэри. Быстрое тестирование. — М.: «Вильямс», 2002. — 374 с.
5. Куликов С. Тестирование программного обеспечения. Базовый курс. – © EPAM Systems, 2015-2017 – 289 с.
6. Сайт об автоматизации тестирования [электронный ресурс]. URL: <http://perfect-test.com> (дата обращения 14.06.2016).
7. I.B. Bourdonov, A.S. Kossatchev, V.V. Kuliamin, A.K. Petrenko. UniTesK Test Suite Architecture. Proc. of FME 2002. LNCS 2391, pp. 77–88, Springer–Verlag, 2002.
8. В.В. Кулямин, А.К. Петренко, А.С. Косачев, И.Б. Бурдонов. Подход UniTesK к разработке тестов. Программирование, 29(6):25–43, 2003.
9. Русскоязычный сайт, посвященный Selenium [электронный ресурс]. URL: <http://selenium2.ru> (дата обращения 14.06.2016).
10. Оценка эффективности автоматизации тестирования [Электронный ресурс]. Технологии качества. URL: <http://alqa.ru/blog/otsenka-effektivnosti-avtomatizatsii-testirovaniya/> (дата обращения 20.05.2017).
11. Элфрид Дастин, Джефф Рэшка, Джон Пол. Автоматизированное тестирование программного обеспечения. Внедрение, управление и эксплуатация. М.: «Лори», 2003.

12. Гленфорд Майерс. Надежность программного обеспечения. – М.: «Мир», 1980. – 360 с.
13. Гленфорд Майерс. Искусство тестирования программ. – М.: «Финансы и статистика», 1982. – 272 с.
14. И. Винниченко. Автоматизация процессов тестирования. – СПб.: «Питер», 2005. – 449 с.
15. Кент Бек. Экстремальное программирование: разработка через тестирование. – СПб.: «Питер», 2003. – 224 с.
16. Луиза Тамре. Введение в тестирование программного обеспечения. – М.: «Вильямс», 2003. – 368 с.
17. Гребенюк В. М. Оценка целесообразности внедрения автоматизированного тестирования. Институт Государственного управления, права и инновационных технологий (ИГУПИТ). Интернет-журнал «НАУКОВЕДЕНИЕ» №1 2013.
18. Чрезвычайные ситуации при работе с ПЭВМ // Студопедия — Ваша школопедия. URL: [http://studopedia.ru/8\\_107307\\_osveshchenie-pomeshcheniy-vichislitelnih-tsentrov.html](http://studopedia.ru/8_107307_osveshchenie-pomeshcheniy-vichislitelnih-tsentrov.html) (дата обращения: 10.03.2017).
19. Долин П.А. Справочник по технике безопасности. М.: Энергоатомиздат, 1984 г. – 824 с.
20. Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 3.07.2016) // Электронный фонд правовой и нормативно-технической документации. URL: <http://docs.cntd.ru/document/901807664> (дата обращения: 11.03.2017).
21. СанПиН 2.2.4.548-96. Гигиенические требования к микроклимату производственных помещений.
22. Программа и методика испытаний «Корпоративная Геоинформационная система» «РН – Пурнефтегаз». – 2016. – 73с.
23. Программа и методика испытаний «Автоматизированная система для создания ведомости подсчета объемов работ по автомобильным дорогам и генеральным планам». – 2016. – 80 с.

24. Охрана труда. Основы безопасности жизнедеятельности // [www.Grandars.ru](http://www.Grandars.ru). URL: <http://www.grandars.ru/shkola/bezopasnost-zhiznedeyatelnosti/ohrana-truda.html> (дата обращения: 11.03.2017).

25. Попов В.М. Психология безопасности профессиональной деятельности: учебное пособие / В. М. Попов; Новосибирский государственный технический университет. – Новосибирск: Изд-во Новосибирского государственного технического университета, 1996 г. – 155 с.

26. СанПиН 2.2.2/2.4.1340-03. Санитарно-эпидемиологические правила и нормы. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы // Электронный фонд правовой и нормативно-технической документации. URL: <http://docs.cntd.ru/document/901865498> (дата обращения: 10.03.2017).

27. ГОСТ Р 12.1.019-2009 ССБТ. Электробезопасность. Общие требования и номенклатура видов защиты // Электронный фонд правовой и нормативно-технической документации. URL: <http://docs.cntd.ru/document/1200080203> (дата обращения: 11.03.2017).

## ПРИЛОЖЕНИЕ А

### Part 1

#### Description of the subject field

Студент:

Группа	ФИО	Подпись	Дата
8ИМ5А	Борисова Анастасия Владимировна		

Консультант кафедры Информационные системы и технологии:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Мирошниченко Евгений Александрович	К.т.н.		

Консультант – лингвист кафедры ИЯ:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель каф.ИЯ	Горбатова Татьяна Николаевна			

## **1. Description of the subject field**

Testing is process of detecting errors in program software by executing its output code based on the test data, gathering working characteristics in the performance dynamics in specific operating environment, detection of different errors, defects, failures and bugs, caused by irregular and abnormal situations or software crash. Important part of verification and testing process is organization, such as activity of specialists group, who perform the planning of those processes, preparation test data and monitoring of testing [3].

### **Classification of testing methods**

Testing process can be classified by a large number of features. This work includes and describes all classification methods of testing. Such work allows us to systemize, to widen knowledge and significantly accelerate planning process of testing and test-cases development along with optimization of labor contribution.

### **Code execution start classification**

- Static testing – without starting code execution. This test can be used for:
  - Documents (requirements, test cases, etc.)
  - Graphic prototypes (user's interface design sketches).
  - Application code
  - Parameters (settings) application run-time environment.
  - Prepared test data.
- Dynamic testing – with start of code execution. The code execution can be run for the whole application code, for several interrelated parts (integration testing) or for separate parts (module or unit testing) and even for separate parts of code. The main idea of that type of testing is to check actual behavior (part) of application [12].

### **Code access and architecture of application classification**

The white-box technique – tester has access to internal structure and to the code of application, he also has enough knowledge to understand what he sees.

- The black-box technique – tester either does not have access to internal structure and to the code of application, or he does not have knowledge to understand

it, or he intentionally ignores it during testing process. However, absolute majority of testing types work according to the black-box technique. Within the framework of black-box technique method, the main information, which is used to create test-cases, is documentation (requirements in particular) and common sense (in cases, where application behavior is not explicitly regulated for some situations; sometimes it is called “testing based on implicit requirements”).

- Gray-box technique – it is a combination of white-box and black-box techniques, what means that tester has access to some part of code and architecture and to some part does not [2].

#### **Automatization degree classification**

- Manual testing - test cases are performed manually by a person without using means of automatization.

- Automated testing is a set of techniques, approaches and device tools, which allow to exclude a person from a number of tasks execution during testing process. Test-cases development, data preparation, performance results evaluation, writing reports about detected faults – all this and much more is still done by a person.

#### **Level of application detalization classification (by level of testing)**

- Module (unit) testing is aimed at testing small individual parts of application, which (in general) can be tested separately from other similar parts. During this type of testing you can test separate functions or techniques of classes, classes themselves, classes interaction small libraries, separate parts of application. Very often this type of testing is executed with the usage of special technologies and tools of automatization of testing process, what significantly simplifies and accelerates related test-cases development.

- Integration testing is aimed at testing of interaction of several parts of application (each one of those, in its turn, was checked separately during module stage testing).

- System testing is aimed at the test of the entire application as a whole, assembled from different parts, which were checked during two previous stages. This stage detects not only “conjunction” units faults, but also gives possibility to fully

interact with application from the point of view of the end user, applying many other listed in this chapter types of testing [3].

### **Classification by the importance degree (descending) of the tested functions (by the level of functional testing)**

- Smoke test is aimed at the test of the main, the most important and the key functionality itself, nonoperability of which makes the whole idea of application usage (or any other object, subjected to the smoke test) meaningless. Smoke test is carried out after the release of new build to define general level of application and to take a decision on (non) reasonability of critical path testing execution and advanced testing.

- Critical path testing is aimed at the analysis of functionality used by typical user in typical daily activity. The idea itself was borrowed from the project management and transformed in the context of testing into the following: there is most part of users, who more often use subset of application functions.

- Advanced testing is aimed at the examination of all functionality listed in requirements, including the one that is low ranked by the importance degree. At the same time, it is noted here which function is more important and which one is less important. Although, if there is enough time and other resources test-cases of that level can reach even the most low-priority requirements [12].

### **Classification by the working principals with application**

- Positive testing is aimed at the examination of application in situation when all the actions are performed strictly by the instruction, without any mistakes, deviations or incorrect data input, etc.

- Negative testing is aimed at the examination of application functioning in situations when incorrect operations are executed and/or data, potentially leading to mistakes (divide by zero), is used with it [11].

### **Application nature classification**

- Web-applications testing involves intensive activity in the field of substitutable testing (in particular cross-browser testing), productivity testing, automatization testing with the usage of large number of tools.

- Mobile applications testing
- Desktop applications testing

### **Classification by the focusing on the architectural level of application**

- Testing of level of the presentation is focused on the part of application which is responsible for interaction with “outside world” (both by the users and other applications). In this part, we examine the questions of ease of use, speed of interface response, compatibility with browsers and accuracy of interface functioning.

- Testing of the domain logic level is responsible for the test of main set of application functions and is built on the base of key requirements to the application, business rules and general check of functionality.

- Testing of the data level is concentrated on the part of application, that is responsible for storing and partial data processing (most often in database or in another storage). In this case, the main interest is data testing, compliance of business-rules check and productivity testing [14].

### **Classification by end-users involvement (operational testing)**

- Alpha-testing is performed inside the development organization with possible partial involvement of end users. It can be a form of internal acceptance testing process. Shortly speaking, the idea of this type is that the product can be periodically shown to outside users, but it is still quite “raw” and the main testing is executed by the development organization.

- Beta-testing is performed outside of the development organization with active involvement of end users. It can serve as a form of outside acceptance testing. Briefly, the essence of that type is that the product can be openly shown to outside users and it is quite stable, but problems still can occur and to detect those feedbacks from real users are necessary.

- Gamma-testing is a final stage of testing before product release, which is aimed at fixing minor defects, detected during beta-testing. Usually, this stage is performed with maximum involvement of end users/customers. It can serve as a form of external acceptance testing. Shortly speaking, the essence of that type is that the

product is almost ready and the feedbacks from real users are being used to fix the latest faults [4].

### **Classification by formalization degree**

- Testing on the base of test-cases is a formal approach, where testing is performed on the base of previously prepared test-cases, sets of test-cases and other documentation.

- Exploratory testing is a partially formal approach, within that type tester can work with application according to the selected scenario, which in its turn is being improved in the implementation process in order to achieve more detailed examination of application. The key factor of success of exploratory testing is exact work according to scenario not execution of random, thoughtless operations. There is even special scenario approach, called sessional testing. As an alternative to scenario, while selecting actions with application, sometimes can be used check-lists then this type of testing is called testing on the base of check-lists.

- Ad hoc testing is completely non-formalized approach, where there is no intention to use any test-cases, check-lists or scenarios, tester fully relies on his professionalism and intuition for spontaneous execution of actions with application, which he thinks can detect an error.

### **Classification by aims and tasks**

- Positive testing (reviewed above)
- Negative testing (reviewed above)
- Functional testing is aimed at examination of accuracy of application functional performance (accurate realization of functional requirements).

- Non-functional testing is aimed at examination of non-functional features of application (realization accuracy of non-functional requirements), such as usability, productivity, security etc.

- Installation testing is aimed at detecting defects that can influence installation stage (setup) of application. In general, such type of testing checks multiple scenarios and aspects of installer's work in such cases as:

- new environment of performance

- updating of existing version;
- changing current version to the older one;
- repeated installation of application;
- repeated installation launch;
- uninstalling application;
- installing new application from the application family;
- automatic installation without user's participation.
- Regression testing is a test, aimed at the examination of the fact that there are no errors in previously working functionality caused by the changes in application or in its functional environment.
  - Re-testing is test-cases execution, which previously detected bugs in order to confirm defects elimination.
  - Acceptance testing is a formal testing aimed at examination of application from the point of view of end-user/customer and taking a decision whether a client accepts work from the executor (project team).
  - Operational testing is a test conducted in real or close to real operational environment, including operating system, database systems, application servers, web-servers, hardware etc.
  - Usability testing is aimed at examining how far end-user understands how to work with application and how much he likes to use the product.
  - Accessibility testing is aimed at examination of suitability of product for usage by people with disabilities (weak eyes, etc.)
  - Interface testing is aimed at application interface examination or its components. According to ISTQB definition, this type of testing belongs to integration testing.
  - Security testing is aimed at examination of application ability to resist malicious attempts to gain access to the data or functions without having access right.
  - Internationalization testing is aimed at examination of readiness of the product to function with the usage of different languages and with taking into account different national and cultural differences.

- Localization testing is aimed at examination of accuracy and adaptation quality of the product to the usage in one or another language taking into account national and cultural peculiarities. It checks accuracy of translation and adaptation of product, not readiness of the product to such actions.
- Compatibility testing is aimed at examination of application ability to function in specified environment. For example: here can be checked:
  - Compatibility with hardware platform, operational system and network infrastructure.
  - Compatibility with browsers and their versions
  - Compatibility with mobile devices.
  - Etc.
- Data and database testing, two similar types of testing, aimed at the study of data characteristics such as completeness, consistency, integrity, structuring, etc. In databases context, we can examine adequacy of the problem space, capacity of the model to ensure integrity and consistency of data, accuracy of triggers functioning, stored procedures, etc.
- Resource utilization testing is a set of testing types checking effectiveness of usage by application available to its resources and dependence of results of application on quantity of available resources.
- Comparison testing is aimed at comparative analysis of advantages and disadvantages of the product being developed in relation to its main competitors.
- Demonstration testing is a formal demonstration to the product customer in order to confirm that product meets all the listed requirements. Unlike the acceptance testing, this process is more exact and comprehensive, but also can be carried out at intermediate stages of product development.
- Redundancy/over testing is testing of application with all possible combinations of all possible input data in all possible execution conditions. It is impossible to use for any level of complexity of the system, but can be used to test separate and very simple components [16].

- Reliability testing means testing the ability of application to perform its functions in specified conditions for specified period of time or specified number of operations.

- Restorability testing is a testing of application capacity to restore its functions and specified level of performance and also to restore data in case of critical situation, which can lead to temporary (partial) loss of working capacity of the application.

- Fault tolerance testing is a testing process that consists of emulating or actual creating critical situations in order to check application ability to use appropriate mechanisms to prevent disruption of working process, performance and data damage.

- Performance testing is an examination of response speed rate to the external influences of different characteristics and load intensity. Performance testing has several subtypes:

- Load testing is the study of the ability of application to maintain target quality level with a load within acceptable limits and with a certain excess of those limits (definition “margin of safety”).

- Scalability testing is the study of the ability of application to increase performance level in accordance with increase of number of available resources to the application.

- Volume testing means the study of application performance while processing different (usually, big) volumes of data.

- Stress testing is the study of application behavior in case of abnormal load changes significantly exceeding calculated level or in situations, where most part of necessary resources for the application is unavailable.

- Competitive testing is the study of application behavior in the situation where it has to process large number of simultaneous requests, what causes competition among requests for the resources. Sometimes, competitive testing is also understood as the study of multithreaded applications and accuracy of action synchronization in different threads.

## Techniques and approaches classification

- Positive testing (reviewed above).
- Negative testing (reviewed above).
- Based on the tester's experience, scenarios, check-lists:
  - Exploratory testing (reviewed above).
  - Free (intuitive) testing (reviewed above).
- By the degree of interference in application performance:
  - Intrusive testing is a testing, execution of which can influence application functioning due to the testing tools operation or due to the power of interference in the code of application itself.
    - Non-intrusive testing is a testing, performance of which is invisible for the application and does not influence its usual performance process.
- By the automatization techniques:
  - Testing under data control is a means of automatic test-cases development, where input data and expected results are placed outside of test-case and stored outside of it – in a file, database etc.
  - Testing under keywords management is a way of development of automatic test-cases, where not only set of input data and expected results are placed outside of test-case, but also behavioral logic of test-case, which is described with the help of keywords (commands).
  - Testing under behavior control is a means of automatic test-cases development, where the main attention is paid to the accuracy of business-scenarios performance, but not to separate details of functionality of application.
- On the base of (knowledge) errors sources:
  - Error guessing testing is tests developed on the tester's experience and his knowledge about what defects are typical for these or those components or areas of application functionality.
  - Heuristic evaluation is a technique of usability testing, aimed at locating problems in user's interface, which is a deviation from generally accepted rules.

- Testing with adding errors – to the application intentionally are added previously known and specially designed errors in order to monitor the process of errors detecting and eliminating, thus, creating more specific estimation of testing process.

- Based on the choice of input data:

- Testing based on the equivalence of classes is aimed at reducing number of test-cases being developed and executed at the same time maintaining enough test coverage (identifying sets of equivalent test-cases).

- Testing based on the boundary conditions; it is based on the equivalence of classes allowing to identify specific features of test parameters, related to the boundaries of classes of equivalence.

- Domain testing is based on the equivalence and boundary classes, what allows effectively create test-cases involving several parameters at once.

- Pair testing implies test-cases are created based on the principal of testing pairs of parameters values, instead of attempts to check all possible combinations of all values of all parameters.

- Testing on the base of orthogonal arrays is tool technique of pair and N-combination testing based on the use of, so called, orthogonal arrays (two-dimensional arrays with following property: if to take two random columns of such an array, then resulting “subarray” will contain all possible pair combinations of values, presented in the original array).

- Based on the execution environment:

- Testing in the development process is a testing being directly performed during development process of application and/or in the execution environment, which is different from the actual environment of application usage.

- Operational testing (reviewed above).

- Testing based on the code (on the base of the structures).

- Testing by the control flaw is a family of techniques, where test-cases are designed to activate and to test execution of different sequences of events that are determined by the analysis of the source code of application.

- Testing by the data flow is a family of techniques based on the selection of separate threads of control flow in order to study events, related to the changing of state of variables.
- Testing by diagram or state table means test-cases are developed to check applications transitions from one state to another.
- Code audit is a family of techniques for enhancing code quality due to the fact that several people participate in the process of development or sophistication of the code.
  - Testing on the base of the code structure implies possibility to examine logic of code execution in different situations and includes:
    - Expression testing examines accuracy (and the fact itself) of separate expressions execution in the code.
    - Testing based of the paths checks execution of individual paths of the code.
    - Conditions-based testing examines execution of individual conditions.
    - Testing based on the combinations of conditions checks execution of complex (compound) conditions.
    - Testing based on the individual conditions, that generate braiding (“decisive conditions”) examines execution of such conditions as a part of complex conditions, which alone determine the result of calculations of the whole complex condition.
    - Testing based on the solutions checks performance of complex paths (with two or more possible options). Despite of the fact that “two options” both fit, formally such situation should be attributed to testing based on conditions.
    - Testing based on the paths examines execution of all or several specially selected paths in the code of application.
    - Testing based on the (model) behavior of application:
      - Testing based on the decision table means that test-cases are developed on the basis of so called decision table, which reflects the input data (and its

combinations) and influences on the application, as well as relevant output data and application reactions.

- Testing by diagram or state table (reviewed above).
- Testing by specifications (reviewed above).
- Testing by the behavioral models means study of application (and test-cases development) is built on some model: decision table, table or state diagram, user scripts, load models, etc.
- Testing based on the use cases means that test-cases are developed on the base of use cases.
- Parallel testing implies comparison of behavior of new (or modified) application with behavior of reference application.
- Random data testing implies that input data, actions or even test-cases themselves are selected on the basis of (pseudo) random values in order to correspond to the operational profile – a subset of actions, corresponding to a certain situation or scenario of working with application.
- A/B testing is a testing technique that studies the impact of changing of one of the input parameters on the result of operation execution [5].

## ПРИЛОЖЕНИЕ Б

Таблица Б.1. Примеры тест-кейсов по функциональности, доступной пользователю модуля «Автоматизированная система для создания ведомости подсчета объемов работ по автомобильным дорогам и генеральным планам».

№	Название	Предварительные шаги	Шаги	Ожидаемый результат	Фактический результат
<b>1</b>	<b>Создание проекта</b>				
1.1	Создание нового проекта с пустыми полями	1. Запустить программу AutoCAD Civil 3D. 2. Перейти на вкладку Полотно.Ведомости	1. Нажать «Создать проект» 2. Оставить все поля пустыми 3. Нажать «Заккрыть»	1. После шага 1 откроется окно «Создать проект» 2. После шага 2 Кнопка «Создать» - неактивна, Кнопка «Заккрыть» - активна (создание проекта невозможно).	+
1.2	Создание нового проекта	1. Запустить программу AutoCAD Civil 3D 2. Перейти на вкладку Полотно.Ведомости	1. Нажать «Создать проект» 2. Заполнить обязательные поля: 2.1. Название 2.2. Заказчик 2.3. Дата создания Остальные поля заполнить по желанию 3. Нажать «Создать»	1. После шага 1 откроется окно «Создать проект» 2. После шага 2 в заполненных полях появятся данные. Кнопка «Создать» стала активна. 3. После шага 3 окно «Создать проект» закроется, система создаст новый проект.	+
1.3	Отмена экспорта ведомостей путем отмены	1. Запустить программу AutoCAD Civil 3D 2. Перейти на вкладку Полотно.Ведомости 3. Создать проект.	1. Нажать «Экспорт в Word» 2. Нажать на папку, в которую ранее экспортировались ведомости 3. Нажать «ОК»	1. После шага 1 откроется окно «Обзор папок» 2. После шага 2 выбранная папка выделится синим 3. После шага 3 появится	Ошибка: система производит частичный экспорт

	индикатора загрузки	<p>Выполнить тест-кейс 1.2 или 1.3</p> <p>4. Открыть окно «Ведомости». Выполнить тест-кейс 3.1.2</p> <p>5. Создать сооружение. Выполнить тест-кейс 3.2.1</p> <p>6. Создать записи в ведомостях В1-В5. Выполнить тест-кейсы 3.3.1-3.3.8</p>	<p>4. Нажать крестик на индикаторе выполнения</p> <p>5. Нажать «ОК»</p>	<p>индикатор выполнения</p> <p>4. После шага 4 появится окно «Операция отменена»</p> <p>5. После шага 5 станет активным окно «Ведомости». Система не произведёт экспорт ведомостей.</p>	
1.4	Создание проекта с некорректной датой	<p>1. Запустить программу AutoCAD</p> <p>2. Перейти на вкладку Полотно.Ведомости</p>	<p>1. Нажать «Создать проект»</p> <p>2. Заполнить обязательные поля:</p> <p>2.1. Название</p> <p>2.2. Заказчик</p> <p>3. Заполнить поле «Дата создания» буквами</p> <p>4. Нажать «Создать»</p>	<p>1. После шага 1 откроется окно «Создать проект».</p> <p>2. После шагов 2 и 3 в заполненных полях появятся данные.</p> <p>4. После шага 4 система создаст новый проект с текущей датой.</p>	+

## ПРИЛОЖЕНИЕ В

Таблица А.1 – Примеры тест-кейсов по функциональности, доступной пользователю web-приложения «Корпоративная геоинформационная система» ООО «РН – Пурнефтегаз»

№	НАИМЕНОВАНИЕ МЕТОДИКИ/ИСПЫТАНИЯ	ШАГИ	ОЖИДАЕМЫЙ РЕЗУЛЬТАТ	АВТОМАТИЗАЦИЯ	ЗАМЕЧАНИЯ
<b>Работа с картой КГИС</b>					
1	Перемещение карты.	<ol style="list-style-type: none"> <li>1. Открыть интернет браузер Internet Explorer версии 9.0 для Windows 7.</li> <li>2. В поле адреса указать адрес модуля КГИС: <a href="http://hydra.tomskasu.ru/cgis#">http://hydra.tomskasu.ru/cgis#</a> .</li> <li>3. Нажать в любом месте карты левой кнопкой мыши.</li> <li>4. Зажав кнопку перетащить карту.</li> </ol>	Карта переместилась в указанном направлении.	+	Успешно
2	Приближение/отдаление на карте.	<ol style="list-style-type: none"> <li>1. Открыть интернет браузер Internet Explorer версии 9.0 для Windows 7.</li> <li>2. В поле адреса указать адрес модуля КГИС: <a href="http://hydra.tomskasu.ru/cgis#">http://hydra.tomskasu.ru/cgis#</a> .</li> </ol> <p><i>1 способ</i></p> <ol style="list-style-type: none"> <li>3.левой кнопкой мыши зажать ползунок масштаба, расположенный в правом нижнем углу окна.</li> <li>4. Переместить ползунок на нужную позицию.</li> </ol> <p><i>2 способ</i></p> <ol style="list-style-type: none"> <li>5. Нажать на выпадающий список масштаба, расположенный в правом нижнем углу окна.</li> <li>6. Выбрать подходящий масштаб и нажать на него.</li> </ol> <p><i>3 способ</i></p> <ol style="list-style-type: none"> <li>7. Покрутить колесико мыши.</li> </ol>	Произошло приближение (отдаление) на карте.	+	Успешно

1.	Создание объекта собственности.	<ol style="list-style-type: none"> <li>1. Открыть интернет браузер Internet Explorer версии 9.0 для Windows 7.</li> <li>2. В поле адреса указать адрес модуля КГИС.</li> <li>3. Нажать в любом месте карты правой кнопкой мыши. В выпадающем списке выбрать пункт «Создание объекта собственности».</li> </ol>	Открылось новая вкладка АРМ «Объекты собственности», в котором отображаются атрибуты выбранного в КГИС объекта.	+	Успешно
4	Добавление пункта маршрута.	<ol style="list-style-type: none"> <li>1. Открыть интернет браузер Internet Explorer версии 9.0 для Windows 7.</li> <li>2. В поле адреса указать адрес модуля КГИС: <a href="http://hydra.tomskasu.ru/cgis#">http://hydra.tomskasu.ru/cgis#</a> .</li> <li>3. Нажать в любом месте карты правой кнопкой мыши.</li> <li>1. В выпадающем списке выбрать пункт «Добавить пункт маршрута».</li> </ol>	Открылось окно поиска маршрута, выбранный объект добавлен в инструмент поиска маршрута.	+	Успешно
5	Измерение координат точки.	<ol style="list-style-type: none"> <li>1. Открыть интернет браузер Internet Explorer версии 9.0 для Windows 7.</li> <li>2. В поле адреса указать адрес модуля КГИС: <a href="http://hydra.tomskasu.ru/cgis#">http://hydra.tomskasu.ru/cgis#</a> .</li> <li>3. Нажать в любом месте карты правой кнопкой мыши.</li> <li>4. В выпадающем списке выбрать пункт «Координаты точки».</li> </ol>	Открылось окно измерения, в котором отобразились координаты выбранной точки.	+	Успешно
6	Создать порыв.	<ol style="list-style-type: none"> <li>1. Открыть интернет браузер Internet Explorer версии 9.0 для Windows 7.</li> <li>2. В поле адреса указать адрес модуля КГИС: <a href="http://hydra.tomskasu.ru/cgis#">http://hydra.tomskasu.ru/cgis#</a> .</li> <li>3. Нажать в любом месте карты правой кнопкой мыши.</li> <li>4. В выпадающем списке выбрать пункт «Создать порыв».</li> <li>5. В открывшемся окне заполнить обязательные поля.</li> <li>6. Нажать кнопку «Сохранить».</li> </ol>	В модуле АРМ УНУ создан порыв.	+	Успешно