

**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 Информационных систем и технологий

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

Тема работы
Разработка технологии автоматизированного тестирования программного обеспечения семейства информационных систем

УДК 004.415.53

Студент

Группа	ФИО	Подпись	Дата
8ИМ5А	Федорченко Алёна Вадимовна		

Руководитель

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

**КОНСУЛЬТАНТЫ:**

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент кафедры менеджмента	Попова Светлана Николаевна	к.э.н., доцент		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент кафедры ЭБЖ	Акулов Петр Анатольевич			

**ДОПУСТИТЬ К ЗАЩИТЕ:**

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
ИСТ	Мальчуков Андрей Николаевич	К.т.н., доцент		

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 (специальность) 09.04.02 Информационные системы и технологии  
Кафедра Информационных систем и технологий

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	Федорченко Алёна Вадимовна

Тема работы:

Разработка технологии автоматизированного тестирования программного обеспечения семейства информационных систем
---

Утверждена приказом директора (дата, номер)	№ 897/с от 20.02.2017
---	-----------------------

Срок сдачи студентом выполненной работы:	
--	--

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ:**

<p><b>Исходные данные к работе</b></p> <p><i>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).</i></p>	<p>Задача разработки автоматизированных тестовых скриптов для 50% функционала настольного модуля программной системы «CAD-GIS» «Автоматизированная система для создания ведомости подсчета объемов работ по автомобильным дорогам и генеральным планам» и 100% функционала настольного приложения «Модуль формирования спецификаций «Specbook».</p>
---	---

<p><b>Перечень подлежащих исследованию, проектированию и разработке вопросов</b></p> <p><i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p>	<p>Поставлены задачи проведения сравнительного анализа программных средств, предназначенных для автоматизации процесса тестирования настольных приложений, задача разработки автоматических тестов для модуля «Модуль формирования спецификаций «Specbook» с помощью скриптового языка AutoIT, разработка автоматической рассылки результатов тестирования, расчет оценки эффективности внедрения автоматического тестирования для программного модуля.</p>
<p><b>Перечень графического материала</b></p> <p><i>(с точным указанием обязательных чертежей)</i></p>	<p>Сравнительная таблица инструментов для тестирования          Пример тестовых сценариев          Пример записи результатов тестирования в лог-файл          Сервер непрерывной интеграции</p>
<p><b>Консультанты по разделам выпускной квалификационной работы</b></p> <p><i>(с указанием разделов)</i></p>	
<p><b>Раздел</b></p>	<p><b>Консультант</b></p>
<p>Финансовый менеджмент</p>	<p>Попова С.Н.</p>
<p>Социальная ответственность</p>	<p>Акулов П. А.</p>
<p>по иностранному языку</p>	<p>Мирошниченко Е.А.</p>
<p><b>Названия разделов, которые должны быть написаны на русском и иностранном языках:</b></p>	
<p>3.2. Разработка автотестов</p>	

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

**Задание выдал руководитель:**

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

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8ИМ5А	Федорченко Алёна Вадимовна		

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

---

Институт кибернетики  
Направление подготовки (специальность) Информационные системы и технологии  
Уровень образования магистратура  
Кафедра Информационных систем и технологий  
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)
30.03.2017	Глава «Анализ предметной области»	10
25.04.2017	Глава «Расчеты и аналитика»:	30
02.06.2017	Глава «Результаты выполнения разработки»	30
08.06.2017	Глава «Финансовый менеджмент»	10
09.06.2017	Глава «Социальная ответственность»	10
09.06.2017	Перевод раздела диплома на английский язык	10

Составил преподаватель:

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

**СОГЛАСОВАНО:**

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
ИСТ	Мальчуков Андрей Николаевич	К.т.н., доцент		

## РЕФЕРАТ

Выпускная квалификационная работа содержит 127 с., 36 рис., 35 табл., 25 источников, 16 формул, 2 приложения.

Ключевые слова: технологии автоматизации тестирования, AutoIT, методологии тестирования, тестовые сценарии, Specbook, TestComplete, Полотно.Ведомости.

Объектом исследования являются методологии и программные средства, предназначенные для автоматизации тестирования настольных приложений.

Целью выпускной квалификационной работы является разработка и внедрение технологии автоматизированного тестирования программного обеспечения.

В рамках работы решались следующие задачи: анализ программных средств, предназначенных для тестирования настольных приложений, разработка и внедрение технологии автоматизированного тестирования программного обеспечения, включающей в себя разработку автоматических тестов с функцией формирования и рассылки отчётов о результатах тестирования для настольного приложения «Модуль формирования спецификаций «Specbook» с помощью скриптового языка AutoIT, подсчет оценки целесообразности внедрения автоматизированного тестирования.

В работе проводились обзор существующих методологий для проведения тестирования настольных приложений, сравнительный анализ программных средств, предназначенных для автоматизации процесса тестирования настольных приложений, изложены результаты сравнительного анализа, описан процесс тестирования приложения, дана экономическая оценка от внедрения автоматического тестирования на предприятии.

В результате разработана стратегия автоматизированного тестирования программного обеспечения, которая включает в себя план тестирования модуля формирования спецификаций «Specbook», а также

набор автотестов, созданных с помощью скриптового языка, которые после прохождения выдают отчет о прохождении теста с указанием обнаруженных ошибок, а также производят рассылку результатов тестирования заинтересованным лицам. Разработанные тесты позволяют сократить время на тестирование программы при ее дальнейшей модификации и поддержке, а так же покрывают требования, предъявляемые к программе на 100%, что позволяет избежать допущения новых ошибок и находить новые быстрее.

Данная работа сокращает время и средства, затраченные на проведение ручного тестирования при внесении изменений в данный модуль штатным тестировщиком.

В будущем планируется интеграция трекер-системы Redmine и Teamcity, многофункционального сервера непрерывной интеграции приложений, которая позволит создавать новые задачи в Redmine по результатам прохождения автотестов.

Выпускная квалификационная работа выполнена с использованием САПР AutoCAD Civil 3D, скриптового языка AutoIT, программного средства TestComplete, программных модулей Specbook и Автоматизированной системы для создания ведомости подсчета объемов работ по автомобильным дорогам и генеральным планам. Пояснительная записка выполнена в текстовом редакторе Microsoft Word 2010.

## ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ

ВКР – выпускная квалификационная работа.

ПО – программное обеспечение.

SpecBook (Система) – система автоматизированной генерации объектных спецификаций.

ОС – операционная система.

СУБД – система управления базой данных.

GUI – graphical user interface (англ. «графический интерфейс пользователя»).

Лог-файл - это файл текстового формата, в который заносятся все данные о неуспешном прохождении автотеста с указанием времени и вида ошибки.

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

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.

САПР – система автоматизированного проектирования.

ГИС – геоинформационная система.

СГИО – система геоинформационного обеспечения.

БГД – база геоданных.

AutoCAD Civil 3D – система автоматизированного проектирования.



## ОГЛАВЛЕНИЕ

Определения, обозначения, сокращения .....	8
Введение.....	10
1. Анализ предметной области .....	14
1.1. Описание деятельности организации ООО «ТомскАСУпроект» .....	14
1.2. Актуальность поставленной задачи.....	15
1.3. Определение процесса тестирования ПО .....	18
1.4. Характеристика тестируемых приложений .....	20
1.5. Критерии эффективности процесса тестирования.....	22
2. Расчеты и аналитика .....	26
2.1. Сравнительный анализ программных средств для автоматизации тестирования.....	26
2.2. Выбор методологии и видов тестирования .....	33
2.3. Расчёт эффективности внедрения автоматизированных тестов.....	34
3. Результаты выполнения разработки.....	40
3.1. Разработка тестовых сценариев .....	40
3.2. Разработка автотестов .....	46
3.3. Настройка непрерывной интеграции.....	55
3.4. Обработка ошибок.....	56
4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	60
5. Социальная ответственность .....	81
Заключение .....	102
Список использованных источников .....	104
Приложение А .....	107
Приложение Б .....	119

## ВВЕДЕНИЕ

Бурное развитие систем автоматизированной разработки программного обеспечения (ПО) и сетевых технологий повлекли за собой рост рынка производства программного обеспечения, а также к пересмотру вопросов, касающихся обеспечения надежности и качества разрабатываемого ПО. Конкуренция между производителями программных продуктов резко усилилась, что потребовало особого внимания к качеству создаваемого ПО, потому что теперь потребитель мог делать выбор: на рынке появилось много фирм, предлагающих свои продукты и услуги по низким ценам, но теперь потребитель мог обратиться не только к тем, кто разработает программу быстро и недорого, но еще и качественно. Компании осознали, что обеспечение высокого качества создаваемых программных продуктов, это верный шанс «обойти» конкурентов, поэтому сейчас компании во всем мире с каждым годом вкладывают всё больше средств, предназначенных для обеспечения качества своих продуктов, и создают свои отделы, которые занимаются тестированием, или прибегают к услугам сторонних организаций, занимающихся тестированием ПО.

На сегодняшний день тестирование стало неотъемлемой частью процесса разработки программных продуктов. С помощью него компании могут своевременно обнаружить и устранить большое число ошибок, что приводит к повышению качества всех характеристик ПО.

Автоматизация процесса тестирования помогает компаниям сокращать время, затрачиваемое на тестирование, а также упрощать весь процесс, потому что применяются специальные программные инструменты для создания и запуска тестов, а также проверки результатов их выполнения. Самой распространенной формой автоматизации тестирования можно считать тестирование приложений с помощью графического пользовательского интерфейса.

Любая компания, занимающаяся разработкой ПО, рано или поздно задумывается о внедрении автоматизированного процесса тестирования. Чаще всего задача по автоматизации применяется для выполнения большого количества похожих действий, которые выполняются в одной части приложения, а также для проведения регрессионного тестирования, чтобы освободить тестировщиков от рутинной работы и освободить ресурсы для новых разработок. Переход на автоматизированное тестирование, которое будет удовлетворять нуждам компании, является актуальной и сложной задачей, потому что нужно одновременно достичь соответствующего требованиям качества тестирования продукта, а также обеспечить необходимую экономию ресурсов.

Целью выпускной квалификационной работы является разработка и внедрение технологии автоматизированного тестирования программного обеспечения.

Для достижения цели необходимо решить ряд задач, а именно: провести сравнительный анализ средств, предназначенных для автоматизации процесса тестирования, разработать автотесты с помощью платного программного средства TestComplete для программного модуля «Автоматизированная система для создания ведомости подсчета объемов работ по автомобильным дорогам и генеральным планам», разработать автоматические тесты для настольного приложения «Модуль формирования спецификаций «Specbook» с функцией формирования отчетов с помощью скриптового языка AutoIT, а также провести оценку эффективности внедрения автоматических тестов для данных проектов.

В первом разделе описана актуальность поставленной задачи, раскрывается понятие процесса автоматизации тестирования, с его положительными и отрицательными сторонами. Кратко описываются программные модули, которые будут подвергнуты тестированию, и основные критерии выбора инструмента для тестирования. Описаны критерии, по которым будет производиться расчет эффективности от внедрения автотестов.

Во втором разделе описан подробный сравнительный анализ существующих программных средств, предназначенных для автоматизации тестирования настольных приложений. Описаны положительные и отрицательные стороны некоторых инструментов. Описаны причины выбора методологии и видов тестирования для данных программных модулей.

В третьем разделе изложены результаты разработки и внедрения технологии автоматизированного тестирования: описан процесс создания тестовых сценариев и разработки автотестов с помощью выбранных инструментов для автоматизации тестирования для программного модуля «Автоматизированная система для создания ведомости подсчета объемов работ по автомобильным дорогам и генеральным планам», а также для модуля формирования спецификаций «Specbook», которые после прохождения выдают отчет о прохождении теста с указанием места обнаружения ошибок, а также позволяют производить рассылку данных отчетов. Посчитана выгода от внедрения технологии автоматизированного тестирования, проведено сравнение критериев эффективности при использовании мощного инструмента с платной лицензией и инструмента со свободно распространяемой лицензией для тестирования.

Для выполнения работ использовались программное средство TestComplete, автоматизированная система для создания ведомости подсчета объемов работ по автомобильным дорогам и генеральным планам, скриптовый язык AutoIT, а также модуль формирования спецификаций «Specbook».

В результате выполнения работы разработана и внедрена стратегия автоматизированного тестирования, включающая в себя серии автоматических тестов для системы создания ведомости подсчета объемов работ по автомобильным дорогам и генеральным планам, а также серии автотестов для модуля формирования спецификаций «Specbook», которые позволяют сократить время на тестирование программы при внесении в нее каких-либо изменений, а так же позволяют наиболее эффективно покрыть

код тестами 100% функционала, согласно требованиям, что позволяет избежать допущения ошибок при доработке приложения. Также была посчитана экономическая эффективность от внедрения автоматических тестов, которая показала, что использование бесплатного инструмента автоматизации тестирования приносит большую выгоду, в отличие от внедрения платного инструмента, при внедрении которого компания понесет только убытки.

## 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

### 1.1. Описание деятельности организации ООО «ТомскАСУпроект»

Данная ВКР выполнена по заданию организации ООО «ТомскАСУпроект», которая осуществляет свою основную деятельность в области разработки и поддержки геоинформационных систем (ГИС) и систем автоматизированного проектирования (САПР) различного назначения. Организация занимается обработкой данных, их управлением, организацией их хранения и использования в разрабатываемых и поддерживаемых системах. ТомскАСУпроект предоставляет широкий спектр услуг по разработке программного обеспечения и IT-консалтингу. Они имеют опыт разработки готовых решений для реализации потребностей в различных отраслях.

Основными заказчиками являются следующие крупные компании: ПАО «НК «Роснефть», ОАО «ТомскНИПИнефть», ОАО «Томскнефть» ВНК, ООО «РН-СахалинНИПИморнефть», ООО «РНЮганскнефтегаз», ПАО «ВСНК», ФГАОУ ВО НИ ТПУ.

В организации применяются следующие технологии:

- Разработка веб-приложений. ASP.NET MVC 5, WebApi 2, JSON, HTML, CSS, JS, jQuery, ORM (NHibernate, Entity Framework), Node JS
- Используемые системы управления базами данных (СУБД)-MS SQL, ORACLE, PostgreSQL и др.
- Разработка настольных приложений. WPF, WCF, AutoCAD .NET, C++.
- Также компания занимается разработкой собственных продуктов, предназначенных для обеспечения полного цикла камеральных работ по подготовке топографических планов инженерно-строительных изысканий, для автоматизации процесса создания выкопировок по направлениям на основе общего сводного плана при ведении проектно-изыскательских работ, для автоматизации ведения ведомостей и реестра элементов дороги,

автоматизации расчёта нормативных противопожарных расстояний между объектами чертежа.

В штате компании работают более 30 высококвалифицированных специалистов: руководители проектов, программисты, системные аналитики, тестировщики, специалисты по обработке данных. Из них около 15 разработчиков и лишь три тестировщика, использующих технологию ручного тестирования, что в значительной степени усложняет процесс проверки ПО, потому что невозможно с помощью ручного тестирования качественно проверить все разрабатываемые приложения. В связи с тем, что основной объем разрабатываемых проектов составляют именно настольные приложения, которые постоянно нуждаются в тестировании, необходимо было провести сравнительный анализ имеющихся средств для автоматизации тестирования разрабатываемых продуктов для того, чтобы в дальнейшем повысить конкурентоспособность компании за счет выпуска более качественного программного обеспечения.

## **1.2. Актуальность поставленной задачи**

Тестирование является важной частью жизненного цикла разработки ПО. Этот процесс позволяет выполнять следующие задачи: выявлять ошибки в работе программы, проверять соответствие качества продукта с заявленными требованиями, предотвращать появление дефектов. Процесс тестирования включает в себя стадии планирования, разработки и реализации тестов, а так же поддержания их в актуальном состоянии.

Для корректного внедрения процесса тестирования на предприятии рекомендуется использовать международные стандарты по тестированию, в частности IEEE 829-1983 Standard for Software Test Documentation.

Современные программные средства имеют сложную структуру, и тестировать ее вручную трудно и дорого. Также результаты ручного тестирования могут пропасть, если их не фиксировать тестировщикам вручную, и их трудно повторить. Для упрощения и ускорения процесса

тестирования, для повышения качества тестирования, обеспечения возможности повторного использования тестов при внесении изменений в разрабатываемый продукт применяются средства для автоматизации тестирования [7].

Автоматизированное тестирование - это процесс проверки ПО, который позволяет запускать тесты и выдавать результат их прохождения автоматически при помощи программных средств автоматизированного тестирования [6].

С автоматизацией тестирования, связано много неверных представлений. Для того чтобы избежать неэффективного применения автоматизации, следует обходить ее недостатки и максимально использовать преимущества.

Автоматизация тестирования обладает рядом преимуществ:

- повторяемость – автотесты выполняются однообразно, без вмешательства человека, что исключает «человеческий фактор».
- быстрота – автотесту не нужно сверяться с документацией ПО, что помогает экономить время его выполнения.
- небольшие затраты на поддержку – написанные автотесты, нуждаются в меньшей поддержке и анализе результатов их выполнения, чем при ручном тестировании.
- построение отчетов – результаты выполнения автотестов можно автоматически рассылать, а также хранить отчеты о выполнении.
- не требует вмешательства – во время выполнения автотестов не требуется участия тестировщика, поэтому он может заниматься другой важной работой, также тесты можно запускать в нерабочее время.

Автоматизация тестирования обладает также рядом недостатков:

- однообразность – автотесты всегда выполняются однообразно, поэтому невозможно заметить недочеты, которые может найти тестировщик при ручном тестировании.



- большие затраты на разработку – разработка автотестов является сложным процессом, потому что фактически разрабатывается новое приложение, которое тестирует другое. Чем сложнее система, тем сложнее и дольше создавать для нее автотесты.

- стоимость инструмента автоматизации – мощный лицензионный инструмент для автоматизации тестирования, чаще всего имеет высокую стоимость. Свободно распространяемые средства имеют скромный функционал и не очень удобный интерфейс.

- пропуск ошибок - автотест пропускает ошибки, проверка которых не была в него заложена. К таким ошибкам можно отнести неточность в позиционировании окон, ошибки в надписях.

Список недостатков внушительный, самые серьезные из них это затрата лишнего времени и денег. В организации применяется только ручное тестирование приложений, потому что ручного тестирования хватало для выпуска качественного программного обеспечения. Но сейчас, с ростом поступающих заказов на разработку ПО организация столкнулась с проблемой нехватки времени на проведение ручного тестирования всех разрабатываемых приложений. Поэтому остро встал вопрос о внедрении на предприятии автоматизированных тестов. Но существует множество программных средств, предназначенных для автоматизации тестирования настольного ПО, и не каждое может подойти под нужды компании, поэтому было принято решение найти оптимальное бесплатное приложения для создания автотестов, которое позволит тестировщикам освободить их от рутинных занятий и появится время на что-то более важное без значительных дополнительных финансовых затрат [7].

Именно поэтому поиски подходящего инструмента для автоматизации тестирования настольных приложений и создание с помощью него автотестов является актуальной для данной организации.

Выпускная квалификационная работа направлена на описание процесса организации тестирования, разработку автоматических тестов для

настольных приложений, а также для расчета целесообразности такого внедрения.

### 1.3. Определение процесса тестирования ПО

По стандарту IEEE Std 829-1983 тестирование является процессом анализа программы, направленный на выявление отличий между его реально существующими и требуемыми свойствами (дефект) и на оценку свойств ПО [5].

По ГОСТ Р ИСО МЭК 12207-99 в жизненном цикле ПО определены среди прочих вспомогательные процессы верификации, аттестации, совместного анализа и аудита. Верификация - это процесс определения того, что программа функционирует в полном соответствии с требованиями или условиями, прописанными в документации к продукту. Этот процесс может включать в себя анализ, проверку и тестирование. Аттестация (валидация) служит для проверки соответствия требований и ожиданий конечного пользователя системы и их функциональному назначению. Совместный анализ служит для оценки состояний и, при необходимости, результатов работ по проекту. Аудит служит для определения соответствия требованиям, планам и условиям договора. Вместе эти процессы и составляют процесс тестирования.

Процесс тестирования схематично показан на рисунке 1.1.



Рисунок 1.1 – Процесс тестирования [5]

Основой тестирования являются тестовые процедуры (сценарии) с конкретными входными значениями, заданными начальными условиями и ожидаемым результатом, которые разрабатываются для проверки отдельной программы или определенного требования.

Перед разработкой теста нужно учесть, в соответствии с каким стандартом или требованием будет проводиться тестирование. Какие инструменты будут применены для поиска и фиксации ошибок. Необходимо всё продумать заранее, чтобы получить максимальную пользу от теста.

Правильное тестирование начинается с составления технического задания, в котором описываются все возможности и функции разрабатываемой системы, на которые в дальнейшем можно описаться при разработке тестовых сценариев. До начала разработки и тестирования ПО необходимо четко сформулировать требования, чтобы заказчик точно получил именно то, что он хотел, а также, чтобы требование можно было использовать как эталон во время проведения тестирования. Тестирование является не чем иным, как способом установления соответствия заранее определенным требованиям (функциональным, требованиям производительности, безопасности).

Для установления соответствия между требованиями и реальными результатами при тестировании создается тестовый план, который описывает, как должна вести себя система в определенной ситуации, что нужно подготовить для проведения тестирования, кто и как часто должен проводить тестирование.

План тестирования включает в себя тестовые сценарии (тест-кейсы), которые описывают последовательность шагов, конкретных условий и параметров, которые необходимы для проверки реализации тестируемой системы. Хорошо составленный тестовый план позволит обеспечить полное покрытие требований тестами, отслеживать изменения тестовых сценариев, а

также определиться с частотой проводимого тестирования, которая бы соответствовала частоте обновления программного продукта.

#### 1.4. Характеристика тестируемых приложений

Перед анализом и выбором инструментов для тестирования настольных приложений необходимо ознакомиться с самими тестируемыми модулями. Ознакомление происходило непосредственно с самой программой, а также путем изучения документации к данным программам.

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

Данная программа обеспечивает оперативное ведение реестра работ, реестра спецификаций работниками системы геоинформационного обеспечения (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 [11]. Модуль имеет интерфейс, представленный на рисунке 1.2.

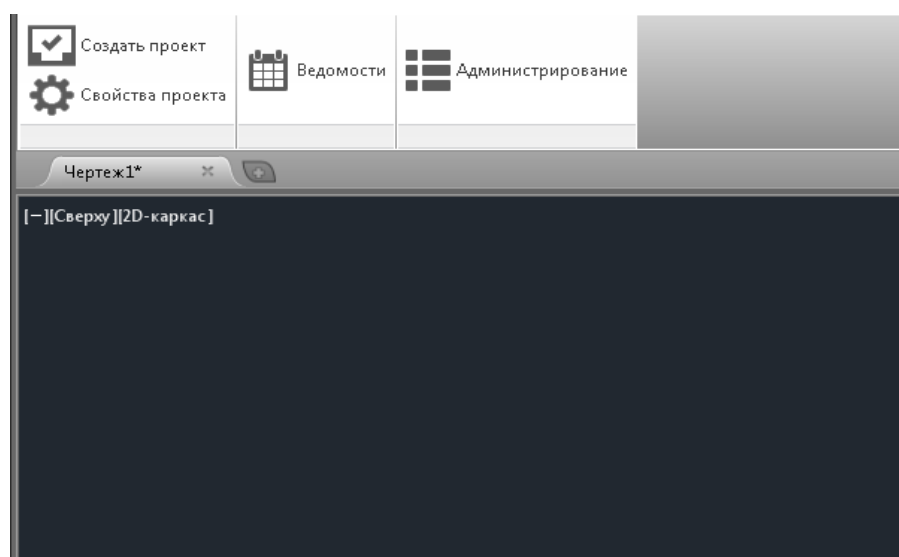


Рисунок 1.2 – Интерфейс модуля Полотно. Ведомости

Модуль формирования спецификаций «Specbook»

Система используется при проектировании обустройства месторождений для создания и выпуска заказных спецификаций. В рамках текущих работ, необходимо с учётом накопленного опыта использования Системы адаптировать её функционал в основном для создания объемов работ строительного отдела.

Интерфейс представлен на рисунке 1.3.

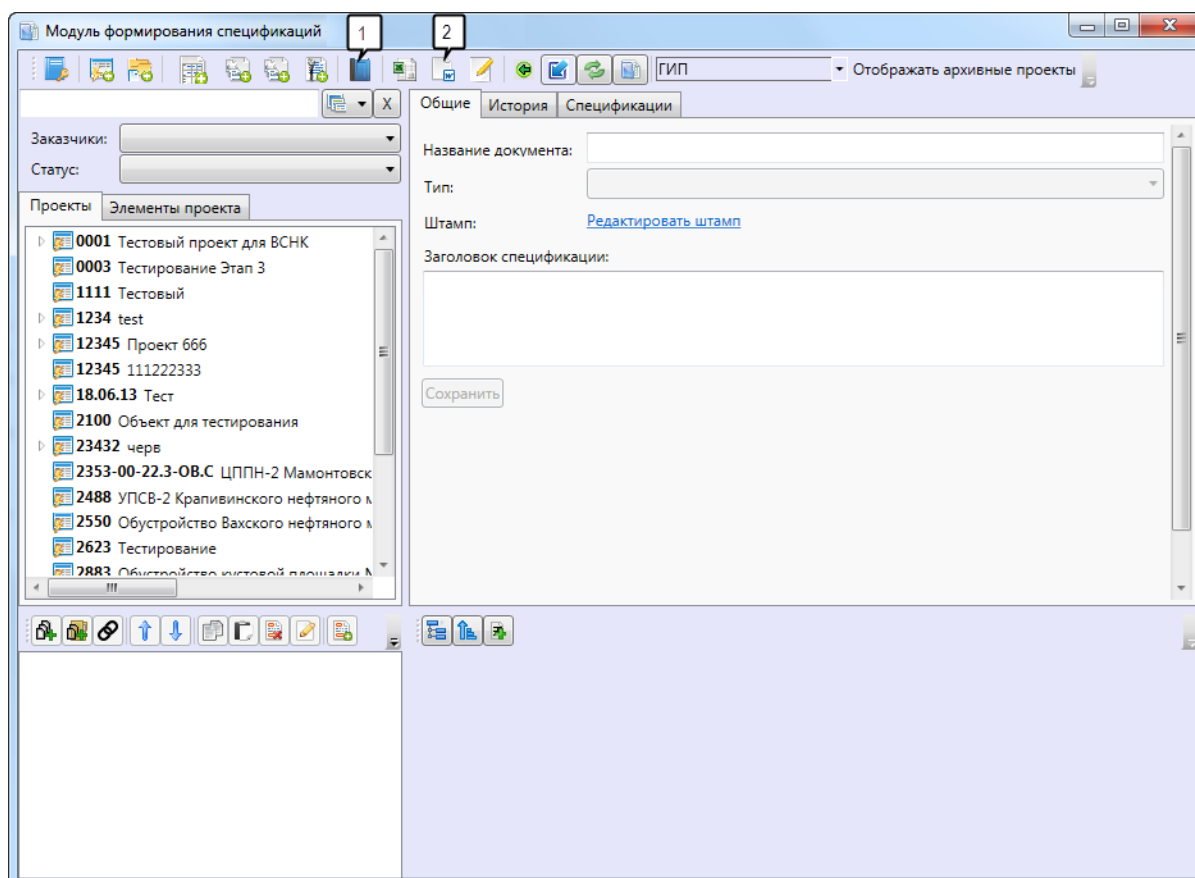


Рисунок 1.3 – Интерфейс модуля Sresbook

В ходе изучения были получены следующие знания о данных модулях, которые необходимо учитывать при выборе инструмента для тестирования.

1. Программный код не доступен для тестирования, поэтому его можно производить только через интерфейс пользователя.
2. Инструмент должен поддерживать приложения, написанные под AutoCAD Civil 3D.

3. Данные программные модули постоянно модифицируются, а это значит необходимо программное средство, которое легко поддерживать тестировщикам.

### **1.5. Критерии эффективности процесса тестирования**

Процесс тестирования должен быть эффективен в первую очередь с точки зрения компании, в которой он протекает. Компании могут быть интересны следующие параметры процесса тестирования:

- Время, необходимое для разработки тестов
- Время, которое занимает один цикл тестирования
- Квалификация персонала, необходимая для разработки и проведения тестов

Изменив любой из этих параметров, компания может повлиять на качество тестирования. Однако важно понимать, что любая комбинация этих параметров может быть выражена в денежном эквиваленте, и, как правило, у любого конкретного процесса тестирования есть оптимальная комбинация, при которой достигается достаточный уровень качества тестирования при минимальных денежных затратах.

Автоматизируя процесс тестирования, мы, разумеется, меняем процесс тестирования, а вместе с ним поменяется и оптимальная комбинация перечисленных выше параметров. Например, можно ожидать, что увеличится время, необходимое для разработки тестов и повысятся требования к квалификации персонала, при этом сильно понизится, время, занимаемое одним циклом тестирования. Учитывая, что комбинация параметров стала новой, вероятно поменяется и качество тестирования вместе с его стоимостью.

Для того чтобы была возможность дать численный эквивалент эффективности процесса тестирования, предлагается зафиксировать параметр качества на определенном уровне. Тогда численной оценкой эффективности определенного способа тестирования будет являться

величина инвестиций, необходимая для того, чтобы он обеспечивал некий определенный уровень качества.

Оценка целесообразности автоматизации тестирования производится с помощью подсчета затрат на ручное и автоматизированное тестирование и их сравнение. Точно посчитать финансовую целесообразность автоматизации тестов обычно невозможно, поскольку она зависит от параметров, которые в процессе разработки продукта могут быть лишь примерно понятны (например, планируемая длина жизненного цикла системы или точный список тестов, подлежащих автоматизации).

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

$$I_p = I_0 + C_0 + \sum_{n=1}^k (C_c + C_a + C_m) \quad (1.1)$$

$I_0$  - Оценка стартовых инвестиции, которые состоят из затрат на лицензии необходимого программного обеспечения для разработки автотестов, стоимости дополнительных аппаратных средств и прочего..

$C_0$  - Оценка стоимости разработки и отладки библиотеки автоматических тестов, которая рассчитывается 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.

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

$C_c$  - Оценка стоимости одного прогона всех автоматизированных тестов, которая рассчитывается как время, необходимое для подготовки к выполнению тестирования, сложенное с 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.

$G_a$  - Оценка затрат на анализ результатов одной итерации цикла автоматизированного тестирования, которая 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.

$G_m$  - 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.

Оценка стоимости ручного тестирования ( $G_p$ ) представлена в следующей формуле:

$$G_p = G_0 + \sum_{n=1}^k (G_e + G_a + G_m) \quad (1.2)$$

$G_0$  - Оценка стоимости разработки базы тест-кейсов для ручного тестирования.

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

$G_e$  - 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 тестировщиком, умноженное на суммарное количество кейсов и на цену одного рабочего часа тестировщика.

$G_a$  - 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 тестировщиком, и на цену одного рабочего часа тестировщика;



Gm - 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 тестировщика [7].

С помощью этих формул компания может посчитать, будет ли внедрение автоматизированных тестов экономически оправдано. В следующей главе на конкретном примере будет рассчитана экономическая целесообразность внедрения таких тестов.

## **2. РАСЧЕТЫ И АНАЛИТИКА**

### **2.1. Сравнительный анализ программных средств для автоматизации тестирования**

В настоящее время для веб-приложений есть непосредственный лидер в части разработки автоматических тестов инструмент, используемый для разработки тестовых сценариев Selenium IDE. При разработке веб-приложений используются открытые стандарты, что позволило разработчикам создать универсальный инструмент для автоматизации тестирования таких приложений. С настольными приложениями разработка универсального приложения усложняется тем, что такие приложения пишутся под различные операционные системы, с использованием различных технологий, поэтому хороший, качественный инструмент для тестирования таких приложений не имеет открытых или бесплатных лицензий.

На данный момент предприятие использует инструмент для автоматизации тестирования Selenium IDE, но с его помощью можно тестировать только веб-приложения

Но так как ООО «ТомскАСУпроект» занимается разработкой не только веб -, но и настольных приложений, необходимо найти программное средство, подходящее для тестирования ранее описанных настольных приложений. Для этого было необходимо провести сравнительный анализ имеющихся программных средств и языков для автоматизации процесса тестирования.

В процессе изучения и выбора средств для автоматизации тестирования была создана сравнительная таблица, приведенная в таблице 2.1.

Таблица 2.1 – Сравнительные характеристики программных продуктов для тестирования настольных приложений

Характеристика/ Инструмент	TestComplete	Ranorex	Microsoft Test Manager + Coded UI	AutoIt	(Winium)Обертка над Selenium от 2ГИС	SilkTest	Sikuli	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	Бесплатная	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	Бесплатная	30-ти дневный триал
<b>Необходимость программирования</b>	низкая	низкая	низкая	высокая	высокая	низкая	высокая	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++)	Jython, Java	VBScript
<b>Запись скриптов</b>	да	да	да	да	да	да	нет	да
<b>Инструменты CI</b>	Automated Build Studio	Jenkins plugin	TFS	-	любой	Open Agent	Любой java-совместимый	Jenkins plugin
<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	Видео (англ), Форумы

Характеристика/ Инструмент	TestComplete	Ranorex	Microsoft Test Manager + Coded UI	AutoIt	(Winium)Обертка над Selenium от 2ГИС	SilkTest	Sikuli	QuickTest Professional (UFT)
<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 самостоятельной разработки дополнительных модулей	Да, но при условии подключения дополнительных библиотек	Нет	Да, но при условии подключения дополнительных модулей, написанных на AutoLisp и AutoCad ActiveX Automation	Да, но требует самостоятельной разработки дополнительных модулей	Да, но требует самостоятельной разработки дополнительных модулей
<b>Возможность тестирования через GUI</b>	да	да	да	да	да	да	да	да
<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	Работает на уровне ОС	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	нет	да

По изначальному заданию компании предстояло найти мощный инструмент, который позволил бы за минимальные сроки автоматизировать процесс тестирования разрабатываемых приложений, который бы удовлетворял следующим критериям:

1. Возможность тестировать приложения, написанные для AutoCAD.
2. Минимальная необходимость программирования.
3. Простота изучения.
4. Невысокая стоимость (по возможности).

Как видно из таблицы 1, наиболее подходящими продуктами оказались TestComplete, Ranorex и Microsoft Test Manager + Coded UI.

Не смотря на высокую 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. Изучение программного средства для тестирования должно было пройти за минимально короткое время, для того, чтобы ознакомиться с процессом автоматизации тестирования. Только TestComplete имеет полноценный учебник на русском языке, в котором описаны основные возможности программы и приведены примеры создания автотестов.

Приведённые преимущества TestComplete, выявленные в ходе сравнительного анализа, так же подтвердил руководитель работы, имеющий успешный опыт работы с TestComplete. Так же он подчеркнул дружелюбный интерфейс TestComplete, позволяющий с минимальными сроками на обучение приступить к созданию автотестов, а также универсальность программы для выполнения автотестов как для настольных, так и для веб-приложений.

Конечно, такое приложение очень удобно для тестировщика, которому не придется тратить лишнее время на разработку скриптов, логирование тестов. Такие инструменты выгодно применять в крупных компаниях, которые разрабатывают одновременно много проектов и для автоматизации тестирования таких проектов должно уходить минимально возможное время. С таким инструментом сможет справиться любой, не обязательно со специальным образованием, человек. Но для маленьких компаний покупка такого инструмента не выгодна. На эту сумму можно нанять команду тестировщиков, которые будут заниматься ручным тестированием настольных приложений. На основании этого, было принято решение, что стоимость лицензии платного инструмента для автоматизации процесса тестирования не рентабельна, поэтому была поставлена задача поиска оптимального бесплатно распространяемого приложения с хорошей поддержкой.

Как видно из таблицы, наиболее подходящими инструментами являются AutoIT, Winium и Sikuli. Однако, хоть инструмент Sikuli и является универсальным, работает на всех платформах и со многими технологиями, он основан на распознавании графических образов, как таковые скрипты не создаются, поэтому такие тесты сложно поддерживать. Например, при обновлении или добавлении нового функционала придется менять все графические изображения, которые затронули эти изменения. Также, тесты, написанные на Sikuli нельзя считать надежными, потому что их проходимость может зависеть от разрешения экрана пользователя, ведь на

экранах с большим расширением оконные приложения могут растягиваться, а данный инструмент позволяет сравнивать изображения только попиксельно, поэтому, работающие тесты могут дать сбой, что может привести к поискам несуществующих ошибок и, следовательно, к лишним потерям времени и средств.

Winium является оберткой над инструментом для тестирования веб-приложений Selenium, которая поддерживает все языки и технологии, что и Selenium. Однако этот инструмент появился на рынке совсем недавно и на сегодняшний день не имеет пособий по работе с ним и библиотек.

AutoIT это свободно распространяемый язык для автоматизации выполнения задач в Microsoft Windows. Он имеет довольно удобный дизайн языка и достаточную для тестирования настольных приложений функциональность, что приблизило AutoIt к языкам общего назначения. Основным назначением является автоматизация работ с Windows GUI (графическим интерфейсом пользователя MS Windows). Для выполнения этой сложной задачи предоставляется комбинация испытанных методов, включающих в себя симуляцию нажатий комбинаций клавиш клавиатуры, перемещения указателя мыши и управления окнами и его визуальными элементами.

Данный скриптовый язык позволяет работать со строками, графическим интерфейсом, с файловой системой, имитировать пользовательский ввод мышью и клавиатурой, а также имеет множество других функций.

Интерпретатор AutoIt имеет небольшой размер и не требует обязательной установки. Для работы достаточно основного файла, а для создания скрипта — любого текстового редактора [9].

Хоть AutoIt и не является средством автоматизации тестирования, а лишь простым скриптовым языком, с помощью которого можно разрабатывать автотесты, и не имеет встроенных инструментов для логирования, хранения версий, построения отчетов об ошибках, это можно

написать самостоятельно, учитывая собственные нужды и возможности. Также, этот язык уже получил широкое распространение среди пользователей, поэтому имеется целая база уже написанных библиотек, подходящих, практически, под любые нужды. Поэтому, именно этот инструмент для автоматизации процесса тестирования был выбран для изучения и дальнейшего использования.



## 2.2. Выбор методологии и видов тестирования

Существуют различные методологии динамического тестирования ПО. Для тестирования выбранных приложений будет применяться метод черного ящика, потому что не представляется возможным узнать внутреннюю структуру и код тестируемого приложения. Поэтому в данном случае будет тестироваться поведение системы через внешние интерфейсы системы, такие как графический интерфейс.

Данный метод назван «Черным ящиком», поскольку в этом методе тестируемое программное обеспечение для тестировщика выглядит как черный ящик, внутри которого происходят некоторые процессы, однако тестировщику о них принципиально ничего не известно. Данная техника позволяет обнаружить ошибки в следующих категориях:

- Ошибки интерфейса.
- Недостающие или неправильно реализованные функции.
- Недостаточная производительность или ошибки поведения системы.
- Некорректные структуры данных или плохая организация доступа к внешним базам данных.

Таким образом, поскольку тестировщик не имеет никакого представления о внутреннем устройстве и структуре системы, ему необходимо сконцентрироваться на том, что делает программа, а не на том, как она это делает [2].

На следующем шаге необходимо выбрать те виды тестирования, которые подходят именно в данном случае.

При выборе видов тестирования, было необходимо руководствоваться типом и функционалом программного модуля. В ходе изучения было принято решение провести следующие виды тестирования:

1. Функциональное тестирование. Данный вид тестирования является очень важным, потому что именно с его помощью можно

проверить, все ли требования заказчика учтены и корректно работают, покрыты ли все случаи.

2. Тестирование установки и удаления программы. Данный вид тестирования позволяет проверить корректность установки (удаления) программного продукта (создаются и удаляются ли нужные файлы и папки в системе).

3. Тестирование лицензирования. Данный вид тестирования применяется для того, чтобы проверить корректность процесса лицензирования, а также, чтобы исключить возможность совершения противоправных действий пользователей по лицензированию программного модуля.

4. Позитивное тестирование. Тесты с позитивным сценарием проверяют способность программы выполнять заложенный в нее функционал.

5. Негативное тестирование. Негативное тестирование программного обеспечения происходит на сценариях, соответствующих нештатному поведению программы.

### **2.3. Расчёт эффективности внедрения автоматизированных тестов**

Перед разработкой автоматических тестов необходимо определить будет ли это выгодно для компании, сколько времени и средств будут экономиться при их использовании. Для расчета используются формулы, описанные в первой главе. В ходе опроса менеджеров были получены данные, необходимые для подсчетов.

Для модуля «Sresbook» и инструмента с открытой лицензией, применяемого для тестирования:

- Оплата тестировщика, оценивается в 250 рублей в час. Без учета затрат на налоги, аренду и прочее, т.к. эти суммы не дадут большой погрешности, потому что и для автотестов, и для ручных, будет использоваться один и тот же показатель.

- Данный проект рассчитан как минимум еще на три года. Обычно, тестирование проводится после выпуска новой версии, раз в месяц, но бывает, что после исправления критичных ошибок, найденных при тестировании, проверки необходимо выполнять заново. И того, примерно 0,3 прогона в неделю. При тестировании используется 250 тестов.

- На подготовку к циклу у ведущего тестировщика обычно уходит порядка 45 минут, преимущественно это время тратится на распределение задач между тестировщиками и другие организационные задачи. Среднее время, необходимое одному тестировщику на выполнение одного тест-кейса, составляет 10 минут.

- При каждом прогоне примерно 5% тестов имеют отрицательные результаты. На определение источника ошибки для каждого теста у ручного тестировщика уходит около 10 минут, в то время как при автоматизированном тестировании анализ ошибки занимает 15 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.

- Вероятность появления необходимости изменения одного теста между циклами тестирования оценена в 3%, Среднее время, необходимое для актуализации одного теста около 6 минут. Для актуализации автоматизированного теста потребуется 30 минут.

- Автоматизация одного теста оценивается в 3 часа

Для расчета затрат на автоматизированное тестирование воспользуемся формулой 1.1.

Начальные инвестиции в данном случае равны нулю, поскольку используется бесплатный скриптовый язык и отсутствует необходимость вкладываться в дополнительное оборудование.

Стоимость разработки автоматизированных тестов равна 187 500 рублей (250 тестов \* 3 часа \* 250 руб/час).

Планируемое количество циклов тестирования – 46,8 раз (3 года\*52 недели\*0,3 раза)

Оценка стоимости однократного выполнения цикла автоматизированного тестирования равна нулю, поскольку подготовка к циклу тестирования не требуется, а само тестирование не нуждается в дополнительном контроле со стороны работника и происходит полностью автономно.

Оценка стоимости анализа результатов выполненного цикла автоматизированного тестирования равна 781,25 рублей (250тестов \* 0.05 \* 0.25часа \* 250руб/час)

Оценка стоимости поддержания автоматизированных тестов в рабочем и актуальном состоянии равна 937,5 рублей (250тестов \* 0.03 \* 0.5часа \* 250руб/час).

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

$$0 + 187\,500 + 46,8 * (0 + 781,25 + 937,5) = 267\,937,5 \text{ рублей.}$$

Для расчета затрат на ручное тестирование воспользуемся формулой 1.2.

Оценка стоимости разработки базы тест-кейсов для ручного тестирования равна нулю, поскольку компания уже обладает базой тест-кейсов

Оценка стоимости однократного выполнения цикла ручного тестирования равна 10 812,5 рублей (0.75 + 250 тестов\* 0.17) \* 250руб/час.

Оценка стоимости анализа результатов для одного прогона цикла ручного тестирования равна 531,25 рублей (250 \* 0.05 \* 0.17 \* 250).

Оценка стоимости поддержания ручных тестов в актуальном состоянии равна 187,5 рублей (250 \* 0.03 \* 0.1 \* 250).

Итоговая стоимость затрат на ручное тестирование равна:

$$0 + 46,8 * (10\,812,5 + 531,25 + 187,5) = 539\,662,5 \text{ рублей.}$$

Следовательно, можно прийти к заключению, что выгода от использования составляет 271 725 рублей.

Далее, необходимо рассчитать экономическую выгоду от использования инструмента с платной лицензией TestComplete для тестирования модуля «Полотно.Ведомости»:

- Оплата тестировщика, оценивается также, в 250 рублей в час.
- Данный проект рассчитан на 2 года. Тестирование проводится после выпуска новой версии, раз в месяц, но бывает, что после исправления критичных ошибок, найденных при тестировании, проверки необходимо выполнять заново. И того, примерно 0,3 прогона в неделю. При тестировании используется 120 тестов.

- На подготовку к циклу у ведущего тестировщика уходит 30 минут. Среднее время, необходимое одному тестировщику на выполнение одного тест-кейса, составляет 10 минут.

- При каждом прогоне примерно 4% тестов имеют отрицательные результаты. На определение источника ошибки для каждого теста у ручного тестировщика уходит около 10 минут, в то время как при автоматизированном тестировании анализ ошибки занимает 15 минут.

- Вероятность появления необходимости изменения одного теста между циклами тестирования оценена в 7%, Среднее время, необходимое для актуализации одного теста около 6 минут. Для актуализации автоматизированного теста потребуется 30 минут.

- Автоматизация одного теста оценивается в 1 час

Для расчета затрат на автоматизированное тестирование воспользуемся формулой 1.1.

Начальные инвестиции в данном случае равны стоимости лицензии за 2 года. Из таблицы 1 видно, что годовая стоимость фиксированной лицензии для настольного приложения на TestComplete 2 067 евро = 131 068,47 рублей. Всего в компании три тестировщика, и, следовательно, чтобы повысить скорость работы, необходимо приобрести три лицензии.

Стоимость разработки автоматизированных тестов равна 30 000 рублей (120 тестов \* 1 часа \* 250 руб/час).

Планируемое количество циклов тестирования – 46,8 раз (2 года\*52 недели\*0,3 раза)

Оценка стоимости однократного выполнения цикла автоматизированного тестирования равна нулю, поскольку подготовка к циклу тестирования не требуется, а само тестирование не нуждается в дополнительном контроле со стороны работника и происходит полностью автономно.

Оценка стоимости анализа результатов выполненного цикла автоматизированного тестирования равна 375 рублей (120 тестов \* 0.05 \* 0.25 часа \* 250 руб/час)

Оценка стоимости поддержания автоматизированных тестов в рабочем и актуальном состоянии равна 1050 рублей (120 тестов \* 0.07 \* 0.5 часа \* 250 руб/час).

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

$$131\,068,47 * 2 * 3 + 30\,000 + 46,8 * (0 + 375 + 1050) = 883\,100,82 \text{ рублей.}$$

Для расчета затрат на ручное тестирование воспользуемся формулой 1.2.

Оценка стоимости разработки базы тест-кейсов для ручного тестирования равна нулю, поскольку компания уже обладает базой тест-кейсов.

Оценка стоимости однократного выполнения цикла ручного тестирования равна 5 225 рублей (0.5 + 120 тестов \* 0.17) \* 250 руб/час.

Оценка стоимости анализа результатов для одного прогона цикла ручного тестирования равна 204 рубля (120 тестов \* 0.04 \* 0.17 \* 250 руб/час).

Оценка стоимости поддержания ручных тестов в актуальном состоянии равна 210 рублей (120 тестов \* 0.07 \* 0.1 \* 250 руб/час).

Итоговая стоимость затрат на ручное тестирование равна:

$$0 + 46,8 * (5\ 225 + 204 + 210) = 263\ 905,2 \text{ рублей.}$$

Следовательно, для данного проекта использование платного инструмента для автоматизации тестирования не целесообразно, потому что затраты на лицензию очень велики, даже, если создавать тесты для нескольких проектов с помощью данной программы, существенную выгоду будет очень сложно получить.

## 3. РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ РАЗРАБОТКИ

### 3.1. Разработка тестовых сценариев

На данном этапе были созданы тестовые сценарии, необходимые для проверки функционала системы.

Для составления качественных тестовых сценариев необходимо ознакомиться с документацией к программе, которую необходимо тестировать. Для изучения функциональных возможностей программных модулей использовались технические задания к системе автоматической генерации спецификаций «Specbook» и к модулю «Автоматизированная система для создания ведомости подсчета объемов работ по автомобильным дорогам и генеральным планам»

По итогам изучения документации, все тест-кейсы были разбиты на подгруппы, это сделано с целью упрощения процесса разработки тестов, дальнейшего сбора тестов в модули для проверки конкретного участка программы, а также для изучения и понимания функционала программы другими тестировщиками.

Тест-кейсы описывают совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой системы. Частично их можно увидеть в приложении А.

Тест-кейсы имеют следующие атрибуты:

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. Если фактический результат совпадает с ожидаемым, то в секции ставится «+» или «пройден». Если же фактический результат отличается от желаемого, тогда в секции ставится «-» или «не пройден» и указывается, что не соответствует ожиданию.

Дополнительно могут описываться критерии окончания тестирования, риски, непредвиденные ситуации, приводятся ссылки на соответствующие разделы в основных документах проекта - план управления требованиями, план конфигурационного управления.

Тестирование модуля «Полотно.Ведомости» предстояло тестировать в паре, поэтому, было принято решение разделить функционал системы для тестирования. На рисунке 3.1 схематично представлен тестируемый функционал, где жёлтым выделен тот функционал, который повторяется в нескольких ведомостях. Белым тот, который выполняется единожды.

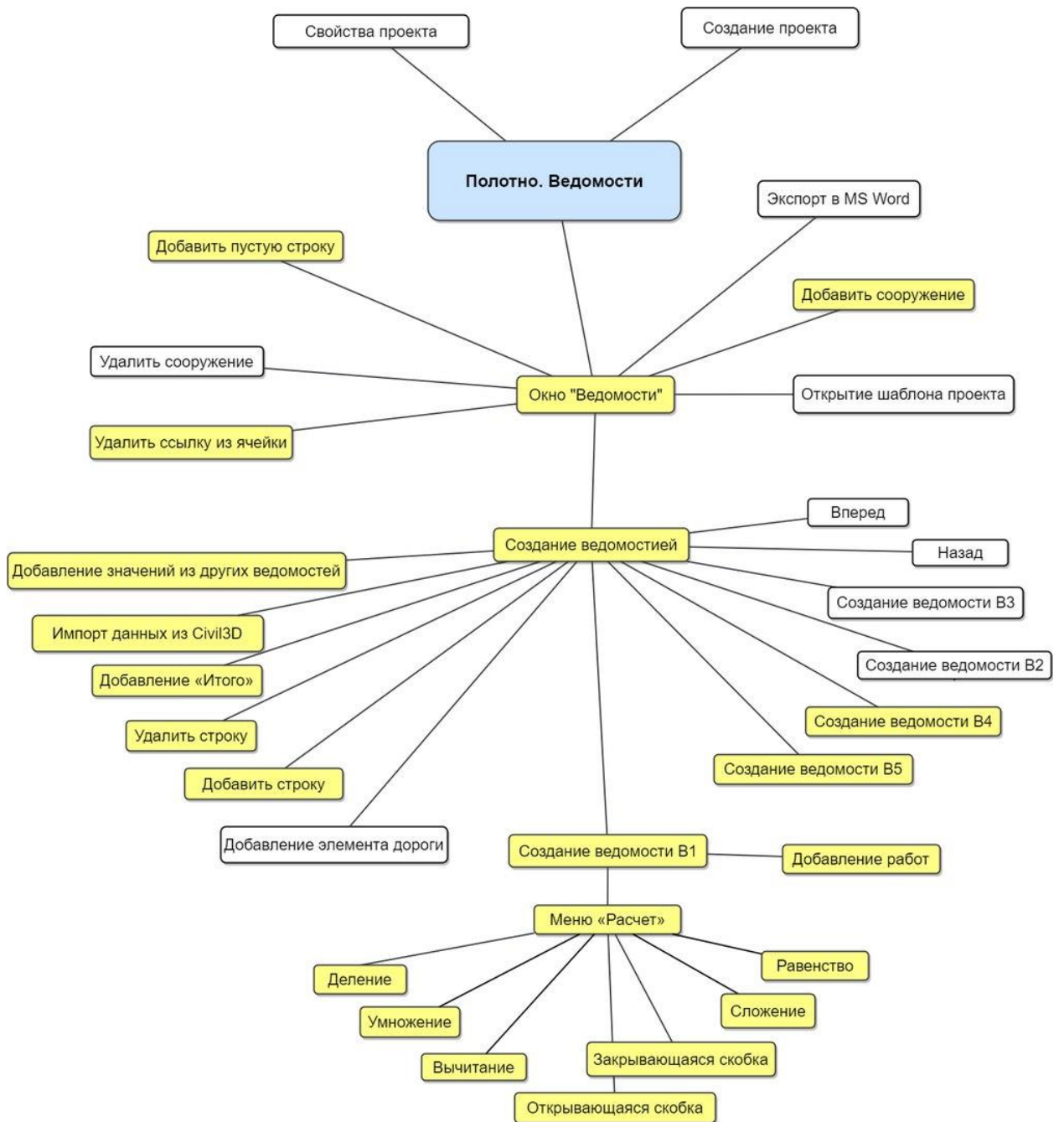


Рисунок 3.1 – Тест-кейсы для модуля «Полотно.Ведомости»

Ниже в таблицах 3.1 и 3.2 приведены некоторые тестовые сценарии, которые будут подвергнуты автоматизации.

Таблица 3.1– Тестовые сценарии для модуля «Specbook»

№	Название	Предварительные шаги	Шаги	Ожидаемый результат	Фактический результат
1.1	Установка модуля «Specbook»	1. Установить на настольный компьютер AutoCAD 2012.	1. Нажать на файл установки 2. Нажать «Далее» 3. Выбрать все устанавливаемые компоненты 4. Нажать «Далее» 5. Нажать «Закреть»	1. После шага 1 откроется окно установки модуля «Specbook». 2. После шага 2 откроется окно со списком устанавливаемых компонент. 3. После шага 3 напротив выбранных компонент появится галочка 4. После шага 4 откроется окно с процессом установки, когда процесс завершится, в окне появится сообщение о том, что установка завершена. 5. После шага 5 окно закроется, система завершит установку модуля.	
1.2	Создание каталога работ	1. Установить на настольный компьютер AutoCAD 2012. 2. Установить Specbook 3. Запустить Specbook	1. Нажать на кнопку  в главном окне модуля 2. Выбрать отдел в дереве объектов	1. После шага 1 откроется окно «Реестр работ» со списками элементов. 2. После шага 2 выбранный отдел выделится. 3. После шага 3 в разделе «Работы» появится новый каталог с именем «Каталог».	


			«Работы» 3. Нажать на кнопку  6.		
1.3	Переименование каталога работ	1. Установить на настольный компьютер AutoCAD 2012. 2. Установить Specbook 3. Запустить Specbook 4. Создать каталог работ (Тест 1.2)	1. Нажать ЛКМ дважды на каталог работ 2. Ввести имя 3. Нажать «Enter»	1. После шага 1 выделится имя каталога 2. После шага 2 введенное имя появится на экране 3. После шага 3 система сохранит имя каталога	
1.4	Удаление каталога работ	1. Установить на настольный компьютер AutoCAD 2012. 2. Установить Specbook 3. Запустить Specbook 1. Создать каталог работ (Тест 1.2)	1. Выбрать каталог из списка 2. Нажать на кнопку 	1. После шага 1 каталог выделится 2. После шага 2 выбранный каталог работ будет удален системой	

Таблица 3.2 – Тестовые сценарии для модуля «Полотно.Ведомости»

№	Название	Предварительные шаги	Шаги	Ожидаемый результат	Фактический результат
<b>1 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 индикатора загрузки	1. Запустить программу AutoCAD Civil 3D 2. Перейти на вкладку Полотно.Ведомости 3. Создать проект. Выполнить тест-кейс 1.2 или 1.3 4. Открыть окно «Ведомости». Выполнить тест-кейс 3.1.2 5. Создать сооружение. Выполнить тест-кейс 3.2.1 6. Создать записи в ведомостях В1-В5. Выполнить тест-кейсы 3.3.1-3.3.8	1. Нажать «Экспорт в Word» 2. Нажать на папку, в которую ранее экспортировались ведомости 3. Нажать «ОК» 4. Нажать крестик на индикаторе выполнения 5. Нажать «ОК»	1. После шага 1 откроется окно «Обзор папок» 2. После шага 2 выбранная папка выделится синим 3. После шага 3 появится индикатор выполнения 4. После шага 4 появится окно «Операция отменена» 5. После шага 5 станет активным окно «Ведомости». Система не произведёт экспорт ведомостей.	Ошибка: система производит частичный экспорт

Рассмотрим первый тест-кейс. Он содержит следующие обязательные атрибуты:

- Уникальный идентификатор тест-кейса: 1.1
- Название: Установка модуля «Specbook»
- Предусловия: в данном тест-кейсе предположением является установка на настольный компьютер AutoCAD 2012, в противном случае мастер установки модуля «Specbook» выдаст ошибку установки.
- Шаги.
- Ожидаемый результат.
- Фактический результат [3].

После разработки тестовых сценариев в табличном виде, были разработаны документ «Тестовые сценарии», в котором описан процесс проведения тестирования, который будет применяться при сдаче разработанного модуля, частично он представлен в приложении.

### **3.2. Разработка автотестов**

Для написания автотестов для программного модуля Полотно.Ведомости использовался программный продукт TestComplete. Он представляет собой функциональную автоматизированную платформу для тестирования, разработанную SmartBear Software. TestComplete дает тестировщикам возможность создавать автоматизированные тесты для Microsoft Windows, Web, Android и ОС IOS приложений. В TestComplete функциональные тесты могут создаваться в форме тестов на основе ключевых слов (keyword tests) или в форме скриптов [10]. Тесты обоих видов можно записать или создать с нуля с помощью встроенных в TestComplete редакторов. Интерфейс программы представлен на рисунке 3.2.

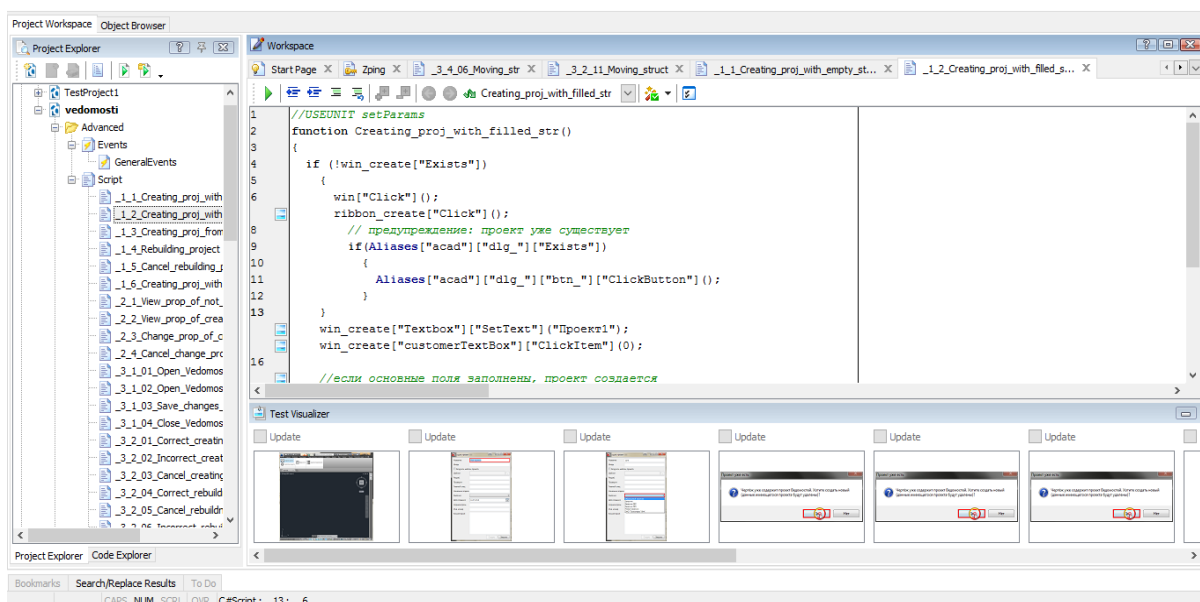


Рисунок 3.2 – Интерфейс программы TestComplete

Автотесты создавались на основе ключевых слов, затем генерировались в скрипты и далее дополнялись различными проверками.

Запись тестов в TestComplete проходит в три этапа:

1. Запуск записи теста (Record Keyword Test).
2. Выполнение тестовых действий, описанных в тест-кейсе.
3. Остановка записи теста и генерация его в скрипт.

Далее в скрипт добавлялись различные проверки: о том, что тестируемое окно открыто, необходимые поля заполнены и т.д.

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 равной длины, но отличающиеся по содержанию (рис.б). Эта проблема была решена заменой каждого символа кода подстановочным знаком (?). Подстановочные знаки - это специальные символы, которые могут заменять неизвестные символы в текстовом значении и удобны для поиска нескольких элементов с похожими, но не одинаковыми данными.

2. Элементы программы имеют длинные имена и для упрощения и понятности кода они были заменены переменными и помещены в отдельный скрипт (рис.3.3).

```
var ind = 1;
var win = Sys["Process"]("acad")["Window"]("Afx:????????????????:8:????????????????:????????????????:????????????????", "AutoCAD");
var ribbon_create = Sys["Process"]("acad")["Window"]("Afx:????????????????:8:????????????????:????????????????:????????????????", "AutoCAD");
var win_create = Aliases["acad"]["HwndSource_ProjectWindow"]["ProjectWindow"]["MainGrid"];
var ribbon_prop = Sys["Process"]("acad")["Window"]("Afx:????????????????:8:????????????????:????????????????:????????????????", "AutoCAD");
var ribbon_vedomosti = Sys["Process"]("acad")["Window"]("Afx:????????????????:2:????????????????:????????????????:????????????????", "AutoCAD");
var ribbon_admin = Sys["Process"]("acad")["Window"]("Afx:????????????????:8:????????????????:????????????????:????????????????", "AutoCAD");

var grid_elem = Aliases["acad"]["HwndSource_AddElementWindow"]["AddElementWindow"]["Grid"];
var grid_vedom = Aliases["acad"]["HwndSource_vedomostiWindow"]["vedomostiWindow"]["Grid"];
var grid_link = Aliases["acad"]["HwndSource_LinksWindow"]["LinksWindow"]["Grid"];
var grid_total = Aliases["acad"]["HwndSource_V2TotalRowWindow"]["V2TotalRowWindow"]["Grid"];

var win_vedomosti = Aliases["acad"]["HwndSource_vedomostiWindow"];
var win_prop = Aliases["acad"]["HwndSource_ProjectWindow2"]["ProjectWindow2"]["MainGrid"];
```

Рисунок 3.3 – Скрипт, содержащий сокращения системных имен элементов

Для написания автотестов для программного модуля Specbook использовался скриптовый язык AutoIT.

При тестировании установки модуля «Specbook» необходимо было произвести следующие проверки:

1. Наличие на компьютере установленного AutoCAD 2012, потому что «Specbook» является модулем данной программы.
2. Наличие ранее установленной версии модуля.
3. Открытие всех окон установки.

Скрипт, который проверяет установку модуля приведен ниже:

```
#include <File.au3>
#include <Log.au3>
#include <Outlook.au3>

If WinExists(@ScriptName) Then Exit
AutoItWinSetTitle(@ScriptName)

;Имя основного окна
Global $Title="Установка - SpecBook"
; Имя окна при отсутствии автокада
```



```

Global $title_1="Ошибка"
; Имя окна при повторной установке
Global $title_2="Установка"

Install_SpecBook()

Func Install_SpecBook()
Run('@ProgramFilesDir & \SpecBook*.exe')
Sleep (5000)

if WinExists($title_1) Then
ControlClick($title_1,'OK','Button1')
_Log("Не установлен AutoCAD",$Title)
Exit
Endif

if WinExists($title_2) Then
ControlClick($title_2,'Да','Button1')
ExitLoop
Endif

WinWaitActive($Title)
If WinExists($Title) then
WinActivate($Title)
ControlClick($Title, "Далее", 'Button2')
Else
_Log ("Окно установки не открылось",$Title) exit
EndIf
Sleep (5000)

WinWaitActive($Title)
If WinExists($Title) then
WinActivate($Title)
ControlClick($title_1, "Каталог Элементов", 'Textbox2')
Else
_Log ("Окно выбора каталогов не открылось",$Title) exit
EndIf
Sleep (5000)

WinWaitActive($Title)
If WinExists($Title) then
WinActivate($Title)
ControlClick($title_1, "Далее", 'Button2')
Else

```

```
_Log ("Окно установки не открылось", $Title) exit  
EndIf  
Sleep (5000)
```

```
WinWaitActive($Title)  
If WinExists($Title) then  
WinActivate($Title)  
ControlClick($title_1, "Установить", 'Button3')  
Esle _Log ("Установка не удалась", $Title) exit  
EndIf  
Sleep (5000)
```

```
WinWaitActive($Title)  
If WinExists($Title) then  
WinActivate($Title)  
ControlClick($title_1, "Завершить", 'Button4')  
Esle _Log ("Не удалось завершить установку", $Title) exit  
EndIf
```

```
If FileNoExists("C:\Program Files (x86)\AutoCAD\*.spb") Then  
Esle _Log ("Проверочный файл не создан", $Title) exit  
EndIf
```

```
_Mail($Title)  
Exit
```

Дополнительные скрипты и библиотеки подключаются с помощью  
#include.

Далее, следует запись, которая предотвращает повторный запуск теста,  
если он был запущен ранее:

```
If WinExists(@ScriptName) Then Exit  
AutoItWinSetTitle(@ScriptName)
```

Если по одной из причин тест не удастся, то сообщение об этом  
записывается в лог-файл, который имеет следующий формат (рис.3.4):

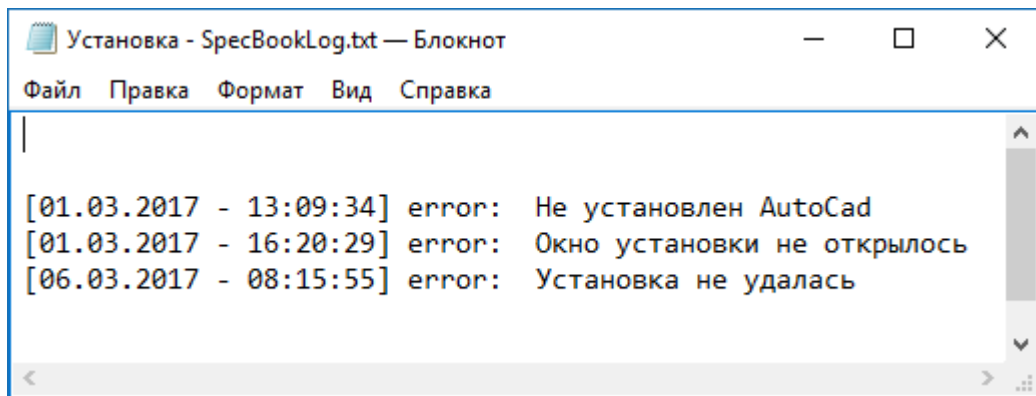


Рисунок 3.4 – Лог-файл процесса установки модуля

Как видно из рисунка, лог-файл представлен документом с расширением \*.txt, заголовок которого описывает назначение тестового скрипта. Записи об ошибках имеют информацию о времени записи и о виде ошибки, по которой можно в дальнейшем сформировать ошибку в системе Redmine.

Такое логирование происходит с помощью следующей функции:

```
Func _Log($LogTe, $Title)
    $LogFile = @ScriptDir & "\"&$Title&"Log.txt"
    If Not FileExists($LogFile) Then _FileCreate($LogFile)
    Local $log_file_id = FileOpen($LogFile, 1)
    FileWriteLine($log_file_id, "["&@YEAR&"-"&@MON&"-"&@MDAY&" :
        "&@hour&"-"&@min&"]"&StringFormat("error: "&$LogTe))
    FileClose($log_file_id)
EndFunc
```

Где \_Log () – название функции,

\$LogTe – переменная с текстом ошибки

\$Title – название теста

Как видно из скрипта, проверке подвергается каждое окно, которое должно появляться при установке, и, при его отсутствии, запись об этом с указанием события, которое не удалось, попадает в лог-файл, а выполнение тестового скрипта завершается.

После окончания установки, результаты тестирования отправляются на почту разработчику письмом, содержащим название теста и файл с указанием ошибки.

Скрипт, который отвечает за отправку почты, представлен ниже.

```
Func _Mail($Title)
    $SmtpServer = "smtp.mail.ru" ; адрес SMTP сервера
```

```

$FromName = "Tester1" ; имя отправителя
$FromAddress = "fedorchenkoa@tomskasu.ru" ;адрес отправителя
$Password = "qweQWE123"; пароль аккаунта, с которого отправляется
$ToAddress = "shabanovsn@tomsasu.ru"; адрес назначения
$Subject = "Результаты проверки - "&$Title; тема письма
$Body = "Результаты выполнения теста "&$Title&" во вложении."; тело письма
(сам текст письма)
$AttachFiles = "C:\"& $Title& ".txt"; прикрепленный файл

```

```

$BlatArgs = "-t "&$ToAddress&" -f "&$FromAddress&" -s "&$Subject&" -body
"&$Body&" -server "&$SmtServer&" -u "&$Username&" -pw "&$Password&" -
attach "&$AttachFiles
Run (@scriptdir & "\blat.exe " & $BlatArgs, @ScriptDir, @SW_HIDE )
_SendMail($BlatArgs)

```

```

Func _SendMail($CMDstring)
$BlatDLL = DllOpen ("blat.dll")
Do
$result=DllCall($BlatDLL,"int","Send","str",$CMDstring)
Select
Case $result[0] = 12
_Log ("Ошибка. Код ошибки " &$result[0] & ". Не указаны или не найдены в
реестре аргументы -server или -f",$Title)
Case $result[0] = 3
_Log("Ошибка. Код ошибки " &$result[0]& ". Не могу прочитать файл с текстом
письма или приложенный файл",$Title)
Case $result[0] = 2
_Log("Ошибка. Код ошибки " &$result[0]& ". Почтовому серверу не понравился
адрес отправителя; или он отказал нам в подключении; или неверный пароль;
или не найден файл с текстом письма",$Title)
Case $result[0] = 1
_Log ("Ошибка. Код ошибки " &$result[0]& ". Не могу открыть SMTP сокет.
Проверьте подключение к Интернету; либо неверные аргументы",$Title)
Case $result[0] <> 0
_Log ("Ошибка. Код ошибки " &$result[0]& ". Неизвестная ошибка",$Title)
EndSelect
Until $result[0] = 0
_Log ("нет ошибки.", "Письмо отправлено" ,$Title)
DllClose ($BlatDLL)
EndFunc
EndFunc

```

Письмо, которое получают разработчики, имеет следующий формат,

представленный на рисунке 3.5.

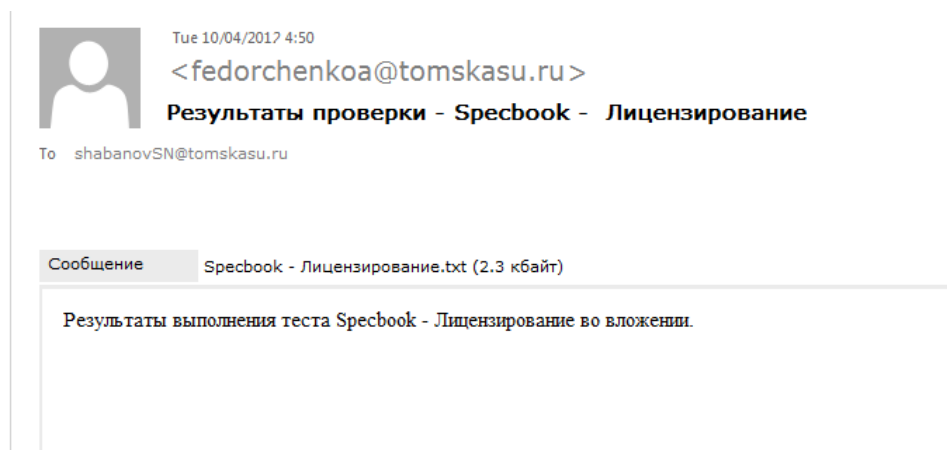


Рисунок 3.5 – Схема запусков модуля для проверки корректной работы лицензирования

Из рисунка видно, что скрипт, с помощью которого осуществляется рассылка сообщений об ошибке, позволяет заполнить следующие поля:

Тему письма:  $\$Subject = \text{"Результаты проверки - " \& \$Title}$ , где  $\$Title$  – это название теста, отчет о котором рассылается.

Сообщение:  $\$Body = \text{"Результаты выполнения теста " \& \$Title \& " во вложении."}$

Прикрепленный файл с отчетом:  $\$AttachFiles = \text{"C:\& \$Title \& ".txt"}$

При тестировании лицензирования модуля «Спебук» необходимо было произвести следующие проверки:

1. Наличие на компьютере установленного модуля «Спебук».
2. Наличие файла с лицензией.
3. Проверка корректности поведения лицензии.

Для оценки корректности лицензирования была разработана схема проверок, представленная на рисунке 3.6, которая имитирует преступные действия пользователей в попытке продления действия лицензии.

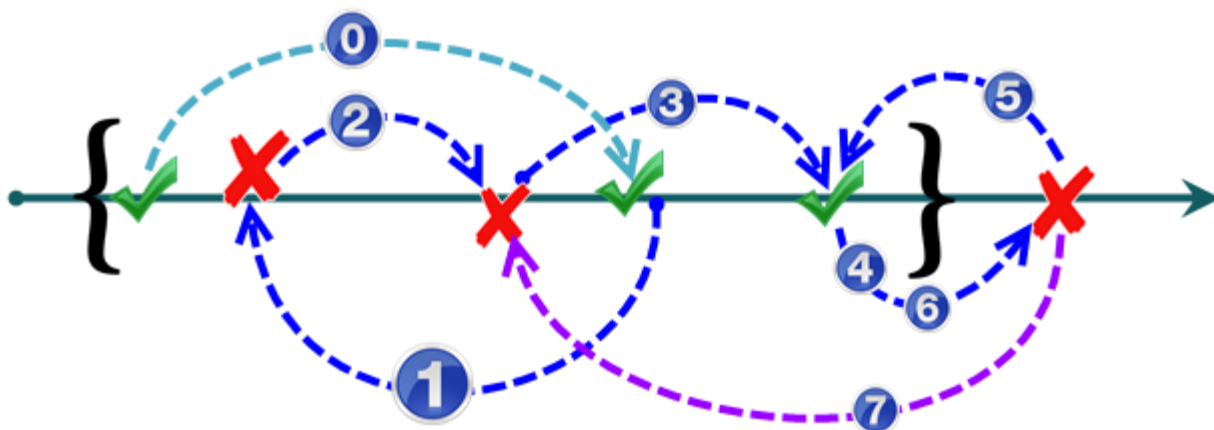


Рисунок 3.6 – Схема запусков модуля для проверки корректной работы лицензирования

Скобки обозначают срок действительной лицензии.

Стрелки показывают, в какую сторону необходимо провести сдвиг системного времени.

Галочки показывают, после каких изменений система должна продолжать корректно работать.

Крестики показывают, после каких изменений система должна выдавать ошибку лицензии.

Для имитации перевода системного времени пользователем использовалась функция:

```
_SetDate(@MDAY, @MON + 1, @YEAR)
```

После едино кратного изменения времени согласно схеме, представленной на рисунке 3.6, выполнялись запуск, удаление и повторная установка программы. Если возникали ошибки в процессе проверки лицензирования, информация о них также записывалась в лог.

При тестировании функционала, согласно требованиям, было разработано порядка 250 тестов, покрывающих 100% всего функционала.

При функциональном тестировании возникали следующие проблемы:

При проверке правильности ввода данных в таблицу возникла проблема с иницированием ячеек. Ячейки как таковые отсутствовали, поэтому пришлось имитировать не нажатие ячейки, а нажатие мышью на определенные координаты окна, и затем уже вводить значение:

MouseClicked ('left',184,1007,1,10)

Send('Ведомость')

Позже, разработанные тесты были сформированы в группы, что позволит производить только определенную группу тестов для проверки отдельного модуля программы.

### **3.3. Настройка непрерывной интеграции**

Разработанные тесты для проверки качества программного обеспечения должны быть доступны любому разработчику и тестировщику, чтобы они могли быстро и оперативно запускать, изменять тесты для различных программ. Для программного кода в организации применяется сервер непрерывной интеграции TeamCity, поэтому было принято решения использовать именно его для хранения и запуска автоматических тестов.

Сначала была создана виртуальная машины, предназначенная для запуска и отслеживания автоматических тестов, на которой был развернут сервер непрерывной интеграции TeamCity.

Автотесты иногда нуждаются в обновлении и модернизации, потому что они разрабатываются параллельно с программным обеспечением, в которое может получить новый или измененный функционал. Важно, чтобы автотесты оставались всегда в актуальном состоянии.

Все изменения, внесенные в тесты, должно фиксироваться тестировщиком, потому что другие должны быть в курсе, какая часть теста изменилась, и кто ее изменить. Потому что людям свойственно ошибаться, а из-за ошибок тестировщика тесты могут не запуститься, и, если не фиксировать, кто и когда внес изменения, будет очень сложно искать эти ошибки. Поэтому очень важно отслеживать версию автотестов. Для этого в компании применяется Gitlab – система управления исходным кодом, которая позволяет отслеживать версию тестов, а также оставлять комментарии тестировщикам с подробным описанием внесенных изменений в каждый тест,

что поможет в дальнейшем оперативно отслеживать внесенные изменения и исправлять ошибки в тестовых скриптах.

Далее, для того, чтобы любой тестировщик или программист мог запустить разработанные тесты, отследить их код выполнения или просмотреть результаты тестирования, они загружаются на сервере TeamCity – системы управления распределенными процессами сборки кода и система непрерывной интеграции процессов разработки, которая позволяет запускать тесты на специально созданной виртуальной машине для тестирования. TeamCity позволяет запускать тесты по расписанию или после загрузки или обновления какой-то части кода тестируемого приложения, а также хранит историю запусков в отчеты, что позволяет оперативно реагировать на ошибки и не требуют дополнительного вмешательства тестировщика.

### **3.4. Обработка ошибок**

С помощью разработанных тестов был выявлен ряд ошибок, требующих оперативное устранение, в их число вошли:

Ошибки, обнаруженные в «Specbook»:

1. Ошибки лицензирования: не записывалась информация о дате истечения лицензии в системный файл, поэтому пользователь мог бы пользоваться одной лицензией постоянно.

2. Ошибка ограничения ввода: поле

Ошибки, обнаруженные в модуле «Полотно.Ведомости»:

1. Ошибка удаления каталога из списка: после удаления окно с ведомостями становилось неактивным до перезапуска модуля.

2. Ошибка ограничение на ввод значений: в числовое значение вводились текстовые символы, в качестве значения длины дороги можно было вводить отрицательные значения.

3. Ошибка ограничение на ввод значений: не было ограничений на ввод значения радиуса, что приводило к некорректным расчетам.

Все ошибки вносились в 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.

**Задание #1705** ✎ Редактировать 📊 Трудозатраты 📈 Следить 📄 Копировать 🗑 Удалить

**Ошибки Ведомостей от Алёны** ◀ Предыдущее | 2/11 | Следующее ▶

Добавил(а) Федорченко Алёна 21 дня назад. Обновлено 21 дня назад.

Статус: Новая Дата начала: 13.07.2016  
 Приоритет: Нормальный Дата завершения: Готовность: 0%  
 Назначена:  Онищенко Михаил Трудозатраты: -  
 Категория: - Приоритет тестирования:  
 Версия: -  
 Вес: 0  
 Этап:  
 Этапы:

**Описание** 💬 Цитировать

Ошибки, найденные при тестировании

Подзадачи	Статус	Исполнитель	Готовность	Добавить
Ошибка #1422: Ведомости. Ошибка выбора значений при расчете B1	Новая	Онищенко Михаил		<input type="button" value="Добавить"/>
Задание #1454: Ведомости. Ошибка удаления записей из ведомости B2	Новая	Онищенко Михаил		<input type="button" value="Добавить"/>
Задание #1470: Ведомости. Ошибка отображения отмены действия	Новая	Онищенко Михаил		<input type="button" value="Добавить"/>
Ошибка #1706: B3. Ошибка при изменении радиуса	Новая	Онищенко Михаил		<input type="button" value="Добавить"/>
Ошибка #1707: Возможен ввод отрицательных значений длин	Новая	Онищенко Михаил		<input type="button" value="Добавить"/>
Ошибка #1708: Ошибка при пересчете зависимых значений	Новая	Онищенко Михаил		<input type="button" value="Добавить"/>
Ошибка #1710: Разный результат при вводе букв(и иных символов) в числовые значения	Новая	Онищенко Михаил		<input type="button" value="Добавить"/>
Ошибка #1711: Ошибки при изменении значений в столбцах «От ПК+» «До ПК+»	Новая	Онищенко Михаил		<input type="button" value="Добавить"/>
Ошибка #1712: Разные результаты при изменении значений в ячейках	Новая	Онищенко Михаил		<input type="button" value="Добавить"/>
Задание #1840: Ошибка удаления сооружения	Новая	Онищенко Михаил		<input type="button" value="Добавить"/>

**Связанные задачи** Добавить

✎ Редактировать 📊 Трудозатраты 📈 Следить 📄 Копировать 🗑 Удалить  
 Экспортировать в

Рисунок 3.7 – Задача в РМ. Ошибки, обнаруженные в программе «Полотно.Ведомости»

Под задачей в данной информационной системе подразумевается объект системы РМ, отражающие суть ошибки, задачи, поручения или работы. Когда ошибка будет исправлена, разработчик изменит статус ошибки на «Тестирование» и она снова вернется тестировщику для проверки.

Рисунок 3.8 – Создание ошибки в RM

Благодаря универсальности разработанных тестов для данного модуля, это позволяет расширять функциональность с минимальными человеческими затратами. В настоящий момент покрытие кода автоматическими тестами для данного модуля составляет 100% от требований, обозначенного в техническом здании к данному модулю.

В дальнейшем планируется разработать систему, которая позволит отслеживать ошибки, обнаруженные при тестировании и автоматически создавать на их основе ошибки в трекер-системе Redmine чтобы исключить ручное создание таких ошибок тестировщиком, что позволит сэкономить временные ресурсы тестировщиков на другие проекты.

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

Студенту:

<b>Группа</b>	<b>ФИО</b>
8ИМ5А	Федорченко Алёна Вадимовна

<b>Институт</b>	<b>ИК</b>	<b>Кафедра</b>	<b>ИСТ</b>
Уровень образования	магистратура	Направление/специальность	09.04.02 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, нормативно-правовых документах, предоставленной организацией</i>
2. <i>Нормы и нормативы расходования ресурсов</i>	<i>Работа с нормативно-правовыми документами</i>
3. <i>Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>	<i>Работа с действующим налоговым кодексом РФ</i>

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

1. <i>Оценка коммерческого и инновационного потенциала НТИ</i>	<i>Оценка потенциальных потребителей исследования, SWOT-анализ, исследование альтернатив проведения НИ, создание графика проведения и бюджета НИ, оценка ресурсной, финансовой и экономической эффективности НИ</i>
2. <i>Разработка устава научно-технического проекта</i>	<i>Планирование этапов работ, определение трудоёмкости и построение календарного графика, формирование бюджета</i>
3. <i>Планирование процесса управления НТИ: график проведения, бюджет, риски и организация закупок</i>	<i>Расчет плановой стоимости затрат, необходимых для реализации проекта, а также анализ возможных рисков</i>
4. <i>Определение ресурсной, финансовой, экономической эффективности</i>	<i>Расчет экономической выгоды от внедрения разрабатываемого проекта</i>

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

1. Оценка конкурентоспособности технических решений
2. Диаграмма FAST
3. Матрица SWOT
4. График проведения и бюджет НТИ
5. Оценка ресурсной, финансовой и экономической эффективности НТИ
6. Потенциальные риски

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

**Задание выдал консультант:**

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
Доцент кафедры менеджмента	Попова Светлана Николаевна	К.Э.Н		

**Задание принял к исполнению студент:**

<b>Группа</b>	<b>ФИО</b>	<b>Подпись</b>	<b>Дата</b>
8ИМ5А	Федорченко Алёна Вадимовна		

## 4. ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ

### 4.1. Предпроектный анализ

#### 4.1.1. Потенциальные потребители результатов исследования

Целевым рынком разрабатываемой технологии автоматизированного тестирования настольного приложения «Модуль формирования спецификаций «Specbook» является компания, для нужд которой создавалась данная технология – ООО «ТомскАСУпроект», данным решением будут пользоваться только сотрудники компании.

#### 4.1.2. Анализ конкурентных технических решений с позиции ресурсоэффективности и ресурсосбережения

Для выбора необходимого инструмента для тестирования настольного приложения необходимо провести сравнительный анализ аналогичных решений.

Для наглядности демонстрации преимуществ выбранного скриптового языка AutoIT, предназначенного для тестирования настольных приложений, проведем анализ с помощью оценочной карты.

В качестве конкурирующих технологий возьмем следующие инструменты, предназначенные для автоматизации тестирования:  $B_{\phi}$  – AutoIT,  $B_{к1}$  – Sikuli,  $B_{к2}$  – Winium.

Таблица 4.1 – Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		$B_{\phi}$	$B_{к1}$	$B_{к2}$	$K_{\phi}$	$K_{к1}$	$K_{к2}$
1	2	3	4	5	6	7	8
<b>Технические критерии оценки ресурсоэффективности</b>							
1. Повышение производительности труда пользователя	0,2	3	2	5	0,6	0,4	1
2. Удобство в эксплуатации	0,1	4	2	5	0,4	0,2	0,5
3. Надежность	0,05	5	2	4	0,25	0,1	0,2

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
4. Поддержка	0,1	4	2	3	0,4	0,2	0,3
5. Функциональная мощность (предоставляемые возможности)	0,1	3	1	5	0,3	0,1	0,5
6. Поддерживаемые технологии	0,05	4	2	3	0,2	0,1	0,15
7. Понятность интерфейса	0,05	2	1	4	0,1	0,05	0,2
<b>Экономические критерии оценки эффективности</b>							
1. Конкурентоспособность продукта	0,07	5	2	4	0,35	0,21	0,21
2. Уровень проникновения на рынок	0,04	5	5	2	0,35	0,21	0,28
3. Цена	0,1	5	5	5	0,35	0,35	0,35
4. Финансирование научной разработки	0,1	5	1	2	0,28	0,21	0,28
5. Срок выхода на рынок	0,04	5	5	5	0,35	0,35	0,35
<b>Итого</b>	<b>1</b>	<b>50</b>	<b>30</b>	<b>47</b>	<b>4</b>	<b>2,29</b>	<b>3,6</b>

Анализ оценочной карты показал, что наивысший уровень конкурентоспособности имеет используемый инструмент AutoIT (4 единицы). Такой уровень конкурентоспособности у данного продукта достигается благодаря хорошей поддержке инструмента, высокой надежности и удобству использования. В отличие от конкурентных систем AutoIT работает на уровне операционных систем, а это значит, что данный инструмент не зависит от языка программирования и с его помощью можно тестировать любое настольное приложение.

Winium занимает второе место по результатам оценочной карты. Он является оберткой над инструментом для тестирования веб-приложений Selenium, которая поддерживает все языки и технологии, что и Selenium. Однако этот инструмент появился на рынке совсем недавно и на сегодняшний день не имеет пособий по работе с ним и библиотек.

Sikuli занимает самую низкую позицию, хоть инструмент Sikuli и является универсальным, работает на всех платформах и со многими технологиями, он основан на распознавании графических образов, как таковые скрипты не создаются, поэтому такие тесты сложно поддерживать. Например, при обновлении или добавлении нового функционала придется менять все графические изображения, которые затронули эти изменения. Также, тесты, написанные на Sikuli нельзя считать надежными, потому что их проходимость может зависеть от разрешения экрана пользователя, ведь на экранах с большим расширением оконные приложения могут растягиваться, а данный инструмент

позволяет сравнивать изображения только попиксельно, поэтому, работающие тесты могут дать сбой, что может привести к поискам несуществующих ошибок и, следовательно, к лишним потерям времени и средств.

Таким образом, конкурентоспособность инструмента AutoIT достигнута благодаря следующим показателям:

- Понятный интерфейс, независимость от языка приложения.
- Высокая надежность и простота поддержки тестов.
- Высокая поддержка инструмента.

#### 4.1.3. FAST-анализ

Стадия 1. Выбор объекта FAST-анализа.

В качестве объекта FAST-анализа выступает в данной ВКР выступают программное средство, предназначенное для автоматизации тестирования настольных приложений – AutoIT.

Стадия 2. Описание функций, выполняемых объектом.

Таблица 4.2. Классификация функций, выполняемых разрабатываемым продуктом

Выполняемые функции	Ранг функции		
	Главная	Основная	Вспомогательная
Ф1. Создание скрипта	X		
Ф2. Запуск скрипта	X		
Ф3. Запись результата действий скрипта		X	
Ф4. Возможность создания exe файла		X	
Ф5. Просмотр объектов рабочего стола			X
Ф6. Возможность проверки кода на корректность			X

Анализируя таблицу 4.2. можно сделать вывод, что главными функциями программы являются создание скрипта и возможность его запуска.

Стадия 3. Для оценки значимости функций использован метод расстановки приоритетов. В основу данного метода положено расчетно-экспертное определение значимости каждой функции. Для этого была построена матрица смежности функций.

Таблица 4.3. Матрица смежности

	<b>Ф1</b>	<b>Ф2</b>	<b>Ф3</b>	<b>Ф4</b>	<b>Ф5</b>	<b>Ф6</b>
<b>Ф1</b>	=	=	=	>	>	>
<b>Ф2</b>	=	=	=	<	>	>
<b>Ф3</b>	=	=	=	<	>	>
<b>Ф4</b>	<	>	>	=	=	>
<b>Ф5</b>	<	<	<	=	=	>
<b>Ф6</b>	<	<	<	<	<	=

После этого выполнено преобразование матрицы смежности в матрицу количественных соотношений функций.

Таблица 4.4. Матрица количественных соотношений функций

	<b>Ф1</b>	<b>Ф2</b>	<b>Ф3</b>	<b>Ф4</b>	<b>Ф5</b>	<b>Ф6</b>	$\Sigma$	<b>k</b>
<b>Ф1</b>	1	1	1	1,5	1,5	1,5	7,5	0,208
<b>Ф2</b>	1	1	1	0,5	1,5	1,5	6,5	0,181
<b>Ф3</b>	1	1	1	0,5	1,5	1,5	6,5	0,181
<b>Ф4</b>	0,5	1,5	1,5	1	1	1,5	7	0,194
<b>Ф5</b>	0,5	0,5	0,5	1	1	1,5	5	0,139
<b>Ф6</b>	0,5	0,5	0,5	0,5	0,5	1	3,5	0,097
Итого							36	1

Из таблицы видно, что наиболее значимыми функциями являются Ф1, Ф2, Ф3, Ф4. Менее значимыми функциями являются Ф5, Ф6.

Стадия 4. Анализ стоимости функций, выполняемых продуктом.

Таблица 4.5. Определение стоимости функций, выполняемых продуктом

<b>Выполняемые функции</b>	<b>Показатели</b>				
	<b>Трудоемкость функции, нормо-ч</b>	<b>Стоимость функции, руб.</b>	<b>Зарботная плата, руб. (150 руб. час)</b>	<b>Себестоимость, руб.</b>	<b>k</b>
Ф1. Создание скрипта	110	2000	<b>16500</b>	<b>18500</b>	<b>0,226</b>
Ф2. Запуск скрипта	90	1800	<b>13500</b>	<b>15300</b>	<b>0,187</b>

Ф3. Запись результата действий скрипта	90	1700	<b>13500</b>	<b>15200</b>	<b>0,185</b>
Ф4. Возможность создания exe файла	100	1800	<b>15000</b>	<b>16800</b>	<b>0,205</b>
Ф5. Просмотр объектов рабочего стола	70	1500	<b>10500</b>	<b>12000</b>	<b>0,147</b>
Ф6. Возможность проверки кода на корректность	20	1000	<b>3000</b>	<b>4000</b>	<b>0,05</b>
Итого				<b>81800</b>	<b>1</b>

Из таблицы видно, что наиболее трудозатратные получились функции Ф1, Ф2, Ф3, Ф4.

Стадия 5. Построена функционально-стоимостная диаграмма (рис. 4.1).

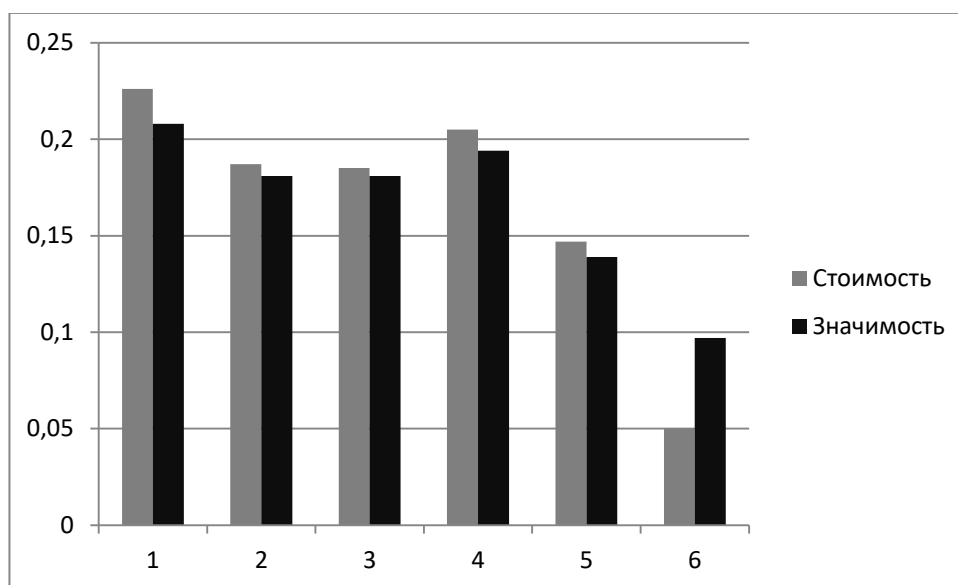


Рисунок 4.1. Функционально-стоимостная диаграмма

Построенная функционально-стоимостная диаграмма позволяет выявить диспропорции между важностью (полезностью) функций и затратами на них.

Анализ приведенной выше ФСД показывает явное наличие рассогласования по функции 6. Данное несоответствие обусловлено различием функций по сложности реализации. Наибольшую стоимость имеют самые значимые функций 1-4.



Стадия 6. Для оптимизации затрат на выполнение проекта можно применить следующие методы:

- Использовать готовые программные компоненты для работы с AutoIT.
- Использовать метод code-review для оптимизации кода программы.
- Использовать паттерны программирования в разработке.

#### 4.1.4. SWOT-анализ

Для проведения комплексного анализа AutoIT была составлена матрица SWOT, которая представлена в таблице 4.6.

Таблица 4.6. SWOT-матрица

	<p><b>Сильные стороны научно-исследовательского проекта:</b></p> <p><i>С1. Бесплатно распространяемое ПО</i></p> <p><i>С2. Возможность интеграции TeamCity;</i></p> <p><i>С4. Уникальность разработанных тестов</i></p>	<p><b>Слабые стороны научно-исследовательского проекта:</b></p> <p><i>Сл1. Отсутствие встроенного интегратора;</i></p> <p><i>Сл2. Возникновение ошибок в процессе разработки</i></p> <p><i>Сл3. Высокая вероятность программирования;</i></p> <p><i>Сл4. Плохая поддержка на русском языке.</i></p>
<p><b>Возможности:</b></p> <p><i>В1. Разработка тестов для всех программ.</i></p> <p><i>В2. Расширение функциональности</i></p> <p><i>В3. Обеспечение хорошей технической поддержки;</i></p>	<p>Возможности позволят усилить сильные стороны продукта.</p> <p>Возможность применения данного языка для автоматизации тестирования всех создаваемых приложений позволит продавать более качественно ПО и, тем самым, завлекать большее число заказчиков.</p>	<p>Расширение функциональности и количества тестов позволит детальнее изучить данный скриптовый язык и не допускать ошибок при написании автотестов.</p>

<p><b>Угрозы:</b></p> <p>У1. Сравнительно медленная скорость разработки тестов</p> <p>У2. <i>Невозможность приспособить тесты для AutoCAD</i></p> <p>У3. <i>Отсутствие опыта программирования на AutoIT</i></p>	<p>Самыми опасными угрозами, которые могут негативно повлиять на разработку, являются отсутствие опыта программирования на данном языке спроса, а также невозможность использования для тестирования AutoCAD. Необходимо детально изучить возможности скриптового языка.</p>	<p>Выявленные самые опасные угрозы могут превратить сильные стороны продукта в слабые стороны. Данные угрозы могут оказать негативное влияние на разработку.</p>
---	--	--

Проведенный SWOT-анализ показал, что к сильным сторонам продукта относится то, что данное ПО является бесплатно распространяемым, а также с его помощью можно внедрить автотесты для любой программы. Возможности позволят усилить сильные стороны продукта.

Возможность применения данного языка для автоматизации тестирования всех создаваемых приложений позволит продавать более качественно ПО и, тем самым, завлекать большее число заказчиков.

Негативно влияние на продукт могут оказать угрозы, самыми опасными из которых являются невозможность приспособить тесты для AutoCAD и отсутствие опыта программирования на AutoIT. Выявленные самые опасные угрозы могут превратить сильные стороны продукта в слабые стороны. Данные угрозы могут оказать негативное влияние на разработку.

#### 4.1.5. Оценка готовности проекта к коммерциализации

В таблице 4.7 приведены показатели о степени проработанности продукта с позиции коммерциализации и компетенциям разработчика научного проекта.

Таблица 4.7. Бланк оценки степени готовности продукта к коммерциализации

№	Наименование	Степень проработанности научного проекта	Уровень имеющихся знаний у разработчика
1	Определен имеющийся научно-технический задел	4	5
2	Определены перспективные направления коммерциализации научно-технического задела	4	5
3	Определены отрасли и технологии (товары, услуги) для предложения на рынке	4	4

№	Наименование	Степень проработанности научного проекта	Уровень имеющихся знаний у разработчика
4	Определена товарная форма научно-технического задания для представления на рынок	4	4
5	Определены авторы и осуществлена охрана их прав	5	5
6	Проведена оценка стоимости интеллектуальной собственности	4	4
7	Проведены маркетинговые исследования рынков сбыта	4	4
8	Разработан бизнес-план коммерциализации научной разработки. Определены пути продвижения научной разработки на рынок	4	5
9	Определены пути продвижения научной разработки на рынок	4	4
10	Разработана стратегия (форма) реализации научной разработки	5	5
11	Проработаны вопросы международного сотрудничества и выхода на зарубежный рынок	3	4
12	Проработаны вопросы использования услуг инфраструктуры поддержки, получения льгот	4	4
13	Проработаны вопросы финансирования коммерциализации научной разработки	5	5
14	Имеется команда для коммерциализации научной разработки	4	4
15	Проработан механизм реализации научного проекта	4	4
<b>Итого баллов</b>		<b>62</b>	<b>65</b>

Полученные результаты показывают, что перспективность коммерциализации является достаточно высокой и перспективной, следовательно, продукт готов к коммерциализации. Однако разработчику продукта необходимо проработать вопросы международного сотрудничества и выхода на зарубежный рынок.

#### 4.2. Инициация проекта

##### Цели и результат проекта

Далее приведена информация о заинтересованных сторонах проекта иерархии целей проекта и критериях достижения целей (табл. 4.8).

Таблица 4.8. Заинтересованные стороны проекта

Заинтересованные стороны проекта	Ожидания заинтересованных сторон
ООО «ТомскАСУпроект»	Ожидают наличие готовых автотестов, с помощью которых можно будет осуществлять тестирования программного продукта для увеличения скорости тестирования, экономии ресурсов, а также для выпуска более качественных продуктов
Разработчик и научный руководитель	Получение работающей программы, для демонстрации полученных результатов, успешная защита магистерской диссертации.

Таблица 4.9. Цель и результат проекта

Цель проекта:	Разработка и внедрение технологии автоматизированного тестирования программного обеспечения
---------------	---

<b>Ожидаемые результаты проекта:</b>	Внедрение разработанной технологии на предприятие ООО «ТомскАСУпроект» Успешная защита магистерской диссертации Демонстрация результатов разработки на конференциях
<b>Критерии приемки результата проекта:</b>	100% покрытие требований тестами, успешные результаты прохождения автотестов, формирование отчетов об ошибках.
<b>Требования к результату проекта:</b>	<b>Требование:</b>
	1. Должны быть разработаны тестовые сценарии 2. Должны быть разработанные автоматические тесты.

## Организационная структура проекта

Таблица 4.10. Рабочая группа проекта

№	ФИО, место работы, должность	Роль в проекте	Функции	Трудо-затраты, час.
1	Федорченко Алена Вадимовна, ТПУ, магистрант	Исполнитель проекта	Аналитика и реализация проекта	576
2	Кудинов Антон Викторович, ТПУ, доцент каф. ИСТ	Руководитель проекта от ТПУ	Экспертная консультационная помощь, координация действий в процессе выполнения проекта	5
3	Валявин Михаил Борисович	Руководитель проекта от предприятия ООО «ТомскАСУпроект»	Экспертная консультационная помощь, координация действий в процессе выполнения проекта	96
<b>Итого:</b>				<b>677</b>

Таблица 4.11. Ограничения и допущения проекта

Фактор	Ограничения/допущения
Бюджет проекта	281 539 руб.
Источник финансирования	Внутренние средства ООО «ТомскАСУпроект»
Сроки проекта	01.03.2016 – 02.06.2017

### 4.3. Планирование управления научно-техническим проектом

#### 4.3.1. Контрольные события проекта

Определены ключевые события проекта по разработке продукта (табл. 4.12).

Таблица 4.12. Контрольные события проекта

№	Контрольное событие	Даты	Результат
1	Составление и согласование задания	01.03.2017 – 05.03.2017	Согласованное задание с руководителем
2	Анализ существующих программных средств для тестирования	06.03.2017 – 20.03.2017	Сравнительная таблица программных средств
3	Составление тестового плана	21.03.2017 – 30.03.2017	Документ с описанием структуры тестового плана
4	Создание тестовых сценариев	01.04.2017 – 15.04.2017	Документ с подробными шагами тестового сценария
5	Программная реализация тестовых скриптов	16.04.2017 – 15.05.2017	Реализованы скрипты с помощью которых можно тестировать программный модуль
6	Финальное тестирование и отладка	16.05.2017 –	Рабочая версия продукта

№	Контрольное событие	Даты	Результат
		20.05.2017	
7	Документирование	21.05.2017 – 02.06.2017	Написание документа с тестовыми сценариями

### 4.3.2. План проекта

Таблица 4.13. Календарный план проекта

Код работы (из ИСР)	Название	Длительность, дни	Дата начала работ	Дата окончания работ	Состав участников (ФИО ответственных исполнителей)
1	Составление и согласование задания	3	01.03.2017	03.03.2017	Федорченко А.В. Валявин М.Б. Кудинов А.В.
2	Анализ существующих программных средств для тестирования	17	04.03.2017	20.03.2017	Федорченко А.В.
3	Составление тестового плана	12	21.03.2017	31.03.2017	Федорченко А.В. Валявин М.Б.
4	Создание тестовых сценариев	13	01.04.2017	15.04.2017	Федорченко А.В. Валявин М.Б.
5	Программная реализация тестовых скриптов	27	16.04.2017	15.05.2017	Федорченко А.В.
6	Финальное тестирование и отладка	9	16.05.2017	20.05.2017	Федорченко А.В.
7	Документирование	12	21.05.2017	02.06.2017	Валявин М.Б. Федорченко А.В.
<b>И т о г о:</b>		92	01.03.2017	02.06.2017	

### Диаграмма Ганта

Для построения диаграммы Ганта необходимо было заполнить таблицу 4.14. Для заполнения используем опытно-статистический метод, который реализуется двумя способами:

- аналоговый;
- вероятностный.

Для определения ожидаемого значения продолжительности работ  $t_{ож}$  применяется вероятностный метод – метод двух оценок  $t_{min}$  и  $t_{max}$ .

$$t_{ож} = \frac{3t_{min} + 2t_{max}}{5}, \quad (4.1)$$

где  $t_{min}$  – минимальная трудоемкость работ, чел/дн.;

$t_{max}$  – максимальная трудоемкость работ, чел/дн.

Для выполнения работ, перечисленных в таблице 4.13, требуются следующие специалисты:

- исполнитель (магистрант) – И;
- научный руководитель от университета – НРУ;
- научный руководитель от предприятия – НРП.

Рассчитаем длительность этапов в рабочих днях, а затем переведем в календарные дни.

$$T_{PD} = \frac{t_{ож}}{k_{вн}} * k_{д}, \quad (4.2)$$

где  $t_{ож}$  – трудоемкость работы, чел/дн.;

$k_{вн}$  – коэффициент выполнения работ ( $k_{вн} = 1$ );

$k_{д}$  – коэффициент, учитывающий дополнительное время на компенсации и согласование работ ( $k_{д} = 1.2$ ).

Расчет продолжительности этапа в календарных днях ведется по формуле:

$$T_{КД} = T_{PD} * T_{к}, \quad (4.3)$$

где  $T_{PD}$  – продолжительность выполнения этапа в рабочих днях;

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

$T_{к}$  – коэффициент календарности.

Коэффициент календарности рассчитывается по формуле:

$$T_{к} = \frac{93}{93 - 28} = 1,43, \quad (4.4)$$

где  $T_{КАЛ}$  – календарные дни ( $T_{КАЛ} = 93$ );

$T_{ВД}$  – выходные дни ( $T_{ВД} = 24$ );

$T_{ПД}$  – праздничные дни ( $T_{ПД} = 4$ ).

Таблица 4.24. Временные показатели для построения диаграммы Ганта

Этап	Исполнители	Продолжительность работ, дни									Длительность работ, чел/дн.					
		$t_{min}$			$t_{max}$			$t_{ож}$			$T_{рд}$			$T_{кд}$		
		<i>НРП</i>	<i>НРУ</i>	<i>И</i>	<i>НРП</i>	<i>НРУ</i>	<i>И</i>	<i>НРП</i>	<i>НРУ</i>	<i>И</i>	<i>НРП</i>	<i>НР</i>	<i>НРУ</i>	<i>НРП</i>	<i>НРУ</i>	<i>И</i>
Составление и согласование задания	НРП, И, НРУ	1	0,1	1	2	0,8	3	1,4	0,38	1,8	1,68	0,456	2,16	2,40	0,65	3,09
Анализ существующих программных средств для тестирования	И	-	-	4	-	-	10	-	-	6,4	-	-	7,68	-	-	10,98
Составление тестового плана	НРП, И	1	-	2	4	-	10	2,2	-	5,2	2,64	-	6,24	3,78	-	8,92
Создание тестовых сценариев	НРП, И	1	-	3	3	-	9	1,8	-	5,4	2,16	-	6,48	3,09	-	9,27
Программная реализация тестовых скриптов	И		-	10		-	16		-	12,4	-	-	14,88	-	-	21,28
Финальное тестирование и отладка	И		-	4		-	8		-	5,6	-	-	6,72	-	-	9,61
Документирование	НРП, И	1		3	2		8	1,4	-	5	1,68	-	6	2,40	-	8,58
<b>Итого:</b>		4	0,1	27	11	0,8	64	6,8	0,38	41,8	8,16	0,456	50,16	11,67	0,65	71,73

На основании данных, приведённых в таблице 4.13 и 4.14, был сформирован календарный план-график проведения ВКР, представленный в таблице 4.15.

Таблица 4.15. Диаграмма Ганта

№Э	Т <sub>кд</sub> НРУ	Т <sub>кд</sub> НРП	Т <sub>кд</sub> И	Март			Апрель			Май			Июнь
				1	2	3	1	2	3	1	2	3	1
1	2,40	2,40	3,09	■									
2	-	-	10,98		■	■							
3	-	3,78	8,92			■	■						
4	-	3,09	9,27				■	■					
5	-	-	21,28					■	■	■			
6	-	-	9,61								■	■	
7	-	2,40	8,58									■	■

В таблице 4.15 обозначено:

- НРУ – белый прямоугольник;
- НРП – серый прямоугольник;
- И – чёрный прямоугольник;

Из таблицы 4.15 видно, что наиболее длительным этапом является программная реализация тестовых скриптов

#### 4.3.3. Бюджет научного исследования

При планировании бюджета НТИ должно быть обеспечено полное и достоверное отражение всех видов расходов, связанных с его выполнением. В процессе формирования бюджета НТИ используется следующая группировка затрат по статьям:

1. Материальные затраты НТИ;
2. Затраты на специальное оборудование для научных работ;
3. Основная заработная плата исполнителей темы;
4. Дополнительная заработная плата исполнителей темы;
5. Отчисления во внебюджетные фонды;
6. Затраты на научные и производственные командировки;



7. Оплата работ, выполняемых сторонними организациями и предприятиями;

8. Прочие прямые расходы;

9. Накладные расходы.

#### 4.3.3.1. Расчёт материальных затрат НИИ

В таблице 4.16 приведены материальные затраты, которые необходимы для разработки информационной системы.

Таблица 4.16. Материальные затраты

Наименование	Количество	Цена за ед., руб.
Персональный компьютер	1	30 750
Принтер	1	9 200
<b>Итого</b>	<b>2</b>	<b>39 950</b>

#### 4.3.3.2. Расчёт основной и дополнительной заработной платы исполнителей

Вычислим месячный должностной оклад работника.

Для этого определен баланс рабочего времени за 2017 год в таблице 4.17.

Таблица 4.17. Баланс рабочего времени

Показатели рабочего времени	Руководитель 1 (НРУ)	Руководитель 2 (НРП)	Студент (И)
Календарное число дней	93	93	93
Количество нерабочих дней: - выходные дни - праздничные дни	28	28	28
Потери рабочего времени: - отпуск - невыходы по болезни	3	3	2
Действительный фонд рабочего времени (часов)	1973	1973	1973

В таблице 4.18 приведен базовый оклад НРП, НРУ и И.

Базовый оклад сотрудников предприятия взят из их трудовых договоров. Согласно документам ТПУ, оклад руководителя от университета равен 26 300 рублей, т.к. он является доцентом и кандидатом технических наук.

Таблица 4.18 Базовый оклад

Специалист	Базовый оклад
НРП	48629,53
НРУ	26300
Исполнитель	15000

Месячный должностной оклад работника:

$$Z_m = Z_b \cdot (k_{пр} + k_d) \cdot k_p \quad (4.5)$$

где  $Z_b$  – базовый оклад, руб.;

$k_{пр}$  – премиальный коэффициент, (определяется Положением об оплате труда);

$k_d$  – коэффициент доплат и надбавок (в НИИ и на промышленных предприятиях – за расширение сфер обслуживания, за профессиональное мастерство, за вредные условия: определяется Положением об оплате труда);

$k_p$  – районный коэффициент, равный 1.3 (для Томска).

Таблица 4.19. Месячный должностной оклад

Специалист	Базовый оклад	Районный коэффициент	Итого
НРП	48629,53	1,3	63218,39
НРУ	26300		34190
И	15000		19500

Основная заработная плата ( $Z_{осн}$ ) руководителя от предприятия (при наличии руководителя от предприятия) рассчитывается по следующей формуле:

$$Z_{осн} = Z_{дн} \cdot T_{раб} \quad (4.6)$$

Где  $Z_{осн}$  – основная заработная плата одного работника;

$T_p$  – продолжительность работ, выполняемых научно-техническим работником, раб. дн.;

$Z_{дн}$  – среднедневная заработная плата работника, руб.

Среднедневная заработная плата рассчитывается по формуле:

$$Z_{дн} = \frac{Z_m \cdot \bar{M}}{F_d} \quad (4.7)$$

где  $Z_m$  – месячный должностной оклад работника, руб.;

$M$  – количество месяцев работы без отпуска в течение года:

- при отпуске в 24 раб. дня  $M = 11,2$  месяца, 5-дневная неделя;
- при отпуске в 48 раб. дней  $M = 10,4$  месяца, 6-дневная неделя;

$F_d$  – действительный годовой фонд рабочего времени научно-технического персонала, раб. Дн.

Среднедневная зарплата (Руководитель от предприятия) =  $(63218,39 * 11,2 * 8) / 1973 = 2870,88$

Среднедневная зарплата (Руководитель от ТПУ) =  $(34190 * 10,4 * 8) / 1973 = 1441,76$

Среднедневная зарплата (Исполнитель) =  $(19500 * 11,2 * 8) / 1973 = 885,55$

Таблица 4.20. Расчёт основной заработной платы

Исполнитель	Месячный Оклад, руб.	Среднедневная заработная плата, руб./дн.	Трудоемкость, раб. дн.	Основная заработная плата, руб.
НРП	63218,39	2870,88	11,67	33503,17
НРУ	34190	1441,76	0,65	937,14
И	19500	885,55	71,73	63520,50
<b>ИТОГО</b>				<b>97960,82</b>

В таблице 4.21 приведен расчёт дополнительной заработной платы для научных руководителей и студента, которые участвуют в научном проекте. Расчет дополнительной заработной платы ведется по следующей формуле:

$$Z_{\text{доп}} = k_{\text{доп}} \cdot Z_{\text{осн}} \quad (4.8)$$

где  $k_{\text{доп}}$  – коэффициент дополнительной заработной платы.

Принимается равным 0,1.

Таблица 4.31. Расчёт дополнительной заработной платы

Исполнитель	Основная заработная плата, руб.	Коэффициент дополнительной заработной платы	Дополнительная заработная плата, руб.
НРП	33503,17	0,1	3350,32
НРУ	937,14		93,71
И	63520,50		6352,05
<b>ИТОГО</b>			<b>9796,08</b>

#### 4.3.3.3. Отчисления во внебюджетные фонды

В данной статье расходов отражаются обязательные отчисления по установленным законодательством Российской Федерации нормам органам государственного социального страхования, пенсионного фонда и медицинского страхования от затрат на оплату труда работников.

Величина отчислений во внебюджетные фонды определяется исходя из следующей формулы:

$$З_{\text{внеб}} = k_{\text{внеб}} \cdot (З_{\text{осн}} + З_{\text{доп}}), \quad (4.9)$$

где  $k_{\text{внеб}}$  – коэффициент отчислений на уплату во внебюджетные фонды (30%).

В таблице 4.22 приведен расчёт заработной платы для научного руководителя и исполнителя, которые участвуют в научном проекте.

Таблица 4.42. Расчёт отчислений во внебюджетные фонды

Исполнитель	Осн. ЗП	Доп. ЗП	к отчислений во внебюд. фонды	Внебюд. отчисления
НРП	33503,17	3350,32	0,3	11056,05
НРУ	937,14	93,71		309,26
И	63520,50	6352,05		20961,77
			<b>ИТОГО</b>	<b>32327,07</b>

#### 4.3.3.4. Накладные и прочие расходы, бюджет затрат НИИ

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов: затраты управленческого персонала (руководитель может непосредственно не принимать участие в разработке продукта, а только руководить процессом, тогда затраты на его заработную плату будут накладными расходами), аренда помещения, амортизация, услуги связи. При этом данные расходы берутся с учётом длительности выполнения проекта (3 месяца) Стоит отметить, что в данной работе руководители принимают непосредственное участие, потому его заработную плату нельзя отнести к накладным расходам.

Их величина определяется по следующей формуле:

$$C_{\text{накл}} = k_{\text{накл}} \cdot (Z_{\text{осн}} + Z_{\text{доп}}) \quad (4.10)$$

где  $k_{\text{накл}}$  – коэффициент накладных расходов.

Коэффициент принимается равным 0,8.

Тогда сумма накладных расходов составит:

$$(97960,82 + 9796,08) \cdot 0,8 = 86205,52$$

К прочим затратам относятся канцелярия, краска в принтер, буфет (вода, чай + кофе, сахар). Результаты расчета представлены в табл. 4.23.

Таблица 4.53. Расчёт прочих расходов

Статья расходов	Расходы за месяц (руб.)	Расходы за 3 месяца (руб.)
Канцелярия	500	6000
Краска в принтер	100	300
Буфет	3000	9000
Зпроч		15300

В таблице 4.24 приведен расчёт бюджета затрат на научно-исследовательский проект. Бюджет затрат НТИ равен сумме статей 1-6.

Таблица 4.24. Расчёт бюджета затрат на НТИ

Наименование статьи	Сумма, руб.
1. Материальные затраты НТИ	39 950
2. Затраты по основной заработной плате исполнителей темы	97960,82
3. Затраты по дополнительной заработной плате исполнителей темы	9796,08
4. Отчисления во внебюджетные фонды	32327,07
5. Накладные расходы	86205,52
6. Прочие затраты	15300
7. Бюджет затрат НТИ	281 539

#### 4.3.4. Реестр рисков проекта

Проведена идентификация рисков, которая приведена в таблице 4.25.

Таблица 4.25. Реестр рисков

№	Риск	Потенциальное воздействие	Вероятность наступления (1-5)	Влияние риска (1-5)	Уровень риска	Способы смягчения	Условия наступления
1	Неполный список требований	Недостаток функциональности	1	3	низкий	Тщательная проработка списка требований	Недостаточно уделено внимания к списку требований
2	Неправильно спроектированный скрипт	Неверная работа скрипта.	4	5	высокий	Привлечение экспертов разработки приложений	Самостоятельное проектирование скрипта без

						на VBA	привлечения экспертов
3	Отсутствие спроса на скрипты	Нет необходимости автотестах	1	5	низкий	Тщательный анализ предметной области. Выступления с докладом на отраслевых конференциях для рекламы продукта.	Недостаточно уделено внимания к анализу предметной области. Недостаточная осведомленность потенциальных заказчиков
4	Несоответствие плану	Отставание по срокам	2	3	низкий	Проработка плана на начальном этапе	Недостаточно уделено внимания к планированию

#### 4.4. Оценка сравнительной эффективности исследования

Определение эффективности происходит на основе расчёта интегрального показателя эффективности научного исследования. Для этого были рассчитаны следующие показатели: интегральный финансовый показатель, интегральный показатель ресурсоэффективности, показатель эффективности вариантов исполнения разработки.

*Интегральный показатель финансовой эффективности* научного исследования получают в ходе оценки бюджета затрат трех (или более) вариантов исполнения научного исследования. Для этого наибольший интегральный показатель реализации технической задачи принимается за базу расчета (как знаменатель), с которым соотносятся финансовые значения по всем вариантам исполнения.

*Интегральный финансовый показатель* разработки определяется как:

$$I_{\text{финр}}^{\text{исп.}i} = \frac{\Phi_{ri}}{\Phi_{\text{max}}}, \quad (4.11)$$

где  $I_{\text{финр}}^{\text{исп.}i}$  – интегральный финансовый показатель разработки;

$\Phi_{ri}$  – стоимость  $i$ -го варианта исполнения;

$\Phi_{\text{max}}$  – максимальная стоимость исполнения научно-исследовательского проекта (в т.ч. аналоги).

Для оценки финансовой эффективности внедрения автотестов взяты затраты на внедрение ручных и автотестов для данной программы. (табл. 4.26).

Таблица 4.26. Стоимости поддержки тестов для сравнения

Наименование проекта	Стоимость лицензии
1. Внедрение и поддержка автотестов	267 937,5
2. Внедрение и поддержка ручных тестов	539 662,5

$$ИП1 (1) = 267\,937,5/539\,662,5 = 0,49.$$

$$ИП2 (2) = 267\,937,5/267\,937,5 = 1.$$

Полученная величина интегрального финансового показателя разработки отражает соответствующее численное увеличение бюджета затрат разработки в разгах, либо соответствующее численное удешевление стоимости разработки в разгах. Самый низкий интегральный финансовый показатель получен при использовании автотестов.

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить следующим образом:

$$I_{pi} = \sum a_i \cdot b_i, \quad (4.12)$$

где  $I_{pi}$  – интегральный показатель ресурсоэффективности для  $i$ -го варианта исполнения разработки.

Интегральный показатель эффективности вариантов исполнения разработки ( $I_{исп.i}$ ) определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{исп.1} = \frac{I_{p-исп1}}{I_{финр}}, \quad I_{исп.2} = \frac{I_{p-исп2}}{I_{финр}} \quad (4.13)$$

Сравнительная эффективность проекта ( $\mathcal{E}_{cp}$ ):

$$\mathcal{E}_{cp} = \frac{I_{исп.1}}{I_{исп.2}} \quad (4.14)$$

Для расчёта интегрального показателя ресурсоэффективности была составлена таблица 4.27.

Таблица 4.27. Сравнительная оценка характеристик аналогов продукта

Критерии \ Объект исследования	Весовой коэффициент параметра	1 Автотесты	2 Ручное тестирование
1. Способствует росту производительности труда пользователя	0,3	5	2
2. Удобство в использовании	0,1	4	4
3. Надежность	0,15	4	2
4. Безопасность	0,15	4	1
5. Простота создания	0,05	1	5
6. Качество тестов	0,15	4	1
7. Функциональная мощность (предоставляемые возможности)	0,1	4	2
Итого	1	26	18

$$I_{p-ucn1} = 0.3*5 + 0.1*4 + 0.15*4 + 0.15*4 + 0.05*1 + 0.15*4 + 0.1*4 = 4.15;$$

$$I_{p-ucn2} = 0.3*2 + 0.1*4 + 0.15*2 + 0.15*1 + 0.05*5 + 0.15*1 + 0.1*2 = 2.05;$$

$$ИПЭ (1) = 4.15/0.49 = 8.47; ИПЭ (2) = 2.05/1 = 2.05.$$

Результаты расчётов интегральных показателей приведены в таблице 4.28.

Таблица 4.28. Сравнительная эффективность разработки

№ п/п	Показатели	1 Автотесты	2 Ручное тестирование
1	Интегральный финансовый показатель разработки	0.49	1
2	Интегральный показатель ресурсоэффективности разработки	4,15	2,05
3	Интегральный показатель эффективности	8,47	2,05
4	Сравнительная эффективность вариантов исполнения	3.76	0,466

Таким образом, из таблицы видно, что разрабатываемые являются более эффективным вариантом решения поставленной задачи с позиции финансовой и ресурсной эффективности, даже не смотря на то, что они уступает по затрачиваемому времени разработки ручным тестам.



**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»**

Студенту:

<b>Группа</b>	<b>ФИО</b>
8ИМ5А	Федорченко Алена Вадимовна

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

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

1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Объектом исследования являются методологии и программные средства, предназначенные для автоматизации тестирования настольных приложений. Цель исследования – разработка и внедрение технологии автоматизированного тестирования программного обеспечения, включающей в себя разработку автоматических тестов. Результаты будут применяться непосредственно сотрудниками предприятия, для которого были созданы автоматические тесты.
--	--

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

<b>1. Производственная безопасность</b> 1.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения 1.2. Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения	1.1 В качестве вредных факторов выделены: отклонения показателей микроклимата, недостаточная освещенность рабочей зоны, монотонный режим работы. 1.2 В качестве опасных факторов выделены:пожар, поражений электрическим током.
<b>2. Экологическая безопасность</b> 2.1. Анализ воздействия объекта на окружающую среду; 2.2. Разработать решения по обеспечению экологической безопасности со ссылками на НТД по охране окружающей среды.	2.1. Влияние объекта исследования на окружающую среду: влияние бумажных и пластиковых отходов на окружающую среду; 2.2. Мероприятия по защите окружающей среды: утилизация бумажных и пластиковых отходов
<b>3. Безопасность в чрезвычайных ситуациях</b> 3.1. Перечень возможных ЧС при разработке и эксплуатации проектируемого решения; 3.2. Выбор наиболее типичной ЧС; 3.3. Разработка действий в результате возникшей ЧС и мер по ликвидации её последствий.	3.1 Перечень возможных чрезвычайных ситуаций. 3.2 Типичные чрезвычайные ситуации:пожар, кибертерроризм. 3.3 Мероприятия по предотвращению наиболее типичных ЧС – пожара и кибертерроризма.
<b>4. Правовые и организационные вопросы обеспечения безопасности</b> 4.1. Специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; 4.2. Организационные мероприятия при компоновке рабочей зоны.	4.1 Описание правовых норм для работ, связанных с работой за ПЭВМ согласно Трудовому кодексу Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 30.12.2015). 4.2 Организационные мероприятия при компоновке рабочей зоны. 4.3. Влияние разработанных автоматических тестов на качество ПО, на трудовые и временные ресурсы, на вероятность устранения ошибок

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

**Задание выдал консультант:**

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
Ассистент кафедры ЭБЖ	Акулов Петр Анатольевич			

**Задание принял к исполнению студент:**

<b>Группа</b>	<b>ФИО</b>	<b>Подпись</b>	<b>Дата</b>
8ИМ5А	Федорченко Алена Вадимовна		

## **5. СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ**

### **Введение**

В рамках выпускной квалификационной работы разработана и внедрена технология автоматизированного тестирования настольного программного приложения «Модуль формирования спецификаций «Specbook» с помощью скриптового языка AutoIT. Так как разработка велась для предприятия ООО «ТомскАСУпроект», то результаты работы смогут применяться всеми сотрудниками данной организации для тестирования программного модуля «Specbook».

Данная работа была выполнена с использованием вычислительной техники: персонального компьютера (ПК), системы ввода-вывода и периферии. В разделе будут рассмотрены вопросы производственной и пожарной безопасности, требования к защите окружающей среды при выполнении и оформлении данного проекта.

Разработка программного обеспечения подразумевает соблюдение определенных норм, таких как санитарные нормы, меры пожарной безопасности, снижение влияния вредных факторов (при их наличии) и прочее. Для такого рода работ особенно важным является соблюдение норм освещенности рабочего места. Помимо описанных факторов человек подвергается психологическим нагрузкам, стрессу, связанным с необходимостью восприятия, обработки и анализа большого объема информации.

### **5.1. Производственная безопасность**

Вредные производственные факторы – это факторы среды и трудового процесса, воздействие которых на работающего при определенных условиях может вызвать профессиональное заболевание, временное или стойкое снижение работоспособности, повысить частоту соматических и инфекционных заболеваний, привести к нарушению здоровья потомства. В

зависимости от уровня концентрации и продолжительности воздействия вредный производственный фактор может стать опасным [12].

Опасные производственные факторы – это факторы, воздействие которых может привести к травме, отравлению, внезапному резкому ухудшению здоровья, или смерти человека [13].

Опасные и вредные производственные факторы подразделяются по природе действия на физические, химические, биологические и психофизиологические [13]. Для данной работы целесообразно рассмотреть физические и психофизические вредные и опасные факторы производства, характерные как для рабочей зоны программиста, так и для рабочей зоны пользователя. Выявленные факторы представлены в таблице 5.5.1.

Таблица 5.1 – Вредные и опасные производственные факторы при выполнении работ за ПЭВМ [13].

Источник фактора, наименование видов работ	Факторы (по ГОСТ 12.0.003-74)		Нормативные документы
	Вредные	Опасные	
1) Работа за ПЭВМ	1) Отклонение показателей микроклимата. 2) Недостаточная освещенность рабочей зоны; 3) Монотонный режим работы	1) Опасность поражения электрическим током; 2) Опасность возникновения пожара.	1) СанПиН 2.2.4.548-96; 2) СП 52.13330.2011; 3) СанПиН 2.2.2/2.4.1340-03

### 5.1.1. Вредные производственные факторы

#### 5.1.1.1. Отклонение показателей микроклимата

Микроклимат – это один из фундаментальных факторов, оказывающих влияние на работоспособность и здоровье человека. Неблагоприятное сочетание факторов обычно приводит к нарушению терморегуляции организма. Микроклимат характеризуется температурой воздуха, относительной влажностью воздуха, скоростью движения воздуха и интенсивностью теплового излучения от нагретых поверхностей [16].

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».

При повышенной температуре кровеносные сосуды расширяются, это приводит к повышению притока крови к поверхности тела и повышает теплоотдачу в окружающую среду. При пониженной температуре кровеносные сосуды сужаются и приток крови к поверхности тела замедляется, это приводит к уменьшению теплоотдачи. Влажность воздуха оказывает влияние на терморегуляцию организма: при высокой влажности (более 85%) терморегуляция снижается из-за снижения испарения пота, а при низкой влажности (менее 20%) происходит пересыхание слизистой оболочки дыхательных путей.

Санитарные нормы устанавливают оптимальные значения величин показателей микроклимата рабочих мест для различных категорий работ в теплый и холодный периоды года. Холодный период года – среднесуточная температура воздуха 10 °С и ниже, теплый период года – среднесуточная температура воздуха выше 10 °С.

На основе интенсивности общих энергозатрат организма в ккал/ч (Вт) вид данной работы (работа оператора ПК или программиста) относится к категории Ia, который характеризуется интенсивностью энергозатрат до 120 ккал/ч, сидячим состоянием и незначительным физическим напряжением [16]. Оптимальные параметры микроклимата в офисных помещениях приведены в таблице 5.5.2.

В нашем случае работа проводилась в холодное время года. Температура в офисе колебалась в пределах 23-25 градусов, относительная влажность в пределах 50-60.

Таблица 5.2 – Оптимальные значения характеристик микроклимата

Период года	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность, %	Скорость движения воздуха, м/с
Холодный	22–24	21–25	60–40	0,1
Теплый	23–25	22–26	60–40	0,1

По данным таблицы можно сделать вывод, что микроклимат рабочей зоны соответствовал оптимальным значениям.

Для поддержания характеристик микроклимата в помещении необходимо правильно организовать вентиляцию и кондиционирование воздуха, и отопление помещений. Вентиляция может осуществляться естественным и механическим путём. В зимнее время в помещении необходимо предусмотреть систему отопления. Также возможно использование ионизации воздуха, что позволит увеличить в воздухе количество аэроионов. Также, в некоторых случаях, целесообразно обеспечить питьевое водоснабжение. В помещениях для работы с ПК должна производиться ежедневная влажная уборка, а также систематическое проветривание после каждого часа работы [15].

#### 5.1.1.2. Недостаточная освещенность рабочей зоны

1. Для обеспечения комфортной и безопасной работы с ПЭВМ, а также с различными документами на рабочем месте необходимо оптимальное освещение. Недостаточность освещения снижает производительность труда, увеличивает утомляемость и количество допускаемых ошибок, а также может привести к появлению профессиональных болезней зрения. Освещение создается естественными и искусственными источниками. Отсутствие какого-либо из источников недопустимо.

2. Системы освещения должны обладать следующими характеристиками:

- соответствие уровня освещенности рабочих мест характеру выполняемой зрительной работы;
- достаточно равномерное распределение яркости на рабочих поверхностях и в окружающем пространстве;
- отсутствие резких теней, прямой и отраженной блескости;
- постоянство освещенности во времени;

- оптимальная направленность излучаемого осветительными приборами светового потока;

- долговечность, экономичность, электробезопасность и пожаробезопасность, эстетичность, удобство и простота эксплуатации.

3. Для обеспечения освещенности рабочего места применяют естественное уличное освещение и только при его нехватке подключаются внутренние системы освещения. В помещениях, предназначенных для работы с ПЭВМ, рекомендуется использование системы общего освещения.

4. Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300-500 лк. Освещение не должно создавать бликов на поверхности экрана. Освещенность поверхности экрана не должна быть более 300 лк.

5. Следует ограничивать неравномерность распределения яркости в поле зрения пользователя ПЭВМ, при этом соотношение яркости между рабочими поверхностями не должно превышать 3:1-5:1, а между рабочими поверхностями и поверхностями стен и оборудования 10:1.

6. В качестве источников света при искусственном освещении следует применять преимущественно люминесцентные лампы типа ЛБ и компактные люминесцентные лампы (КЛЛ). При устройстве отраженного освещения в производственных и административно-общественных помещениях допускается применение металлогалогенных ламп. В светильниках местного освещения допускается применение ламп накаливания, в том числе галогенные.

7. Если экран дисплея расположен к окну, необходимы специальные экранирующие устройства (светорассеивающие шторы, регулируемые жалюзи, солнцезащитная пленка с металлизированным покрытием).

Разряд зрительных работ программиста и оператора ПЭВМ относится к разряду III, подразряду г (работы высокой точности). В таблице 5.3

представлены нормативные показатели искусственного освещения при работах заданной точности.

Таблица 5.3 – Требования к освещению помещений для операторов ПЭВМ [20]

Характеристика зрительной работы	Наименьший или эквивалентный размер объекта различения, мм	Разряд зрительной работы	Подразряд зрительной работы	Контраст объекта с фоном	Характеристика фона	Искусственное освещение		
						Освещённость, лк		
						При системе комбинированного освещения		При системе общего освещения
всего	В том числе							
Высокой точности	От 0,3 до 0,5	III	г	Средний, большой	Светлый, средний	400	200	200

Для создания и поддержания благоприятных условий освещения для операторов ПЭВМ, их рабочие места должны соответствовать санитарно-эпидемиологическим правилам СанПиН 2.2.2/2.4.1340-03. Рабочее помещение должно иметь естественное и искусственное освещение, соответствующее показателям, представленным в таблице 5.3. Для рассеивания естественного освещения следует использовать жалюзи на окнах рабочих помещений. В качестве источников искусственного освещения должны быть использованы люминесцентные лампы, лампы накаливания – для местного освещения [20].

### 5.1.1.3. Монотонный режим работы

При работе с ПЭВМ основным фактором, влияющим на нервную систему программиста или пользователя, является огромное количество информации, которое он должен воспринимать. Это является сложной задачей, которая очень сильно влияет на сознание и психофизическое состояние из-за монотонности работы. Поэтому меры, позволяющие снизить воздействие этого вредного производственного фактора, которые регулируются СанПиН 2.2.2/2.4.1340-03, являются важными в работе оператора ПЭВМ. Они позволяют увеличить производительность труда и предотвратить появление профессиональных болезней.

Организация работы с ПЭВМ осуществляется в зависимости от вида и категории трудовой деятельности. Виды трудовой деятельности разделяются на 3 группы: группа А – работа по считыванию информации с экрана с предварительным запросом; группа Б – работа по вводу информации; группа В – творческая работа в режиме диалога с ПЭВМ. Работа программиста-разработчика рассматриваемой в данной работе к группе В. Категории трудовой деятельности различаются по степени тяжести выполняемых работ. Для снижения воздействия рассматриваемого вредного фактора предусмотрены регламентированные перерывы для каждой группы работ – таблица 5.4.

Таблица 5.4 – Суммарное время регламентированных перерывов в зависимости от продолжительности и категории работы с ПЭВМ [18]

Категория работы с ПЭВМ	Уровень нагрузки за рабочую смену при видах работ с ПЭВМ			Суммарное время регламентированных перерывов, мин.	
	группа А, количество знаков	группа Б, количество знаков	группа В, ч	при 8-часовой смене	при 12-часовой смене
I	до 20 000	до 15 000	до 2	50	80
II	до 40 000	до 30 000	до 4	70	110
III	до 60 000	до 40 000	до 6	90	140

Согласно ТОИ Р-45-084-01, продолжительность непрерывной работы с компьютером без регламентированного перерыва не должна превышать 2-х часов. А во время регламентированных 15-минутных перерывов с целью снижения нервно-эмоционального напряжения, утомления зрительного анализатора, устранения влияния гиподинамии и гипокинезии, предотвращения развития познотонического утомления необходимо выполнять комплексы упражнений.

## 5.1.2. Опасные производственные факторы

### 5.2.2.1. Электробезопасность

Поражение электрическим током является опасным производственным фактором и, поскольку оператор ПЭВМ имеет дело с электрооборудованием, то вопросам электробезопасности на его рабочем



месте должно уделяться много внимания. Нормы электробезопасности на рабочем месте регламентируются 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.

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

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

Не следует работать с ПЭВМ в условиях повышенной влажности (относительная влажность воздуха длительно превышает 75%), высокой температуры (более 35°C), наличии токопроводящей пыли, токопроводящих полов и возможности одновременного соприкосновения к имеющим соединение с землёй металлическим элементам и металлическим корпусом электрооборудования.

Основным организационным мероприятием по обеспечению безопасности является инструктаж и обучение безопасным методам труда, а также проверка знаний правил безопасности и инструкций в соответствии с занимаемой должностью применительно к выполняемой работе.

Специфическая опасность электроустановок: токоведущие проводники, корпуса стоек ЭВМ и прочего оборудования, оказавшегося под напряжением в результате повреждения (пробоя) изоляции, не подают каких-либо сигналов, которые предупреждали бы человека об опасности. Это напряжение опасно для жизни, поэтому обязательны следующие меры предосторожности:

- перед началом работы нужно убедиться, что выключатели и розетка закреплены и не имеют оголённых токоведущих частей;

- при обнаружении неисправности оборудования и приборов необходимо, не делая никаких самостоятельных исправлений, сообщить человеку, ответственному за оборудование;
- запрещается загромождать рабочее место лишними предметами;
- при возникновении несчастного случая следует соблюдать инструкции техники безопасности и, соответственно проводить обучение сотрудников технике безопасности [18, 21].

#### 5.1.1.2. Опасность возникновения пожара

Возникновение пожара является опасным производственным фактором, т.к. пожар на предприятии наносит большой материальный ущерб, а также часто сопровождается травмами и несчастными случаями. Регулирование пожаробезопасности производится СНиП 21-01-97.

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

Возможные виды источников воспламенения:

- Искра при разряде статического электричества;
- Искры от электрооборудования;
- Искры от удара и трения;
- Открытое пламя [22].

Для профилактики организации действий при пожаре должен проводиться следующий комплекс организационных мер: должны обеспечиваться регулярные проверки пожарной сигнализации, первичных средств пожаротушения; должен проводиться инструктаж и тренировки по действиям в случае пожара; не должны загромождаться или блокироваться пожарные выходы; должны выполняться правила техники безопасности и

технической эксплуатации электроустановок; во всех служебных помещениях должны быть установлены «Планы эвакуации людей при пожаре и других ЧС», регламентирующие действия персонала при возникновении пожара.

Для предотвращения пожара помещение с ПЭВМ должно быть оборудовано первичными средствами пожаротушения: углекислотными огнетушителями типа ОУ-2 или ОУ-5; пожарной сигнализацией, а также, в некоторых случаях, автоматической установкой объемного газового пожаротушения [23].

## **5.2. Экологическая безопасность**

### **5.2.1. Влияние объекта исследования на окружающую среду**

В данном разделе рассматривается воздействие на окружающую среду деятельности по разработке проекта.

При разработке используется ПК, который потребляет сравнительно небольшое количество электроэнергии (мощностью 220 Вт).

Под охраной окружающей среды характеризуется различного рода мероприятиями, влияющие на следующие природные зоны: атмосфера, гидросфера, литосфера.

В ходе выполнения ВКР и дальнейшем использовании алгоритмов отсутствуют выбросы каких-либо вредных веществ в атмосферу, следовательно, загрязнение воздуха не происходит. Не происходит также и сбросов в водоемы, поэтому не оказывается никакого влияния на гидросферу.

Но непосредственно во время разработки системы образовывался мусор, такой как канцелярские принадлежности и бумага, неправильная утилизация которых может нанести вред литосфере и атмосфере при сжигании.

### **5.2.2. 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. Одним из наиболее рациональных и эффективных методов обращения с различными категориями мусора признана вторичная переработка.

Использованная бумага становится сырьем для производства различных товаров народного потребления и это позволяет уменьшить количество вырубки лесные массивы. Поэтому вся использованная и ненужная бумага сдается в макулатуру, где подвергается вторичной переработке.

Основная проблема утилизации пластика состоит в том, что он имеет очень высокую устойчивость к разложению (в зависимости от вида пластика период распада составляет от 100 до 1000 лет), а при сжигании выделяет в атмосферу токсичные вещества.. Тем самым, делая вопрос по утилизации пластика более чем актуальным для скорейшего разрешения, и не только с точки зрения охраны окружающей среды и, соответственно, здоровья человека, но и с позиции существующего в данное время дефицита полимерного сырья. Поэтому пластиковые части канцелярских принадлежностей утилизировались в специальные контейнеры для пластика, чтобы потом данный пластик был правильно утилизирован или использован повторно.

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

#### **5.3.1. Перечень возможных чрезвычайных ситуаций**

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

При работе в кабинете могут возникнуть чрезвычайные ситуации следующих видов:

- 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;
- комбинированные.

### 5.3.2. Типичные чрезвычайные ситуации при работе в офисе

При работе с ПК основными источниками чрезвычайных опасностей являются возникновение пожара, а также кибертерроризм, который может повлечь за собой порчу или потерю данных. Возникновение других видов ЧС маловероятно [22].

Вероятность возникновения пожара высока, так как в современных ЭВМ очень высокая плотность размещения элементов электронных схем, в непосредственной близости друг от друга располагаются соединительные провода и кабели, при протекании по ним электрического тока выделяется значительное количество теплоты, при этом возможно оплавление изоляции и возникновение возгорания,

Обеспечение пожарной безопасности учреждений достигается, прежде всего, установлением жесткого противопожарного режима и обучением обслуживающего персонала и учащихся мерам пожарной безопасности и действиям во время пожара.

Важно контролировать состояние дорог, проездов, подъездов и проходов к зданиям, следить за тем, чтобы они ничем не загромождались, а в зимнее время регулярно очищались от снега и льда.

В зданиях, относящихся к объектам с массовым пребыванием людей, особое внимание должно уделяться содержанию путей эвакуации. Каждое здание должно иметь не менее двух эвакуационных выходов: если один из них отрезан огнем, для спасения используется другой. Запасные выходы должны быть свободны и иметь надпись: «Запасный выход». Все двери

эвакуационных выходов свободно открываются в сторону выхода из помещений.

На случай отключения электроэнергии, у обслуживающего персонала должны быть электрические фонари – не менее одного на каждого работника дежурного персонала.

На каждом этаже здания, на видном месте должен быть вывешен план эвакуации с этажа (здания). На плане эвакуации кроме путей выхода (стрелками) указываются места размещения средств пожаротушения, телефонов.

Одной из причин возникновения ЧС, которым могут подвергаться компании, занимающиеся разработкой программного обеспечения, является кибертерроризм.

Кибертерроризм представляет собой серьезную социально опасную угрозу для человечества, сравнимую с ядерным, бактериологическим и химическим оружием, причем степень этой угрозы в силу своей новизны, не до конца еще осознана и изучена. По мнению американских экспертов, наиболее уязвимыми точками инфраструктуры являются энергетика, телекоммуникации, авиационные диспетчерские, финансовые электронные и правительственные информационные системы, а также автоматизированные системы управления войсками и оружием [17].

Искажение информации или блокирование работы информационных систем в финансовой сфере может иметь следствием экономический кризис. Для ИТ-компаний искажение или кража данных может привести к сбоям работы системы, тем самым, будут произведены срывы сроков работ, нарушена конфиденциальность компаний-партнеров. Также кибертеррористы могут удалить важную информацию с серверов или передать ее третьи лицам, которые могут опубликовать ее в открытом доступе, что может повлечь раскрытие конфиденциальной информации и компании придется платить неустойки за нарушение условий договора, также кибертеррористы могут украсть деньги со счетов компании. Могут

быть введены из строя оборудование, компьютеры, что также приведет к срывам сроков выполнения договоров.

### 5.3.3. Мероприятия по предотвращению наиболее типичных ЧС

Необходимо проводить следующие пожарно-профилактические мероприятия на предприятии:

а) Организационные мероприятия:

- 1) противопожарный инструктаж обслуживающего персонала;
- 2) обучение персонала правилам техники безопасности;

б) Эксплуатационные мероприятия:

- 1) соблюдение эксплуатационных норм оборудования;
- 2) обеспечение свободного подхода к оборудованию.
- 3) содержание в исправности изоляции токоведущих

проводников.

в) Технические мероприятия:

1) соблюдение противопожарных мероприятий при устройстве электропроводок, оборудования, систем отопления, вентиляции и освещения. В здании должны присутствовать порошковые огнетушители, установлен рубильник, обесточивающий все помещение. Если возгорание произошло в электроустановке, для его устранения должны использоваться углекислотные огнетушители или порошковые;

- 2) профилактический осмотр, ремонт и испытание оборудования.

В случае возникновения пожара сотрудники должны предпринять следующие меры [23]:

- сообщить о пожаре в пожарную охрану, задействовать систему оповещения;
- задействовать план эвакуации (открыть запасные двери и включить светоуказатели эвакуационных путей);
- вывести людей в безопасное место в соответствии с планом эвакуации;
- проверить поименно, все ли эвакуированы;

- приступить к тушению пожара первичными средствами;
- встретить пожарные подразделения и сообщить, где могли остаться люди, как туда можно подойти;
- принять меры к эвакуации имущества.

Для тушения пожаров необходимо применять углекислотные и порошковые огнетушители, которые обладают высокой скоростью тушения, большим временем действия, возможностью тушения электроустановок, высокой эффективностью борьбы с огнем. Воду разрешено применять только во вспомогательных помещениях [23].

Меры предотвращения и защиты от кибертерроризма:

1. Необходимо обеспечить разграничение доступа в помещения, где находятся информационные ресурсы.
2. Необходимо установить системы защиты от несанкционированного доступа к ресурсам компьютеров и локальных сетей.
3. Необходимо защитить (межсетевыми экранами) входы в сеть извне, в том числе и при наличии соединения с Internet.
4. Необходимо организовать постоянный мониторинг и аудит сети;
5. Необходимо использовать антивирусные защиты.
6. Необходимо быть осмотрительным в отношении писем с вложенными картинками, поскольку файлы могут содержать вирусы. Открывайте вложения только от известных вам отправителей. И всегда проверяйте вложения на наличие вирусов, если это возможно.
7. Нельзя необдуманно переходить по ссылкам, содержащимся в спам-рассылках. Необходимо удостовериться в правильности ссылки, прежде чем переходить по ней из электронного письма.
8. Нельзя заполнять полученные по электронной почте формы и анкеты. Личные данные безопасно вводить только на защищенных сайтах.
9. Необходимо проверять запросы персональных данных из каких-либо деловых и финансовых структур. Лучше обратиться в эти структуры по контактам, указанным на официальном сайте, а не в электронном письме.



10. Нельзя хранить логины и пароли на персональном компьютере. Желательно запоминать или записывать на бумажный носитель и хранить в защищенном месте [17].

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

##### **5.4.1. Правовые нормы трудового законодательства для рабочей зоны оператора ПЭВМ**

Законодательством РФ регулируются отношения между организацией и работниками, касающиеся оплаты труда, трудового распорядка, социальных отношений, особенности регулирования труда женщин, детей, людей с ограниченными способностями и др.

Продолжительность рабочего дня не должна превышать 40 часов в неделю. Для работников до 16 лет – не более 24 часов в неделю, от 16 до 18 лет – не более 35 часов, как и для инвалидов I и II группы. Для работников, работающих на местах, отнесенных к вредным условиям труда 3 и 4 степени – не более 36 часов.

Возможно установление неполных рабочих дней для беременной женщины; одного из родителей (опекуна, попечителя), имеющего ребенка в возрасте до четырнадцати лет (ребенка-инвалида в возрасте до восемнадцати лет). Оплата труда при этом производится пропорционально отработанному времени. Ограничений продолжительности ежегодного основного оплачиваемого отпуска, исчисления трудового стажа и других трудовых прав при этом не имеется.

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

Организация обязана предоставлять ежегодные отпуска продолжительностью 28 календарных дней. Для работников, занятых на работах с опасными или вредными условиями, предусматривается дополнительный отпуск.

Работнику в течение рабочего дня должен предоставляться перерыв не более двух часов и не менее 30 минут, который в рабочее время не включается. Всем работникам предоставляются выходные дни, работа в выходные дни производится только с письменного согласия работника.

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

Законодательством РФ запрещены дискриминация по любым признакам, а также принудительный труд [24].

#### 5.4.2. Организационные мероприятия при компоновке рабочей зоны

Исходя из общих принципов организации рабочего места, в нормативно-методических документах сформулированы требования к конструкции рабочего места.

Основными элементами рабочего места оператора являются: рабочий стол, рабочий стул (кресло), дисплей, клавиатура [25].

Рабочие места с ЭВМ должны располагаться на расстоянии не менее 1,5 м от стены с оконными проемами, от других стен – на расстоянии 1 м, между собой – на расстоянии не менее 1,5 м. При размещении рабочих мест необходимо исключить возможность прямой засветки экрана источником естественного освещения. Источник естественного освещения (окно) не должен также попадать в зону прямого наблюдения пользователя. По отношению к световым проемам рабочие места целесообразно располагать таким образом, чтобы естественный свет падал на него сбоку, преимущественно слева.

При размещении ЭВМ на рабочем месте должно обеспечиваться пространство для пользователя величиной не менее 850 мм с учетом выступающих частей оборудования и применения (при необходимости) спецодежды. Для стоп должно быть предусмотрено пространство по глубине и высоте не менее 150 мм, по ширине – не менее 530 мм.

Располагать ЭВМ на рабочем месте необходимо так, чтобы поверхность экрана находилась на расстоянии 400 – 700 мм от глаз пользователя. Рекомендуется размещать элементы рабочего места таким образом, чтобы выдерживалось равное расстояние глаз пользователя от экрана, клавиатуры, держателя документов.

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. Сиденье и спинка кресла должны быть полумягкими, с нескользящим, неэлектризующимся и воздухопроницаемым покрытием, материал которого обеспечивает возможность легкой очистки от загрязнения.

Поверхность пола в помещениях эксплуатации компьютеров должна быть ровной, без выбоин, нескользкой, удобной для очистки и влажной уборки, обладать антистатическими свойствами.

Площадь на одно рабочее место с компьютером (жидкокристаллические, плазменные дисплеи) для взрослых пользователей должна составлять не менее 4,5 м<sup>2</sup>.

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].

#### 5.4.3. Влияние разработанных автоматических тестов.

На сегодняшний день тестирование стало неотъемлемой частью процесса разработки программных продуктов. С помощью него компании могут своевременно обнаружить и устранить большое число ошибок, что приводит к повышению качества всех характеристик ПО.

Автоматизация процесса тестирования помогает компаниям сокращать время, затрачиваемое на тестирование, а также упрощать весь процесс, потому что применяются специальные программные инструменты для создания и запуска тестов, а также проверки результатов их выполнения [1].

Автоматизация тестирования позволяет достичь следующих результатов:

- Позволяет повысить качество разрабатываемого ПО. Внедрение автотестов позволяет производить тестирование продуктов чаще и почти не требует вмешательства тестировщика. Поэтому полное тестирование можно проводить после внесения даже незначительных изменений в программу. Это позволит быстрее обнаружить и исправить ошибки, подмеченные разработчиками. Кроме того, автотесты выполняются однообразно, без вмешательства человека, что исключает «человеческий фактор», который может повлечь за собой появление новых ошибок. Следовательно, это позволит выпускать на рынок более качественное программное обеспечение.

- Сокращение затрат. Внедрение автотестов позволяет сэкономить как временные затраты, так и денежные. На выполнение автоматических тестов тратится гораздо меньше времени, чем на ручные, потому что программа не переутомляется и не сбивается во время работы. Также, результаты выполнения автотестов можно автоматически рассылать, а также хранить отчеты о выполнении. Это уменьшает время поиска возникшей ошибки. Во время выполнения автотестов не требуется участия тестировщика, поэтому он может заниматься другой важной работой, также тесты можно запускать в нерабочее время, а после только анализировать полученный результат. Также, внедрение написанные автотесты, нуждаются

в меньшей поддержке и анализе результатов их выполнения, чем при ручном тестировании.

Однако автоматические тесты имеют существенный недостаток:

- Повышается возможность пропуска ошибки. Из-за того, что автотесты всегда выполняются однообразно, невозможно заметить недочеты в работе программного обеспечения, которые смог бы найти тестирующий при ручном тестировании. Также, автотест будут пропускать ошибки, проверка которых не была в него заложена [2].

Но, несмотря на этот недостаток, внедрение автоматического тестирования на предприятии имеет много существенных преимуществ, в частности, экономит временные ресурсы тестирующего, что позволит ему уделить больше времени на анализ частей программы, которые подвержены наибольшей вероятности возникновения ошибок, и создать автотесты, которые будут качественнее прорабатывать данные части, что минимизирует возможность пропуска ошибок.

## ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы были решены следующие задачи:

- Изучены существующие подходы и алгоритмы автоматизации тестирования и выбраны методики, наиболее подходящие для тестирования настольных приложений.

- Проведен подробный анализ существующих инструментов для автоматизации тестирования, выбраны и изучены инструменты для автоматизации процесса тестирования, такие как AutoIT и TestComplete, исходя из специфики проектов компании.

- Разработана и внедрена технология автоматизированного тестирования, включающая в себя разработку тестовых планов для модулей «Полотно.Ведомости» и «Specbook». В соответствии с планом был реализован набор автоматических тестов для данных модулей.

В ходе работы для модуля «Specbook» было создано 8 пакетов автоматических тестов, реализующих около 250 тестовых случаев. В настоящее время все полученные результаты применяются в процессе разработки и поддержки модуля «Specbook». С помощью данных тестов было найдено 30 дефектов, вследствие чего, это поможет избежать сопутствующих ошибок в будущем. Для модуля «Полотно.Ведомости» были созданы 2 пакета автоматических тестов, реализующих около 60 тестовых случаев. В настоящее время тестирование данного модуля с помощью разработанных автотестов не происходит из-за отказа компании использовать данный инструмент для автоматизации тестирования настольных приложений. С помощью данных тестов было найдено 15 дефектов, вследствие чего, это поможет избежать сопутствующих ошибок в будущем.

Также была произведена оценка эффективности внедрения автоматических тестов с использованием платного инструмента

TestComplete, так и с использованием свободно распространяемого инструмента AutoIT. Было выявлено, что для автоматизации одного проекта в 4 раза выгоднее использовать бесплатную программу. Хотя на разработку тестов на скриптовом языке тратится в 3 раза больше времени, чем разработка тестов с помощью TestComplete, применять его в компании, которая не ведет одновременной большое число проектов, не выгодно из-за высокой стоимости лицензии.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Элфрид Дастин, Джефф Рэшка, Джон Пол \ Е. Молодцова, М. Павлов. Автоматизированное тестирование программного обеспечения. Внедрение, управление и эксплуатация. / Москва: ЛОРИ, 2003.
2. И. Винниченко. Автоматизация процессов тестирования. / СПб: Питер, 2005.
3. Борис Бейзер. Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем. / СПб: Питер, 2004.
4. Савин Роман. Тестирование DOT COM. / СПб: Дело, 2007.
5. Сэм Канер, Джек Фолк, Енг Кек Нгуен. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений. / К: ДиаСофт, 2001.
6. Виды Тестирования. Про Тестинг - Тестирование Программного Обеспечения [Электронный ресурс] URL: <http://www.protesting.ru/testing/types/sanity.html> дата обращения: 20.02.2017.
7. Оценка эффективности автоматизации тестирования. Технологии качества. [Электронный ресурс] URL: <http://a1qa.ru/blog/otsenka-effektivnosti-avtomatizatsii-testirovaniya/> дата обращения: 22.02.2017.
8. Техническое задание на выполнение работ по теме: «Техническая поддержка и сопровождение системы автоматической генерации спецификаций SpecBook».
9. Русское сообщество AutoIT [Электронный ресурс] URL: <http://autoit-script.ru> дата обращения: 09.03.2017.
10. Учебник по TestComplete [Электронный ресурс] URL: <http://tctutorial.ru>, дата обращения: 12.07.2016.
11. Технический дизайн «Автоматизированная система для создания ведомости подсчета объемов работ по автомобильным дорогам и



генеральным планам» - 2014. – 31с.

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

13. ГОСТ 12.0.003-74. Система стандартов безопасности труда. Опасные и вредные производственные факторы. Классификация / Ассоциация инженеров-программистов по охране труда - Догма URL: <http://dogma.su/normdoc/rospotrebnadzor/sreda-factor/other/detail.php?ID=1327> дата обращения: 11.03.2017.

14. Ефремова О.С. Требования охраны труда при работе на персональных электронно-вычислительных машинах. – 2-е изд., перераб. и доп. – М. : Издательство «Альфа-Пресс», 2008. – 176 с.

15. Назаренко О.Б. Безопасность жизнедеятельности: учебное пособие / О.Б. Назаренко, Ю. А. Амелькович; Томский политехнический университет. – 3-е изд., перераб. и доп. – Томск: Изд-во Томского политехнического университета, 2013. – 178 с.

16. СанПиН 2.2.4.548-96. Санитарные правила и нормы. Гигиенические требования к микроклимату производственных помещений / Документы системы ГАРАНТ URL: <http://base.garant.ru/4173106/> дата обращения: 10.03.2017.

17. Старостина Е.В. Защита от компьютерных преступлений и кибертерроризма. Вопросы и ответы. – 2-е изд. / Е. В. Старостина, Д. Б. Фролов. – М. : Издательство «Эксмо», 2005.

18. 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.

19. 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.

20. СП 52.13330.2011. Естественное и искусственное освещение /Электронный фонд правовой и нормативно-технической документации.URL: <http://docs.cntd.ru/document/1200084092> дата обращения: 11.03.2017.

21. 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.

22. 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.

23. СНиП 21-01-97. Пожарная безопасность зданий и сооружений / Электронный фонд правовой и нормативно-технической документации. URL: <http://docs.cntd.ru/document/871001022> дата обращения: 11.03.2017.

24. 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.

25. ГОСТ Р 50923-96 Дисплеи. Рабочее место оператора. Общие эргономические требования и требования к производственной среде. Методы измерения / Электронный фонд правовой и нормативно-технической документации.URL: <http://docs.cntd.ru/document/1200025975> дата обращения: 11.03.2017.

## ПРИЛОЖЕНИЕ А

Отрывок из документа тестовых сценариев к настольному приложению «Модуль формирования спецификаций «Specbook»

### 1.1. Работа с проектом

В окне навигации по проектам представлены все проекты, когда-либо созданные на предприятии. Перед тем как создавать новый проект необходимо убедиться, что в выпадающем списке наверху выбран нужный отдел. Поскольку именно этому отделу будет доступна информация, хранящаяся в проекте: объекты иерархии, спецификации и сводные спецификации.

Подробнее о работе с проектом:

- Создать проект
- Редактировать проект

#### 1.1.1. Создать проект

1. Чтобы создать проект можно воспользоваться кнопкой «Создать новый проект» на панели инструментов (см. Рисунок А2), либо пунктом «Создать проект» в контекстном меню дерева проектов (см. Рисунок А3).

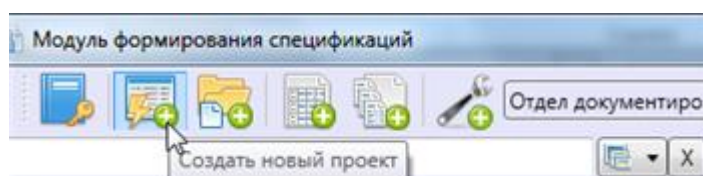


Рисунок А2 – Кнопка «Создать новый проект» на панели инструментов главного окна модуля

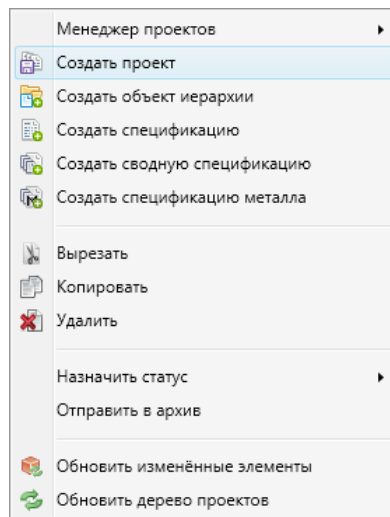


Рисунок А3 – Команда «Создать проект» в контекстном меню дерева проектов

2. После нажатия на кнопку, появится мастер создания проекта (см. Рисунок А4).

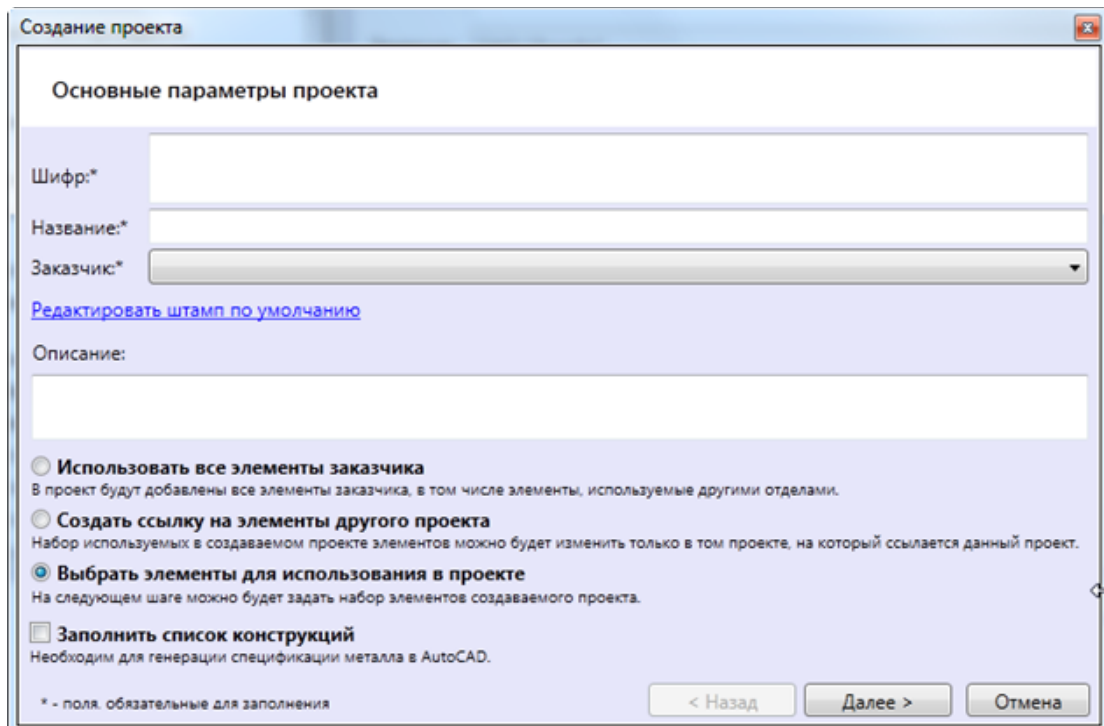


Рисунок А4 – Мастер «Создание проекта»

3. В появившемся окне необходимо ввести «Название проекта», «Шифр» и выбрать заказчика проекта, также можно ввести небольшое описание проекта в поле ниже.

Поля «Название», «Шифр» и «Заказчик» обязательны для заполнения.

4. Далее необходимо выбрать способ добавления элементов в проект:

- «Использовать все элементы заказчика» - выбрав данную опцию, все элементы добавленные заказчику автоматически будут добавлены в проект.
- «Создать ссылку на элементы другого проекта» - выбрав этот пункт, будет доступна возможность использовать элементы других проектов.
- «Выбрать элементы для использования в проекте» - позволит выбрать элементы заказчика для использования в проекте.

В случае если необходимо заполнить список конструкций для спецификации металла в AutoCAD то в окне мастера необходимо поставить галочку напротив соответствующего пункта.

5. Следующее окно мастера создания проекта будет зависеть от выбранного способа добавления элементов в проект.

5.1. Если выбрана опция «Использовать все элементы заказчика», то мастер уведомит пользователя об успешном добавлении всех элементов заказчика в проект и создаст проект.

5.2. Если выбрана опция «Создать ссылку на элементы другого проекта», то следующее окно мастера предложит пользователю выбрать проект, элементы которого будут использоваться в создаваемом проекте (см. Рисунок А5).

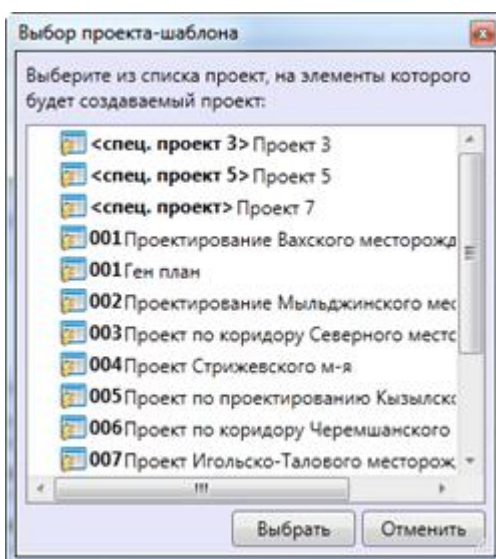


Рисунок А5 – Окно выбора проекта-шаблона

В данном окне необходимо кликнуть мышкой по проекту, элементы которого нужно использовать в своем проекте, затем нажать на кнопку «Выбрать».

5.3. Если выбрана опция «Выбрать элементы для использования в проекте», то следующее окно мастера предложит пользователю выбрать элементы заказчика, или элементы других проектов (см. Рисунок А6)

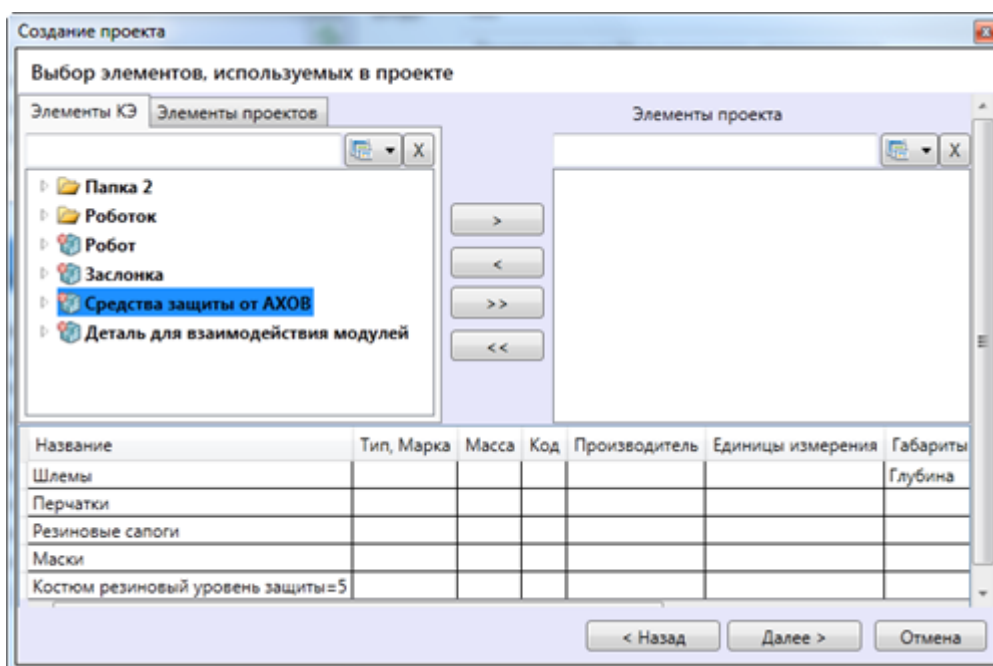
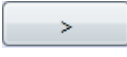
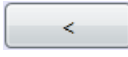
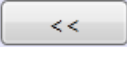
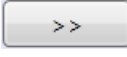


Рисунок А6 – Окно мастера создания проекта. Выбор элементов используемых в проекте

Чтобы добавить элемент в проект, необходимо выбрать нужный элемент в левом дереве и нажать на кнопку . После этого выбранный элемент исчезнет в левом дереве и появится в правом. Для удаления элементов из проекта служит кнопка . При ее нажатии элементы, выделенные в правом дереве, переместятся в левое дерево. Кнопка  удаляет из проекта все элементы. Кнопка  добавляет в проект все элементы заказчика.

Под деревьями расположена таблица с атрибутивной информацией по выбираемым в деревьях элементам.

5.4. Также можно добавлять элементы других проектов. Для этого необходимо перейти во вкладку «Элементы проектов» над левым деревом (см. Рисунок А7).

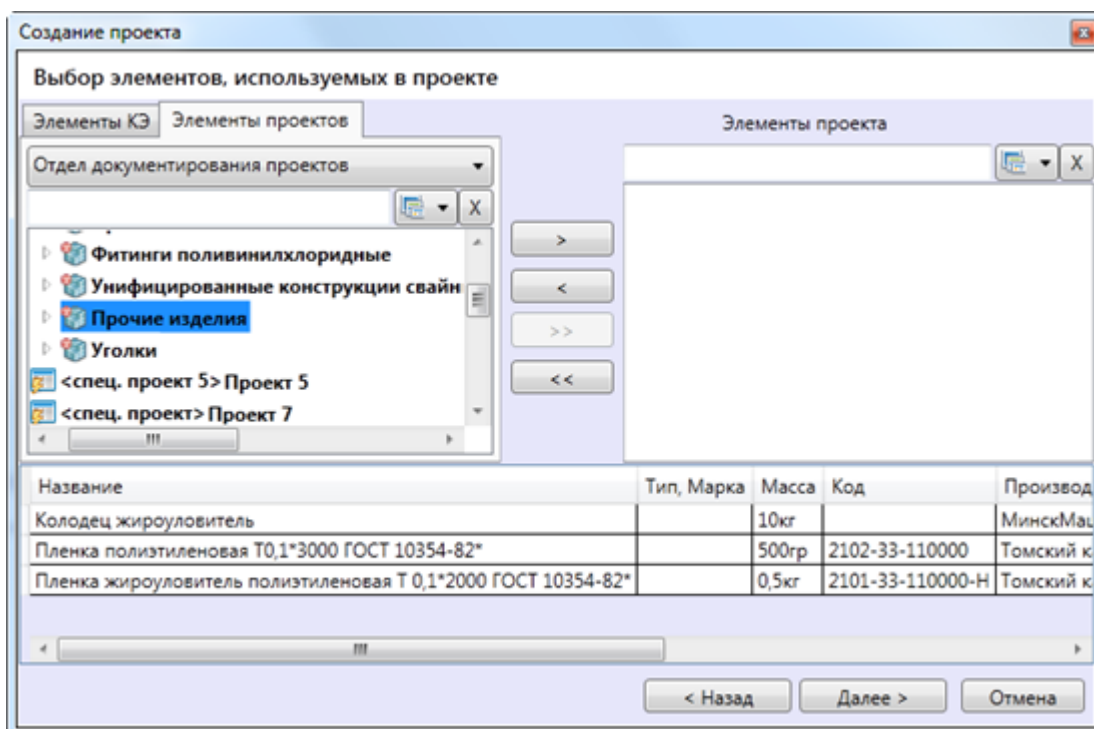


Рисунок А7 – Окно мастера создания проекта. Выбор элементов, используемых в проекте

Также необходимо выбрать в выпадающем списке нужный отдел. Поскольку именно этому отделу будут доступны данные, хранящиеся в проекте: папки, типы или элементы.

б. После добавления в правое дерево всех необходимых объектов нужно нажать на кнопку «Далее». В появившемся окне завершающего этапа создания проекта необходимо нажать на кнопку «Создать проект» (см. Рисунок А8).

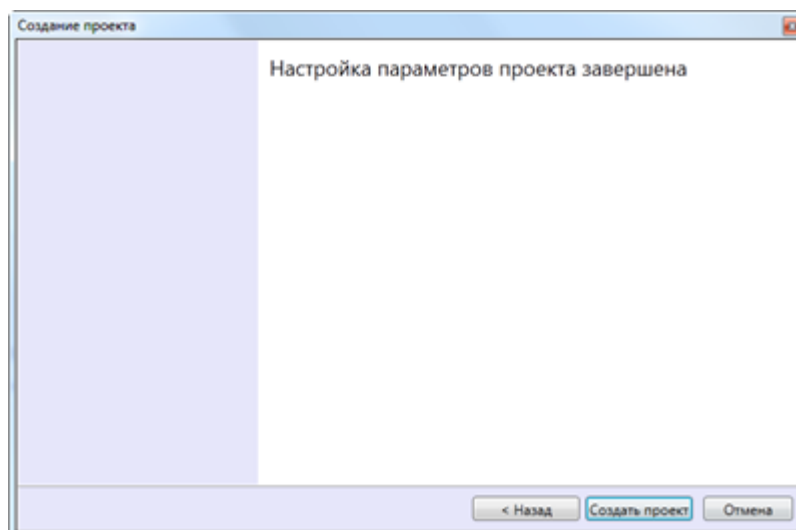


Рисунок А8 – Окно мастера создания проектов. Окно завершения создания проекта

**Результат:** После нажатия на кнопку, окно мастера закроется, и созданный проект добавится в дерево проектов.

### 1.1.2. Создание конструкций для спецификации металла в AutoCAD

1. Если в окне мастера создания проекта была выбрана опция «Заполнить список конструкций» (см. Рисунок А9), то следующим окном мастера будет окно редактирования списка конструкций (см. Рисунок А10).

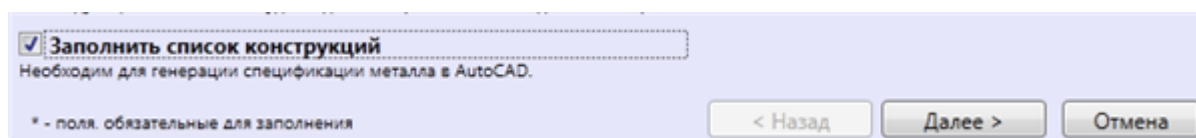


Рисунок А9 – Выбранная опция «Заполнить список конструкций»

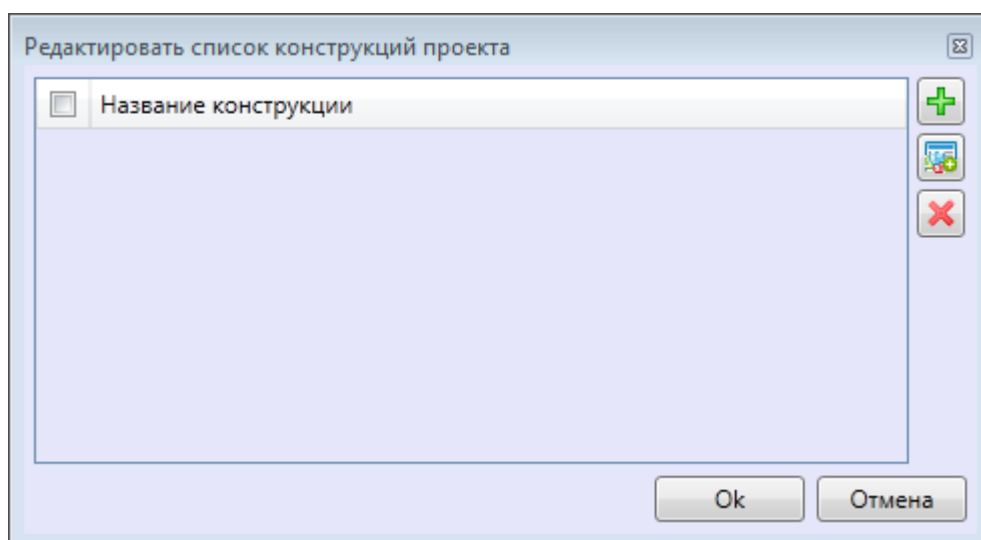


Рисунок А10 – Окно редактирования списка конструкций



2. Чтобы добавить конструкцию нужно нажать на кнопку «Добавить собственную конструкцию» на панели инструментов (см. Рисунок А11). В списке добавится строка для ввода названия конструкции. Необходимо ввести название конструкции и нажать Enter для сохранения конструкции в списке.

2.1. Чтобы использовать уже созданную конструкцию нужно нажать на кнопку «Выбрать конструкцию из проекта» на панели инструментов (см. Рисунок А11).

2.2. Чтобы удалить конструкцию нужно выделить конструкцию, поставив галочку напротив названия и нажать на кнопку «Удалить конструкцию» (см. Рисунок А11). После чего конструкция исчезнет из списка.

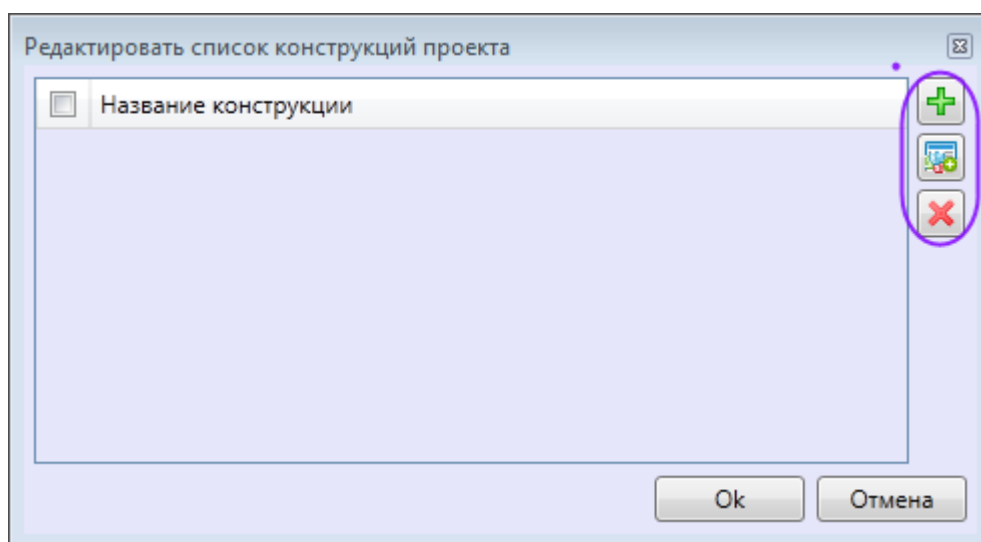


Рисунок А11 – Окно «Создание проекта», панель инструментов

3. После заполнения списка конструкций нажать «Ок».

1.1.3. Редактировать проект

Для того чтобы изменить данные проекта, необходимо:

1 В дереве проектов найти нужный проект, выделить его, левым кликом мыши,

2 В появившемся справа окне свойств изменить данные.

3 Нажать на кнопку «Сохранить» (см. Рисунок А12).

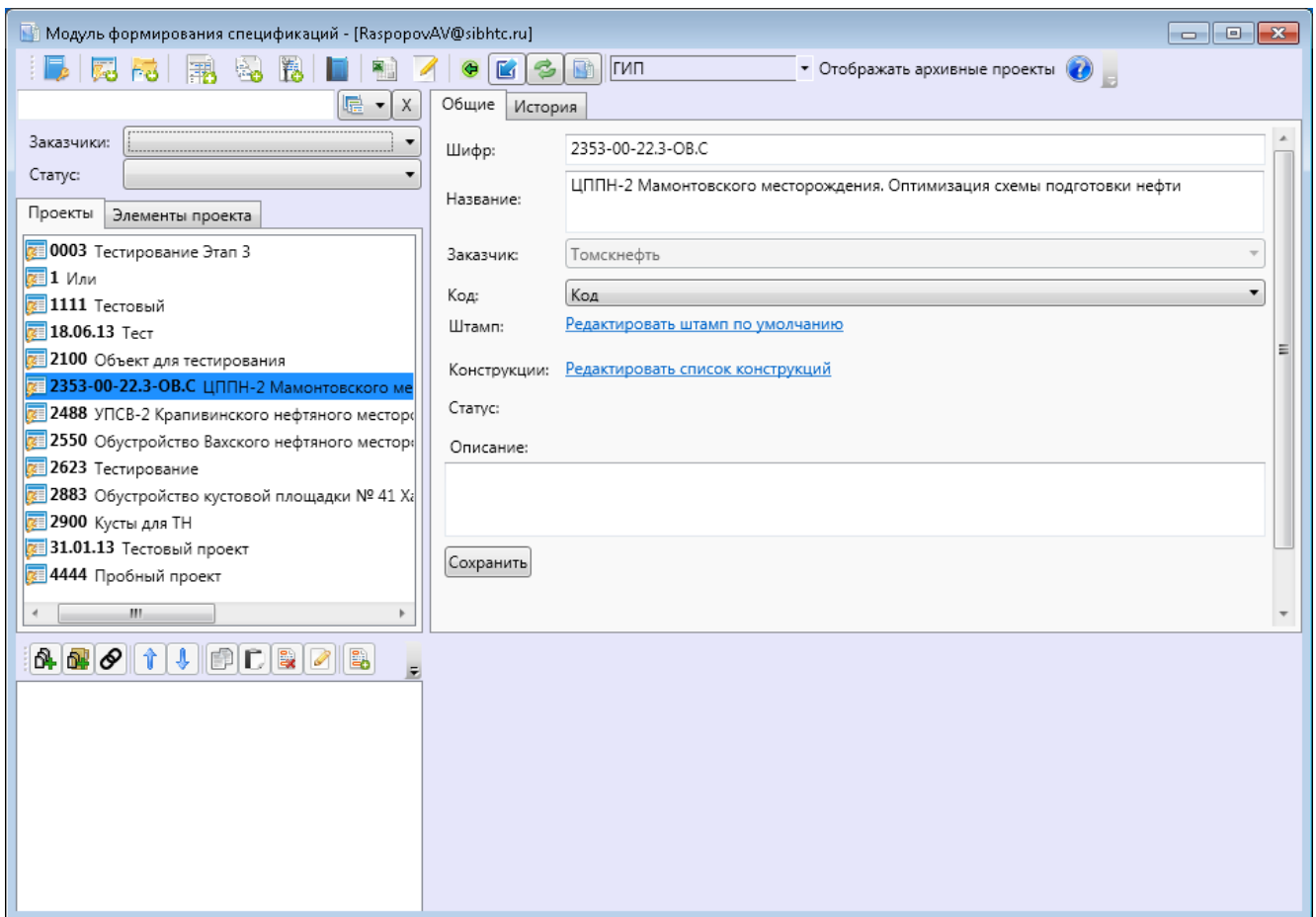


Рисунок А12 – Окно редактирования информации о проекте

На панели справа располагаются ссылки, кликнув по которым можно перейти к соответствующим :

- Редактировать штамп по умолчанию. Кликнув по этой ссылке пользователь попадет в окно «Редактирование штампа» (см. Рисунок А12).

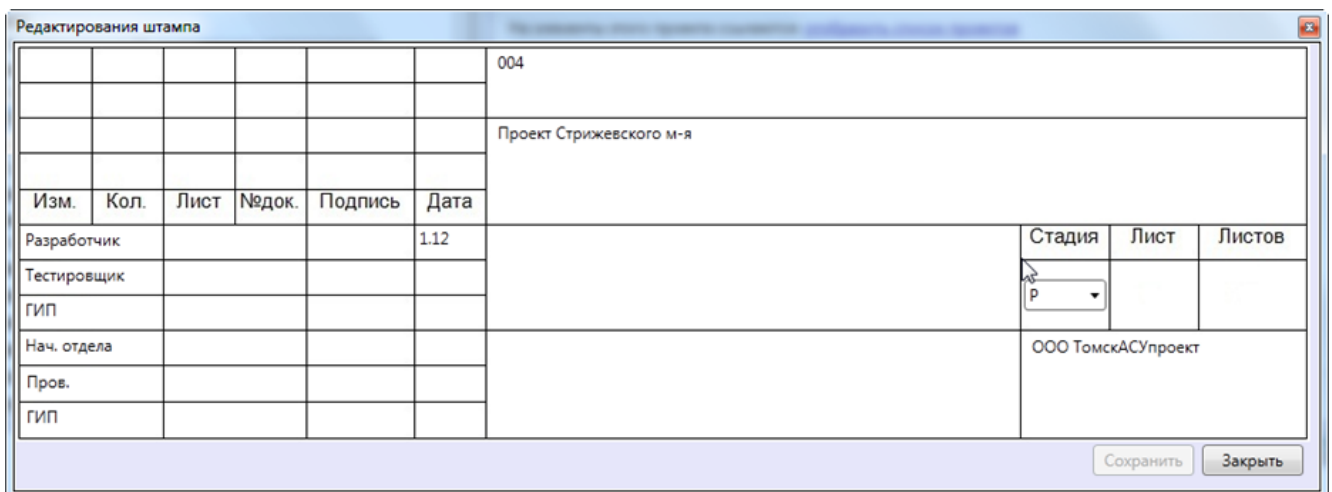


Рисунок А13 – Окно «Редактирование штампа»

- Редактировать список конструкций(см. Рисунок А14).

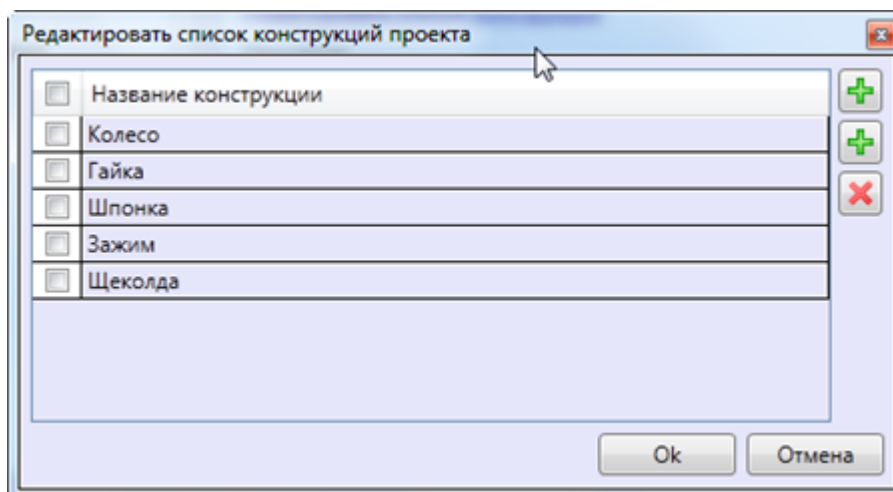


Рисунок А14 – Окно «Редактирование списка конструкций»

При редактировании проекта имеется возможность выполнения как одиночных, так и групповых операций редактирования. Для того чтобы редактировать (копировать/вырезать/вставить/удалить) группу проектов необходимо в дереве объектов выбрать несколько элементов (для этого нужно удерживать клавишу «Shift» либо «Ctrl») и затем выбрать нужную команду.

### **Команда «Копировать»**

Для того чтобы скопировать проект, необходимо кликнуть правой кнопкой мыши по нужному проекту в дереве проектов и затем в контекстном меню выбрать пункт «Копировать» (см.Рисунок А15).

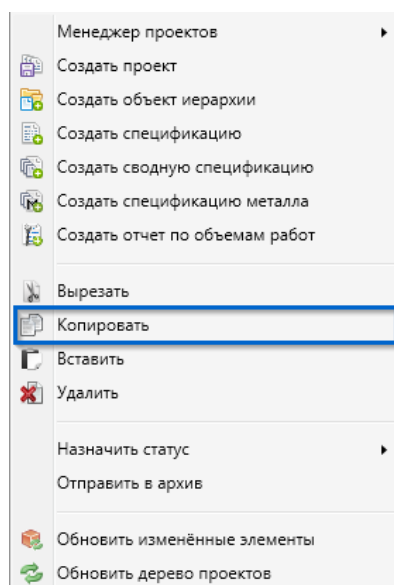


Рисунок А15 – Команда «Копировать» в контекстном меню дерева проектов модуля

### **Команда «Вырезать»**

Для того чтобы вырезать проект, необходимо кликнуть правой кнопкой мыши по нужному проекту в дереве проектов и затем в контекстном меню выбрать пункт «Вырезать» (см. Рисунок А16).

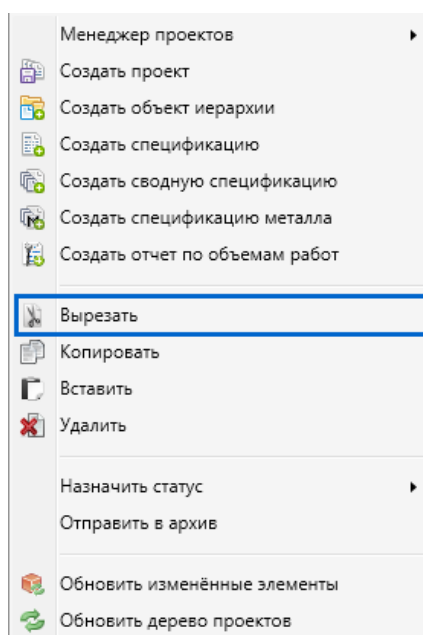


Рисунок А16 – Команда «Вырезать» в контекстном меню дерева проектов модуля

### **Команда «Вставить»**

Чтобы вставить проект необходимо кликнуть правой кнопкой мыши в дереве проектов по месту, куда нужно вставить проект и затем в контекстном меню выбрать пункт «Вставить» (см. Рисунок А17).

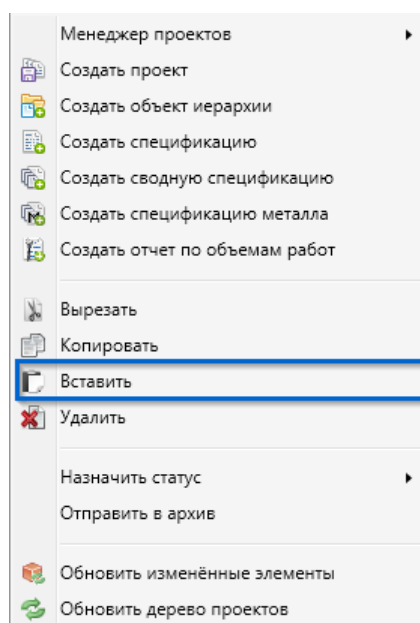


Рисунок А17 – Команда «Вставить» в контекстном меню дерева проектов модуля

При вставке проекта в папку где проект с таким именем уже существует, к имени проекта прибавляется конструкция «- копия (порядковый номер копии)».

### Команда «Удалить»

Для того чтобы удалить проект, необходимо кликнуть правой кнопкой мыши по нужному проекту в дереве проектов и затем в контекстном меню выбрать пункт «Удалить» (см. Рисунок А18).

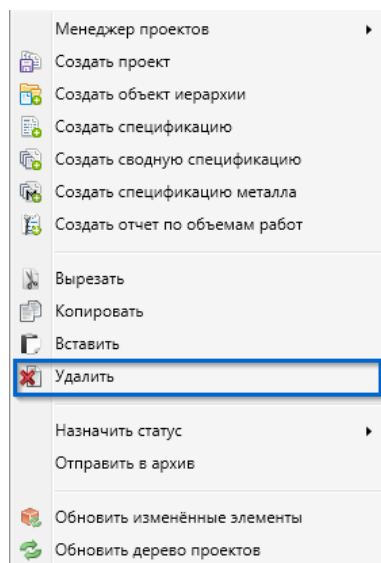


Рисунок А18 – Команда «Удалить» в контекстном меню дерева проектов модуля

#### 1.1.4. Менеджер проектов

Для того чтобы в дереве проектов отображались только нужные для работы проекты, а остальные скрыть, необходимо:

1. Кликнуть по дереву проектов правой кнопкой мыши, и в контекстном меню выбрать команду «Менеджер проектов» (см. Рисунок А19).

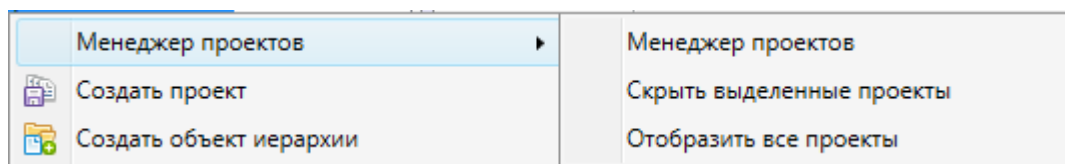


Рисунок А19 – Команда «Менеджер проектов» в контекстном меню дерева проектов

2. В появившемся окне отображаются все существующие проекты (см. Рисунок А20). При помощи галочек выбора проекты разделены на «Скрытые» и «Отображаемые».

3. Для отображения нужных проектов перед их шифром и наименованием необходимо сделать выделение, для скрытия проектов – снять выделение.

4. Для завершения работы менеджера проектов необходимо нажать на кнопку «ОК», чтобы все изменения выполнились, либо «Отмена», чтобы отменить изменения.

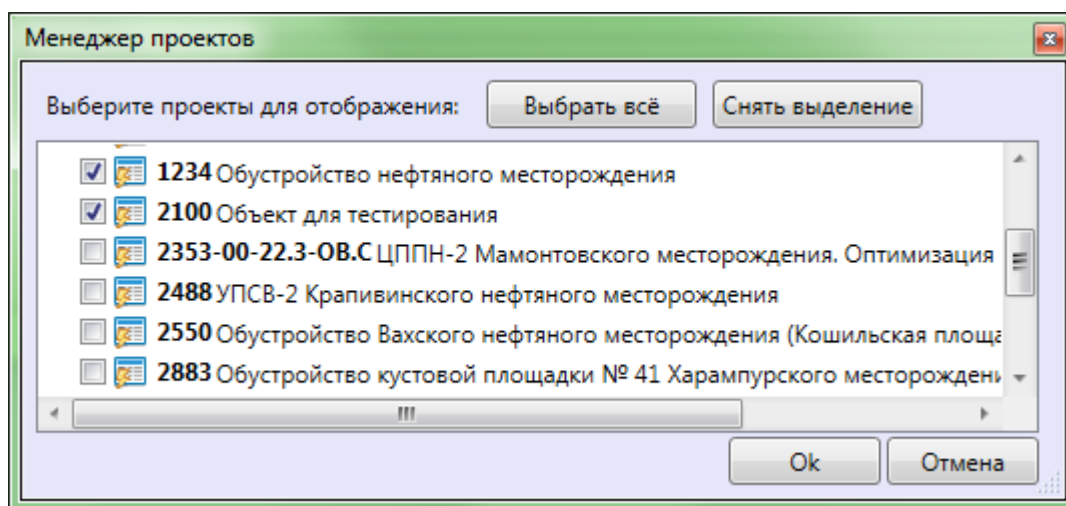


Рисунок А20 – Диалоговое окно «Менеджер проектов»

5. Для того чтобы быстро скрыть нужные проекты или отобразить все проекты без использования диалогового окна «Менеджер проектов», нужно в контекстном меню дерева проектов соответственно выбрать команду «Скрыть выделенные проекты», «Отобразить все проекты» (см. Рисунок А19).

## ПРИЛОЖЕНИЕ Б

### Раздел 3.2. Creating autotests

Студент:

Группа	ФИО	Подпись	Дата
8ИМ5А	Федорченко Алёна Вадимовна		

Консультант кафедры ИСТ:

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

Консультант – лингвист кафедры ИСТ:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель кафедры ИЯКЦ	Горбатова Татьяна Николаевна			

## Creating autotests

For autotests of the software module Polotno.Vedomosti was used a software product TestComplete. It is a functional automated testing platform developed by SmartBear Software. TestComplete gives testers the ability to create automated tests for Microsoft Windows, Web, Android and OS iOS applications. In TestComplete, functional tests can be created in the form of tests based on keywords (keyword tests) or in the form of scripts [11]. Tests of both kinds can be written down or created from scratch using the built-in TestComplete editors. The program interface is shown in Figure B1.

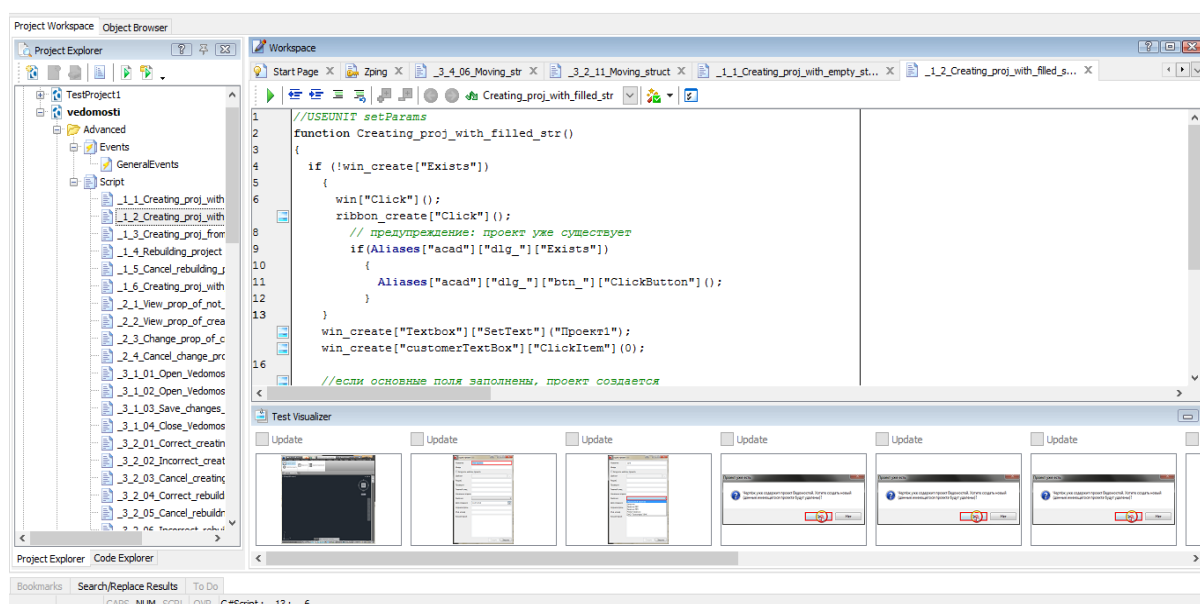


Figure B1 – Program interface TestComplete

Autotests were created on the basis of keywords, then generated in scripts and then supplemented with various checks.

The test recording in TestComplete takes place in three stages:

1. Record Keyword Test.
2. Perform the test actions described in the test case.
3. Stop recording the test and generate it in the script.

Further in the script were added various checks: that the test window was open, the required fields were filled, etc.



When writing autotests, the following problems occurred:

3. Each time the program test is running, a new code is automatically generated for this dialog application (so using of the autotests before was impossible). Since Windows initializes windows applications, it assigns codes of equal length each time, but differing in content (Fig. 3.3). This problem was solved by replacing each character code with a wildcard character (?). Wildcards are special characters that can replace unknown characters in a text value and are useful for finding multiple items with similar but not identical data.

4. Elements of the program have long names and to simplify and understand the code were replaced by variables and placed in a separate script (Fig. B2).

```
var ind = 1;
var win = Sys["Process"]("acad")["Window"]("Afx:????????????????8:????????????????????????????????????????????????????????????", "AutoCAD");
var ribbon_create = Sys["Process"]("acad")["Window"]("Afx:????????????????8:????????????????????????????????????????????????????????????");
var win_create = Aliases["acad"]["HwndSource_ProjectWindow"]("ProjectWindow")["MainGrid"];
var ribbon_prop = Sys["Process"]("acad")["Window"]("Afx:????????????????8:????????????????????????????????????????????????????????????");
var ribbon_vedomosti = Sys["Process"]("acad")["Window"]("Afx:?????????????????:????????????????????????????????????????????????????????????");
var ribbon_admin = Sys["Process"]("acad")["Window"]("Afx:????????????????8:????????????????????????????????????????????????????????????");

var grid_elem = Aliases["acad"]["HwndSource_AddElementWindow"]("AddElementWindow")["Grid"];
var grid_vedom = Aliases["acad"]["HwndSource_vedomostiWindow"]("vedomostiWindow")["Grid"];
var grid_link = Aliases["acad"]["HwndSource_LinksWindow"]("LinksWindow")["Grid"];
var grid_total = Aliases["acad"]["HwndSource_V2TotalRowWindow"]("V2TotalRowWindow")["Grid"];

var win_vedomosti = Aliases["acad"]["HwndSource_vedomostiWindow"];
var win_prop = Aliases["acad"]["HwndSource_ProjectWindow2"]("ProjectWindow")["MainGrid"];
```

Figure B2 – A script containing abbreviations of system element names

The scripting language AutoIT was used to write autotests for the program module "Specbook".

When testing the installation of the module "Specbook" it was necessary to make the following checks:

1. The availability of the installed AutoCAD 2012 on the computer, because the "Specbook" is the module of this program.
2. Presence of the previously installed version of the module.
3. Opened all the installation windows.

The script that checks the installation of the module is shown below:

```
#include <File.au3>
#include <Log.au3>
#include <Outlook.au3>

If WinExists(@ScriptName) Then Exit
AutoItWinSetTitle(@ScriptName)

;Main window name
Global $Title="Installing - SpecBook"
; Window name when Autocad is not installed
Global $title_1="Error"
; Windows name for
Global $title_2="Install"

Install_SpecBook()

Func Install_SpecBook()
Run('@ProgramFilesDir & \SpecBook*.exe')
Sleep (5000)

if WinExists($title_1) Then
ControlClick($title_1,'OK','Button1')
_Log("AutoCAD is not installed")
Exit
Endif

if WinExists($title_2) Then
ControlClick($title_2,'Yes','Button1')
ExitLoop
Endif

WinWaitActive($Title)
If WinExists($Title) then
WinActivate($Title)
ControlClick($Title, "Next", 'Button2')
Else
_Log ("Install window was not opened") exit
EndIf
Sleep (5000)

WinWaitActive($Title)
If WinExists($Title) then
WinActivate($Title)
```

```
ControlClick($title_1, "Elements catalog", 'Textbox2')
Else
_Log ("Elements catalog was not opened") exit
EndIf
Sleep (5000)
```

```
WinWaitActive($Title)
If WinExists($Title) then
WinActivate($Title)
ControlClick($title_1, "Next", 'Button2')
Else
_Log ("Install window was not opened ") exit
EndIf
Sleep (5000)
```

```
WinWaitActive($Title)
If WinExists($Title) then
WinActivate($Title)
ControlClick($title_1, "Install", 'Button3')
Esle _Log ("Install incomplete") exit
EndIf
Sleep (5000)
```

```
WinWaitActive($Title)
If WinExists($Title) then
WinActivate($Title)
ControlClick($title_1, "Finish", 'Button4')
Esle _Log ("Install incomplete ") exit
EndIf
```

```
If FileNoExists("C:\Program Files (x86)\AutoCAD\*.spb") Then
Esle _Log ("Checking file was not created") exit
EndIf
```

```
_Mail($Title)
Exit
```

Additional scripts and libraries are connected using #include.

Next, there is an entry that prevents the test from being restarted if it was started earlier:

```
If WinExists(@ScriptName) Then Exit
AutoItWinSetTitle(@ScriptName)
```

If for one of these reasons the test failed, then the message about this was recorded in a log file, which has the following format (Fig. B3):

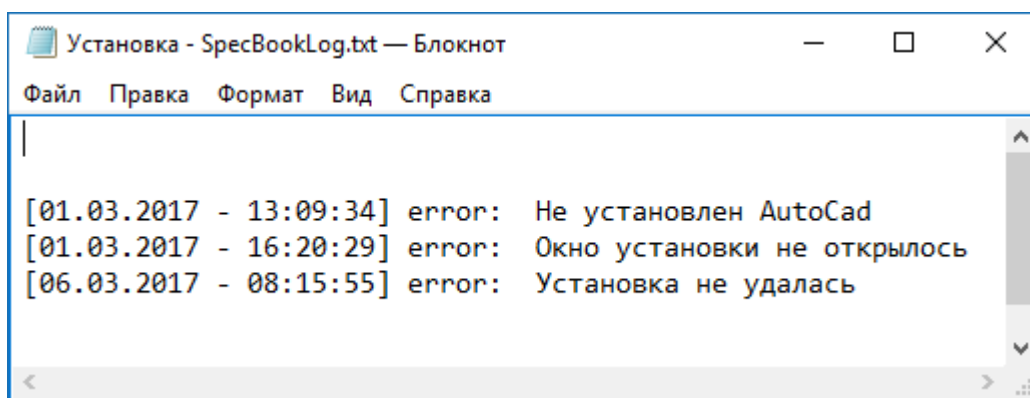


Figure B3 – Installation process log file

As can be seen from the figure, the log file is represented by a document with the extension \*.txt, the title of which describes the purpose of the test script. The error records have information about the recording time and the type of error that can be used to generate an error in the Redmine system.

This logging is done using the following function:

```
Func _Log($LogTe, $Title)
    $LogFile = @ScriptDir & "\"&$Title&"Log.txt"
    If Not FileExists($LogFile) Then _FileCreate($LogFile)
    Local $log_file_id = FileOpen($LogFile, 1)
    FileWriteLine($log_file_id, "["&@YEAR&"-"&@MON&"-"&@MDAY&" :
"&@hour&"-"&@min&"]"&StringFormat("error: "&$LogTe))
    FileClose($log_file_id)
EndFunc
Где _Log () – function name,
$LogTe – error parameter
$Title – text
```

As you can see from the script, every window that should appear during installation is tested, and, if it is not there, an entry about this with an indication of the event that failed fails to get into the log file, and the test script execution is completed.

After the installation is complete, the test results are sent to the developer's mail with a letter containing the name of the test and a file indicating the error.

The script that is responsible for sending mail is shown below.

```
Func _Mail($Title)
$SmtServer = "smtp.mail.ru" ; SMTP server
$FromName = "Tester1" ; sender name
$FromAddress = "*****" ;sender address
$Password = "*****"; sender password
$ToAddress = "*****"; recipient address
$Subject = "Checking - "&$Title; letter theme
$Body = " Results of the test in the attachment "; body of the letter (the text
of the letter itself)
$AttachFiles = "C:\"& $Title& ".txt"; attached file

$BlatArgs = "-t "&$ToAddress& " -f "&$FromAddress& " -s "&$Subject& "
-body "&$Body& " -server "&$SmtServer& " -u "&$Username& " -pw
"&$Password& " -attach "&$AttachFiles
Run (@scriptdir & "\blat.exe " & $BlatArgs, @ScriptDir, @SW_HIDE )
_SendMail($BlatArgs)

Func _SendMail($CMDstring)
$BlatDLL = DllOpen ("blat.dll")
Do
$result=DllCall($BlatDLL,"int","Send","str",$CMDstring)
Select
Case $result[0] = 12
MsgBox(16, "Error. Error code " & $result[0], " The arguments are not
specified or not found in the registry -server or -f")
Case $result[0] = 3
MsgBox(16, " Error. Error code " & $result[0], "I can't read the file with
the text of the letter or attached file ")
Case $result[0] = 2
MsgBox(16, " Error. Error code " & $result[0], "The mail server did not
like the sender's address; Or he refused to connect us; Or incorrect password; Or
the file with the text of the letter was not found ")
Case $result[0] = 1
MsgBox(16, " Error. Error code " & $result[0], "Can't open SMTP socket
")
Case $result[0] <> 0
MsgBox(16, " Error. Error code " & $result[0], "Unknown error")
EndSelect
Until $result[0] = 0
MsgBox(16, "no error.", "Letter was sent :-)" & @CRLF)
DllClose ($BlatDLL)
EndFunc
```

The letter received by the developers has the following format, shown in Figure B4.

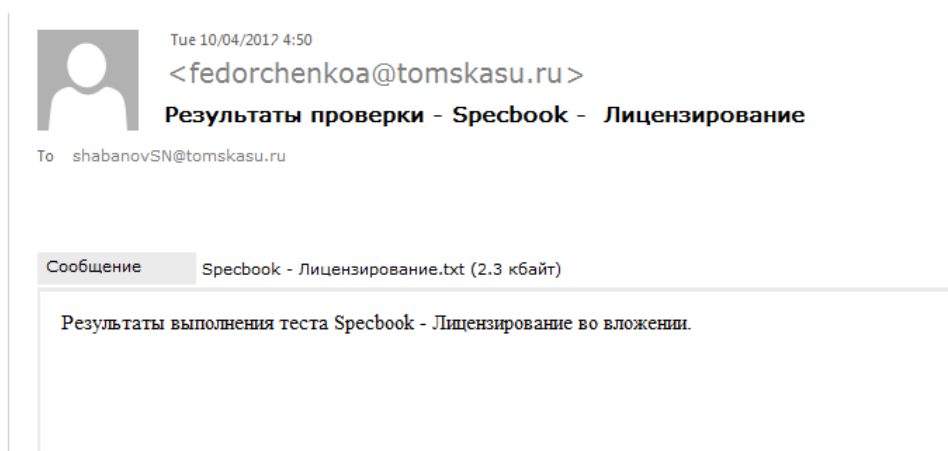


Figure B4 – The scheme of module starts to check the correct operation of the Licensing

From the figure you can see the script used to send out error messages allows filling in the following fields:

The subject of the letter: \$ Subject = "Check Results -" & \$ Title, where \$ Title is the name of the test the report of which is sent.

Message: \$ Body = "Results of running the test" & \$ Title & "in the attachment."

Attached file with report: \$ AttachFiles = "C: \" & \$ Title & ". Txt"

When testing the Licensing module "Specbook" it was necessary to make the following checks:

1. The presence of the installed module "Specbook" on the computer.
2. Availability of the license file.
3. Verify the correctness of the license behavior.

To assess the correctness of licensing, a verification scheme was developed, shown in Figure B5. It simulates the criminal actions of users in an attempt to renew the license.

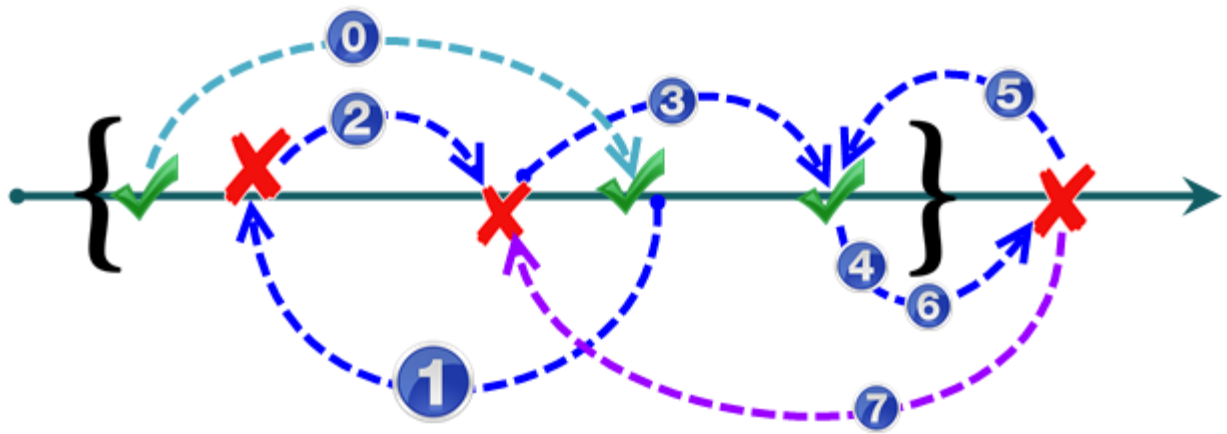


Figure B5 – The scheme of the module starts to verify the correct operation of licensing

Brackets indicate the validity of the license.

Arrows show which way shift the system time.

The check marks indicate after which changes the system should continue to work correctly.

Crosses show after what changes the system should issue a license error.

To simulate the translation of the system time, the user used the function:

`_SetDate (@MDAY, @MON + 1, @YEAR)`

After a single change of time, according to the scheme shown in Figure 9, the program was started, deleted, and reinstalled. If errors occurred during the licensing verification process, information about them was also recorded in the log.

When testing the functionality, according to the requirements, about 250 tests were developed, covering 100% of all functionality.

Functional testing resulted in the following problems:

When checking the correctness of data entry in the table, there was a problem with the initiation of cells. Cells as such were absent, so we had to simulate not pressing the cell, but clicking the mouse on certain coordinates of the window, and then enter the value:

`MouseDown ('left',184,1007,1,10)`

`Send('Statement')`