

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

Институт Кибернетики
Направление подготовки - Информационные системы и технологии
Кафедра Автоматики и Компьютерных Систем

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Разработка системы для автоматизированной сборки дистрибутивов программного обеспечения

УДК 004.42:004.63-047.84

Студент

Группа	ФИО	Подпись	Дата
8ИЗА	Черепанов Иван Сергеевич		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент каф. АиКС	Скирневский Игорь Петрович			

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент каф. Менеджмента	Спицын Владислав Владимирович			

По разделу «Социальная ответственность»

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

ДОПУСТИТЬ К ЗАЩИТЕ:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
АиКС	Суходоев Михаил Сергеевич	К.Т.Н.		

Томск – 2017 г.

Планируемые результаты обучения

Код результатов	Результат обучения (выпускник должен быть готов)
<i>Профессиональные и общепринятые компетенции</i>	
P1	Применять глубокие естественнонаучные и математические знания для решения научных и инженерных задач в области анализа, синтеза, проектирования, производства и эксплуатации средств автоматизации и систем управления техническими объектами.
P2	Уметь обрабатывать, анализировать и обобщать научно-техническую информацию, передовой отечественный и зарубежный опыт в области теории, проектирования, производства и эксплуатации средств автоматизации и систем управления техническими объектами
P3	Ставить и решать инновационные задачи инженерного анализа, связанные с разработкой технических систем управления с использованием аналитических методов и сложных моделей.
P4	Выполнять инновационные инженерные проекты по разработке программно-аппаратных средств автоматизированных систем различного назначения с использованием современных методов проектирования, систем автоматизированного проектирования, передового опыта разработки конкурентно способных изделий.
P5	Планировать и проводить теоретические и экспериментальные исследования в области проектирования аппаратных и программных средств автоматизированных систем с использованием новейших достижений науки и техники, передового отечественного и зарубежного опыта. Критически оценивать полученные данные и делать выводы.
P6	Осуществлять авторское сопровождение процессов проектирования, внедрения и эксплуатации программно-аппаратных средств автоматизированных систем различного назначения.
<i>Универсальные (общекультурные) компетенции</i>	
P7	Владеть иностранным языком на уровне, позволяющем работать в интернациональной профессиональной среде с пониманием культурных, языковых и социально-экономических различий партнеров
P8	Осуществлять коммуникации в профессиональной среде и в обществе в целом, активно владеть иностранным языком, разрабатывать документацию, презентовать и защищать результаты инновационной инженерной деятельности, в том числе на иностранном языке.
P9	Эффективно работать индивидуально и в качестве члена и руководителя группы, в том числе междисциплинарной и международной, при решении инновационных инженерных задач.
P10	Демонстрировать личную ответственность и ответственность за работу возглавляемого коллектива, приверженность и готовность следовать профессиональной этике и нормам ведения инновационной инженерной деятельности. Демонстрировать глубокие знания правовых, социальных, экологических и культурных аспектов инновационной инженерной деятельности.
P11	Демонстрировать способность к самостоятельному обучению, непрерывному самосовершенствованию в инженерной 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 - Информационные системы и технологии
Кафедра Автоматики и Компьютерных Систем

УТВЕРЖДАЮ:

Зав. кафедрой

(Подпись) _____ (Дата) М.С. Суходоев
(Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Бакалаврской работы

Студенту:

Группа	ФИО
8ИЗА	Черепанову Ивану Сергеевичу

Тема работы:

Разработка системы для автоматизированной сборки дистрибутивов программного обеспечения	
Утверждена приказом директора (дата, номер)	№2439/с от 07.04.2017 г.

Срок сдачи студентом выполненной работы:	1.06.2017
--	-----------

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	
<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>	Спроектировать и реализовать информационную систему учета для автоматизированной сборки дистрибутивов лабораторной информационной системы «Химик-аналитик»; система должна обеспечивать функционал необходимый для создания дистрибутивов различных конфигураций системы (хранение и заполнение заявок на сборку, программное скачивание исходного кода, сборка решений, прикрепление вложений, компоновка файлов в дистрибутив, архивирование);

<p>Перечень подлежащих исследованию, проектированию и разработке вопросов</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>	<ol style="list-style-type: none"> 1) Исследование предметной области для построения ее информационной модели; 2) Исследование способов программного получения исходного кода проектов из системы контроля версий; 3) Исследование способов программной сборки решений; 4) Исследование способов программной архивации данных; 5) Проектирование объектно-ориентированной структуры, описывающей предметную область и автоматизируемый процесс
<p>Перечень графического материала</p>	<p>Презентация в формате *.ppt на 15 слайдах.</p>

<p>Консультанты по разделам выпускной квалификационной работы</p>	
<p>Раздел</p>	<p>Консультант</p>
<p>Финансовый менеджмент, ресурсоэффективность и ресурсосбережение</p>	<p>Спицын Владислав Владимирович</p>
<p>Социальная ответственность</p>	<p>Акулов Петр Анатольевич</p>
<p>Названия разделов, которые должны быть написаны на русском и иностранном языках:</p>	
<p>Заключение</p>	

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

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
<p>Ассистент каф. АиКС</p>	<p>Скирневский Игорь Петрович</p>			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
<p>8ИЗА</p>	<p>Черепанов Иван Сергеевич</p>		

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

Институт кибернетики

Направление подготовки – 09.03.02 Информационные системы и технологии

Уровень образования – бакалавриат

Кафедра автоматике и компьютерных систем

Период выполнения – весенний семестр 2017 учебного года

Форма представления работы:

Бакалаврская работа

(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)
	Основная часть	75
	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	15
	Социальная ответственность	10

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент каф. АиКС	Скирневский И.П.			

СОГЛАСОВАНО:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
Зав. каф. АиКС	Суходоев М.С.	К.Т.Н.		

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

Студенту:

Группа	ФИО
8ИЗА	Черепанов Иван Сергеевич

Институт	Институт кибернетики	Кафедра	Автоматики и компьютерных систем
Уровень образования	Бакалавриат	Направление/специальность	Информационные системы и технологии

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

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Работа с информацией, представленной в российских и иностранных научных публикациях, аналитических материалах, статистических бюллетенях и изданиях, нормативно-правовых документах.
2. Нормы и нормативы расходования ресурсов	

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения	Оценка потенциальных потребителей исследования, анализ конкурентных технических решений, SWOT-анализ.
2. Определение возможных альтернатив проведения научных исследований	Определение возможных альтернатив с помощью морфологического подхода.
3. Планирование и формирование бюджета научных исследований	Планирование этапов работ, определение трудоемкости работы и построение календарного графика.

Дата выдачи задания для раздела по линейному графику	06.02.2017
---	------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент каф. Менеджмента	Спицын В.В.			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИЗА	Черепанов Иван Сергеевич		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8ИЗА	Черепанов Иван Сергеевич

Институт	Кибернетики	Кафедра	Автоматики и Компьютерных Систем
Уровень образования	Бакалавриат	Направление/специальность	Информационные системы и технологии

Исходные данные к разделу «Социальная ответственность»:

<i>1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения</i>	<i>Объектом исследования является система для автоматизированной сборки дистрибутивов программного обеспечения.</i>
---	---

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

<p>1. Производственная безопасность</p> <p>1.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности.</p> <p>1.2. Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности.</p> <p>1.3 Рекомендации по минимизации вредных и опасных факторов</p>	<p>1.1 Рассмотрены вредные факторы:</p> <ul style="list-style-type: none"> – отклонение показателей микроклимата; – недостаточная освещенность рабочей зоны. <p>1.2 Рассмотрены опасные факторы:</p> <ul style="list-style-type: none"> – электрический ток; – пожаровзрывобезопасность. <p>1.3 Рекомендации по минимизации вредных и опасных факторов:</p> <ul style="list-style-type: none"> – Рекомендации по режиму труда и отдыха оператора – Мероприятия по оптимизации зрительной работоспособности и световой обстановки – Мероприятия по улучшению состояния воздушной среды рабочих помещений – Мероприятия по снижению шума – Рекомендации при работе с системой по автоматизированной сборке дистрибутивов – Мероприятия по предотвращению причин возникновения пожара – Мероприятия по предотвращению возможности поражения электрическим током
<p>2. Экологическая безопасность:</p> <p>1.1 Анализ воздействия объекта на окружающую среду;</p> <p>1.2 Разработать решения по обеспечению экологической безопасности со ссылками на НТД по охране окружающей среды.</p>	<p>2.1 Рассмотрены негативно влияющие на экологию факторы при эксплуатации компьютера.</p> <p>2.2 Решения по обеспечению экологической безопасности согласно нормативным документам.</p>
<p>3. Безопасность в чрезвычайных ситуациях:</p> <p>3.1 Перечень возможных ЧС при разработке и эксплуатации проектируемого решения;</p> <p>3.2 Разработка действий в результате возникшей ЧС и мер по ликвидации её последствий.</p>	<p>3.1 Перечень возможных ЧС, которые могут возникнуть при работе в помещении офиса.</p> <p>3.2 Способы защиты от пожара и ликвидация последствий.</p>
<p>4. Правовые и организационные вопросы обеспечения безопасности:</p>	<p>4.1 Организационные мероприятия по обеспечению безопасности трудящихся за персональным компьютером.</p>

4.1 Организационные мероприятия при компоновке рабочей зоны. 4.2 Специфика влияния продукта на рабочий процесс.	4.2 Влияние разработанной системы для анализа и управления сетевым трафиком на рабочий процесс
--	--

Дата выдачи задания для раздела по линейному графику	06.02.2017
---	------------

Задание выдал консультант:

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

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИЗА	Черепанов Иван Сергеевич		

РЕФЕРАТ

Выпускная квалификационная работа 102 с., 35 рис., 11 табл., 20 источников, 9 прил.

Ключевые слова: информационная система, Microsoft .Net, дистрибутив, автоматизация бизнес-процесса, Team foundation server, Windows Presentation Foundation.

Объектом исследования является деятельность предприятия ООО «Химсофт», его внутренние бизнес-процессы по созданию дистрибутивов лабораторной информационной системы «Химик-аналитик».

Цель работы – разработка системы для автоматизированной сборки дистрибутивов программного обеспечения.

В процессе исследования проводился теоретический анализ средств программного скачивания исходного кода, сборки решений, архивации. Также были произведены проектирование и разработка информационной системы.

В результате исследования была спроектирована и разработана информационная система для автоматизированной сборки дистрибутивов программного обеспечения.

Основные конструктивные, технологические и технико-эксплуатационные характеристики: система реализована в среде .Net на платформе WPF и обладает всеми заявленными функциями.

Степень внедрения: опытная эксплуатация.

Область применения: автоматизированное создание дистрибутивов лабораторной информационной системы «Химик-аналитик».

Экономическая эффективность/значимость работы выражается в оптимизации временных затрат на создание дистрибутивов. Рабочее место пользователя удовлетворяет стандартам и нормам безопасности.

В будущем планируется доработка информационной системы согласно пожеланиям пользователей.

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

GUI – Graphical User Interface

TFS – Team Foundation Server

WPF – Windows Presentation Foundation

XAML – eXtensible Application Markup Language

JSON – JavaScript Object Notation

DLL – Dynamic Link Library

BPMN – Business Process Model and Notation

UML – Unified Modeling Language

ЛИС – Лабораторная информационная система

ООО – Общество с ограниченной ответственностью

SWOT – Strengths Weaknesses Opportunities Threats

ПЭВМ – Персональная электронно-вычислительная машина

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	13
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	15
1.1 Актуальность	15
1.2 Постановка проблемы	17
1.3 Анализ существующих решений	18
2 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ.....	24
2.1 Выявление функциональных требований	24
2.2 Технологии и средства разработки	26
2.3 Средства хранения данных	28
3 ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ	30
3.1 Обработка «заявок на сборку»	30
3.2 Автоматизированное получение исходного кода.....	38
3.3 Программная сборка решений	42
3.4 Прикрепление дополнительных файлов	43
3.5 Архивация	45
3.6 Логирование	48
4 ДЕМОНСТРАЦИЯ РАБОТЫ ПРИЛОЖЕНИЯ	49
4.1 Обработка «заявок на сборку»	49
4.2 Получение дистрибутивов по «заявкам на сборку».....	53
5 Финансовый менеджмент.....	59
5.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения	59
5.1.1 Потребители результатов исследования.....	59
5.1.2 Анализ конкурентных технических решений	59
5.1.3 SWOT-анализ	61
5.2 Определение возможных альтернатив научных исследований.....	63
5.3 Планирование научно-исследовательских работ	64
5.3.1 Структура работ в рамках научного исследования	64
5.3.2 Определение трудоемкости выполнения работ	65
5.3.3 Разработка графика проведения научного исследования	66
6 СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ	67
6.1 Производственная безопасность	68
6.1.1 Вредные производственные факторы	69
6.1.1.1 Отклонение показателей микроклимата в помещении	69
6.1.1.2 Недостаточная освещенность рабочей зоны	71
6.1.2 Опасные производственные факторы.....	72

6.1.2.1	Опасность поражения электрическим током	72
6.1.2.2	Пожаровзрывобезопасность	73
6.1.3	Рекомендации по минимизации вредных и опасных факторов	74
6.1.3.1	Рекомендации по режиму труда и отдыха оператора	74
6.1.3.2	Рекомендуемые мероприятия по оптимизации зрительной работоспособности и световой обстановки	75
6.1.3.3	Мероприятия по улучшению состояния воздушной среды рабочих помещений	76
6.1.3.4	Мероприятия по снижению шума	77
6.1.3.5	Рекомендации при работе с системой по автоматизированной сборке дистрибутивов.	77
6.1.3.6	Мероприятия по предотвращению возможности поражения электрическим током.	78
6.1.3.7	Мероприятия по предотвращению причин возникновения пожаров	79
6.2	Экологическая безопасность	80
6.2.1	Анализ воздействия продукта на окружающую среду	80
6.2.2	Решения по обеспечению экологической безопасности	80
6.3	Безопасность в чрезвычайных ситуациях	81
6.3.1	Перечень возможных ЧС при разработке и эксплуатации проектируемого решения	81
6.3.2	Разработка действий в результате возникшей ЧС и мер по ликвидации её последствий	82
6.4	Правовые и организационные вопросы обеспечения безопасности	84
6.4.1	Организационные мероприятия при компоновке рабочей зоны	85
6.4.2	Специфика влияния продукта на рабочий процесс	86
	ЗАКЛЮЧЕНИЕ	88
	Список использованных источников	89
	ПРИЛОЖЕНИЕ А	91
	ПРИЛОЖЕНИЕ Б	92
	ПРИЛОЖЕНИЕ В	94
	ПРИЛОЖЕНИЕ Г	96
	ПРИЛОЖЕНИЕ Д	98
	ПРИЛОЖЕНИЕ Е	99
	ПРИЛОЖЕНИЕ Ж	100
	ПРИЛОЖЕНИЕ З	101
	ПРИЛОЖЕНИЕ И	102

ВВЕДЕНИЕ

В условиях рыночной конкуренции предприятиям для процветания и получения прибыли приходится максимально оптимизировать свою деятельность. Автоматизация бизнес процессов является эффективным способом снижения издержек и позволяет экономить время ценных сотрудников, упрощая выполнение часто повторяющихся рутинных задач.

На предприятии ООО «Химсофт» существует задача автоматизации – сборка дистрибутивов лабораторной информационной системы «Химик-аналитик». Сложность процесса заключается в том, что у предприятия имеется большое количество заказчиков разной специализации, эксплуатирующих лабораторную систему – от небольших химических лабораторий до атомных электростанций.

У различных заказчиков используются различные СУБД, модули и конфигурации лабораторной системы. По этой причине на предприятии инструкция по сборке дистрибутива составляет последовательность из более чем двадцати действий. Их выполнение у неподготовленного работника может занять множество часов и потребовать массу разъяснений. Также стоит отметить, что сборка дистрибутивов осуществляется не только при внедрении системы «Химик-аналитик» на предприятие, но и при обновлении заказчиков, заключивших договор на сопровождение.

Проанализировав отчетность по трудозатратам, руководство предприятия решило автоматизировать процесс создания дистрибутивов путем создания системы по программной сборке дистрибутивов.

Актуальность данной работы заключается в том, что разработанный программный продукт позволит сократить трудозатраты на создание дистрибутивов, освободит программистов предприятия от рутинной деятельности. Также автосборщик дистрибутивов позволит упростить документооборот на предприятии. Основанием для сборки дистрибутива является документ «Заявка на сборку», ранее составляемая в текстовом редакторе MS Word. Поля заявки были перенесены в разрабатываемый продукт,

позволяющий ускорить документирование за счет выпадающих списков и различных шаблонов заполнения.

Целью данной работы является разработка системы для автоматизированной сборки дистрибутивов лабораторной информационной системы «Химик-аналитик».

Разработанная информационная система, являющаяся результатом данной работы должна предоставлять следующие возможности:

- Обработка полей «Заявки на сборку», ее сохранение и загрузка;
- Программное создание директорий;
- Дополнение дистрибутива дополнительными файлами и вложениями, документацией, инструкциями, патчами, текстом лицензии, утилитами;
- Программное скачивание исходного кода из системы контроля версий;
- Программное изменение конфигурационных файлов;
- Программная сборка библиотек и запускаемых файлов из исходного кода;
- Программная архивация в CAB архив и защищенный RAR архив;
- Логирование процесса и предоставление данных о возникших ошибках при их наличии;
- Текстовое отображение процесса сборки дистрибутива

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Актуальность

Для сборки дистрибутивов существуют готовые решения, позволяющие создать инсталлятор в несколько кликов. В связи с этим возникает следующий вопрос, почему для достижения поставленной цели потребовалось разрабатывать информационную систему?

Некоторые существующие программные приложения по сборке инсталляторов обладают бесплатной лицензией, также они не требуют особых профессиональных знаний для их применения и достаточно просты и удобны в использовании. При этом сам процесс сборки проходит достаточно быстро и без нареканий. Однако любые распространяемые программные средства не могут учитывать особенностей бизнес процесса предприятия, особенностей системы, чью сборку требуется получить.

Невозможность применения существующего ПО со схожим функционалом для автоматизации процесса связана в первую очередь с двумя проблемами. Первая состоит в том, что для реализации защиты от злоумышленников, а также для сборки системы определенной конфигурации требуется изменение исходных кодов этой системы. Также ООО «Химсофт» обслуживает достаточно большое количество заказчиков, чтобы не представлялось рациональным реализовать хранение конфигурации системы для каждого из заказчиков в отдельной ветке репозитория с исходным кодом. По той причине, что сторонние разработки не работают с исходным кодом системы, они не могут его изменять. Таким образом сторонние разработки не могут обеспечить должную степень автоматизации процесса.

Состав дистрибутива полностью устанавливается содержимым документа «заявка на сборку», который является обязательным основанием для создания дистрибутива. Автоматизированием изменений исходного кода в зависимости от заполнения полей «заявки на сборку» решается множество проблем. Обеспечение защиты и настройка конфигурации вручную оказывали

наибольшую умственную нагрузку, а также требовали наличия профессиональных знаний.

Вторая важная проблема сторонних сборщиков инсталляторов заключается в следующем – им требуется вручную указывать файлы [12][13], которые должны быть развернуты при инсталляции. Это означает, что работник должен скачать последние версии исходного кода проектов, используемые в устанавливаемой системе, а также произвести сборку этих исходников, затем указать инсталлятору на собранные файлы. Эти процессы должен выполнять создатель дистрибутива вручную, чего требуется избежать.

Однако, современные системы непрерывной интеграции обладают возможностью задавать директорию размещения результатов сборки проектов [14]. Также при настройке сервера непрерывной интеграции, сборка будет производиться на сервере автоматически при внесении изменений в исходный код. При этом по-прежнему нерешенной остается проблема выбора только необходимых файлов для сборки дистрибутива. Например, при поставке ЛИС «Химик-аналитик» на АЭС и в химическую лабораторию, предоставляемые дистрибутивы будут сильно отличаться. Также не ясно каким образом производить поставку системы максимальной и минимальной конфигурации. Скорее всего для решения этой проблемы требуется ведение диалога с оператором ПЭВМ при создании дистрибутива. Это означает что потребуются дублирование ввода данных – ведь заявка на сборку к этому моменту уже должна быть заполнена. На этом этапе анализа был сделан вывод о том, что использование сторонних разработок не позволяет достичь нужной степени автоматизации процесса.

Стоит отметить еще раз, что документ «заявка на сборку» полностью определяет состав дистрибутива. Разработанный автосборщик дистрибутивов позволяет получить лишь необходимые файлы автоматически на основе заполненных полей документа. Таким образом, можно сказать, что разработанная система автоматически подбирает необходимые файлы для формирования дистрибутива в зависимости от требуемой конфигурации

устанавливаемой системы, описанной в заявке, все поля которой перенесены в интерфейс автосборщика дистрибутивов.

Высокая степень автоматизации рассматриваемого процесса может быть достигнута лишь с применением ПО, учитывающим его особенности и специфику. Разрабатываемая система без доработок не применима для сборки дистрибутивов любых проектов, однако для сборки дистрибутивов ЛИС «Химик-аналитик» она является лучшим вариантом. Это достигается за счет высокой степени специализации и глубокого анализа бизнес-процессов предприятия.

1.2 Постановка проблемы

Согласно отчетам о трудозатратах за 2016 год, ведущий программист ООО «Химсофт» потратил около двух месяцев рабочего времени на выполнение задач по сборке дистрибутивов ЛИС «Химик-аналитик». Такой большим объемом работ объясняется выходом крупного обновления поставляемой системы.

Предприятие заботится о себе и своих заказчиках, и руководство старается добиться обновления подавляющего большинства своих продуктов у их пользователей. С новыми версиями ЛИС становится все более стабильна, устойчива и проста в использовании и сопровождении. Очень часто при обновлении добавляется новый востребованный функционал. Обновления системы, как и ее поставка, являются важным источником дохода предприятия.

Для обновления ЛИС часто является недостаточно применения патчей. Поэтому приходится переустанавливать систему. При этом конечно же все наработки в БД заказчика сохраняются. Переустановка системы, как и ее поставка требует создания дистрибутива на основании документа «заявка на сборку».

Процесс сборки дистрибутивов занимает много времени, так как разные заказчики используют разные СУБД, приобретают систему с различной конфигурацией. Иногда для заказчиков разрабатываются уникальные модули системы, также заказчикам требуются разные наборы вспомогательных утилит.

Все это приводит к значительному усложнению процесса сборки и увеличению затрат времени на него. Размер инструкции по сборке значительно вырос, пополнился как рутинными операциями, так и сложными для выполнения пунктами. Также созданные дистрибутивы требуется протестировать перед передачей. В связи с вышеперечисленным, массовые обновления заказчиков стали приводить к большим трудозатратам и невозможности переключения на другие задачи.

Таким образом во время выполнения обновления системы ЛИС «Химик-аналитик» в 2016 году было выявлено узкое место в организации работы предприятия, которое требуется устранить.

1.3 Анализ существующих решений

На рынке существует множество решений, позволяющих создавать инсталляторы различной сложности. В данном пункте будут рассмотрены самые популярные из них.

Одной из альтернатив является приложение **Actual Installer**, которое позволит достаточно легко создавать установочные программы для Windows приложений. Программа позволяет за короткий промежуток времени создать полноценную одно-файловую программу готовую для дистрибуции.

Созданная установочная программа будет обладать всеми базовыми возможностями инсталлятора: от многоязычной установки до регистрации шрифтов в операционной системе. Также, Actual Installer позволяет защитить установочные программы с помощью пароля.

Основные возможности программы:

- Создание одно-файловой установочной программы
- Компрессия файлов
- Функция деинсталляции программы
- Создание ярлыков
- Модификация Реестра и INI-файлов
- Показ лицензии и справки в формате TXT или RTF

- Автоматический запуск файлов либо перезагрузка компьютера после инсталляции
- Добавление записей в раздел "Установка и Удаление программ" в Панели Управления

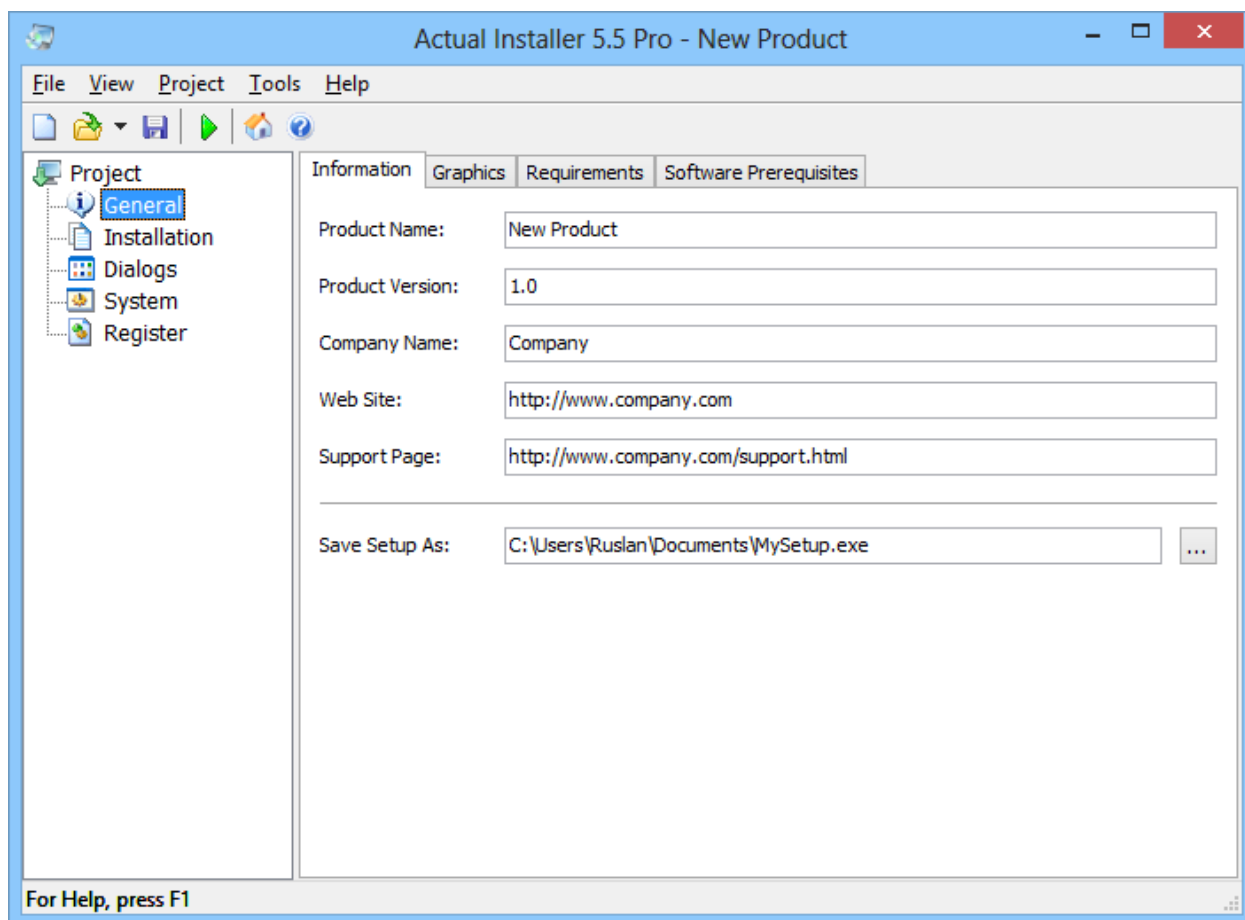


Рисунок 1 - Интерфейс приложения Actual Installer

Inno Setup - система создания инсталляторов для Windows программ с открытым исходным кодом. Впервые выпущенный в 1997 году, Inno Setup сегодня конкурирует и даже превосходит многие коммерческие системы по функциональности и стабильности.

К сожалению, интерфейс пользователя Inno Setup не является удобным. Однако существует несколько программ-оболочек, предназначенных для упрощения и ускорения написания кода, например, Inno Script Studio.

Ключевые особенности:

- Поддержка всех современных версий Windows: 10, 8, 7, 2008 R2, Vista, XP, 2003, 2000, 9x, NT 4.0. Также поддерживает установку 64-битных программ на 64-битных выпусках ОС.
- Поддерживает создание одно-файловых инсталляторов.
- Возможность выбора типа установки: полный, минимальный, выборочный.
- Поддержка DEFLATE, bzip2 и 7-Zip LZMA/LZMA2 сжатий.
- Сравнение версии файлов, установка разделяемых файлов.
- Регистрация DLL/OCX библиотек и установка шрифтов.
- Создание ярлыков в меню «Пуск» и на «Рабочем столе».
- Создавать записей в реестре и .ini-файлов.
- Поддержка скриптов на Object Pascal (посредством Pascal Script).
- Поддерживает создание многоязычных инсталляторов программ.
- Поддерживает установку пароля и шифрование инсталляторов программ.
- Предоставляет доступ к исходному коду на Borland Delphi 2.0-5.0 и 2009.

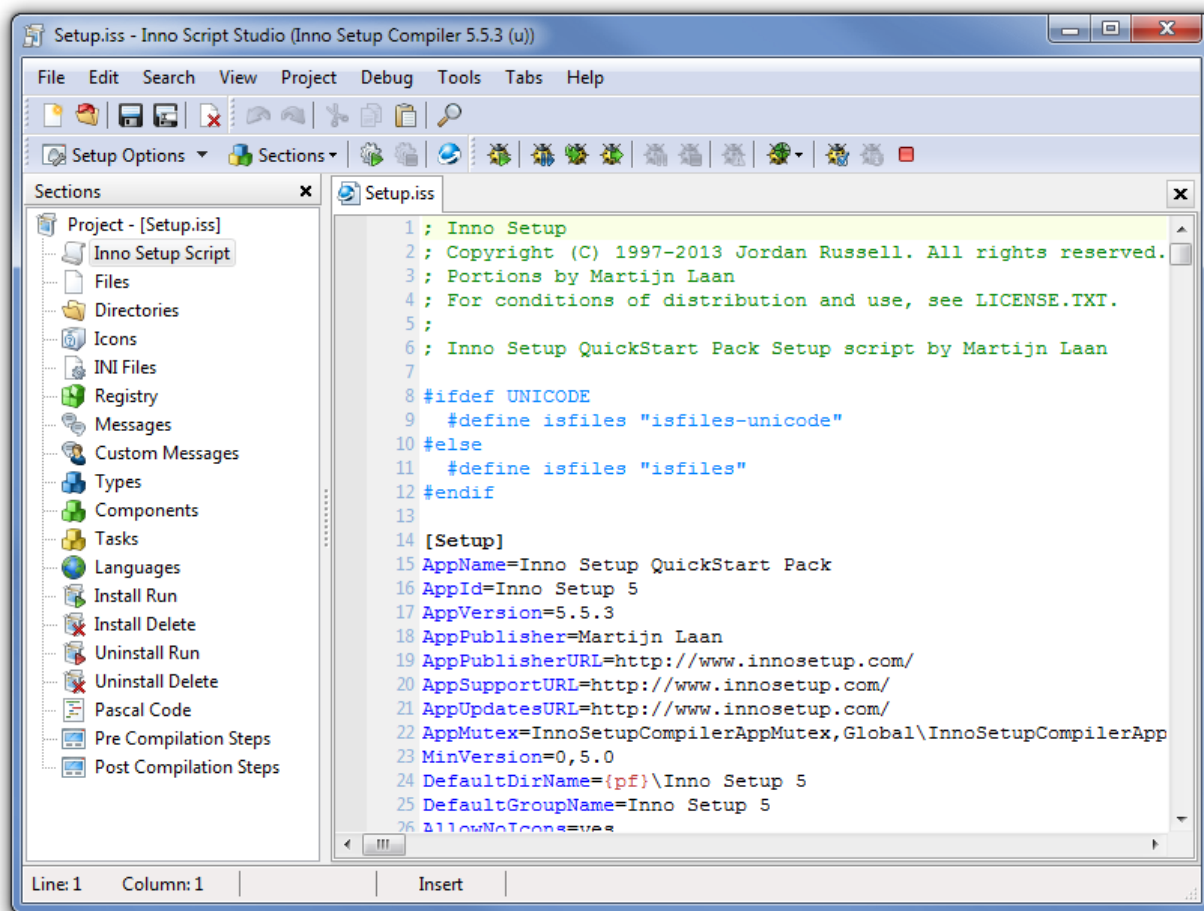


Рисунок 2 - Интерфейс приложения Inno Setup

CreateInstall – система создания инсталляторов, позволяющая наиболее полно контролировать процесс инсталляции. Весь сценарий установки представляется в виде последовательности команд. Пользователь может использовать как уже готовые команды, так и самостоятельно создавать любые необходимые сценарии.

Использование универсального языка программирования Gentee в качестве скриптового языка установок, существенно расширяет возможности создаваемых установочных программ. Пользователь получает возможность полностью управлять процессом установки и деинсталляции. Также пользователю предоставляется функционал по модифицированию диалоговых окон, добавлению собственных ресурсов, графических элементов.

Одной из составляющих инсталлятора являются используемые им алгоритмы сжатия. Пользователю предоставляется выбор между двумя

базовыми алгоритмами сжатия. Один из них обеспечивает хорошую степень компрессии, а другой высокую скорость распаковки файлов. При разработке возможно использование обоих алгоритмов в одной инсталляции для разных групп файлов, что позволяет производить оптимальную установку.

CreateInstall является сложной для освоения программой. Однако разработчики заявляют, что ее возможности по гибкой настройке процесса установки окупают время, затраченное на изучение.

Для создания простых инсталляторов при CreateInstall предоставляет дополнительное приложение CreateInstall Assistant. Оно имеет сильно упрощенный интерфейс и позволяет быстро создавать дистрибутивы без освоения основной программы.

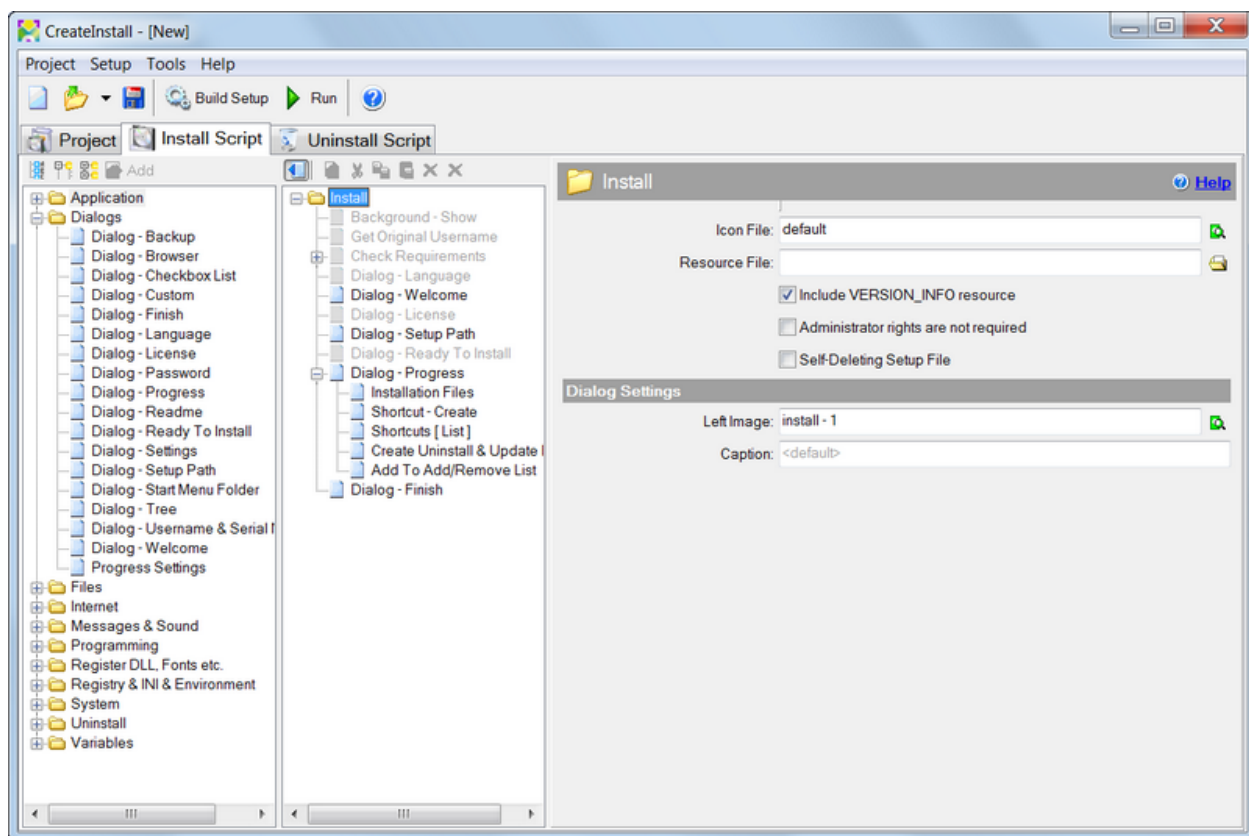


Рисунок 3 - Интерфейс приложения CreateInstall

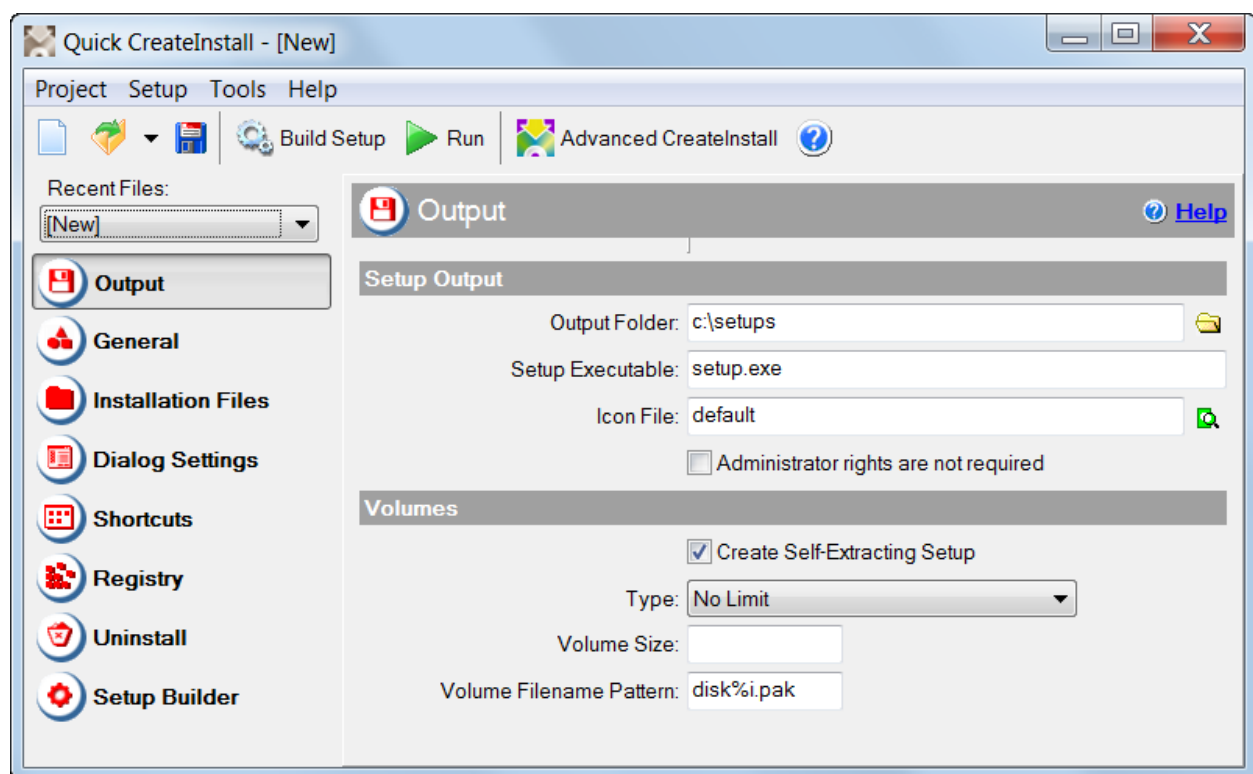


Рисунок 4 - Интерфейс приложения CreateInstall Assistant

По функциональным возможностям и расширяемости CreateInstall опережает конкурентов. Однако для реализации потенциала программы требуется длительное обучение, прочтение документации и изучение языка программирования Gentee.

При анализе актуальности разработки собственной системы по автоматизированной сборке дистрибутивов в пункте 1.1 были выявлены слабые стороны использования стороннего ПО. Данные программы требуют покупки лицензии, изучения их функционала, написания и отладки скриптов, хотя при этом не достигается нужная степень автоматизации.

2 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

2.1 Выявление функциональных требований

В данном пункте будут приведены поясняющие диаграммы, отражающие суть автоматизируемого процесса, а также структуру пользовательских классов, описывающих предметную область. Для достижения наглядности некоторые детали бизнес-процессов на диаграммах нотации BPMN были опущены, однако основная суть процессов осталась неизменной.

Большинство дистрибутивов ЛИС были собраны при обновлении заказчиков. На один заказ поставки ЛИС приходится около пяти ее обновлений. Бизнес-процесс обновления пользователей ЛИС «Химик-аналитик» представлен на рисунке 5.

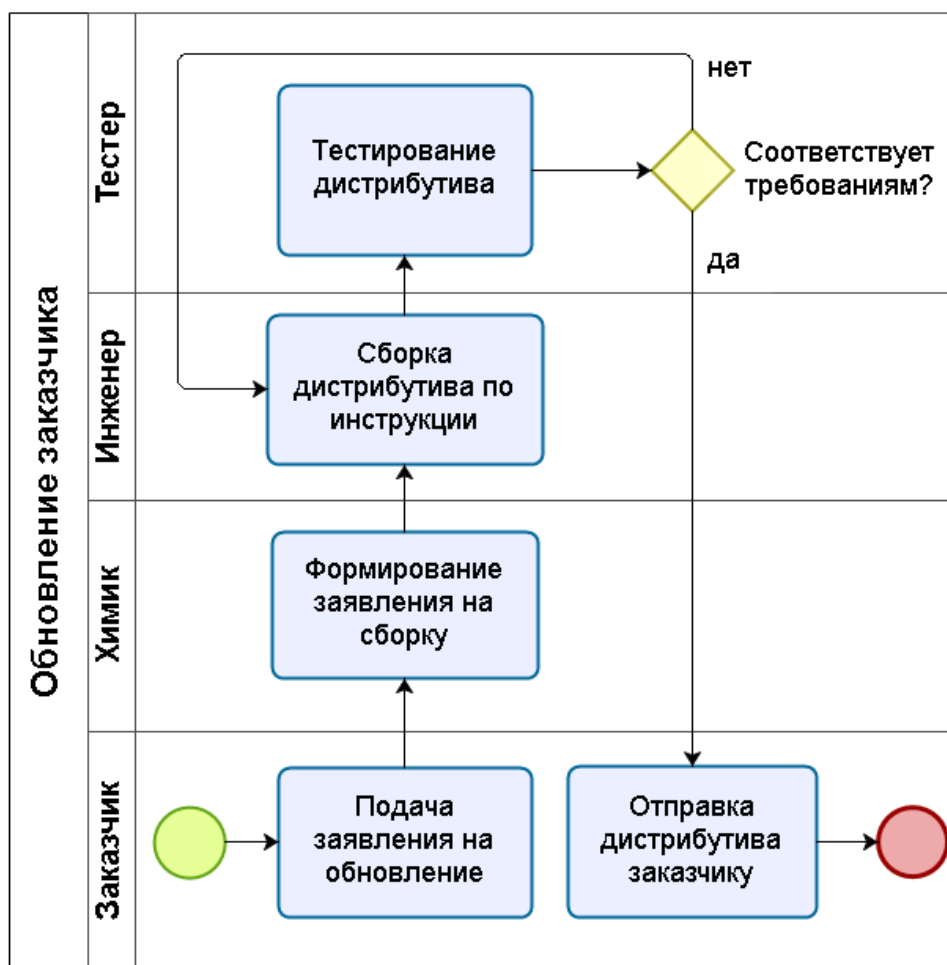


Рисунок 5 – BPMN диаграмма бизнес-процесса обновления заказчика

Большинство из составляемых дистрибутивов являются типовыми, и для их сборки можно применять разработанную на предприятии инструкцию. На рисунке 6 приведена диаграмма процесса сборки дистрибутива по инструкции.

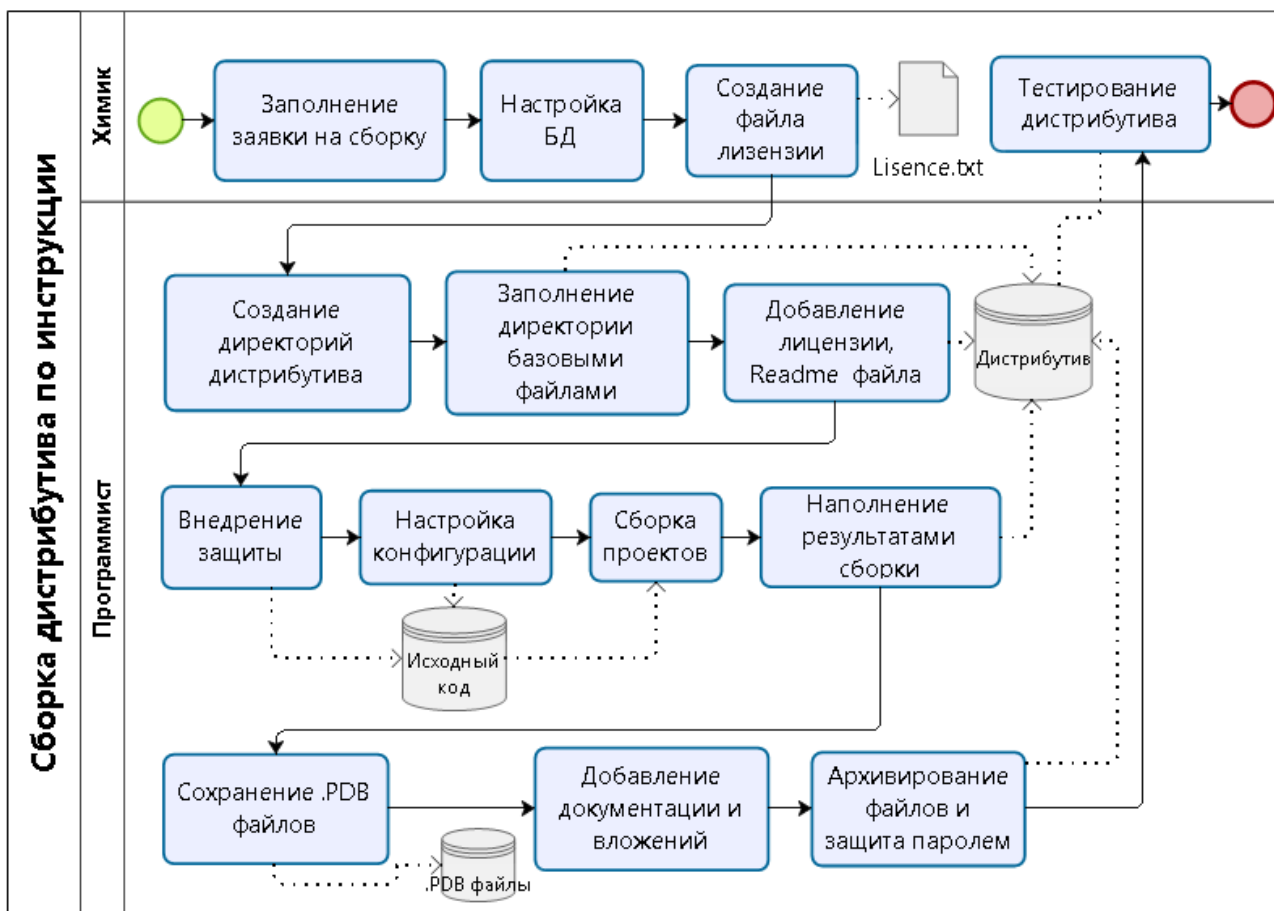


Рисунок 6 - BPMN диаграмма бизнес-процесса сборки дистрибутива

При анализе вышеописанного бизнес-процесса было установлено, что его можно автоматизировать, при этом сократив трудозатраты. Большинство подзадач, отраженных на диаграмме, можно выполнить программно, при наличии заполненной «заявки на сборку». Исходя из этого были выявлены функциональные требования для системы автоматизированной сборки дистрибутивов. В них входят следующие функции:

- Сохранение, загрузка и редактирование заявок на сборку, открытие последней активной заявки при начале работы;
- Программная загрузка исходного кода проектов из системы контроля версий, реализованной на TFS;

- Программная сборка решений и проектов;
- Программный перенос результатов сборки в состав дистрибутива;
- Автоматическое создание директорий, прикрепление текстового файла-лицензии, генерация «readme» файла;
- Автоматическая архивация ресурсов дистрибутива в CAB архив;
- Автоматическая архивация готового дистрибутива в защищенный RAR архив;
- Прикрепление дополнительных файлов к дистрибутиву;
- Отображение выполнения этапов и задач процесса сборки дистрибутива в реальном времени;
- Логирование ошибок на протяжении всего процесса, уведомление о некорректной сборке.

2.2 Технологии и средства разработки

В связи с тем, что для достижения поставленной цели требуется решать сложные специфические в качестве языка программирования был выбран C#. Строгая типизация языка позволяет минимизировать шанс допущения ошибок, что является большим плюсом при разработке проектов с наличием высокой степенью ответственности за полученный продукт.

C# предоставляет Программисту выбор из огромного количества подключаемых библиотек. Данный выбор оказался удачным - с помощью изучения и использования сторонних библиотек были решены самые трудные задачи данной работы.

Данный язык программирования достаточно распространен и имеет качественную документацию, поддерживаемую корпорацией Microsoft. Более того, C# использовался на протяжении всего периода обучения в высшем учебном заведении. Таким образом знание языка, его строгая типизация, наличие качественного справочного материала и множества подключаемых библиотек стали основанием для его выбора в качестве основного языка разработки.

Система автоматизированной сборки была реализована на платформе Windows Presentation Foundation. Windows Presentation Foundation (WPF) – система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML.

WPF предоставляет более широкие возможности для создания интерфейса, нежели ее альтернатива – платформа Windows Forms. Однако использование WPF сопряжено с некоторыми трудностями – требуется изучение языка XAML для создания графического интерфейса приложения. Также создание интерфейса методом перетаскивания (Drag and Drop) часто приводит к нежелательным результатам. Для создания качественного интерфейса его приходится полностью прописывать вручную на языке XAML.

После реализации можно с уверенностью утверждать, что при использовании платформы Windows Forms время, затраченное на разработку, было бы существенно меньше. При этом возможностей Windows Forms полностью хватило бы для отображения GUI. Однако реализация на платформе WPF являлась требованием заказчика – ООО «Химсофт», также это было прописано в техническом задании.

По завершению разработки интерфейса был освоен XAML, механизмы привязки интерфейса к объектам – «bindings», а также разные методы компоновки элементов интерфейса с использованием множества контейнеров.

В качестве среды разработки была выбрана Microsoft Visual Studio 2015. Именно эта среда разработки используется в организации ООО «Химсофт». В данной среде используется Team Foundation Server с настроенной системой контроля версий, необходимой при разработке средних и крупных проектов.

Обоснование выбора технологий и способов программного архивирования, выгрузки исходного кода и сборки представлено в соответствующих пунктах главы номер 3.

2.3 Средства хранения данных

После анализа функциональных требований было установлено, что для их реализации не требуется создание БД. Данные о документах-заявках хранятся в текстовых файлах. Их создание производится при помощи сериализации, а загрузка заявок в систему осуществляется при помощи десериализации.

Для облегчения осуществления этих операций применяется подключаемая библиотека Newtonsoft.Json. Данная библиотека содержит производительные методы, которые просты в использовании. Сериализацию можно произвести в одну строку исходного кода. Пример использования библиотеки приведен на рисунке 7.

```
public void Serialize(string saveFilePath, Options options)
{
    options.userOptions.LastUsedDocPath = saveFilePath;

    string data = JsonConvert.SerializeObject(this, Formatting.Indented);
    using (StreamWriter sw = new StreamWriter(saveFilePath, false, Encoding.GetEncoding(1251)))
    {
        sw.Write(data);
    }
}

public static TaskOptions DeserializeFromPath(string path)
{
    string json = File.ReadAllText(path, Encoding.GetEncoding(1251));
    TaskOptions loadedFromFile = JsonConvert.DeserializeObject<TaskOptions>(json);
    return loadedFromFile;
}
```

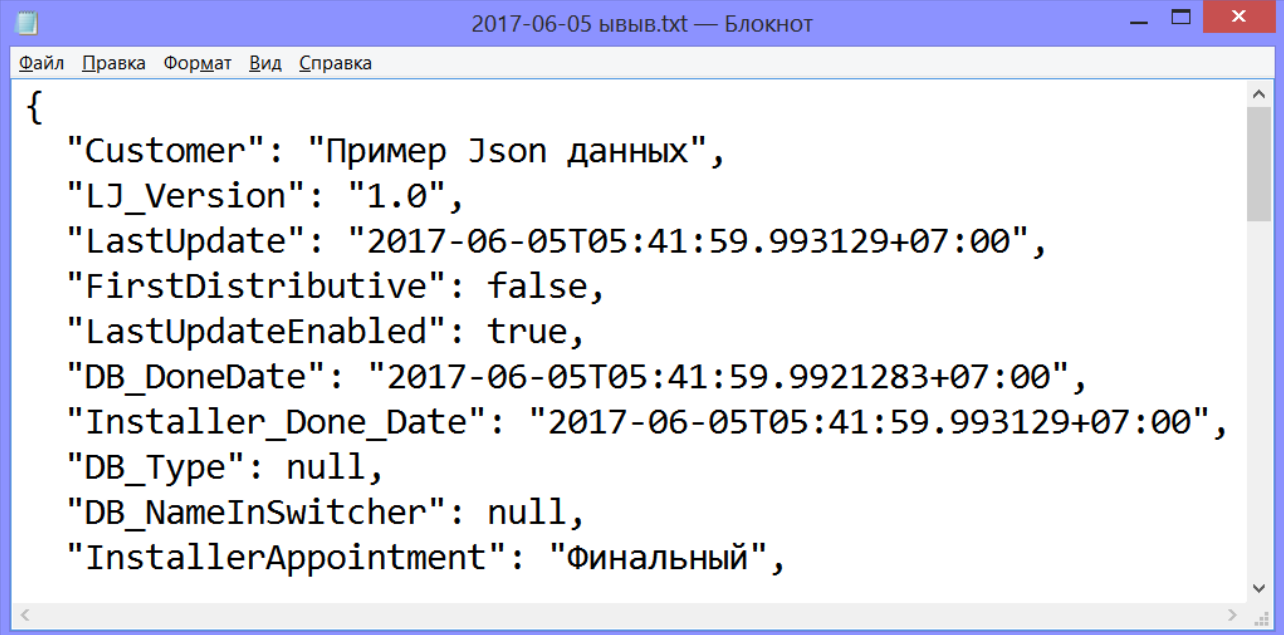
Рисунок 7 - Сериализация и десериализация

Данная библиотека, как и многие другие сторонние используемые библиотеки была подключена с использованием системы управления пакетами для платформ разработки Microsoft - NuGet.

В данной работе производится сериализация не только заявок, но и пользовательской конфигурации системы. Таким способом сохраняются, например, размеры окон и последний активный документ-заявка.

Пример сериализованной информации внутри текстового файла представлен на рисунке 8. Newtonsoft.Json позволяет сохранять информацию с

использованием кодировок для корректного отображения русскоязычного текста и системы отступов для понятной визуализации структуры данных.

A screenshot of a Notepad window titled "2017-06-05 ывыв.txt — Блокнот". The window contains a JSON object with the following fields: "Customer" (Russian text), "LJ_Version", "LastUpdate" (ISO 8601 timestamp), "FirstDistributive" (boolean), "LastUpdateEnabled" (boolean), "DB_DoneDate" (ISO 8601 timestamp), "Installer_Done_Date" (ISO 8601 timestamp), "DB_Type" (null), "DB_NameInSwitcher" (null), and "InstallerAppointment" (Russian text). The JSON is formatted with indentation for readability.

```
{  
  "Customer": "Пример Json данных",  
  "LJ_Version": "1.0",  
  "LastUpdate": "2017-06-05T05:41:59.993129+07:00",  
  "FirstDistributive": false,  
  "LastUpdateEnabled": true,  
  "DB_DoneDate": "2017-06-05T05:41:59.9921283+07:00",  
  "Installer_Done_Date": "2017-06-05T05:41:59.993129+07:00",  
  "DB_Type": null,  
  "DB_NameInSwitcher": null,  
  "InstallerAppointment": "Финальный",  
}
```

Рисунок 8 - Хранение данных в формате JSON

3 ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ

3.1 Обработка «заявок на сборку»

Данные о заявке представляют поля интерфейса главного окна разрабатываемой системы. В приложении Е приведен пример документа, используемого в качестве заявки на сборку до внедрения системы.

Для того, чтобы не перегружать интерфейс приложения, заявка на сборку была разбита на 4 логических блока, которые отображаются в соответствующих вкладках автосборщика:

- Данные о заказчике
- Состав ЛИС
- Состав дистрибутива
- Задача и исполнители

На вкладке «Данные о заказчике» представлены поля для описания предприятия заказчика, например, официальное наименование и контактные данные представителей организации. Подробные данные о наименовании заказчика необходимы для корректной генерации документов в ЛИС. Данные о последней дате обновления необходимы для ведения статистики. При первом внедрении системы отмечается флажок «первая сборка» и блокируется поле «дата последнего обновления». Интерфейс вкладки представлен на рисунке 9.

Autobuilder Заявка на сборку

Файл Настройки

Данные о заказчике Состав ЛИС Состав дистрибутива Задача и исполнители Защита

Наименование заказчика

Дата последнего обновления 05.06.2017 15 Первая сборка

Точное имя владельца версии

Полное

Обычное

Краткое

Сжатое

Наличие города

Пожелания к наименованию владельца версии

Контактный e-mail, телефон, ФИО заказчика

Далее Сохранить заявку Загрузить заявку К сборке

Рисунок 9 – Вкладка «Данные о заказчике»

Вкладка «Состав ЛИС» агрегирует набор флагов для настройки внутреннего состава поставляемой системы. Флаги группы «Конфигурация ЛИС» определяют какие участки исходного кода будут задействованы при сборке системы. Таким образом обеспечивается разнообразное функционирование системы в одинаковых условиях, либо разблокировка дополнительного функционала. Флаги группы «Модули ЛИС» определяют какие дополнительные

вспомогательные программы и утилиты будут включены в дистрибутив. Интерфейс вкладки «Состав ЛИС» представлен на рисунке 10.

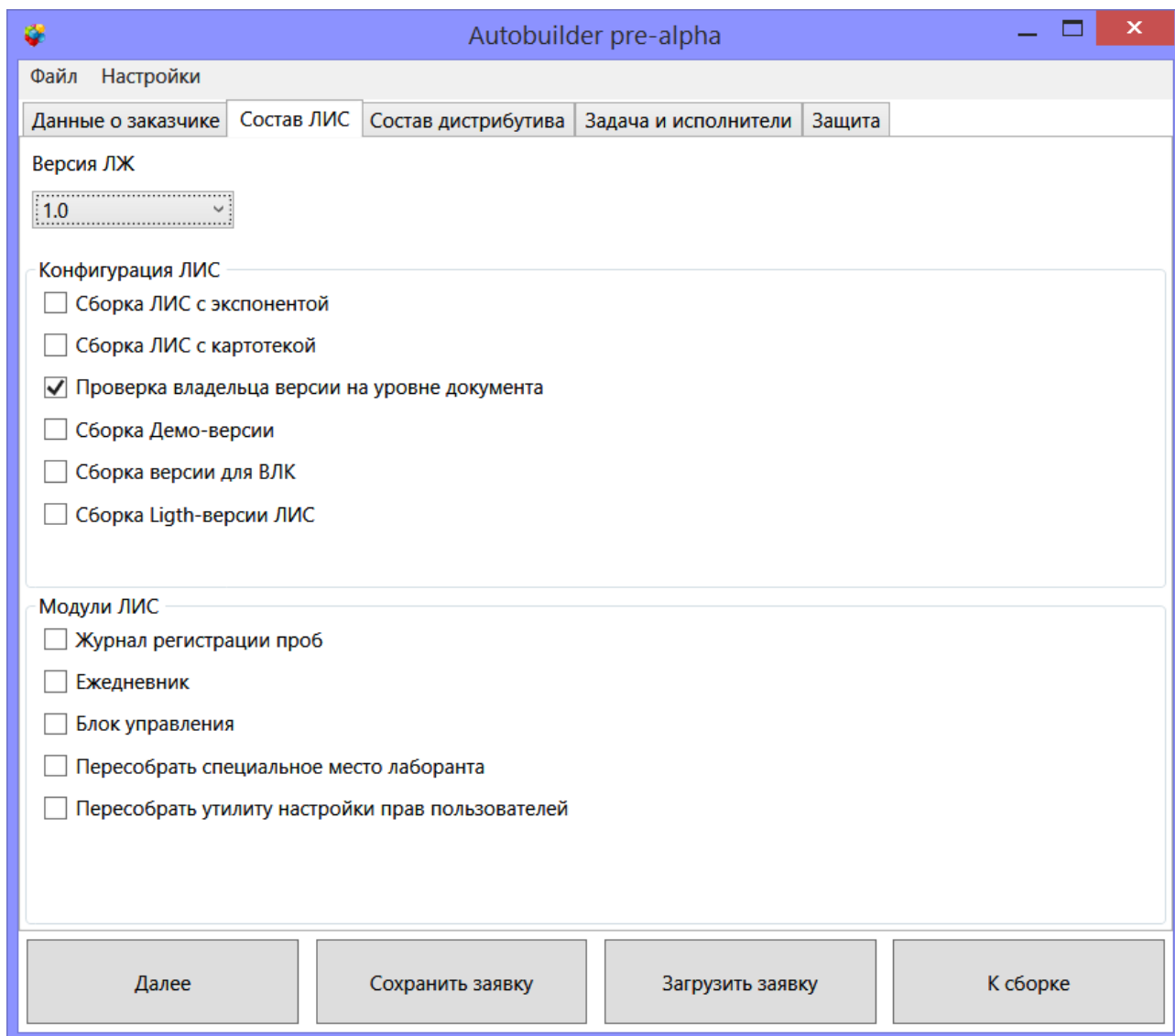


Рисунок 10 –Вкладка «Состав ЛИС»

На рисунке 11 представлен интерфейс вкладки «Состав дистрибутива». Данный блок интерфейса позволяет пользователям включать настроенную для заказчика БД, файл лицензии, документацию и прочие файлы в состав дистрибутива.

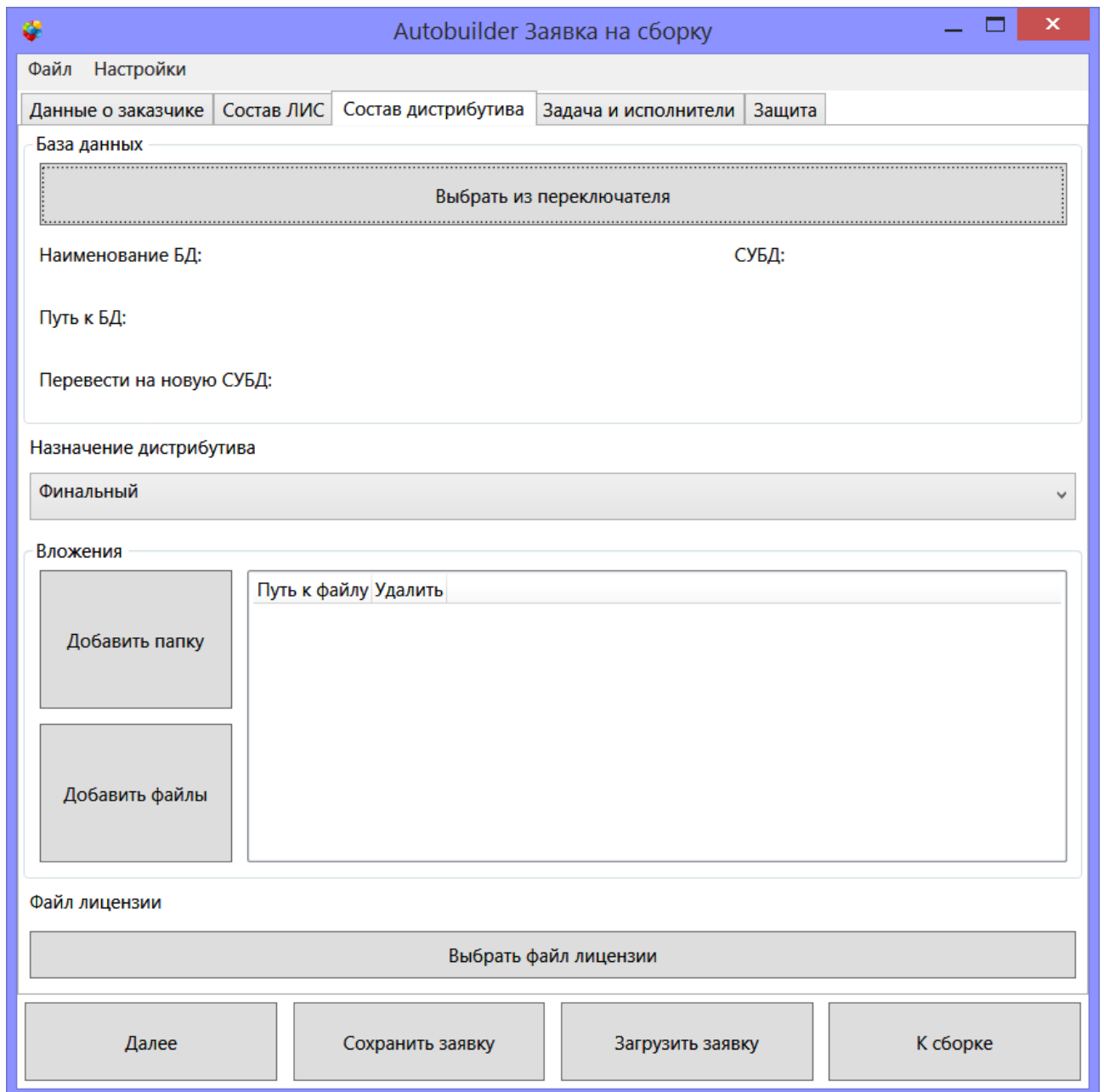


Рисунок 11 – Вкладка «Состав дистрибутива»

Последняя вкладка, относящаяся к документу «заявка на сборку», позволяет задать ответственных исполнителей и сроки выполнения работ, для контроля исполняемости. Она имеет название «Задача и исполнители» и представлена на рисунке 12.

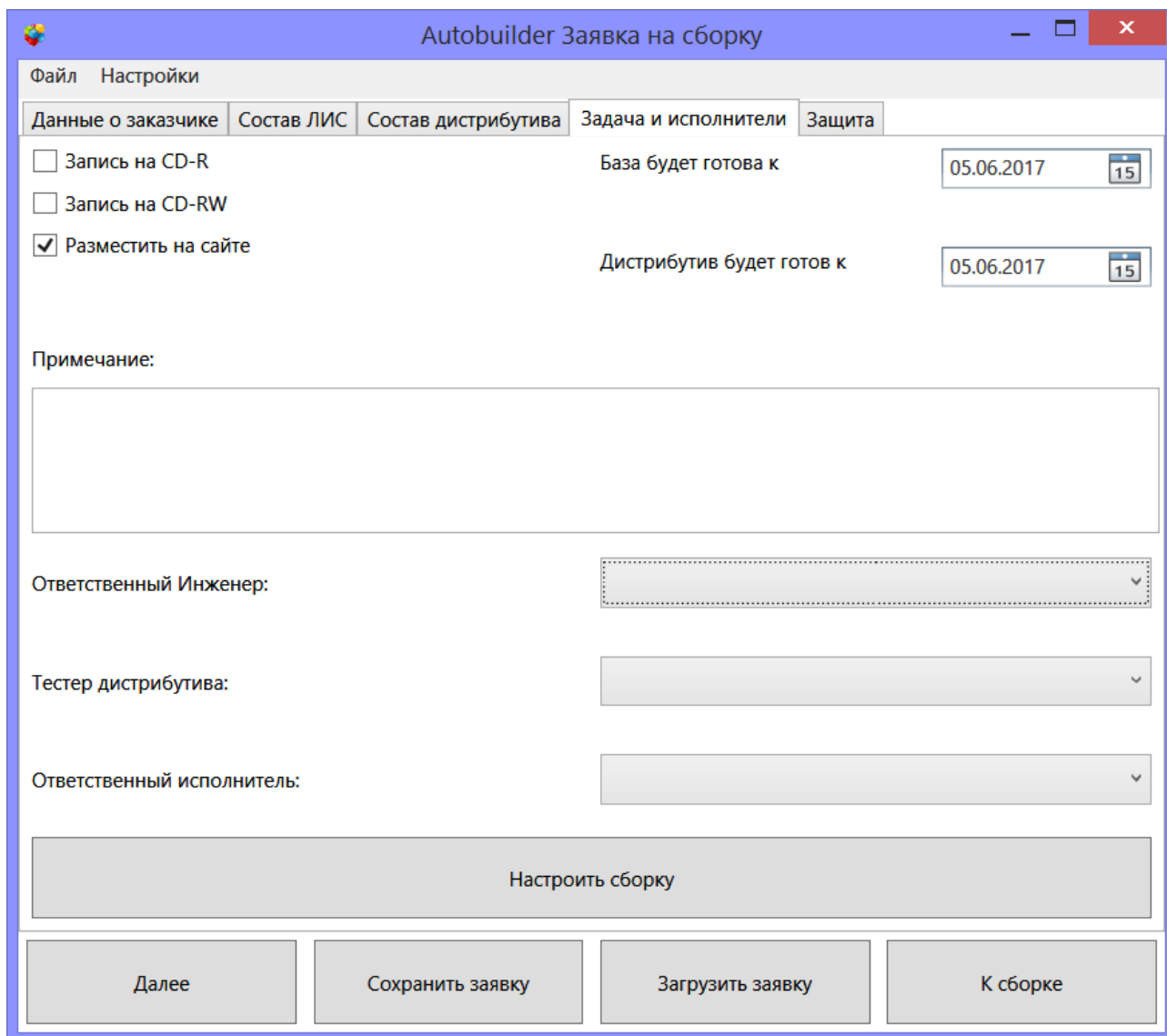


Рисунок 12 – Вкладка «Задача и исполнители»

Редактирование полей заявок реализовано с применением привязок. В WPF привязка (binding) является мощным инструментом программирования. Привязка подразумевает взаимодействие двух объектов: источника и приемника. Объект-приемник создает привязку к определенному свойству объекта-источника. В случае модификации объекта-источника, объект-приемник также будет модифицирован.

Все редактируемые элементы интерфейса связаны со свойствами, описанными в классе Options. Чаще всего эти свойства являются строками, либо булевыми значениями. Свойства привязываются в различных режимах, наиболее используемый в данной работе - двунаправленная привязка. При ее

использовании данные перемещаются в обоих направлениях: от источника к цели и от цели к источнику. Использование режимов привязки заключается в установке свойства Mode объекта Binding. [15]

Для удобно и быстрого заполнения документа были настроены таб-индексы, позволяющие переключаться между полями документа без использования мыши.

При заполнении некоторых полей может использоваться лишь определенный набор значений. Такие поля были реализованы в виде выпадающих списков, что также упрощает ввод данных и не дает возможности произвести опечатку.

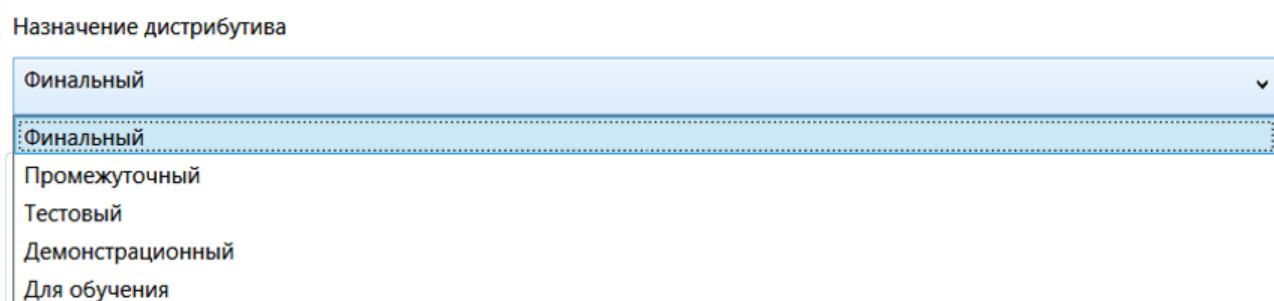


Рисунок 13 – Пример выпадающего списка

В программе была разработана проверка заполнения обязательных для ввода полей перед созданием дистрибутива для предотвращения возникновения исключительных ситуаций. Оставляя обязательные поля незаполненными, пользователь теряет возможность создавать дистрибутивы.

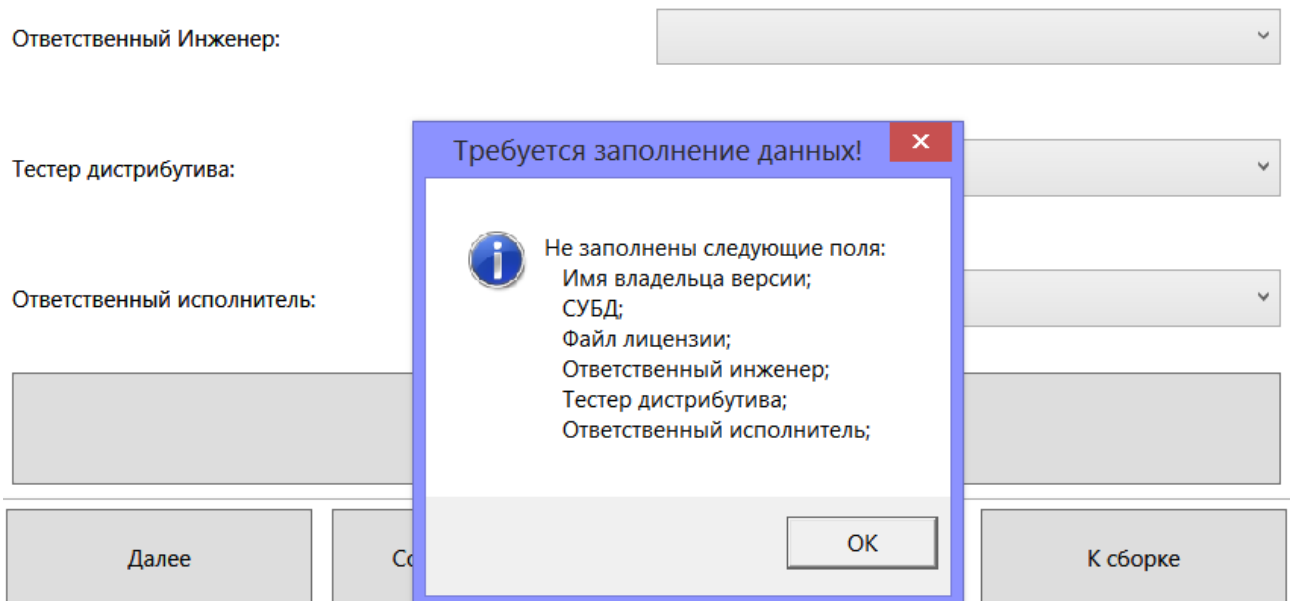


Рисунок 14 – Результат проверки заполнения обязательных полей

Некоторые поля могут потребовать большого объема текста для ввода. Для его удобного просмотра весь интерфейс поддерживает масштабирование. Также многострочные поля ввода поддерживают прокрутку.

Были реализованы некоторые логические проверки, предотвращающие ввод некорректных данных. Например, при сборке демонстрационной версии ЛИС невозможно настроить внедрение защиты от злоумышленников. Демонстрационная версия не требует данной защиты. Пользователь не имеет возможности задавать даты прошедшего времени для полей, отражающих крайние сроки готовности БД и дистрибутива.

При создании новой заявки некоторые поля заполняются значениями по умолчанию. Эти значения были отобраны в ходе анализа нескольких десятков созданных заявок и являются наиболее часто встречаемыми. Их предварительная установка призвана облегчить ввод данных.

Для повышения надежности заявки сохраняются как в папку, выбранную пользователем, так и в предустановленную директорию, расположенную на сетевом диске. Названия сохраненных файлов генерируются автоматически из наименования заказчика и префикса даты, но пользователь может ввести свое название при желании.

Система «запоминает» последние папки, использованные для загрузки и сохранения заявок, а также предлагает их к использованию в следующий раз.

Для прикрепления БД, которая была создана и настроена для заказчика, к дистрибутиву был использован пользовательский элемент управления – «Список подключений». Он был разработан на платформе Windows Forms как модифицированный список ListView. Элемент управления входит в состав утилиты переключателя баз данных, используемых на предприятии для выбора БД, с которой будет работать ЛИС. Элементы этого списка содержат в своих данных путь к БД, который и используется автосборщиком. Данный элемент управления был доработан и приведен на рисунке 15. Его использование обеспечивает целостность и стилистическое единство интерфейса с утилитами, используемыми на предприятии.

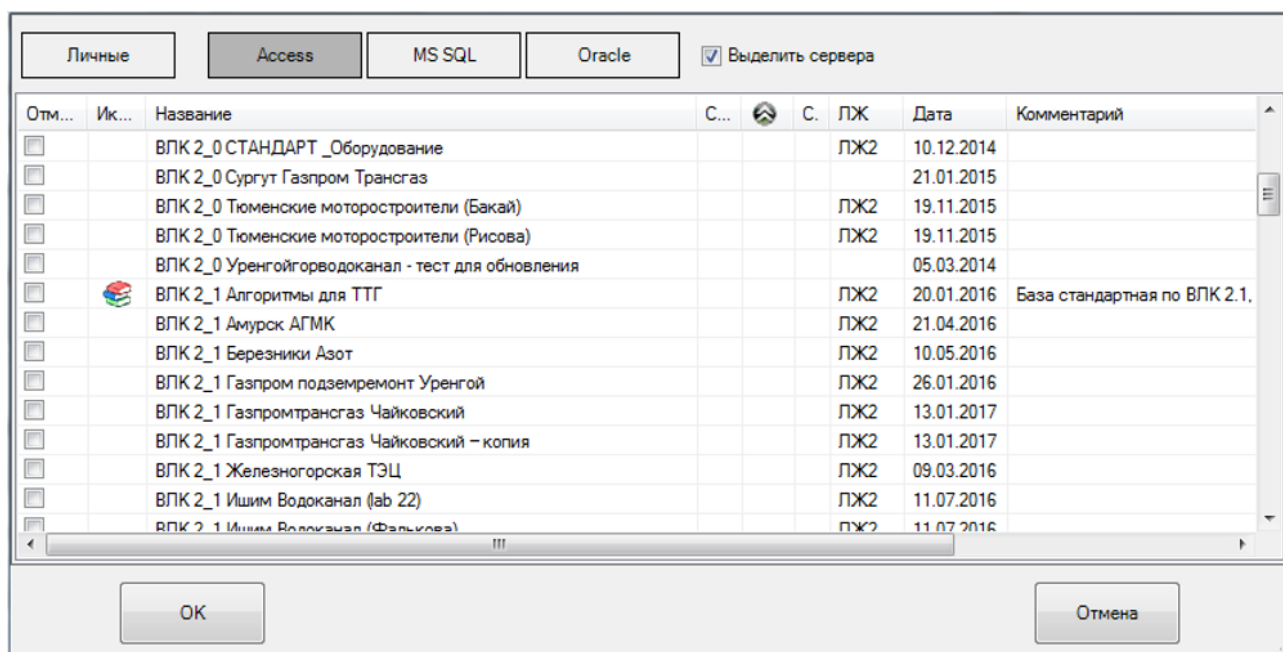


Рисунок 15 – Подключенный элемент управления для выбора БД

Все данные, вводимые пользователем хранятся в классе Options, представленном в приложении Ж. Данный класс агрегирует в себе следующие классы:

- TaskOptions – класс, ответственный за хранение данных о заявке на сборку, он предоставляет методы проверки корректности данных, их сохранения и загрузки;
- DistributiveOptions – класс, ответственный за хранение данных о настройках дистрибутива, например его корневой директории, о возникших ошибках, времени сборки;
- BuildOptions – класс, ответственный за хранение данных о конфигурации сборки проектов, например ее имени и директории размещения результатов сборки;
- UserOptions – класс предназначенный для сохранения пользовательских настроек, например размеров окон. Данный класс выполняет загрузку последней активной заявки при запуске автосборщика.

Данный раздел системы наиболее интенсивно взаимодействует с пользователем, поэтому он будет дорабатываться согласно пожеланиям наиболее часто. Уже на данный момент на предприятии имеется документ, содержащий предложения по улучшению интерфейса. По ним ведутся активные работы.

3.2 Автоматизированное получение исходного кода

Главная задача системы автоматизированной сборки дистрибутивов – программная компоновка файлов, входящих в состав инсталлятора. Иными словами - программное получение библиотек и исполняемых файлов, для формирования дистрибутива. При анализе предметной области были выявлены следующие способы решения задачи:

- Настройка сборки на сервере TFS;
- Использование системы контроля версий для скачивания исходного кода с последующей программной сборкой;

Изначально предполагалось использование первого варианта решения задачи, однако он оказался неподходящим в связи со сложностью собираемой системы.

При выполнении сборки на сервере TFS не нужно сохранять исходные файлы на локальный компьютер. Сборку выполняет агент сервера согласно настроенным шагам сборки, описанных в ее определении. Сборка начинается в момент наступления события, запускающего сборку - триггера.

Было установлено, что настроить определение сборки и запустить ее можно программно. Данная задача оказалась трудоемкой. Несмотря на то, что удалось программно получить артефакты сборки, настроить нужную конфигурацию сборки не удалось. На рисунке 16 представлено программное определение сборки и ее запуск.

```
void TriggerNewBuildDefenition()
{
    string teamProject = "TFS_Testing";
    string buildController = "DevPC - контроллер";

    TfsTeamProjectCollection server = new TfsTeamProjectCollection(new Uri("http://devpc:8080/tfs"));
    server.EnsureAuthenticated();
    IBuildServer buildServer = (IBuildServer)server.GetService(typeof(IBuildServer));

    //настройка определения сборки
    var buildDefinition = buildServer.CreateBuildDefinition(teamProject);
    buildDefinition.Name = "ProgramBuild";
    buildDefinition.Description = "GeneratedFromCode";
    buildDefinition.ContinuousIntegrationType = ContinuousIntegrationType.Individual;
    buildDefinition.Workspace.AddMapping($"{TFS_Testing/simpleProject_for_testing_autoBuilds", "${SourceDir}", WorkspaceMappingType.Map);
    buildDefinition.BuildController = buildServer.GetBuildController(buildController);
    buildDefinition.DefaultDropLocation = @"\\localhost\Users\Public\Drop";

    var defaultTemplate = buildServer.QueryProcessTemplates(teamProject).Where(p => p.TemplateType == ProcessTemplateType.Default).First();
    buildDefinition.Process = defaultTemplate;
    var process = WorkflowHelpers.DeserializeProcessParameters(buildDefinition.ProcessParameters);


    //программно определим конфигурацию
    BuildSettings settings = new BuildSettings();
    settings.ProjectsToBuild = new StringList($"{TFS_Testing/simpleProject_for_testing_autoBuilds/Project1.sln");
    settings.PlatformConfigurations = new PlatformConfigurationList();
    settings.PlatformConfigurations.Add(new PlatformConfiguration("Any CPU", "Debug"));
    process.Add("BuildSettings", settings);
    buildDefinition.ProcessParameters = WorkflowHelpers.SerializeProcessParameters(process);

    buildDefinition.Save();
    // программно триггерим сборку
    buildServer.QueueBuild(buildDefinition);
}
```

Рисунок 16 – Программное определение сборки

Командой QueueBuild() сборка ставится в очередь для исполнения на сервере Team Foundation. Собранные файлы выгружаются в указанную директорию. Результат отображается на веб-портале сервера.

Сборка выполнено

Build сборка_конф_2_20161215.3 

Выполнялось 15 с (DevPC — контроллер), завершено 11 дн назад

[Сводка](#) [Артефакты](#) [Диагностика](#) [Журнал](#)

Сведения о сборке

Определение	сборка_конф_2
Исходная версия	30
Автор запроса	[DefaultCollection]\Project Collection Service Accounts от имени Dev
В очереди	15 декабря 2016 г. 17:13:56
Запущено	15 декабря 2016 г. 17:14:01
Готово	15 декабря 2016 г. 17:14:15

Рисунок 17 – Отображение результата сборки

Для сборки проекта система контроля версий использует либо последнюю версию исходного кода проекта, либо прошлые версии. Однако редактировать исходный код для определения конфигурации и реализации защиты системы простым способом не представляется возможным. Это достижимо лишь при программном сохранении изменений в коде внутри системы контроля версий. Данная процедура осуществима, но не оправдывает затраты в связи с высокой трудоемкостью.

Для решения данных проблем было решено выгружать программно исходный код на локальный компьютер для последующего редактирования. Получить исходный код из системы контроля версий, реализованной на TFS можно несколькими способами:

- Использование TFS REST API для получения исходного кода;
- Использование библиотеки Microsoft.TeamFoundation для получения исходного кода.

Использование TFS REST API наиболее эффективный вариант для веб-приложений, однако разрабатываемая система таковым не является. В связи с этим была изучена и подключена библиотека Microsoft.TeamFoundation, позволяющая скачивать проекты. Используемая в приложении функция для

выгрузки исходного кода выбранного проекта в заданную директорию отражена на рисунке 18.

```
public static void MapProject(string projectServerPath, string localPath)
{
    Uri serverUri = Options.DefaultMapperOptions.TfsServerUri;

    TfsTeamProjectCollection tpc = new TfsTeamProjectCollection(serverUri);
    VersionControlServer versionControl = (VersionControlServer)tpc.GetService(typeof(VersionControlServer));

    Deleteworkspace();
    var workspace = versionControl.CreateWorkspace("NewProgramChemicMapping", versionControl.AuthorizedUser);
    // Свяжем с локальной директорией, затем получаем файлы из репозитория
    workspace.Map(projectServerPath, localPath);
    workspace.Get();
    workspace.Delete();
}
```

Рисунок 18 – Функция выгрузки исходного кода проекта

Данная функция вызывается первой в процессе работы автосборщика как с проектом самой лабораторной системы, так и с выбранными дополнительными модулями. За каждый модуль и за саму ЛИС отвечают отдельные классы, каждый из которых реализует интерфейс с функцией Run, выполнение которой начинается с функции выгрузки исходного кода. Она реализована в классе Mapper. После скачивания исходного кода, он, при необходимости, модифицируется. Следующим этапом происходит сборка проекта классом Builder, и в заключении собранные файлы копируются в состав дистрибутива методом класса InstallerFiller. Классы, ответственные за выполнение данного процесса отображены в приложении 3. Пример исходного кода, демонстрирующий протекание процесса формирования дистрибутива отражен на рисунке 19.

```
// Здесь устанавливается порядок выполнения процесса сборки дистрибутива
private static void SetSteps()
{
    // Подготовка к созданию дистрибутива - создание директорий, заполнение
    StepManager.Add(new InstallerDefaultsSetter());

    // Сборки проектов
    StepManager.Add(new Chemic());
    StepManager.Add(new UserRights());
    StepManager.Add(new SML());
    StepManager.Add(new JRP());
    StepManager.Add(new ControlBlock());
    StepManager.Add(new DiaryChemicT());

    // Модули в разработке
    //StepManager.Add(new ChemicBal());
    //StepManager.Add(new ChemicBel());

    // Сборка дистрибутива
    StepManager.Add(new InstallerCreator());
}

public void Run(Options options)
{
    Map();
    ChangeLisConfig(options);
    ChangeLisParacodes(options);
    Build();
    FillInstallerWithArtifacts();
    RenameChemic15ExeAndPdb();
}
```

Рисунок 19 – Декомпозиция процесса формирования дистрибутива

3.3 Программная сборка решений

Программная сборка решений наступает после того как исходные файлы решений были скачаны, и, при необходимости, модифицированы. За данный процесс отвечает класс Builder. При выполнении сборки происходит логирование процесса классом Logger, что отражено на диаграмме классов в приложении 3.

При реализации программной сборки были проанализированы несколько альтернативных подходов:

- Настройка сборки на сервере TFS;
- Сборка с использованием библиотеки MS Build

Невозможность использовать сборку на сервере TFS была обоснована в пункте 3.2, поэтому для программных сборок была использована библиотека MS Build. Она является стандартным средством сборки решений от корпорации Microsoft и предоставляет богатые возможности конфигурированию сборки, при этом обеспечивая хорошую производительность. На рисунке 20 представлена реализованная функция по программной сборке решений.

```

// сборка решения по .sln файлу
public static void BuildSolution(FileInfo solutionToBuild, Options options)
{
    options.distributiveOptions.CurrentProcess = $"Сборка решения {solutionToBuild.Name}";
    List<ILogger> loggers = CreateLoggers();
    ProjectCollection projectCollection = new ProjectCollection();

    Dictionary<string, string> GlobalProperty = new Dictionary<string, string>();
    GlobalProperty.Add("Configuration", "Release");

    BuildParameters parameters = new BuildParameters(projectCollection);
    parameters.Loggers = loggers;

    BuildRequestData BuildRequest = new BuildRequestData(solutionToBuild.FullName,
        GlobalProperty, "14.0", new string[] { "Build" }, null);
    BuildResult buildResult = BuildManager.DefaultBuildManager.Build(parameters, BuildRequest);

    if (buildResult.OverallResult == BuildResultCode.Success)
        Debug.WriteLine($"{solutionToBuild.Name} build succeeded");
    else
        Debug.WriteLine($"{solutionToBuild.Name} build failed");
}

```

Рисунок 20 – Функция программной сборки решений

При использовании библиотеки MS Build задействуется процесс msbuild.exe. Таким образом самым простым методом программной сборки решений является программный запуск процесса msbuild.exe, с указанием пути к решению в качестве параметра. Метод не используется в данной работе по той причине, что указание нескольких дополнительных параметров сборки в нем является крайне неудобным.

3.4 Прикрепление дополнительных файлов

Перед тем как файлы, собранные из выгруженного исходного кода, попадут в директорию с дистрибутивом, данная директория должна быть создана. Более того, к тому моменту она уже будет наполнена начальными данными. Среди этих данных содержатся утилиты и библиотеки, не требующие обновлений. За создание этих директорий и прикрепление начальных файлов ответственен класс `InstallerDefaultSetter`. Он, как и остальные классы, выполняющие процесс сборки, реализует интерфейс `IExecute`.

После создания начальных условий для сборки дистрибутива классом `InstallerDefaultSetter` выполняется сборка ЛИС и ее модулей. Прикрепление

файлов БД, лицензии и вложений в заключении сборки выполняет класс InstallerCreator. Данные классы представлены на UML диаграмме в приложении И

Необходимость дополнять дистрибутивы файлами-вложениями возникает достаточно часто. Это могут быть текстовые файлы с инструкциями, документация по эксплуатации, SQL-скрипты для модификации БД заказчика, графические материалы.

Задача по автоматическому внедрению их в дистрибутив не решена в связи с тем, что данные файлы имеют разное применение. Полезная работа автосборщика в данном случае состоит в том, чтобы скопировать все прикрепленные вложения в директорию на сетевом диске, где и создаются дистрибутивы. Собранные дистрибутивы являются целостными и хранятся централизованно.

При добавлении вложений можно указывать как файлы, так и целые директории. Для копирования содержимого директорий был реализован рекурсивный метод, позволяющий учитывать вложенные директории.

Вложения можно откреплять, удаляя их из списка. При добавлении вложений открываются стандартные диалоги выбора файла, либо папки. Интерфейс списка вложений представлен на рисунке 21.

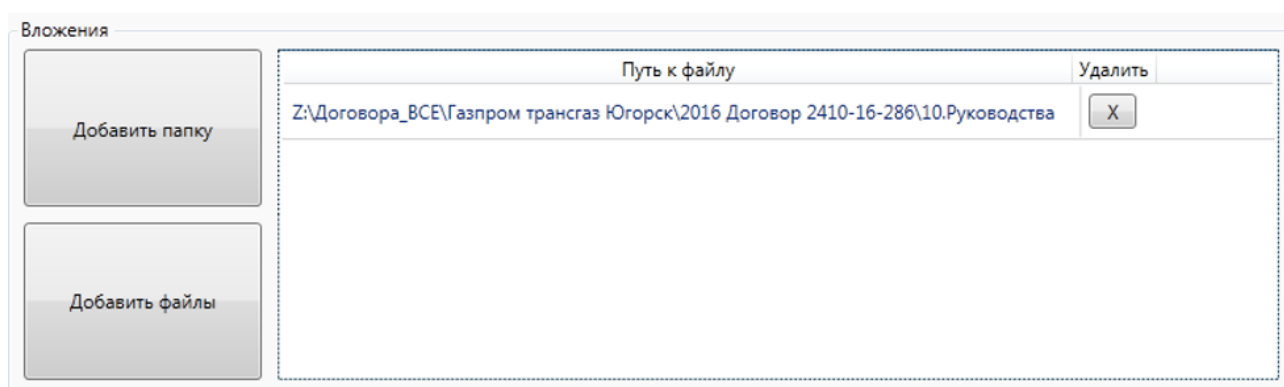


Рисунок 21 – Список вложений

3.5 Архивация

Дистрибутив ЛИС запаковывается в защищенный RAR архив перед отправкой заказчику. Перед этим CAB архив вкладываются файлы, которые должны быть развернуты при установке системы. Преимущество формата CAB состоит в хорошей степени сжатия. RAR архив предоставляет возможность защитить паролем не только вложенные файлы, но и сам архив, заблокировав просмотр внутренней структуры. В дополнение RAR архиватор является наиболее распространенным. Таким образом при использовании данных форматов достигается минимизация размера дистрибутива и его защита.

Программное выполнение CAB архивирования является простой задачей. Для ее выполнения был произведен поиск библиотек, обладающих данным функционалом. Среди найденных была выбрана библиотека Microsoft.Deployment.Compression, так как она является стандартным инструментом сжатия, поставляемым Microsoft. Она позволяет задавать файлы для сжатия, название архива, управлять степенью сжатия, отслеживать прогресс архивации, что является достаточным функционалом.

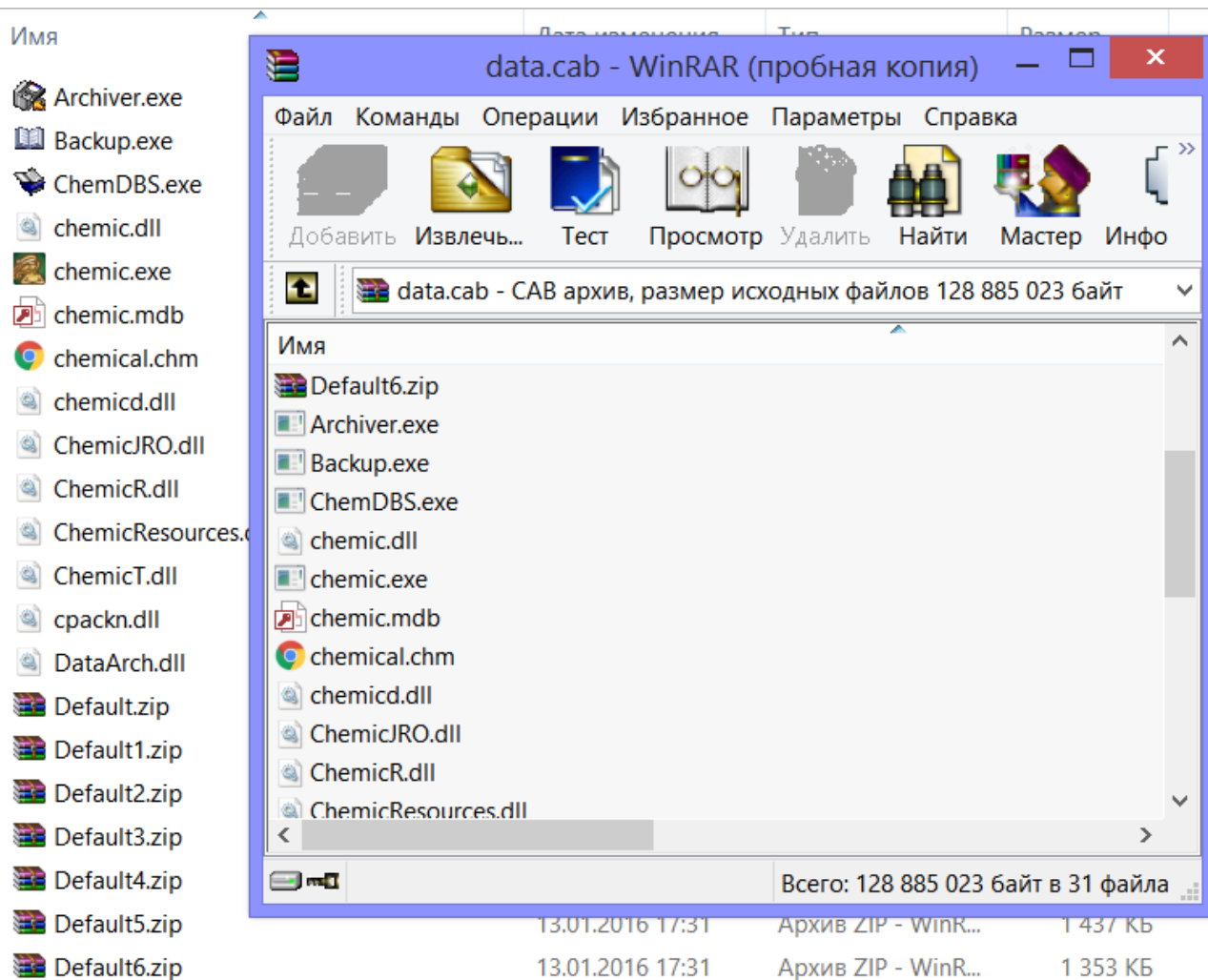


Рисунок 22 - Созданный CAB архив

```
public static void MakeArchive(string resultArchivePath, DirectoryInfo directoryToPack)
{
    CabInfo archive = new CabInfo(resultArchivePath);
    archive.Pack(directoryToPack.FullName, true,
        Microsoft.Deployment.Compression.CompressionLevel.Max, null);
}
```

Рисунок 23 - Функция программной CAB архивации

Для программного создания защищенного архива рассматривался также формат ZIP. Были изучены и проанализированы библиотеки DotNetZip и SharpZipLib. Данные библиотеки имели интуитивно понятный синтаксис и хорошую документацию. К сожалению данные библиотеки некорректно работают с кодировками и не поддерживают русскоязычный текст. Более того защищенные архивы, созданные с применением данных библиотек открыты для

просмотра. Файлы защищены от распаковки, но их названия и структуру архива можно просматривать.

Данные недостатки библиотек привели к поиску более достойных альтернатив. В данной работе программная архивация осуществляется с помощью обращения к консольной версии архиватора RAR. Недостатки данного метода в том, что необходимо иметь консольную версию архиватора по заданному пути, а также требуется изучить команды для обращения к ней. При ее удалении, либо перемещении данный метод окажется неработоспособен. Однако данная версия предоставляет полную защиту архива и корректно работает с кодировками. На рисунке 24 представлено обращение к консольной версии архиватора RAR. Далее на рисунке 25 представлен результат программного архивирования.

```
var startInfo = new ProcessStartInfo("rar.exe")
{
    WorkingDirectory = "\"" + WorkingDirectories.installerRoot + "\"",
    Arguments = command,
    UseShellExecute = true,
    CreateNoWindow = true,
    WindowStyle = ProcessWindowStyle.Hidden,
};
```

Рисунок 24 – Обращение к консольной версии архиватора RAR

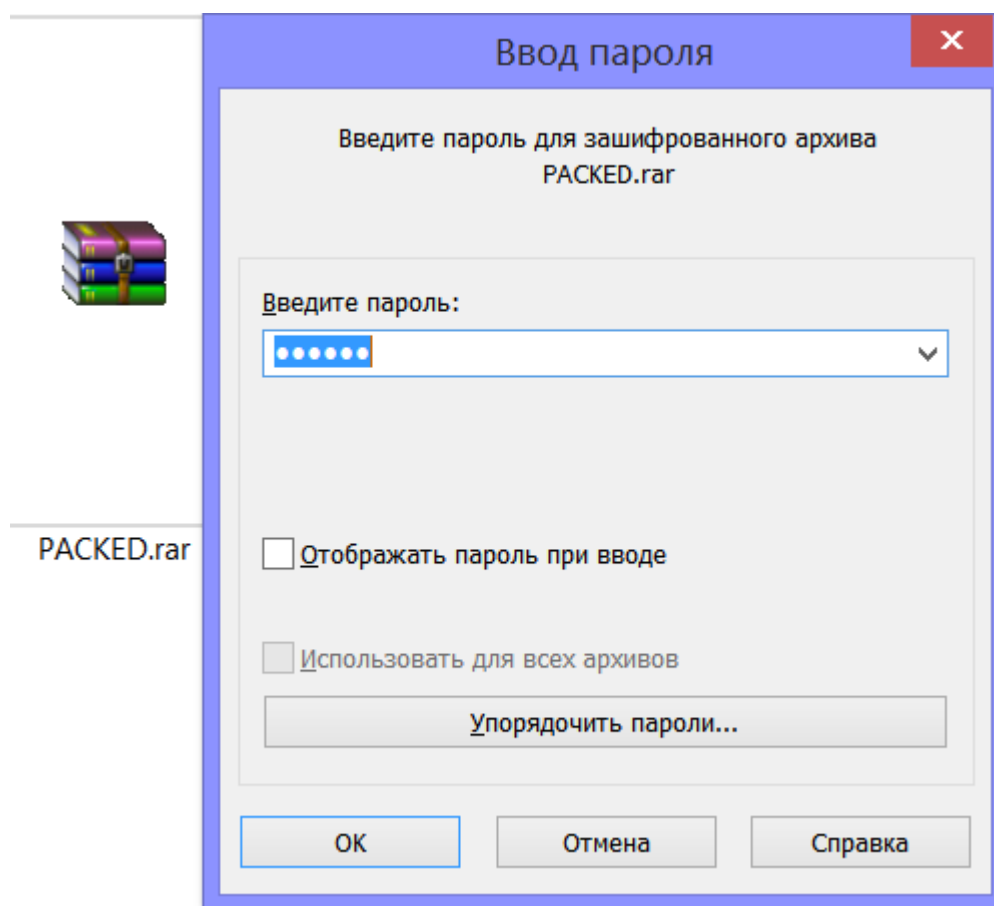


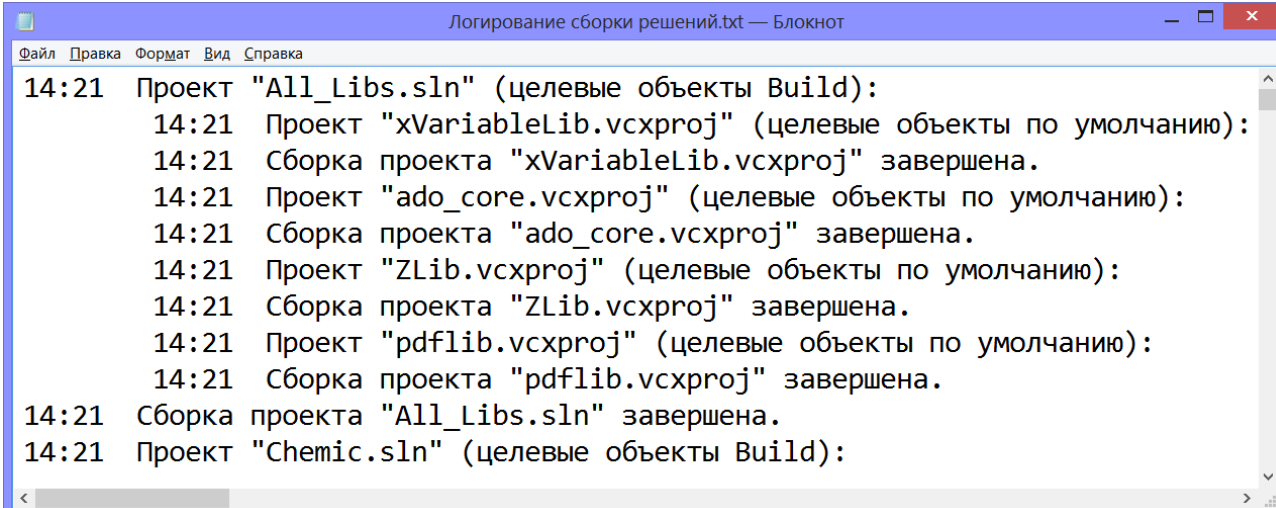
Рисунок 25 – Созданный защищенный архив

СAB и RAR архивация – последние этапы сборки дистрибутива. Они выполняются классом Compressor, который используется классом InstallerCreator, завершающим установку дополнением дистрибутива файлом БД, текстом лицензии и вложениями и соответственно архивацией. Данные классы отражены на диаграмме в приложении И.

3.6 Логирование

Возникновение исключительных ситуаций при сборке проектов библиотекой MS Build не приводит ни к остановке работы программы, ни к уведомлению пользователя о внутренних ошибках. Для проверки результата сборки используется экземпляр класса BuildResult. Навигация по нему в режиме отладки не является удобной. Поэтому был реализован класс Logger для логирования результатов сборки.

Лог ошибок сборки являются задокументированным свидетельством неисправности собираемого проекта, что позволяет своевременно выявить проблему и приступить к ее решению. Фрагмент лога процесса сборки представлен на рисунке 26.



```
Логирование сборки решений.txt — Блокнот
Файл  Правка  Формат  Вид  Справка
14:21 Проект "All_Libs.sln" (целевые объекты Build):
      14:21 Проект "xVariableLib.vcxproj" (целевые объекты по умолчанию):
      14:21 Сборка проекта "xVariableLib.vcxproj" завершена.
      14:21 Проект "ado_core.vcxproj" (целевые объекты по умолчанию):
      14:21 Сборка проекта "ado_core.vcxproj" завершена.
      14:21 Проект "ZLib.vcxproj" (целевые объекты по умолчанию):
      14:21 Сборка проекта "ZLib.vcxproj" завершена.
      14:21 Проект "pdflib.vcxproj" (целевые объекты по умолчанию):
      14:21 Сборка проекта "pdflib.vcxproj" завершена.
14:21 Сборка проекта "All_Libs.sln" завершена.
14:21 Проект "Chemic.sln" (целевые объекты Build):
```

Рисунок 26 - Фрагмент лога сборки проектов

Логируются не только ошибки сборки проектов, но и ошибки, возникающие при формировании дистрибутива. Большинство проводимых операций автосборщика имеют проверки на наиболее вероятные ошибки. Например, при работе в файловой системе Windows, разработанной системой проверяется наличие запрашиваемых файлов и директорий, наличие доступа на операции, проводимые с файлами.

Также логируются ошибки, возникающие при программном скачивании исходного кода, архивировании. После ввода логирования, процесс отладки приложения был упрощен. При наличии обнаруженных ошибок, они выводятся предупреждающим сообщением по окончании сборки дистрибутива.

4 ДЕМОНСТРАЦИЯ РАБОТЫ ПРИЛОЖЕНИЯ

4.1 Обработка «заявок на сборку»

Для демонстрации способности системы обрабатывать заявки на сборку, в ней была отражена одна из использованных заявок. Изменениям подверглось

лишь то, что была выбрана самая полная конфигурация ЛИС и отмечены все дополнительные модули. Данная заявка на сборку размещена в приложении Е. Все данные документа были успешно перенесены в систему. Далее приведены вкладки интерфейса системы автоматизированной сборки дистрибутивов, отражающие данные бумажного документа.

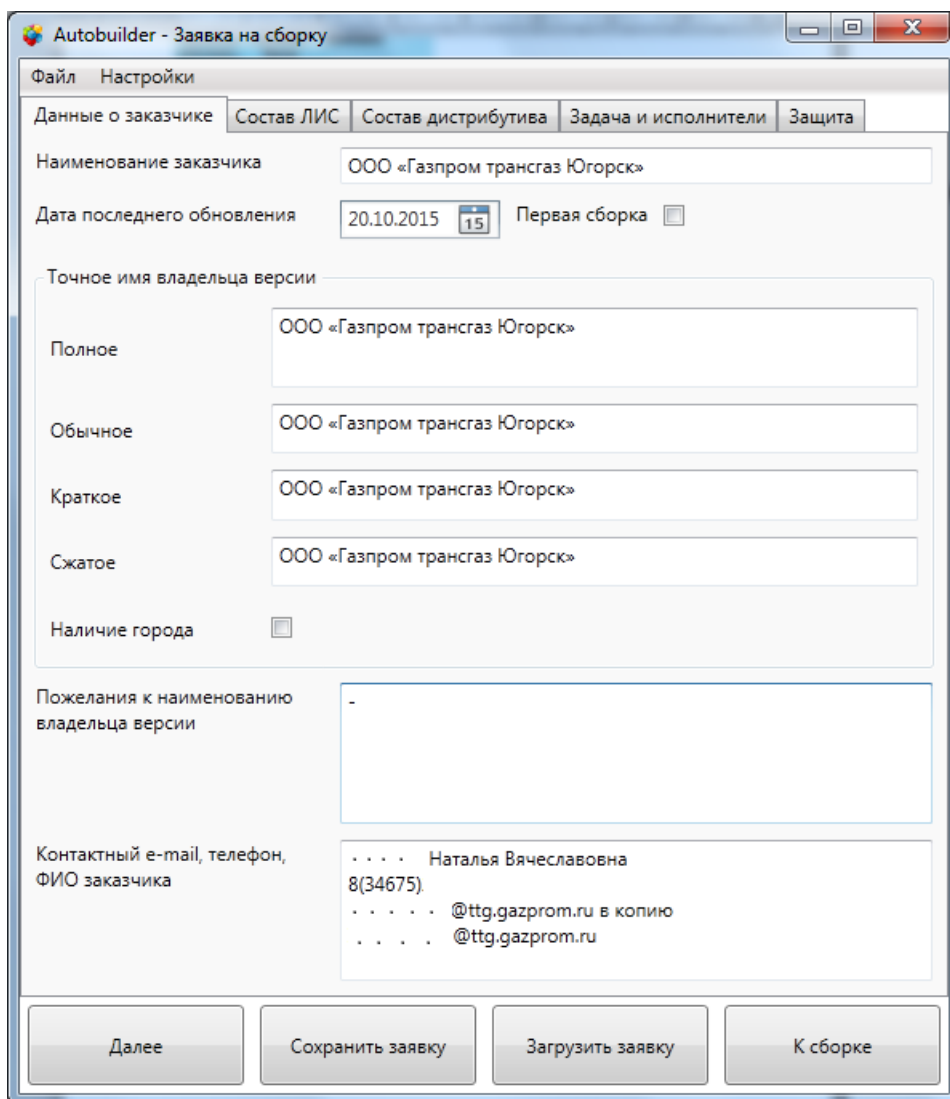


Рисунок 27 – Заполнение вкладки «Данные о заказчике»

После заполнения всех вкладок по данной заявке будет собран тестовый дистрибутив. Для того, чтобы проверить сборку модулей и конфигураций ЛИС на вкладке «Состав ЛИС» были отмечены все дополнительные модули и выбрана самая полная конфигурация.

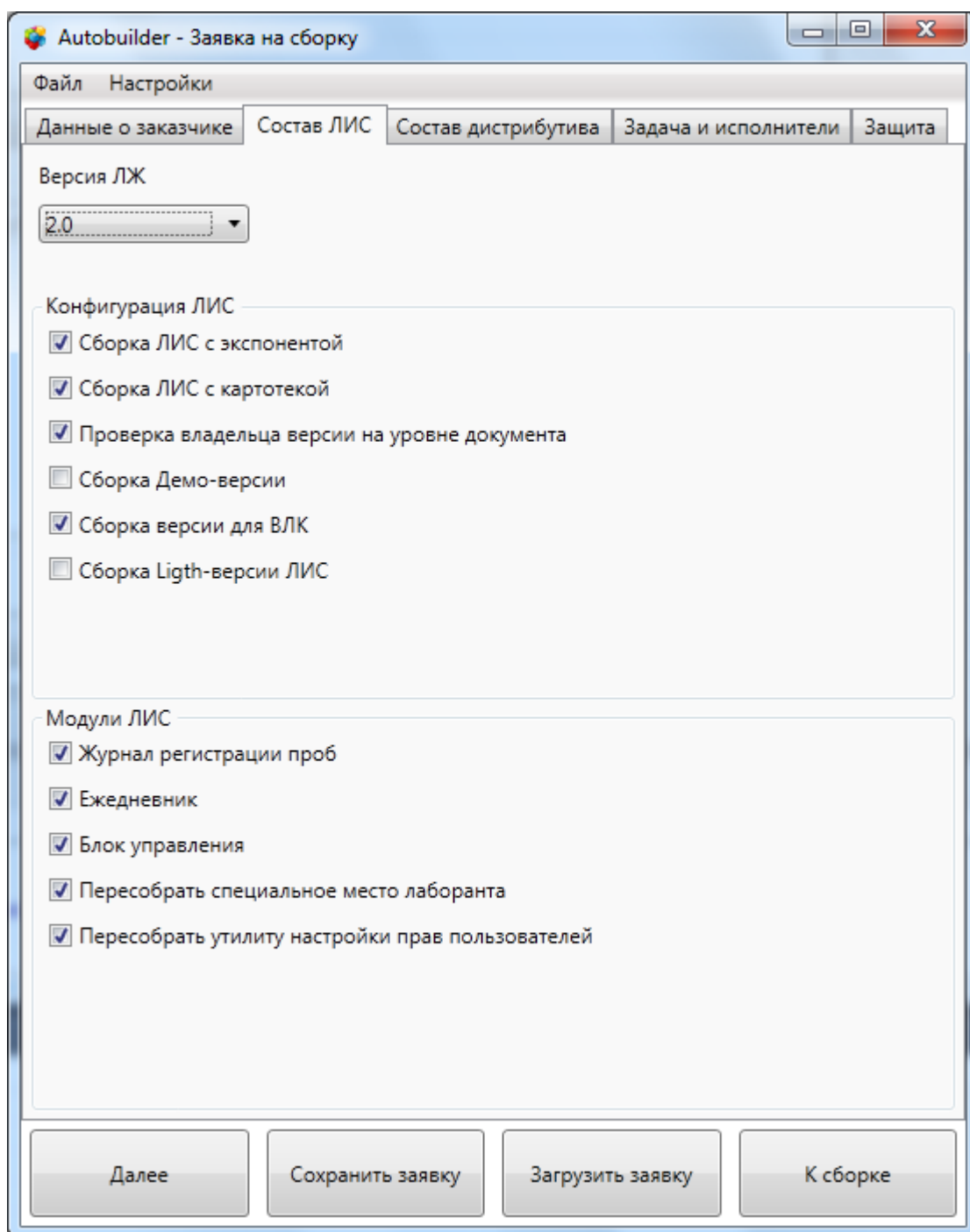


Рисунок 28 – Выбор наиболее полной конфигурации ЛИС

В заполнение следующих вкладок не вносилось поправок. Интерфейс системы отвечал основным требованиям и позволял внести данные корректно. Возможность заполнять поля с использованием выпадающих списков и без участия компьютерной мыши ускорила работу. В составе дистрибутива были указаны все необходимые файлы, были внесены данные о задаче и ее исполнителях, что отражено на рисунках 29, 30.

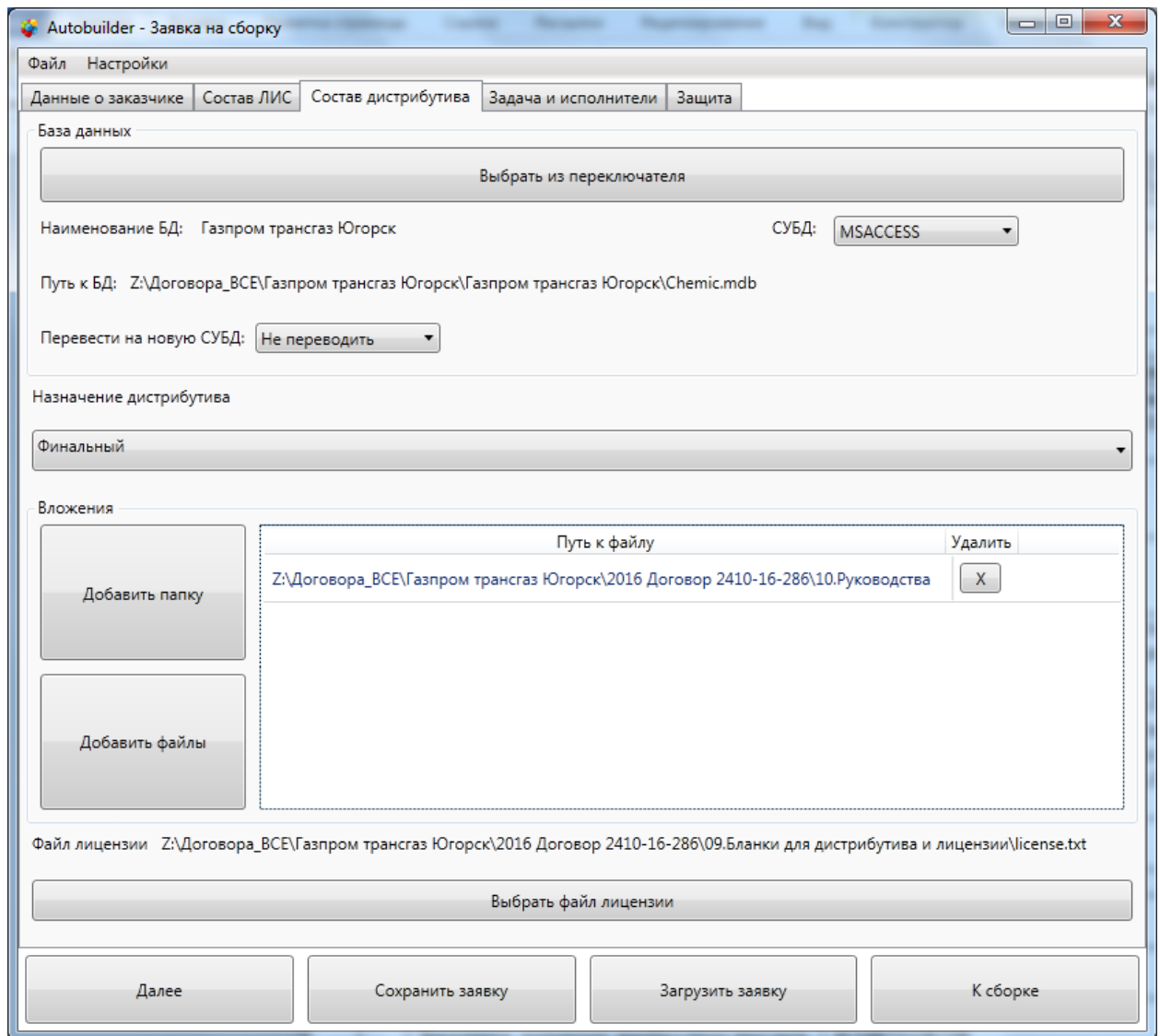


Рисунок 29 – Заполнение вкладки «Состав дистрибутива»

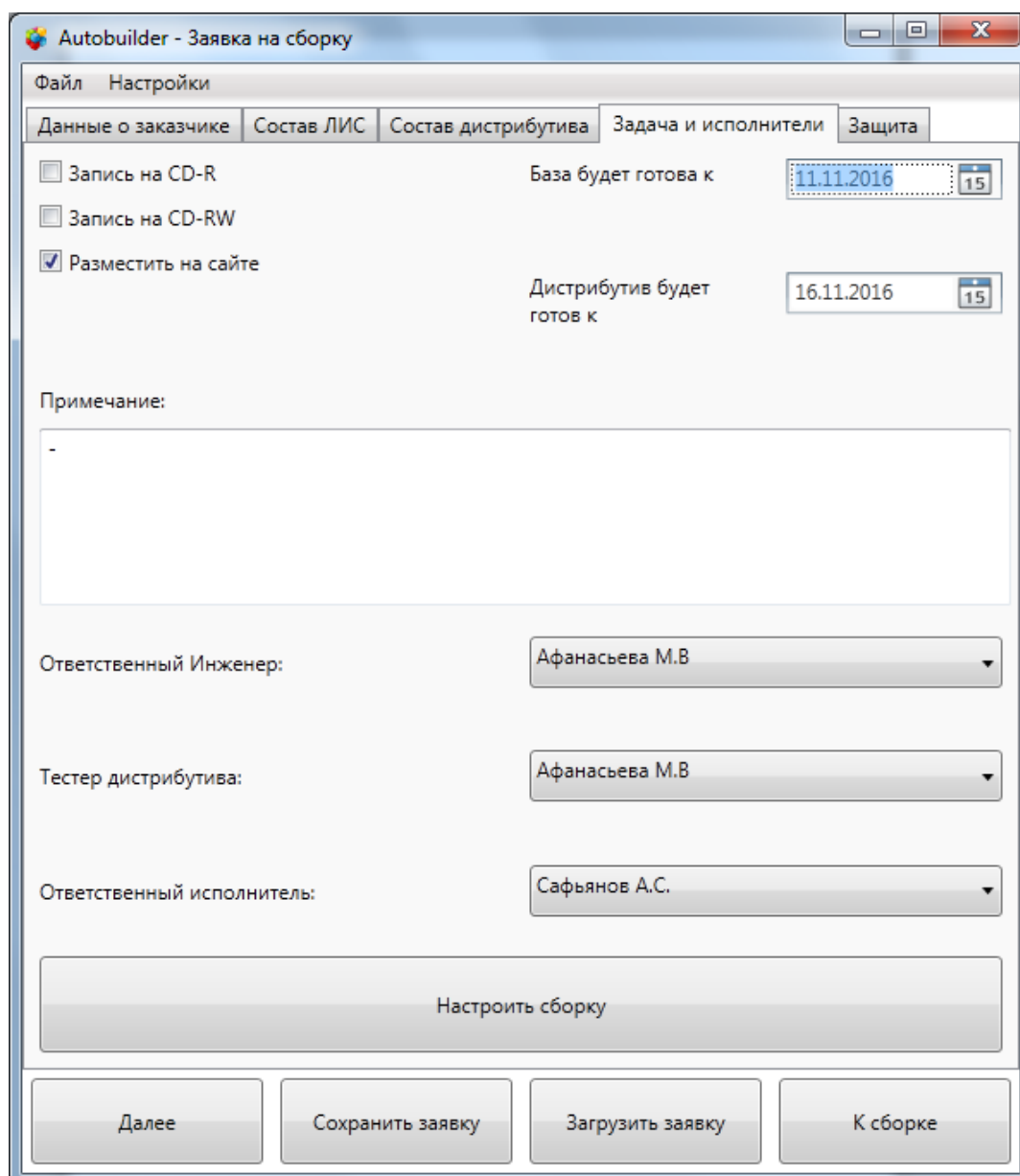


Рисунок 30 – Заполнение вкладки «Задача и исполнители»

4.2 Получение дистрибутивов по «заявкам на сборку»

Для проверки возможности системы автоматизировано создавать дистрибутивы, по заполненной в пункте 4.1 заявке на сборку был собран инсталлятор ЛИС.

Процесс сборки отражается в окне сборки дистрибутива и выполняется в отдельном потоке. В связи с этим его можно прервать по желанию. Интерфейс окна сборки дистрибутива представлен на рисунке 31.

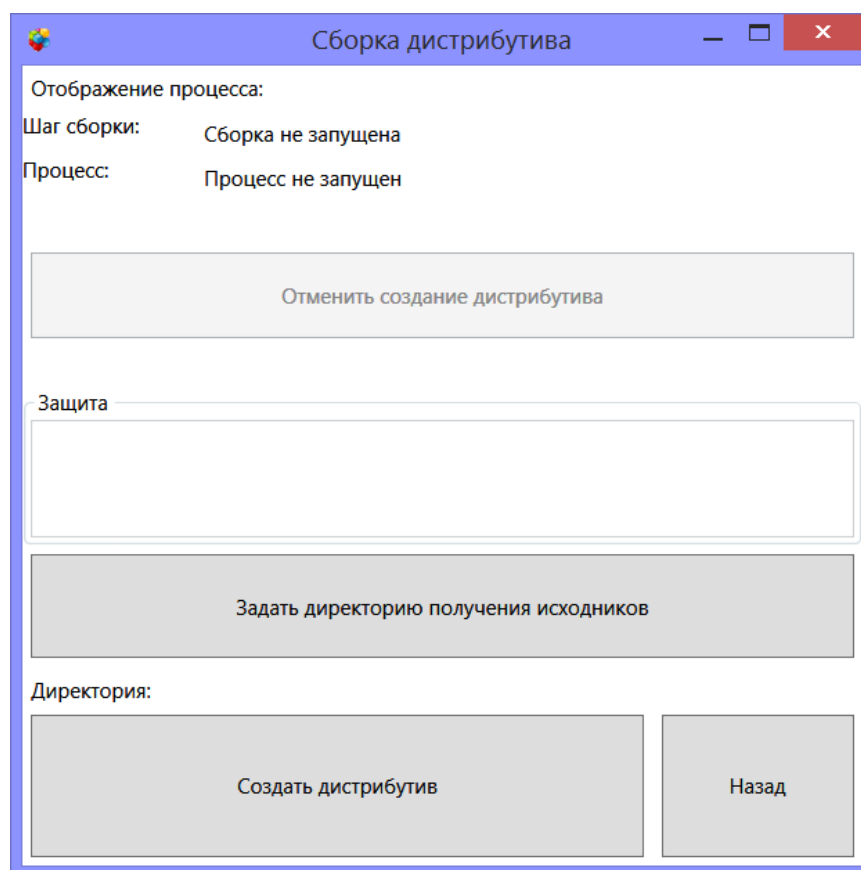


Рисунок 31 - Окно сборки дистрибутива

Результатом сборки по заявке стали требуемые директории и файлы. Выгрузка исходного кода проектов была произведена удачно, сборка и перенос библиотек и исполняемых файлов прошел без ошибок. Были корректно перенесены вложения, сформированы CAB и RAR архивы. Результат работы автосборщика продемонстрирован на рисунках 32, 33.

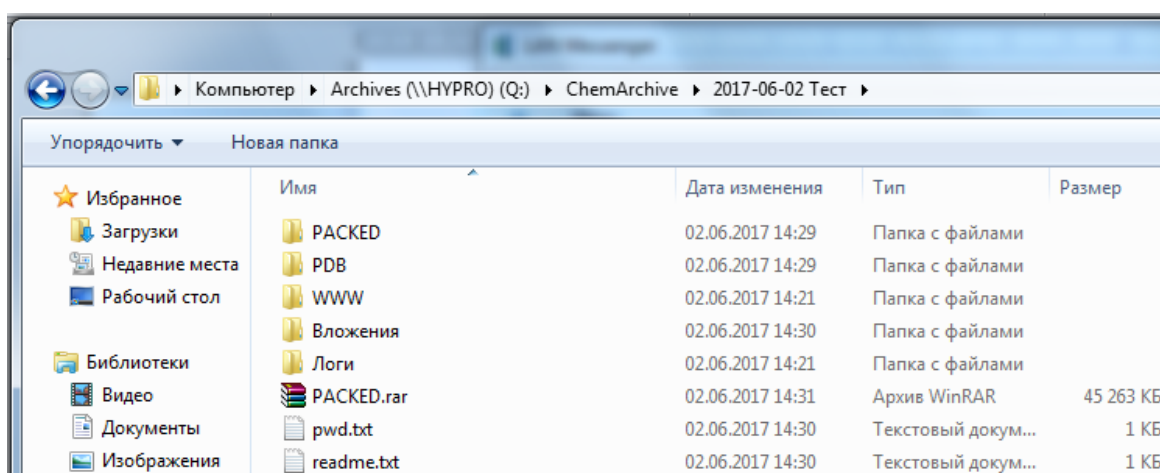


Рисунок 32 – Результат работы автосборщика

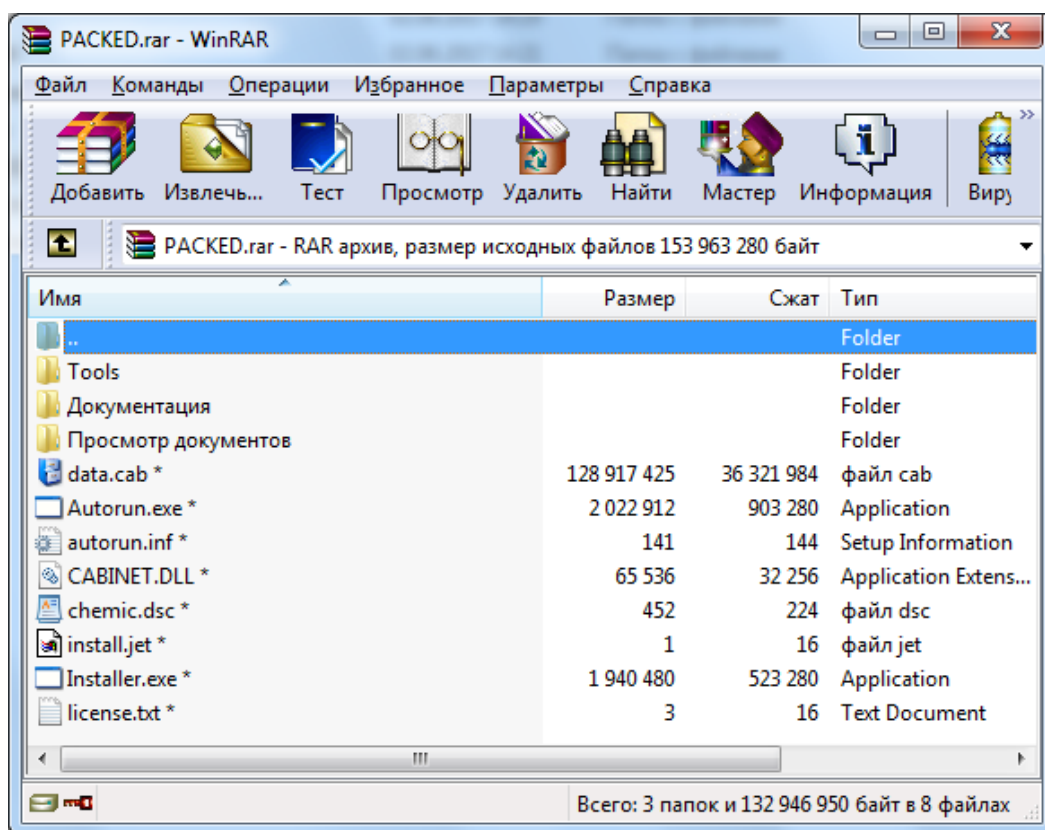


Рисунок 33 – Результат работы автосборщика

При запуске исполняемого файла инсталлятора Autorun.exe начинается процесс установки ЛИС. При этом открывается меню, представленное на рисунке 34.



Рисунок 34 – Начало установки

Далее пользователю предлагается ознакомиться с лицензионным соглашением, выбрать пути для установки ЛИС и развертывания БД. После этого необходимо выбрать компоненты для установки. Окно выбора конфигурации представлено на рисунке 35.

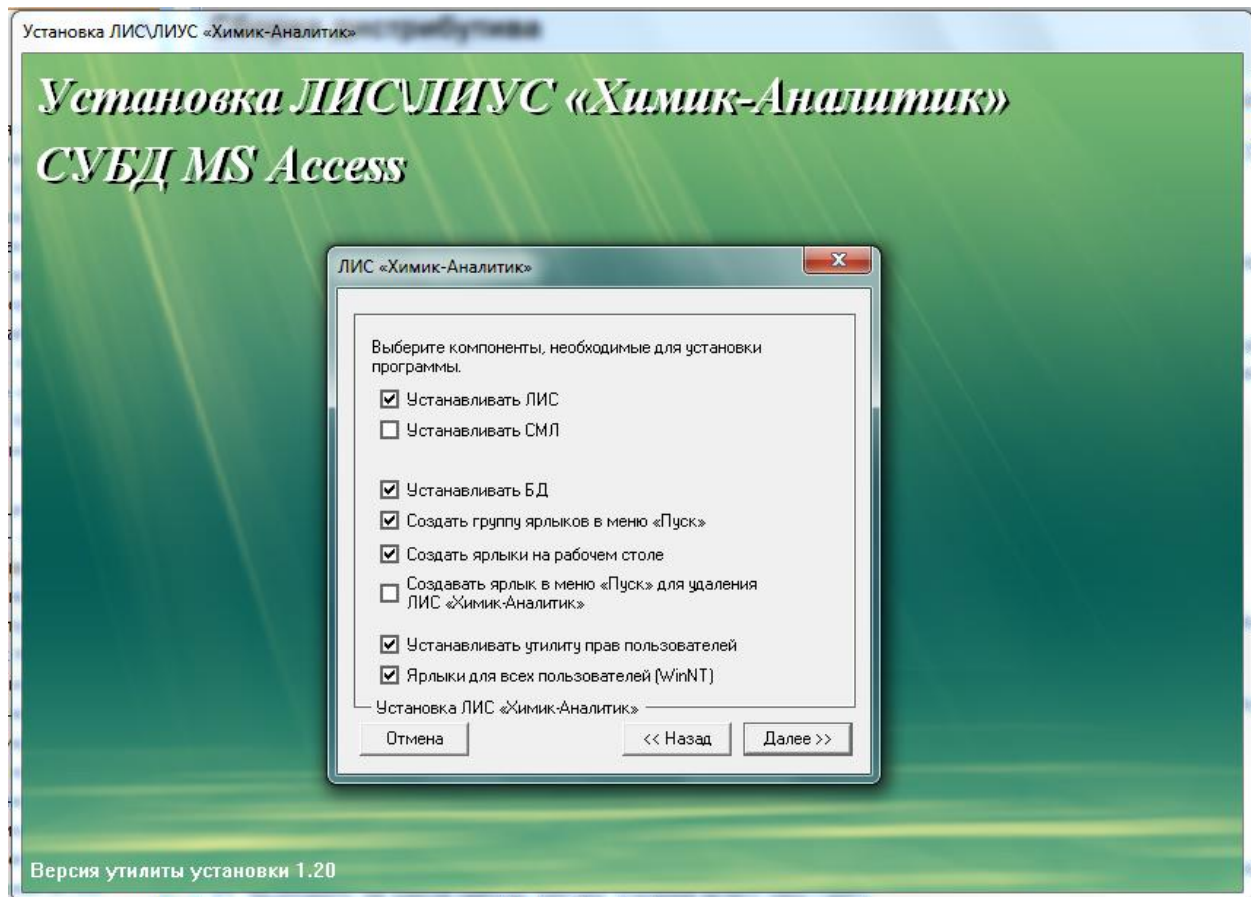


Рисунок 35 – Окно выбора конфигурации

Установка ЛИС занимает непродолжительное время. После установки были модифицированы ветки реестра на рабочей станции. На рисунке 36 представлены записи в реестре в ветке текущего пользователя. Также требуемые записи были созданы в ветке HKEY_LOCAL_MACHINE.

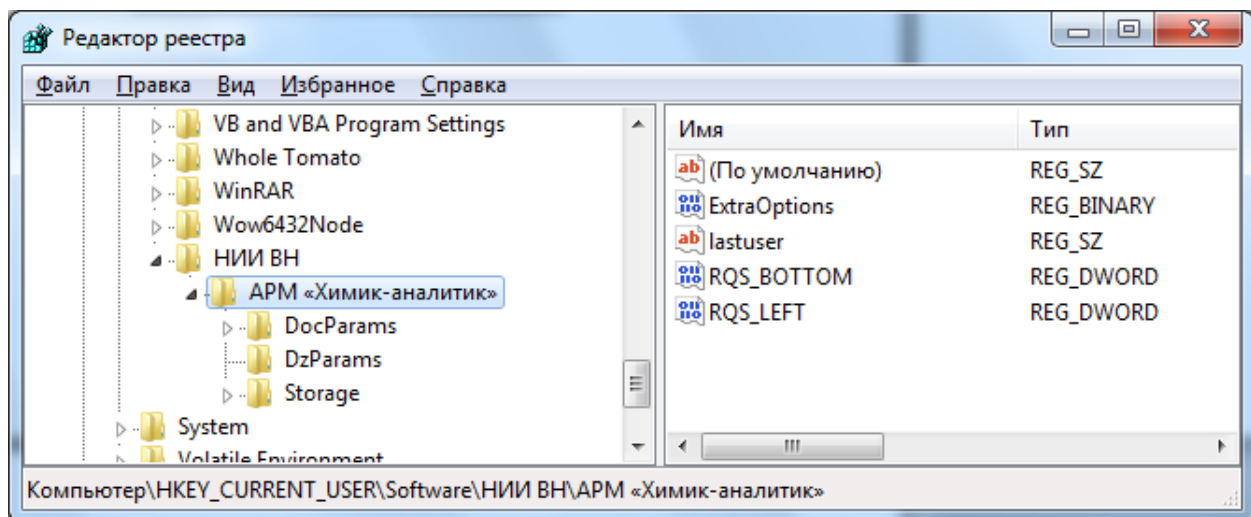


Рисунок 36 – Созданные записи в реестре

На рабочем столе были созданы ярлыки для запуска ЛИС и утилиты «Специальное место лаборанта». В меню «Пуск» были созданы папки, содержащие все необходимые ярлыки для быстрого доступа к системе и ее компонентам

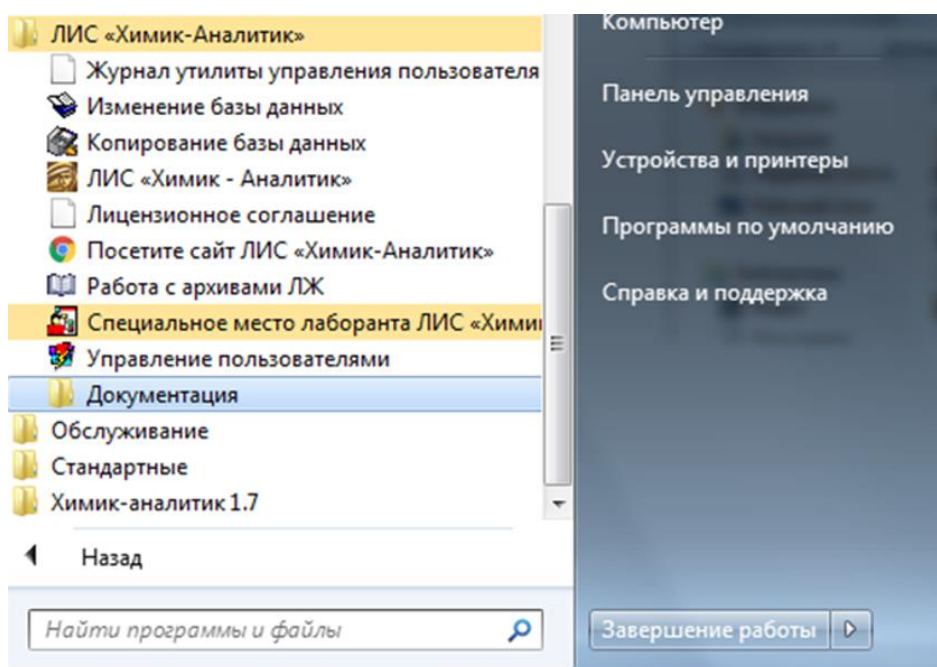


Рисунок 37 – Созданные папки в меню «Пуск»

Развернутая система была протестирована на соответствие заявленной конфигурации и наличие указанных модулей. Проверка была пройдена, ошибок не обнаружено.

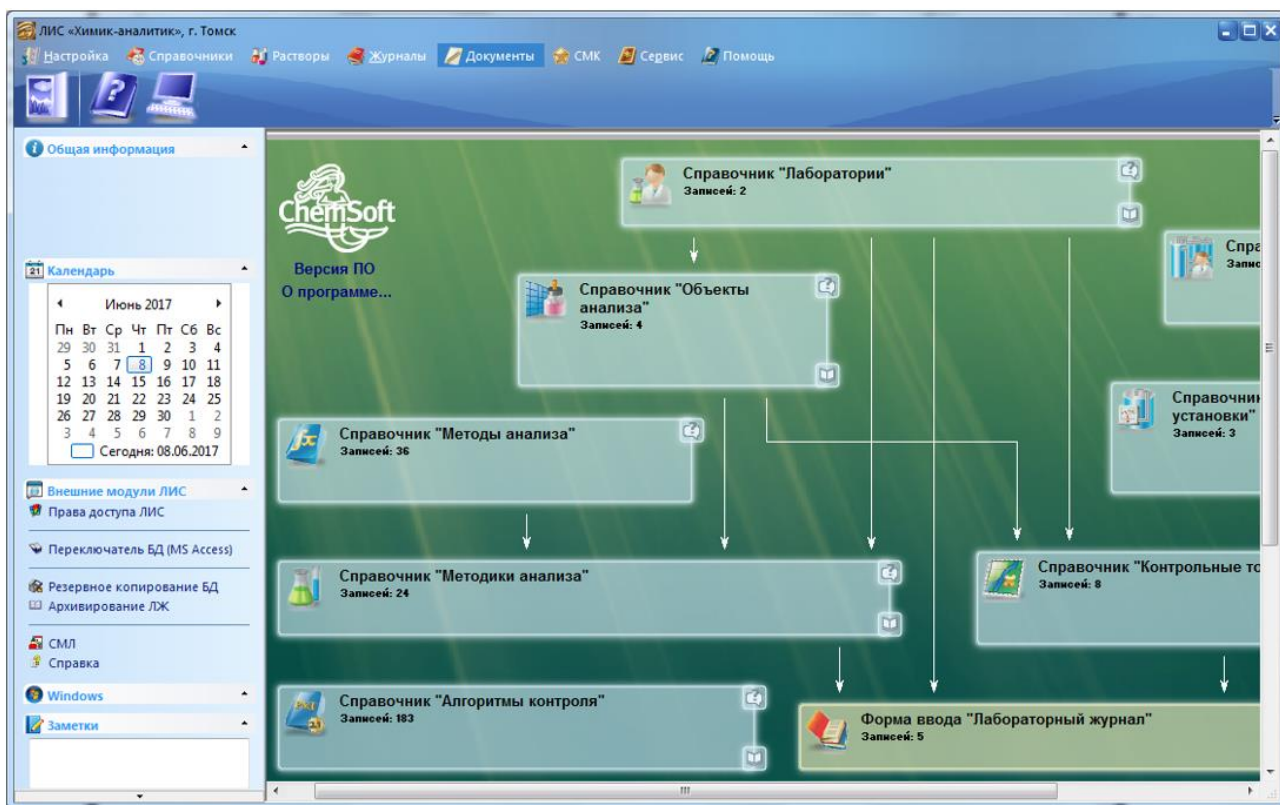


Рисунок 38 – Развернутая ЛИС «Химик-аналитик»

Время, затраченное на автоматизированную сборку данного дистрибутива, составляет 22 минуты. В среднем, согласно данным о трудозатратах, программист выполняет сборку дистрибутива ЛИС по данным заявки 45 минут. Более того, после заполнения заявки автоматизированная сборка выполняется в фоновом режиме и не требует внимания работника.

5 ФИНАНСОВЫЙ МЕНЕДЖМЕНТ

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

5.1.1 Потребители результатов исследования

Целевым потребителем системы автоматизированной сборки дистрибутивов является предприятие ООО «Химсофт». Руководством предприятия была поставлена задача по автоматизации снижения трудозатрат направленных на создание дистрибутивов лабораторной информационной системы «Химик-аналитик». Данная система состоит из множества модулей и имеет множество конфигураций, что сильно осложняет сборку ее дистрибутивов и приводит к большим трудозатратам.

При решении поставленной задачи рассматривались варианты использования стороннего программного обеспечения наряду с разработкой собственной системы по автоматизации данного процесса.

5.1.2 Анализ конкурентных технических решений

В данный момент на рынке существует множество аналогов, позволяющих создавать дистрибутивы различных проектов. При анализе предметной области и конкурентных технических решений были выделены и рассмотрены самые качественные разработки:

- Приложение «Actual Installer» (К1);
- Приложение «Inno Setup» (К2);
- Приложение «CreateInstall» (К3);

В приложении А представлена таблица сравнения данных программных продуктов.

Оценочная карта для сравнения конкурентных программных решений представлена в таблице 1.

Таблица 1 – Оценочная карта для сравнения конкурентных программных решений

Критерии оценки	Вес критерия	Баллы				Конкурентоспособность			
		Бф	Бк1	Бк2	Бк3	Кф	Кк1	Кк2	Кк3
Технические критерии оценки ресурсоэффективности									
Степень автоматизации процесса сборки дистрибутива ЛИС «Химик-аналитик»	0,4	5	1	2	2	2	0,4	0,8	0,8
Функциональная мощность	0,2	3	4	4	5	0,6	0,8	0,8	1
Простота использования	0,2	5	3	2	1	1	0,6	0,4	0,2
Область применения	0,1	1	4	4	5	0,1	0,4	0,4	0,5
Скорость выполнения сборки дистрибутива	0,1	5	4	4	5	0,5	0,4	0,5	0,5
Итого	1					4,2	2,6	2,9	3,0
Экономические критерии оценки ресурсоэффективности									
Стоимость внедрения решения	0,2	1	3	4	3	0,2	0,6	0,8	0,6
Поддержка продукта	0,6	5	1	1	2	3	0,6	0,6	1,2
Сроки внедрения	0,2	2	5	4	3	0,4	1	0,8	0,6
Итого	1					3,6	2,2	2,2	2,4

Экспертная оценка основных технических и экономических характеристик конкурентных программных решений показывает, что разрабатываемая система автоматизированной сборки дистрибутивов ЛИС «Химик-аналитик» является конкурентоспособной по сравнению с представленными аналогами.

Основным недостатком конкурентных программных продуктов являются невозможность достижения требуемого уровня автоматизации процесса сборки дистрибутивов - ПО рассчитанное на широкое применение просто не в состоянии учесть все особенности бизнес-процесса конкретного предприятия. Вторым

недостатком достаточно сложный пользовательский интерфейс, чрезмерно сложный способ расширения функционала по средству написания скриптов.

Разрабатываемая система предоставляет узкий спектр функциональных возможностей, однако она учитывает все тонкости бизнес-процессов предприятия и сводит к минимуму трудозатраты на сборку дистрибутивов. Интерфейс системы предоставляет широкие возможности по автозаполнению, его разработка согласовывалась с работниками предприятия, что обеспечивает максимальную простоту и удобство использования.

Таким образом, преимущество над разработками конкурентов обеспечивает высокая степень специализации.

5.1.3 SWOT-анализ

Для комплексного анализа научно-исследовательского проекта на основе анализа конкурентных решений была составлена матрица SWOT-анализа, содержащая сильные и слабые стороны проекта, а также возможности и угрозы для разработки проекта. Данная матрица представлена в приложении Б.

Для того, чтобы разобраться с различными комбинациями взаимосвязей областей матрицы SWOT, были построены интерактивные матрицы проекта, показывающие соответствия параметров SWOT-анализа.

Интерактивная матрица проекта полей «Сильные стороны и возможности» представлена в таблице 2.

Таблица 2 – Интерактивная матрица сильных сторон и возможностей проекта

Сильные стороны проекта								
		C1	C2	C3	C4	C5	C6	C7
Возможности проекта	B1	-	-	+	-	-	+	+
	B2	+	+	+	-	-	+	+
	B3	-	-	+	+	-	+	+
	B4	+	+	+	+	+	+	+
	B5	+	+	+	-	-	+	-

Интерактивная матрица проекта полей «Слабые стороны и возможности» представлена в таблице 3.

Таблица 3 – Интерактивная матрица слабых сторон и возможностей проекта

Слабые стороны проекта							
Возможности проекта		Сл1	Сл2	Сл3	Сл4	Сл5	Сл6
	В1	-	+	+	+	-	+
	В2	0	0	0	0	0	0
	В3	+	+	-	+	+	+
	В4	+	+	-	+	+	+
	В5	+	+	-	+	+	+

Интерактивная матрица проекта полей «Сильные стороны и угрозы» представлена в таблице 4.

Таблица 4 – Интерактивная матрица сильных сторон и угроз проекта

Сильные стороны проекта								
Угрозы проекта		С1	С2	С3	С4	С5	С6	С7
	У1	0	0	0	+	+	0	+
	У2	-	-	-	-	-	-	-
	У3	0	0	0	+	+	0	+
	У4	-	-	-	-	-	-	-
	У5	-	-	-	+	-	-	+

Интерактивная матрица проекта полей «Слабые стороны и угрозы» представлена в таблице 5.

Таблица 5 – Интерактивная матрица слабых сторон и угроз проекта

Слабые стороны проекта							
Угрозы проекта		Сл1	Сл2	Сл3	Сл4	Сл5	Сл6
	У1	+	-	-	+	+	-
	У2	+	-	-	-	-	-
	У3	-	-	-	-	-	-
	У4	-	-	-	-	-	-
	У5	0	0	-	+	+	-

Результаты анализа интерактивных матриц проекта представлены в матрице SWOT-анализа в приложении Б.

5.2 Определение возможных альтернатив научных исследований

Для определения возможных альтернативных путей проведения научных исследований использовался морфологический подход.

Морфологическими характеристиками являются:

- интегрированная среда разработки;
- язык программирования;
- формат хранения данных;
- графический интерфейс пользователя;
- способ программной выгрузки исходного кода
- способ программной сборки проектов
- способ программного архивирования

В таблице 6 представлена морфологическая матрица проекта.

Таблица 6 – Морфологическая матрица проекта

	Исполнение 1	Исполнение 2	Исполнение 3
А. Интегрированная среда разработки	Microsoft Visual Studio 2015	Microsoft Visual Studio 2017	-
Б. Язык программирования	C#	VB.NET	C++
В. Формат хранения данных	JSON	YAML	XML
Г. Графический интерфейс пользователя	WPF	WinForms	MFC
Д. Способ программной выгрузки исходного кода	Использование TFS API	Настройка Post-build events на сервере TFS	-

Е. Способ программной сборки проектов	Использование библиотеки MS Build	Настройка непрерывной интеграции	Написание скриптов с обращением к msbuild.exe
Ж. Способ программного архивирования	Написание скриптов с обращением к консольной версии архиватора RAR	Использование библиотеки DotNetZip	Использование библиотеки ZLib

Из данной морфологической матрицы проекта было выделено три варианта решения технической задачи:

И1. А1Б1В1Г1Д1Е1Ж1;

И2. А2Б2В2Г2Д2Е2Ж2;

И3. А1Б3В3Г3Д2Е3Ж3.

Эти варианты исполнения будут использованы в далее расчетах временных показателей научного исследования в приложении Г. Согласно проведенным расчетам вариант исполнения И1 был признан самым эффективным и был реализован на практике. Стоит отметить, что данный вариант является также наиболее гибким и позволяет легко производить сопровождение внедренной системы, вводить новый функционал.

5.3 Планирование научно-исследовательских работ

5.3.1 Структура работ в рамках научного исследования

Для планирования комплекса предполагаемых работ был составлен перечень этапов и работ в рамках проведения научного исследования, определены исполнители проекта и распределено время и этапы работ между исполнителями проекта. Исполнителями проекта являются студент и научный руководитель. Перечень этапов, работ и распределение исполнителей по данным видам работ в рамках проводимого научно-исследовательского проекта представлен в приложении В.

5.3.2 Определение трудоемкости выполнения работ

Определение трудоемкости работ каждого из участников проекта является важным этапом планирования научно-исследовательских работ, так как трудовые затраты в большинстве случаев образуют основную часть стоимости разработки.

Трудоемкость выполнения научного исследования оценивается экспертным путем в человеко-днях и носит вероятностный характер и рассчитывается с помощью длительности работ в рабочих и календарных днях каждого этапа работ.

По формуле 1, рассчитывается ожидаемое (среднее) значение трудоемкости выполнения каждой работы $t_{ожі}$ в человеко-днях.

$$T_{ожі} = \frac{3t_{mini} + 2t_{maxi}}{5}, \quad (1)$$

где $t_{ожі}$ – ожидаемая трудоёмкость выполнения i -ой работы, человеко-дни;

t_{mini} – минимально возможная трудоемкость выполнения заданной i -ой работы (оптимистическая оценка: в предположении наиболее благоприятного стечения обстоятельств), человеко-дни;

t_{maxi} – максимально возможная трудоемкость выполнения заданной i -ой работы (пессимистическая оценка: в предположении наиболее неблагоприятного стечения обстоятельств), человеко-дни.

По формуле 2, рассчитывается продолжительность каждой работы в рабочих днях T_{pi} , с учетом численности исполнителей на каждом этапе выполнения работ.

$$T_{pi} = \frac{t_{ожі}}{Ч_i}, \quad (2)$$

где T_{pi} – продолжительность i -ой работы, рабочие дни;

$t_{ожі}$ – ожидаемая трудоёмкость выполнения i -ой работы, человеко-дни;

$Ч_i$ – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, человек.

Для удобства построения графика проведения научного исследования необходимо перевести длительность каждого из этапов работ из рабочих в календарные дни с помощью формулы 3.

$$T_{ki} = T_{pi} \cdot k_{\text{кал}}, \quad (3)$$

где T_{ki} – продолжительность выполнения i -й работы в календарных днях;

T_{pi} – продолжительность выполнения i -й работы в рабочих днях;

$k_{\text{кал}}$ – коэффициент календарности.

Для расчёта длительности каждого из этапов работ в календарных днях необходимо рассчитать коэффициент календарности $k_{\text{кал}}$ используя формулу 4.

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - T_{\text{вых}} - T_{\text{пр}}}, \quad (4)$$

где $k_{\text{кал}}$ – коэффициент календарности;

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

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

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

В 2017 году количество календарных дней составляет 365 дней, а сумма выходных и праздничных дней равна 118 дням. Из этого следует, что коэффициент календарности для 2017 года равен $k_{\text{кал}} = 1,478$.

Для построения календарного плана-графика необходимо рассчитать временные показатели проведения научного исследования. Данные расчеты представлены в приложении Г.

5.3.3 Разработка графика проведения научного исследования

Для наглядного представления распределения работ участников проекта и затраченного времени была построена диаграмма Ганта – горизонтальный ленточный график, на котором работы по теме представляются протяженными во времени отрезками, характеризующимися датами начала и окончания выполнения данных работ. Построенная диаграмма Ганта представлена в приложении Д.

6 СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ

Целью выпускной квалификационной работы является разработка системы для автоматизированной сборки дистрибутивов программного обеспечения. Актуальность данной работы заключается в том, что разработанный программный продукт позволит сократить трудозатраты на создание дистрибутивов, что освободит работников предприятия от рутинной деятельности. При этом приложение предоставляет возможность снизить время, проводимое работниками за ЭВМ, что сократит влияние соответствующих вредных факторов, рассмотренных далее.

Также стоит отметить, что при разработке системы проводились различные теоретические исследования, анализ и структуризация полученных данных, а также проектирование и программирование системы с использованием ЭВМ.

Как разработка, так и эксплуатация программного обеспечения проходит за рабочим местом оператора персональной электронной вычислительной машины (ПЭВМ). Именно поэтому в данной части выпускной квалификационной работы приведено влияние вредных факторов, получаемых от использования средств вычислительной техники, а также способы его минимизации. При работе с ЭВМ на человека накладывается целый ряд вредных факторов, что впоследствии снижает производительность его труда и может привести к существенным проблемам со здоровьем.

Обеспечение производственной и экологической безопасности является необходимым условием реализации любых проектов, в том числе конструкторских и исследовательских. В общем, обеспечение безопасности предполагает создание безопасных и благоприятных рабочих условий для лиц, задействованных в работе над проектом, а также условий, обеспечивающих экологическую безопасность окружающей среды.

Первичным этапом в задаче обеспечения безопасности труда является выявление и анализ вредных и опасных факторов труда оператора ПЭВМ,

возможных причин потенциальных аварий и пожаров, производственных травм, профессиональных заболеваний. Следующими этапами в задачи обеспечения безопасности труда являются разработка мероприятий по защите вредных и опасных факторов, оценка условий труда и микроклимата рабочей среды.

Поэтому данный раздел посвящен анализу вредных и опасных факторов производственной среды для операторов ПЭВМ, разработке программ по минимизации воздействия вредоносного и опасного влияния выявленных факторов, а также программ по снижению вредных воздействий на окружающую среду, экономии невозможных ресурсов и защите в чрезвычайных ситуациях.

6.1 Производственная безопасность

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

Производственные условия на рабочем месте характеризуются наличием различных опасных и вредных производственных факторов, оказывающих негативное влияние на работников.

Производственный фактор считается вредным, если воздействие этого фактора на работника может привести к его заболеванию. Производственный фактор считается опасным, если его воздействие на работника может привести к его травме [1].

Вредные факторы характеризуются потенциальной опасностью для здоровья, в частности способствуют развитию каких-либо заболеваний, приводят к повышенной утомляемости и снижению работоспособности. При этом, вредные факторы проявляются при определенных условиях таких как интенсивность и длительность воздействия. Опасные производственные факторы способны моментально оказать влияние на здоровье работника: привести к травмам, ожогам или к резкому ухудшению здоровья работников в результате отравления или облучения.

В таблице 7 представлены возможные вредные и опасные факторы, возникающие при работе за ПЭВМ.

Таблица 7 – Вредные и опасные факторы, возникающие при работе за ПЭВМ

Наименование видов работ	Факторы	Нормативные документы
Вредные факторы		
Работа за ПЭВМ	Отклонение показателей микроклимата (температуры и влажности воздуха)	СанПиН 2.2.2/2.4.1340-03 [2] СанПиН 2.2.4.548-96 [3]
	Недостаточная освещенность рабочей зоны	СанПиН 2.2.2/2.4.1340-03
Опасные факторы		
Работа за ПЭВМ	Опасность поражения электрическим током	ГОСТ 12.1.038–82 [4]
	Пожаровзрывоопасность	ФЗ от 22.07.2008 №123-ФЗ [5]

6.1.1 Вредные производственные факторы

6.1.1.1 Отклонение показателей микроклимата в помещении

Одним из необходимых благоприятных условий труда является обеспечение в помещениях нормальных условий микроклимата, оказывающих существенное влияние на тепловое самочувствие человека. Микроклимат в производственных помещениях, зависит от особенностей технологического процесса, а также внешних условий (категории работ, периода года, условий вентиляции и отопления).

К параметрам, характеризующим микроклимат в производственных помещениях, относятся:

- Температура воздуха (t , °С);
- Температура поверхностей (t , °С);
- Относительная влажность воздуха (ϕ , %);
- Скорость движения воздуха (v , м/с);
- Интенсивность теплового облучения (I , Вт/м²).

В производственных помещениях для работы с ПЭВМ происходит постоянное выделение тепла самой вычислительной техникой, вспомогательными приборами и средствами освещения. Поскольку оператор расположен в непосредственной близости с источниками выделения тепла, то данный фактор является одним из важнейших вредных факторов производственной среды оператора ПЭВМ, а высокая температура воздуха способствует быстрому перегреву организма и быстрой утомляемости [6].

Влажность оказывает большое влияние на терморегуляцию организма. Так, например, высокие показатели относительной влажности (более 85 %) затрудняют терморегуляцию снижая возможность испарения пота, низкие показатели влажности (менее 20 %) вызывают пересыхание слизистых оболочек человека [7].

Работа программиста относится к категории Ia, которые производятся сидя и сопровождаются незначительными физическим напряжением. Интенсивность энерготрат организма для данной категории работ составляет до 120 ккал/ч (до 139 Вт).

Оптимальные значения показателей микроклимата на рабочих местах производственных помещений согласно СанПиН 2.2.4.548-96 для категории работ Ia представлены в таблице 8. [3]

Таблица 8 – Оптимальные величины показателей микроклимата на рабочих местах производственных помещений

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

Оптимальные микроклиматические условия обеспечивают общее и локальное ощущение теплового комфорта при минимальном напряжении механизмов терморегуляции, не вызывают отклонений в состоянии здоровья,

создают предпосылки для высокого уровня работоспособности и являются предпочтительными на рабочих местах.

В таблице 9 приведены допустимые величины показателей микроклимата на рабочих местах производственных помещений согласно СанПиН 2.2.4.548-96 для категории работ Ia. [3]

Таблица 9 – Допустимые величины показателей микроклимата на рабочих местах производственных помещений

Период года	Категория работ	Температура воздуха, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Ia	20 – 25	15 – 75	0,1
Теплый	Ia	21 – 28	15 – 75	0,1 – 0,2

Допустимые микроклиматические условия не вызывают повреждений или нарушений состояния здоровья, но могут приводить к возникновению общих и локальных ощущений теплового дискомфорта, напряжению механизмов терморегуляции, ухудшению самочувствия и понижению работоспособности.

Согласно требованиям СанПиН 2.2.2/2.4.1340–03, в кабинете поддерживается температура равная 19–20 С°, при относительной влажности в 55–58%. Для этого в помещении проводится ежедневная влажная уборка и систематическое проветривание после каждого часа работы на ПЭВМ [2].

6.1.1.2 Недостаточная освещенность рабочей зоны

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

Работа с компьютером подразумевает постоянный зрительный контакт с дисплеем ПЭВМ и занимает от 80 % рабочего времени.

Недостаточный уровень освещенности в помещении приводит к снижению остроты зрения, головным болям, снижению концентрации внимания и, как следствие, к ухудшению производительности труда.

Причиной недостаточной освещенности являются недостаточность естественного освещения, недостаточность искусственного освещения, пониженная контрастность.

Рабочее помещение должно иметь как естественное, так и искусственное освещение. Коэффициент естественного освещения должен быть не менее 1,2%. Согласно СанПиН 2.2.2/2.4.1340-03 освещенность на поверхности рабочего стола в зоне размещения документа должна быть 300 – 500 лк, что может достигаться установкой местного освещения, не создающего бликов на поверхности экрана. Освещенность поверхности экрана не должна превышать 300 лк. Яркость светящихся поверхностей (окон, светильников), находящихся в поле зрения должна быть не более 200 кд/м². Для источников искусственного освещения следует применять люминесцентные лампы типа ЛБ и компактные люминесцентные лампы (КЛЛ). Коэффициент пульсации при работе с ПЭВМ не должен превышать 5%.

Следует ограничивать отраженную блескость на рабочих поверхностях (экран, стол, клавиатура) за счет правильного выбора и расположения светильников, яркость бликов на экране не должна превышать 40 кд/м². Светильники местного освещения должны иметь непросвечивающий отражатель.

6.1.2 Опасные производственные факторы

6.1.2.1 Опасность поражения электрическим током

Поражение электрическим током является опасным производственным фактором и, поскольку оператор ПЭВМ имеет дело с электрооборудованием, то вопросам электробезопасности на его рабочем месте должно уделяться много внимания.

Опасность поражения человека электрическим током оценивается величиной тока I (А), проходящего через его тело, или напряжением прикосновения U (В). Степень опасного воздействия на человека электрического тока зависит от рода и величины напряжения тока, частоты электрического тока,

пути тока через тело человека, продолжительности его воздействия на организм человека, а также условий внешней среды.

Электрический ток, протекая через тело человека, производит термическое, механическое и световое воздействие – электролитическое разложение жидкости (в том числе и крови), судорожное сокращение мышц, разрыв тканей и поражение глаз.

Работа с ПЭВМ является опасной с точки зрения поражения током, так как практически во всех частях компьютера течет электрический ток. Поражение электрическим током при работе в ПЭВМ возможно при наличии оголенных участков на кабеле, нарушении изоляции распределительных устройств и от токоведущих частей компьютера в случае их пробоя и нарушении изоляции, при работе с ПЭВМ во влажной одежде и влажными руками.

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

Мероприятия по снижению риска поражения электрическим током приведены в пункте 6.1.3.6.

6.1.2.2 Пожаровзрывобезопасность

Возникновение пожара является опасным производственным фактором, т.к. пожар на предприятии наносит большой материальный ущерб, а также часто сопровождается травмами и несчастными случаями.

Пожарная безопасность представляет собой единый комплекс организационных, технических, режимных и эксплуатационных мероприятий по предупреждению пожаров и взрывов.

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.

Мероприятия по снижению риска возникновения пожара приведены в пункте 6.1.3.7.

6.1.3 Рекомендации по минимизации вредных и опасных факторов

6.1.3.1 Рекомендации по режиму труда и отдыха оператора

В целях снижения нервного напряжения, утомления зрительной и опорно-двигательной систем оператора рекомендуется следующий режим его работы.

Таблица 10 -Режим работы оператора ПК

Категория работы с ПК	Уровень нагрузки за рабочую смену			Суммарное время перерывов, мин	
	Группа А, кол-во знаков	Группа Б, кол-во знаков	Группа В, ч	при 8-часовой смене	при 12-часовой смене
I	До 20 тыс.	15 тыс.	До 2	30	70
II	До 40 тыс.	30 тыс.	До 4	50	90
III	До 60 тыс.	40 тыс.	До 6	70	120

Перерывы в течение рабочего дня при 8-часовой смене по количеству и продолжительности распределяются следующим образом:

- Для I категории - 2 перерыва по 15 минут через 2 часа после начала смены и через 2 часа после обеденного перерыва;

- Для II категории - через 2 часа после начала смены и через 1,5-2 часа после обеденного перерыва по 15 минут каждый или по 10 минут через каждый час работы;

- Для III категории - через 1,5-2 часа после начала смены и через 1,5-2 часа после обеденного перерыва по 20 минут каждый или по 15 минут через каждый час.

При 12-часовой смене перерывы в первые 8 часов такие же, как и при 8-часовой смене; в течение последних 4 часов (независимо от категории и вида работ) - каждый час по 15 минут. Не рекомендуется работать на ПК более 2 часов подряд без перерыва.

В процессе работы для уменьшения отрицательного влияния монотонности рекомендуется менять вид работы, например, чередовать ввод данных и редактирование, считывание информации и ее осмысление.

6.1.3.2 Рекомендуемые мероприятия по оптимизации зрительной работоспособности и световой обстановки

Среди данных мероприятий стоит выделить следующие:

- Улучшение световой обстановки путем обеспечения помещений естественным и достаточным искусственным освещением, рациональным расположением рабочих мест по отношению к оконным проемам и светильникам искусственного освещения;

- Снижение зрительного утомления путем снижения пульсации светового потока, исключения бликов отражения на экранах мониторов, использования экранов защиты, фильтров с антибликовым покрытием, очков для пользователей ПЭВМ и рационального использования режимов труда и отдыха.

Общее освещение следует выполнять светильниками с экранирующими решетками и отражателями, обеспечивающими отсутствие зеркальных отражений светящихся поверхностей светильников на экранах дисплеев и на горизонтальных поверхностях, а также необходимую степень контраста машинописных и рукописных текстов.

Светильники общего освещения следует располагать над рабочими поверхностями в равномерно-прямоугольном порядке.

Осветительные установки должны обеспечивать равномерную освещенность с помощью преимущественно отраженного или рассеянного светораспределения; они не должны создавать слепящих бликов на клавиатуре и других частях пульта, а также на экране видеотерминала в направлении глаз оператора.

При рядном размещении оборудования (ПК) не допускается расположение дисплеев экранами друг к другу. Также пульсация освещенности используемых люминесцентных ламп не должна превышать 10%.

Улучшение условий считывания информации с экрана монитора ПК осуществляется путем применения:

- очков со специальным покрытием.
- защитных экранов
- применение стереопар
- регулярный прием витаминов группы А, В и С и бета-каротина

6.1.3.3 Мероприятия по улучшению состояния воздушной среды рабочих помещений

Мероприятия по улучшению состояния воздушной среды рабочих помещений с ПК включают:

- Применение вентиляции и кондиционирования воздуха;
- Уменьшение тепловыделений от мониторов ПК;
- Применение ионизаторов (или люстр Чижевского);
- Использование специальных увлажнителей, комнатных растений;
- Влажную ежедневную уборку помещений.

Улучшение состояния воздушной среды рабочих помещений достигается обеспечением соответствующей площади и объема производственного

помещения и устройством эффективных систем вентиляции и кондиционирования.

Кондиционирование воздуха обеспечивает автоматическое поддержание параметров микроклимата в необходимых пределах в течение всех сезонов года, очистку воздуха от пыли и вредных веществ, создание небольшого избыточного давления в чистых помещениях для исключения поступления неочищенного воздуха.

Для повышения влажности воздуха в помещениях с ПЭВМ (ПК) следует:

- применять увлажнители воздуха, заправляемые ежедневно дистиллированной или прокипяченной питьевой водой;
- использовать комнатные растения и цветы;
- в помещениях иметь аквариумы.

6.1.3.4 Мероприятия по снижению шума

В помещениях с ПЭВМ (ПК), где уровень шума превышает допустимые значения, для его устранения должны проводиться организационные, строительно-акустические и другие мероприятия (такие как облицовка стен, покрытие пола, использование средств индивидуальной защиты).

Шумящее оборудование (АЦПУ, принтеры и т.п.), уровни шума которого превышают нормированные, должно находиться вне помещения с ПЭВМ.

Мерой борьбы с шумом является рациональная планировка помещений с ПК, позволяющая исключить проникновение шумов из соседних помещений.

6.1.3.5 Рекомендации при работе с системой по автоматизированной сборке дистрибутивов.

Процесс сборки дистрибутива включает в себя скачивание исходного кода, сборку из него самой системы, архивирование файлов. Эти процессы наиболее требовательны к ресурсам ЭВМ, поэтому центральный процессор нагревается достаточно сильно, также при их выполнении система охлаждения работает максимально интенсивно. Работа вентиляторов становится слышна и

вызывает у некоторых пользователей дискомфорт при работе, хотя шумовые показатели и находятся в пределах нормы.

Таким образом были сформированы следующие рекомендации:

- запускать процесс сборки дистрибутива наиболее удобно перед перерывом в работе. После начала сборки процесс не требует вмешательства, а при возникновении ошибок будет предоставлено их описание;
- необходимо убедиться, что система охлаждения ЭВМ функционирует исправно – процесс сборки очень требователен к ресурсам ЭВМ и сильно разогревает аппаратную часть;
- при наличии требований и пожеланий к интерфейсу и функционалу системы следует обращаться к разработчику. Это позволит сократить время работы с ЭВМ, а также умственную нагрузку при составлении документа «заявка на сборку» и создании дистрибутива;
- после сборки дистрибутива рекомендуется проветрить рабочее помещение.

6.1.3.6 Мероприятия по предотвращению возможности поражения электрическим током

К мероприятиям по предотвращению возможности поражения электрическим током относятся:

- При производстве монтажных работ необходимо использовать только исправный инструмент, аттестованный службой КИПиА;
- С целью защиты от поражения электрическим током, возникающим между корпусом приборов и инструментом при пробое сетевого напряжения на корпус, корпуса приборов и инструментов должны быть заземлены;
- При включенном сетевом напряжении работы на задней панели должны быть запрещены;
- Все работы по устранению неисправностей должен производить квалифицированный персонал;

Необходимо постоянно следить за исправностью электропроводки [2].

Согласно ГОСТ 12.1.038-82 на рабочем месте программиста допускаются уровни напряжений прикосновения и токов, представленные в таблице 11. [4]

Таблица 11 – Предельно допустимые напряжения прикосновения и токи

Род тока	Напряжение прикосновения, В	Ток, мА
	не более	
Переменный, 50 Гц	2,0	0,3
Постоянный	8,0	1,0

Значения напряжения прикосновения и токов приведены при продолжительности воздействия не более 10 минут в сутки.

6.1.3.7 Мероприятия по предотвращению причин возникновения пожаров

Для устранения возможных причин возникновения пожаров необходимо проводить следующие мероприятия:

- **Организационные мероприятия:**
 - противопожарный инструктаж обслуживающего персонала;
 - обучение персонала техники безопасности;
 - разработка инструкций, плакатов, планов эвакуации.
- **Эксплуатационные мероприятия:**
 - соблюдение эксплуатационных норм оборудования;
 - выбор и использование современных автоматических средств тушения пожаров.
- **Технические мероприятия:**
 - профилактический осмотр и ремонт оборудования;
 - соблюдение противопожарных мероприятий при устройстве электропроводок, оборудования, систем отопления, вентиляции и освещения.

6.2 Экологическая безопасность

6.2.1 Анализ воздействия продукта на окружающую среду

Разработанный программный продукт, не наносит вреда окружающей среде ни на стадиях его разработки, ни на стадиях эксплуатации. Однако, средства, необходимые для его разработки и эксплуатации могут наносить вред окружающей среде.

Объект, на котором производилась разработка продукта, а также объекты, на которых будет производиться его использование операторами ПЭВМ относятся к предприятиям пятого класса, размер санитарной зоны для которых равен 50 м.

Основными факторами, оказывающими негативные действия на экологию, являются факторы, связанные с производством и эксплуатацией компьютерной техники. В частности, отходы и выбросы, имеющие место на этапе производства компьютеров, а также отходы, связанные с неполной их утилизацией.

Эксплуатация компьютерной техники может сопровождаться следующими негативными факторами влияния на окружающую среду:

локальное повышение электромагнитного и радиоактивного фона;

неоправданное потребление электроэнергии (связано с использованием компьютера не на полную мощность в течение всего его время работы) и прочее.

6.2.2 Решения по обеспечению экологической безопасности

При разработке любых автоматизированных систем возникает необходимость утилизировать производственные отходы, в качестве которых в данном случае выступают бумажные отходы (макулатура) и неисправные детали персональных компьютеров, плат, контроллеров.

Бумажные отходы должны передаваться в соответствующие организации для дальнейшей переработки во вторичные бумажные изделия. Неисправные комплектующие персональных компьютеров должны передаваться либо

государственным организациям, осуществляющим вывоз и уничтожение бытовых и производственных отходов, либо организациям, занимающимся переработкой отходов. Важнейшим этапам обращения с отходами является их сбор, а в дальнейшем переработка, утилизация и захоронение [8].

6.3 Безопасность в чрезвычайных ситуациях

6.3.1 Перечень возможных ЧС при разработке и эксплуатации проектируемого решения

Возможными чрезвычайными ситуациями могут быть:

- техногенные: взрывы, пожары, обрушение помещений, аварии на системах жизнеобеспечения;
- природные: наводнения, ураганы, бури, природные пожары;
- экологические: разрушение озонового слоя, кислотные дожди;
- биологические: эпидемии, пандемии;
- антропогенные: война, терроризм.

Общие правила поведения при чрезвычайных ситуациях:

- 2 Не паниковать и не поддаваться панике. Призывать окружающих к спокойствию.
- 3 По возможности немедленно позвонить по телефону «01», сообщить что случилось, указать точный адрес места происшествия, назвать свою фамилию и номер своего телефона.
- 4 Включить устройства передачи звука (радио, телевизор), а также прослушать информацию, передаваемую через уличные громкоговорители и громкоговорящие устройства. В речевом сообщении будут озвучены основные рекомендации и правила поведения.
- 5 Выполнять рекомендации специалистов (сотрудников полиции, медицинских работников, пожарных, спасателей).
- 6 Не создавать условия, которые препятствуют и затрудняют действия сотрудников полиции, медицинских работников, спасателей, пожарных.

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

Причинами возникновения данного вида ЧС могут являться:

- возникновением короткого замыкания в электропроводке;
- возгоранием устройств ПЭВМ из-за неисправности аппаратуры;
- возгоранием устройств искусственного освещения;
- возгоранием мебели по причине нарушения правил пожарной безопасности, а также неправильного использования дополнительных бытовых электроприборов и электроустановок.

Помещение для работы операторов ПЭВМ по системе классификации категорий помещений по взрывопожарной и пожарной опасности относится к категории Д (из 5-ти категорий А, Б, В1-В4, Г, Д), т.к. относится к помещениям с негорючими веществами и материалами в холодном состоянии [9].

6.3.2 Разработка действий в результате возникшей ЧС и мер по ликвидации её последствий.

Пожарная безопасность подразумевает надлежащее состояние объекта с исключением возможности возникновения очага возгорания (пожара) и его распространения в пространстве. Обеспечение пожарной безопасности — приоритетная задача для любого предприятия. Создание системы защиты регламентировано законом и нормативными документами различных ведомств.

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

Запрещается использовать электроприборы в условиях, не соответствующих требованиям инструкций изготовителей, или имеющие неисправности, которые в соответствии с инструкцией по эксплуатации могут привести к пожару, а также эксплуатировать электропровода и кабели с поврежденной или потерявшей защитные свойства изоляцией. Электроустановки и бытовые электроприборы в помещениях по окончании

рабочего времени должны быть обесточены (вилки должны быть вынуты из розеток). Под напряжением должны оставаться дежурное освещение и пожарная сигнализация. Недопустимо хранение легковоспламеняющихся, горючих и взрывчатых веществ, использование открытого огня в помещениях офиса.

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

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

Немедленно прекратить работу и вызвать пожарную охрану по телефону «01», сообщив при этом адрес, место возникновения пожара и свою фамилию;

Принять по возможности меры по эвакуации людей и материальных ценностей;

Отключить от сети закрепленное за ним электрооборудование;

Приступить к тушению пожара имеющимися средствами пожаротушения;

Сообщить непосредственному или вышестоящему начальнику и оповестить окружающих сотрудников;

При общем сигнале опасности покинуть здание согласно «Плану эвакуации людей при пожаре и других ЧС».

Для тушения пожара применять ручные углекислотные огнетушители (типа ОУ-2, ОУ-5), находящиеся в помещениях офиса, и пожарный кран внутреннего противопожарного водопровода. Они предназначены для тушения начальных возгораний различных веществ и материалов, за исключением веществ, горение которых происходит без доступа воздуха. Огнетушители должны постоянно содержаться в исправном состоянии и быть готовыми к

действию. Категорически запрещается тушить возгорания в помещениях офиса при помощи химических пенных огнетушителей (типа ОХП-10) [10].

6.4 Правовые и организационные вопросы обеспечения безопасности

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

Режим труда и отдыха предусматривает соблюдение определенной длительности непрерывной работы на персональном компьютере (ПК) и перерывов, регламентированных с учетом продолжительности рабочей смены, видов и категории трудовой деятельности.

Вид трудовой деятельности на персональном компьютере в рамках данной работы соответствует группе В – творческая работа в режиме диалога с ПК, категория трудовой деятельности – III (до 6 часов непосредственной работы на ПК).

При 8-часовой рабочей смене и работе на ПК, соответствующей описанным выше критериям необходимо через 1,5- 2,0 часа от начала рабочей смены и через 1,5-2,0 часа после обеденного перерыва устраивать регламентированные перерывы продолжительностью 20 минут каждый или продолжительностью 15 минут через каждый час работы.

Продолжительность непрерывной работы на ПК без регламентированного перерыва не должна превышать 2 часа.

Эффективными являются нерегламентированные перерывы (микропаузы) длительностью 1-3 минуты.

Регламентированные перерывы и микропаузы целесообразно использовать для выполнения комплекса упражнений и гимнастики для глаз, пальцев рук, а также массажа. Комплексы упражнений целесообразно менять через 2-3 недели.

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

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

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

Организация обязана предоставлять ежегодный отпуск продолжительностью 28 календарных дней. Дополнительные отпуска предоставляются работникам, занятым на работах с вредными или опасными условиями труда, работникам имеющими особый характер работы, работникам с ненормированным рабочим днем и работающим в условиях Крайнего Севера и приравненных к нему местностях. [11]

6.4.1 Организационные мероприятия при компоновке рабочей зоны

Большое значение для профилактики статических физических перегрузок имеет правильная организация рабочего места человека, работающего с ПЭВМ. Рабочее место должно быть организовано в соответствии с требованиями стандартов, технических условий и (или) методических указаний по безопасности труда. Оно должно удовлетворять следующим требованиям:

обеспечивать возможность удобного выполнения работ;

учитывать физическую тяжесть работ;

учитывать размеры рабочей зоны и необходимость передвижения в ней работающего;

учитывать технологические особенности процесса выполнения работ.

Невыполнение требований к расположению и компоновке рабочего места может привести к получению работником производственной травмы или развития у него профессионального заболевания. Рабочее место программиста должно соответствовать требованиям СанПин 2.2.2/2.4.1340-03.

Конструкция оборудования и рабочего места при выполнении работ в положении сидя должна обеспечивать оптимальное положение работающего, которое достигается регулированием высоты рабочей поверхности, высоты сидения, оборудованием пространства для размещения ног и высотой подставки для ног. Схемы размещения рабочих мест с персональными компьютерами должны учитывать расстояния между рабочими столами с мониторами: расстояние между боковыми поверхностями мониторов не менее 1,2 м, а расстояние между экраном монитора и тыльной частью другого монитора не менее 2,0 м. Клавиатура должна располагаться на поверхности стола на расстоянии 100-300 мм от края, обращенного к пользователю. Быстрое и точное считывание информации обеспечивается при расположении плоскости экрана ниже уровня глаз пользователя, предпочтительно перпендикулярно к нормальной линии взгляда (нормальная линия взгляда 15 градусов вниз от горизонтали). Рабочие места с ПЭВМ при выполнении творческой работы, требующей значительного умственного напряжения или высокой концентрации внимания, рекомендуется изолировать друг от друга перегородками высотой 1,5 - 2,0 м.

6.4.2 Специфика влияния продукта на рабочий процесс

Разрабатываемый в ходе выполнения ВКР программный продукт служит для автоматизации бизнес-процесса предприятия ООО «Химсофт» по сборке дистрибутивов лабораторной информационной системы «Химик-аналитик».

Инструкция по сборке дистрибутива содержит 23 пункта, практически все ее операции можно повторить программно. Разработанный автосборщик дистрибутивов позволит выполнить эти операции в фоновом режиме, требуя для работы заполненную заявку на сборку. Таким образом дистрибутив позволяет сэкономить время на выполнение бизнес-процесса, а также снизить умственную нагрузку на работников предприятия, так как первые сборки дистрибутива у неподготовленного работника могут занять длительное время и потребовать разъяснения по пунктам инструкции.

После внедрения, заявления на сборку хранятся и заполняются в разработанной системе. Написав нескольких заявок, освоив разработанную систему, работники предприятия смогут создавать их быстрее, чем ранее. Это достигается благодаря выпадающим спискам, предлагающим варианты заполнения, шаблонам заполнения, также влияет тот факт, что система требует гораздо меньше ресурсов, чем редактор MS Word.

Для обеспечения защиты от злоумышленников и конфигурирования системы «Химик-аналитик», ранее в обязательном порядке требовалось производить манипуляции с исходным кодом системы в интегрированной среде разработки MS Visual Studio, что требовало наличия профессиональных знаний и запуска требовательного ПО. Разработанная система по автоматизированной сборке дистрибутивов позволяет выполнить эти действия заполнив лишь несколько полей «заявки на сборку». Это наиболее существенно снижает умственную нагрузку на создателя дистрибутива.

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы достигнута ее основная цель – была спроектирована и разработана система автоматизированной сборки программного обеспечения.

Данное приложение, созданное на платформе WPF, отвечает всем заявленным требованиям и позволяет автоматизировать процесс создания инсталляторов ЛИС «Химик-аналитик» на предприятии ООО «Химсофт».

В процессе разработки были проанализированы различные варианты программной сборки проектов, выгрузки исходных файлов из системы контроля версий, программной архивации. Из них были выбраны и использованы самые надежные и эффективные альтернативы.

Программный продукт позволяет сократить трудозатраты на сборку дистрибутивов, освободить инженеров предприятия от задачи, требующей проявления высокой концентрации.

Данная работа является многогранной и сочетает в себе множество сложных программных решений – от рекурсивных функций до создания пользовательских элементов управления.

В рамках развития проекта в будущем планируется дорабатывать его согласно пожеланиям работников предприятия.

Список использованных источников

1. Охрана труда [Электронный ресурс] / Безопасность жизнедеятельности. URL: <http://www.grandars.ru/shkola/bezопасnost-zhiznedeyatelnosti/ohrana-truda.html>, свободный. Яз. Рус. Дата обращения: 16.05.2017.
2. СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы. Яз. Рус. Дата обращения: 16.05.2017.
3. СанПиН 2.2.4.548-96 Гигиенические требования к микроклимату производственных помещений. Яз. Рус. Дата обращения: 16.05.2017.
4. ГОСТ 12.1.038–82 Система стандартов безопасности труда. Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов. Яз. Рус. Дата обращения: 16.05.2017.
5. Федеральный закон от 22.07.2008 N 123-ФЗ (ред. От 13.07.2015) «Технический регламент о требованиях пожарной безопасности» [Электронный ресурс] / КонсультантПлюс. URL: http://www.consultant.ru/document/cons_doc_LAW_78699/, свободный. Яз. Рус. Дата обращения: 16.05.2017.
6. Ефремова О. С. Требования охраны труда при работе на персональных электронно-вычислительных машинах. – 2-е изд., перераб. и доп. – М. : Издательство «Альфа-Пресс», 2008. Яз. Рус. Дата обращения: 16.05.2017.
7. 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. Яз. Рус. Дата обращения: 16.05.2017.

8. ГОСТ Р 55090-2012 Ресурсосбережение. Обращение с отходами. Рекомендации по утилизации отходов бумаги. Яз. Рус. Дата обращения: 16.05.2017.
9. НПБ 105-03 Определение категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности [Электронный ресурс] / Электронный фонд правовой и нормативно-технической документации. URL: <http://docs.cntd.ru/document/1200032102>, свободный. Яз. Рус. Дата обращения: 16.05.2017.
10. ППБ 01–03. Правила пожарной безопасности в Российской Федерации. – М.: Министерство Российской Федерации по делам гражданской обороны, чрезвычайным ситуациям и ликвидации последствий стихийных бедствий, 2003.
11. Трудовой кодекс Российской Федерации" от 30.12.2001 N 197-ФЗ (ред. от 03.07.2016) (с изм. и доп., вступ. в силу с 01.01.2017) [Электронный ресурс] / Консультант Плюс. URL: https://www.consultant.ru/document/cons_doc_LAW_34683/, свободный. Яз. Рус. Дата обращения: 16.05.2017.
12. <https://habrahabr.ru/post/255807/>
13. <http://www.actualinstaller.com/articles/how-to-create-installer.html>
14. [https://msdn.microsoft.com/en-us/library/ff977206\(v=vs.100\).aspx](https://msdn.microsoft.com/en-us/library/ff977206(v=vs.100).aspx)
15. https://professorweb.ru/my/WPF/binding_and_styles_WPF/level8/8_2.php

ПРИЛОЖЕНИЕ А

Таблица сравнения программных продуктов, используемых для создания дистрибутивов

Программный продукт	Поддержка редактирования реестра и INI файлов	Поддержка выборочной и многоязычной установки	Возможность выбора разных алгоритмов сжатия	Создание полноценного деинсталлятора	Тип лицензии	Сложность освоения	Возможность проверки цифровой подписи	Необходимость использования скриптовых языков
Приложение «Actual Installer»	Да	Да	Нет	Нет	Бесплатная	Низкая	Нет	Нет
Приложение «Inno Setup»	Да	Да	Нет	Да	Условно-бесплатная	Высокая	Да	Да
Приложение «CreateInstall»	Да	Да	Да	Да	Платная	Высокая	Да	Да
Приложение «Smart Install Maker» "	Да	Да	Нет	Нет	Условно-бесплатная	Низкая	Нет	Нет

ПРИЛОЖЕНИЕ Б

Матрица SWOT-анализа

	<p>Сильные стороны научно-исследовательского проекта:</p> <p>С1. Высокая степень автоматизации процесса сборки дистрибутива.</p> <p>С2. Простой интерфейс пользователя.</p> <p>С3. Механизмы автозаполнения «заявок на сборку».</p> <p>С4. Возможность интеграции системы с действующими утилитами предприятия и системой контроля версий.</p> <p>С5. Программное скачивание исходного кода и сборка проектов без участия пользователя.</p> <p>С6. Замена бумажного документооборота «заявок на сборку»</p> <p>С7. Отличная масштабируемость проекта</p>	<p>Слабые стороны научно-исследовательского проекта:</p> <p>Сл1. Узкая направленность разработки.</p> <p>Сл2. При внесении изменений в ЛИС «Химик-аналитик» может потребоваться доработка продукта.</p> <p>Сл3. Требуется наличие Интернет-соединения для корректной работы.</p> <p>Сл4. Высокая сложность и большие трудозатраты разработки</p> <p>Сл5. Использование большого количества сторонних библиотек, чей код не контролируется.</p> <p>Сл6. Отсутствие команды разработчиков</p>
<p>Возможности:</p> <p>В1. Автоматизация доставки дистрибутива заказчикам (программная загрузка на сайт).</p> <p>В2. Доработка по пожеланиям пользователей.</p> <p>В3. Интеграция с действующими разработками предприятия ООО «Химсофт»</p>	<p>Направления развития:</p> <p>1. В1В2В3С3С6С7 – Увеличение степени автоматизации процесса создания дистрибутивов</p> <p>2. В2В5С1С2 – Повышение удобства использования</p> <p>3. В4В5С3С6 – Повышение надежности системы</p> <p>4. В3В4С4С7 – Интеграция с</p>	<p>Сдерживающие факторы:</p> <p>1. В1В3В4В5Сл2Сл4Сл6 – В связи со сложностью разработки возможен выход за предполагаемые временные и бюджетные рамки.</p> <p>2. В1Сл3 – Усиление зависимости от наличия Интернет-соединения</p> <p>3. В3В4Сл1Сл5 – Усиление зависимости от стороннего ПО и подключаемых библиотек</p>

<p>В4. Развитие ЛИС «Химик-аналитик», создание новых модулей и утилит.</p> <p>В5. Развитие проверки корректности ввода заявок на сборку.</p>	<p>используемыми разработками</p>	<p>4. В3В5Сл4Сл5 – Ввод сложных алгоритмов и дополнительных зависимостей приведет к запутанности кода и увеличит сложность сопровождения</p>
<p>Угрозы:</p> <p>У1. Некорректная работа используемых библиотек</p> <p>У2. Нежелание пользователей ЛИС обновлять систему</p> <p>У3. Чрезмерно быстрое развитие ЛИС «Химик-аналитик»</p> <p>У4. Переход на новый способ распространения системы.</p> <p>У5. Возможность наличия не выявленных ошибок</p>	<p>Угрозы развития:</p> <p>1. У1У3С4С5С7 – сложности сопровождения продукта могут возникнуть при нахождении ошибок в используемых библиотеках, либо при слишком быстром развитии распространяемой ЛИС.</p> <p>2. У1У5С4С7 – Интеграция с разработками и дополнение функционала повышают нестабильность системы.</p>	<p>Уязвимости:</p> <p>1. У1У5Сл4Сл5 – сбои работы программы при сборке дистрибутива могут привести к непредвиденным последствиям после установки ЛИС.</p> <p>2. У2Сл1 – Отказ пользователей ЛИС обновляться лишает разработку актуальности.</p> <p>3. У4Сл1 – Переход на принципиально новый способ распространения ЛИС лишает разработку актуальности.</p>

ПРИЛОЖЕНИЕ В

Перечень этапов, работ и распределение исполнителей

Основные этапы	№ раб	Содержание работ	Должность исполнителя
Постановка задачи	1	Постановка задачи и сроков выполнения	Научный руководитель
Анализ предметной области	2	Аналитический обзор аналогичных конкурентных систем, обзор методов и средств программной архивации, программной выгрузки исходного кода и сборки проектов.	Студент
Проектирование программной системы	3	Выбор методов и средств разработки.	Студент
	4	Проектирование архитектуры системы.	Студент, научный руководитель
Программирование	5	Разработка графического интерфейса системы.	Студент
	6	Реализация обработки заявок на сборку дистрибутивов системы «Химик-аналитик», а именно их сохранение, загрузку, редактирование и заполнение данными по шаблону.	Студент
	7	Реализация программной выгрузки исходного кода системы «Химик-аналитик» из системы контроля версий для создания ее дистрибутива.	Студент
	8	Реализация программной сборки проекта системы, ее дополнительных модулей и вспомогательных утилит.	Студент

	9	Реализация программной архивации в CAB архив и в защищенный RAR архив.	Студент
	10	Разработка и внедрение различных функций, обеспечивающих стабильную работу программной системы.	Студент
	11	Реализация логирования процесса сборки проектов, процесса составления дистрибутива.	Студент
Тестирование	12	Тестирование разработанной системы на наличие ошибок.	Студент, научный руководитель
Исправление ошибок	13	Исправление ошибок, найденных на этапе тестирования, улучшение качества работы системы.	Студент
Анализ и оформление результатов	14	Оценка полученных результатов.	Научный руководитель
	15	Оформление сопровождающей документации.	Студент

ПРИЛОЖЕНИЕ Г

Временные показатели научного исследования

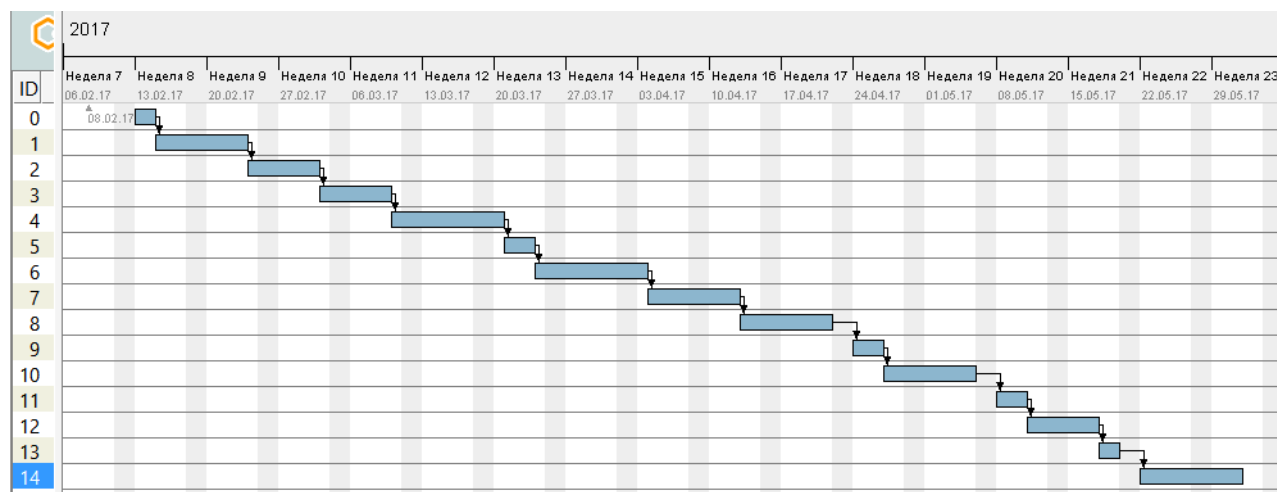
Название работы	Исполнители	Трудоемкость работ, человеко-дни									Длительность работ					
		tmin			tmax			toжi			Тр, рабочие дни			Тк, календарные дни		
		И1	И2	И3	И1	И2	И3	И1	И2	И3	И1	И2	И3	И1	И2	И3
Постановка задачи	1	1	1	1	2	2	2	1,4	1,4	1,4	1,4	1,4	1,4	2	2	2
Анализ предметной области	1	6	6	6	8	8	8	6,8	6,8	6,8	6,8	6,8	6,8	10	10	10
Выбор методов и средств разработки	1	4	4	4	6	6	6	4,8	4,8	4,8	4,8	4,8	4,8	7	7	7
Проектирование архитектуры системы	2	9	9	9	11	11	11	9,8	9,8	9,8	4,9	4,9	4,9	7	7	7
Разработка графического пользовательского интерфейса	1	6	3	6	8	4	8	6,8	3,4	6,8	6,8	3,4	6,8	10	5	10
Реализация обработки заявок на сборку	1	3	5	6	4	6	7	3,4	5,4	6,4	3,4	5,4	6,4	5	8	10
Реализация программной выгрузки исходного кода	1	5	3	3	6	4	4	5,4	3,4	3,4	5,4	3,4	3,4	8	5	5

Реализация программной сборки проектов	1	3	8	2	4	10	4	3,4	8,8	2,8	3,4	8,8	2,8	5	13	4
Реализация программного создания защищенного архива	1	6	5	5	8	6	6	6,8	5,4	5,4	6,8	5,4	5,4	10	8	8
Реализация логирования	1	3	3	3	4	4	4	3,4	3,4	3,4	3,4	3,4	3,4	5	5	5
Реализация конечной версии программного продукта	1	5	6	7	6	7	8	5,4	6,4	7,4	5,4	6,4	7,4	8	10	11
Тестирование	2	3	3	3	4	4	4	3,4	3,4	3,4	3,4	3,4	3,4	5	5	5
Исправление ошибок	1	3	4	4	5	6	6	3,8	4,8	4,8	3,8	4,8	4,8	6	7	7
Оценка полученных результатов	1	1	1	1	2	2	2	1,4	1,4	1,4	1,4	1,4	1,4	2	2	2
Оформление сопровождающей документации	1	7	7	7	10	10	10	8,2	8,2	8,2	8,2	8,2	8,2	12	12	12
Итого								74,2	76,8	76,2	69,3	71,9	71,3	102	106	105

ПРИЛОЖЕНИЕ Д

Календарный план-график проведения работ

ID	Название	Исполнителей	Дата начала	Дата оконча...
0	• Постановка задачи	Студент,Руководитель	13.02.17	14.02.17
1	• Анализ предметной области	Студент	15.02.17	23.02.17
2	• Выбор методов и средств разработки	Студент	24.02.17	02.03.17
3	• Проектирование архитектуры системы	Студент,Руководитель	03.03.17	09.03.17
4	• Разработка графического пользовательского интерфейса	Студент	10.03.17	20.03.17
5	• Реализация обработки заявок на сборку	Студент	21.03.17	23.03.17
6	• Реализация программной выгрузки исходного кода	Студент	24.03.17	03.04.17
7	• Реализация программной сборки проектов	Студент	04.04.17	12.04.17
8	• Реализация программного создания защищенных архивов	Студент	13.04.17	21.04.17
9	• Реализация логирования	Студент	24.04.17	26.04.17
10	• Реализация конечной версии программного продукта	Студент	27.04.17	05.05.17
11	• Тестирование	Студент,Руководитель	08.05.17	10.05.17
12	• Исправление ошибок	Студент	11.05.17	17.05.17
13	• Оценка полученных результатов	Руководитель	18.05.17	19.05.17
14	• Оформление сопровождающей документации	Студент	22.05.17	31.05.17



ПРИЛОЖЕНИЕ Е

Пример «заявки на сборку»

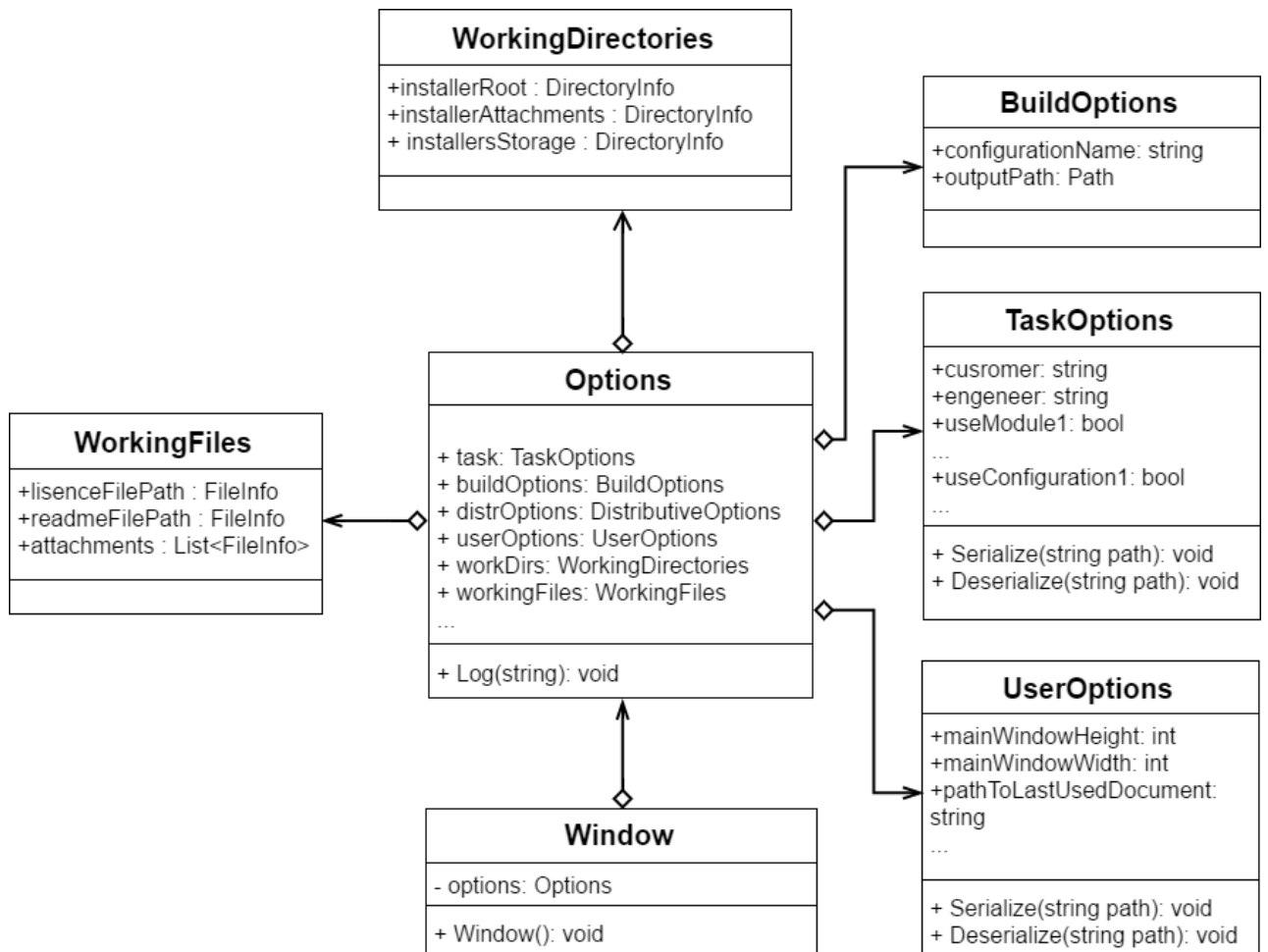
**Ф ЛИТ ИФВТ 2.4.7-01/01. Заявка. Формирование дистрибутива ВЛК\ЛИС\ЛИУС
«Химик-аналитик»**

Дата: **09.11.2016**

№	Наименование показателя	Значение
1.	Заказчик, наименование, принятое внутри лаборатории	ООО «Газпром трансгаз Югорск»
2.	Версия ЛЖ	2.0
3.	Дата (время), к которой будет готова БД для дистрибутива	11.11.2016
4.	Дата (время), к которому необходим дистрибутив	16.11.2016 10:00
5.	Тип СУБД, версия	MS Access
6.	Наименование БД в переключателе	Югорск Общая ВЛК 2.1 (обновленная, финальная)
7.	Состав дистрибутива: обычный, ВЛК, демоверсия или обновление отдельных модулей (патчи)	Обновление (БД и клиента с заменой файлов) Предыдущее обновление 02/02/2016
8.	Назначение дистрибутива (промежуточный, финальный, для обучения, тестовый)	Финальный
9.	Состав ЛИС и дополнительные модули	Обычный, Ежедневник
10.	Лицензия (файл <i>license.txt</i> с текстом)	Z:\Договора_ВСЕ\Газпром трансгаз Югорск\2016 Договор 2410-16-286\09.Бланки для дистрибутива и лицензии\
11.	Точное имя владельца версии и все его варианты. Отдельно указать необходимость наличия города во владельце версии	Полное - ООО "Газпром трансгаз Югорск" Обычное – ООО "Газпром трансгаз Югорск" Краткое - ООО "Газпром трансгаз Югорск" Сжатое - ООО "Газпром трансгаз Югорск" Наличие города – не требуется
12.	Отключение проверки владельца версии на уровне документов: да\нет	нет
13.	Сборка ЛИС с «Е»: да\нет	нет
14.	Запись дистрибутива на CD-диск. Тип CD: CD-R\CD-RW	нет
15.	Публикация дистрибутива на сайте лаборатории www.chemsoft.ru	да
16.	Контактный e-mail, телефон и ФИО Заказчика, которому необходимо выслать уведомление по сайту	***** Наталья Вячеславовна 8(34675) ***** *****@ttg.gazprom.ru *****@ttg.gazprom.ru
17.	Наличие в дистрибутиве доп. информации и вложений	Z:\Договора_ВСЕ\Газпром трансгаз Югорск\2016 Договор 2410-16-286\10.Руководства\
18.	Примечание	-
19.	ФИО ответственного инженера	Афанасьева М.В.
20.	ФИО Тестера дистрибутива	Афанасьева М.В.
21.	ФИО Исполнителя	Сафьянов А.С.

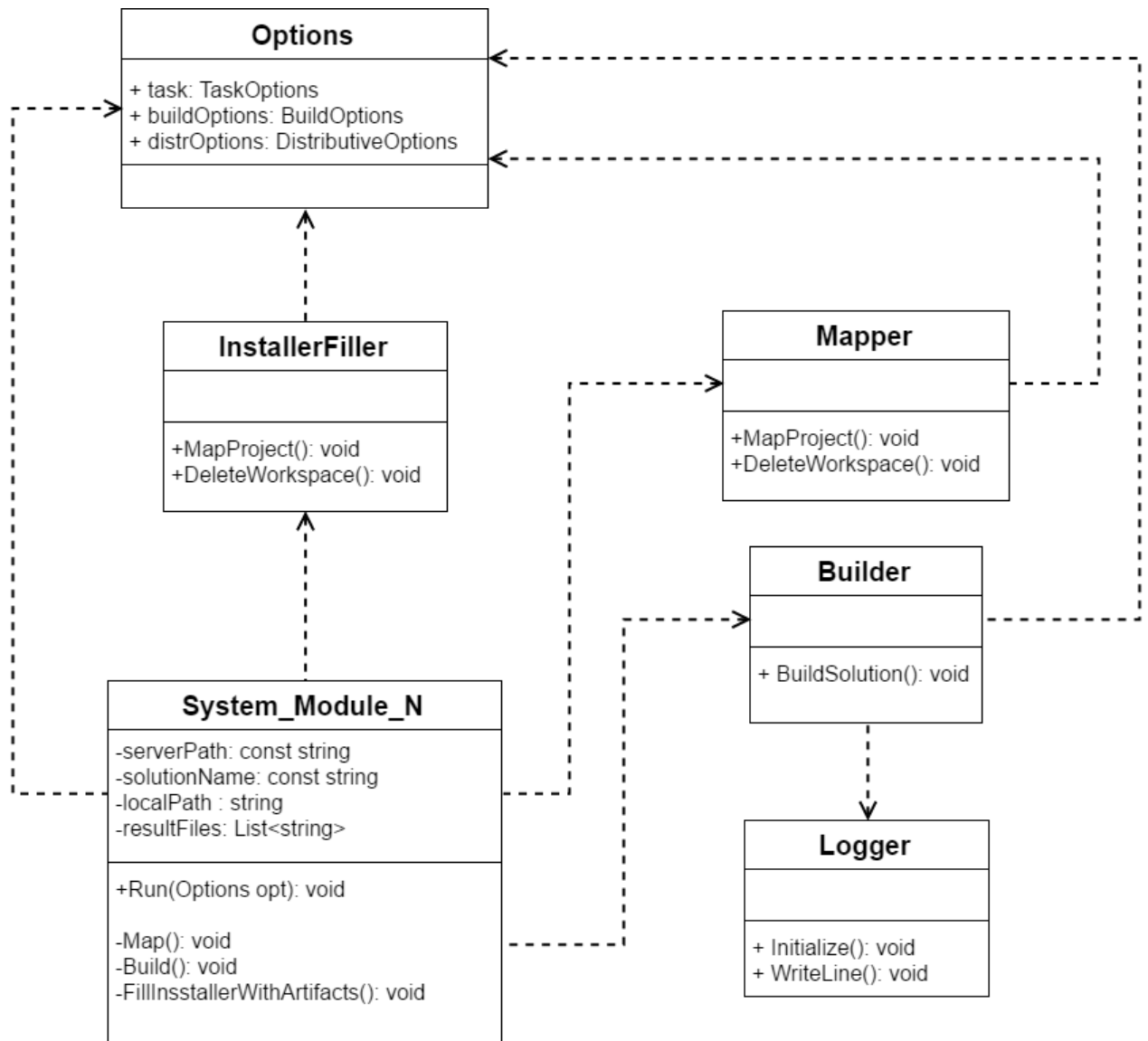
ПРИЛОЖЕНИЕ Ж

UML диаграмма классов



ПРИЛОЖЕНИЕ 3

UML диаграмма классов



ПРИЛОЖЕНИЕ И

UML диаграмма классов

