

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ  
Заведующий кафедрой ЭВМ  
\_\_\_\_\_ Г.И. Радченко  
«\_\_» \_\_\_\_\_ 2020 г.

Модуль формирования отчетов в системе управления производством  
РФЯЦ-ВНИИТФ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

Руководитель работы,  
к.т.н., доцент каф. ЭВМ  
\_\_\_\_\_ В.А. Парасич  
«\_\_» \_\_\_\_\_ 2020 г.

Автор работы,  
студент группы КЭ-405  
\_\_\_\_\_ В.Е. Лукашов  
«\_\_» \_\_\_\_\_ 2020 г.

Нормоконтролёр,  
ст. преп. каф. ЭВМ  
\_\_\_\_\_ С.В. Сяськов  
«\_\_» \_\_\_\_\_ 2020 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ  
Заведующий кафедрой ЭВМ  
\_\_\_\_\_ Г.И. Радченко  
« \_\_\_\_ » \_\_\_\_\_ 2020 г.

### **ЗАДАНИЕ**

**на выпускную квалификационную работу бакалавра**  
студенту группы КЭ-405  
Лукашову Виталию Евгеньевичу  
обучающемуся по направлению  
09.03.01 «Информатика и вычислительная техника»

- 1. Тема работы:** «Модуль формирования отчетов в системе управления производством РФЯЦ-ВНИИТФ» утверждена приказом по университету от 24 апреля 2020 г. №627.
- 2. Срок сдачи студентом законченной работы:** 1 июня 2020 г.
- 3. Исходные данные к работе:**
  - комплекс систем управления производством (CRM, ERP, MES и PLM-системы);
  - операционная система не ниже Windows 7.Обеспечить основной функционал приложения:
  - автоматизированное составление отчетов и сопутствующей документации;
  - интегрирование ПО в имеющуюся систему управления производством.

**4. Перечень подлежащих разработке вопросов:**

- рассмотрение существующих систем автоматизации и управления производством;
- анализ современных программных технологий для организации автоматизации и управления производством;
- выбор инструментов для разработки продукта;
- разработка собственного, вспомогательного программного продукта для автоматизации и управления документооборотом на производстве;
- тестирование разработанного программного обеспечения;
- интеграция интерфейса разработанного ПО в имеющуюся PLM-систему;
- оценка работоспособности разработанного программного обеспечения в реальных условиях работы.

**5. Дата выдачи задания:** 1 декабря 2019 г.

Руководитель работы \_\_\_\_\_ /В.А. Парасич/

Студент \_\_\_\_\_ /В.Е Лукашов /

## КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы	01.03.2020	
Разработка модели, проектирование	01.04.2020	
Реализация системы	01.05.2020	
Тестирование, отладка, эксперименты	15.05.2020	
Компоновка текста работы и сдача на нормоконтроль	24.05.2020	
Подготовка презентации и доклада	30.05.2020	

Руководитель работы \_\_\_\_\_ /В.А. Парасич /

Студент \_\_\_\_\_ /В.Е. Лукашов/

## Аннотация

В.Е. Лукашов. Модуль формирования отчетов в системе управления производством РФЯЦ-ВНИИТФ. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШЭКН; 2020, 100 с., 37 ил., библиогр. список – 15. наим.

В рамках выпускной квалификационной работы производится анализ имеющихся и возможных для внедрения технологий автоматизации и управления производством в рамках предприятия: ФГУП РФЯЦ-ВНИИТФ. Разрабатывается программное обеспечение для автоматизации документооборота на уровне систем PLM и MES с использованием объектно-ориентированного языка программирования C#. Реализуется интеграция интерфейса ПО в имеющиеся системы управления и автоматизации производства. Производится тестирование и анализ результатов работы системы управления и автоматизации производства с новым программным продуктом. После успешных испытаний ПО введено в эксплуатацию. В результате внедрения программного обеспечения, был увеличен общий коэффициент автоматизации производства и решены такие проблемы как: допуск ошибки, связанной с человеческим фактором, при заполнении документации вручную, несвоевременность оформления документов отчетности, а также сокращено время формирования отчетных и сопроводительных документов.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	8
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	9
1.1. АНАЛИЗ ИМЕЮЩЕЙСЯ СИСТЕМЫ И ПОСТАНОВКА ЗАДАЧ .....	11
1.2. АНАЛИЗ ОСНОВНЫХ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ .....	12
1.2.1. Совместимость с имеющимися системами.....	13
1.2.2. Скорость разработки .....	13
1.2.3. Производительность.....	14
1.2.4. Удобство создания графического интерфейса .....	15
1.2.5. Поддержка .....	15
1.2.6. Вывод.....	15
2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ.....	16
2.1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ.....	16
2.2. НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ.....	19
2.2.1. Требования к платформе.....	19
2.2.2. Требования к безопасности .....	19
2.2.3. Требования к пользователю .....	19
3. ПРОЕКТИРОВАНИЕ.....	20
3.1. АРХИТЕКТУРА ПРЕДЛАГАЕМОГО РЕШЕНИЯ .....	21
3.2. ОПИСАНИЕ ДАННЫХ.....	23
3.2.1. Данные блока «Серверный менеджер» .....	23
3.2.2. Данные блока «Получение отчета MES» .....	24
3.2.3. Данные блока «Получение документа КЗО».....	25

3.2.4.Данные блока «Получение отчета ТЯИ» .....	28
3.2.5.Данные блока «Получение отчета ПИП».....	29
4.РЕАЛИЗАЦИЯ.....	32
4.1.РЕАЛИЗАЦИЯ ИНТЕГРАЦИИ МОДУЛЯ В СИСТЕМУ PLM.....	32
4.2.РЕАЛИЗАЦИЯ БЛОКА «ФОРМИРОВАНИЯ ОТЧЕТА КЗО».....	37
4.3.РЕАЛИЗАЦИЯ БЛОКА «ФОРМИРОВАНИЕ ДОКУМЕНТА ТЯИ» .....	41
4.4.РЕАЛИЗАЦИЯ БЛОКА «ФОРМИРОВАНИЕ ДОКУМЕНТА ПИП» .....	45
4.5.РЕАЛИЗАЦИЯ БЛОКА «СЕРВЕРНЫЙ МЕНЕДЖЕР».....	49
4.6.РЕАЛИЗАЦИЯ БЛОКА «ПОЛУЧЕНИЕ ОТЧЕТА MES» .....	53
5.ТЕСТИРОВАНИЕ .....	56
5.1.МЕТОДОЛОГИЯ ТЕСТИРОВАНИЯ.....	56
5.2.ПРОВЕДЕНИЕ ПРОЦЕДУРЫ ТЕСТИРОВАНИЯ.....	56
ЗАКЛЮЧЕНИЕ.....	65
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	67
ПРИЛОЖЕНИЕ А ИСХОДНЫЙ КОД Plugin1 .....	69
ПРИЛОЖЕНИЕ Б ИСХОДНЫЙ КОД KZOcard .....	73
ПРИЛОЖЕНИЕ В ИСХОДНЫЙ КОД LPcard .....	77
ПРИЛОЖЕНИЕ Г ИСХОДНЫЙ КОД TPcard.....	80
ПРИЛОЖЕНИЕ Д ИСХОДНЫЙ КОД PLMServer .....	84
ПРИЛОЖЕНИЕ Е ИСХОДНЫЙ КОД MEScard .....	99

## ВВЕДЕНИЕ

Для эффективного управления процессами предприятия все чаще используются специализированные решения, такие как: CRM, ERP, MES, PLM-системы, позволяющие автоматизировать деятельность предприятия на всех уровнях, установить контроль над процессами, вести оперативный учет на основе показателей, делегировать задачи, управлять ресурсами. Потребность в подобных системах с каждым годом растет, поскольку удобный и эффективный инструмент управления позволяет оценивать текущую ситуацию и наиболее оперативно реагировать на все изменения, как на предприятии, так и в мире в целом. [1].

Автоматизация процессов на предприятии позволяет значительно сократить временные издержки, предотвратить ошибки, связанные с человеческим фактором, легко контролировать ход выполнения производственных процессов.

На данный момент на рынке существует множество готовых систем и комплексов автоматизации и управления производством. Однако невозможно сделать универсальный продукт, удовлетворяющий требованиям сразу всех предприятий. В связи с этим, при установке того или иного готового программного продукта, на предприятии рано или поздно возникает необходимость в его доработке, усовершенствовании имеющихся возможностей или введения дополнительных модулей с новыми техническими решениями.

Целью представленной выпускной квалификационной работы является разработка модуля системы автоматизированного управления производством ФГУП РFYЦ-ВНИИТФ, отвечающего за автоматизированное формирование отчетной и сопроводительной документации на уровне PLM-системы и передачу информации в MES-систему. Так как существующий функционал не полностью удовлетворял требованиям заказчика, возникла необходимость



создания дополнительного модуля, расширяющего возможности существующей системы управления производством.

Актуальность автоматизированной системы отчетности и документации связана в первую очередь с возможностью возникновения ошибок вследствие человеческого фактора в процессе составления отчетности вручную, что приводит к неправильному получению информации руководством, некорректно составленным планам предприятия, к потерям во времени и к непредвиденным финансовым издержкам. В связи с этим большинство предприятий пытается автоматизировать документооборот на всех уровнях производства.

## 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Высокая конкуренция на рынке производственных предприятий заставляет ведущих менеджеров искать возможности сокращения издержек и усовершенствования производства. Одним из применяемых инструментов для этого является автоматизация процессов на предприятии.

Первые попытки автоматизации на производстве были предприняты еще в девятнадцатом веке и за более чем двести лет развития превратились в сложные системы, способные не только заменить человека в некоторых операциях, но и выполнить техническое задание качественней и быстрее. В наше время автоматизированная система управления производством представляет собой иерархию контрольно-информационных систем, регулирующих производственные процессы всех уровней.

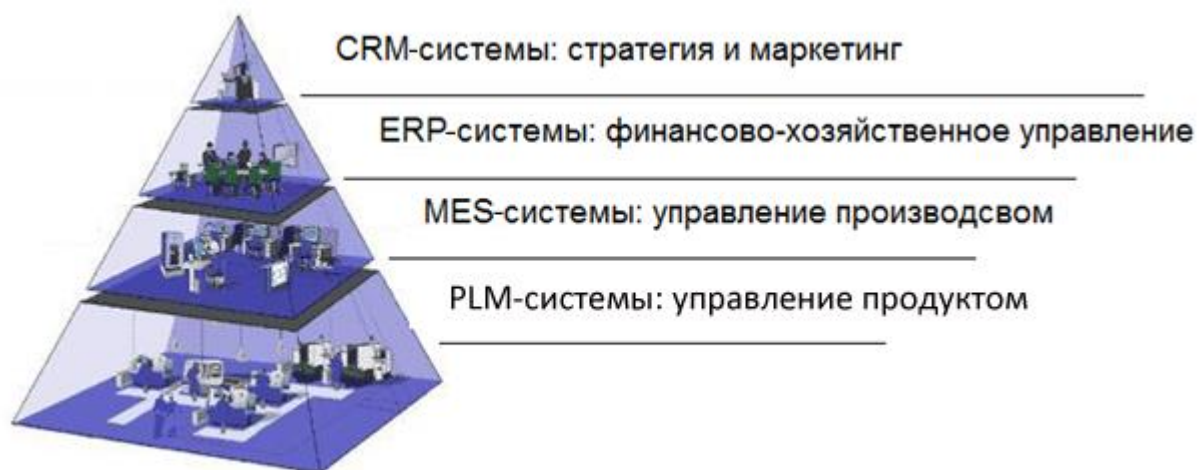


Рисунок 1 – Схема иерархии систем автоматизации

CRM-система (Customer Relationship Management, Система управления взаимоотношениями с клиентами) — прикладное программное обеспечение для организаций, предназначенное для автоматизации стратегий взаимодействия с заказчиками (клиентами), в частности, для повышения уровня продаж, оптимизации маркетинга и улучшения обслуживания клиентов путём сохранения информации о клиентах и истории взаимоотношений с ними, установления и улучшения бизнес-процессов и последующего анализа результатов.

ERP (Enterprise Resource Planning, планирование ресурсов предприятия) — организационная стратегия интеграции производства и операций, управления трудовыми ресурсами, финансового менеджмента и управления активами, ориентированная на непрерывную балансировку и оптимизацию ресурсов предприятия посредством специализированного интегрированного пакета прикладного программного обеспечения, обеспечивающего общую модель данных и процессов для всех сфер деятельности. ERP-система — программный пакет, реализующий стратегию ERP.

MES-система (Manufacturing Execution System, производственная исполнительная система) — специализированное прикладное программное обеспечение, предназначенное для решения задач синхронизации, координации, анализа и оптимизации выпуска продукции в рамках какого-либо производства. С 2004 года термин расшифровывается как Manufacturing Enterprise Solutions — корпоративные системы управления производством. MES-системы относятся к классу систем управления уровня цеха[2].

PLM-система (Product Lifecycle Management, управление жизненным циклом изделия) — организационно-техническая система, обеспечивающая управление всей информацией об изделии (продукте) и связанных с ним процессах на протяжении всего его жизненного цикла, начиная с проектирования и производства до снятия с эксплуатации. При этом в качестве изделий могут рассматриваться различные сложные технические объекты, например, самолеты. Информация об объекте, содержащаяся в PLM-системе, является цифровым макетом этого объекта [3].

С помощью перечисленных четырех систем на предприятии возможно организовать автоматизированное управление на всех уровнях производства от склада с заготовками изделий до высшего менеджмента компании.

### 1.1. АНАЛИЗ ИМЕЮЩЕЙСЯ СИСТЕМЫ И ПОСТАНОВКА ЗАДАЧ

На предприятии ФГУП РФЯЦ-ВНИИТФ имеется полный цикл систем автоматизации производства. Существуют системы класса: CRM, ERP, MES и PLM. Все системы функционируют в штатном режиме, и позволяют вести автоматизированное управление производством. Однако, согласно стратегии развития предприятия, руководством было принято решение об увеличении

коэффициента автоматизации документооборота на уровне системы PLM и реализации возможности передачи технологических данных из системы PLM в MES-систему.

Цель данной выпускной квалификационной работы является разработка модуля автоматизации формирования отчетного и сопроводительного документооборота на уровнях PLM и MES-систем или уровне цеха предприятия. Данное программное обеспечение направлено на автоматизацию составления отчетов и документов, формируемых в настоящее время в ручном режиме операторами систем PLM и MES. Для достижения поставленной цели необходимо решить следующие задачи:

- выбрать инструменты для разработки программного продукта;
- разработать структуру программного продукта;
- интегрировать ПО в имеющиеся системы;
- провести тестирование и внедрить ПО на предприятии.

## 1.2. АНАЛИЗ ОСНОВНЫХ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ

Перед началом разработки необходимо определиться, на каком языке будет написано программное обеспечение. Данный выбор является важным вопросом, так как от него будет зависеть, насколько качественно будет работать программа, и насколько быстрым и комфортным будет создание продукта.

Исходя из задач, необходимо создать программный продукт, способный взаимодействовать с имеющимися системами MES и PLM, а также с API Microsoft Office. Приложение должно работать на операционной системе Windows.

По данным критериям подходят три популярных языка программирования для разработки приложений:

1. C#;
2. Python;
3. Java.

Проведем анализ выбранных языков согласно интересующим критериям:

1. Совместимость с имеющимися системами.
2. Скорость разработки.
3. Производительность.
4. Удобство создания графического интерфейса.
5. Поддержка.

#### 1.2.1. Совместимость с имеющимися системами

Большинство современных систем автоматизаций и управления производством уровня CRM, ERP, MES и PLM разрабатываются на языках программирования C# и Java. Следовательно, при выборе этих двух языков, интеграция разработанного приложения будет проще, в то время как при использовании языка Python необходимо разрабатывать дополнительный модуль интерпретации. В нашем случае имеющиеся системы управления были реализованы на языке C#.

#### 1.2.2. Скорость разработки

В данном критерии Python, в отличие от языков C# и Java, имеет лаконичный и минималистичный синтаксис, что делает разработку простой и быстрой. Во многом это простота обусловлена динамической типизацией

данных, однако данный вид типизации может привести к ухудшению других параметров программы. Языки Java и C# имеют статическую типизацию объектов, что обеспечивает безопасность типов, при которой все потенциальные ошибки устанавливаются во время компиляции. Из-за динамической типизации данных в языке Python ошибки появляются в процессе выполнения программы. Таким образом, вероятность появления ошибок в готовых программах на языках Java и C# минимальна. В конечном итоге, все это упрощает разработку и управление большими приложениями. Ошибки во время выполнения программы сложнее идентифицировать и исправлять, чем ошибки во время компиляции.

Python позволяет в короткие сроки реализовать код, охватывающий большой круг задач. C# и Java являются более сложными в изучении и применении языками, однако эта сложность оправдана другими, в некоторых случаях, более важными критериями. Так как разработка модуля не была ограничена строгими временными рамками, но требовала точности и безошибочности в работе, предпочтительней было использование языков C# и Java.

### 1.2.3. Производительность

Все три языка имеют достаточно высокую производительность на сегодняшний день, они могут запускаться на Windows, однако, язык C# имеет более глубокую интеграцию в операционную систему Windows. Платформа .Net активно развивается, что хорошо отображается на производительности и оптимизации компиляции программного кода на языке C#.

#### 1.2.4. Удобство создания графического интерфейса

В данном критерии, безусловно, побеждает C# с Windows формами, которые позволяют очень быстро, красиво и гибко настроить и запрограммировать форму под нужды конкретного проекта. Конечно, GUI десктопа можно писать на Java и Python, но это будет более трудозатратно по времени и усложнит разработку модуля.

#### 1.2.5. Поддержка

Безусловно, в данном проекте очень важна поддержка языка и регулярные обновления. Так как C# разработан и работает в экосистеме Windows, то у него имеется очень большая поддержка, как со стороны сообщества, так и со стороны экспертов самой компании Microsoft. Java также имеет очень большое сообщество, однако, если мы имеем дело с Windows, то у C# здесь имеется преимущество. Python - это язык с открытым кодом, его развитие идет только со стороны сообщества, которое не может гарантировать тот же уровень поддержки.

#### 1.2.6. Вывод

Из анализа соответствия обозначенным выше критериям рассмотренных языков программирования можно сделать вывод, что для выполнения данной работы (с учетом всех условий, требований, временных ограничений, предыдущего опыта исполнителя и требуемой поддержки) наиболее оптимальным является язык программирования C#.

## 2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ

### 2.1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

Для определения функциональных требований к системе построим диаграмму прецедентов. Она отображает события, возникающие в системе с точки зрения пользователя. Диаграмма прецедентов в нотации UML изображена на рисунке 2.

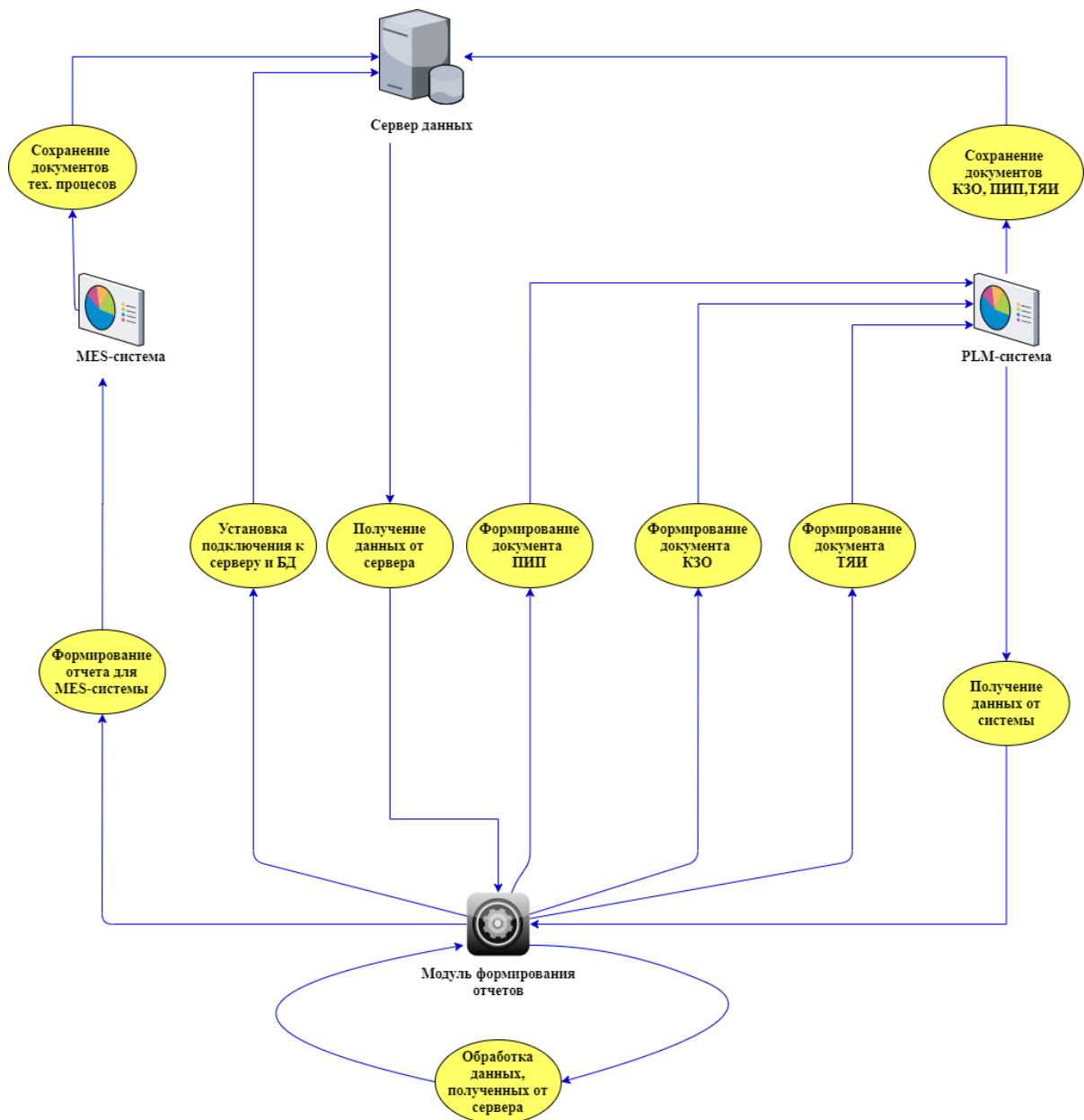


Рисунок 2 – Диаграмма вариантов использования модуля



Согласно диаграмме, программное обеспечение должно выполнять 8 основных функций:

- получение данных от системы PLM;
- подключение к серверу и БД;
- получение данных от сервера;
- обработка данных, полученных от сервера;
- формирование документа Контрольного листа Заявки Оснастки (КЗО);
- формирование документа Протокола Испытаний Продукции (ПИП);
- формирование документа Транспортного Ярлыка Изделия (ТЯИ);
- формирование отчета для MES-системы.

Исходя из перечня основных функций, для реализации программного продукта необходим набор блоков, представленных в таблице 1.

Таблица 1 – Программные блоки модуля

<b>Название блока</b>	<b>Описание блока</b>
Блок интеграции модуля в PLM-систему	С помощью данного блока обеспечивается управление модулем формирования отчетов из системы управления производством.
Блок «ФОРМИРОВАНИЯ ОТЧЕТА КЗО»	Документ Контрольного листа Заявки Оснастки (КЗО) предназначен для выполнения заказа оснастки, необходимой для изготовления изделия. Данный блок обеспечивает вывод информации о необходимой оснастке, полученной из системы PLM, в заготовку документа, преобразование её в готовый файл формата PDF и экспортирование готового файла в систему управления.

Продолжение таблицы 1

Название блока	Описание блока
<p>Блок «ФОРМИРОВАНИЕ ТЯИ»</p>	<p>Транспортный Ярлык Изделия (ТЯИ) предназначен для маркировки детали перед отправкой на склад хранения. Данный блок обеспечивает вывод информации об изделии, полученной из системы PLM, в заготовку документа, преобразование её в готовый файл формата PDF и экспортирование готового файла в систему управления.</p>
<p>Блок «ФОРМИРОВАНИЕ ОТЧЕТА ПИП»</p>	<p>Протокол Испытаний Продукции (ПИП) является обязательным документом для каждого продукта и содержит в себе информацию обо всех испытаниях конкретного продукта. Данный блок обеспечивает вывод информации о прохождении испытаний изделием, полученной из системы PLM, в заготовку документа, преобразование её в готовый файл формата PDF и экспортирование готового файла в систему управления.</p>
<p>Блок «СЕРВЕРНЫЙ МЕНЕДЖЕР»</p>	<p>Блок обеспечивает возможность подключения модуля к серверу и базам данных PLM-системы для получения информации напрямую из файлов базы данных системы. Данный блок необходим для получения информации, которую невозможно получить от PLM-системы. Полученная в виде XML-файлов информация от сервера обрабатывается в блоке и передается другим блокам в качестве переменных типа string, int32, bool и других.</p>

Окончание таблицы 1.

<b>Название блока</b>	<b>Описание блока</b>
Блок «ПОЛУЧЕНИЕ ОТЧЕТА MES»	Отчет для системы MES представляет собой описание структуры продукта с указанием стадии изготовления. Данный блок обеспечивает вывод информации о стадиях изготовления частей продукта и продукта в целом, полученной из блока «СЕРВЕРНЫЙ МЕНЕДЖЕР», в заготовку документа, преобразование её в готовый файл формата PDF и экспортирование готового файла на диск MES-системы.

## 2.2. НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

### 2.2.1. Требования к платформе

Программное обеспечение должно работать на персональном компьютере с операционной системой Windows не ниже версии 7.

### 2.2.2. Требования к безопасности

Система безопасности реализована на уровне доступа пользователя к компьютеру и соответствует всем требованиям предприятия. Система безопасности не нуждается в доработке в рамках программного продукта.

### 2.2.3. Требования к пользователю

Для работы с системой пользователь должен иметь базовые навыки работы с компьютером, а так же уметь пользоваться имеющейся системой управления производством.

### 3. ПРОЕКТИРОВАНИЕ

Перед проектированием модуля необходимо рассмотреть функциональную составляющую части системы управления производством, которую затронет данный модуль. Разрабатываемый модуль будет взаимодействовать с MES и PLM-системами, являющимися частями системы управления на уровне производственного цеха.

На рисунках 3 и 4 представлена функциональная схема до внедрения модуля и после внедрения модуля соответственно.



Рисунок 3 – Функциональная схема части СУ до внедрения модуля

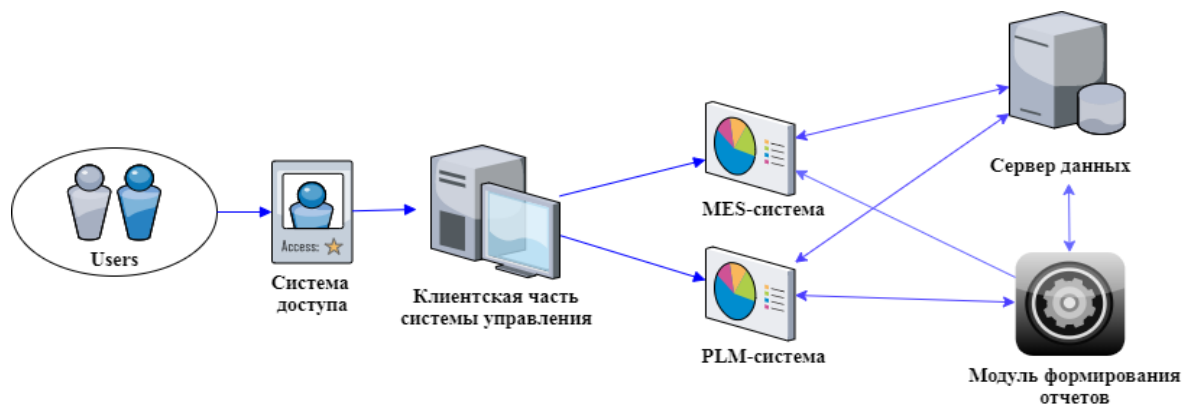


Рисунок 4 – Функциональная схема части СУ после внедрения модуля

Из рисунка 3 видно, что связи между системами PLM и MES не существует, таким образом, исключается получение информации одной системы из другой. К тому же, при эксплуатации существующей системы управления производством, были обнаружены некоторые недоработки в части документооборота системы PLM. Все недостатки системы можно исключить, введя в систему управления производством дополнительный модуль формирования отчетности.

### 3.1. АРХИТЕКТУРА ПРЕДЛАГАЕМОГО РЕШЕНИЯ

Модуль формирования отчетов можно разбить на шесть подпрограмм или блоков модуля, реализующих основные функции программного обеспечения.

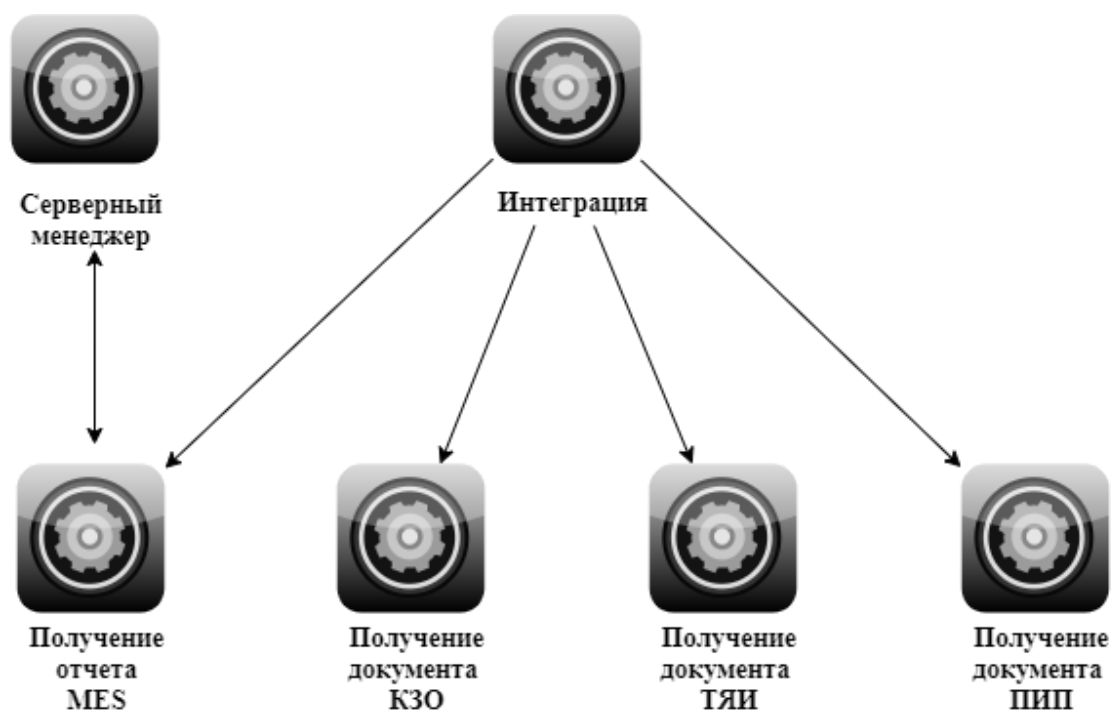


Рисунок 5 – Схема элементов модуля

1. «Интеграция» – блок интеграции интерфейса модуля в PLM-систему.

Интерфейс представляет собой четыре функциональные кнопки в меню системы:

– «Документ КЗО»;

– «Документ ТЯИ»;

– «Документ ПИП»;

– «Документ MES».

С помощью данных кнопок происходит запуск блоков получения соответствующих документов.

2. «Серверный менеджер» – данный блок обеспечивает подключение модуля к серверу и базам данных системы PLM. Кроме функций подключения сервер так же получает информацию из базы данных, обрабатывает ее и представляет в виде переменных для передачи другим блокам модуля.

3. «Получение отчета MES» – блок получает необходимую для формирования отчета информацию, используя блок «Серверный менеджер», автоматизировано составляет документ отчетности и экспортирует его на диск системы MES. Формирование и экспорт документа осуществляется путем обращения к API Microsoft Office.

4. «Получение документа КЗО» – блок получает необходимую информацию об оснастке для изготовления изделия из системы управления производством, используя обращение к API PLM-системы. После получения данных и представления их в виде переменных блока, данные заполняются в шаблон документа и экспортируются в PLM-систему путем обращения к API Microsoft Office.

5. «Получение документа ТЯИ» – блок получает необходимую информацию об изделии из системы управления производством, используя обращение к API PLM-системы. После получения данных и представления их в виде

переменных блока, данные заполняются в шаблон документа и экспортируются в PLM-систему путем обращения к API Microsoft Office.

- б. «Получение документа ПИП» – блок получает необходимую информацию об испытаниях изделия из системы управления производством, используя обращение к API PLM-системы. После получения данных и представления их в виде переменных блока, данные заполняются в шаблон документа и экспортируются в PLM-систему путем обращения к API Microsoft Office.

## 3.2. ОПИСАНИЕ ДАННЫХ

### 3.2.1. Данные блока «Серверный менеджер»

Блок «Серверный менеджер» получает информацию от сервера в виде XML-документа. Пример документа представлен на рисунке 6.

```
<?xml version="1.0" encoding="UTF-16" ?>
<ROOT>
  <fieldset>
    <field Id="c0" Name="_ID_LINK" DataType="int" />
    <field Id="c1" Name="_ID_VERSION" DataType="int" />
    <field Id="c2" Name="_ID_TYPE" DataType="int" />
    <field Id="c3" Name="_PRODUCT" DataType="string" />
    <field Id="c4" Name="_VERSION" DataType="string" />
    <field Id="c5" Name="_ID_STATE" DataType="int" />
    <field Id="c6" Name="_DOCUMENT" DataType="int" />
    <field Id="c7" Name="_MIN_QUANTITY" DataType="float" />
    <field Id="c8" Name="_MAX_QUANTITY" DataType="float" />
    <field Id="c9" Name="_ACCESSLEVEL" DataType="int" />
    <field Id="c10" Name="_LABEL" DataType="int" />
    <field Id="c11" Name="_LABEL_NAME" DataType="string" />
    <field Id="c12" Name="_ID_LOCK" DataType="int" />
    <field Id="c13" Name="inId" DataType="int" />
    <field Id="c14" Name="inIdVersion" DataType="int" />
    <field Id="c15" Name="inId_1" DataType="int" />
    <field Id="c16" Name="inId_2" DataType="int" />
    <field Id="c17" Name="inId_3" DataType="int" />
    <field Id="c18" Name="inIdVersion_1" DataType="int" />
    <field Id="c19" Name="inIdUser" DataType="int" />
  </fieldset>
  <rowset>
    <row c0="105892" c1="00051" c2="60" c3="A5B.13.3.1" c4="1" c5="2" c6="0" c7="1" c8="1" c9="3" c10="-1" c11="" c12="" />
    <row c0="105912" c1="00054" c2="60" c3="A5B.13.3.2" c4="1" c5="2" c6="0" c7="1" c8="1" c9="3" c10="-1" c11="" c12="" />
    <row c0="105914" c1="00055" c2="60" c3="A5B.13.3.3" c4="1" c5="2" c6="0" c7="2" c8="2" c9="3" c10="-1" c11="" c12="" />
    <row c0="105916" c1="00056" c2="60" c3="A5B.13.3.4" c4="1" c5="2" c6="0" c7="2" c8="2" c9="3" c10="-1" c11="" c12="" />
    <row c0="105918" c1="00057" c2="60" c3="A5B.13.3.5" c4="1" c5="2" c6="0" c7="2" c8="2" c9="3" c10="-1" c11="" c12="" />
    <row c0="105926" c1="00061" c2="60" c3="A5B.13.3.7" c4="1" c5="2" c6="0" c7="1" c8="1" c9="3" c10="-1" c11="" c12="" />
    <row c0="105927" c1="00062" c2="60" c3="A5B.13.3.8" c4="1" c5="2" c6="0" c7="1" c8="1" c9="3" c10="-1" c11="" c12="" />
    <row c0="105938" c1="00063" c2="60" c3="A5B.13.3.6" c4="1" c5="2" c6="0" c7="2" c8="2" c9="3" c10="-1" c11="" c12="" />
  </rowset>
</ROOT>
```

Рисунок 6 – Пример XML-документа, полученного от сервера

Документ содержит в себе описание всех параметров изделия, хранящихся в виде пары (ключ, значение). Документ является полным описанием всех характеристик изделия, процессов, связанных с изделием, и всех его связей с другими изделиями и их процессами.

В начале каждого документа задаются все параметры изделия, хранящиеся в данном документе. После объявления параметров и их типа данных, идет описание изделия и связанных с ним объектов или процессов.

После получения, документ обрабатывается блоком. Необходимая информация сохраняется в виде переменных для дальнейшей передачи другим блокам модуля.

### 3.2.2. Данные блока «Получение отчета MES»

Все данные поступают в блок «Получение отчета MES» от блока «Серверный менеджер» в виде переменных различных типов.

После получения данных, происходит формирование отчета внутренними функциями блока. Сформированные данные в виде текстового набора строк передаются в шаблон документа.

Сформированный документ содержит в себе информацию о составляющих продукта и стадии их производства. Описание полей документа приведено в таблице 2.

Таблица 2 – Поля отчета MES

<b>Название поля</b>	<b>Описание</b>
Наименование	Наименование изделия по стандартам производства
ID	Внутренний идентификатор изделия в PLM-системе
Уровень	Уровень изделия в иерархии системы PLM



## Окончание таблицы 2

Название поля	Описание
Готовность	Стадия производства изделия (Готово, Изготавливается, Не готово)

Выходными данными модуля получения MES является готовый PDF документ, пример сформированного документа представлен на рисунке 7.

```
-- ID: АБВГ111.11110.0 Наименование: Изделие
  Уровень: 1 Готовность: Изготавливается

-- -- ID: АБВГ111.11111.0 Наименование: Деталь 1
  Уровень: 2 Готовность: Не готово

-- -- ID: АБВГ111.11112.0 Наименование: Деталь 2
  Уровень: 2 Готовность: Готово

-- -- -- ID: АБВГ111.11112.1 Наименование: Деталь 2.1
  Уровень: 3 Готовность: Готово

-- -- -- ID: АБВГ111.11112.2 Наименование: Деталь 2.2
  Уровень: 3 Готовность: Готово

-- -- ID: АБВГ111.11113.0 Наименование: Деталь 3
  Уровень: 2 Готовность: Изготавливается
```

Рисунок 7 – Пример Отчета MES

### 3.2.3. Данные блока «Получение документа КЗО»

Блок «Получение документа КЗО» получает все данные из системы управления путем обращения к API PLM-системы. Данные поступают в блок в виде текста и сохраняются в переменных блока.

Формирование документа происходит с помощью переноса готовых текстовых переменных блока в поля шаблона документа. Описание полей документа представлено в таблице 3.

Таблица 3 – Поля документа КЗО

<b>Название поля</b>	<b>Описание</b>
№ наряда	Номер наряда согласно отделу составителю
Заказ №	Номер заказа (зарезервирован в системе)
Для цеха	Цех заказчик
Обознач. узла	Идентификатор узла детали (в случае отсутствия, присваивается значение 0)
Наименование детали	Наименование детали
Обознач. оснастки	Идентификатор листа оснастки
Обознач. детали	Идентификатор изготавливаемой детали
К-во изготовл. оснастки	Количество комплектов оснастки
Составил	Фамилия составителя заказа
Трудоемкость	Трудоемкость изготовления детали
Содержание заказа оснастки	Содержание листа оснастки, необходимого для изготовления изделия

Сформированный документ содержит в себе информацию о необходимых средствах оснастки для изготовления продукта.

Выходными данными модуля получения КЗО является готовый PDF документ, пример сформированного документа представлен на рисунке 8.

№ наряда 0001	Заказ № 001		Для цеха 222		
Обознач. узла	Трудоемкость		Тема №		
<i>АБВ0.00000.00</i>	1000		Карта №		
Обознач. детали	Наименование детали		Обознач. оснастки		
<i>АБВ0.00000.01</i>	<i>Компрессор</i>		<i>111.222.3</i>		
Кооперация	К-во изготовл. оснастки				
	2				
Содержание заказа					
- Набор СК40 с латунными подкладками (3шт.)					
- 2002-Т(1шт.)					
-Заготовка КВ010103 (1шт.)					
	Фамилия	Подпись	Дата	Срок выд. КД	
Составил	Лукашов			Срок изгот., оснастки	
Проверил				111/222	
Утвердил				Нач. БИХ	
Констр				Накладная №	
Рук. гр.				Дата выполн. заказа	

Рисунок 8 – Пример документа КЗО

### 3.2.4. Данные блока «Получение документа ТЯИ»

Блок «Получение документа ТЯИ» получает все данные из системы управления путем обращения к API PLM-системы. Данные поступают в блок в виде текста и сохраняются в переменных блока.

Формирование документа происходит с помощью переноса готовых текстовых переменных блока в поля шаблона документа. Описание полей документа представлено в таблице 4.

Таблица 4 – Поля документа ТЯИ

<b>Название поля</b>	<b>Описание</b>
Цех изготовитель	Номер цеха изготовителя изделия
Узел	Идентификатор узла детали (в случае отсутствия, присваивается значение 0)
Деталь	Идентификатор детали
Количество деталей	Количество составляющих детали
Количество сопроводительной документации	Количество документации к детали
Масса	Масса детали
Технолог	Фамилия технолога, разработавшего технологию изготовления детали
Инженер	Фамилия инженера-конструктора

Сформированный документ содержит в себе краткую информацию об изделии, необходимую для идентификации изделия на складе хранения.

Выходными данными модуля получения ТЯИ является готовый PDF документ, пример сформированного документа представлен на рисунке 9.

<b><i>ТРАНСПОРТНЫЙ ЯРЛЫК ИЗДЕЛИЯ</i></b>	
<i>Деталь</i>	<i>АБВ.0000.000.1</i>
<i>Узел</i>	<i>АБВ.0000.000.0</i>
<i>Кол-во деталей</i>	<i>8</i>
<i>Кол-во сопр. док.</i>	<i>4</i>
<i>Масса(кг)</i>	<i>1224</i>
<i>Цех изготовитель</i>	<i>222</i>
<i>Технолог</i>	<i>Лукашов</i>
<i>Инженер</i>	<i>Лукашов</i>

Рисунок 9 – Пример документа ТЯИ

### 3.2.5. Данные блока «Получение документа ПИП»

Блок «Получение документа ПИП» получает все данные из системы управления путем обращения к API PLM-системы. Данные поступают в блок в виде текста и сохраняются в переменных блока.

Формирование документа происходит с помощью переноса готовых текстовых переменных блока в поля шаблона документа. Описание полей документа представлено в таблице 5.

Таблица 5 – Поля документа ПИП

Название поля	Описание
Статические испытания	Показывают этап прохождения испытаний (Пройдены, Не пройдены, Планируются)
Вибрационные испытания	Показывают этап прохождения испытаний (Пройдены, Не пройдены, Планируются)
Термические испытания	Показывают этап прохождения испытаний (Пройдены, Не пройдены, Планируются)
Аэродинамические испытания	Показывают этап прохождения испытаний (Пройдены, Не пройдены, Планируются)
Баллистические испытания	Показывают этап прохождения испытаний (Пройдены, Не пройдены, Планируются)
Деталь	Идентификатор детали
Узел	Идентификатор узла детали (в случае отсутствия, присваивается значение 0)
Ответственный 1-5	Фамилии ответственных за испытания лиц
Руководитель испытаний	ФИО руководителя всех испытаний

Сформированный документ содержит в себе краткую информацию об изделии, необходимую для идентификации изделия на складе хранения.

Выходными данными модуля получения ПИП является готовый PDF документ, пример сформированного документа представлен на рисунке 10.

<b>ПРОТОКОЛ ИСПЫТАНИЙ ПРОДУКЦИИ</b>			
<b>Деталь</b>	<i>АБВ.0000.000.1</i>	<b>Руководитель испытаний</b>	<b>Подпись</b>
Узел	<i>АБВ.0000.000.0</i>	<i>Лукашов Виталий Евгеньевич</i>	
<b>Вид испытаний</b>	<b>Этап испытаний</b>	<b>Ответственный</b>	
<i>Статические испытания</i>	<i>Пройдены</i>	<i>Лукашов</i>	
<i>Вибрационные испытания</i>	<i>Не пройдены</i>	<i>Лукашов</i>	
<i>Гермические испытания</i>	<i>Планируются</i>	<i>Лукашов</i>	
<i>Аэродинамические испытания</i>	<i>Пройдены</i>	<i>Лукашов</i>	
<i>Баллистические испытания</i>	<i>Пройдены</i>	<i>Лукашов</i>	
М.П.			

Рисунок 10 – Шаблон документа ПИП

## 4. РЕАЛИЗАЦИЯ

### 4.1. РЕАЛИЗАЦИЯ ИНТЕГРАЦИИ МОДУЛЯ В СИСТЕМУ PLM

Согласно требованиям к программному обеспечению интерфейс модуля должен быть интегрирован в систему управления производством на уровне PLM-системы. Общий вид интерфейса системы PLM и интерфейс стандартного набора функциональных кнопок представлены на рисунках 11 и 12 соответственно.

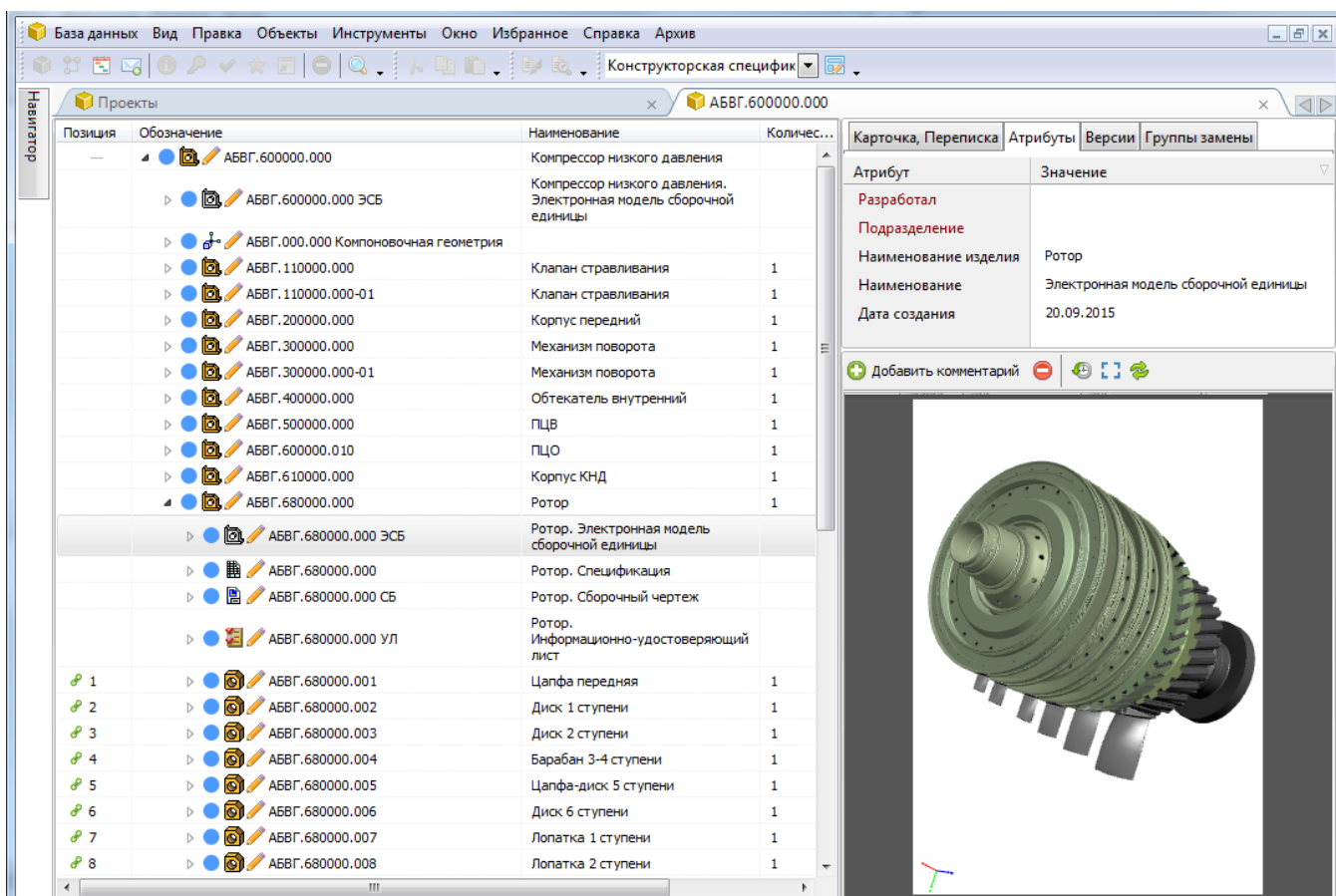


Рисунок 11 – Общий интерфейс системы PLM



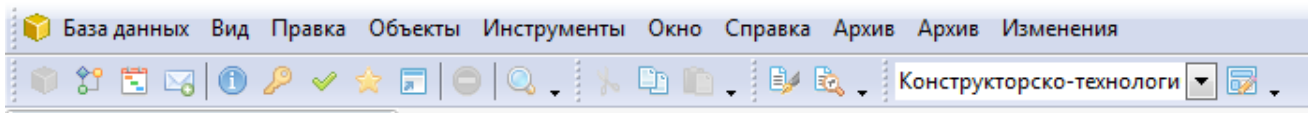


Рисунок 12 – Интерфейс стандартного набора функциональных кнопок PLM-системы

После внедрения кнопок интерфейса управления модулем формирования отчетов в интерфейс функциональных кнопок PLM-системы стандартный набор кнопок расширился, новые функциональные кнопки указаны на рисунке 13.

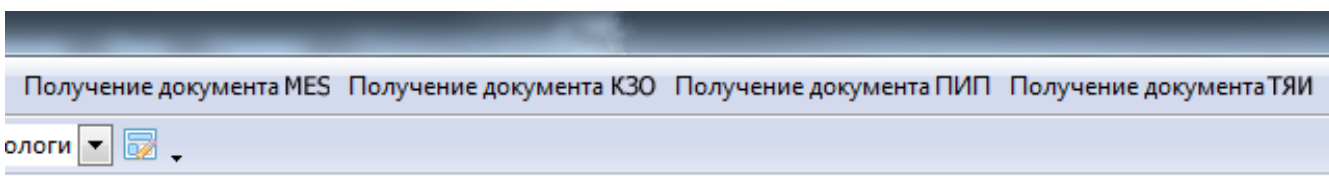


Рисунок 13 – Интерфейс доработанного набора функциональных кнопок PLM-системы

Реализация интеграции интерфейса модуля, в систему PLM осуществлена с помощью класса Plugin1, представленного в листинге А.1 приложения А.

Класс Plugin1 содержит в себе 8 методов. Описание назначения методов указано в таблице 6.

Таблица 6 – Методы класса Plugin1

Название метода	Назначение
InitUserDLLCom	Сборка меню функциональных кнопок
setmenu	Описания пункта меню
PgiCheckMenuItemCom	Проверка включенности кнопки
Mes_card	Получение отчета MES системы

Окончание таблицы 6

<b>Название метода</b>	<b>Назначение</b>
Kzo_card	Получение документа КЗО
TP_card	Получение документа ПИП
LP_card	Получение документа ТЯИ
CurrentDomain_DLL	Проверка файлов DLL после составления меню

Описание данных, обрабатываемых методами класса Plugin1, представлено в таблице 7.

Таблица 7 – Данные методов класса Plugin1

<b>Название метода</b>	<b>Входящие данные</b>	<b>Выходящие данные</b>
InitUserDLLCom	Значение ссылки дескриптора PLM-системы и с помощью метода setmenu заполняет функциональное меню системы.	DLL файл для PLM-системы с меню функциональных кнопок.
setmenu	Значение ссылки дескриптора, название кнопки и функции, заданной кнопке.	Элемент меню для метода InitUserDLLCom.

Окончание таблицы 7

Название метода	Входящие данные	Выходящие данные
PgiCheckMenuItemCom	Значение ссылки дескриптора.	Значение в виде переменной bool.
Mes_card	Значение ссылки дескриптора.	Создает объект PLMserver и запускает модуль формирования документа MES.
Kzo_card	Значение ссылки дескриптора.	Создает объект Kzocard и запускает модуль формирования документа КЗО.
TP_card	Значение ссылки дескриптора.	Создает объект TPcard и запускает модуль формирования документа ПИП.
LP_card	Значение ссылки дескриптора.	Создает объект LPcard и запускает модуль формирования документа ТЯИ.

Интеграция модуля с системой осуществляется посредством обращения модуля к API PLM-системы – IPluginCall.

Используемые методы IPluginCall приведены в таблице 8.

Таблица 8 – Используемые методы IPluginCall

<b>Название метода</b>	<b>Назначение</b>
RunMethod	Запуск метода PLM-системы
GetDataSet	Получение набора данных из системы
Selected	Получение свойств выделенного объекта

Описание данных, обрабатываемых методами класса IPluginCall, представлено в таблице 9.

Таблица 9 – Данные используемых методов IPluginCall

<b>Название метода</b>	<b>Входные данные</b>	<b>Выходные данные</b>
RunMethod	Идентификатор метода в системе PLM.	Запуск метода в системе PLM.
GetDataSet	Идентификатор метода в системе PLM.	XML-файл с данными, соответствующими заданному методу.
Selected	Идентификатор свойства выделенного объекта.	Значение заданного свойства.

При инициализации класса Plugin1 задается кодировка данных, поддерживаемая системой PLM. После чего создается объект IPluginCall для

взаимодействия модуля с API PLM-системы. С помощью данного объекта мы можем получать информацию из PLM-системы.

С помощью метода `InitUserDLLCom` и вложенного в него метода `setmenu` создается меню функциональных кнопок и экспортируется как файл DLL в папку файлов PLM-системы, что проверяется функцией `CurrentDomain_DLL`.

После загрузки меню функциональных кнопок в системе PLM и активации кнопок «Получение документа MES», «Получение документа КЗО», «Получение документа ПИП», «Получение документа ТЯИ» инициализируются функции `Mes_card`, `Kzo_card`, `TP_card`, `LP_card`, соответствующие активированным кнопкам. Данные функции запускают работу блоков выдачи документов MES, КЗО, ПИП, ТЯИ.

#### 4.2. РЕАЛИЗАЦИЯ БЛОКА «ФОРМИРОВАНИЯ ОТЧЕТА КЗО»

Для решения данной задачи используются возможности обращения к API PLM-системы – `IPluginCall` для инициализации внутренних функций PLM-системы и получения необходимых данных в виде переменных типа `string`. Используемые методы `IPluginCall` и функции PLM-системы указаны в таблицах 10 и 11 соответственно.

Таблица 10 – Используемые методы `IPluginCall`

<b>Название метода</b>	<b>Назначение</b>
<code>RunMethod</code>	Запуск метода PLM-системы
<code>GetDataSet</code>	Получение набора данных из системы

Таблица 11 – Используемые функции PLM-системы

<b>Название функции</b>	<b>Назначение</b>
GetPropObjects	Возвращает свойства объекта
FieldValue [*]	Выдача значения заданного параметра * объекта
CheckOut	Выдача значения идентификатора открытого проекта
ConnectToCheckOut	Получение доступа к данным открытого проекта
GetLinkedFast2	Возвращает список версий объекта
KillVersionById	Помечает объект для удаления
CheckIn	Возвращает измененный объект в базу данных
GetInfoAboutVersion	Возвращает информацию о версии объекта
GetAttrPlainTextValue	Получает текстовое значение атрибута
NewObject	Создает новый объект
RegistrationOfFile	Регистрирует в базе данных новый файл, хранящийся на диске модуля

Получаемые от системы значения сохраняются в виде переменных в модуле. Переменные перечислены во втором столбце таблицы 12. Для формирования конечного документа переменные с помощью обращения к библиотеке Microsoft.Office.Interop.Word, входящей в API Microsoft Office, передаются в поля заранее созданного шаблона документа Microsoft Office. После чего документ сохраняется в виде PDF-файла.

Таблица 12 показывает полученные значений из системы и соответствующие им параметры модуля и поля документа.

Таблица 12 – Таблица соответствия данных

<b>Значения в системе</b>	<b>Параметр в модуле</b>	<b>Поле в документе</b>
Распорядительный документ	nom_naryad	№ наряда
№ заказа	nom_zak	Заказ №
Цех-потрибитель	dlya_cex	Для цеха
Обозначение узла	obozn_uzla	Обознач. узла
Наименование СТО	naim_osn	Наименование оснастки
Обозначение СТО	obozn_osn	Обознач. оснастки
Обозначение ДСЕ	obozn_det	Обознач. детали
Количество СТО	kol_osn	К-во изготовл. оснастки
Разработал	fam_sost	Составил
Трудоемкость	trudoemkos	Трудоемкость
Комплектующие	sod_zak	Содержание заказа оснастки

Таблица 13 показывает используемые функции библиотеки Microsoft.Office.Interop.Word, входящей в API Microsoft Office.

Таблица 13 – методы Microsoft.Office.Interop.Word

Название функции	Назначение
Documents.Add	Открытие документа для редактирования
Bookmarks	Закладка в документе
ExportAsFixedFormat	Экспорт файла с заданием адреса и формата
Close	Закрытие документа

При успешном составлении отчета и привязке его к системе выдается сообщение об успешном формировании файл. И файл становится виден в системе.

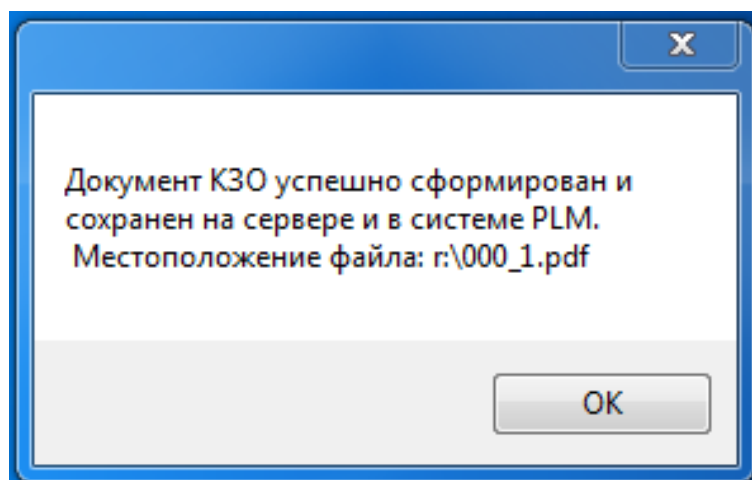


Рисунок 14 – Окно уведомления об успешном формировании документа

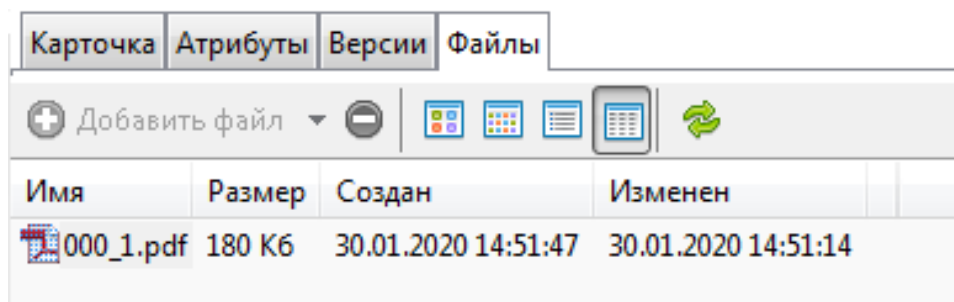


Рисунок 15 – Сформированный модулем файл в системе PLM



Блок формирования документа КЗО реализован с помощью класса KZOcard, представленного в листинге Б.1 приложения Б.

#### 4.3. РЕАЛИЗАЦИЯ БЛОКА «ФОРМИРОВАНИЕ ДОКУМЕНТА ТЯИ»

Для формирования транспортного ярлыка изделия так же, как и в блоке «Получение документа КЗО» используются возможности обращения к API PLM-системы – IPluginCall для инициализации внутренних функций PLM-системы и получения необходимых данных в виде переменных типа string. Используемые методы IPluginCall и функции PLM-системы указаны в таблицах 14 и 15 соответственно.

Таблица 14 – Используемые методы IPluginCall

<b>Название метода</b>	<b>Назначение</b>
RunMethod	Запуск метода PLM-системы
GetDataSet	Получение набора данных из системы

Таблица 15 – Используемые функции PLM-системы

<b>Название функции</b>	<b>Назначение</b>
GetPropObjects	Возвращает свойства объекта
FieldValue [*]	Выдача значения заданного параметра * объекта
CheckOut	Выдача значения идентификатора открытого проекта
ConnectToCheckOut	Получение доступа к данным открытого проекта

Окончание таблицы 15

Название функции	Назначение
GetLinkedFast2	Возвращает список версий объекта
KillVersionById	Помечает объект для удаления
CheckIn	Возвращает измененный объект в базу данных
GetInfoAboutVersion	Возвращает информацию о версии объекта
GetAttrPlainTextValue	Получает текстовое значение атрибута
NewObject	Создает новый объект
RegistrationOfFile	Регистрирует в базе данных новый файл, хранящийся на диске модуля

Получаемые значения от системы сохраняются в виде переменных в модуле. Переменные приведены во втором столбце таблицы 16. Для формирования конечного документа переменные с помощью обращения к библиотеке Microsoft.Office.Interop.Word, входящей в API Microsoft Office, передаются в поля заранее созданного шаблона документа Microsoft Office. После чего документ сохраняется в виде PDF-файла.

Таблица 16 показывает полученные из системы значения и соответствующие им параметры модуля и поля документа.

Таблица 16 – Таблица соответствия данных

<b>Значения в системе</b>	<b>Параметр в модуле</b>	<b>Поле в документе</b>
Цех-потребитель	ceh	Цех изготовитель
Обозначение узла	obozn_uzla	Узел
Обозначение ДСЕ	obozn_det	Деталь
Составляющих элементов	kolv_det	Количество деталей
Количество документации	kolv_doc	Количество сопроводительной документации
Масса	mass	Масса
Разработал технологию	fam_tec	Технолог
Изготовил	fam_ing	Инженер

Таблица 17 показывает используемые функции библиотеки Microsoft.Office.Interop.Word, входящей в API Microsoft Office.

Таблица 17 – методы Microsoft.Office.Interop.Word

<b>Название функции</b>	<b>Назначение</b>
Documents.Add	Открытие документа для редактирования
Bookmarks	Закладка в документе

## Окончание таблицы 17

Название функции	Назначение
ExportAsFixedFormat	Экспорт файла с заданием адреса и формата
Close	Закрытие документа

При успешном составлении отчета и привязке его к системе выдается сообщение об успешном формировании файла. Файл становится виден в системе.

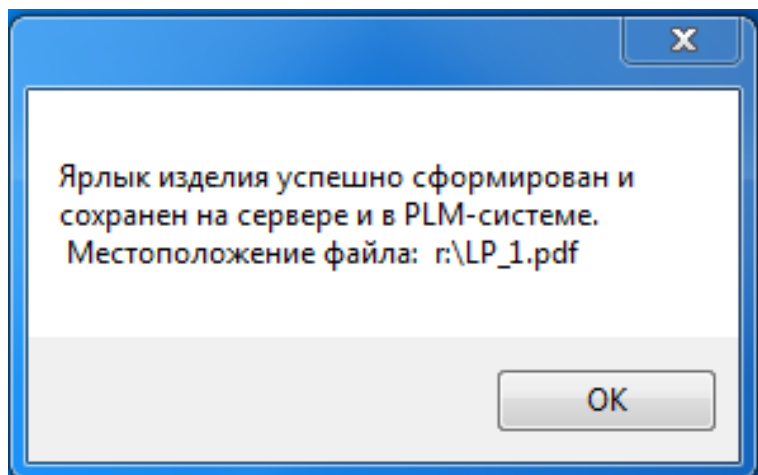


Рисунок 16 – Окно уведомления об успешном формировании документа

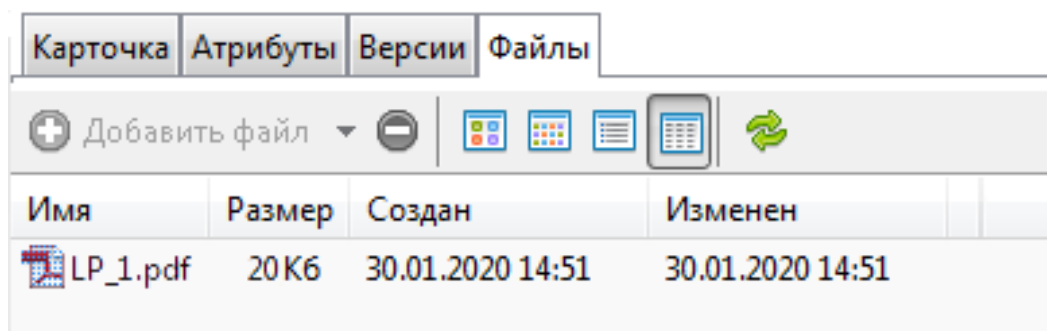


Рисунок 17 – Сформированный модулем файл в системе PLM

Блок формирования транспортного ярлыка изделия реализован с помощью класса LPcard, представленного в листинге В.1 приложения В.

#### 4.4. РЕАЛИЗАЦИЯ БЛОКА «ФОРМИРОВАНИЕ ДОКУМЕНТА ПИП»

Для формирования отчета по испытаниям продукции так же, как и в блоке «Получение документа КЗО» и блоке «Формирования ТЯИ» используются возможности обращения к API PLM-системы – IPluginCall для инициализации внутренних функций PLM-системы и получения необходимых данных в виде переменных типа string. Используемые методы IPluginCall и функции PLM-системы указаны в таблицах 18 и 19 соответственно.

Таблица 18 – Используемые методы IPluginCall

<b>Название метода</b>	<b>Назначение</b>
RunMethod	Запуск метода PLM-системы
GetDataSet	Получение набора данных из системы

Таблица 19 – Используемые функции PLM-системы

<b>Название функции</b>	<b>Назначение</b>
GetPropObjects	Возвращает свойства объекта
FieldValue [*]	Выдача значения заданного параметра * объекта
CheckOut	Выдача значения идентификатора открытого проекта
ConnectToCheckOut	Получение доступа к данным открытого проекта
GetLinkedFast2	Возвращает список версий объекта
KillVersionById	Помечает объект для удаления

Окончание таблицы 19

<b>Название функции</b>	<b>Назначение</b>
CheckIn	Возвращает измененный объект в базу данных
GetInfoAboutVersion	Возвращает информацию о версии объекта
GetAttrPlainTextValue	Получает текстовое значение атрибута
NewObject	Создает новый объект
RegistrationOfFile	Регистрирует в базе данных новый файл, хранящийся на диске модуля

Значения, полученные от системы, сохраняются в виде переменных в модуле. Переменные приведены во втором столбце таблицы 20. Для формирования конечного документа переменные с помощью обращения к библиотеке Microsoft.Office.Interop.Word, входящей в API Microsoft Office, передаются в поля заранее созданного шаблона документа Microsoft Office. После чего документ сохраняется в виде PDF-файла.

Таблица 20 показывает полученные значений из системы, соответствующие им параметры модуля и поля документа.

Таблица 20 – Таблица соответствия данных

<b>Значения в системе</b>	<b>Параметр в модуле</b>	<b>Поле в документе</b>
Этап статических испытаний	stat	Статические испытания
Ответственный за стат. испытания	Fam_stat	Ответственное лицо 1

Окончание таблицы 20

<b>Значения в системе</b>	<b>Параметр в модуле</b>	<b>Поле в документе</b>
Этап вибрационных испытаний	vibro	Вибрационные испытания
Ответственный за вибр. испытания	fam_vibro	Ответственное лицо 2
Этап термических испытаний	term	Термические испытания
Ответственный за терм. испытания	fam_term	Ответственное лицо 3
Этап аэродинамических испытаний	aero	Аэродинамические испытания
Ответственный за аэро. испытания	fam_aero	Ответственное лицо 4
Этап баллистических испытаний	balis	Баллистические испытания
Ответственный за бал. испытания	fam_bal	Ответственное лицо 5
Обозначение ДСЕ	obozn_det	Деталь
Обозначение узла	obozn_uzla	Узел
Руководитель испытаний	ryk_isp	Руководитель испытаний

Таблица 21 показывает используемые функции библиотеки Microsoft.Office.Interop.Word, входящей в API Microsoft Office.

Таблица 21 – методы Microsoft.Office.Interop.Word

Название функции	Назначение
Documents.Add	Открытие документа для редактирования
Bookmarks	Закладка в документе
ExportAsFixedFormat	Экспорт файла с заданием адреса и формата
Close	Закрытие документа

При успешном составлении отчета и привязке его к системе выдается сообщение об успешном формировании файла. Файл становится виден в системе.

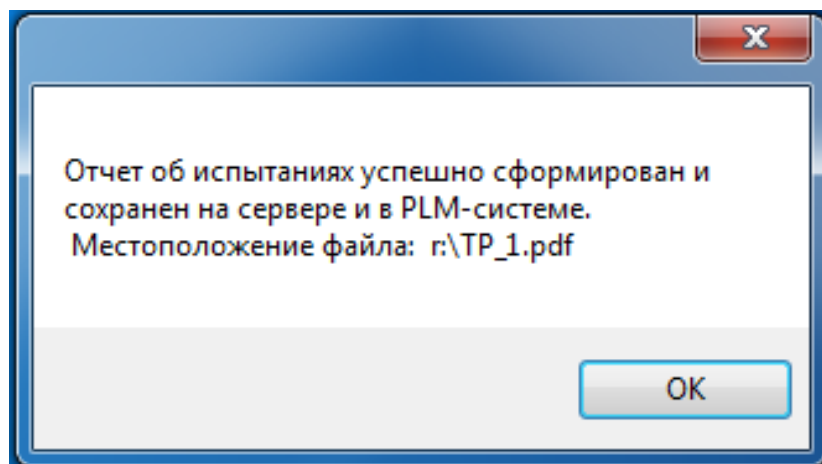


Рисунок 18 – Окно уведомления об успешном формировании документа



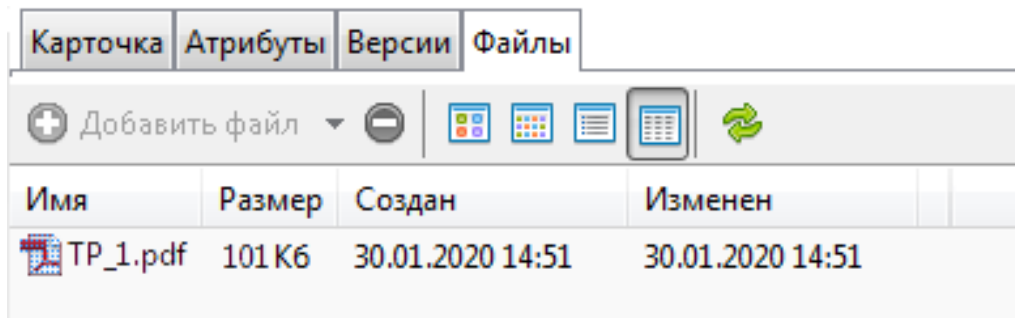


Рисунок 19 – Сформированный модулем файл в системе PLM

Блок формирования отчета испытаний продукта реализован с помощью класса TPcard, представленного в листинге Г.1 приложения Г.

#### 4.5. РЕАЛИЗАЦИЯ БЛОКА «СЕРВЕРНЫЙ МЕНЕДЖЕР»

Реализация блока обращения к серверу и обработки полученной информации основана на применении методов собственного класса PLMserver. Класс PLMserver содержит в себе 21 метод. Назначение методов указано в таблице 22.

Таблица 22 – Методы класса PLMserver

Название метода	Назначение
Disconnect	Отключение от сервера
Connect	Подключение к серверу и задание формата обмена данных: XML
SelectDB	Подключение к базе данных
GetDBList	Получение списка баз данных сервера
GetTypeList	Получение списка всех типов баз данных
GetLinkList	Получение списка всех типов связи

## Окончание таблицы 22

<b>Название метода</b>	<b>Назначение</b>
ParseFields	Обработчик информации, полученной от сервера
EnterToEditMode	Вход в режим редактирования данных
ExitFromEditMode	Выход из режима редактирования данных
ObjectExists	Проверка объекта на существование
GetAtrListFromTxt	Извлечение списка атрибутов
ListAtrGetTwo	Извлечение списка необходимых атрибутов
GetIdList	Получение листа идентификаторов атрибутов
GetAtrLink	Получение id связи объекта
CreateDate	Получение даты создания объекта
CreateName	Получение наименования объекта
GetChild	Поиск составляющих объекта
GetLinkAtr_Poz	Получение атрибута позиции составляющей объекта
GetChild_Id	Получение ID связей составляющих объекта
InfoElm	Получение информации о составляющей объекта для отчета MES
Cycle_elm	Получение списка составляющих объекта

Описание данных, обрабатываемых методами класса PLMserver, представлено в таблице 23.

Таблица 23 – Данные методов класса PLMserver

<b>Название метода</b>	<b>Входные данные</b>	<b>Выходные данные</b>
Disconnect	Имя сервера	Логическая переменная
Connect	Имя сервера	Логическая переменная
SelectDB	Имя базы данных, имя пользователя, пароль	Логическая переменная
GetDBList	XML-документ	Список имеющихся баз данных на сервере
GetTypeList	XML -документ	Список типов имеющихся на сервере баз данных
GetLinkList	XML -документ	Список типов связи
ParseFields	XML -документ	Словарь атрибутов и их значений
EnterToEditMode	ID объекта, атрибут изменяемого значения, значение	Отредактированный объект
ExitFromEditMode	ID объекта	Закрытие редактирование объекта
ObjectExists	ID объекта, тип, версия	Объект системы
GetAttrListFromTxt	Текстовые параметры системы	Список всех атрибутов

Окончание таблицы 23

<b>Название метода</b>	<b>Входные данные</b>	<b>Выходные данные</b>
ListAttrGetTwo	Текстовые параметры системы	Список необходимых атрибутов
GetIdList	XML-файл	Список идентификаторов атрибутов
GetAttrLink	ID объекта, название атрибута	Id-связи
CreateDate	ID объекта	Дата создания
CreateName	ID объекта	Наименование объекта
CreateGotov	ID объекта	Переменная типа int, где 0-Не готов, 1-Изготавливается, 2-Готов
GetChild	ID объекта	Словарь составляющих объекта
GetLinkAttr_Poz	ID объекта	Значение атрибута
GetChild_Id	ID объекта	Словарь составляющих объекта с ID
InfoElm	Элемент структуры elm	Строка отчета MES
Cycle_elm	ID объекта	Словарь составляющих объекта с необходимыми атрибутами

При инициализации, класс PLMserver устанавливает соединение с сервером и задает формат обмена данными с помощью метода Connect.

В дальнейшем используются методы GetDBList и SelectDB для выбора и подключения к необходимой базе данных, используя локальную аутентификацию.

В случае возникновения ошибки подключения на любом уровне, блок информирует пользователя.

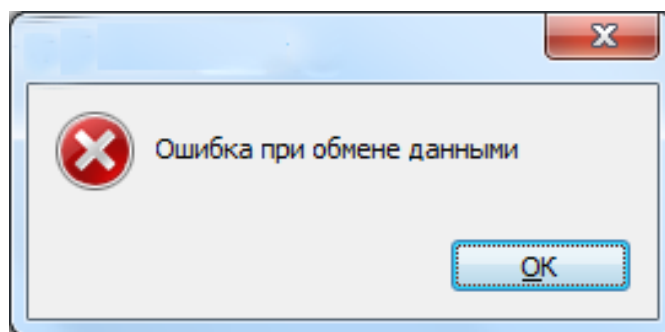


Рисунок 20 – Окно ошибки блока серверного менеджера

Остальные методы класса PLMserver служат для редактирования объектов системы или получения информации об этих объектах. Необходимость данных методов обусловлена ограниченностью функционала API PLM-системы.

Блок серверного менеджера реализован с помощью класса PLMserver, представленного в листинге Д.1 приложения Д.

#### 4.6. РЕАЛИЗАЦИЯ БЛОКА «ПОЛУЧЕНИЕ ОТЧЕТА MES»

Для формирования отчета об изделии для MES-системы не используются возможности обращения к API PLM-системы – IPluginCall. Вместо них, для получения информации для отчета, используется обращение к серверу с помощью класса PLMserver. С помощью методов Connect и SelectDB из класса PLMserver происходит подключение к серверу и базе данных системы PLM.

В классе PLMserver происходит получение необходимой информации с помощью методов ParseField, GetAttrListFromTxt, ListAttrGetTwo, GetIdList, GetAttrLink, CreateDate, CreateName, GetChild, GetChild\_Id. Далее полученная информация представляется в необходимом формате с помощью функций InfoElm, Cycle\_elm и данные передаются в класс MEScard в виде сформированных строк. Информационные данные в классе MEScard хранятся в виде трех переменных, представленных в таблице 24.

Таблица 24 – Информационные переменные класса MEScard

<b>Название переменной</b>	<b>Назначение</b>
rmInfoElm	Информационная строка одного элемента отчета
srmInfoElm	Сумма строк всех элементов отчета
otchet	Конечные данные, передаваемые в отчет

Для формирования конечного документа, единственная переменная otchet, представляющая собой полный набор информации, с помощью обращения к библиотеке Microsoft.Office.Interop.Word, входящей в API Microsoft Office, передается в заранее созданный шаблон документа MES. После чего документ сохраняется в виде PDF-файла. Для реализации действий переноса и сохранения информации используются методы, указанные в таблице 25.

Таблица 25 – методы Microsoft.Office.Interop.Word

<b>Название функции</b>	<b>Назначение</b>
Documents.Add	Открытие документа для редактирования
Bookmarks	Закладка в документе

## Окончание таблицы 25

Название функции	Назначение
ExportAsFixedFormat	Экспорт файла с заданием адреса и формата
Close	Закрытие документа

При успешном составлении отчета и экспорте его на диск системы MES выдается сообщение об успешном формировании файла. Файл становится виден в системе.

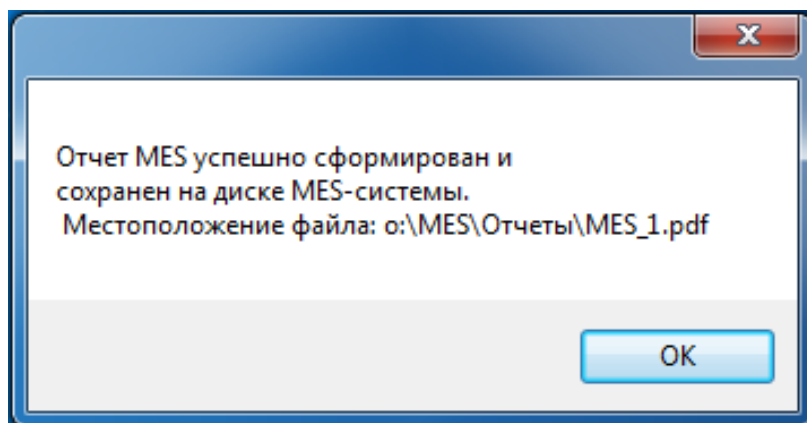


Рисунок 21 – Окно уведомления об успешном формировании документа

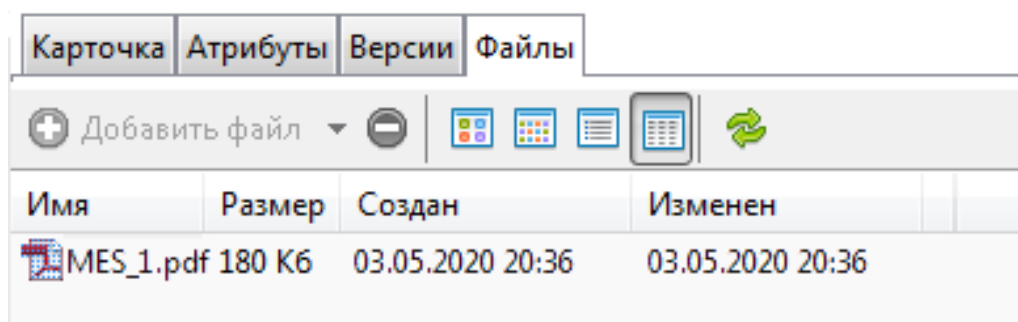


Рисунок 22 – Сформированный модулем файл в системе PLM

Блок формирования отчета об изделии для MES-системы реализован с помощью класса MEScard, представленного в листинге E.1 приложения E.

## 5. ТЕСТИРОВАНИЕ

### 5.1. МЕТОДОЛОГИЯ ТЕСТИРОВАНИЯ

Модуль успешно прошел отладку.

Программа прошла альфа тестирование, которое проводилось на тестовом изделии. Тестовое изделие создавалось в системе PLM. После создания изделия, нажатием на функциональные кнопки «Получение документа КЗО», «Получение документа MES», «Получение документа ПИП», «Получение документа ТЯИ» в PLM-системе, активировались скрипты с классами KZOcard, MEScard, TPcard, LPCard, отвечающие за создание нормативных документов. Информационное окно и появление файла в разделе документации, свидетельствовали об успешном формировании документа.

Альфа-тестирование было завершено проверкой данных в сформированных документах. После чего, в связи с отсутствием ошибок, модуль был передан заказчику. Далее тестирование проводилось на реальных данных.

Чтобы проверить работоспособность и безошибочность системы, работа программы контролировалась и отслеживалась в течение месяца для выявления сбоев и ситуаций, не предусмотренных ранее. После тестирования программа была полностью введена в эксплуатацию.

### 5.2. ПРОВЕДЕНИЕ ПРОЦЕДУРЫ ТЕСТИРОВАНИЯ

В данном пункте приведен пример тестирования на изделии АБВГ.600000.000 ЭСБ. Тестовый элемент был создан в системе PLM и полностью описан, согласно требованиям конструкторской спецификации.



После создания изделия, нажатием на соответствующую функциональную кнопку, была активирована функция «Получение документа КЗО». В течение 3 секунд модуль отчетов выдал информационное окно о создании документа, представленное на рисунке 23.

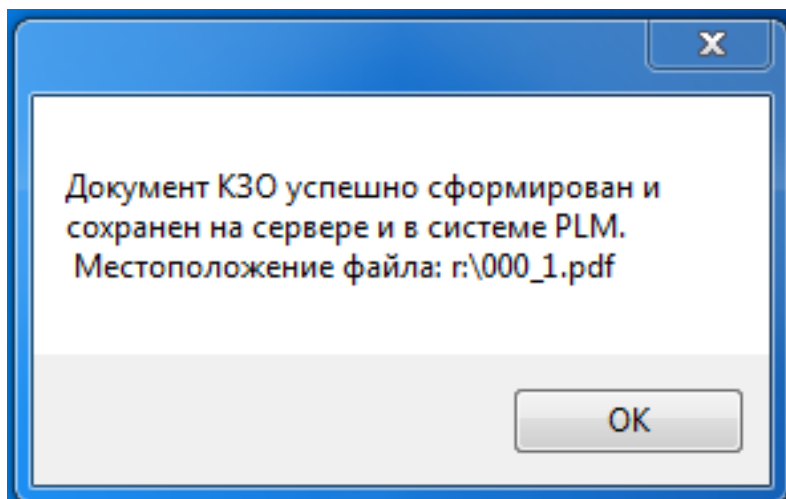


Рисунок 23 – Окно уведомления об успешном формировании документа

Документ появился в разделе документации системы PLM.

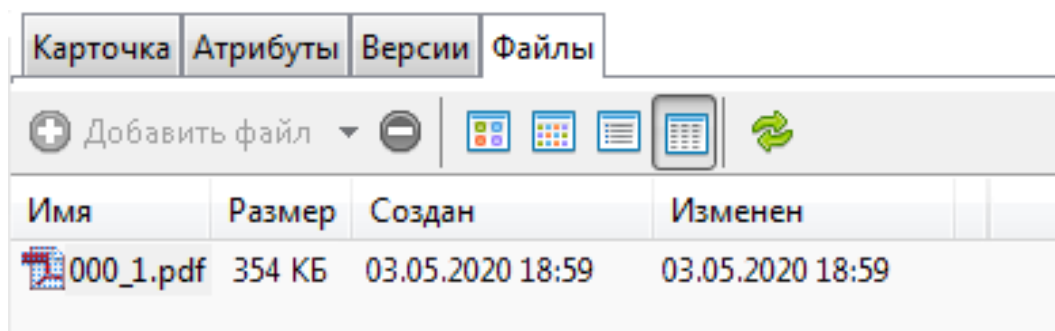


Рисунок 24 – Сформированный модулем файл в системе PLM

Как видно из рисунка 25, документ был успешно сформирован.

№ наряда 0001	Заказ № 002		Для цеха 222		
Обознач. узла	Трудоемкость		Тема №		
<i>АБВГ.000000.000</i>	1000		Карта №		
Обознач. детали	Наименование детали		Обознач. оснастки		
<i>АБВГ.600000.000 ЭСБ</i>	<i>Компрессор низкого давления</i>		<i>1100001110</i>		
Кооперация	К-во изготвл. оснастки				
	<i>1</i>				
Содержание заказа					
- Набор СК40 с латунными подкладками (3шт.)					
- 2002-Т(1шт.)					
-Заготовка КВ010103 (1шт.)					
	Фамилия	Подпись	Дата	Срок выд. КД	
Составил	Лукашов			Срок изгот., оснастки	
Проверил				111/222	
Утвердил				Нач. БИХ	
Констр				Накладная №	
Рук. гр.				Дата выполн. заказа	

Рисунок 25 – Сформированный документ КЗО

После получения документа КЗО, нажатием на соответствующую функциональную кнопку, была активирована функция «Получение документа ПИП». В течение 3 секунд модуль отчетов выдал информационное окно о создании документа, представленное на рисунке 26.

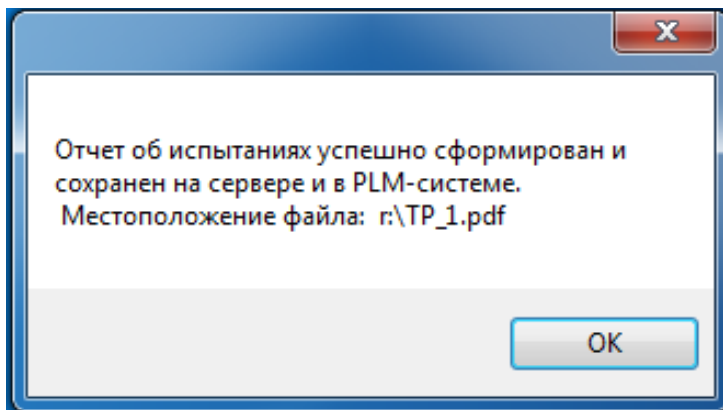


Рисунок 26 – Окно уведомления об успешном формировании документа

Документ появился в разделе документации системы PLM.

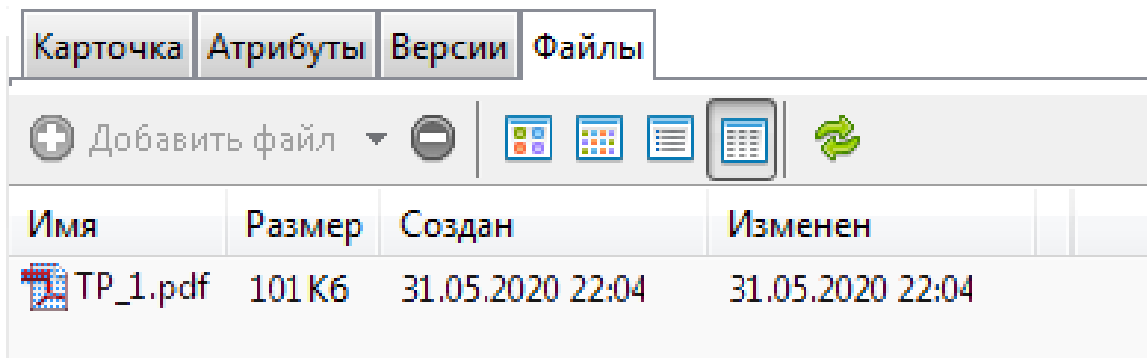


Рисунок 27 – Сформированный модулем файл в системе PLM

Таким образом, как видно из рисунка 28, документ был успешно сформирован согласно шаблону.

**ПРОТОКОЛ ИСПЫТАНИЙ ПРОДУКЦИИ**

Деталь	<i>АБВГ.600000.000 ЭСБ</i>
Узел	<i>АБВГ.000000.000</i>

<i>Руководитель испытаний</i>	<i>Подпись</i>
<i>Лукашов Виталий Евгеньевич</i>	

<i>Вид испытаний</i>	<i>Этап испытаний</i>	<i>Ответственный</i>
<i>Статические испытания</i>	<i>Пройдены</i>	<i>Лукашов</i>
<i>Вибрационные испытания</i>	<i>Не пройдены</i>	<i>Лукашов</i>
<i>Термические испытания</i>	<i>Планируются</i>	<i>Лукашов</i>
<i>Аэродинамические испытания</i>	<i>Планируются</i>	<i>Лукашов</i>
<i>Баллистические испытания</i>	<i>Планируются</i>	<i>Лукашов</i>

М.П.

Рисунок 28 – Сформированный документ ПИП

После получения документа ПИП, нажатием на соответствующую функциональную кнопку, была активирована функция «Получение документа ТЯИ». В течении 3 секунд модуль отчетов выдал информационное окно о создании документа, представленное на рисунке 29.

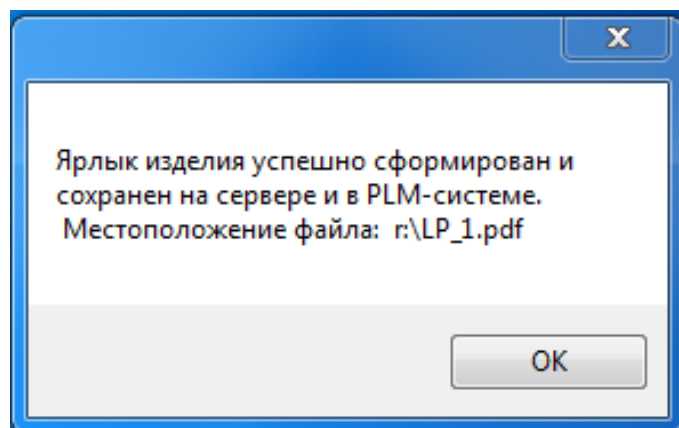


Рисунок 29 – Окно уведомления об успешном формировании документа

Документ появился в разделе документации системы PLM.

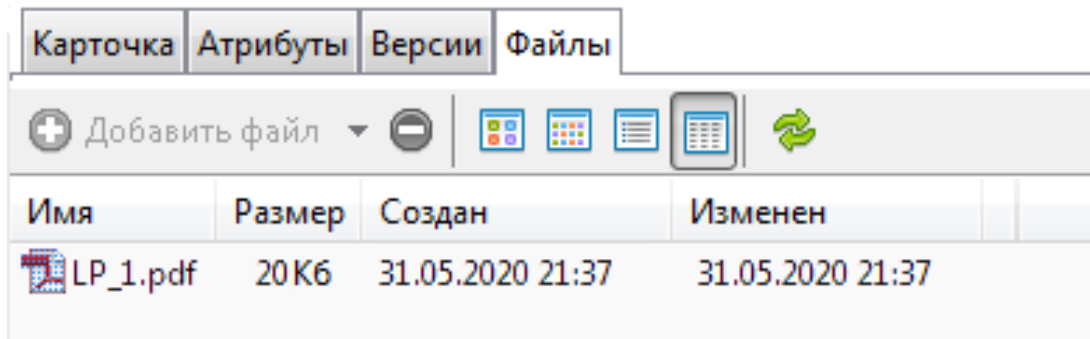


Рисунок 30 – Сформированный модулем файл в системе PLM

На рисунке 31 показан документ, сформированный согласно шаблону.

<b><i>ТРАНСПОРТНЫЙ ЯРЛЫК ИЗДЕЛИЯ</i></b>	
<i>Деталь</i>	<i>АБВГ.600000.000 ЭСБ</i>
<i>Узел</i>	<i>АБВГ.000000.000</i>
<i>Кол-во деталей</i>	<i>10</i>
<i>Кол-во сопр. док.</i>	<i>3</i>
<i>Масса(кг)</i>	<i>2548</i>
<i>Цех изготовитель</i>	<i>000</i>
<i>Технолог</i>	<i>Лукашов</i>
<i>Инженер</i>	<i>Лукашов</i>

Рисунок 31 – Сформированный документ ТЯИ

После получения документа ТЯИ, нажатием на соответствующую клавишу, была активирована функция «Получение документа MES». В течении

10 секунд модуль отчетов выдал информационное окно о создании документа, представленное на рисунке 32.

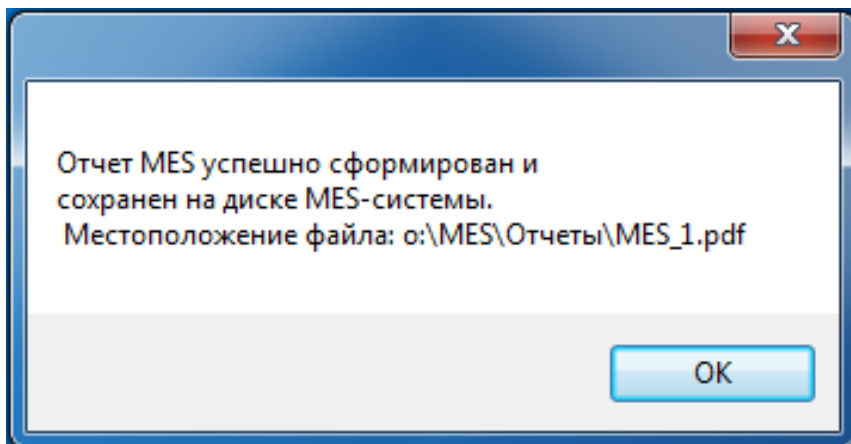


Рисунок 32 – Окно уведомления об успешном формировании документа

Документ был сформирован, появился в системе и был отправлен на диск системы MES в папку отчетов.

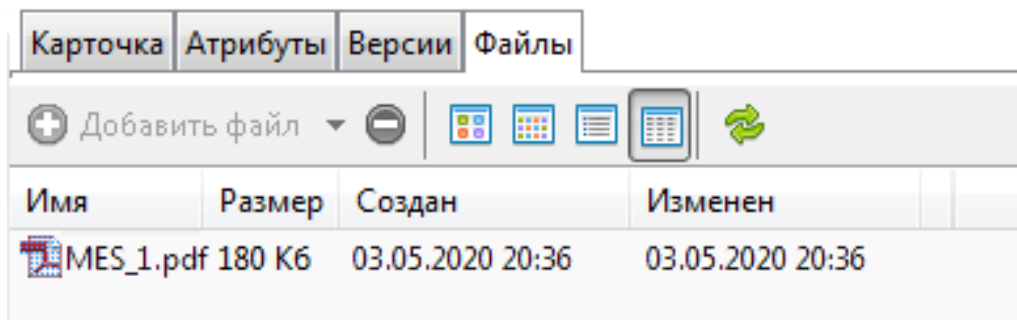


Рисунок 33 – Сформированный модулем файл в системе PLM

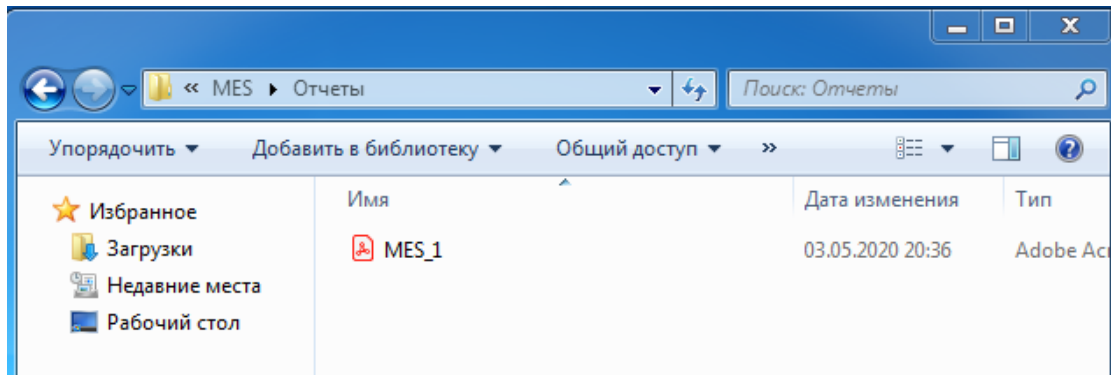


Рисунок 34 – Файл в папке системы MES

На рисунке 35 показано, что документ успешно сформирован согласно шаблону документа.

-- ID: АБВГ.600000.000 Наименование: Компрессор низ. давления Уровень: 1 Готовность: Изготавливается	----- ID: АБВГ.680000.004 Наименование: Барабан 3-4 ступени Уровень: 3 Готовность: Готово
--- ID: АБВГ.110000.000 Наименование: Клапан стравливания Уровень: 2 Готовность: Не готово	----- ID: АБВГ.680000.005 Наименование: Цапфа-диск 5 Уровень: 3 Готовность: Изготавливается
--- ID: АБВГ.110000.000-01 Наименование: Клапан стравливания Уровень: 2 Готовность: Готово	----- ID: АБВГ.680000.006 Наименование: Диск 6 ступени Уровень: 3 Готовность: Изготавливается
--- ID: АБВГ.200000.000 Наименование: Корпус передний Уровень: 2 Готовность: Готово	----- ID: АБВГ.680000.007 Наименование: Лопатка 1 ступени Уровень: 3 Готовность: Не готово
--- ID: АБВГ.300000.000 Наименование: Механизм поворота Уровень: 2 Готовность: Готово	----- ID: АБВГ.680000.008 Наименование: Лопатка 2 ступени Уровень: 3 Готовность: Готово
--- ID: АБВГ.300000.000-01 Наименование: Механизм поворота Уровень: 2 Готовность: Готово	
--- ID: АБВГ.400000.000 Наименование: Обтекатель внутренний Уровень: 2 Готовность: Готово	
--- ID: АБВГ.500000.000 Наименование: ПВЦ Уровень: 2 Готовность: Готово	
--- ID: АБВГ.600000.000 Наименование: ПЦО Уровень: 2 Готовность: Готово	
--- ID: АБВГ.610000.000 Наименование: Корпус КНД Уровень: 2 Готовность: Готово	
--- ID: АБВГ.680000.000 Наименование: Ротор Уровень: 2 Готовность: Изготавливается	
----- ID: АБВГ.680000.001 Наименование: Цапфа передняя Уровень: 3 Готовность: Готово	
----- ID: АБВГ.680000.002 Наименование: Диск 1 ступени Уровень: 3 Готовность: Готово	
----- ID: АБВГ.680000.003 Наименование: Диск 2 ступени Уровень: 3 Готовность: Готово	

Рисунок 35 – Сформированный документ MES

Тестирование на возникновение неполадок в обмене данными системы с сервером проводилось с помощью преднамеренного нарушения канала связи и отслеживания реакции системы. В случае возникновения неполадок система информировала пользователя с помощью информационного окна.

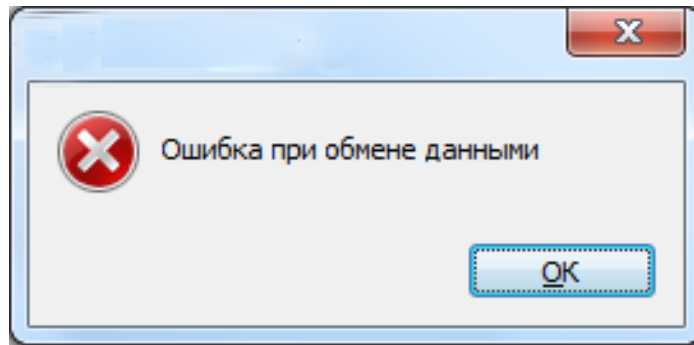


Рисунок 36 – Окно ошибки блока подключения к серверу

Таким образом, программное обеспечение прошло тестирование всех внутренних блоков. В ходе тестирования было установлено, что формирование всех отчетов производится правильно.



## ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы были решены все основные задачи, связанные с проектированием, разработкой и вводом в эксплуатацию модуля формирования отчетов в системе управления производством.

В ходе выполнения работы:

- рассмотрены существующие системы автоматизации и управления производством;
- проведен анализ имеющихся на предприятии систем управления и их недостатков;
- проведено сравнение трех языков программирования для разработки ПО;
- разработан программный продукт автоматизации отчетности на уровне PLM-системы;
- ПО интегрировано в имеющуюся систему управления производством;
- проведено тестирование разработанного программного обеспечения;
- ввод программного продукта в эксплуатацию.

В настоящее время модуль введен в эксплуатацию и выполняет все заданные функции. С помощью модуля удалось повысить уровень автоматизации производства на уровне PLM и MES-систем, уменьшить издержки предприятия.

В дальнейшем планируется увеличение функциональных возможностей модуля формирования отчетов за счет увеличения количества форм отчетов и реализации взаимодействия с системами ERP и CRM. Взаимодействие модуля со всеми системами управления и автоматизации производства обеспечит безошибочное формирование документооборота и предоставит возможность обмена документами на всех уровнях предприятия.

Схема перспективного взаимодействия модуля с системами автоматизации и управления производством представлена на рисунке 37.

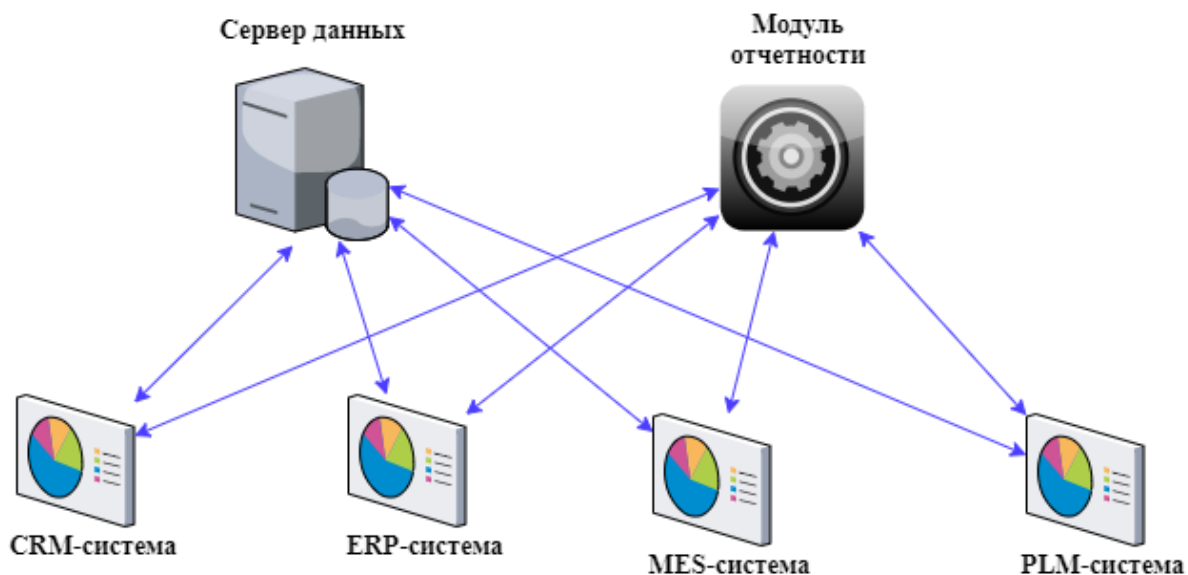


Рисунок 37 – Схема перспективного взаимодействия модуля с системами

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Кинзябулатов, Р.Х. CRM. Подробно и по делу: справочник. / Р.Х. Кинзябулатов. – М.: Издательские решения, 2009. – 430 с.
2. Балашова, Ю. Эффективные системы управления производством. – <https://www.cfin.ru/itm/kis/choose/Manufacturing.shtml> Дата обращения: 20.03.2020.
3. Системы управления жизненным циклом сложных объектов – <https://constructor.ru/solutions/967/> Дата обращения: 20.03.2020.
4. Бородин, И.Ф. Автоматизация технологических процессов и системы автоматического управления: учебник. / И.Ф. Бородин, С.А. Андреев. – 2-е изд. – М.: Юрайт, 2018. – 386 с.
5. Альбитер, А.М. Концепция рационального управления производственной инфраструктурой промышленных комплексов / А.М. Альбитер, С.А. Лочап. – М.: Палеотип, 2009. – 92 с.
6. Яшин, А.А. Логистика. Основы планирования и оценки эффективности логистических систем: учебное пособие / А.А. Яшин, М.Л. Ряшко. – М.: Бибком, 2014. – 60 с.
7. Шилдт, Г. С# 4.0. Полное руководство / Г. Шилдт. – М.: Вильямс, 2019. – 1056 с.
8. Евгеньев, Г.В. Основы автоматизации технологических процессов и производств. Учебное пособие. В 2 т. Т. 1: Информационные модели / Г.В. Евгеньев, А.В. Грошев, С.С. Гаврюшин. – М.: МГТУ им. Н. Э. Баумана, 2015. – 448 с.
9. Щагин, А.В. Основы автоматизации техпроцессов. Учебное пособие для вузов / А. В. Щагин, А. Б. Кабанова, В. И. Демкин, В. Ю. Кононов. – М.: Юрайт, 2009. – 164 с.

10. Гарольд, Э.Р. XML. Справочник / Э.Р. Гарольд, У.С. Минс; пер. с англ. Л. Фрейдин. – М.: Символ-Плюс, 2017. – 567 с.
11. Хантер, Д. XML. Базовый курс / Д. Хантер, Д. Рафтер, Д. Фосетт, Э. Влист и др.; пер. с англ. И.В. Красиков. – 4-е изд. – М.: Вильямс, 2018. – 1344 с.
12. Троелсен, Э. Язык программирования C# 7 и платформы .NET и .NET Core / Э. Троелсен, Ф. Джепикс; пер. с англ. Ю. Н. Артеменко. – 8-е изд. – М.: Вильямс, 2018. – 1328 с.
13. Саймон, Р. Microsoft Windows API. Справочник системного программиста / Р. Саймон. – 4-е изд. – М.: DiaSoft, 2004. – 1217 с.
14. Прайс, М. C# 7 и .NET Core. Кросс-платформенная разработка для профессионалов / М. Прайс; пер. с англ. С.В. Черников. – 3-е изд. – СПб.: Питер, 2018. – 640 с.
15. Фримен, А. ASP.NET Core MVC 2 с примерами на C# для профессионалов / А. Фримен. – 7-е изд. – М.: Диалектика, 2019. – 1008 с.