

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(институт)

ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА

(кафедра)

09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

(код и наименование направления подготовки, специальности)

БИЗНЕС-ИНФОРМАТИКА

(наименование профиля, специализации)

БАКАЛАВРСКАЯ РАБОТА

на тему «Разработка автоматизированного решения для обмена данными между
PLM системой и представителями поставщиков»

Студент Кубасов Александр Сергеевич _____
(личная подпись)

Руководитель Шляпкин Андрей Владимирович _____
(личная подпись)

Допустить к защите

Заведующий кафедрой к.т.н., доцент, А.В. Очеповский _____
(личная подпись)

« _____ » _____ 20 _____ г.

Тольятти 2018



Росдистант
ВЫСШЕЕ ОБРАЗОВАНИЕ ДИСТАНЦИОННО

АННОТАЦИЯ

К БАКАЛАВРСКОЙ РАБОТЕ

Студента 5 курса, группы ПИБд-1302а

КУБАСОВ АЛЕКСАНДРА СЕРГЕЕВИЧА

(ФИО)

На тему: Разработка автоматизированного решения для обмена данными между PLM системой и представителями поставщиков

Целью бакалаврской работы является разработка информационной системы автоматизированного обмена данными между PLM системой и представителями поставщиков.

Объектом исследования бакалаврской работы является бизнес-процесс обмена данными между PLM системой и представителями поставщиков.

Предметом исследования бакалаврской работы является ресурсоемкость обмена данными между PLM системой и представителями поставщиков.

Структура дипломной работы представлена введением, тремя главами, заключением, списком использованной литературы и приложениями.

В первой главе дано описание компании S&C и ее деятельности, представлена организационная структура. Описан существующий бизнес-процесс обмена данными и выявлены его недостатки. Разработана модель бизнес-процесса автоматизированного обмена данными и сформированы требования к проектируемой ИС.

Во второй главе разработаны логическая и физическая модель данных исследуемой предметной области. Разработана ИС и описана её функциональность.

В третьей главе произведен расчет экономической целесообразности разрабатываемой системы.

Работа представлена на 83 листах, включает 26 иллюстраций, 3 таблицы и 9 приложений на 29 листах. Биография содержит 33 источника.

ОГЛАВЛЕНИЕ

| | |
|---|----|
| ВВЕДЕНИЕ..... | 5 |
| ГЛАВА 1. АНАЛИЗ КОМПАНИИ S&C И ОТДЕЛА PSS..... | 8 |
| 1.1. Техничко-экономическая характеристика процесса обмена данными между PLM системой S&C и аутсорсинговыми компаниями..... | 8 |
| 1.1.1. Характеристика компании S&C | 8 |
| 1.1.2. Краткая характеристика подразделения Post Sale Services и его видов деятельности..... | 9 |
| 1.1.3. Сущность задачи автоматизации процесса обмена данными..... | 10 |
| 1.2. Концептуальное моделирование процесса обмена данными..... | 12 |
| 1.3. Постановка задачи разработки автоматизированного решения для процесса обмена данными..... | 15 |
| 1.3.1. Цель и назначение автоматизации процесса обмена данными..... | 15 |
| 1.3.2. Общая характеристика разрабатываемой системы для решения поставленной задачи..... | 16 |
| 1.4. Анализ существующих разработок и обоснование выбора технологии проектирования системы для интеграции с PLM Enovia..... | 17 |
| Выводы по главе 1..... | 19 |
| ГЛАВА 2. РАЗРАБОТКА И РЕАЛИЗАЦИЯ ПРОЕКТНОГО РЕШЕНИЯ | 21 |
| 2.1. Логическое моделирование процесса обмена данными между PLM системой и поставщиками..... | 21 |
| 2.1.1. Логическая модель процесса обмена данными и ее описание..... | 21 |
| 2.1.2. Характеристика входной информации для работы подсистем File Sharing System..... | 24 |
| 2.1.3. Характеристика базы данных подсистемы Web Exchange | 25 |
| 2.1.4. Характеристика результатной информации подсистемы PLM Integration | 26 |
| 2.2. Физическое моделирование File Sharing System..... | 27 |
| 2.2.1. Выбор архитектуры File Sharing System..... | 27 |
| 2.2.2. Функциональная схема подсистем File Sharing System | 30 |
| 2.2.3. Структурная схема File Sharing System | 31 |

| | |
|---|----|
| 2.2.4. Описание программных модулей File Sharing System | 33 |
| 2.3. Технологическое обеспечение процесса обмена данными | 35 |
| 2.3.1. Организация технологии сбора, передачи, обработки и выдачи информации..... | 35 |
| 2.3.2. Схема технологического процесса сбора, передачи, обработки и выдачи информации..... | 37 |
| 2.4. Контрольный пример реализации File Sharing System и ее описание | 39 |
| Выводы по главе 2..... | 41 |
| ГЛАВА 3. ОЦЕНКА И ОБОСНОВАНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ РАЗРАБОТКИ FILE SHARING SYSTEM..... | 42 |
| 3.1. Выбор и обоснование методики расчета экономической эффективности.... | 42 |
| 3.2. Расчет показателей экономической эффективности File Sharing System | 43 |
| Выводы по главе 3..... | 48 |
| ЗАКЛЮЧЕНИЕ | 49 |
| СПИСОК ЛИТЕРАТУРЫ И ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ..... | 50 |
| Приложение 1. Диаграмма потоков данных, отражающая схему процесса обмена данными с поставщиком, по которой S&C работали до внедрения автоматизированного решения..... | 54 |
| Приложение 2. Диаграмма потоков данных, на которой изображена общая схема, по которой S&C начали работать после внедрения автоматизированного решения..... | 55 |
| Приложение 3. Диаграмма взаимодействия РМ и поставщика с File Sharing System..... | 57 |
| Приложение 4. JSON структура контейнера | 61 |
| Приложение 5. JSON структура письма..... | 63 |
| Приложение 6. JSON структура уникальной ссылки..... | 64 |
| Приложение 7. Блок-схемы процессов в подсистемах..... | 65 |
| Приложение 8. Контрольный пример реализации File Sharing System и ее описание..... | 68 |
| Приложение 9. Пример Container REST API..... | 74 |

ВВЕДЕНИЕ

Обмен данными представляет собой сложный и очень ресурсозатратный процесс как в материальном отношении, так и человеко-часах. На каждом предприятии этот процесс имеет свою специфику, но общая цель каждого предприятия, организации, фирмы едина – снизить ресурсозатратность, повысить безопасность передачи данных и упростить весь процесс. В данной бакалаврской работе в качестве предметной области рассмотрим в общем деятельность компании S&C Electric Company, особое внимание уделим взаимодействию сотрудников компании с аутсорсинговыми компаниями через PLM систему Enovia. Именно этот процесс требовал автоматизации, цель которой заключается не только в упрощении и ускорении документооборота, но также в снижении ресурсозатратности. В итоге автоматизация прошла успешно за счет разработки и внедрения расширения File Sharing System к PLM Enovia.

Актуальность бакалаврской работы заключается в следующем:

- обеспечить безопасность передаваемых исполнителям данных от несанкционированного доступа;
- снизить вероятность потери данных или их части в процессе обмена;
- упростить работу с большими объемами пересылаемых данных;
- сократить финансовые затраты на пользование лицензионным соглашением.

С PLM системами работают, в основном, крупные организации, которым важно сохранение большого объема данных об изготавливаемой продукции. Есть научно-популярные работы, посвященные разбору конкретных PLM систем, систем для разработки PLM, сравнению PLM и PDM (системы управления данными об изделии). На рынке без труда можно найти поставщиков PLM систем. Тема данной бакалаврской работы удовлетворяет веяниям рынка информационных систем.

Целью выполнения работы является упрощение и снижение ресурсозатрат процесса документооборота между PLM и внешними поставщиками услуг.

Задачи, которые были поставлены при написании данной бакалаврской работы:

- собрать информацию о существующих подходах обмена данными между PLM и внешними поставщиками услуг;
- проанализировать и выявить текущие проблемы и потребности пользователей PLM Eovia и поставщиков услуг;
- описать процесс обмена данными между PLM и внешними поставщикам услуг после внедрения системы автоматизации;
- разобрать интерфейс по управлению системы обмена данными между PLM и внешними поставщиками услуг;
- реализовать систему обмена данными.

Объектом исследования является процесс обмена данными между PLM системой и поставщиками услуг.

Предмет исследования – ресурсоемкость процесса обмена данными.

Используемые методы исследования – структурный подход к анализу и проектированию, CASE-технологии структурного и объектно-ориентированного анализа и проектирования.

Основная структура данной работы имеет следующий вид:

1. Глава первая – Анализ компании S&C и отдела PSS.
2. Глава вторая – Разработка и реализация проектных решений.
3. Глава третья – Оценка и обоснование экономической эффективности проекта.

Каждая глава подразделяется на параграфы, в которых раскрывается составляющая часть общей темы главы. Во Введении и Заключение представлены те элементы работы, которые являются входящими (мы опирались на них в течение всего процесса работы) и выходящими

(полученные результаты в краткой форме, итог). В Списке литературы и используемых источников перечислены материалы, используемые в работе.

Новизна работы состоит в том, что предложенное автоматизированное решение по отношению к PLM системе Enovia, еще никем не было реализовано в доступной форме. Разработанный пример, на данный момент, является уникальным и, главное, работоспособным. В этом также состоит теоретическая и практическая значимость полученных результатов.

ГЛАВА 1. АНАЛИЗ КОМПАНИИ S&C И ОТДЕЛА PSS

1.1. Техничко-экономическая характеристика процесса обмена данными между PLM системой S&C и аутсорсинговыми компаниями

1.1.1. Характеристика компании S&C

Дадим характеристику компании. Компания S&C – это крупный поставщик оборудования и услуг для электроэнергетических систем. Компания ведет свою историю с 1911 года. Головной офис находится в Чикаго, США. Продукция S&C предназначена для обеспечения эффективной и надежной передачи электроэнергии. Часть производимой продукции предназначена для внедрения в электрические цепи. Другая часть используется как средства защиты электрического оборудования от поломок, а также как средство перенаправления электрической энергии от дефектного узла в функционирующие, чтобы не создавались простои. Кроме того, S&C предлагает широкий спектр инженерных, исследовательских и испытательных услуг для различных целей связанных с электрическими сетями. S&C предлагает свои услуги заказчикам по всему миру. Инженерные бюро и производственные цеха компании расположены в Чикаго (США), Франклин (США), Аламида (США), Орландо (США). Дочерние компании расположены в Торонто (Канада), Уэльсе (Великобритания), Мельбурне (Австралия), в также в Китае и Мексике. Компания имеет большую и сложную организационную структуру, в которую вовлечено около 5 тысяч сотрудников. Годовой доход компании варьируется от 500 миллионов до 1 миллиарда долларов США.

Примеры выпускаемого оборудования – коммутационное оборудование, контроллеры, накопители, счетчики, трансформаторы и др. Компания ведет полный цикл разработки и производства изделий. В основном компания реализует готовые изделия, но возникают случаи, когда в компанию обращаются заказчики с особыми требованиями к изделию, которые необходимо реализовать ограниченной партией или произвести штучные

модификации только для определенного клиента. Накладные затраты на изменение отлаженной работы собственного производства для выпуска «особой» продукции зачастую оказываются больше получаемой прибыли по данному заказу. Поэтому предприятию невыгодно из-за подобных заказов перенастраивать производство. Но и отказ от них значит потерять лояльность клиентов, что отрицательно скажется в долгосрочной перспективе на дальнейшей работе компании. Также возможно клиент захочет модифицировать уже установленное изделие, что не дает возможности его транспортировать. В таких случаях компания берет на себя заказ с «особыми» требованиями и перепоручает его компаниям-аутсорсерам. Таким образом, и заказчик получает то, что хотел, и компания получает выгоду. Для таких целей в компании S&C было создано подразделение Post Sale Services (PSS), на которое возложены обязанности по послепродажному обслуживанию клиентов и доработке готовой продукции под особые требования клиента.

1.1.2. Краткая характеристика подразделения Post Sale Services и его видов деятельности

Подразделение Post Sale Services (в дальнейшем будем писать просто PSS) занимается обслуживанием и поддержкой клиентов уже купивших и использующих продукцию компании. В их задачи входит: общение с клиентами, сбор информации о дефектах в изделиях, предоставления услуг по их доработке, а также сопутствующие услуги, связанные с установкой, покраской, модификацией и т.п. Примером таких услуг могут служить: изменения элементов крепления, дополнительная защита от вандализма, оградительные периметры, окраска и т.п.

Несмотря на то, что компания S&C имеет достаточно большой личный штат, данные работы в большинстве случаев выгодно выполнять не самой компании, а внешним подрядчикам. В каких-то случаях это компании, работающие на месте использования изделия, в каких-то это узко

специализирующиеся компании на определенных услугах (проектирование, изготовление, отливка, поставка и т.п.).

Подразделение состоит из проектных управляющих, проектировщиков, ассистентов, инженеров. Их основные задачи – проработка запросов (требований) поступающих от клиентов, поиск исполнителей, контроль за выполнением работ. Работа внутри организации ведется в нескольких корпоративных системах, основной является система управления жизненным циклом изделий (PLM – Product Lifecycle Management). Компания использует продукт «Enovia VPLM», разработанный компанией «Dassault Systèmes». Который предоставляет своим пользователям возможность создавать и поддерживать полноту, точность и актуальность составляемой документации, возможность управления спецификациями. В этой системе хранится вся инженерная документация: чертежи, технические задания, инструкции и др.

Для обработки запросов сотрудниками подразделения PSS им необходимо получить всю информацию по используемым изделиям, уточнить новые требования и разработать детальный план работ. Для определения сроков и стоимости работ, сотрудникам необходимо запросить эти данные у потенциальных исполнителей (поставщиков услуг – в дальнейшем будем внешних исполнителей называть поставщиками), для этого им необходимо передать информацию, полученную из Enovia PLM, а также информацию от клиента и после обработки потенциальными поставщиками собирать с них предложения и загружать обратно в систему PLM для дальнейшей работы.

1.1.3. Сущность задачи автоматизации процесса обмена данными

Сотрудники подразделения PSS регулярно предоставляет информацию из системы PLM поставщикам, а также получает данные от них и загружает в корпоративную систему. Данный процесс обмена требует от исполнителя внимательности, осведомленности о проектах, знание с какими поставщиками ведется работа в каждом из проектов и дополнительных знаний и умений по

работе с системой PLM. Наличие небольшого количества проектов, находящихся в одновременной работе у исполнителя, не вызывает особых проблем, но при увеличении нагрузки, увеличивается количество допускаемых ошибок, что ведет к потере информации в процессе обмена, потери контроля над исполнением, увеличение затрачиваемого времени на сам обмен, увеличением сроков реализации проекта и возможно срыв проекта. Также процесс обмена осложняется наличием документов большого размера и/или большого количества документов, подлежащих обмену, что делает невозможным рассылку данных документов через стандартные почтовые сервисы. Поэтому исполнителю необходимо совершать много дополнительных действий для их передачи, связанные с загрузкой документов в файлообменные сервисы, что в свою очередь влечет и проблемы с их безопасностью. Во избежание возможных ошибок, необходимо максимально автоматизировать текущий процесс обмена документами. Так как источником документов является Enovia, то идеальный конечный результат состоит в том, чтобы исполнитель работал только с ней, документы рассылались из нее и возвращались от поставщиков обратно, без дополнительных действий исполнителя.

Выводы по параграфу (актуальность задачи)

Решение проблемы компании S&C будет соответствовать актуальности, заявленной во введении данной бакалаврской работы, по следующим пунктам:

1. Обеспечение безопасности передаваемых исполнителям данных от несанкционированного доступа.
2. Снижение вероятности потери данных или их части в процессе обмена.
3. Упрощение работы с большими объемами пересылаемых данных.
4. Отсутствие необходимого функционала для обмена данными или высокая стоимость лицензии для каждого дополнительного пользователя в существующих системах.

1.2. Концептуальное моделирование процесса обмена данными

Процесс обмена данными с поставщиками входит в состав более комплексного процесса: «сбор предложений поставщиков». Данный процесс направлен на получение данных о готовности поставщика, сроках и стоимости работ подлежащих выполнению. Входными данными для процесса «обмена документами» является утвержденный список документов необходимых для рассылки, находящихся в система PLM, и список поставщиков, кому необходимо их направить. Результатам обмена должны стать документы от поставщиков с их предложениями загруженными в Enovia PLM. Опишем модель процесса как сейчас (AS IS) происходит обмен в компании S&C.

После утверждения списка документов подлежащих рассылки и списка поставщиков, начинается формирование пакета документов на отправку поставщикам. Исполнитель выгружает системы PLM необходимые документы и формирует папки с копиями документов для каждого поставщика в файлообменном сервисе DropBox. Процесс загрузки данных в DropBox осложняется наличием файлов больших размеров, что требует затрачивать больше времени на загрузку. После этого каждому поставщику необходимо отправить письмо с уникальной ссылкой на созданную папку с документами для него и дополнительной информацией по работам. В течение некоторого времени поставщики должны сформировать свои предложения по работам, оформив их в виде документа-предложения. Поставщик высылает сформированный документ письмом по электронной почте. После получения предложения, сотрудник отдела PSS загружает полученный документ-предложение в PLM систему. После сбора предложений от всех поставщиков или по истечению установленного времени, процесс обмена считается законченным, данные из файлообменного сервиса удаляются и к анализу полученных документов приступают аналитики отдела. Ниже в приложении 1 приведена DFD схема, иллюстрирующая движение данных в процессе обмена.

Основным и главным недостатком существующего подхода является человеческий фактор. Большое количество операций производится вручную, что увеличивает вероятность совершения ошибок сотрудниками. Ошибки могут быть как критическими, связанными с неправильным указанием адресатов или неправильной настройкой доступа к распространяемым данным, что влечет за собой проблемы с безопасностью. Так и незначительными, но на их исправление расходуется дополнительное время сотрудников. Процесс ожидания предложений от поставщиков занимает продолжительное время, в течение которого могут быть заменены люди, работающие по данному проекту, что приводит к потере контроля над процессом и требует дополнительных ресурсов на его восстановление. Резюмирование недостатков:

- безопасность данных;
- нарушение обратной связи;
- потеря контроля над процессом;
- высокие трудозатраты.

Все работы в рамках процесса обмена данными можно разбить на три логических этапа. Первый этап – это получение необходимых данных из системы PLM, сначала для выгрузки необходимо найти все документы, подлежащие рассылке, и сохранить их на локальный диск. Вторым этапом связан с обменом данными между поставщиками и компанией. Третьим этапом – это загрузка данных в систему PLM.

Сам процесс можно разбить на три подсистемы. Подсистема загрузки и выгрузки данных из PLM, подсистема обмена данными и подсистема взаимодействия с поставщиком.

Далее представим некоторую характеристическую информацию по производительности в отделе до внедрения автоматизированного решения. Для этого от сотрудников отдела была получена аналитическая информация по показателям, связанным или влияющим на процесс обмена и позволяющий подсчитать экономическую целесообразность проведения автоматизации системы. Вот какую информацию мы получили:

- среднее число проектов в год (СЧПГ) – 53;
- среднее число поставщиков в одном проекте (СЧП) – 11 человек;
- среднее время, затрачиваемое на одного поставщика (СВОД) – 5 часов;
- размеры отправляемых документов – от 500 Кб до 5 Гб;
- средний размер отправляемого документа (СРД) – 63 мб;
- количество документов, подлежащих обмену – не более 40 в одном проекте;
- среднее количество документов (СКД) – 6 шт.

Опираясь на средние показатели, можем подсчитать среднее затрачиваемое время одним сотрудником в год (СЗВГ) и определить трудозатраты:

$$\text{СЗВГ} = \text{СЧПГ} * \text{СЧП} * \text{СВОД} = 53 * 11 * 5 = 2915 \text{ часов на все проекты.}$$

Отсюда вычислим сколько рабочих дней уходит на работу с поставщиками:

Количество рабочих дней = $2915 / 8 = 364,375 \approx 364$. Получаем без одного дня целый год.

Теперь сформируем модель потока данных в формате DFD как это должно быть (ТОВЕ).

В приложении 2 показана обобщенная диаграмма потоков данных, отражающая общую схему работы системы после автоматизации. Ниже на рисунке в том же приложении показана декомпозиция предыдущей диаграммы, в которой добавлена сущность «Поставщик» и процесс работы поставщика с документами компании.

В модели ТОВЕ акцент делается на минимизацию участия сотрудника компании в процессе формирования пакетов документов и писем с уведомлениями поставщикам, основной его задачей является указание автоматизированной системе документов для выгрузки из Enovia PLM и отправки заранее выбранным поставщикам. По завершению обмена автоматизированная система уведомит сотрудника о получении документов от поставщика.

1.3. Постановка задачи разработки автоматизированного решения для процесса обмена данными

Компания S&C сделала заказ на разработку автоматизированного решения для упрощения процесса обмена документами с поставщиками и как следствие значительного сокращения трудозатрат и увеличения контроля над данным процессом.

Задача – разработать недорогое, безопасное и удобное решение, предоставляющее передавать документы из внутренней системы PLM поставщикам (услуг, товара, внешним исполнителям, партнерам и заказчикам) и загрузкой их предложений в эту систему. Основными критериями, которыми должна обладать система, являются простота и удобство в использовании, защищенность от вирусов и несанкционированного доступа к данным, надежность.

В разработке проект был поделен на три крупных этапа – сбор информации и планирование проекта, разработка программных модулей и интерфейсов, тестирование и ввод в эксплуатацию. Со стороны заказчика исполнителю были отправлены документы с перечислением всех бизнес требований к разрабатываемой системе, технические инструкции по нахождению и структуре данных в системе Enovia PLM.

1.3.1. Цель и назначение автоматизации процесса обмена данными

Основными целями ставится улучшения текущих показателей по сокращению времени на работу с данными, повышение их защищенности и полноты получаемой информации.

Назначение реализации работы по автоматизации обмена информацией может служить:

1. Автоматизация процесса выгрузки и загрузки данных в/из PLM системы.

2. Автоматизация рассылка уведомлений пользователям на почту о поступлении данных и изменении их статуса.

3. Удобный способ получения данных поставщиками и из загрузки.

4. Контроль доступа к данным.

1.3.2. Общая характеристика разрабатываемой системы для решения поставленной задачи

В системе, подлежащей разработке, определены два основных действующих лица, это «внутренний пользователь» сотрудник отдела PSS, и «внешний пользователь» представитель поставщика. Внутренний пользователь в системе PLM в специально разработанной форме должен сформировать список документов подлежащих обмену и сформировать список поставщиков, которым предназначается информация. Внешний пользователь получает уведомление по почте, с предоставленной специально для него уникальной ссылкой, по которой он может скачать документу и загрузить свое предложение.

Также установлены технических требования к системе:

1. Почтовые уведомления должны рассылаться с личного почтового сервера компании со специального почтового ящика.

2. Все получаемые данные от поставщиков должны проходить обязательную проверку корпоративным антивирусом.

3. Для каждого поставщика создается своя уникальная ссылка доступа к документам.

4. Должна существовать система блокировки созданных ссылок.

5. Ссылка блокируется автоматически после истечения заданного периода в настройках, а данные загружаются в систему PLM.

6. Должна существовать возможность загрузки данных раньше срока по решению сотрудника компании.

7. У сотрудника должна быть возможность просмотра данных, которые загрузил поставщик, до непосредственной загрузки в систему Enovia.

8. Уведомления на события в системе должны быть конфигурируемые.

9. Возможность просмотра ссылок, по которым ведется работа, через специальный веб-интерфейс.

1.4. Анализ существующих разработок и обоснование выбора технологии проектирования системы для интеграции с PLM Enovia

Для реализации обмена данными в PLM системе существует встроенный стандартный вариант обмена данными с внешними поставщиками. Но для его использования необходимо для каждого поставщика покупать дополнительно специальную лицензию. При большом количестве поставщиков общая стоимость оказывается очень большой. К тому же, данная лицензия позволяет произвести замену пользователя, т.е. поставщика, только один раз в три месяца, что не позволяет оперативно переключать лицензии между поставщиками и таким образом сэкономить на лицензиях.

Произведенный поиск подобных решений для данной системы не дал результатов. Система Enovia PLM – узкоспециализированная и ее конфигурируют под специфику и нужды компании, использующей ее, что затрудняет разработку универсального программного продукта, который бы можно было использовать «из коробки», т.е. без необходимости предварительного конфигурирования и доработки. Вероятно, существуют подобные разработки, но они все носят закрытый характер, получить о них информацию в открытых источниках затруднительно.

Для наглядности составим таблицу сравнения достоинств и недостатков, ручного обмена данными, стандартными средствами системы при покупке лицензий и разрабатываемой нами автоматизированной системы обмена.

Таблица 1.1. Сравнительная характеристика разработок

| | Обмен средствами Enovia PLM | Ручная работа с использованием DropBox | Дополнение к системе Enovia PLM |
|-------------|---|---|--|
| Достоинства | <ol style="list-style-type: none"> 1. Надежность. 2. Проверенность. 3. Известность пользователям. 4. Официальная поддержка. 5. Благодаря ограниченному доступу в систему, вероятность порчи документов и внесения несогласованных изменений в них резко снижается. | <ol style="list-style-type: none"> 1. Полный контроль над процессом. 2. Не необходимости дополнительных расходов, связанных с лицензиями. | <ol style="list-style-type: none"> 1. Экономичность 2. Простота в использовании. 3. Для работы с данными достаточно только специальной ссылки. 4. Настраиваемость режимов доступа для поставщиков. 5. Контроль исполнения поручений. 6. Хороший уровень информированности пользователей. 7. Конфигурируемое время жизни данных. |
| Недостатки | <ol style="list-style-type: none"> 1. Большая стоимость лицензии. 2. Сложность в использовании системой. 3. Сложность увеличения числа поставщиков. | <ol style="list-style-type: none"> 1. Потеря контроля 2. Проблемы с безопасностью информации. 3. Много ручной работы. | <ol style="list-style-type: none"> 1. Дополнительные расходы на поддержку системы самой организацией. |

Компания пришла к выводу, что затраты на разработку и содержание своего решения по обмену данными будет дешевле чем оплата лицензии на пользование встроенного функционала в Enovia PLM.

Руководством S&C было решено разработать собственное решение и интегрировать его с Enovia PLM, тем самым получить достаточно гибкую систему обмена, а не ограниченную по использованию. Таким образом, заявка точно будет исполнена, а S&C значительно расширяют свои возможности по выбору поставщиков и сможет работать с большим их количеством.

Выводы по параграфу (решение)

Проблемы, с которыми столкнулась компания S&C в работе с корпоративной PLM системой:

1. Существующая система документооборота с поставщиками, заказчиками и партнерами неудобна в плане привлечения к этому процессу нескольких человек из числа персонала компании на ручное формирование пакетов с документами и отправку каждого пакета адресатам по отдельности.

2. Система предполагает плату большой суммы за каждого нового участника этой системы.

3. Смена зарегистрированных участников может происходить не чаще одного раза в месяц.

Решением перечисленных проблем компания S&C видит в следующем: спроектировать и реализовать дополнительную интегрируемую в Enovia систему, которая обладала бы возможностью к генерации ссылок доступа к документам для внешних поставщиков, стала бы средством уменьшения объемов ручной работы посредством автоматизации процессов загрузки и выгрузки документов из PLM системы, рассылки уведомлений по электронной почте.

Выводы по главе 1

В процессе работы над первой главой бакалаврской работы было выполнено следующее:

1. Дано описание компании, ее деятельности и масштабах, для которой было необходимо разработать автоматизированное решение возникших проблем.

2. Дано описание конкретного отдела компании, для которого, в основном, разрабатывалось автоматизированное решение.

3. Разъяснена сущность задачи автоматизации с указанием решаемых проблем и их актуальности.

4. В рамках концептуального моделирования построены диаграммы потоков данных, отображающих процесс документооборота в компании до внедрения автоматизированного решения (модель AS IS) и после внедрения автоматизированного решения (модель TO BE).

5. Составлена таблица, в которой представлено сравнение существующих разработок с предложенной и реализованной нами.

Таким образом, дадим обобщенную формулировку решения: спроектировать и реализовать дополнительную интегрируемую в Enovia систему, которая обладала бы возможностью к генерации ссылок доступа к документам для нескольких внешних исполнителей одновременно, стала бы средством уменьшения объемов ручной работы посредством автоматизации процессов загрузки и выгрузки документов из PLM системы, рассылки уведомлений по электронной почте.

ГЛАВА 2. РАЗРАБОТКА И РЕАЛИЗАЦИЯ ПРОЕКТНОГО РЕШЕНИЯ

2.1. Логическое моделирование процесса обмена данными между PLM системой и поставщиками

2.1.1. Логическая модель процесса обмена данными и ее описание

Разработаем логическую модель для системы на основе построенной в первой главе модели TO BE. Данная модель приведена в приложении 1 данной работы.

Структурная модель «Как должно быть» в общем отражает процесс работы поставщиков услуг (исполнителей) с документами из PLM системы Eovia. Разобьем разрабатываемую систему, далее её будем называть «File Sharing System» (сокращенно FSS), на подсистемы, логические объекты и действующие лица. В системе выделяются три больших логических подсистемы, каждая из которых выполняет свой круг задач и обусловлена ограничениями, накладываемыми изначальными требованиями. Выделим три подсистемы: PLM Integration, Web Exchange, Data Sync. Рассмотрим каждую подсистему по отдельности.

PLM Integration – подсистема взаимодействия с PLM системой, ее главной задачей является взаимодействием с самой PLM системой: выгрузка и загрузка данных, предоставление пользовательских интерфейсов внутри PLM для удобной работы пользователя, для выгрузки данных, скрипты загрузки данных в систему PLM. Данная подсистема разрабатывается под конкретную версию используемой PLM системы и может быть адаптирована под другие продукты.

Web Exchange – подсистема, предоставляющая поставщикам удобный доступ к данным, которая всегда доступна через интернет и обеспечивает свободный доступ к скачиванию и загрузке. Для выполнения этих условий

данный сервис должен располагаться вне внутренней сети организации и быть независимым.

Data Sync – подсистема передачи данных между PLM Integration и Web Exchange.

Выделим логические объекты, которые будут использоваться внутри системы подсистемами. Для распространения и хранения данных одного набора документов введем понятия «Container», данная логическая сущность будет содержать в себе данные, подлежащие распространению, а также логические сущности «LoadBox», в которых будут содержаться данные, загруженные поставщиком, для каждого поставщика создается отдельная такая сущность в контейнере.

Выделим действующие лица и изобразим взаимодействие поставщиков и работников отдела Post Sale Services (возьмем одного project manager (PM)) с PLM системой с использованием Use Case Diagram методологии UML.

У нас есть три основных действующих лица (actors) – поставщик (supplier), PM (project manager), система (System). Они взаимодействуют с разрабатываемой нами системой «File Sharing System». Внутри разработанной системы они взаимодействуют с контейнерами, посредством которых происходит обмен данными между сотрудником S&C и поставщиком. Контейнер в диаграммах будем обозначать словом «Container». Диаграмма вариантов использования представлена в приложении 3.

Диаграмма вариантов использования полезна в качестве средства изображения взаимодействия сущностей (актеров) с разработанной системой. Для большей конкретизации взаимодействующих сущностей используется диаграмма модулей, классов, которая представляется в виде поименованных таблиц с указанием атрибутов и методов.

Используемые классы: Project manager (PM), Supplier (Поставщик), Container (Контейнер), Documents, PLM, LoadBox.

Краткое описание диаграммы (смотри приложение 3): PM и Supplier взаимодействуют с Container. Container по структуре состоит из LoadBox-ов, в

которые загружаются документы. Соответственно в LoadBox содержатся Documents, которые загружены из PLM или поставщиками через Container и LoadBox. Вместе с тем, PM может напрямую взаимодействовать с PLM в ручном режиме. От Container идут стрелки к PM и Supplier, так как во время работы с ним оба субъекта получают уведомления на электронную почту.

Далее работу с File Sharing System рассмотрим в динамике. Для этого воспользуемся еще одним инструментом UML – диаграммой последовательности (Sequence Diagram). Этот тип диаграмм позволяет подробно рассмотреть взаимодействие объектов предметной области по совершаемым действиям на каждом этапе в их строгой последовательности (смотри приложение 3).

Анализ деятельности вокруг системы завершим логической моделью данных. Логическая модель или ER-диаграмма (диаграмма сущность-связь) строится на основе диаграммы классов. ER-диаграмма нужна для отображения сущностей и связей между ними, важных для рассматриваемой системы и последующей реализации базы данных, а также для указания кратности каждого отношения (многие ко многим, один ко многим и др.). Для этого типа диаграммы характерны понятия первичный ключ (PK) и внешний ключ (FK). Они нужны для того, чтобы по минимальному числу атрибутов в таблице однозначно найти требуемый экземпляр сущности.

Полученная логическая модель представлена в приложении 3.

При реализации данной структуры в базе данных мы получим, что из PLM можно просмотреть документы, созданные контейнеры и сотрудников, которые загрузили те или иные документы и создали контейнеры. Из контейнера нам доступна информация об авторе (PM), адресате (Supplier), источнике отправки документов, LoadBox-ах и самих документах. Поставщики работают с документами только через контейнер.

2.1.2. Характеристика входной информации для работы подсистем File Sharing System

Формирование контейнера с документами происходит в системе Enovia. Таким образом, вся входная информация формируется с использованием подсистемы PLM Integration. Благодаря этому формирование контейнера возможно непосредственно из PLM.

Есть некоторые ограничения, наложенные на разработанное расширение к PLM, которые определяют количество и размеры загружаемых документов. Размеры отправляемых документов – от 500 Кб до 5 Гб. Количество документов, подлежащих обмену – не более 40 в одном проекте.

В Enovia хранится полный список всех проектов компании. Проекты, переданные на исполнение в отдел PSS, хранятся в специальной папке «PSS Projects». Чтобы просмотреть список документов по тому или иному проекту, нужно кликнуть на нужный проект, и тогда раскроется новая вкладка «Project Documents».

Для рассылки выбираем нужные документы, отмечая их галочками в чекбоксах слева. Отмеченные строки с документами отмечаются голубым. Для формирования самого контейнера в системе предусмотрена специальная форма, которая является частью реализованного модуля PLM Integration. Внутри системы эта операция называется Create Drawing Set. Перед нами возникает форма с полями для заполнения.

В этой форме вводим всю необходимую информацию для контейнера с документами – наименование контейнера, описание этого контейнера, выбираем электронные адреса поставщиков, еще раз подтверждаем ранее выбранные документы на отправку. В конце нажимаем на кнопку Send для рассылки контейнера по указанным адресам. Сформированный контейнер попадает в список уже существующих, ранее созданных контейнеров.

По нажатию на наименование контейнера раскрывается окно с полной информацией об этом контейнере. Также отображаются адреса тех, кому был

отправлен контейнер и указывается уникальная ссылка, по которой можно получить доступ к данному контейнеру через подсистему Web Exchange.

Кликнув на ссылку, можно перейти к содержимому контейнера, представленному в подсистеме Web Exchange. Документы, которые были выбраны в Enovia, успешно добавились в контейнер и стали доступны для пользования получателем письма с указанной ссылкой. Далее в приложении 8 «Контрольный пример и схема его реализации» представлены скриншоты описанных выше модальных окон.

2.1.3. Характеристика базы данных подсистемы Web Exchange

Предоставим структуру хранения данных для подсистемы Web Exchange. Данные представлены в виде JSON для хранения в документо-ориентированной базе данных. В проекте используется три вида коллекций документов: для контейнеров, для хранения уникальных ссылок, для хранения писем, подлежащих рассылке. Приведем структуру каждого из них в приложениях 4, 5 и 6 к данной бакалаврской работе.

Рассмотрим структуру документа каждой коллекции. Документ «Контейнер» состоит из трех основных элементов, это уникальный идентификатор, список, описывающий документы, подлежащие рассылке, в каждом из них описывается id, author, internalPath, expiredAt, subject, message и список LoadBox, каждый из которых содержит свой уникальный идентификатор, имя, размер и статус.

JSON-структура уникальных ссылок описывает следующие поля: уникальный идентификатор ссылки, уникальный номер доступа (случайно сгенерированная строка), активна ли данная ссылка, время создания и время последнего доступа по ней. Также содержится информация о том, кому доступна данная ссылка, и к какому контейнеру и LoadBox она предоставляет доступ.

JSON-структура письма содержит уникальный идентификатор, заголовок, имя шаблона письма, в который будет подставлена информация из поля данных «data» данного документа. В данный раздел может входить различная информация в зависимости от шаблона для уведомлений о предоставлении информации – это список распространяемых документов, их суммарный размер, имя пользователя, от которого происходит рассылка, комментарии и уникальная ссылка для доступа. Раздел системной информации, в который включена информация о статусе письма, времени, когда было создано и времени успешной или не успешной отправки. Также указывается список адресатов.

Данный подход к организации хранения был выбран для удобства и наглядности информации, что упрощает администрирование и поддержку базы данных.

2.1.4. Характеристика результатной информации подсистемы PLM Integration

В результате разработки и реализации проектного решения мы получили в том числе подсистему PLM Integration к системе Enovia. Эта подсистема позволила облегчить работу внутри системы Enovia с рассылаемыми поставщикам и получаемыми от поставщиков документами.

В данной подсистеме предусмотрено отображение созданных контейнеров. Для каждого контейнера отображаются его статусы – empty (пустой) и full (заполненный), sent (отправлен) и create (создан) – для удобства отслеживания их состояния отображаются в общей таблице без непосредственного перехода к содержимому контейнера.

PLM Integration предоставляет возможность для просмотра полного перечня загруженных документов через определенный интерфейс в Enovia. Документы, также как и контейнеры, отображаются в общей таблице, в которой помимо их названия выведена информация по объему, автору, дате загрузки.

Выведенная информация позволяет быстро найти нужный документ без ручного перебора всех наличных документов.

Документы от поставщиков загружаются в PLM Enovia в двух режимах – ручном (на тот случай, если работа с поставщиками завершится раньше оговоренного срока) и автоматическом. Автоматическая загрузка документов из контейнера происходит по истечению срока его жизни. В этом случае подсистемой деактивируются ссылки, документы загружаются в PLM, поставщикам и администратору на электронную почту приходят уведомления об автоматическом срабатывании подсистемы.

Выводы по параграфу

В результате работы над логическим представлением разработанной системы, мы выполнили следующее:

1. Разработали несколько разновидностей моделей для четкого представления предметной области: диаграмму прецедентов, классов, последовательностей и логическую модель.
2. Дали характеристику входной информации, базам данных и результатной информации.
3. Пошагово описали формирование контейнера с документами из PLM.

2.2. Физическое моделирование File Sharing System

2.2.1. Выбор архитектуры File Sharing System

Для выбора архитектуры и используемого программного обеспечения сформулируем общие правила:

- архитектура должна быть масштабируема и легко модифицироваться для использования под разными платформами;
- по возможности используемое программное обеспечение (ПО) должно быть бесплатным;

— выбранное ПО должно быть популярным, что упростит разработку и поддержку продукта в дальнейшем;

— использоваться должно только проверенное ПО в других проектах, что позволит избежать потенциальных проблем.

Каждая разрабатываемая подсистема имеет свои особенности, поэтому рассмотрим выбор архитектуры для каждой из них в отдельности.

PLM Integration – данная подсистема разрабатывается и интегрируется в систему Enovia PLM, поэтому для ее реализации будет использоваться та же архитектура, что и для основной PLM. В Enovia PLM используется клиент-серверный подход. Сама система реализована при помощи веб-сервера Apache Tomcat. Разработка под него ведется путем написания сервлетов (JSP) на языке Java. Генерируемый код веб-страничек оптимизирован под Internet Explorer, отображение под другими браузерами может осуществляться некорректно. Подсистема будет состоять из дополнительно разработанных сервлетов (JSP), которые будут отвечать за отображение форм и дополнительных библиотек на Java, которые позволят сервлетам взаимодействовать с подсистемой передачи данных Data Sync.

Data Sync – подсистема, отвечающая за обмен данными между Web Exchange и PLM Integration. С ее помощью происходит перенос данных из одной подсистемы в другую. Перенос данных это всегда определенная последовательность действий, которую необходимо выполнить. Для ее выполнения лучше всего подходит система с определением некоторой последовательности задач, подлежащих выполнению, за каждой из них закрепляется свой обработчик. Для описания последовательности задач, подлежащих выполнению, была выбрана методология BPMN (Business Process Model and Notation). Так как обработчики задач необходимо запускать на разных компьютерах, использование клиент-серверной архитектуры позволит решить эту проблему. Серверная часть предоставляется Workflow REST API сервисом, для доступа к задачам процессов и для хранения их состояний. Такой подход позволяет реализовать платформу-независимую подсистему, в которой

обработчики задач могут быть реализованы под разными операционными системами и платформами.

Web Exchange – подсистема, представляющая собой точку обмена документами для поставщиков. Так как доступ к данным должен быть максимально простым и привычным для поставщиков, использование протоколов FTP, SFTP, SSH, требующих установки дополнительного программного обеспечения, усложняет использование. В данном контексте самое простое для использования – это создание веб-сервиса, работающего по протоколу HTTPS, который позволяет скачивать и загружать данные, предоставив уникальную ссылку поставщику. Так как сервис не представляет из себя большого веб-портала с данными, для него рациональнее использовать подход SPA (single page application). SPA – позволяет создавать страницы, которые обрабатывают логику на стороне клиента, получение необходимых данных происходит методом обращения к REST API сервисам. Для раздачи контента используется сервер Nginx. Реализацию Container REST API, который предоставляет информацию о контейнерах и позволяет скачивать и загружать документы, целесообразнее выполнять с использованием Node.js, это связано с небольшим объемом кода и быстрой скоростью разработки. Данные по контейнеру, описывающие его структуру и содержащиеся в нем информацию, описывающую документы, удобнее всего хранить в документо-ориентированной СУБД, одной из популярных и бесплатных баз данных такого типа является MongoDB. Так как суммарный размер хранимых документов (файлов) может быть большим, подсистема должна уметь хранить и оперативно выдавать большие объемы данных. Для этого лучше всего подходит хранилище данных от Amazon S3, которое гарантирует сохранность данных, предоставляет неограниченное пространство и достаточно быстрые каналы связи для доступа к данным.

2.2.2. Функциональная схема подсистем File Sharing System

Для изображения функциональной схемы проекта используем блок-схемы. Для удобства понимания для каждой подсистемы составим отдельные блок-схемы. Блок-схемы помещены в приложение 7 данной работы.

Блок-схемы описывают следующие подсистемы:

Для PLM Integration:

1. Формирование данных для рассылки:
 - a. выбор проекта и формирование списка документов;
 - b. введение адресов получателей документов.
2. Передача данных:
 - a. выгрузка документов из базы данных Enovia на жесткий диск;
 - b. формирование описания данных, подлежащих рассылке;
 - c. передача информации подсистеме Data Sync.
3. Загрузка данных в Enovia:
 - a. проверка на наличие данных, подлежащих загрузке;
 - b. загрузка данных в Enovia.

Для подсистемы Web Exchange, функции работы с поставщиками:

- переход по уникальной ссылке;
- просмотр списка файлов;
- скачивание файлов;
- работа с файлами;
- загрузка файлов.

Для подсистемы Data Sync:

- формирование контейнера во внешней подсистеме Web Exchange;
- загрузка документов с жесткого диска в сервис Amazon S3;
- рассылка уведомлений пользователям;
- проверка передаваемых документов антивирусом;
- получение данных из Web Exchange и передача их в PLM Integration.

Перейдем к описанию получившегося диалога между субъектами и подсистемами File Sharing System.

На графе для Web Exchange (который также можно посмотреть в приложении 3) в центре указано «Main window», что подразумевает главное окно веб-сайта, с которым пользователь взаимодействует и к которому постоянно будет возвращаться, так как каждое действие сопровождается возникающими модальными окнами. Например, на главном окне отображается список документов. По отношению к документам возможно действие «Download All». Кликнув на соответствующую кнопку, всплывает диалоговое окно, на котором нужно подтвердить свое выбранное действие. После подтверждения окно исчезает, мы возвращаемся к главному окну и параллельно видим, что идет загрузка всех документов. С остальными действиями точно также.

Граф для подсистемы PLM Integration демонстрирует последовательность действий внутри Enovia. Подробнее указанные шаги описаны в приложении 8 «Контрольный пример реализации проекта и его описание».

2.2.3. Структурная схема File Sharing System

В логической модели мы разбили подсистему на три основных подсистемы: PLM Integration, Data Sync, Web Exchange. Рассмотрим каждую.

PLM Integration. Разобьем данную подсистему на логические модули, каждый из которых будет отвечать за определенный функционал. PLMUI – визуальный модуль, который будет отвечать за работу с пользователем, за ввод и отображение информации. Данный модуль будет взаимодействовать с кодом основной PLM для визуализации и получения данных из нее. Модуль Workflow Rest Api Client Library реализовывает взаимодействие с сервисом Workflow, входящим в состав подсистемы DataSync, реализация логики работы протокола обмена данными поверх HTTPS и предоставление данных в удобном виде.

Модуль Loader Library содержит функционал для загрузки и выгрузки файлов из Enovia PLM с учетом специфики данной системы.

DataSync. Все модули данной системы можно разбить на две группы. Модули, организующие работу workflow процессов, и модули, выполняющие задачи (tasks) в рамках процесса. Workflow Server (WS) выполняет управленческие функции над процессами, обеспечивает сохранение состояния процессов и предоставление информации по ним. Workflow Client Library (WCL) реализует взаимодействие с сервером и предоставляет информацию в удобном виде. Модули выполнения задач в процессах: Antivirus Task (AT), Upload Task (UT), Download Task (DT), Import to PLM Task (IT). Все модули выполнения задач взаимодействуют с WS через WCL, получают назначенные задачи для выполнения с данными, после завершения выполнения передают результат обратно серверу. AT – выполняет проверку полученных файлов от поставщиков корпоративным антивирусом, UT – выполняет загрузку данных с указанного места в корпоративном файловом сервере в хранилище данных S3, DT – получает данные из S3 и сохраняет их в указанное место на внутреннем корпоративном файловом сервере, IM – загружает данные с корпоративного файлового сервера в указанное расположение системы Enovia PLM.

Модули подсистемы WebExchange разделяются на две группы. Модули, обеспечивающие работу Container Rest Api Server'a (CRAS), и модули, взаимодействующие с ним и предоставляющие визуальный функционал. В CRAS входят модули, отвечающие за работу интерфейсной части REST ссервиса через HTTPS протокол (Restify); модуль авторизации - Permission; модуль, осуществляющий взаимодействие с базой данных Mongoose; модуль, отвечающий за логику работы сервиса Logic. Два визуализационных модуля SPA Ext User – главное рабочее пространство поставщика и SPA Admin – модуль статистики и управления системой.

2.2.4. Описание программных модулей File Sharing System

Количество кода в программных модулях достаточно велико, система разбита на простые подмодули, выполняющие определенные задачи, и не имеет сложных алгоритмов, представляющих необходимость в визуализации.

На рисунке на блок-схемах представлено описание двух программных модулей. Блок-схема 1 описывает процесс создания контейнера для загрузки в него документов для дальнейшей рассылки поставщикам. Это модуль Loader Library подсистемы PLM Integration. Данный модуль предоставляет уровень абстракции над взаимодействием с системой Data Sync, что позволяет в удобной форме производить взаимодействие. Блок-схема 2 демонстрирует алгоритм функционирования модуля Upload Task в подсистеме Data Sync. Данный модуль запускается с определенной периодичностью, производит загрузку файлов и завершается. Аналогичным образом работает модель Download Task. Только в отличие от Upload Task он осуществляет скачивание файлов. Различие состоит в одном блоке на блок-схеме. Вместо загрузки происходит скачивание. Схема работы Antivirus Task тоже очень похожа, различие заключается в запуске функции проверки вместо загрузки файлов. Реализация модулей не содержит в себе сложных алгоритмов. Это сделано для того, чтобы не перегружать систему и сделать ее простой для понимания и поддержки.

Модуль Logic подсистемы WebExchange представляет уровень абстракции над моделью базы данных и содержит в себе бизнес логику работы с данными из базы. Определяет доступность действий при определенных состояниях объектов и доступность выполнения этих действий. Модели или схемы объектов базы данных описаны отдельными классами используя библиотеку Mongoose. С ее помощью предоставляется удобный способ доступа к данным в базе MongoDB.

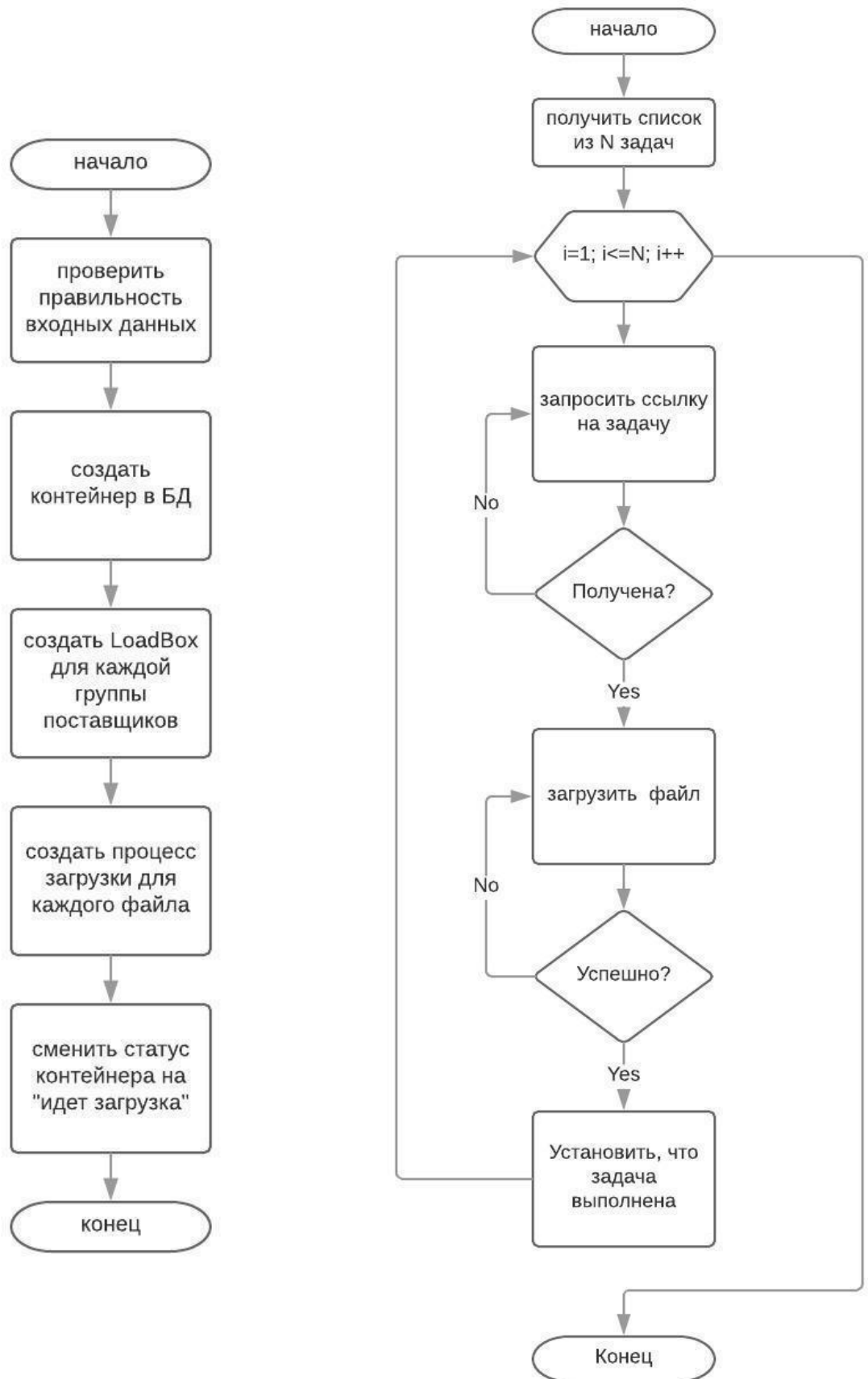


Рис.2.1. Блок-схемы модуля Loader Library для создания контейнера и последующей загрузки в него документов и модуля Upload Task

Выводы по параграфу

В данном параграфе дается описание разработанной системы с точки зрения ее физической реализации. Дано описание структурной и функциональной схем File Sharing System. Представлены диалоги project manager и supplier (поставщика) с PLM системой. Дано описание программных модулей с использованием блок-схем.

2.3. Технологическое обеспечение процесса обмена данными

2.3.1. Организация технологии сбора, передачи, обработки и выдачи информации

В процессе деятельности компании S&C значительное место занимает разнообразная документация – технические задания, чертежи, инструкции, отчеты и др. Без такого рода документов было бы сложно соблюдать предписания по производству электрического оборудования, всевозможных деталей. Отдел качества, например, только по официальным документам проводит проверку производимой продукции. Грамотно задокументированная и доступная для сотрудников S&C информация необходима для бесперебойного производства. С этими целями и была внедрена в процесс документооборота PLM система Enovia.

Задокументированная информация загружается в систему, только после тщательной проверки на соответствие требованиям по оформлению документов.

Рассмотрим процесс трансформирования информации в данной компании. У S&C есть крупные заказчики, для которых производится большая часть от всей продукции. От них компания получает документы с требованиями к производимому продукту. На этих требованиях и массе нормативной документации строится массовое производство в цехах S&C. С мелкими заказами компания также работает, как уже говорилось ранее. Для работы с

такими заказами предусмотрен неоднократно упоминаемый в тексте отдел PSS. В этом случае также работа начинается с требований заказчика.

Поток информации в компании следующий:

1. Получение информации от заказчиков в виде технических заданий, чертежей, требований.
2. Соотнесение требований заказчиков с обязательными нормативными документами, по которым обязана работать компания.
3. Переработка информации от заказчиков во внутренние технические задания, инструкции.
4. Разработка собственных чертежей и согласование их с заказчиками.
5. Приведение документации к общему виду для удобства работы с ними в дальнейшем (документация, как правило, составляется с учетом возможных будущих изменений).
6. Загрузка подготовленных документов в Enovia PLM.
7. При необходимости внесения изменений в существующую документацию, Enovia предоставляет инструментарий для поддержки актуальности документации внутри самой системы без предварительной выгрузки.
8. При получении небольших заказов на продукцию с «особыми» требованиями (не реализуемыми в массовом производстве) формируется задание на выполнение заказа отделом Post Sale Services (PSS).
9. Сотрудники PSS обрабатывают запрос и подбирают потенциальных исполнителей (поставщиков) из имеющейся базы.
10. Затем сотрудники отдела PSS на основе полученного заказа формируют рассылку, в которой указывается список необходимой документации из имеющейся и хранящейся в PLM, и список поставщиков.
11. При формировании рассылки в расширении к Enovia создается контейнер, в который из PLM загружаются указанные в рассылке документы.
12. Поставщики работают с этим контейнером, оригиналы документов из которого в течение всей их работы остаются неизменными, так как внесение в

них изменений извне ограничено (есть возможность только загружать себе копии документов и работать с ними).

13. Работа с информацией из контейнеров завершается загрузкой поставщиками своих ответных документов.

14. Далее сотрудником S&C просматривается загруженная документация.

15. После проверки документов сотрудником осуществляется их загрузка в Enovia с последующей деактивацией контейнера и ссылок на него.

2.3.2. Схема технологического процесса сбора, передачи, обработки и выдачи информации

Процесс работы с информацией, описанный в предыдущем пункте 2.3.1, представим в виде схемы технологического процесса. Сам процесс разделяется на три стадии. В первой происходит обработка полученной заявки, определения необходимых данных для проведения анализа и их нахождения в системе, получение доступа к необходимым данным, определения действительно ли данная заявка выполнима организацией. Как вся необходимая информация собрана, происходит переход в второй стадии, где определяется виды работ и их исполнители. Формируются рассылки с запросами цен и сроков исполнения. Третья стадия начинается, когда поставщики (потенциальные исполнители) проанализируют полученные ими документы и сформируют свои предложение по запрашиваемым работам. До наступления данной стадии идет конкретизация требований со стороны исполнителей. На это выделяется обычно несколько месяцев, чтобы провести полноценный и равноправный отбор. В третьей стадии полученные данные во всех поставщиков загружаются в систему, по ним выбирается исполнитель.

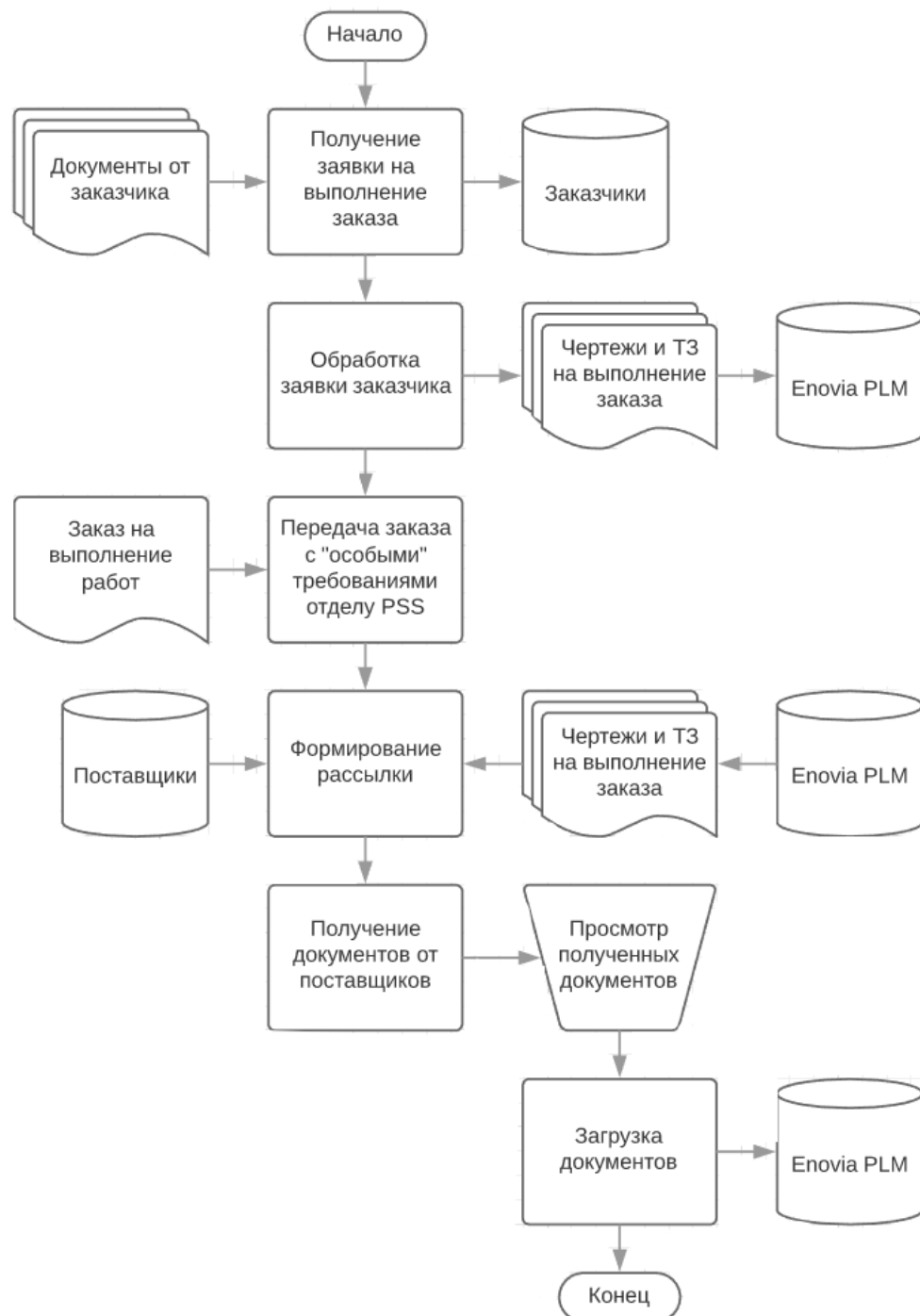


Рис.2.2. Схема технологического процесса сбора, передачи, обработки и выдачи информации в компании S&C

Выводы по параграфу

В данном параграфе дано словесное и графическое описание организации технологии сбора, передачи, обработки и выдачи информации в компании S&C с использованием разработанной File Sharing System. Цепочка работы с

информацией начинается с передачи заказчиком заявки на выполнение заказа, а завершается процесс загрузкой задокументированных данных в PLM систему Enovia.

2.4. Контрольный пример реализации File Sharing System и ее описание

Представим реализацию проекта в подсистемах Web Exchange (интерфейс веб-сайта, с которым работают поставщики и администратор) и PLM Integration (подсистема, интегрированная в Enovia, доступная только сотрудникам S&C). Подробное описание работы со скриншотами представлено в приложении 8.

Ручное создание контейнера в Web Exchange. Работа с File Sharing System начинается с входа сотрудника отдела PSS в систему под учеткой администратора. Для этого ему нужно ввести логин, пароль.

Далее администратор должен сформировать контейнер для отправки. На первом шаге выбираются документы для отправки, потом отмечаются электронные адреса поставщиков. Когда контейнер полностью будет сформирован, можно нажимать на кнопку Send.

После нажатия на кнопку Send системой автоматически генерируются уникальные ссылки, которые отображаются в отправленных письмах. Клик по кнопке View открывает содержимое контейнера для работы с ним.

Работа с содержимым контейнера осуществляется следующим образом. Поставщик получает доступ к контейнеру по ссылке. В пространстве контейнера поставщику доступны документы, которые были загружены администратором системы. Эти документы можно либо все сразу скачать себе на компьютер, либо скачивать их по отдельности. Удалять загруженные администратором документы поставщик не может. Свои документы поставщик может загрузить либо кнопкой Add files, либо перетаскиванием документов в область окна (предусмотрена функция Drag and Drop).

По истечению срока жизни контейнера происходит автоматическая деактивация ссылки на него и загрузка документов в PLM Enovia. Сотруднику отдела PSS, который занимался формированием данного контейнера, и поставщикам, которые работали с содержимым этого контейнера, на электронную почту придут сообщения. Поставщики получают уведомление, внешний вид которого показан на рисунке ниже.

В системе помимо представленных окон предусмотрена еще масса модальных окон с уведомлениями о, например, том, что загружаемый файл не доступен к загрузке, так как он уже кем-то был удален. Такое может случиться, когда с одним контейнером работает сразу несколько человек, а страница не успела перезагрузиться. Перед удалением выбранных файлов возникает окно для подтверждения удаления перечисленных файлов и др.

PLM Integration предусмотрена для удобства работы с PLM и ее интерфейс выполнен в том же стиле, что и сама PLM Enovia. Рассмотрим работу с данной подсистемой.

В Enovia хранится полный список всех проектов компании. Проекты, переданные на исполнение в отдел PSS, хранятся в специальной папке PSS Projects.

Кликнув по какому-нибудь проекту из списка, перейдем к перечню документов, которые используются в рамках выбранного проекта.

Для формирования рассылки выбираем нужные документы (строки с этими документами отмечаются голубым).

Дальше переходим к формированию самого контейнера. Данная форма является частью реализованного модуля PLM Integration. Внутри системы эта операция называется Create Drawing Set. Перед нами возникает форма с полями для заполнения.

В этой форме вводим всю необходимую информацию для контейнера с документами – наименование контейнера, описание этого контейнера, выбираем электронные адреса поставщиков, еще раз подтверждаем ранее

выбранные документы на отправку. В итоге остается нажать на кнопку Send для рассылки контейнера по указанным адресам.

Сформированный контейнер попадает в список уже существующих, ранее созданных контейнеров.

По нажатию на наименование контейнера раскрывается окно с полной информацией об этом контейнере. Также отображаются адреса тех, кому был отправлен контейнер и указывается уникальная ссылка, по которой можно получить доступ к данному контейнеру через подсистему Web Exchange.

Кликнув на ссылку, можно перейти к содержимому контейнера, представленному уже в подсистеме Web Exchange.

Выводы по параграфу

В параграфе пошагово описан алгоритм работы администратора (сотрудника отдела PSS) и поставщиков с разработанным расширением к PLM системе Enovia.

Выводы по главе 2

В результате работы над второй главой мы представили следующие результаты:

1. Логическую модель предметной области с ее подробным описанием.
2. Функциональную структуру разработанной системы.
3. Пошаговое описание технологии получения, обработки, хранения и выдачи информации с использованием автоматизированного решения.
4. Технологическую схему процесса работы с информацией.
5. Пошаговое описание работы подсистем Web Exchange и PLM Integration, из которых состоит File Sharing System.

ГЛАВА 3. ОЦЕНКА И ОБОСНОВАНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ РАЗРАБОТКИ FILE SHARING SYSTEM

3.1. Выбор и обоснование методики расчета экономической эффективности

Экономическую эффективность будем рассчитывать по следующей методике: рассчитаем прямую эффективность от внедрения информационной системы по сравнению с существующей в организации до внедрения автоматизированного решения.

В таком случае описание экономической эффективности базируется на сравнении технологических процессов до автоматизации и после внедрения автоматизированного решения; анализе затрат на проектирование, разработку и реализацию проекта.

Данная методика расчета хороша тем, что она подразумевает работу с реальными цифрами по основным производственным показателям, таким как: затраты по трудоемкости и стоимости расходного и составляющего материала.

В общем, чтобы точнее рассчитать экономическую эффективность, нужно учитывать два ее типа:

1. Косвенный эффект – возникает как следствие внедрения каких-либо нововведений в бизнес-процесс. Это, например, рост прибыли, уменьшение затрат на материалы и сырье, снижение брака на производстве и др.

2. Прямой эффект – возникает непосредственно от внедрения нововведений в бизнес-процесс, проявляется практически сразу же после начала функционирования нововведений. Характеризуется снижением трудовых и стоимостных показателей за счет уменьшения затрачиваемого времени на обработку и получение данных, сокращение трудоемкости работ, увеличения достоверности и точности информации.

Мы при подсчете экономической эффективности воспользуемся только показателями прямого эффекта. Во-первых, мы располагаем данной

информацией, во-вторых, косвенный эффект сложнее поддается подсчету и регистрированию (многие аспекты косвенного эффекта остаются тайной организации).

3.2. Расчет показателей экономической эффективности File Sharing System

Для расчета экономической эффективности от внедрения автоматизированного решения рассмотрим показатели трудовых и стоимостных затрат. Для этого от сотрудников отдела Post Sale Services (PSS) была получена аналитическая информация по показателям, связанным или влияющим на процесс обмена и позволяющий подсчитать экономическую целесообразность проведения автоматизации системы. На основе полученных данных будем производить расчеты необходимые расчеты. Вот какими данными об отделе мы располагаем:

- среднее число проектов в год (СЧПГ) – 53;
- среднее число поставщиков в одном проекте (СЧП) – 11 человек;
- среднее время, затрачиваемое на одного поставщика (СВОД) – 5 часов (до автоматизации);
- размеры отправляемых документов – от 500 Кб до 5 Гб;
- средний размер отправляемого документа (СРД) – 63 мб;
- количество документов, подлежащих обмену – не более 40 в одном проекте;
- среднее количество документов (СКД) – 6;
- стоимость лицензии на пользование PLM 15 000 \$;
- за каждого нового пользователя вне компании S&C приходится сверх выплачивать 0,2 % от общей стоимости лицензии, т.е. 30 \$;
- помимо единовременной выплаты за PLM компании каждый год приходится выплачивать 3 % от общей стоимости системы за продление лицензии, т.е. 450 \$;

— средняя заработная плата сотрудника отдела PSS в год составляет 70 000 \$.

После внедрения автоматизированного решения значения относительно времени выполнения необходимого объема работ и численности обслуживающего персонала предполагаются что снизятся до:

— среднее время, затрачиваемое на одного поставщика (СВОД) – 2,5 часа;

— на обслуживание системы выделен один человек.

Приступим к расчетам. Опираясь на средние показатели, можем подсчитать среднее затрачиваемое время одним сотрудником в год (СЗВГ) и определить трудозатраты до автоматизации:

$$T_0 = \text{СЧПГ} * \text{СЧП} * \text{СВОД} = 53 * 11 * 5 = 2915 \text{ часов на все проекты.}$$

Отсюда вычислим сколько рабочих дней уходит на работу с поставщиками:

$$\text{Количество рабочих дней (КРД)} = 2915 / 8 = 364,375 \approx 364.$$

Если рассчитать из расчета 5-дневной рабочей недели (без учета праздничных дней), то получим в год 104 выходных дня, то есть $364 - 104 = 260$ рабочих дней за год.

Теперь рассчитаем трудозатраты после внедрения автоматизированного решения (к сокращениям добавим индекс «А» - после автоматизации):

$$T_1 = 53 * 11 * 2,5 = 1457,5 \text{ часов на все проекты.}$$

$$\text{КРД}_A = 1457,5 / 8 = 182 \text{ дня.}$$

Получили выигрыш в 182 рабочих дня от результата до автоматизации. Рассмотрим финансовые затраты до и после проведения автоматизации за год.

До проведения автоматизации:

1. Стоимость PLM системы Eovia составляет 15000 \$, эту стоимость компания выплатила единовременно.

2. Расчеты затраченного времени на обслуживание системы демонстрируют необходимость компании в администраторе, который бы работал с данной системой, то есть компании нужно выплачивать заработную

плату в размере 70000 \$ в год. Если учесть, что на обслуживание системы нужно затратить 364 дня, то получаем сумму $(70000/260)*364 = 98000\$$.

3. Каждый год приходится выплачивать 450 \$ за продление лицензии на пользование PLM Enovia.

4. Добавление к системе нового пользователя составляет 30 \$, в месяц можно подключать только одного нового пользователя, соответственно, за год организация потратит $12*30 \$ = 360 \$$.

Рассчитаем суммарные затраты:

- Разовые затраты: 15000 \$;
- Постоянные затраты в год: $98000 \$ + 450 \$ + 360 \$ = 98810 \$$.

После проведения автоматизации:

1. Разработка автоматизированного решения обошлась в 35500 \$.
2. Затраты на техническое обслуживание системы (системного администратора, сервера) в год составляет 5400 \$ в год.
3. Затраты на выплату заработной платы сотруднику за поддержку системы составляет 49000 \$.
4. Плата за продление лицензии PLM Enovia остается в размере 450\$ в год.

Рассчитаем суммарные затраты:

- Разовые затраты: 35500 \$;
- Постоянные затраты в год: $49000 \$ + 5400 \$ + 450 \$ = 54850 \$$.

Перейдем к расчету трудовых и стоимостных показателей. Для их расчета используем следующие формулы:

Таблица 3.1. Формулы расчета затрат

| | |
|--------------------------------------|---|
| $T = T_0 - T_1$ | абсолютное снижение трудовых затрат |
| $K_T = \frac{\Delta T}{T_0} * 100\%$ | коэффициент относительного снижения трудовых затрат |

Продолжение таблицы 3.1.

| | |
|--------------------------------------|--|
| $Y_T = \frac{T_0}{T_1}$ | индекс снижения трудовых затрат |
| $C = C_0 - C_1$ | абсолютное снижение стоимостных затрат |
| $K_C = \frac{\Delta C}{C_0} * 100\%$ | коэффициент относительного снижения стоимостных затрат |
| $Y_C = \frac{C_0}{C_1}$ | индекс снижения стоимостных затрат |
| $T_{OK} = \frac{K_{\Pi}}{\Delta C}$ | срок окупаемости затрат на внедрение проекта |

Подставим найденные значения в формулы выше. Значения трудовых показателей следующие:

$$T = 2915 - 2040,5 = 874,5 \text{ часов}$$

$$K_T = \frac{874,5}{2915} * 100\% = 30\%$$

$$Y_T = \frac{2915}{2040,5} = 1,43$$

Значения стоимостных показателей:

$$C = C_0 - C_1 = 98810 - 54850 = 43960 \$$$

$$K_C = \frac{\Delta C}{C_0} * 100\% = \frac{43960}{98810} * 100\% = 44\%$$

$$Y_C = \frac{C_0}{C_1} = \frac{98810}{54850} = 1,8$$

Таблица 3.2. Показатели эффективности автоматизации

| | Затраты | | Абсолютное изменение затрат | Коэффициент изменения затрат | Индекс изменения затрат |
|--------------|------------------|---------------------|-----------------------------|------------------------------|-------------------------|
| | До автоматизации | После автоматизации | | | |
| Трудоемкость | 2915 ч. | 2040,5 ч. | 874,5 ч. | 30 % | 1,43 |
| Стоимость | 98810 \$ | 54850\$ | 43960\$ | 44 % | 1,8 |

Рассчитаем срок окупаемости затрат на внедрение.

$$T_{OK} = \frac{K_{П}}{\Delta C} = \frac{35500 \$}{43960 \$} = 0,8 \text{ года} \approx 9 \text{ месяцев.}$$

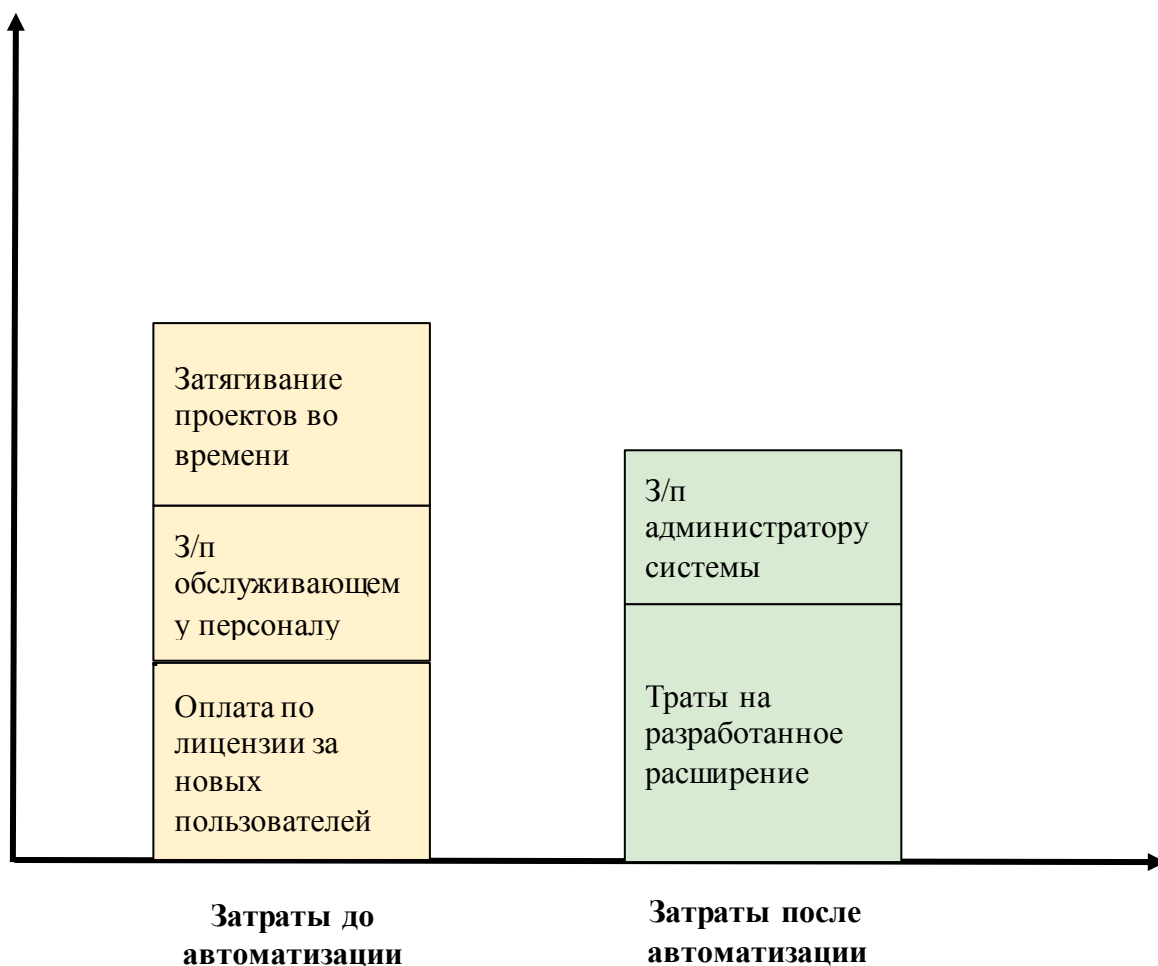


Рис.3.1. Графическое представление затрат до автоматизации и после проведения автоматизации

Выводы по главе 3

В данной главе проводится расчет экономической эффективности внедренного решения, дается оценка и обоснование экономической эффективности. Для подсчетов используется методика расчета прямой эффективности от внедрения информационной системы.

Основой для проделанных расчетов стали трудовые и стоимостные показатели.

Индексы изменения затрат трудоемкости и стоимости показывают значения соотношения между затратами до и после проведения автоматизации. По этому индексу можно судить о динамике изменения показателей. Например, значение единица будет говорить о том, что затраты остались на прежнем уровне. В описанном же случае значение больше единицы, что свидетельствует о снижении затрат.

Внедрение автоматизированного решения позволило снизить затраты по трудоемкости на 30 %. Это связано с тем, что внедренное решение стало причиной уменьшения времени на выполнение ряда операций, которые в базовом варианте выполнялись вручную. Отсюда вытекает еще одна причина: автоматизация привела к частичному снижению влияния человеческого фактора на процесс документообмена, что помимо экономии времени обеспечивает бесперебойную работу системы за счет внедрения дополнительных средств защиты от ошибочных действий со стороны пользователей. Финансовые затраты снизились на 44,9 %. Снижение затрат связано, в основном, с тем, что компании больше не нужно платить за новых пользователей и сокращение трудозатрат работников. Срок окупаемости затрат на внедрение проекта также небольшой для такой крупной компании, как S&C. Проект с финансовой и трудозатратной точек зрения можно считать успешным.

ЗАКЛЮЧЕНИЕ

Итогом написания данной бакалаврской работы является внедренная и успешно функционирующая система-расширение к PLM Enovia. Определяющим результатом является положительный отзыв заказчика о работе File Sharing System, которая функционирует на производстве уже на протяжении двух месяцев. На сегодняшний момент проект находится в стадии сопровождения.

Поставленная во введении цель, заключающаяся в упрощении и снижении ресурсозатратности обмена данными между PLM и внешними поставщиками услуг, достигнута вследствие проведения автоматизации. Задачи, вытекающие из цели, успешно выполнены – проведен анализ процесса обмена данными до проведения автоматизации, выявлены проблемы, требующие устранения; спроектировано и описано автоматизированное решение возникших у компании проблем и, наконец, реализована и сдана заказчику File Sharing System – результат полугодовой работы команды российских программистов для американской компании.

Работа, также, укладывается в указанную во введении актуальность – внедренное автоматизированное решение обеспечивает достаточный уровень безопасности передачи данных, снижает вероятность потери данных при передаче, упрощает работу с большими объемами данных, также снижает ресурсозатратность как в финансовом отношении, так и в человеко-часах.

СПИСОК ЛИТЕРАТУРЫ И ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Беляевский И. К. Маркетинговое исследование [Электронный ресурс] : Информация, анализ, прогноз : учеб. пособие / И. К. Беляевский. - Москва : Финансы и статистика, 2014. - 319 с. - ISBN 5-279-02220-9.
2. Горбенко А.О. Информационные системы в экономике [Электронный ресурс]: учебное пособие/ Горбенко А.О.- Электрон. текстовые данные.- М.: БИНОМ. Лаборатория знаний, 2013.- 292 с.
3. Граничин О. Н. Информационные технологии в управлении [Электрон-ный ресурс] : учебное пособие / О. Н. Граничин, В. И. Кияев. - 2-е изд., испр. - Москва : ИНТУИТ, 2016. - 377 с. : ил. - (Основы информационных технологий). - ISBN 978-5-94774-986-1. -С. 235-245
4. Долженко А. И. Технологии командной разработки программного обеспечения информационных систем [Электронный ресурс] : [курс лекций] / А. И. Долженко. - 2-е изд., испр. - Москва : ИНТУИТ , 2016. - 300 с. : ил. - (Основы информационных технологий).
5. Золотов С. Ю. Проектирование информационных систем [Электронный ресурс] : учеб. пособие / С. Ю. Золотов ; Томский гос. ун-т систем управления и радиоэлектроники. - Томск : Эль Учебное пособие Контент, 2013. - 86 с. - ISBN 978-5-4332-0083-8. -С.17-23
6. Инвестиции [Электронный ресурс] : Системный анализ и управление : [учеб. пособие] / под ред. К. В. Балдина. - 4-е изд., испр. - Москва : Дашков и К°, 2013. - 287 с. - ISBN 978-5-394-01870-1.
7. Кандрашина Е.А. Финансовый менеджмент [Электронный ресурс] : учебник / Кандрашина Е.А. – Электрон. текстовые данные.- М.: Дашков и К, Ай Пи Эр Медиа, 2017. - 220 с. – ISBN 978-5-394-01579-3
8. Граничин О. Н. Информационные технологии в управлении [Электрон-ный ресурс] : учебное пособие / О. Н. Граничин, В. И. Кияев. - 2-е изд., испр. - Москва : ИНТУИТ, 2016. - 377 с. : ил. - (Основы информационных технологий). - ISBN 978-5-94774-986-1. -С. 24-35

9. Кастанова А.А. Реинжиниринг бизнес- процессов [Электронный ресурс]: методические указания к лабораторным работам/ Кастанова А.А.- Электрон. текстовые данные.- М.: Российский новый университет, 2014.- 32 с.- Режим доступа: <http://www.iprbookshop.ru/21308>.— ЭБС «IPRbooks»

10. Котляров В. П. Основы тестирования программного обеспечения [Электронный ресурс] : [учеб. пособие] / В. П. Котляров. - 2-е изд., испр. - Москва : ИНТУИТ, 2016. - 335 с. : ил. - (Основы информационных технологий). - ISBN 5-94774-406-4.

11. Красина Ф. А. Финансовый менеджмент [Электронный ресурс] : учебное пособие / Красина Ф. А. - Электрон. текстовые данные.- Томск : Томский государственный университет систем управления и радиоэлектроники, 2015. - 266 с.

12. Маклаков С.В. ВРwin и ERwin. CASE-средства разработки информационных систем. / С.В. Маклаков. - 3-е изд., испр. и доп. - М.: ДИАЛОГ-МИФИ, 2013. - 306 с.

13. Михеев А. Г. Процессное управление на свободном программном обеспечении [Электронный ресурс] : [учеб. пособие] / А. Г. Михеев. - 2-е изд., испр. - Москва : ИНТУИТ, 2016. - 230 с.

14. Николаева И. П. Инвестиции [Электронный ресурс] : учебник / И. П. Николаева. - Москва : Дашков и К°, 2013. - 253 с. - (Учебные издания для бакалавров). - ISBN 978-5-394-01410-9 .

15. О продукте Enovia [Электронный ресурс]. - URL: <https://www.3ds.com/products-services/enovia>

16. Официальный сайт S&C [Электронный ресурс]. - URL: <https://www.sandc.com/en/company/>

17. Привалов И. М. Основы аппаратного и программного обеспечения [Электронный ресурс] : учеб.-метод. пособие / И. М. Привалов ; Сев.-Кавказ. федерал. ун-т. - Ставрополь : СКФУ, 2015. - 145 с.

18. Прыкина Л. В. Экономический анализ предприятия [Электронный ресурс] : учебник / Л. В. Прыкина. - Москва : ЮНИТИ-ДАНА, 2017. - 407 с. -

ISBN 978-5-238-00503-2.

19. Реинжиниринг бизнес-процессов [Электронный ресурс]: учебное пособие/Блинов А.О.; под редакцией Блинова А.О. - М.: ЮНИТИ-ДАНА, 2015.- 343 с.

20. Сайт Glassdoor/Информация о S&C [Электронный ресурс]. - URL: https://www.glassdoor.com/Overview/Working-at-S-and-C-Electric-EI_IE4505.11,27.htm

21. Силич В.А. Реинжиниринг бизнес-процессов [Электронный ресурс]: учебное пособие/Силич В.А., Силич М.П.-Томск:ТУСУР, 2014.-199с.

22. Сорокин А. А. Реинжиниринг бизнес-процессов [Электронный ресурс] : учеб. пособие / А. А. Сорокин, А. Ю. Орлова ; Сев.-Кавказ. федерал. ун-т. - Ставрополь : СКФУ, 2014. - 212 с.

23. Стешин А. И. Информационные системы в организации [Электронный ресурс] : учебное пособие / А. И. Стешин. - Саратов : Вузовское образование, 2013. - 194 с. -С. 86-119.

24. Тельнов Ю. Ф. Инжиниринг предприятия и управление бизнес-процессами [Электронный ресурс] : методология и технология : учеб. пособие для студентов магистратуры, обучающихся по направлению «Прикладная информатика» / Ю. Ф. Тельнов, И. Г. Фёдоров. - Москва : ЮНИТИ-ДАНА, 2015. - 207 с. - ISBN 978-5-238-02622-0.

25. Фадеева О. Ю. Информационные системы в экономике [Электронный ресурс] : учеб. пособие / О. Ю. Фадеева, Е. А. Балашова. - Омск : Омский гос. ин-т сервиса, 2015. - 99 с. - ISBN 978-5-93252-360-5. - С. 44-49.

26. Швецов В. И. Базы данных [Электронный ресурс] : [учебное пособие] / В. И. Швецов. - 2-е изд., испр. - Москва : ИНТУИТ, 2016. - 218 с. : ил. - С.67-84.

27. Сайт Amazon / Облачное хранилище S3 [Электронный ресурс]. - URL: <https://aws.amazon.com/ru/s3>

28. Sbarski Peter / Serverless Architectures on AWS 2nd Edition / Sbarski Peter - 2 edition - : Manning Publications, 2018. – 500 с.

29. Stephen Corona / nginx: A Practical Guide to High Performance 1st Edition / Stephen Corona - 1 edition - : O'Reilly Media, 2018. – 250 с.

30. Jan Machacek / Reactive Systems Architecture: Designing and Implementing an Entire Distributed System / Jan Machacek, Martin Zapletal, Michal Janousek, Anirvan Chakraborty - 1 edition - : O'Reilly Media, 2018. – 200 с.

31. Shannon Bradshaw / MongoDB: The Definitive Guide: Powerful and Scalable Data Storage 3rd Edition / Shannon Bradshaw, Kristina Chodorow - 3 edition - : O'Reilly Media, 2018. – 425 с.

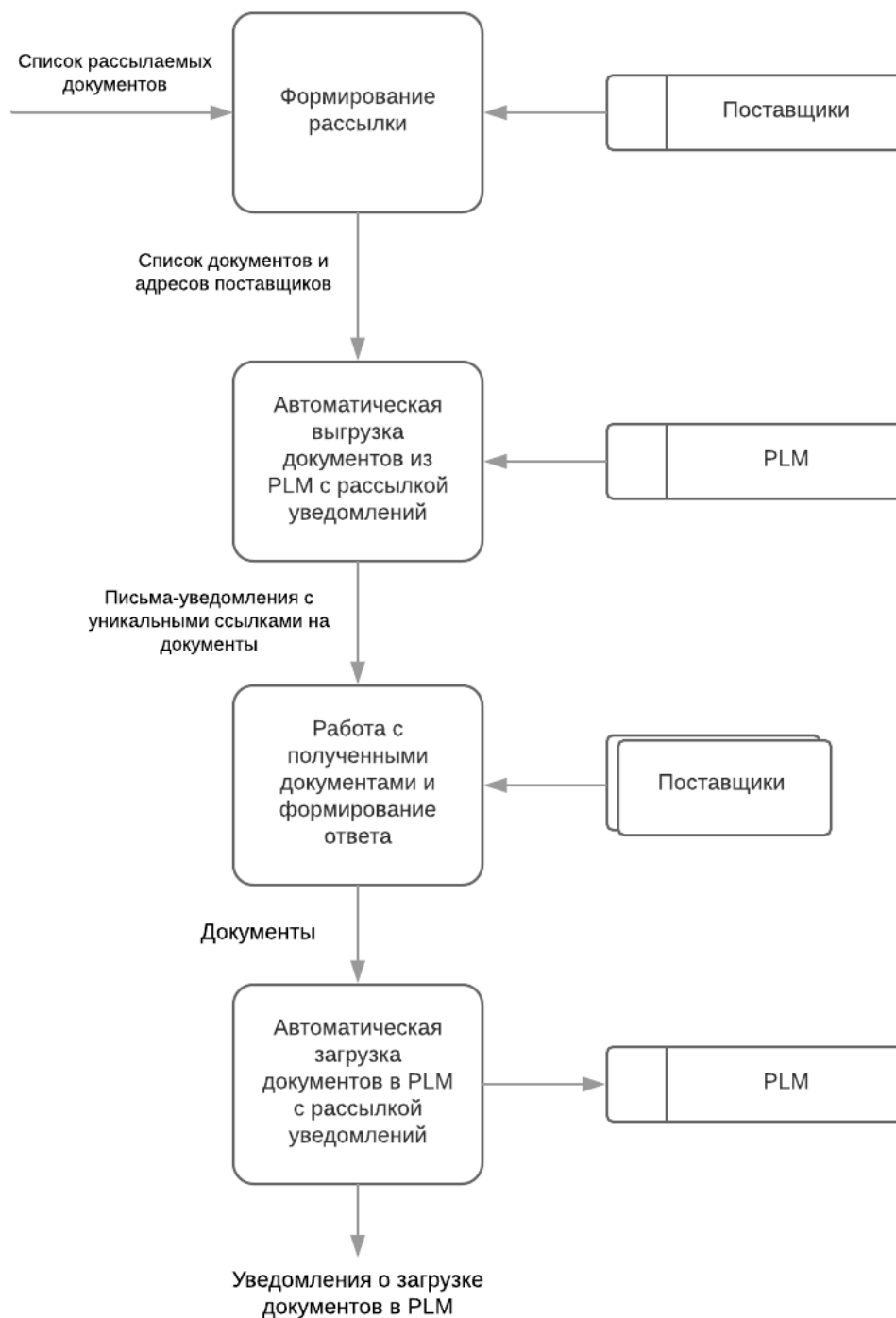
32. Robert C. Martin / Clean Architecture: A Craftsman's Guide to Software Structure and Design / Robert C. Martin - 1 edition - : Prentice Hall, 2017. – 432 с.

33. Янова П. Г. Экономический анализ [Электронный ресурс] : учеб.-метод. комплекс / П. Г. Янова. - Саратов : Вузовское образование, 2013. - 295 с.

Приложение 2. Диаграмма потоков данных, на которой изображена общая схема, по которой S&C начали работать после внедрения автоматизированного решения



Декомпозиция диаграммы потоков данных общей схемы, по которой начали работать S&C после автоматизации



Приложение 3. Диаграмма взаимодействия РМ и поставщика с File Sharing System

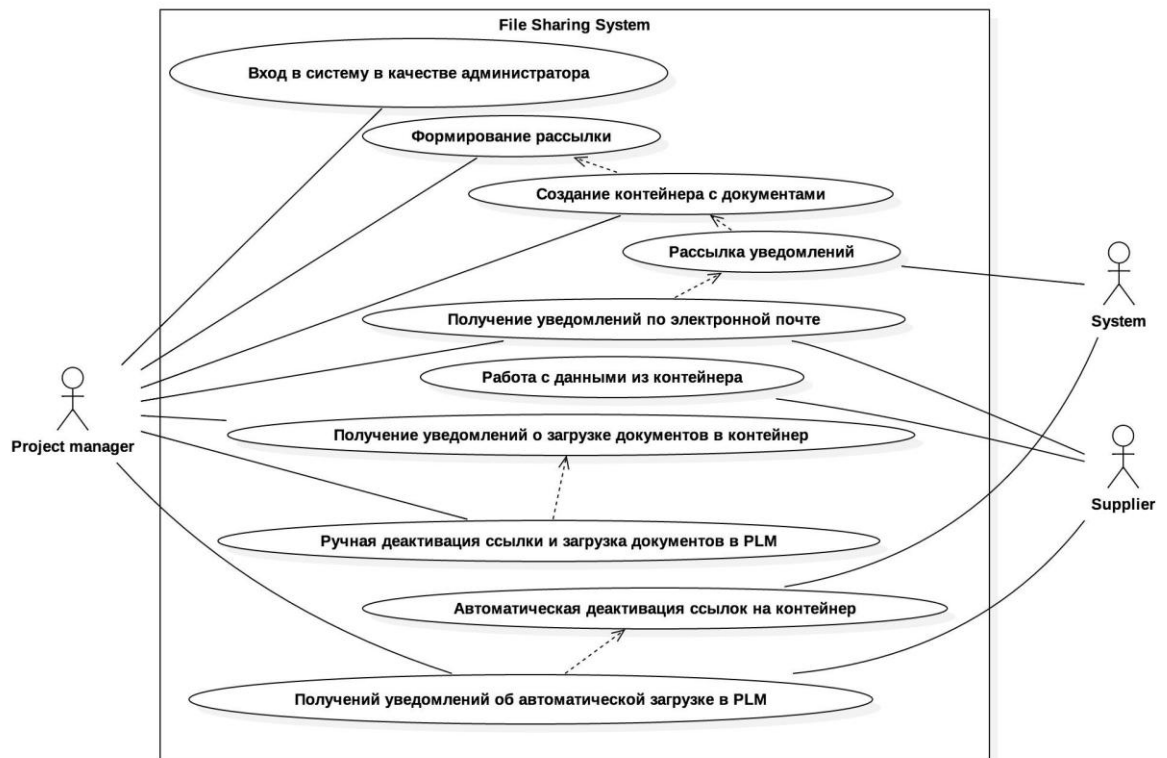


Диаграмма взаимодействия классов в File Sharing System

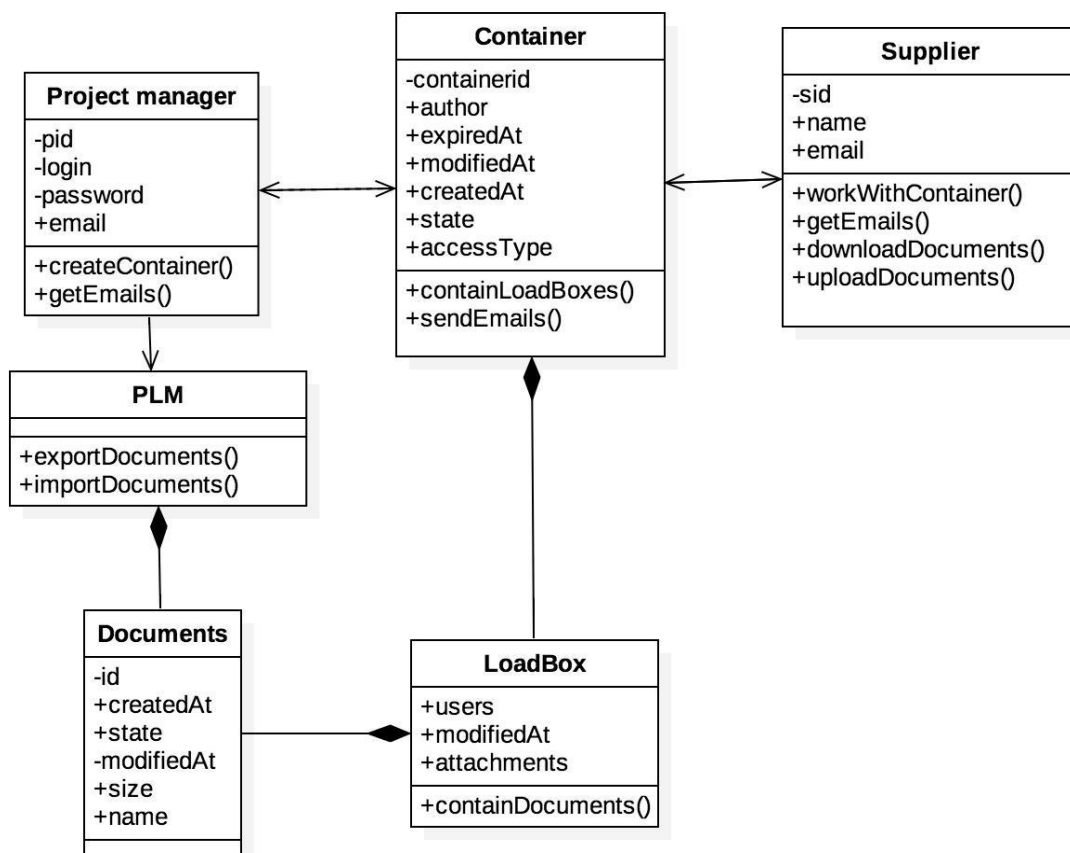
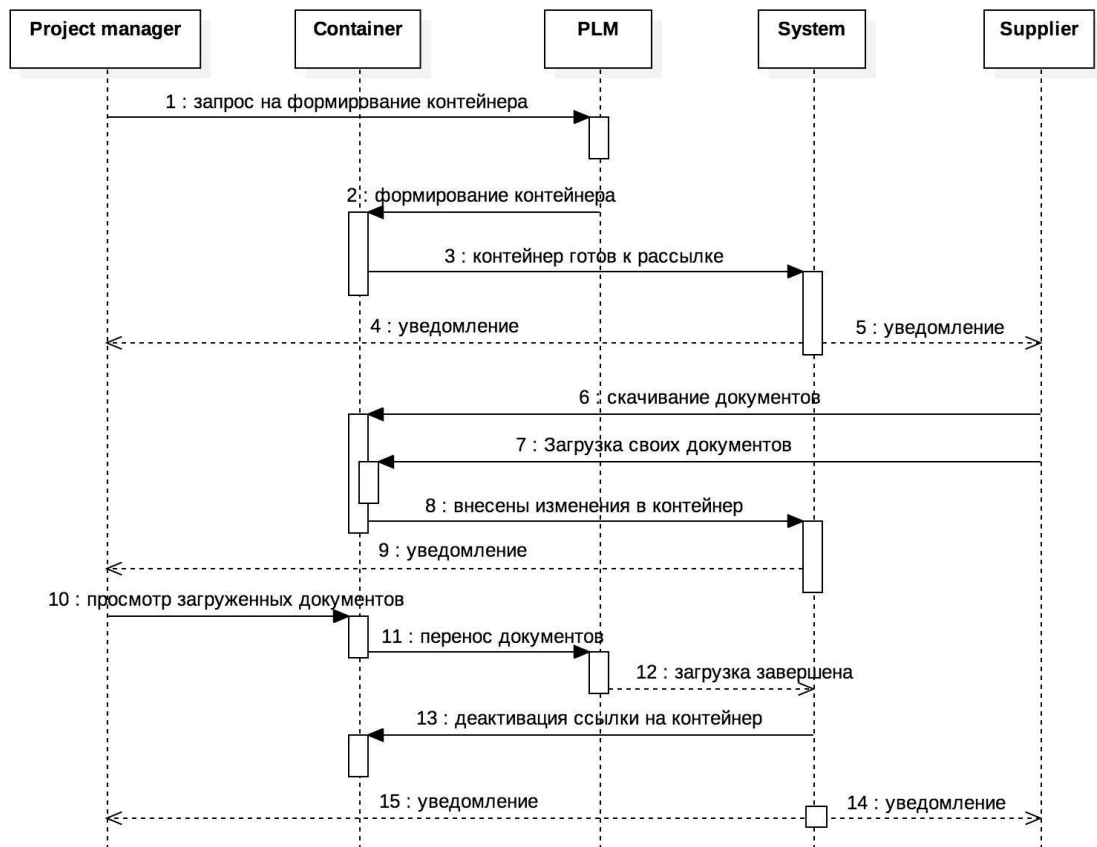
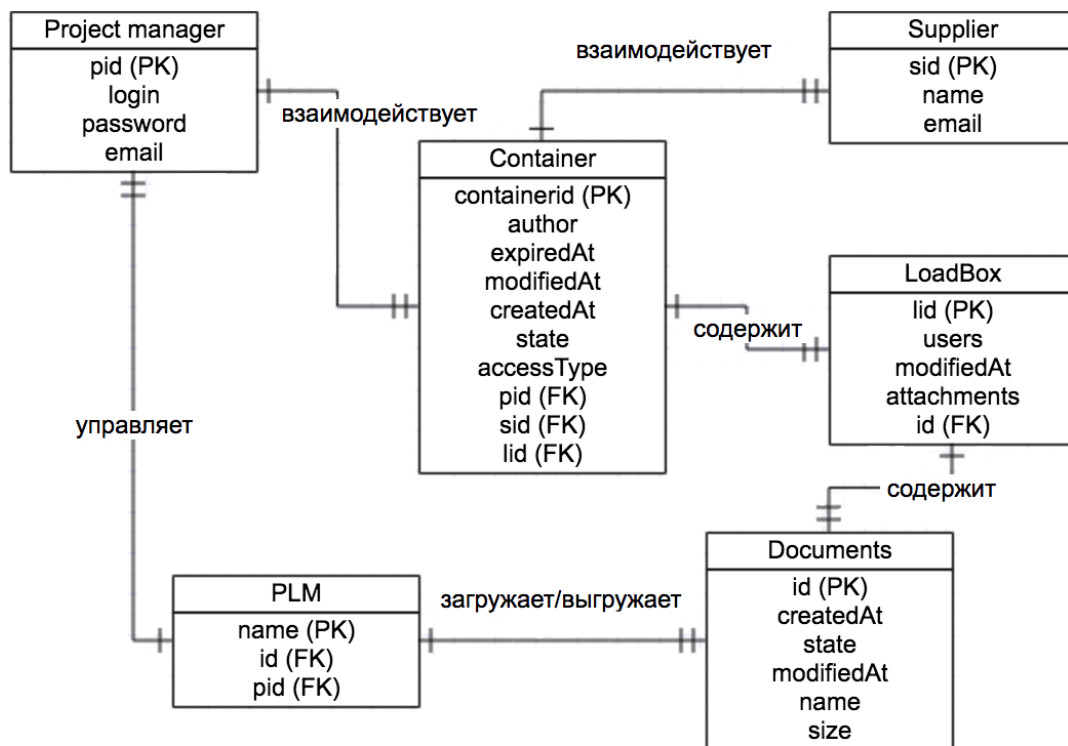


Диаграмма последовательности предметной области



Логическая модель рассматриваемой предметной области



Приложение 4. JSON структура контейнера

```
.
{
  "_id" : ObjectId("5aeabe14dda04e048d0e557a"),
  "containerId" : "f06436f0-4ea5-11e8-bc85-5966e1566112",
  "author" : "_@senticore.com",
  "internalPath" : "cont1/",
  "expiredAt" : ISODate("2018-05-11T05:08:42.000Z"),
  "subject" : "test subject",
  "message" : "Hello! Here is the files that you asked for. I hope you'l send me changes soon.",
  "tag" : "_",
  "loadBoxes" : [
    {
      "users" : [
        "_@gmail.com"
      ],
      "modifiedAt" : ISODate("2018-05-03T07:45:24.458Z"),
      "attachments" : [
        {
          "createdAt" : ISODate("2018-05-03T08:15:02.596Z"),
          "state" : "available",
          "modifiedAt" : ISODate("2018-05-03T08:15:02.596Z"),
          "_id" : ObjectId("5aeac5068152fc0d86b4b40b"),
          "size" : 14,
          "name" : "file2",
          "id" : "9855fd96-519e-4fbc-a120-890200eea138"
        },
        {
          "createdAt" : ISODate("2018-05-03T08:15:04.226Z"),
          "state" : "available",
          "modifiedAt" : ISODate("2018-05-03T08:15:04.226Z"),
          "_id" : ObjectId("5aeac5088152fc0d86b4b40c"),
          "size" : 14,
          "name" : "file3",
          "id" : "b8319512-b1e6-4f67-8e67-8b9da06d0e74"
        },
        {
          "createdAt" : ISODate("2018-05-03T08:15:05.966Z"),
          "state" : "available",
          "modifiedAt" : ISODate("2018-05-03T08:15:05.966Z"),
          "_id" : ObjectId("5aeac5098152fc0d86b4b40d"),
          "size" : 14,
          "name" : "file11",
          "id" : "64f37a9a-4ffb-4666-a59d-70eb81439a52"
        }
      ],
      "_id" : ObjectId("5aeabe14dda04e048d0e557b"),
      "loadboxId" : "6d829dbf-b21b-4859-b937-54a2bc116181",
      "state" : "available"
    }
  ],
}
```

```

"attachments" : [
  {
    "createdAt" : ISODate("2018-05-03T07:45:24.474Z"),
    "state" : "available",
    "modifiedAt" : ISODate("2018-05-03T07:45:24.474Z"),
    "_id" : ObjectId("5aeabe14dda04e048d0e557e"),
    "size" : 14,
    "id" : "c8a73241-a523-4f79-86d7-caf03543d1f6",
    "name" : "file1 one two three.exe"
  },
  {
    "createdAt" : ISODate("2018-05-03T07:45:24.477Z"),
    "state" : "available",
    "modifiedAt" : ISODate("2018-05-03T07:45:24.477Z"),
    "_id" : ObjectId("5aeabe14dda04e048d0e557f"),
    "size" : 14,
    "id" : "5f83c5a8-daed-4491-9faf-f39fb33dd90b",
    "name" : "file1 ;,@&=+$#.a test .docx"
  },
  {
    "createdAt" : ISODate("2018-05-03T07:45:24.478Z"),
    "state" : "available",
    "modifiedAt" : ISODate("2018-05-03T07:45:24.478Z"),
    "_id" : ObjectId("5aeabe14dda04e048d0e5580"),
    "size" : 14,
    "id" : "8ebcd2e9-fffb-4ff5-8667-6262fdb598f3",
    "name" : "file3"
  },
  {
    "createdAt" : ISODate("2018-05-03T07:45:24.479Z"),
    "state" : "available",
    "modifiedAt" : ISODate("2018-05-03T07:45:24.479Z"),
    "_id" : ObjectId("5aeabe14dda04e048d0e5581"),
    "size" : 45777,
    "id" : "b9de1842-6f4a-479b-9efd-c07dacao038e",
    "name" : "img.png"
  }
],
"modifiedAt" : ISODate("2018-05-03T07:45:24.439Z"),
"createdAt" : ISODate("2018-05-03T07:45:24.439Z"),
"state" : "available",
"accessType" : "RW",
"__v" : 0
}

```

Приложение 5. JSON структура письма

```
{
  "_id" : ObjectId("5aeabe75f565a405beb73438"),
  "subject" : "test subject",
  "templateName" : "extTemplate",
  "data" : {
    "documentsList" : [
      { "title" : "file1 one two three.exe"
      },
      { "title" : "file1 ;,@&=+$.a test .docx"
      },
      { "title" : "file3"
      },
      { "title" : "img.png"
      }
    ],
    "fileCountShow" : 6,
    "filesSize" : "44.7 Kb",
    "externalUser" : "_@gmail.com",
    "packageComments" : "Click View to see the package.",
    "eventName" : "sent you a transmittal package",
    "linkForUser" : "http://192.168.1.111:888/ext/9a2b3792-f3d6-4d60-b001-77be2e5bf1b9/",
    "internalUser" : "_@senticore.com"
  },
  "system" : {
    "state" : "created",
    "status" : "the email was created, not yet sent",
    "isNeedSend" : true,
    "sendFailedAt" : null,
    "sentAt" : null,
    "createdAt" : ISODate("2018-05-03T07:47:01.562Z")
  },
  "mailTo" : [
    "_@gmail.com"
  ],
  "__v" : 0 }
}
```

Приложение 6. JSON структура уникальной ссылки

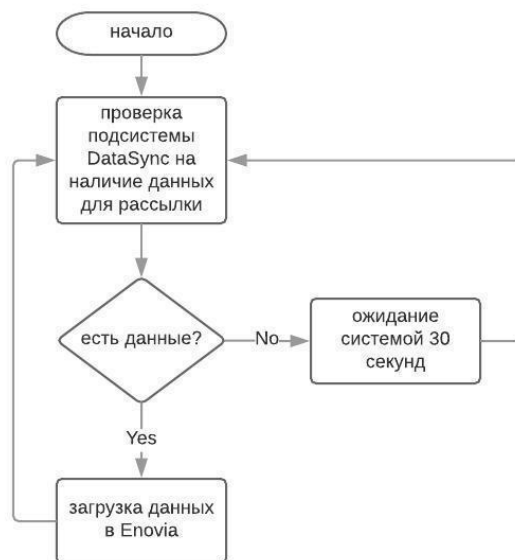
```
{
  "_id" : ObjectId("5a9d2b1436f8776533651008"),
  "linkId" : "96671f2b-6720-4a9e-881b-eccd826bab13",
  "enable" : false,
  "createdAt" : ISODate("2018-03-05T11:33:40.603Z"),
  "modifiedAt" : ISODate("2018-03-05T11:33:40.603Z"),
  "data" : {
    "cid" : "0d8d9e80-2069-11e8-aa61-4791c70f19e3",
    "lid" : "39eba5ad-74b4-4293-ac7f-8725db961ee9",
    "role" : "ext"
  },
  "__v" : 0
}
```


Приложение 7. Блок-схемы процессов в подсистемах

Процесс передачи данных в подсистеме PLM Integration



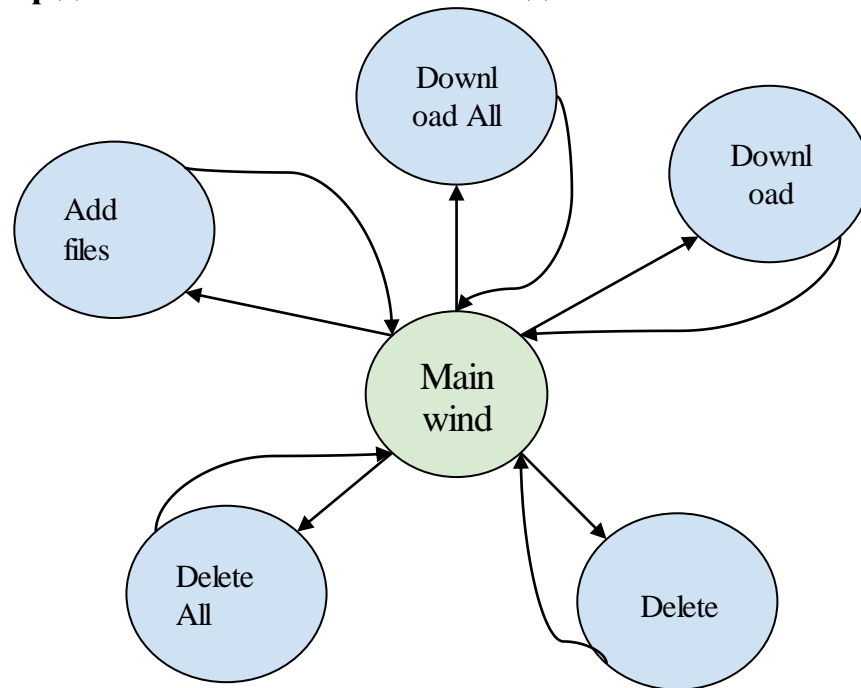
Процесс загрузки данных в подсистеме PLM Integration



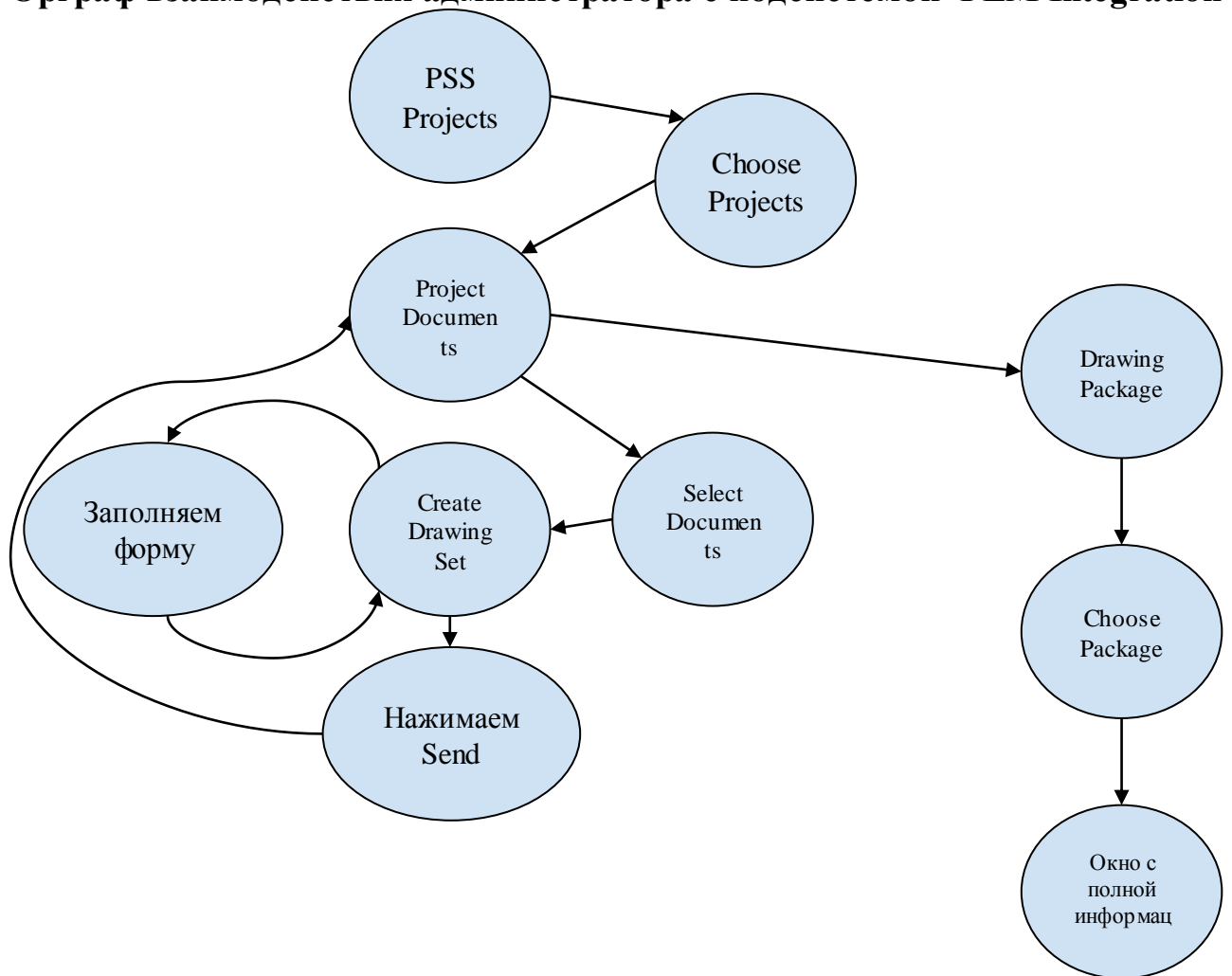
Работа поставщика с подсистемой Web Exchange



Орграф диалога пользователя с подсистемой Web Exchange

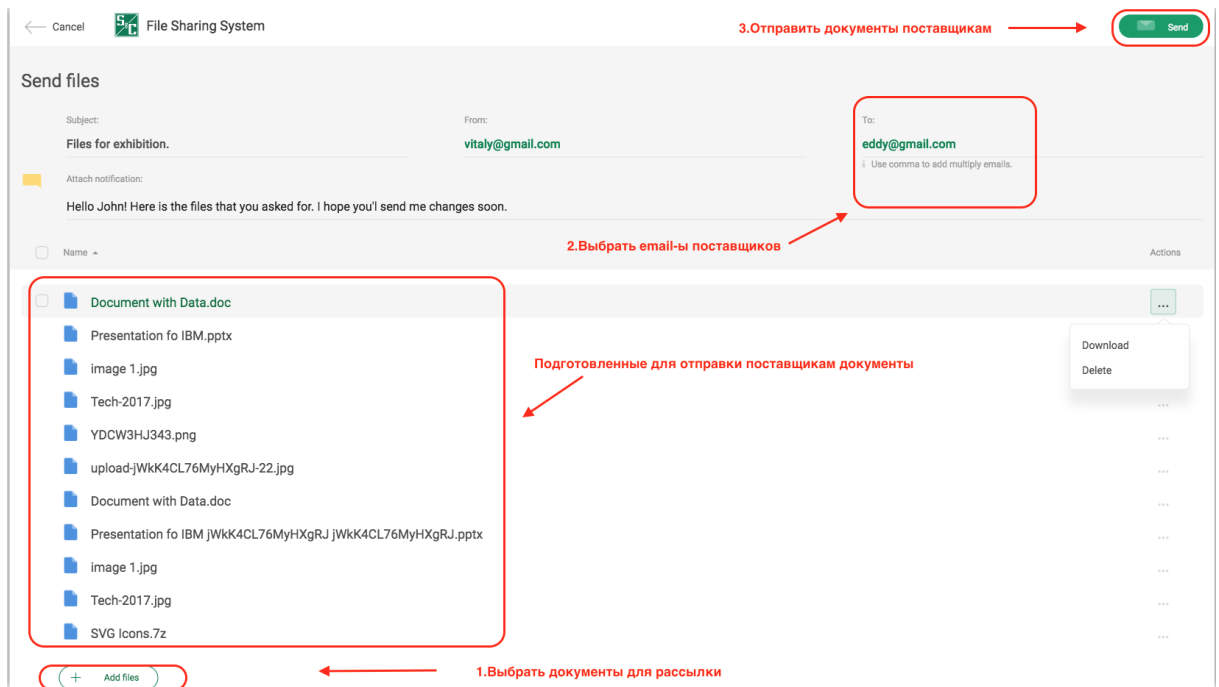


Орграф взаимодействия администратора с подсистемой PLM Integration



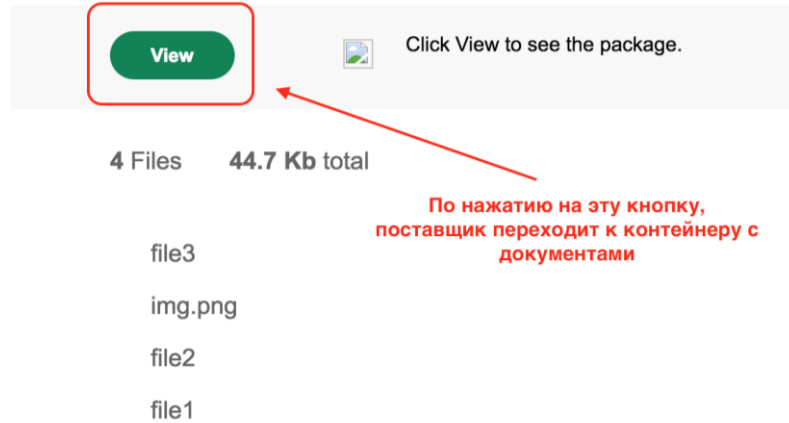
Приложение 8. Контрольный пример реализации File Sharing System и ее описание

Формирование контейнера для отправки поставщикам



Письмо с ссылкой на содержимое контейнера

... @senticore.
com
sent you some files



Контейнер, доступный поставщику для выгрузки и загрузки документов

File Sharing System

Files package: Files for exhibition

from: vitaly@gmail.com
to: eddy@gmail.com

Notification: Hello John! Here is the files that you asked for. I hope you'll send me changes soon.

Download All

Download Delete

Скачать документы

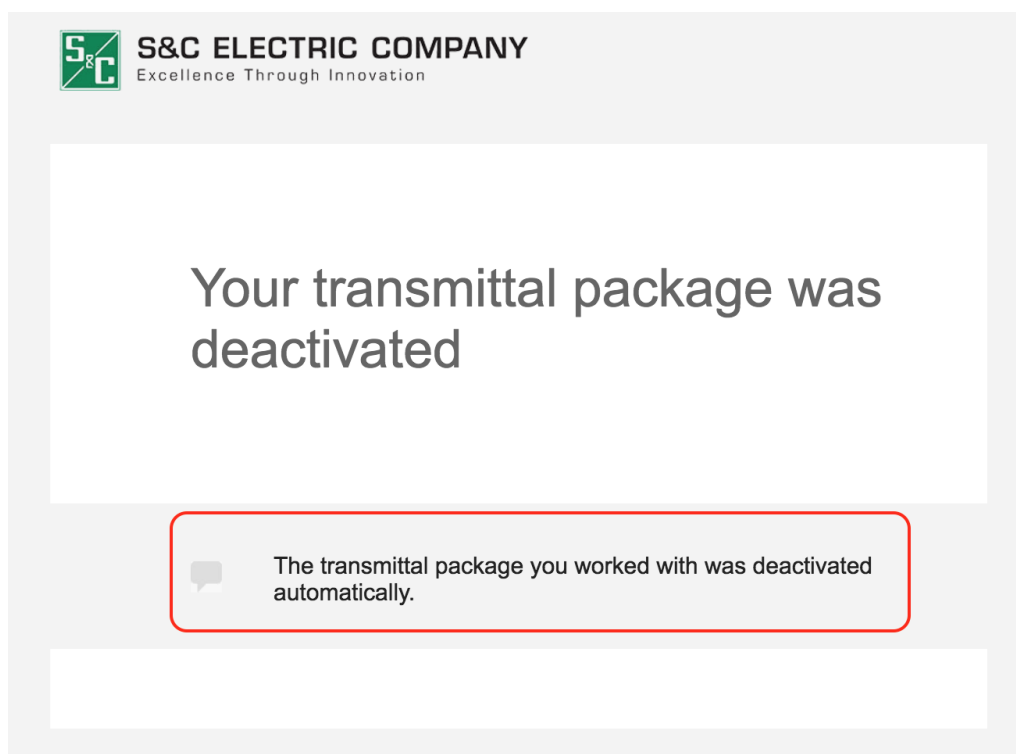
| Name | Added |
|--|-----------------|
| Document with Data.doc | - |
| Presentation fo IBM.pptx | - |
| image 1.jpg | - |
| Tech-2017.jpg | 4/8/2017 10:34 |
| YDCW3HJ343.png | 4/8/2017 10:00 |
| upload-jWkK4CL76MyHXgRJ-22.jpg | - |
| Document with Data.doc | - |
| Presentation fo IBM jWkK4CL76MyHXgRJ jWkK4CL76MyHXgRJ.pptx | 14/8/2017 16:00 |
| image 1.jpg | - |
| Tech-2017.jpg | 26/1/2017 12:30 |
| SVG Icons.7z | 4/8/2017 11:00 |

Документы, отправленные S&C

Документы, загружаемые поставщиком

Загрузить документы

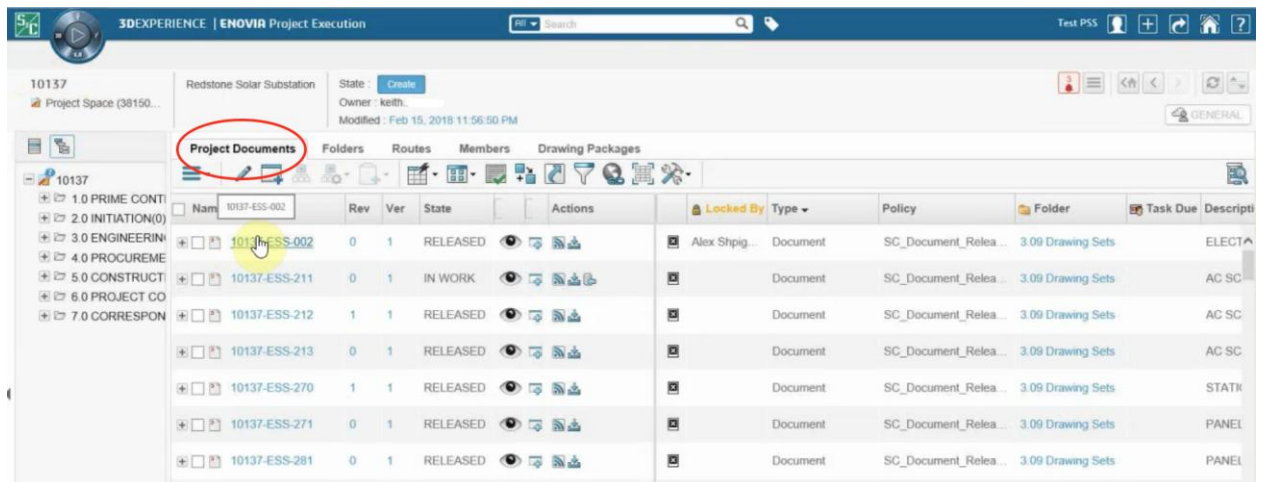
Уведомление, которое приходит поставщикам при автоматической деактивации контейнера, с которым они работали



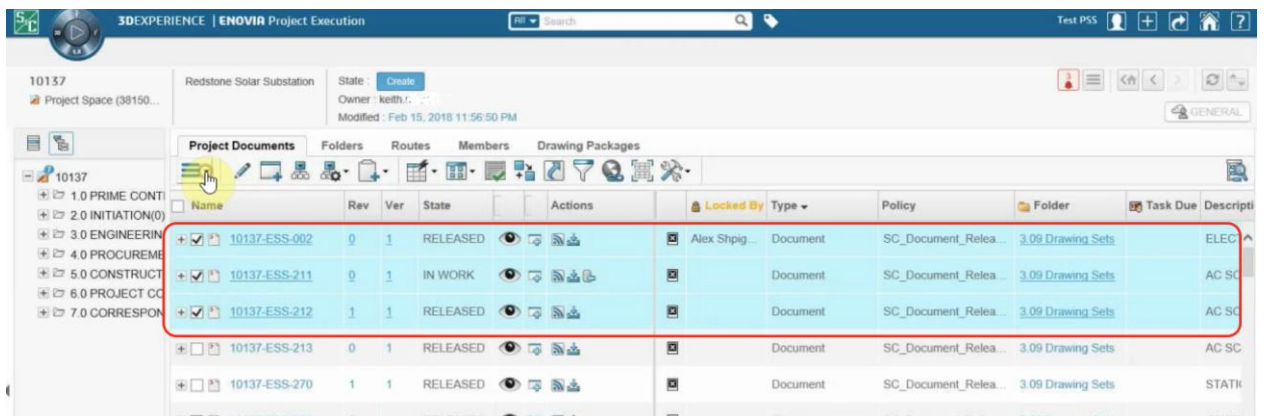
Отображение списка проектов отдела PSS в системе Enovia

| Name | Customer | Description | Due Date | Owner | Created Date |
|-------|----------|----------------------------|--------------|-------|--------------|
| 00001 | | New Folder Structure | Nov 30, 2017 | Alex | Nov 30, 2017 |
| 10137 | | Redstone Solar Substation | Sep 21, 2017 | Keith | Sep 21, 2017 |
| 11084 | | Villa Rose Microgrid | Mar 1, 2017 | Richa | Mar 13, 2017 |
| 11323 | | ULL PV Project | Feb 1, 2017 | Richa | Mar 1, 2017 |
| 11590 | | Duke Energy Oneal Retail | Apr 4, 2017 | Richa | Apr 4, 2017 |
| 11691 | | NXP Express Feeder | Jul 31, 2017 | Davic | Jul 31, 2017 |
| 11721 | | Port Fourchon DSTATCOM | May 17, 2017 | Davic | May 17, 2017 |
| 11727 | | Leeville DSTATCOM | May 17, 2017 | Davic | May 17, 2017 |
| 11762 | | Duke Energy - New Shac... | Aug 24, 2017 | Richa | Aug 24, 2017 |
| 11841 | | Exeter Solar | Jul 31, 2017 | Davic | Jul 31, 2017 |
| 11844 | | Richmond Solar | Jul 31, 2017 | Davic | Jul 31, 2017 |
| 11846 | | North Smithfield Solar | Jul 31, 2017 | Davic | Jul 31, 2017 |
| 12206 | | Duke Energy - Katherine... | Sep 29, 2017 | Micha | Sep 29, 2017 |

Отображение перечня документов в Enovia внутри определенного проекта



Выделение документов на отправку поставщикам внутри проекта в PLM Enovia



Отображение списка контейнеров в системе Enovia

The screenshot shows the 'Drawing Packages' tab in the ENOVIA Project Execution interface. A table lists various drawing sets. The entry 'New_Test020' is highlighted with a red circle, and a red arrow points to it from the text 'Созданный нами тестовый контейнер'.

| Name | Type | Drop | Rev | Ver | Latest Rev | Actions | Locked By | State | Owner | Description | Task | Task Title |
|---------------|---------------|------|------|-----|------------|---------|-----------|---------|-------------|--------------------------------|--------|------------|
| 10137 | Project Space | | | | True | | | Create | Keith Moses | Redstone Solar Substation | | |
| New_Test | Drawing Set | | 1... | | True | | | Sent... | Test PSS | Button test | Feb... | |
| New_Test002 | Drawing Set | | 1... | | True | | | Sent... | Test PSS | Testing Multi-value attribu... | Feb... | |
| New_Test003 | Drawing Set | | 1... | | True | | | Cre... | Test PSS | Document OID save to at... | Feb... | |
| New_Test004 | Drawing Set | | 1... | | True | | | Sent... | Test PSS | Test Docstring | Feb... | |
| New_Test005 | Drawing Set | | 1... | | True | | | Sent... | Test PSS | Test Multivalue attribute | Feb... | |
| New_Test008 | Drawing Set | | 1... | | True | | | Sent... | Test PSS | again | Feb... | |
| New_Test010 | Drawing Set | | 1... | | True | | | Cre... | Test PSS | again | Feb... | |
| New_Test020 | Drawing Set | | 1... | | True | | | Sent... | Test PSS | This is a test to quick sen... | Feb... | |
| New_Test-001 | Drawing Set | | 1... | 1 | True | | | Cre... | Test PSS | added attribute for Recept... | Feb... | |
| Test_Carl-001 | Drawing Set | | 1... | | True | | | Cre... | Test PSS | Carl Breving, User testing | Feb... | |
| Test_Carl-002 | Drawing Set | | 1... | | True | | | Cre... | Test PSS | Email Recipients Test | Feb... | |
| Test_KRL001 | Drawing Set | | 1... | | True | | | Sent... | Test PSS | Test | Jan... | |

Отображение полной информации о созданном контейнере с указанием уникальной ссылки на него

The screenshot shows the 'New_Test020' container details in the ENOVIA Collaboration and Approvals interface. The 'Recipient URLs' field contains a unique URL, which is highlighted with a red circle.

| Field | Value |
|-------------------------|---|
| Name | New_Test020 |
| Revision | 1519102842377 |
| Type | Drawing Set |
| Vault | eService Production |
| Title | New_Test020 |
| Description | This is a test to quick send of Drawing Package |
| State | Sent to External |
| Owner | Test PSS |
| Originated | Feb 19, 2018 |
| Modified | Feb 19, 2018 |
| Policy | SC_Drawing Set |
| Title | New_Test020 |
| Package Due Date | Feb 26, 2018 |
| Instructions | This is a test to quick send of Drawing Package |
| Project | 10137 |
| Folder Name | 10137_New_Test020_1519102842377/ |
| Email Recipients | ... |
| Recipient URLs | ... https://fileshare.plm.sandc.com/nt/0be990b-14b8-4373-8e72-9c12538cccd/ |
| Comments | |
| Classification Path (s) | |

Приложение 9. Пример Container REST API

Пример Container REST API подсистемы Web Exchange, выполненный с использованием Node.js, который предоставляет информацию о контейнерах и позволяет скачивать и загружать документы.

```
«use strict»;

const mongoose = require('mongoose');
const ContainerAPI = require('./logics/containerApi');
const Container = require('./models/container');
const ContainerModel = Container.Model;

const Link = require('./models/linkModel');

const restify = require('restify');
const errs = require('restify-errors');
const corsMiddleware = require('restify-cors-middleware');
const jwtMiddleware = require('restify-jwt');

const _ = require('underscore');
const moment = require('moment');
const uuid = require('uuid');

const log = require('./log');
const aws_s3 = require('./aws_s3_integration');
const s3BrowserDirectUpload = require('./s3client');

const containersRoute = "/containers";
const containerRoute = "/containers/:containerId";
const containerAttachmentsRoute = "/containers/:containerId/attachments";
const containerAttachmentRoute = "/containers/:containerId/attachments/:attachmentId";
const containerLoadboxesRoute = "/containers/:containerId/loadboxes";
const containerLoadboxRoute = "/containers/:containerId/loadboxes/:loadboxId";
const containerLoadboxAttachmentsRoute =
"/containers/:containerId/loadboxes/:loadboxId/attachments";
const containerLoadboxAttachmentRoute =
"/containers/:containerId/loadboxes/:loadboxId/attachments/:attachmentId";

const containerLoadboxAttachmentDownloadLinkRoute =
"/containers/:containerId/loadboxes/:loadboxId/attachments/:attachmentId/downloadLink";
const containerAttachmentDownloadLinkRoute =
"/containers/:containerId/attachments/:attachmentId/downloadLink";

const userContainerFields = ["containerId", "author", "accessType", "state", "createdAt",
"publishedAt", "expiredAt", "subject", "message", "internalPath", "tag"];
const userLoadboxFields = ["loadboxId", "state", "users"];
const userAttachmentFields = ["id", "name", "hash", "size", "state", "createdAt", "modifiedAt"];
```

```

const userContainerAttachmentFields = [ "name", "state", "id" ];

// filter in pos
const userLoadboxPutFields = [ "state", "users"];
const userAttachmentPutFields = [ "name", "state", "size"];
const userContainerPutFields = [ "state" ];

var containersPathInS3 = "containers";

const loadBoxAvailable = "available";
const resourceNotFound = 404;
const resourceNotAvailable = 403;
const loadBoxBlocked = "blocked";
const loadBoxExpired = "expired";
const containerExpired = "expired";
const containerUnavailable = "unavailable";
const containerCreated = "created";

const acl = require("./acl_restapi");

exports.createServer = function (options, restifyOptions) {
  log.info("Container RESTAPI server creating...");

  let settings = options || {};
  let serverOptions = restifyOptions || {};

  let path = settings.path || ""

  serverOptions.name = 'ContainerRESTAPIServer'

  var server = restify.createServer(serverOptions);

  server.use(restify.plugins.queryParser({ mapParams: false }));
  server.use(restify.plugins.bodyParser({ mapParams: false }));

  server.use(function (req, res, next) {
    res.noCache();
    return next();
  });

  server.use(function(req, res, next) {
    if ((req.header('X-API-Key') != undefined) && (settings.allowApiKeys !=
undefined)) {
      if (_.includes(settings.allowApiKeys, req.header('X-API-Key'))) {
        req['user'] = {"role": "system"};
        return next();
      }
    }

    jwtMiddleware({ secret: settings.jwt.secret})(req,res,next);
  });
}

```

```

// CORS
const cors = corsMiddleware(settings.cors)

server.pre(cors.preflight)
server.use(cors.actual)

server.on('restifyError', function(req, res, err, callback) {
  //log.debug("error = ", req);
  err.toJSON = function customToJSON() {
    return {
      name: err.name,
      message: err.message
    };
  };
  err.toString = function customToString() {
    return 'i just want a string';
  };
  return callback();
});

/*
  Configure AWS S3
*/
containersPathInS3 = settings.s3uploader.containersPath;

// S3 move
aws_s3.init({ accessKeyId: settings.aws.accessKeyId, secretAccessKey:
settings.aws.secretAccessKey, region: settings.aws.region}, settings.s3uploader.bucket);

// S3 uploader
let s3client = new s3BrowserDirectUpload(settings.aws, settings.s3uploader.allowedTypes);

/*
  Containers REST link to s3
*/

// server.post(path + "/getUploadLink", acl.middlewareAcl(['ext']), function(req, res, next) {
server.post(path + "/getUploadLink2", acl.middlewareAcl([]), function(req, res, next) {
  // log.debug("TEst1 s3uploader", req.body);
  s3uploader(s3client, settings.s3uploader, req, res, next);
});

server.post(path + "/getUploadLink", acl.middlewareAcl(['ext']), function(req, res, next) {
  // log.debug("TEst2 s3uploaderLoadBoxAttachments");
  ContainerModel.findOne({"containerId" : req.body.containerId}, function(err,
container) {
    if (err) {
      res.send(resourceNotFound);
    }
  });
});

```

```

        next();
    }
    if ((container.accessType == "RO")||(container.state ==
containerCreated))||(container.state == containerUnavailable))||(container.state == containerExpired))
{
        res.send(resourceNotAvailable);
        next();
    } else {
        ContainerModel.findOne({"containerId" : req.body.containerId,
"loadBoxes.loadboxId" : req.body.loadboxId} , {"loadBoxes.$":1}, function(err, loadBoxesArr) {
            if (err) {
                res.send(resourceNotFound);
                next();
            }
            if (loadBoxesArr.loadBoxes[0].state == loadBoxAvailable) {

                s3uploaderLoadBoxAttachments(s3client,
settings.s3uploader, req, res, next);

            } else {
                res.send(resourceNotAvailable);
                next();
            }
        });
    }
});
});

/--OLD dont use--
server.post(path + "/getUploadLink1", acl.middlewareAcl(['ext']), function(req, res, next) {
    // log.debug("TEst2 s3uploaderLoadBoxAttachments");
    ContainerModel.findOne({"containerId" : req.body.containerId,
"loadBoxes.loadboxId" : req.body.loadboxId} , {"loadBoxes.$":1}, function(err, loadBoxesArr) {
        if (err) {
            return sendError(resourceNotFound, next);
        }
        ContainerModel.findOne({"containerId" : req.body.containerId},
function(err, container) {
            if (err) {
                return sendError(resourceNotFound, next);
            }

            if ((loadBoxesArr.loadBoxes[0].state ==
loadBoxAvailable)&&(container.accessType != "RO")) {
                s3uploaderLoadBoxAttachments(s3client, settings.s3uploader,
req, res, next);

            } else {
                res.send(resourceNotAvailable);
                next();
            }
        });
    });
});

```

```

    });
  });
  //--/OLD--

  server.get(path + containerLoadboxAttachmentDownloadLinkRoute, acl.middlewareAcl([]),
function(req, res, next) {
  //log.debug(req);
  // log.debug("TEst3 getLoadboxAttachmentDownloadLink");
  getLoadboxAttachmentDownloadLink(settings.s3uploader, req, res, next);
});

  server.get(path + containerAttachmentDownloadLinkRoute, acl.middlewareAcl(['ext']),
function(req, res, next) {
  //log.debug(req);
  // log.debug("TEst4 getContainerAttachmentDownloadLink");
  getContainerAttachmentDownloadLink(settings.s3uploader, req, res, next);
});

/*
Containers REST
*/

  server.get(path + containersRoute, acl.middlewareAcl(['system','admin','manager']),
getContainers);
  server.post(path + containersRoute, acl.middlewareAcl(['system']), newContainer);

  server.get(path + containerRoute, acl.middlewareAcl(['system','admin','manager','int','ext']),
acl.middlewareCondition("cid"), getContainer);
  server.del(path + containerRoute, acl.middlewareAcl(['system','admin']), delContainer);

  server.put(path + containerRoute, acl.middlewareAcl(['system','admin','manager']),
putContainer);

/*
Container Attachments REST
*/

  server.get(path + containerAttachmentsRoute,
acl.middlewareAcl(['system','admin','manager','int','ext']), acl.middlewareCondition("cid"),
getContainerAttachments);
  server.post(path + containerAttachmentsRoute, acl.middlewareAcl(['system']),
newContainerAttachment);

  server.get(path + containerAttachmentRoute,
acl.middlewareAcl(['system','admin','manager','int','ext']), acl.middlewareCondition("cid"),
getContainerAttachment);
  server.put(path + containerAttachmentRoute, acl.middlewareAcl(['system']),
putContainerAttachment);

  server.del(path + containerAttachmentRoute, acl.middlewareAcl([]),
delContainerAttachment);

/*

```

```

        LoadBoxes REST
    */
    server.get(path + containerLoadboxesRoute,
    acl.middlewareAcl(['system','admin','manager']), acl.middlewareCondition("cidlid"),
    getLoadboxes);
    server.post(path + containerLoadboxesRoute, acl.middlewareAcl(['system']), newLoadbox);

    server.get(path + containerLoadboxRoute,
    acl.middlewareAcl(['system','admin','manager','int','ext']), acl.middlewareCondition("cidlid"),
    getLoadbox);
    server.del(path + containerLoadboxRoute, acl.middlewareAcl([]), delLoadbox);

    server.put(path + containerLoadboxRoute, acl.middlewareAcl(['system']), putLoadbox);

    /*
        LoadBox Attachments REST
    */
    server.get(path + containerLoadboxAttachmentsRoute,
    acl.middlewareAcl(['system','admin','manager','int','ext']), acl.middlewareCondition("cidlid"),
    getLoadboxAttachments);
    server.post(path + containerLoadboxAttachmentsRoute, acl.middlewareAcl(['system','ext']),
    acl.middlewareCondition("cidlid"), newLoadboxAttachment);

    server.get(path + containerLoadboxAttachmentRoute,
    acl.middlewareAcl(['system','admin','manager','int','ext']), acl.middlewareCondition("cidlid"),
    getLoadboxAttachment);
    server.del(path + containerLoadboxAttachmentRoute, acl.middlewareAcl(['system','ext']),
    acl.middlewareCondition("cidlid"), delLoadboxAttachment);

    return server;
}

function sendError(err, next) {
    let restError = new errs.RestError({
        restCode: "ContainerError",
        message: err.toString()
    });

    next(restError);
}

function s3uploaderLoadBoxAttachments(s3client, options, req, res, next) {
    let file_id = uuid.v4();

    var uploadPostFormOptions = {
        key: options.inboxPath + "/" + file_id,
        //key: req.body.filename, // required
        bucket: options.bucket, // required
    }

```

```

        //extension: 'ext', // optional (pass if You want to check with allowed extensions or
set ContentType)
        acl: 'public-read', // optional, default: 'public-read'
        expires: moment.utc().add(options.uploadExpiredSec, 'seconds').toDate(), // optional
(date object with expiration date for urls), default: +60 minutes
        algorithm: 'AWS4-HMAC-SHA256', // optional, default: 'AWS4-HMAC-SHA256'
        //region: 'eu-central-1', // optional, default: s3client.region
        /*conditionMatching: [
            {"success_action_redirect": "http://google.com"},
            ["condition", "key", "pattern"]
        ] */// optional
    };

    try {
        s3client.uploadPostForm(uploadPostFormOptions, function (err, params) {

            params['file_id'] = file_id;

            res.send(200, params);
            next();
        });
    } catch (e) {
        return sendError(e, next);
    }
}

```

```

function s3uploader(s3client, options, req, res, next) {
    log.info("generating link for upload " + req.body.filename + " User : ???");

    let file_id = uuid.v4();

    var uploadPostFormOptions = {
        key: options.inboxPath + "/" + file_id,
        //key: req.body.filename, // required
        bucket: options.bucket, // required
        //extension: 'ext', // optional (pass if You want to check with allowed extensions or
set ContentType)
        acl: 'public-read', // optional, default: 'public-read'
        expires: moment.utc().add(options.uploadExpiredSec, 'seconds').toDate(), // optional
(date object with expiration date for urls), default: +60 minutes
        algorithm: 'AWS4-HMAC-SHA256', // optional, default: 'AWS4-HMAC-SHA256'
        //region: 'eu-central-1', // optional, default: s3client.region
        /*conditionMatching: [
            {"success_action_redirect": "http://google.com"},
            ["condition", "key", "pattern"]
        ] */// optional
    };

    try {
        s3client.uploadPostForm(uploadPostFormOptions, function(err, params) {

            params['file_id'] = file_id;

```



```

        res.send(200,params);
        next();
    });
} catch (e) {
    return sendError(e, next);
}
}

function getLoadboxAttachmentDownloadLink(options, req, res, next) {
    try {
        log.debug("getLoadboxAttachmentDownloadLink",
req.params.containerId,req.params.loadboxId,req.params.attachmentId);

        Container.getLoadbox(req.params.containerId, req.params.loadboxId, function(err,
loadBox) {
            if (err) {
                return sendError(resourceNotFound,next);
            }
            if (loadBox.state == loadBoxAvailable) {
                aws_s3.existsFile(containersPathInS3 + "/" + req.params.containerId +
"/" + req.params.loadboxId + "/" + req.params.attachmentId, function (err, obj) {
                    if (err) {
                        log.debug("File in S3 not found = " +
containersPathInS3 + "/" + req.params.containerId + "/" + req.params.loadboxId + "/" +
req.params.attachmentId);
                        return sendError(new errs.NotFoundError(), next);
                    }
                    Container.getLoadboxAttachment(req.params.containerId,
req.params.loadboxId, req.params.attachmentId, function (err, attachment) {
                        if (err) {
                            return sendError(err, next);
                        }
                        try {
                            let filename = attachment.name;
                            let url =
aws_s3.getDownloadSignedUri(containersPathInS3 + "/" + req.params.containerId + "/" +
req.params.loadboxId + "/" + req.params.attachmentId, filename, options.downloadExpireSec);
                            res.send(url);
                        } catch (e) {
                            return sendError(e, next);
                        }
                    }
                    next();
                });
            });
        });
    } else {
        res.send(resourceNotAvailable);
        next();
    }
});
}

```

```

    } catch (e) {
        sendError(e, next);
    }
}

//ContFileDownload
function getContainerAttachmentDownloadLink(options, req, res, next) {
    log.debug("getContainerAttachmentDownloadLink");

    try {
        // console.log("REQ.PARAMS",req.user)

        Container.getContainer(req.user.cid, function(err, container) {
            if (err) {
                return sendError(resourceNotFound,next);
            }
            // console.log("REQUEST",req);
            if ((container.accessType ==
"WO")||(container.state==containerCreated)||(container.state==containerUnavailable)||(container.sta
te==containerExpired)){
                res.send(resourceNotAvailable);
                next();
            } else {
                Container.getLoadbox(req.user.cid, req.user.lid, function(err,
loadBox) {
                    if (err) {
                        return sendError(resourceNotFound,next);
                    }
                    if ((loadBox.state == loadBoxBlocked)||(loadBox.state ==
loadBoxExpired)) {
                        // console.log("LOADBOX BLOCKED");
                        res.send(resourceNotAvailable);
                        next();
                    } else {
                        aws_s3.existFile(containersPathInS3 + "/" +
req.params.containerId + "/data/" + req.params.attachmentId, function (err, obj) {
                            if (err) {
                                log.debug("File in S3 not found = " +
containersPathInS3 + "/" + req.params.containerId + "/data/" + req.params.attachmentId);
                                return sendError(new
errs.NotFoundError(), next);
                            }
                        }

                        Container.getContainerAttachment(req.params.containerId, req.params.attachmentId,
function (err, attachment) {
                            if (err) {
                                return sendError(err);
                            }

                            if (attachment == null) {

```

```

    errs.NotFoundError());
    "available"){
    aws_s3.getDownloadSignedUri(containersPathInS3 + "/" + req.params.containerId + "/data/" +
    req.params.attachmentId, filename, options.downloadExpireSec);
    return next(new
    } else if (attachment.state ==
        let filename = attachment.name;
        let url =
        res.send(url);
        next();
    } else {
        res.send(resourceNotAvailable);
        next();
    }
    });
    });
    });
    });
    } catch (e) {
        sendError(e, next);
    }
}

```