###### **Функциональная спецификация**

###### **Интерфейс с поставщиками**

**Содержание**

1 Общие положения 3

1.1 Цель документа 3

1.2 Контроль версий 3

1.2.1 Функциональная спецификация 3

1.2.2 Техническая спецификация 3

1.3 Согласование 3

1.4 Описание разработки 3

2 Функциональная спецификация *(заполняется консультантом)* 4

2.1 Общее описание 4

2.2 Бизнес требования 4

2.3 Описание процесса 4

2.4 Требования к интерфейсу пользователя 10

3 Техническая спецификация *(заполняется разработчиком)* 10

3.1 Интерфейс пользователя 10

3.2 Объекты разработки 10

3.3 Алгоритм (необязательно) 10

3.3.1 Шаг N: Название шага (необязательно) 10

# Общие положения

Цель документа

Контроль версий

### Функциональная спецификация

| Версия | Статус | Число | Автор | Описание изменений |
| --- | --- | --- | --- | --- |
| [v1] | [Draft] | 01.09.15 | Михед А. |  |
| [v2] | [Draft] |  |  |  |

### Техническая спецификация

| Версия | Статус | Число | Автор | Описание изменений |
| --- | --- | --- | --- | --- |
| [v1] | [Draft] |  |  |  |
| [v2] | [Draft] |  |  |  |

Согласование

| Имя | Роль на проекте/  позиция в Эльдорадо | Подтвержденная версия | Подпись | Дата |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |

Описание разработки

Для бесстоковой торговли требуется на стороне Поставщика разработать Web-сервис для обмена данными между Поставщиком и «Эльдорадо» – Customer Exchange Interface (CEI).

|  |  |
| --- | --- |
| **Модуль SAP** | SD |
| **Приоритет (**Обязательно/Высокий/Средний/Низкий) | Высокий |
| **Срок исполнения** |  |

# Функциональная спецификация *(заполняется консультантом)*

## Общее описание

Web-сервис должен работать по протоколу SOAP (Simple Object Access Protocol) версии поверх HTTPS. Обмен данными должен вестись посредством XML-сообщений, что позволяет реализовать удаленный вызов процедур (RPC - Remote Procedure Call).

Для удобства работы с CEI размещается WSDL-описание (Web Services Description Language) версии 1.1. — это язык описания Web-сервисов, основанный на XML. На WSDL описываются методы, входные и результирующие структуры данных, типы данных, сетевые адреса для обращения к сервису и т.д.

Web-сервис должен позволять выполнять удаленные запросы к базе данных поставщика для получения актуальных данных, и обработки заказов.

Работа с CEI осуществляется по HTTPS (Hypertext Transfer Protocol Secure) протоколу с использованием авторизации.

На стороне Поставщика должны быть предусмотрены тестовая и продуктивные системы со своими точками входа.

**Перечень методов**

* [GetItemsAvail – доступное количество товаров;](#getItemsAvail)
* [GetMaterialData – получение информации по товару;](#getItemsAvail)
* [GetExcludedDates – справочник дат исключаемых для отгрузки;](#getShipmentDates)
* [GetOrder – чтение заказа;](#getOrderLines)
* [SetOrderCreate – создание заказа;](#setOrderHeaderCommand)
* [SetOrderChange – редактирование заказа;](#setOrderLineCommand)
* [SetSignOrder – подпись заказа;](#setSignOrderCommand)
* [SetDeleteOrder – удаление заказа;](#setDeleteOrderCommand)
* SetFinalOrder – сверка заказа после нарезки;
* [GetOperationResult – результаты выполнения set-запроса;](#getCommandResult)

**Основные параметры**

При подключении Поставщика к процессу Бесстоковой торговли в системе «Эльдорадо» Поставщику будет заведен код кредитора, для его складов – коды заводов. Если у Поставщика физически несколько складов, но информацию по запасам этих складов он транслирует, как общие, и формирование заказа идет одновременно с этих складов, а также машина может привозить товары одновременно с этих складов, то допустимо заводить один код завода. Например, если в Москве у Поставщика физически 2 склада, но при заказе от «Эльдорадо» возможно видеть запас и резервировать его под один заказ, а также машина может забирать товар с обоих складов, а потом везти товар на склад «Эльдорадо», то будет заводиться один код завода. Это должно обговариваться бизнесом. Для заводов в системе «Эльдорадо» предусмотрен четырехзначный код «MX\*\*» (МХ01, МХ02 и т.д.). Коды заводов могут отличаться в тестовой системе и продуктивной, поэтому в системе Поставщика должна быть возможность настройки данного параметра для разных систем. Аналогично и для параметра кода кредитора (Поставщика) - он может отличаться в тестовой и продуктивной системах, для этого поля предусмотрен тип (CHAR 10).

Код кредитора и коды заводов используются при заливке цен по спецификациям и мэппинге между номенклатурами «Эльдорадо» и Поставщика, при запросе остатков и дат исключений. Данные коды присваиваются в системе «Эльдорадо» и сообщаются Поставщику при его подключении или при подключении новых складов в регионах.

Со стороны бизнеса должна быть достигнута договоренность о предоставлении информации по номенклатурам. Результатом этой договоренности является получение Excel-файла (спецификации) с информацией по мэппингу кодов номенклатур «Эльдорадо» и кодов номенклатур Поставщика, с указанием соответствующей группы товара Поставщика, EAN, цен и краткого наименования товаров.

**Общее описание алгоритма**

На стороне «Эльдорадо» предусмотрено несколько программ, которые запускаются периодически: программа обновления остатков, программа получения справочника дат исключений на стороне Поставщика, основная программа и программа нарезки.

Программа обновления остатков ZBST\_STOCK\_UPDATE запускается раз в 30 минут, она анализирует все номенклатуры, заведенные для данного Поставщика с признаком Бесстоковой торговли, группирует номенклатуры по группам товаров и вызывает метод GetItemsAvail по всем найденным группам товаров, сохраняет полученный результат в специальной таблице. Таким образом, в системе «Эльдорадо» обновляется информация по доступным запасам Поставщика.

Программа получения справочника дат исключений на стороне Поставщика ZBST\_GET\_EXCL\_DATES запускается на ежедневной основе, вызывает метод GetExcludedDates, служит для обновления информации справочника дат исключений на стороне Поставщика. При создании клиентских заказов в системе «Эльдорадо» эти исключения, как и исключения со стороны «Эльдорадо», учитываются при расчете даты поставки в клиентских заказах, также это учитывается при расчете даты поставки в заявках на закупку товаров от Поставщика.

Алгоритм основной программы предусматривает отслеживание вновь появившихся потребностей по клиентским заказам с сайта «Эльдорадо», а также изменение в меньшую сторону этих заказов, их отмену. При получении первых потребностей на данную дату в разрезе Поставщиков и их заводов потребности группируются (агрегируются) по номенклатурам. Далее вызывается метод создания заказа SetOrderCreate в системе Поставщика. Т.к. в системе Поставщика возможны задержки с ответом по созданному заказу, и ответ не может быть получен сразу, программа также занимается отправкой запросов на получение результата команд GetOperationResult. В соответствии с полученным ответом по запросу на создание заказа в системе Поставщика, по-разному обрабатываются ситуации с полным комплектом, частичным и полным некомплектам по позициям.

Аналогичным образом обрабатываются изменение клиентских заказав и их отмена. Изменение заказа возможно при добавлении в течение дня новых клиентских потребностей в общий заказ от Поставщика. Редактирование заказа Поставщика осуществляется с помощью вызова метода SetOrderChange и последующей обработки полученных результатов команд методом GetOperationResult. При обработке результата команд следует учитывать, что команды должны обрабатываться на стороне Поставщика в порядке очередности. По каждому GUID команды результат должен сохраняться в системе Поставщика, и должна быть возможность считать данный результат в любой момент времени. Более подробно об обработке очереди команда написано там, где представлено описание метода GetOperationResult.

На стороне Поставщика должен быть организован процесс резервирования товара – при увеличении количества товаров в заказе – зарезервированный запас должен увеличиваться, при изменении количества товаров в меньшую сторону – высвобождаться. При чем, резерв в заказе Поставщика может меняться только по запросу от Эльдорадо.

Для каждого региона в разрезе Поставщиков в системе «Эльдорадо» предусмотрен настроечный параметр «Время фиксации», это то время, при наступлении которого заказ фиксируется и больше меняться не должен. Это время оговаривается между представителями бизнеса Поставщика и «Эльдорадо». От времени фиксации зависит – когда будет подписан и нарезан заказ, и когда он будет отгружен. Например, если текущая дата равна 01.11.15 и, если заказ создан до времени фиксации (для Москвы – до 16.50), то датой заказа будет являться текущая дата, на эту дату будет подписан заказ. Но при этом фактическая дата отгрузки со склада Поставщика может отличаться от данной даты. Если же заказ создан 01.11.15, но после времени фиксации (для Москвы после 16.50), то датой заказа будет являться текущая дата плюс один день = 02.11.15, на эту дату будет подписан заказ. Но фактическая дата отгрузки со склада Поставщика также может отличаться от данной даты.

При наступлении времени фиксации заказ в системе «Эльдорадо» анализируется на существование блокированных клиентских потребностей (неподтвержденных со стороны КЦ и клиента). Если таковые имеются, то они должны быть переложены в новый заказ. Такие позиции в старом заказе помечаются на удаление и генерируются такие же позиции для создания нового заказа в системе Поставщика. Затем отправляется запрос во внешнюю систему на снятие резерва по данным позициям из старого заказа и создание резерва по ним в новом заказе.

Таким образом, после наступления времени фиксации в систему Поставщика отправляется финальное состояние заказа (корзина заказа), после этого момента количество товаров и позиций увеличиваться в данном заказе не может. После получения результата команды на данный запрос, заказ подписывается – в систему поставщика отправляется на подпись заказа. Если при подписи заказа в запрос на подпись попали не все товары, которые были в заказе, то не попавшие товары должны удаляться из заказа Поставщика, далее запас на складе по ним нужно восстановить для возможности последующего резервирования (создание новых позиций, заказа).

После подписи заказа в системе «Эльдорадо» запускается программа нарезки, которая нарезает заказы на поставку с учетом оговоренных с Поставщиком ограничений по объему машин. Каждый заказ на поставку – это одно транспортное средство. В случае, если общий заказ у Поставщика превышает по объему одно транспортное средство, заказ разбивается на столько заказов, сколько нужно в соответствии с данным ограничением. После нарезки в системе «Эльдорадо» в систему Поставщика отправляется запрос SetFinalOrder на сверку заказов, который и содержит информацию о том, как разбился общий заказ (поле PurchaseOrderNumber – будет содержать номер закупочного заказа Эльдорадо). Кроме того, в этом же запросе к каждой позиции отправляется информация о наименовании данного товара для того, чтобы на стороне Поставщика была проведена сверка – соответствуют ли наименования товаров кодам заказанных номенклатур во избежание при поступлении товаров на склад «Эльдорадо» пересорта и претензий.

Если при запросе на сверку заказа в запрос попали не все товары, которые были в заказе, то не попавшие товары должны удаляться из заказа Поставщика, далее запас по ним нужно восстановить.

В ответе по сверке заказа от Поставщика в поле DocumentNumber должен прийти соответствующий номер документа поставщика.

После этого на стороне Поставщика печатаются документы, товар подбирают физически на складе и везут на склад «Эльдорадо» - если предусмотрено договором условие доставки товара.

Заказ может удаляться целиком (например, в случае обнаружения критичных ошибок, которые нельзя исправить другим способом), в этом случае будет вызван метод SetDeleteOrder, после которого в системе Поставщика необходимо восстановить запас.

На стороне Поставщика должна быть предусмотрена обработка ошибочных ситуаций, например, при попытке повторно подписать заказ (повторный запрос от «Эльдорадо» на подпись уже подписанного заказа) должен приходить результат операции с ошибкой.

**WSDL:**



Бизнес требования

| № | Процесс/  Подпроцесс | Описание требования | Выбранное решение | Тип RICEFW |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |

Описание процесса

**методы, общая информация**

Для работы с методами Set\* используется следующая методика: первым шагом выполняется Set\* метод, в котором передаются данные для обработки. Этот метод в случае успешного вызова должен вернуть значение «0» в поле Result и уникальный идентификатор в поле OperationID, по которому на втором шаге будет получена информация о выполнении операции либо, в случае ошибки выполнения запроса значение «1» в поле Result и сообщение об ошибке в поле ErrorMessage.

Далее, через заданный промежуток времени, вызывается метод GetOperationResult с тем же уникальным идентификатором. Этот метод должен вернуть результат выполнения операции, номер заказа поставщика и список позиций заказа для методов [SetOrderCreate,](#setOrderHeaderCommand) [SetOrderChange](#setOrderLineCommand), [SetSignOrder](#setSignOrderCommand), SetFinalOrder.

Параметры помеченные \* - обязательны к заполнению.

**Метод** [**GetItemsAvail**](#getItemsAvail)

Метод для получения доступного количества материалов на складе поставщика. На вход метода подается номер завода Эльдорадо, соответствующего поставщику и дата, а также либо перечень материалов, либо перечень групп материалов. Одна из таблиц обязательно должна быть заполнена. Метод возвращает таблицу с перечнем материалов и доступным количеством. В случае, если на момент запроса количества, запаса у Поставщика этого товара нет, то нулевое количество товара также должно возвращаться в ответе.

>>Входные параметры:

**Werks (char4)\*** – код склада поставщика (используется кодировка Эльдорадо)

**Date (тип даты)\* –** дата на которую требуется получить доступность товара

**MaterialID\_Tab** – таблица номеров материалов

Структура строки:

* MaterialID (char35) – номер материала

**MaterialGroup\_Tab** – таблица групп материалов

Структура строки:

* MaterialGroup (char18) – группа материала

<<Выходные параметры:

**Material\_Tab –** таблица со списком материалов и доступным количеством

Структура строки:

* MaterialID (char35) – номер материала
* AvailableCount (numc9) – доступное количество

**Result (numc1) –** результат запроса (0 – успешно, 1 – ошибка)

**ErrorMessage (string) –** в случае ошибки, содержит сообщение об ошибке

**Метод GetMaterialData**

Метод для получения данных материала поставщика - используется справочно для идентификации товаров.

>>Входные параметры:

**MaterialID** **(char35)** – номер материала

<<Выходные параметры:

**MaterialText (string) –** наименование материала

**MaterialGroup (char18)** – группа материала

**UnitOfMeasurement (char3)** – единицы измерения

**MaterialCharacteristics** – таблица с характеристиками материала

Структура строки:

* CharName (char40) – название характеристики
* CharValue (string) – значение характеристики

**Result (numc1) –** результат запроса (0 – успешно, 1 – ошибка)

**ErrorMessage (string) –** в случае ошибки, содержит сообщение об ошибке

**Метод** [**GetExcludedDates**](#getShipmentDates)

Метод для получения перечня дат, в которые поставщик не производит отгрузку товара. Возвращаемые даты ограничиваются интервалом, заданным во входных параметрах.

>>Входные параметры

**Date\_From (тип даты)\*** – начальная дата интервала

**Date\_To (тип даты)\*** – конечная дата интервала

<<Выходные параметры

**ExcludedDate\_Tab –** таблица дат-исключений

Структура строки:

* ExcludedDate (тип даты)
* ExcludedCode (char10) – код Поставщика БСТ (соответствует коду Поставщика в «Эльдорадо»), или код завода (соответствует коду завода в «Эльдорадо»), по которым будет исключена отгрузка

**Result (numc1) –** результат запроса (0 – успешно, 1 – ошибка)

**ErrorMessage (string) –** в случае ошибки, содержит сообщение об ошибке

**Метод** [**GetOrder**](#getOrderLines)

Метод для чтения заказа в системе поставщика по номеру заказа. В выходных параметрах возвращается таблица с позициями заказа.

>>Входные параметры

**DocumentNumber (numc10)\*** – номер документа поставщика

<<Выходные параметры

**OrderDate (тип даты)** – дата заказа

**OrderItems** – таблица позиций заказа

Структура строки:

* MaterialID (char35) – номер материала
* Quantity (numc9) – зарезервированное в заказе количество

**Result (numc1) –** результат запроса (0 – успешно, 1 – ошибка)

**ErrorMessage (string) –** в случае ошибки, содержит сообщение об ошибке

**Метод** [**SetOrderCreate**](#setOrderHeaderCommand)

Метод для создания заказа поставщика.

>>Входные параметры

**Werks (char4)\*** – код завода (склада) поставщика

**OrderDate (тип даты)\*** – дата заказа. Дата заказа – это собственно та дата, на которую предполагается подпись и нарезка заказа.

**OrderItems\*** – таблица позиций заказа

Структура строки:

* MaterialID (char35) – номер материала
* Quantity (numc9) – затребованное количество

<<Выходные параметры

**OperationID (guid32) -** уникальный идентификатор операции

**Result (numc1) –** результат запроса (0 – успешно, 1 – ошибка)

**ErrorMessage (string) –** в случае ошибки, содержит сообщение об ошибке

**Метод** [**SetOrderChange**](#setOrderLineCommand)

Метод для изменения заказа. На вход передается таблица с позициями заказа для изменения. Если позиции с указанной номенклатурой не существует, создается новая позиция заказа. Если позиция уже существует, то обновляется количество.

>>Входные параметры

**DocumentNumber (numc10)\*** – номер документа поставщика

**OrderItems\*** – таблица позиций заказа

Структура строки:

* MaterialID (char35) – номер материала
* Quantity (numc9) – затребованное количество

<<Выходные параметры

**OperationID (guid32) -** уникальный идентификатор операции

**Result (numc1) –** результат запроса (0 – успешно, 1 – ошибка)

**ErrorMessage (string) –** в случае ошибки, содержит сообщение об ошибке

При чем, в Quantity передается общее количество по позиции, а не дельта. Т.е., например, если изначально создали заказ с данным товаром с количеством – 2шт, то при добавлении 3шт в запросе на изменение заказа придет 5шт – общее количество по данной позиции.

**Метод** [**SetSignOrder**](#setSignOrderCommand)

Метод для «подписи» заказа. Выполняется после времени фиксации, когда прошло финальное подтверждение заказа. На вход подается номер заказа.

В будущем на вход, возможно, будет передаваться таблица с перечнем позиций заказа, которые необходимо перенести в новый заказ. Новый заказ создается на следующую возможную дату поставки, которая передается в поле NewOrderDate. Номер вновь созданного заказа должен возвращаться при получении результата выполнения метода. Сейчас пока эта функциональность не используется.

Если при подписи заказов на стороне «Эльдорадо» часть позиций (или часть количества) не подана на вход (корзина товаров неполная из-за каких-то технических проблем), в этом случае на стороне Поставщика не вошедшие позиции (или количество) не должны больше учитываться в заказе, а затем по ним должен восстанавливаться запас на стороне Поставщика.

>>Входные параметры

**DocumentNumber (numc10)\*** – номер документа поставщика

**OrderItems\*** – таблица позиций заказа для подтверждения

Структура строки:

* MaterialID (char35) – номер материала
* Quantity (numc9) – затребованное количество

**NewOrderDate (тип даты) –** дата заказа для переноса позиций – пока не используется

**ItemsForTransfer** – таблица позиций заказа для переноса в новый заказ – пока не используется

Структура строки:

* MaterialID (char35) – номер материала
* Quantity (numc9) – затребованное количество

<<Выходные параметры

**OperationID (guid32) -** уникальный идентификатор операции

**Result (numc1) –** результат запроса (0 – успешно, 1 – ошибка)

**ErrorMessage (string) –** в случае ошибки, содержит сообщение об ошибке

**Метод** [**SetDeleteOrder**](#setDeleteOrderCommand)

Метод для удаления заказа.

>>Входные параметры

**DocumentNumber (numc10)\*** – номер документа поставщика

<<Выходные параметры

**OperationID (guid32) -** уникальный идентификатор операции

**Result (numc1) –** результат запроса (0 – успешно, 1 – ошибка)

**ErrorMessage (string) –** в случае ошибки, содержит сообщение об ошибке

Если после удаления заказа будет вызван метод чтения заказа, то внешняя система должна выдать сообщение об ошибке в формате:

[**-**](file:///C:\Users\alehina\AppData\Local\Temp\1c1c2bc8_0604HTML000004.xml) <ns1:GetOrder\_Resp\_MT xmlns:ns1="**urn:eldorado.ru:holodilnik.ru:CEI**" xmlns:SOAP-ENV="**http://schemas.xmlsoap.org/soap/envelope/**">

<OrderDate>2015-10-13</OrderDate>

<OrderItems />

<Result>1</Result>

<ErrorMessage>Order 10240 deleted</ErrorMessage>

</ns1:GetOrder\_Resp\_MT>

После удаления заказа все методы Set\* должны получать в ответе ошибку:

[**-**](file:///C:\Users\alehina\AppData\Local\Temp\1c1c2bc8_0803HTML000004.xml) <ns1:SetFinalOrder\_Resp\_MT xmlns:ns1="**urn:eldorado.ru:holodilnik.ru:CEI**" xmlns:SOAP-ENV="**http://schemas.xmlsoap.org/soap/envelope/**">

<OperationID />

<Result>1</Result>

<ErrorMessage>Wrong DocumentNumber</ErrorMessage>

</ns1:SetFinalOrder\_Resp\_MT>

**Метод SetFinalOrder**

Метод для сверки заказа после выполнения программы нарезки. В результате нарезки может быть сформировано несколько закупочных заказов, к примеру, из-за ограничений по объему транспортируемого товара. На вход подается номер заказа поставщика и список позиций с разбивкой в соответствии с созданными закупочными заказами Эльдорадо. Если при этом поставщик делит свой заказ на заказы, соответствующие закупочным заказам Эльдорадо, то в результате выполнения метода требуется вернуть список позиций с новой разбивкой по заказам поставщика и с указанием номеров заказов.

Если в результате нарезки заказов на стороне «Эльдорадо» часть позиций (или часть количества) не нарезалась (корзина товаров неполная из-за каких-то технических проблем), то на вход будут поданы только те позиции с указанием номера закупочного заказа, которые смогли нарезаться. В этом случае на стороне Поставщика ненарезанные позиции (или количество) не должны больше учитываться в заказе, а затем по ним должен восстанавливаться запас.

>>Входные параметры

**DocumentNumber (numc10)\*** – номер документа поставщика

**OrderItems\*** – таблица позиций заказа с указанием номеров закупочных заказов Эльдорадо

Структура строки:

* PurchaseOrderNumber (char10) – номер закупочного заказа Эльдорадо
* MaterialID (char35) – номер материала
* MaterialText (char40) – краткий текст материала Эльдорадо
* Quantity (numc9) – затребованное количество

<<Выходные параметры

**OperationID (guid32) -** уникальный идентификатор операции

**Result (numc1) –** результат запроса (0 – успешно, 1 – ошибка)

**ErrorMessage (string) –** в случае ошибки, содержит сообщение об ошибке

**Метод GetOperationResult**

Метод для получения результата выполнения Set\* методов.

>>Входные параметры

**OperationID (guid32)\* -** уникальный идентификатор операции

<<Выходные параметры

**DocumentNumber (numc10)** – номер документа поставщика

**OrderItems** – таблица позиций заказа

Структура строки:

* DocumentNumber (numc10) – номер документа поставщика (при получении результата метода SetFinalOrder это поле будет содержать № нового документ в системе Поставщика (не прошлый №, который был присвоен в системе Поставщика при создании заказа))
* PurchaseOrderNumber (char10) – номер закупочного заказа Эльдорадо (заполняется только при получении результата метода SetFinalOrder)
* MaterialID (char35) – номер материала
* Quantity (numc9) – зарезервированное количество
* PosResult (numc1) – результат обработки позиции (0 – успешно, 1 – ошибка)
* PosError (string) - в случае ошибки (если PosResult = 1), должно содержать сообщение об ошибке по данной позиции

**Result (numc1) –** результат запроса (0 – успешно, 1 – ошибка, 2 – запрос еще обрабатывается)

**ErrorMessage (string) –** в случае ошибки, содержит сообщение об ошибке

При обработке результата команд следует учитывать, что команды должны обрабатываться на стороне Поставщика в порядке очередности. Если на стороне Поставщика возможны задержки по обработке результата команды, то такая команда должна возвращать в заголовке Result = 2, при этом последующие команды не должны обрабатываться, пока не будет обработана данная команда. Такой же алгоритм заложен на стороне системы «Эльдорадо» в разрезе позиций заказа, т.е. если во внешнюю систему был отправлен запрос на добавление товара1 и товара2 и не получен успешный результат этой команды (Result = 0), то в случае появления новых потребностей в системе «Эльдорадо» по товарам: товара1, товар2, товар3, запрос на добавление товаров в общий заказ Поставщика будет отправлен только на товар3. Потребности по товару1 и товару2 будут «ожидать» результата обработки предыдущего запроса, по ним запрос отправится только после получения Result = 0 по предыдущей команде.

Кроме того, по GUID команды должна быть возможность считывания результата в любой момент, в системе Поставщика должны сохраняться логи с результатами команд.

* Диаграмма процесса (необязательно)
* Алгоритм (необязательно)
* Описание шагов процесса (необязательная таблица)

| Шаг | Название | Тип | Описание и дополнительные комментарии | Ответственный | Система |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Требования к интерфейсу пользователя

# Техническая спецификация *(заполняется разработчиком)*

Интерфейс пользователя

* схема расположения элементов окна

Объекты разработки

| Объект | Тип | Description |
| --- | --- | --- |
|  |  |  |
|  |  |  |

Алгоритм (необязательно)

### Шаг N: Название шага (необязательно)

* Описание (необязательно)