

МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное
образовательное учреждение высшего образования
«Череповецкий государственный университет»

Институт (факультет)	<u>Институт информационных технологий</u>
Направление подготовки (специальность)	<u>09.03.02 Информационные системы и технологии</u>
Выпускающая кафедра	<u>Математического и программного обеспечения ЭВМ</u>

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Название работы	<u>Разработка информационной системы контроля и</u> <u>анализа первичных документов на технологической платформе 1С8</u>
-----------------	---

Студента	<u>Соколовой Светланы Сергеевны</u> Ф.И.О.
----------	---

ДОПУСТИТЬ К ЗАЩИТЕ

Директор института (декан факультета)	<u>Ершов Евгений Валентинович</u>
Заведующий выпускающей кафедрой	<u>Ершов Евгений Валентинович</u>
Руководитель выпускной квалификационной работы	<u>Ганичева Оксана Георгиевна</u>
Нормоконтролер	<u>Виноградова Людмила Николаевна</u>
Выпускник	<u>Соколова Светлана Сергеевна</u>

Введение.....	4
Часть 1 Основная.....	5
1.1 Сравнительный анализ отечественных аналогов проектируемой системы	5
1.2 Выбор технологии, среды и языка программирования.....	7
1.2.1 Выбор модели жизненного цикла.....	7
1.2.2 Выбор подхода к разработке.....	10
1.2.3 Выбор языка и среды программирования.....	12
1.3 Анализ процесса обработки информации, выбор структур данных для её хранения.....	13
1.4 Разработка спецификаций проектируемой системы.....	14
1.4.1 Построение функциональных диаграмм.....	14
1.4.2 Построение диаграмм потоков данных.....	20
1.4.3 Проектирование структур данных и построение диаграмм отношений компонентов данных.....	25
1.4.4 Построение диаграмм переходов состояний.....	29
1.5 Проектирование системы.....	30
1.5.1 Проектирование структуры системы.....	31
1.5.2 Разработка функциональной схемы.....	32
1.5.3 Разработка модульной структуры.....	35
1.6 Проектирование интерфейса пользователя.....	41
1.6.1 Построение графа диалога.....	41
1.6.2 Разработка форм ввода-вывода информации.....	43
1.7..... Выбор стратегии тестирования, разработка тестов, программа и методика испытаний.....	48
1.7.1 Объект и цель испытаний.....	48
1.7.2 Требования к информационному, аппаратно-программному обеспечению и документации.....	49
1.7.3 Состав, порядок и методы испытаний.....	53

1.7.4 Результаты проведения испытаний.....	54
Часть 2. Техничко-экономическое обоснование.....	61
2.1 Создание иерархической структуры работы, выполнение назначения ресурсов.....	61
2.2 Выравнивание загрузки ресурсов.....	61
2.3 Диаграмма Ганта.....	62
2.4 Критический путь.....	63
2.5 Расчет себестоимости программного продукта.....	64
2.6 Расчет цены программного продукта.....	68
2.7 Расчет экономической эффективности.....	69
Заключение.....	72
Список литературы.....	74
ПРИЛОЖЕНИЕ 1 Техническое задание.....	77
ПРИЛОЖЕНИЕ 2 Текст программы.....	86
ПРИЛОЖЕНИЕ 3 Спецификации.....	96
ПРИЛОЖЕНИЕ 4 Руководство пользователя.....	97

В современном обществе своевременная обработка информации способствует совершенствованию организации производства, оперативному и долгосрочному планированию, прогнозированию и анализу хозяйственной деятельности. Каждая организация стремится минимизировать затраты времени, материальных, трудовых ресурсов в ходе своей деятельности и упростить процесс обработки информации. Эти задачи можно решить с помощью использования расширений информационных систем на платформе 1С:Предприятие.

Информационная система (ИС) — система, предназначенная для хранения, поиска и обработки информации, и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию [2,15].

Расширения в 1С представляют собой что-то вроде параллельных конфигураций, которые автоматически объединяются с основной конфигурацией поставщика. Они созданы для облегчения внесения изменений в программу. При использовании расширений, трансформация типовой конфигурации не требуется. Все изменения будут делаться при помощи расширений, которые тоже являются конфигурациями. Таким образом, основная конфигурация останется на полной поддержке [17].

Компания «Северсталь-инфоком» столкнулась с проблемой контроля и анализа первичных документов. Сейчас в компании уходит большое количество времени на поиск и исправление ошибок в документации.

Целью выпускной квалификационной работы является создание информационной системы «Контроль и анализ первичных документов», направленной на повышение качества ввода первичных документов в 1С:Бухгалтерии, с целью минимизировать затраты рабочего времени на предприятии «Северсталь-инфоком» и сократить сроки финансового закрытия, путем обнаружения пользовательских ошибок, в момент занесения информации.

1.1 Сравнительный анализ отечественных аналогов проектируемой системы

На сегодняшний день рынок программного обеспечения (ПО) не богат расширениями для контроля первичных документов. В основном, такие решения разрабатываются на заказ.

Рассмотрим некоторые аналоги отечественных и зарубежных производителей наиболее похожие на проектируемую ИС (расширение).

1) Jetlex - онлайн-сервис для автоматической загрузки первичных документов в 1С. Служба использует машинное обучение, благодаря которому она адаптируется под новые данные почти из любой документации. Продукт после автоматического извлечения данных вводит их в программу 1С или другие системы учета. Использование программы позволяет сократить время заполнения систем бухгалтерского учета в 10 раз [18].

Преимущества Jetlex:

- достаточно высокая скорость обработки документов (5 файлов за 2 минуты);
- есть опция выбора типа документа вручную.

Недостатки Jetlex:

- плохое качество распознавания документов;
- Jetlex достаточно сложно интегрируется с 1С и требует предоставления полного доступа к бухгалтерской системе;
- Jetlex не осуществляет математических проверок документов, не распознает акцизы, ГТД, кассовые чеки, форматы Excel и Word, а также файлы с нарушенным порядком страниц [18].

2) Entera - это сервис для загрузки данных из первичных документов в вашу систему складского или бухгалтерского учета [18].

Преимущества Entera:

- нет необходимости покупать лицензию;
- у программы один из наиболее удобных интерфейсов, позволяющих контролировать точность распознавания (для каждой строки документа появляется всплывающая подсказка с указанием соответствующей строки скана) и проводить групповое сопоставление документов;
- высокая степень точности распознавания нестандартные и нечеткие документы;
- в сервис встроены: математическая проверка таблиц, сверка с базой и проверка банковских реквизитов, что дополнительно минимизирует ошибки;
- большим преимуществом Entera с точки зрения экономии времени является возможность групповой обработки документов.

Недостатки Entera:

- высокое качество распознавания занимает чуть больше времени относительно других программ. Одну страницу придется « подождать » около минуты;
- для работы с сервисом необходимо подключение к интернету, так как он находится на удаленном сервере [18].

3) Efsol, он же 42clouds - одним из первых продуктов по вводу первичной информации в 1с появился на российском рынке [18].

Преимущества Efsol:

- простой запуск и быстрая интеграцией с 1С;
- удобный и простой интерфейс, контроль дубликатов, наличие выбора типа создаваемых документов, единый словарь сопоставлений и математическая проверка табличной части документов;
- быстро распознает страницы хорошего качества и позволяет создавать из них большинство существующих в 1С форм документов.

Недостатки Efsol:

- плохо распознаются документы среднего качества и совершенно не распознаются нечеткие и нестандартные документы;
- интерфейс не оптимизирован, и требуется выполнять множество типовых действий вручную;
- для использования данного сервиса необходимо дополнительно приобретать годовую лицензию на каждое рабочее место [18].

4) Сервис Gendalf – единственный из сравниваемых – работает без подключения к Интернету. Он устанавливается как приложение на конкретный компьютер, при этом требуя от него высоких мощностей: если ресурсов устройства не хватает – программа зависает [18].

Преимущества Gendalf:

- для использования программы приобретается лицензия на каждое рабочее место.

Недостатки Gendalf:

- установка Gendalf достаточно трудоёмкая: требуется создать свежие библиотеки, подготовить информацию для активации и пр. Может потребоваться техническая поддержки специалиста;
- у сервиса отсутствует электронный архив документов и история обработанных файлов [18].

Существующие аналоги решения проблемы оказались неподходящими для данного предприятия. Так как каждый из них работает не достаточно хорошо и точно, контроль происходит в основном печатных документов, а также отсутствуют функции, нужные для компании «Северсталь-инфоком». Разработка собственной ИС для производства предоставит возможность более точно реализовать все необходимые запросы предприятия.

1.2 Выбор технологии, среды и языка программирования

1.2.1 Выбор модели жизненного цикла

Жизненный цикл программного обеспечения (ЖЦ) — период времени, который начинается с момента принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации [10].

На сегодняшний день существует множество моделей жизненного цикла. Некоторые из них представлены в следующем списке:

- каскадная модель разработки программного обеспечения;
- V-образная модель разработки программного обеспечения;
- модель прототипирования жизненного цикла разработки программного обеспечения;
- модель быстрой разработки приложений (Rapid Application Development (RAD));
- инкрементальная модель жизненного цикла разработки программного обеспечения;
- спиральная модель жизненного цикла разработки программного обеспечения.

Модель, выбранная для какого-либо проекта, должна обеспечивать потребности организации, соответствовать типу выполняемых работ, а также навыкам и инструментальным средствам, которые имеются у специалистов-практиков [10].

После рассмотрения различных моделей ЖЦ, для проектирования информационной системы была выбрана каскадная модель жизненного цикла с промежуточным контролем.

Каскадная модель с промежуточным контролем (рис.1) - это итерационная модель разработки ИС с циклами обратной связи между этапами. Преимущество такой модели заключается в том, что межэтапные корректировки обеспечивают меньшую трудоемкость по сравнению с каскадной моделью,

однако время жизни каждого из этапов растягивается на весь период разработки [10].



Рис. 1. Каскадная модель с промежуточным контролем

Модель характеризуется следующими свойствами взаимодействия этапов:

- модель состоит из последовательно расположенных этапов (точно так же, как и «водопад»);
- каждый этап имеет обратную связь с предыдущими этапами;
- исправление ошибок происходит на каждом из этапов, сразу при выявлении проблемы - это промежуточный контроль;
- этапы перекрываются во времени по причине наличия обратной связи: следующий этап не начинается, пока не завершится предыдущий; при первом проходе по модели вниз, как только обнаружена ошибка, осуществляется возврат снизу вверх, к предыдущим этапам, которые повлекли ошибку. Таким образом, фактически этапы оказываются растянутыми во времени;
- результат появляется только в конце разработки, как и в модели «водопад».

Критерием появления результата при модели «водопад» является приемлемое качество продукта, то есть такое состояние продукта, когда наиболее критические для клиента ошибки устранены, а с наличием не принципиальных для жизнедеятельности системы ошибок клиент согласился

- данные ошибки описаны в документации и фактически переведены таким образом в разряд особенностей системы [10].

1.2.2 Выбор подхода к разработке

На сегодняшний день для проектирования ИС, существует две основные методологии (подхода), принципиальное различие между которыми обусловлено разными способами декомпозиции систем. Первый подход называют функционально-модульным или структурным. В его основу положен принцип функциональной декомпозиции, при которой структура системы описывается в терминах иерархии ее функций и передачи информации между отдельными функциональными элементами [5].

Второй, объектно-ориентированный подход, использует объектную декомпозицию. При этом структура системы описывается в терминах объектов и связей между ними, а поведение системы описывается в терминах обмена сообщениями между объектами [5].

Для проектирования данной информационной системы используется структурный подход.

Структурный подход базируется на ряде общих принципов:

- принцип решения сложных проблем путем их разбиения на множество меньших независимых задач, легких для понимания и решения;
- принцип иерархического упорядочивания, т.е. принцип организации составных частей проблемы в иерархические древовидные структуры с добавлением новых деталей на каждом уровне;
- принцип структурирования данных: заключается в том, что данные должны быть структурированы и иерархически организованы [14].

Преимущества данного подхода заключаются в следующем:

- возможность проведения глубокого анализа бизнес-процессов, выявления узких мест: комплексное применение позволяет выявить все возможные рассогласования и неточности;

- применение универсальных графических языков моделирования IDEF0 (Icam DEFinition for Function Modeling), IDEF3 и DFD (Data Flow Diagrams) обеспечивает логическую целостность и полноту описания, необходимую для достижения точных и непротиворечивых результатов;

- проверенность временем и широкое распространение среди аналитиков и разработчиков [8].

Структурный подход включает в себя следующие методологии:

- IDEF0. Методология функционального моделирования IDEF0 – это технология описания системы в целом как множества взаимосвязанных действий, или функций. Важно отметить функциональную направленность IDEF0 – функции системы исследуются независимо от объектов, которые обеспечивают их выполнение [7].

Наиболее часто IDEF0 применяется как технология исследования и проектирования систем на логическом уровне. По этой причине он, как правило, используется на ранних этапах разработки проекта [8].

- IDEF3. Способ описания процессов, основной целью которого является обеспечение структурированного метода, используя который эксперт в предметной области может описать положение вещей как упорядоченную последовательность событий с одновременным описанием объектов, имеющих непосредственное отношение к процессу [8].

- DFD. Диаграммы потоков данных моделируют систему как набор действий, соединенных друг с другом стрелками. Диаграммы потоков данных также могут содержать два новых типа объектов: объекты, собирающие и хранящие информацию – хранилища данных и внешние сущности – объекты, которые моделируют взаимодействие с теми частями системы, которые выходят за границы моделирования [8].

Вся документация оформляется в соответствии с ГОСТ 19.101-77 Единая система программной документации (ЕСПД).

Таким образом, для разрабатываемой ИС более подходит структурный подход. При его использовании система разбивается на функциональные

подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи и так далее. Процесс разбиения продолжается вплоть до конкретных процедур. При этом автоматизируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимосвязаны.

1.2.3 Выбор языка и среды программирования

Для разработки информационной системы используется язык программирования 1С и технологическая платформа «1С:Предприятие».

Платформа «1С:Предприятие» — основа для запуска «конфигураций» или прикладных решений, разработанных программистами 1С. Ее гибкость позволяет объединить в единой информационной среде решения для различных задач по организации управления бизнесом. Таким образом, сама система «1С:Предприятие» состоит из технологической платформы и различных конфигураций. Пользователь чаще всего работает с каким-то прикладным решением, а не с самой платформой. Она лишь служит базой для создания, запуска, изменения конфигураций и работы с ними. Так на базе одной информационной среды можно автоматизировать разные виды деятельности [21].

Конфигурируемость — главное свойство системы «1С:Предприятие». Она дает возможность менять типовые прикладные решения под нужды бизнеса, дорабатывать их, учитывая специфику предприятия, и создавать новые версии конфигураций самостоятельно. Платформа не зависит от отрасли, в которой работает компания, она универсальна для всех конфигураций [21].

Встроенный язык программирования 1С:Предприятие — язык программирования, который используется в семействе программ «1С:Предприятие». Данный язык является интерпретируемым языком высокого уровня. Интерпретация текста программного модуля в байт-код выполняется в момент обращения к этому модулю в процессе работы, таким образом обычно интерпретируется только часть текстов программных модулей [22].

Таким образом, исходя из вышесказанного, можно сделать вывод, что платформа «1С: Предприятие» и встроенный язык программирования подходят для разработки данной ИС.

1.3 Анализ процесса обработки информации, выбор структур данных для её хранения

Информационная система «Контроль и анализ первичных документов» разрабатывается для предприятия «Северсталь-инфоком» с целью повышения качества ввода первичных документов в 1С:Бухгалтерии и минимизацией затрат рабочего времени.

На сегодняшний момент, оперативной проверки первичных документов на предприятии нет. Процесс ведения бухгалтерского учета ведется следующим образом (рис.2). Пользователь заносит первичные документы в информационную систему предприятия. Далее при закрытии отчетности за период, бухгалтер может получить список неточностей и несоответствий информации в этих документах. Для исправления этих ошибок уходит большое количество времени. Некоторые виды бумаг после исправления нужно отправлять обратно контрагентам, чтобы он соответствующим образом скорректировал свой бухгалтерский учет.

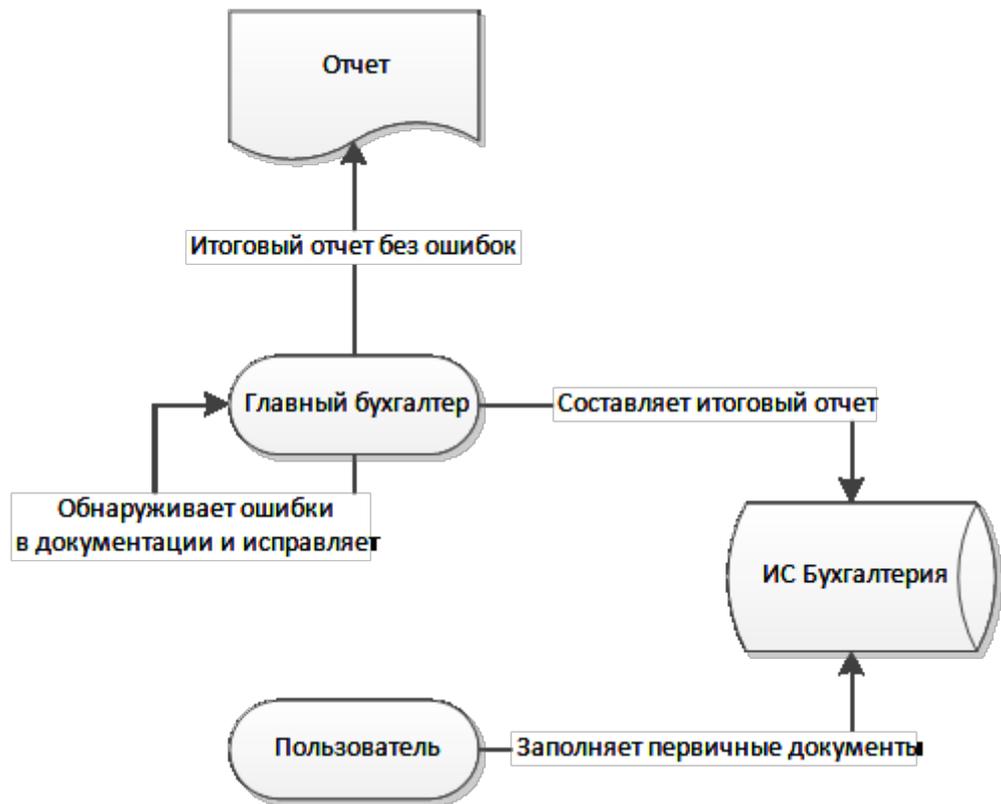


Рис. 2. Процесс занесения первичной документации на предприятии
Разрабатываемая ИС будет выполнять следующий функционал:

- настройка условий контроля изменений реквизитов любого документа;
- любое количество условий контроля для одного документа;
- настройка сообщения, которое будет выводиться пользователю;
- настройка пользователей, для которых будет выполняться контроль.

Информационная система позволит проверять корректность заполнения реквизитов и исключить ошибки ввода данных (например, менеджер указал в документе с подразделением А склад от подразделения Б или перевел заказ в состояние к отгрузке, а в заказе еще не были выполнены какие-то условия):

Решение позволит ограничить допустимые значения реквизитов. Например, можно каждому пользователю разрешить продажу только с определенных складов, подразделений и организаций. А также разрешить пользователям редактировать только свои документы, в которых они указаны как ответственные.

С появлением информационной системы контроля и анализа первичных документов процесс закрытия отчетности пойдет гораздо быстрее. И будет иметь следующие преимущества:

- сокращение времени на обмен документами;
- снижение ошибок из-за исключения человеческого фактора;
- сокращение времени на обработку документов;
- сокращение времени на подготовку и отправку документации в инспекцию федеральной налоговой службы (ИФНС);
- высокая безопасность и конфиденциальность электронного документооборота.

1.4 Разработка спецификаций проектируемой системы

1.4.1 Построение функциональных диаграмм

Методология SADT (Structured Analysis and Design Technique) – это совокупность правил и процедур, предназначенных для построения функциональной модели (схемы) предметной области разрабатываемой информационной системы [9].

Функциональная модель отображает функциональную структуру объекта, то есть производимые им действия и связи между этими действиями [9].

Моделирование процессов в нотации IDEF0 начинается с создания так называемой контекстной диаграммы (рис.3). Эта диаграмма описывает деятельность организации или процесса в целом [16].



Рис. 3. Контекстная диаграмма IDEF0

В результате проектирования системы были выделены основные элементы. Входными данными в систему будут являться первичные документы и данные администратора. На выходе из системы будет выдаваться или сообщение пользователю о недопустимых действиях, или сохраненные корректно заполненные документы. Механизмом управления ИС является регламентированная форма заполнения документов на предприятии, т.е. правила разработанные компанией по тому, как должны быть заполнены документы. В роли исполнения выступает пользователь, администратор.

Затем, на рис. 4 представлена декомпозиция контекстной диаграммы IDEF0 на три функциональных блока. Блок «Настройка условий контроля» - в нем администратор выбирает необходимый документ, создает условие контроля. В табличной форме выбирает реквизиты и значения реквизитов, по которым будет анализироваться данные первичных документов. Далее блок «Внесение первичной информации» пользователь заполняет первичный документы. Следующим этапом происходит обработка данных – внесенная, пользователем информация анализируется в соответствии с установленными условиями контроля.

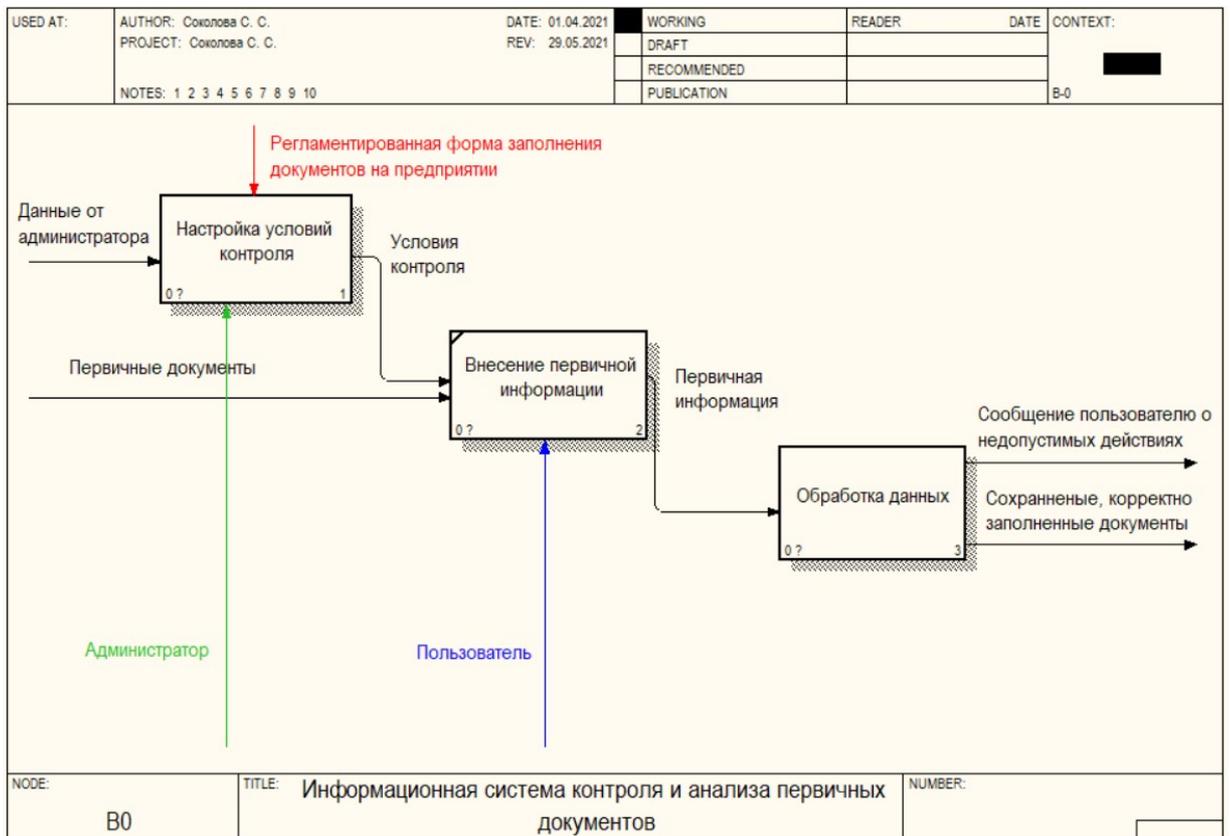


Рис. 4. Декомпозиция контекстной диаграммы IDEF0

На рис. 5 показана декомпозиция блока «Настройка условий контроля» на три более простых блока. Первым делом, администратор выбирает необходимый документ, в котором данные должны контролироваться. После этого откроется табличная форма, для заполнения реквизитов, значений и выбора пользователей, для которых будет действовать настраиваемое условие. Далее администратор вносит текст сообщения, которое будет выдаваться, если пользователь внесет данные отличные от условий контроля.

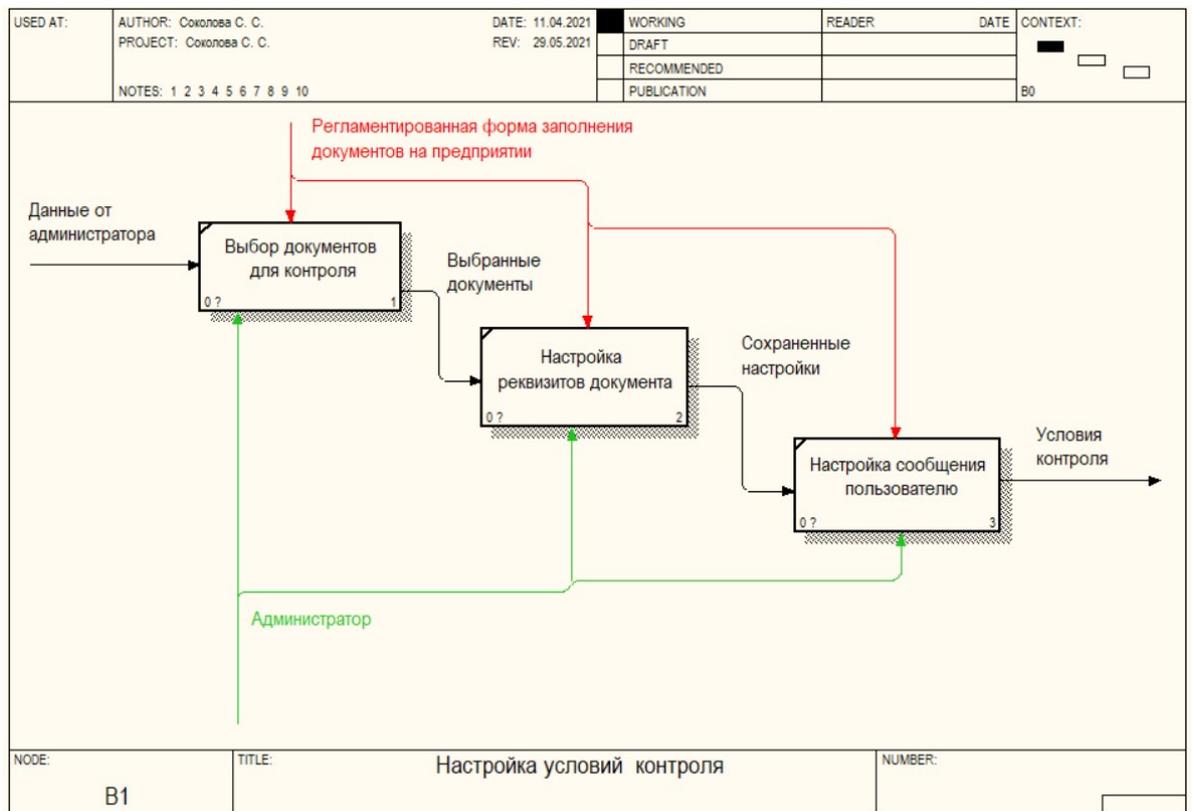


Рис. 5. Декомпозиция функционального блока «Настройка условий контроля»

«Обработка данных» также декомпозирован на три более простых блока (рис.6). После того, как пользователь ввел информацию и пытается сохранить документ, происходит ее обработка. Сначала проверяется, все ли поля документа заполнены. Далее происходит анализ введенной информации на основе условий контроля, после чего, проанализированная информация или сохраняется, если все данные удовлетворяют настроенным условиям, или пользователю высвечивается сообщение о некорректных действиях.

Дальнейшая декомпозиция блоков не требуется, на этом можно закончить разработку модели IDEF0.

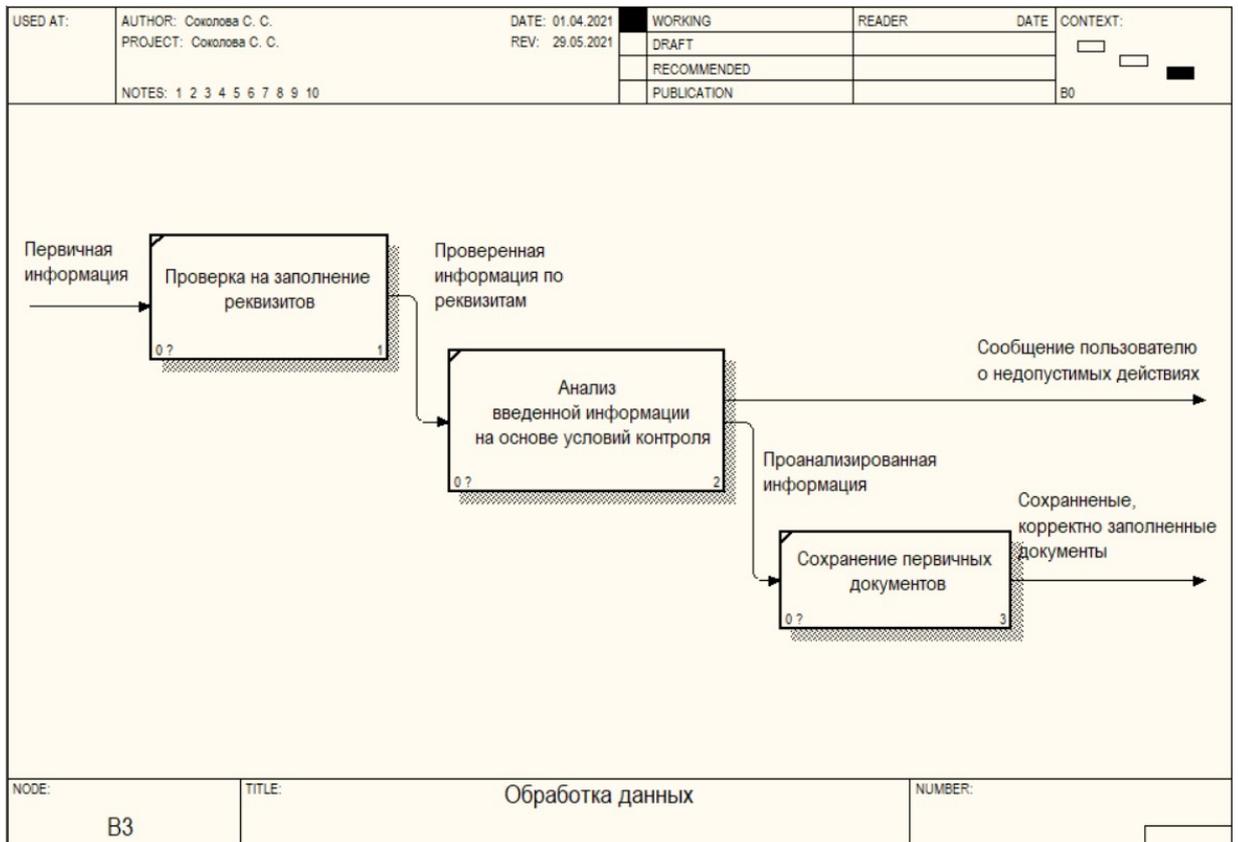


Рис. 6. Декомпозиция функционального блока «Обработка данных»

В процессе построения диаграмм была описана вся детализированная информация, представленная в описании элементарных функций (табл.1) и в словаре терминов (табл. 2).

Таблица 1

Описание элементарных функций IDEF0

Элементарный процесс	Описание функций
1	2
Выбор документов для контроля	Выбор конкретных документов, по которым будет производиться контроль с установленными настройками
Настройка реквизитов документа	Выбор конкретных реквизитов документа, по которым будет производиться контроль и анализ в соответствующих документах
Настройка сообщения пользователю	Текст сообщения, выводимого пользователю при ошибке, является настраиваемым

1	2
Внесение первичной информации	Блок предназначен, для занесения первичных документов
Проверка на заполнение реквизитов	Происходит проверка, заполнено ли поле реквизита или нет
Анализ введенной информации на основе условий контроля	Служит для обработки данных по запросу из класса «База данных» и передачи на представление. Введенные данные анализируются на правильность заполнения форм на основе установленных настроек контроля
Сохранение первичных документов	Сохранение документов происходит, если они заполнены корректно, согласно условиям контроля. Если нет, но вызывается сообщение для пользователя

Таблица 2

Словарь терминов IDEF0

Данные	Описание
1	2
Первичные документы	Содержит электронный документ, свидетельствующий о каких-либо деловых или финансовых транзакциях, то есть значимый бухгалтерский документ, используемый для подготовки, проверки и аудита финансовой отчетности организации
Данные от администратора	Содержит список контрольных настроек. Какому пользователю можно/нельзя редактировать документ. Значение, какого реквизита доступно для определенного документа, а какое нет
Условия контроля	Содержит настройки условий контроля, которые формирует администратор
Первичная информация	Содержит введенные данные (значения реквизитов документов) в информационную систему
Регламентированная форма заполнения документов на предприятии	Содержит правила разработанные компанией, по нормам заполнения документации
Администратор	Пользователь подсистемы администрирования ИС, обладающий расширенным набором прав и выполняющий настройку контроля документов.

1	2
Пользователь	Человек, работающий в системе «Контроль и анализ пользовательских действий», реагирующий на ее сообщения
Сообщение пользователю недопустимых действиях	Всплывающее уведомление с предупреждением, при недопустимых действиях пользователя
Сохраненные, корректно заполненные документы	Документы прошедшие обработку данных, в результате которой, не было выявлено разногласий с настройками администратора
Выбранные документы	Документы, по которым будет производиться контроль с соответствующими контрольными настройками
Сохраненные настройки	Таблица, заполненная администратором, и содержащая информацию о контрольных настройках
Проверенная информация по реквизитам	Содержит документы, проверенные на заполнение полей реквизитов
Проанализированная информация	Содержит документы, по которым не выявлено недочетов в соответствии с условиями контроля

1.4.2 Построение диаграмм потоков данных

Цель DFD диаграммы – продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами. Необходимость использования DFD-диаграмм заключается в потребности описать существующие в структуре организации потоки данных [16].

Основными компонентами диаграмм потоков данных являются:

- внешние сущности;
- системы и подсистемы;
- процессы;
- накопители данных;
- потоки данных.

Контекстная DFD-диаграмма чаще всего состоит из одного функционального блока и из одной или нескольких внешних сущностей (рис.7).

Для «Информационная система контроля и анализа первичных документов» составим DFD (рис. 7-8). На контекстной диаграмме показана система в целом и взаимодействие её с внешним миром (рис. 7).

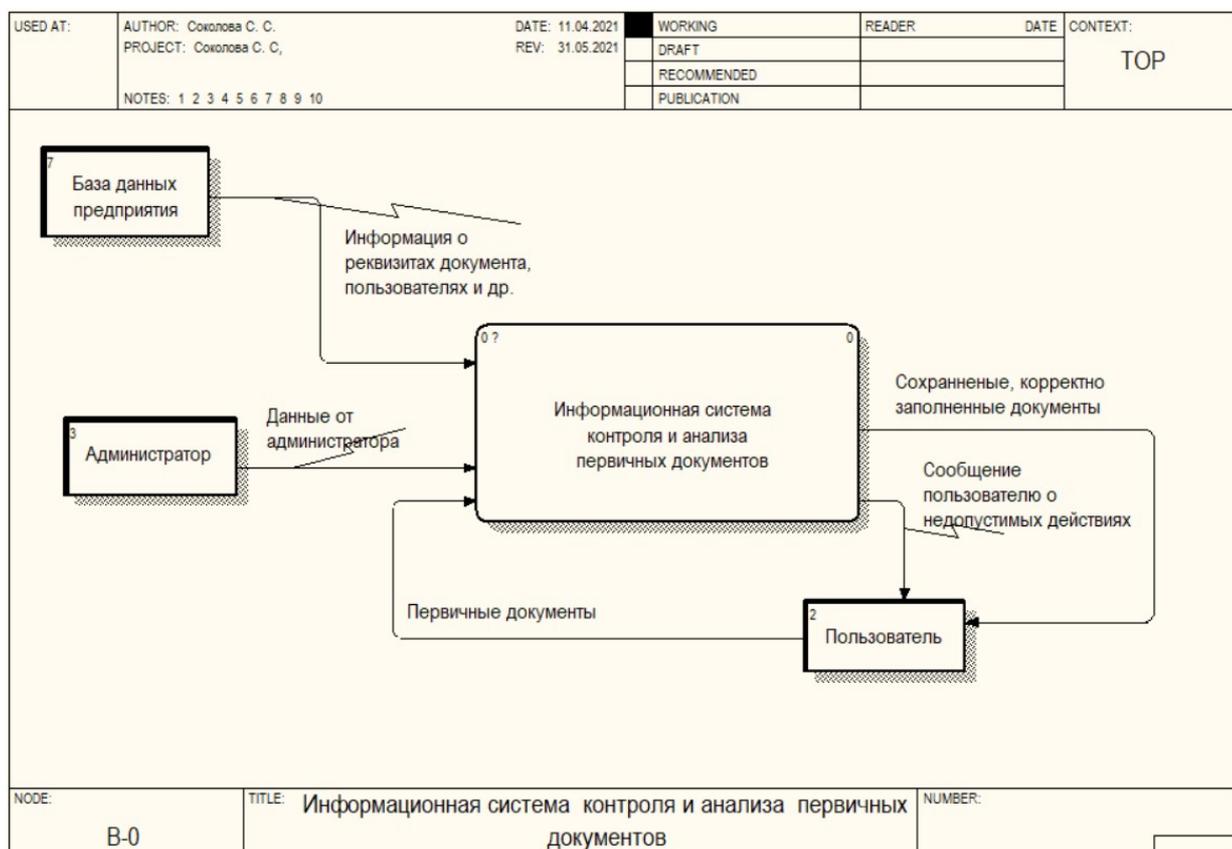


Рис. 7. Контекстная DFD-диаграмма

Диаграмма потоков данных (DFD) - это граф, на котором показано движение значений данных от их источников через преобразующие их процессы к их потребителям в других объектах [9,16].

Информация о реквизитах документа, пользователей и др.; данные администратора, первичные документы это исходные данные, которые предоставляют внешние сущности: БД предприятия, администратор и пользователь. Стрелки, выходящие справа: сохраненные, корректно заполненные документы; сообщение пользователю о недопустимых действиях – это результат работы системы.

Данную диаграмму можно декомпозировать на следующие блоки (рис.8):

1. Настройки условия контроля;
2. Внесение первичной информации;
3. Обработка данных.

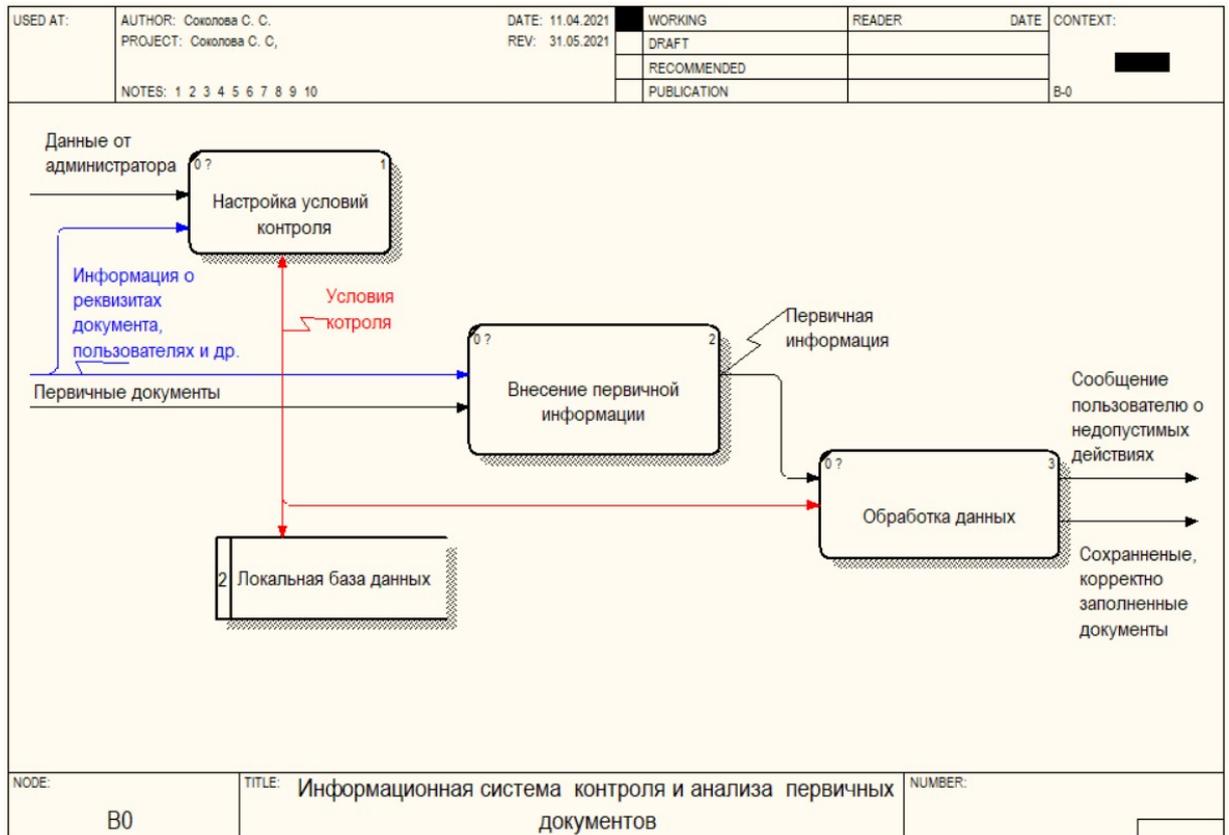


Рис. 8. Декомпозиция контекстной DFD-диаграммы

Главной особенностью декомпозиции DFD-диаграммы является то, что на ней отображается хранилище – локальная база данных. Оно хранит условия контроля, внесенные администратором, а также предоставляет эти условия при обработке информации, введенной пользователем.

Описание элементарных процессов представлено в табл. 3.

Таблица 3

Описание элементарных процессов DFD

Элементарный процесс	Описание
Настройка условий контроля	Основной процесс информационной системы. Входными данными являются список контрольных настроек для дальнейшего анализа, вносимый администратором, и информация о реквизитах, пользователях и др. данных ИС предприятия. Контроль можно настроить на изменение реквизитов и для любых других документов
Внесение первичной информации	Пользователь вносит в информационную систему первичную информацию
Обработка данных	Служит для обработки данных по запросу из класса «База данных» и передачи на представление. Введенные данные анализируются на правильность заполнения форм

Словарь терминов представлен ниже в табл. 4.

Таблица 4

Словарь терминов DFD

Данные	Описание
1	2
Данные от администратора	Содержит список контрольных настроек. Какому пользователю можно/нельзя редактировать документ. Значение, какого реквизита доступно для определенного документа, а какое нет
Условия контроля	Содержит настройки условий контроля, которые формирует администратор
Первичные документы	Содержит электронный документ свидетельствующий о каких-либо деловых или финансовых транзакциях, то есть значимый бухгалтерский документ, используемый для подготовки, проверки и аудита финансовой отчетности организации
Первичная информация	Содержит введенные данные (значения реквизитов документов) в информационную систему

1	2
Информация о реквизитах документа, пользователях и др.	Содержит информацию о реквизитах документах, пользователях и других данных хранящихся в базе данных предприятия
Сообщение пользователю о недопустимых действиях	Всплывающее уведомление с предупреждением, при недопустимых действиях пользователя
Сохраненные, корректно заполненные документы	Документы прошедшие обработку данных, в результате которой, не было выявлено разногласий с настройками администратора

Описание хранилищ представлено в табл. 5.

Таблица 5

Описание хранилищ DFD

Название	Описание
Локальная база данных	Хранит условия контроля, внесенные администратором, и обработанную информацию

Описание внешних сущностей представлено в табл. 6.

Таблица 6

Описание внешних сущностей DFD

Внешняя сущность	Описание
Администратор	Пользователь подсистемы администрирования ИС, обладающий расширенным набором прав и выполняющий настройку контроля документов.
Пользователь	Человек, работающий в системе «Контроль и анализ пользовательских действий», реагирующий на ее сообщения
База данных предприятия	Представляет собой информацию о всех реквизитах, пользователях, и др данных ИС предприятия

1.4.3 Проектирование структур данных и построение диаграмм отношений компонентов данных

ERwin является одним из CASE средств, позволяющих моделировать бизнес процессы. ERwin обеспечивает интеграцию моделей верхнего уровня с моделями нижнего уровня. Модели верхнего уровня разрабатываются на начальных стадиях проектирования информационных систем. Модели нижнего уровня разрабатываются на этапе создания программного кода и тестирования [4].

Диаграмма ERwin строится из трех основных блоков - сущностей, атрибутов и связей. Если рассматривать диаграмму как графическое представление правил предметной области, то сущности являются существительными, а связи - глаголами. Выбор между логическим и физическим уровнем отображения осуществляется через линейку инструментов или меню [20]. Внутри каждого из этих уровней есть следующие режимы отображения:

- Режим "сущности" - внутри прямоугольников отображается имя сущности (для логической модели) или имя таблицы (для физического представления модели); служит для удобства обзора большой диаграммы или размещения прямоугольников сущностей на диаграмме.

- Режим "определение сущности" служит для презентации диаграммы другим людям.

- Режим "атрибуты". При переходе от предметной области к модели требуется вводить информацию о том, что составляет сущность. Эта информация вводится путем задания атрибутов (на физическом уровне - колонок таблиц). В этом режиме прямоугольник-сущность делится линией на две части - в верхней части отображаются атрибуты (колонки), составляющие первичный ключ, а в нижней - остальные атрибуты. Этот режим является основным при проектировании на логическом и физическом уровнях.

- Режим "первичные ключи" - внутри прямоугольников - сущностей показываются только атрибуты/колонки, составляющие первичный ключ.

- Режим "пиктограммы". Для презентационных целей каждой таблице может быть поставлена в соответствие пиктограмма.
- Режим "показ глагольной фразы". На дугах связей показываются глагольные фразы, связывающие сущности (для логического уровня) или имена внешних ключей (для физического уровня) [4].

ERwin не содержит полного алгоритма нормализации и не может проводить нормализацию автоматически, однако его возможности облегчают создание нормализованной модели данных. Поэтому для начала необходимо выполнить нормализацию.

База данных (БД) спроектированная в CASE-средстве ERWin представлена на рис. 9.

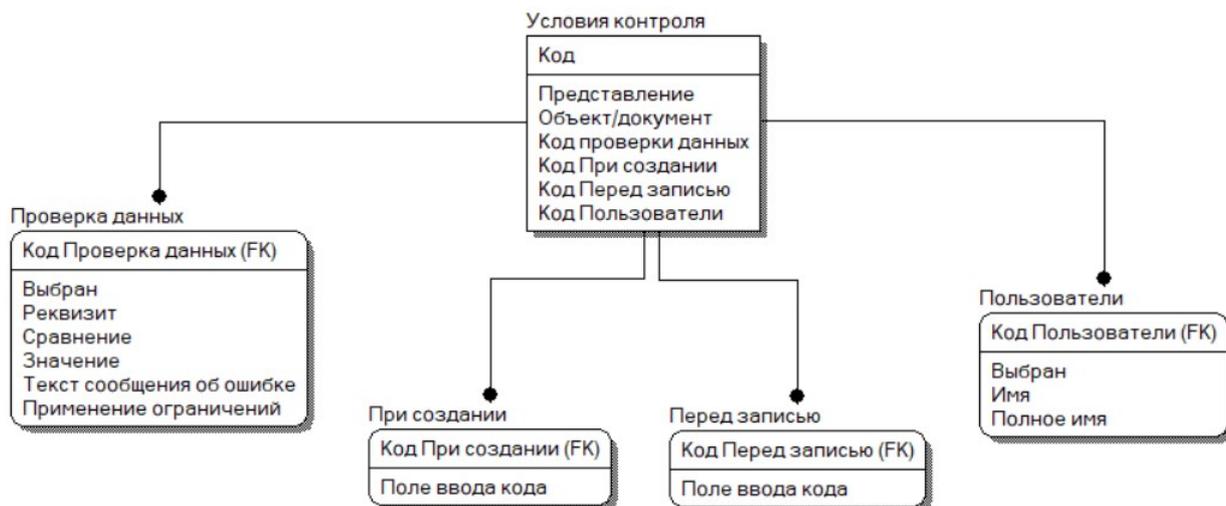


Рис. 9. ER-диаграмма БД

Атрибуты для каждой сущности БД представлены в табл. 7 – 11.

Таблица 7

Атрибуты сущности «Условия контроля»

Название атрибута	Тип данных	Размерность	Назначение
1	2	3	4
Код	Number	Длинное целое	Уникальный идентификатор условия

Продолжение табл. 7

1	2	3	4
---	---	---	---

Представление	Varchar2	100	Краткое название условия
Объект/документ	Varchar2	100	Краткое название документа
Код Проверка данных	Number	Длинное целое	Уникальный идентификатор данных проверки
Код При создании	Number	Длинное целое	Уникальный идентификатор кода в поле «При создании»
Код Перед записью	Number	Длинное целое	Уникальный идентификатор кода в поле «Перед записью»
Код Пользователи	Number	Длинное целое	Уникальный идентификатор списка пользователей для условия контроля

Таблица 8

Атрибуты сущности «Проверка данных»

Название атрибута	Тип данных	Размерность	Назначение
1	2	3	4
Код Проверка данных	Number	Длинное целое	Уникальный идентификатор данных проверки
Выбран	Boolean	True (истина, «да», логическая единица «1»); false (ложь, «нет», логический ноль «0»)	Обозначение включено условие контроля или нет
Реквизит	Char	Произвольный символ	Название реквизита, по которому будет производиться контроль
Сравнение	Data layout comparison view	-	Уникальный идентификатор проверки

1	2	3	4
Значение	Char	Произвольный символ	Название значения, по которому будет производиться контроль
Текст сообщения	Varchar2	1000	Уникальный идентификатор кода в поле «Перед записью»
Применение ограничения	Boolean	True (истина, «да», логическая единица «1»); false (ложь, «нет», логический ноль «0»)	Обозначение включенного ограничения

Таблица 9

Атрибуты сущности «При создании»

Название атрибута	Тип данных	Размерность	Назначение
Код При создании	Number	Длинное целое	Уникальный идентификатор текста кода при создании
Поле вода кода	Varchar2	1000	Текс кода для дополнительных функций контроля

Таблица 10

Атрибуты сущности «Перед записью»

Название атрибута	Тип данных	Размерность	Назначение
Код Перед записью	Number	Длинное целое	Уникальный идентификатор текста кода перед записью
Поле вода кода	Varchar2	1000	Текс кода для дополнительных функций контроля

Атрибуты сущности «Договор»

Название атрибута	Тип данных	Размерность	Назначение
Код Пользователи	Number	Длинное целое	Уникальный идентификатор списка пользователей для условия контроля
Выбран	Boolean	True (истина, «да», логическая единица «1»); false (ложь, «нет», логический ноль «0»)	Обозначение, для каких пользователей действует условие контроля
Имя	Varchar2	100	Записываются имена пользователей системы
Полное имя	Varchar2	100	Записываются полные имена пользователей системы

1.4.4 Построение диаграмм переходов состояний

На этапе анализа требований и определения спецификации диаграмма переходов состояний демонстрирует поведение разрабатываемой программной системы при получении управляющих воздействий [12].

Под управляющими воздействиями или сигналами в данном случае понимают управляющую информацию, получаемую системой извне, например, управляющими воздействиями считают команды пользователя и сигналы датчиков, подключенных к компьютерной системе. Получив такое управляющее воздействие, разрабатываемая система должна выполнить определенные действия, а затем, или остаться в том же состоянии, или перейти в другое состояние, зафиксировав некоторые изменения в системе [12].

После запуска системы на экран выводится главное окно программы. После этого идет переход к расширению «Контроль и анализ первичных документов». Из состояния ожидания возможен переход в состояния «Настройка условий контроля» по команде «Открыть настройки условий контроля», «Внесение

первичных документов» по команде «Открыть формы внесения документов» и переход в состояние завершения.

Перейдя в состояние «Настройка условий контроля», возможен переход в состояние «Выбор документов для контроля», где можно изменить параметры. Из предыдущего состояния можно перейти в «Выбор параметров контроля» и добавить или изменить данные, а также ввести текст сообщения пользователю.

Перейдя в состояние «Внесение первичных документов», можно изменить, добавить или удалить данные, а также при введении некорректных данных, пользователь получит сообщение об ошибке.

Диаграмма переходов состояний представлена на рис.10.

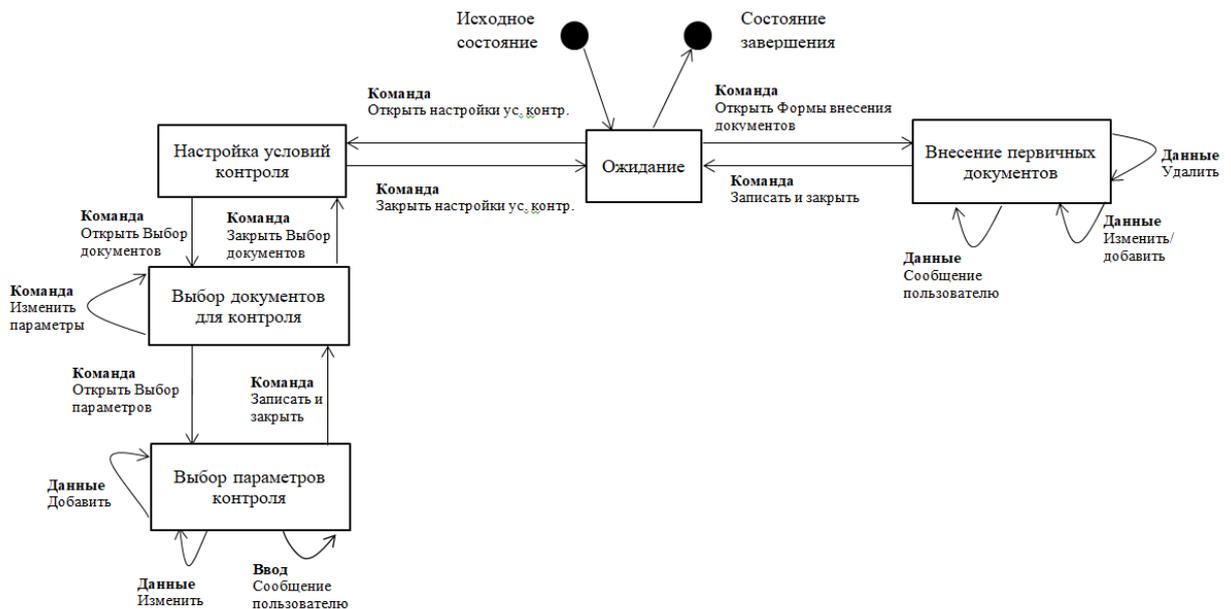


Рис. 10. Диаграмма переходов состояний

1.5 Проектирование системы

Процесс проектирования сложного программного обеспечения начинают с уточнения его структуры, т. е. определения структурных компонентов и связей между ними. Результат уточнения структуры может быть представлен в виде структурной и/или функциональной схем и описания (спецификаций) компонентов [11].

1.5.1 Проектирование структуры системы

Структурной называют схему, отражающую состав и взаимодействие по управлению частей разрабатываемого программного обеспечения [12].

Структурная схема проектируемой информационной системы представлена на рис. 11.



Рис. 11. Структурная схема системы

«ИС контроля и анализа первичных документов» состоит из трех подсистем: настройка условий контроля, внесение первичной информации, обработка данных. В первой формируется список документов, в которых будет происходить контроль, далее выбираются реквизиты и значения для анализа, а также можно ввести сообщение, которое будет получать пользователь. Вторая подсистема предоставляет информацию введенную пользователем. В третьей подсистеме идет обработка внесенных данных. Проверяется, какие реквизиты заполнены, и анализируются на основе настроенных условий контроля.

Описание структурной схемы представлено в табл. 12

Таблица 12

Описание структурной схемы

Наименование	Описание
1	2
Подсистема Настройка условий контроля	Предоставляет перечень условий, контрольных настроек для дальнейшего анализа, и информацию о реквизитах, пользователях и др. данных ИС предприятия

1	2
Подсистема Внесение первичной информации	Предоставляет информацию из первичных документов, внесенных пользователем
Подсистема Обработка данных	Предоставляет анализ первичной информации, на основе введенных условий контроля
Выбор документов для контроля	Составление списка документов, которые будут проходить контроль системы
Настройка реквизитов документа	Составление списка реквизитов, которые будут проходить контроль системы
Настройка сообщения пользователю	Написание сообщения пользователю, которое будет высвечиваться при выявлении ошибки
Проверка на заполнение реквизитов	Проверяется, все ли реквизиты заполнены, которые были указаны в «Настройке реквизитов для контроля»
Анализ введенной информации на основе условий контроля	Служит для обработки данных по запросу из класса «База данных» и передачи на представление. Введенные данные анализируются на правильность заполнения форм на основе установленных настроек контроля
Сохранение первичных документов	Сохранение документов происходит, если они заполнены корректно, согласно условиям контроля. Если нет, но вызывается сообщение для пользователя

1.5.2 Разработка функциональной схемы

Функциональные схемы, более информативны, чем структурные. Все компоненты структурных и функциональных схем должны быть описаны. При структурном подходе особенно тщательно необходимо прорабатывать спецификации межпрограммных интерфейсов, так как от качества их описания зависит количество самых дорогостоящих ошибок. К самым дорогим относятся ошибки, обнаруживаемые при комплексном тестировании, так как для их устранения могут потребоваться серьезные изменения уже отлаженных текстов [7].

Функциональная схема или схема данных – схема взаимодействия компонентов программного обеспечения с описанием информационных

потоков, состава данных в потоках и указанием используемых файлов и устройств [7].

Функциональная схема информационной системы (рис.12) соответствует структурной схеме.

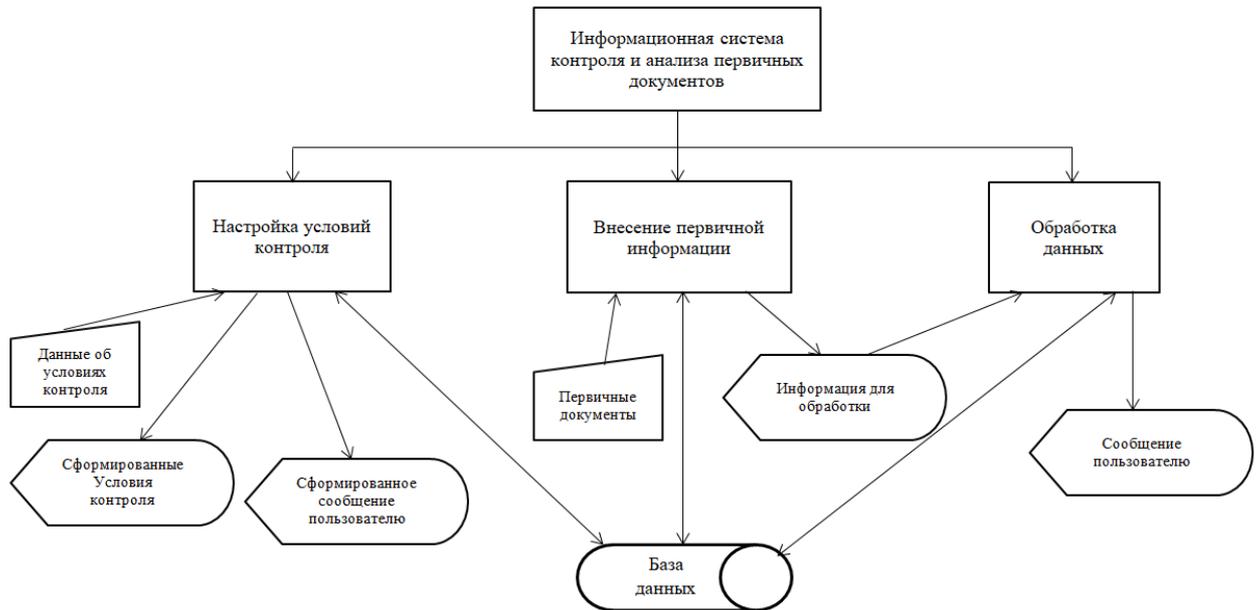


Рис. 12. Функциональная схема информационной системы

Администратор вводит данные об условиях контроля в подсистему «Настройка условий контроля», которые в дальнейшем хранятся в БД. В результате происходит вывод условий контроля и сообщения пользователю. Пользователь вносит первичные документы в систему, которые тоже сохраняются в БД, откуда в дальнейшем эта информация отправляется в обработчик данных. После обработки, если в первичной информации нашлись несоответствия условиям контроля, то на экран высветится сообщение для пользователя. Если первичные документы заполнены согласно условиям контроля, то они сохраняются в БД.

Полное описание функциональной схемы представлено в табл. 13.

Таблица 13

Описание функциональной схемы

Наименование	Тип блока	Описание
Настройка условий контроля	Процесс	Предоставляет перечень условий, контрольных настроек для дальнейшего анализа, и информацию о реквизитах, пользователях и др. данных ИС предприятия
Внесение первичной информации	Процесс	Предоставляет информацию из первичных документов, внесенных пользователем
Обработка данных	Процесс	Предоставляет анализ первичной информации, на основе введенных условий контроля
Данные об условиях контроля	Ручной ввод	Предоставляет набор документов, реквизитов, пользователей и т.п, по которым будет производиться анализ и контроль
Первичные документы	Ручной ввод	Предоставляет ввод информации из первичных документов (сотрудник, контрагент, склад, товар, дата и др.)
Обработанная информация	Ручной ввод	Получение обработанной информации
Сформированные условия контроля	Дисплей	Вывод списка контрольных настроек по которым будет производиться контроль и анализ
Сформированное сообщение пользователю	Дисплей	Вывод текста сообщения пользователю
Информации для обработки	Дисплей	Вывод наименования первичного документа, реквизитов и др., для дальнейшего анализа и контроля
Вывод сообщения пользователю	Дисплей	Вывод текста сообщения при несоответствии действий пользователя и условий контроля
База данных	Запоминающее устройство с прямым доступом	Хранит в себе список условий контроля, первичную информацию и текст сообщения пользователю

1.5.3 Разработка модульной структуры

Модуль – фрагмент программного текста, являющийся строительным блоком для физической структуры системы. Как правило, модуль состоит из интерфейсной части и части-реализации [7].

Базовыми строительными блоками программной системы являются модули. Все виды модулей в любом языке программирования имеют ряд общих свойств, нижеперечисленные из которых существенны при структурном проектировании [7]:

- модуль состоит из множества операторов языка программирования, записанных последовательно;
- модуль имеет имя, по которому к нему можно ссылаться как к единому фрагменту;
- модуль может принимать и/или передавать данные как параметры в вызывающей последовательности или связывать данные через фиксированные ячейки или общие области.

Модульная структура представлена на рис. 13, описание модульной структуры в табл. 14.

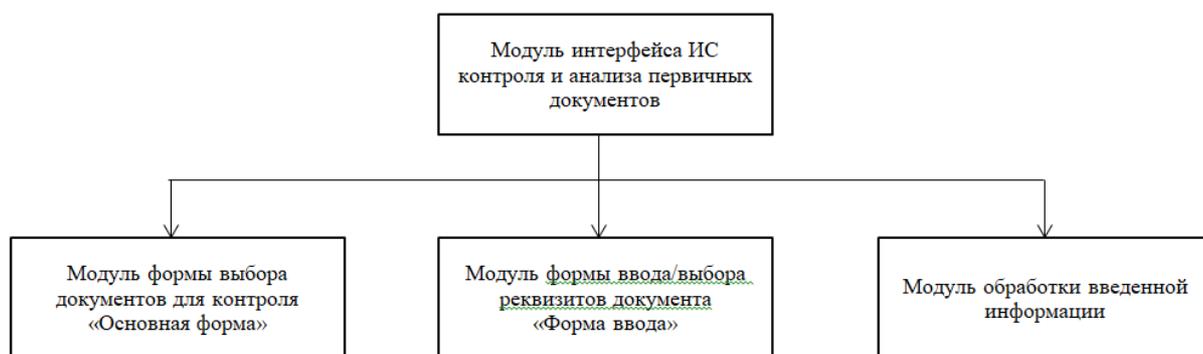


Рис. 13. Модульная структура информационной системы

Разрабатываемая ИС состоит из четырех модулей. Главный модуль «Модуль интерфейса ИС контроля и анализа первичных документов» обеспечивает взаимодействие модулей «Модуль формы выбора документов для контроля «Основная форма», «Модуль формы ввода/выбора реквизитов документа

«Форма ввода» «Модуль обработки введенной информации» между собой, и предоставляет через пользовательский интерфейс пользователю, все функции ИС. Модуль формы выбора документов для контроля «Основная форма» предназначен для формирования списка условий, по которым происходит контроль и анализ первичных документов. Модуль формы выбора документов для контроля «Основная форма» для формирования самих условий контроля. Модуль обработки введенной информации предназначен для контроля и анализа первичной информации, на основе введенных условий контроля.

Таблица 14

Описание модульной структуры

Наименование	Входные данные	Выходные данные	Описание
1	2	3	4
Модуль интерфейса ИС контроля и анализа первичных документов	-	Форма интерфейса пользователя	Модуль обеспечивает взаимодействие всех модулей между собой, разграничение доступа, предоставление пользовательского интерфейса к функциям системы
Модуль формы выбора документов для контроля «Основная форма»	Все возможные документы, по которым будет производиться анализ и контроль	Список условий контроля для каждого документа, которым будет производиться контроль и анализ. Текст сообщения пользователю	Модуль предназначен для формирования списка условий, по которым происходит контроль и анализ первичных документов

1	2	3	4
Модуль формы ввода/выбора реквизитов документа «Форма ввода»	Информация из первичных документов (сотрудник, контрагент, склад, товар, дата и др.)	Наименование условия контроля, и сформированные условия для дальнейшего анализа и контроля	Модуль предназначен для формирования самих условий контроля
Модуль обработки введенной информации	Первичная информация	Сохраненные документы, или же, при введении некорректных данных, получение сообщения об ошибке	Модуль предназначен для контроля и анализа первичной информации, на основе введенных условий контроля

Для анализа технологичности полученной иерархии модулей целесообразно использовать структурные карты Константайна или Джексона.

На структурной карте отношения между модулями представляют в виде графа, вершинам которого соответствуют модули и общие области данных, а дугам - межмодульные вызовы и обращения к общим областям данных [7].

Карта Константайна для анализируемой модульной структуры представлена на рис. 14.

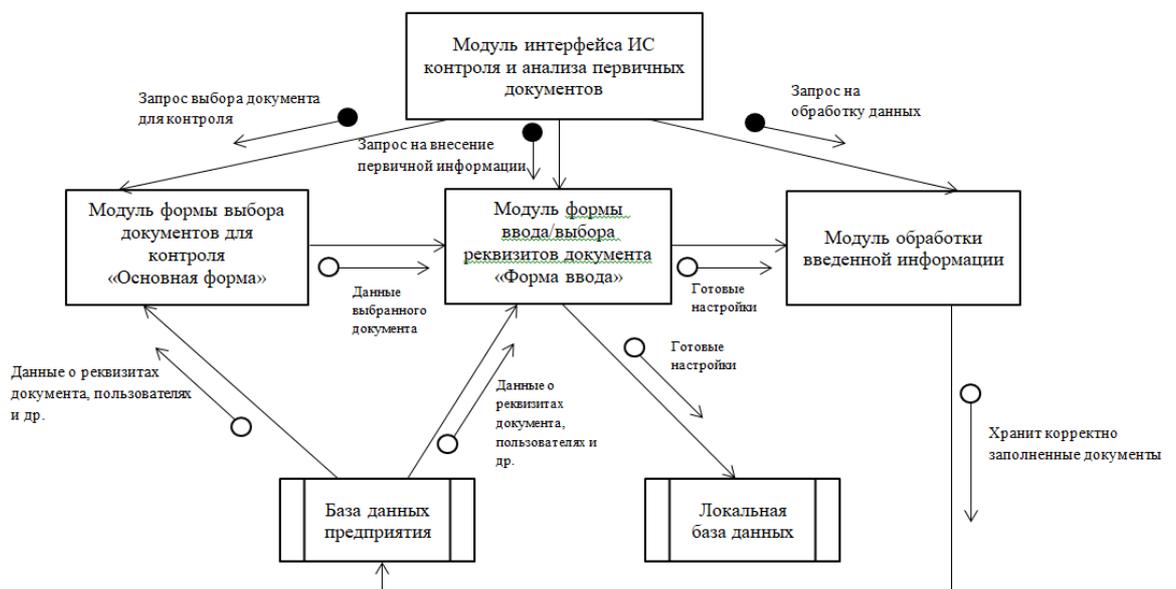


Рис. 14. Карта Константайна информационной системы

Степень независимости модулей оценивают двумя критериями: сцеплением и связностью [19].

Сцепление является мерой взаимозависимости модулей, которая определяет, насколько хорошо модули отделены друг от друга [19].

Связность - мера прочности соединения функциональных и информационных объектов внутри одного модуля. Если сцепление характеризует качество отделения модулей, то связность характеризует степень взаимосвязи элементов, реализуемых одним модулем [19].

В программных системах должны присутствовать модули, имеющие следующие три меры связности: функциональная, последовательная и информационная, так как другие типы связности являются крайне нежелательными и осложняют понимание и сопровождение системы [19].

Информация о связности модулей представлена в табл. 15.

Таблица 15

Связность модулей

Наименование модуля	Вид связи
1	2
Модуль формы выбора документов для контроля «Основная форма»	Информационная, функциональная
Модуль формы ввода/выбора реквизитов документа «Форма ввода»	Информационная, функциональная
Модуль обработки введенной информации	Последовательная, функциональная

Информация о сцеплении модулей представлена в табл. 16

Таблица 16

Сцепление модулей

Наименование модуля	Вид сцепления
Модуль формы выбора документов для контроля «Основная форма»	По данным
Модуль формы ввода/выбора реквизитов документа «Форма ввода»	По данным
Модуль обработки введенной информации	По данным

Полученная модульная структура является оптимальной по критериям связности и сцепления модулей.

Подробная спецификация модулей ИС представлена в табл. 17.

Таблица 17

Спецификация модулей ИС

Имя модуля	Заголовок процедуры или функции	Формальные параметры	Выполняемое действие
1	2	3	4
Модуль формы выбора документов для контроля «Основная форма»	Процедура ЗаполнитьДеревоОбъектов()	-	Заполняется список всех документов и справочников, имеющихся в ИС предприятия
	Процедура ЗаполнитьСписокНастроек	знач СтрокаДерева = Неопределено, знач ИмяОбъекта, ИмяОсновногоПользователя = Неопределено	Заполнение списка настроек у каждого документа соответственно
	Процедура ДобавитьОткрытьНастройку	ИдентификаторОбъекта, ИдентификаторНастройки, Отказ = Ложь	Служит для добавление новой настройки условий контроля
	Функция ПолучитьНовыйИдентификаторНастройки()	-	Функция для определения наименования настройки
	Процедура УдалитьНастройкуНаСервере	ИдентификаторОбъекта, ИдентификаторНастройки	Служит для удаления настроек из списка

1	2	3	4
	Процедура ПересохранитьНастро йкиНаСервере()	-	Служит для пересохранения настройки условий контроля, при внесенных изменениях
Модуль формы ввода/выбора реквизитов документа «Форма ввода»	Процедура ПриСозданииНаСерве ре	Отказ, Стандартная Обработка	Вызывается при Открытии формы заполнения новой настройки условий
	Функция ЗаписатьНаСервере()	-	Возникает при нажатии кнопки «Записать», служит для сохранения условия контроля
	Процедура ЗаписатьИЗакреть	Команда	Служит для сохранения и закрытия формы заполнения условия контроля
	Процедура ИнициализироватьКом поновщикНастроек()	-	Создается новая настройка условий контроля
	Процедура СнятьФлажкиУВсехП ользователей	Команда	Служит для отмены выбора всех пользователей
	Процедура ПоставитьФлажкиУВс ехПользователей	Команда	Служит для выбора всех пользователей
Модуль обработки введенной информации	Функция ПолучитьСтруктуруНа строек	КлючНастро ек="", ИмяПользова теля = Неопределен о, Идентификат орНастройки =""	Вызывается при анализе информации, введенной пользователем, и условий контроля, установленными администратором

1	2	3	4
	Процедура ПриСозданииНаСервере	Процедура ПриСозданииНаСервере	Служит для анализа (сравнения) заполненных полей документа и условий контроля
	Процедура ПередЗаписьюНаСервере	Форма, Отказ, ТекущийОбъект, ПараметрыЗаписи, ИмяОсновногоОбъекта="Объект"	Служит для обработки данных документа на основе кода, введенного администратором
	Процедура ПриЗаписиНаСервере	Форма, Отказ, ТекущийОбъект, ПараметрыЗаписи	Служит для обработки данных документа на основе кода, введенного администратором
	Функция СсылкаСоответствуетНастройкамКомпоновки	КлючНастройки, СсылкаНаЭлемент, НастройкиКомпоновщика	Служит для сохранения корректно заполненных документов

Текст программы представлен в прил.2 и спецификация на программное изделие представлена в прил.3.

1.6 Проектирование интерфейса пользователя

1.6.1 Построение графа диалога

Пользовательский интерфейс – элементы и компоненты программы способные оказывать влияние на взаимодействие пользователя с программным обеспечением [13]. В том числе:

- средства отображения информации;

- командные режимы, язык «пользователь – интерфейс»;
- устройства и технологии ввода данных;
- диалоги, взаимодействия между пользователем и компьютером;
- обратная связь с пользователем.

Под диалогом в данном случае понимают регламентированный обмен информацией между человеком и компьютером, осуществляемый в реальном масштабе времени и направленный на совместное решение конкретной задачи: обмен информацией и координация действий. Каждый диалог состоит из отдельных процессов ввода-вывода, которые физически обеспечивают связь пользователя и компьютера [13].

Граф диалога — ориентированный взвешенный граф, каждой вершине которого сопоставлена конкретная картинка на экране (кадр) или определенное состояние диалога, характеризующееся набором доступных пользователю действий. Дуги, исходящие из вершин, показывают возможные изменения состояний при выполнении пользователем указанных действий. В качестве весов дуг указывают условия переходов из состояния в состояние и операции, выполняемые во время перехода [13].

Граф диалога представлен на рис.15. Как видно, сначала отображается начальная страница информационной системы. Переход к настройке условий контроля происходит через вкладку «Администрирование» - > «Сервис» - > «Настройки условий контроля». В этом окне можно добавлять или удалять условия, а также, нажав на интересующий документ, можно перейти в окно «Форма выбора параметров контроля». В ней можно переходить на любую вкладку («Проверка данных», «При создании», «Перед записью», «Пользователи») табличной формы, чтобы внести реквизиты, значения реквизитов, по которым будет происходить анализ. Так же в них можно прописать программный код, для дополнительных настроек, и выбрать пользователей, для которых будет действовать данное условие контроля.

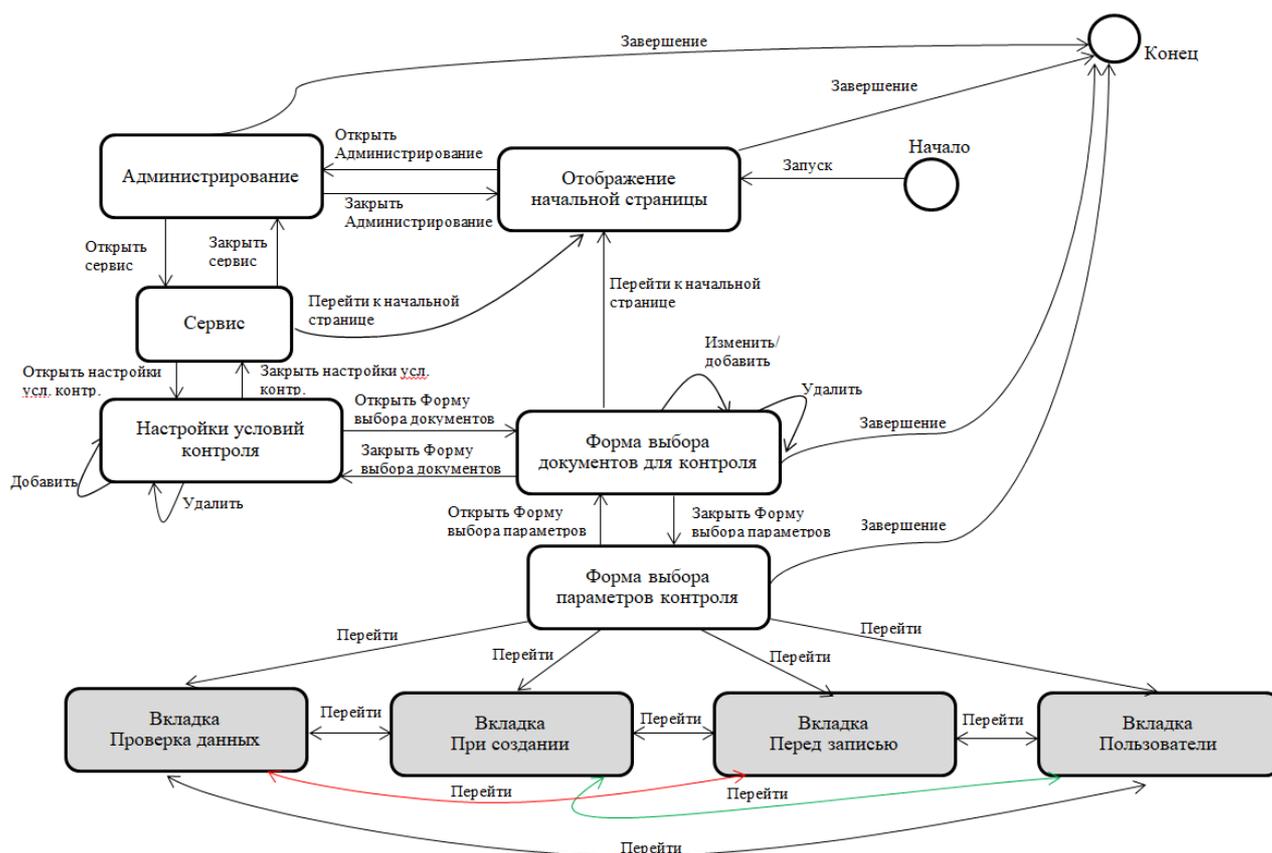


Рис.15. Граф диалога информационной системы

1.6.2 Разработка форм ввода-вывода информации

Разрабатывать информационную систему необходимо в программе «1С: Предприятие 8.3» с интерфейсом «Такси». Отличительными особенностями интерфейса «Такси» являются крупный шрифт, улучшенная навигация, адаптация под стандарты веб-приложений, а также максимизация рабочего пространства. Пользователи программ «1С: Предприятие 8.3» смогут настроить внешний вид системы под себя и расположить панели так, как удобно.

К основным элементам разрабатываемой подсистемы относятся:

1. Главное окно ИС (расширения) (рис. 16). На главном окне расположены: панель разделов с подсистемами, панель инструментов, строка поиска, настройки главного окна, строка состояния. Главное окно предназначено для перехода в один из элементов подсистемы. Кнопка «Сервис» содержит переход к списку «Настройки условий контроля».

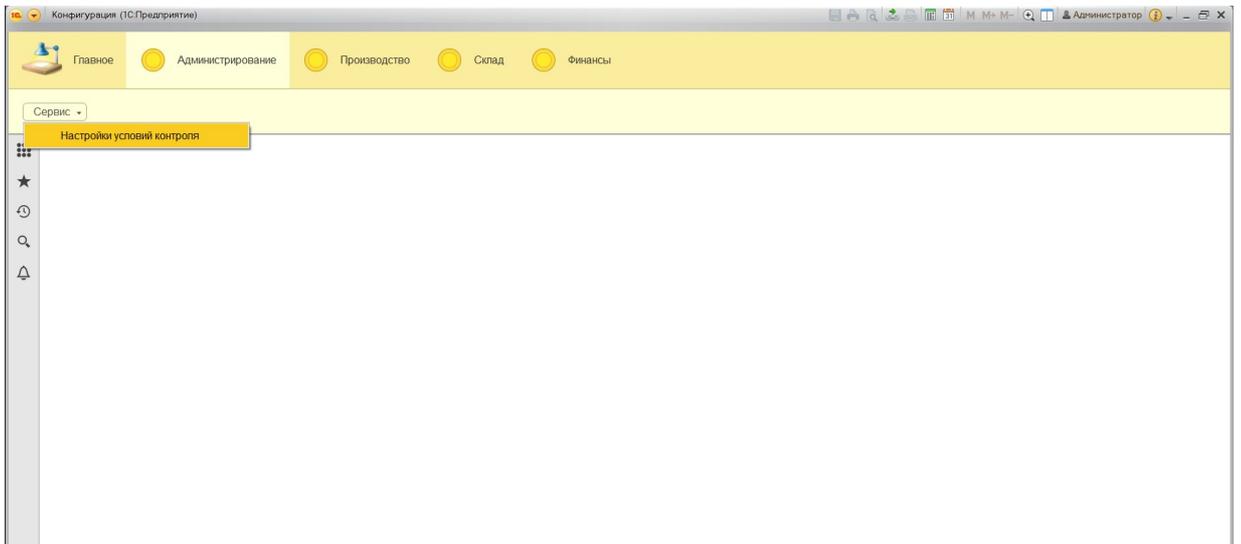


Рис. 16. Интерфейс платформы

2. Для того чтобы перейти к настройкам контроля первичных документов нужно нажать «Настройки условий контроля», откроется окно со всеми возможными документами и справочниками по которым будет производиться контроль (рис. 17).

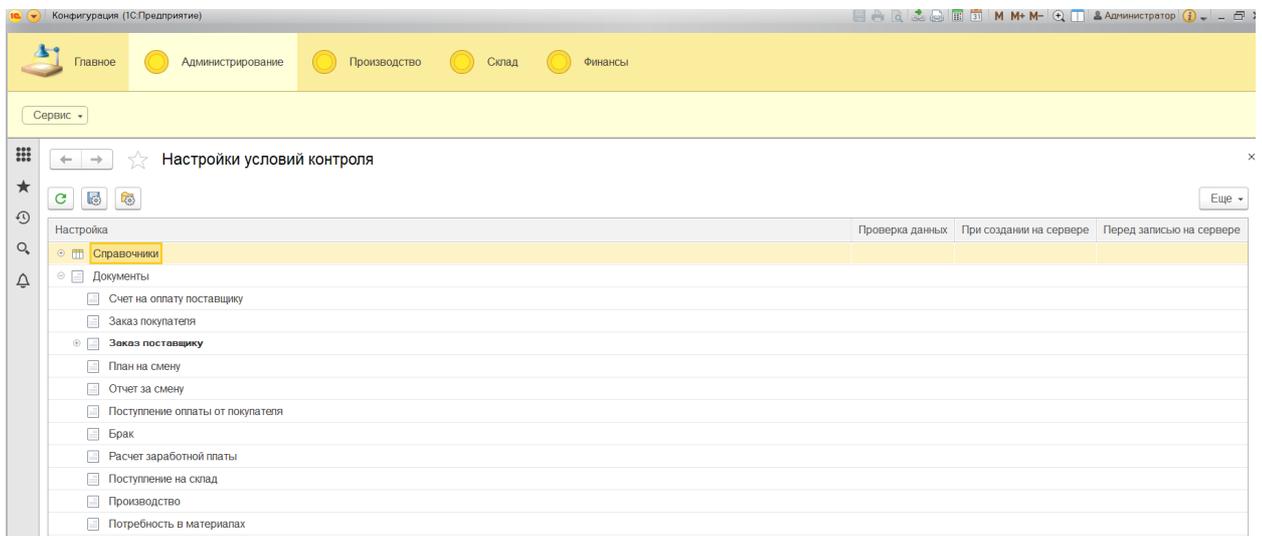


Рис. 17. Переход к «Настройки условий контроля»

3. Окно занесения информации первичных документов (рис. 18). При нажатии на дубой документ Откроется окно занесения условий контроля по данному документу. Включает в себя заполнения реквизиты, строки, отдельные страницы с заполнением, кнопки. Строка «Представление» предназначена для названия настройки. Справа находится блок ограничений для соответствующего

условия настройки. Он имеет флажки, которые устанавливаются для выбора ограничения.

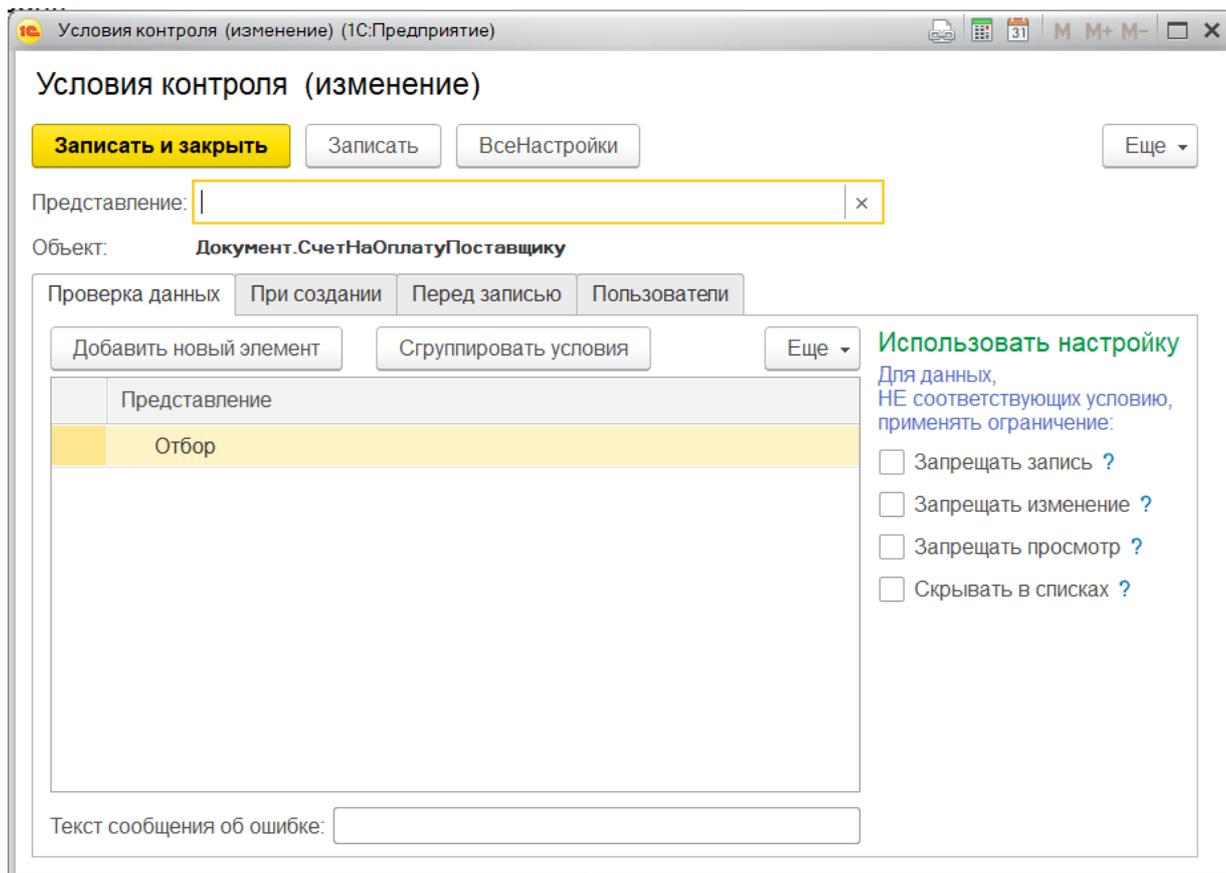


Рис. 18. Окно занесения информации первичных документов

На рис. 19-20 представлены вкладки «Перед записью» и «При создании», она содержит текстовое поле, предназначенное для ввода кода дополнительных условий контроля.

Вкладка «Пользователи» (рис. 21) имеет табличную часть с полями «Имя» и «Полное имя», предназначенные для занесения наименования пользователей системы. В поле «Выбран» указывается флажок у пользователя, для которого будут действовать данные условия контроля.

Условия контроля (изменение) (1С:Предприятие)

Записать и закрыть Записать ВсеНастройки Еще ▾

Представление: ×

Объект: **Документ.СчетНаОплатуПоставщику**

Проверка данных При создании **Перед записью** Пользователи

Текст перед записью на сервере:

Рис.19. Вкладка «Перед записью»

Условия контроля (изменение) (1С:Предприятие)

Записать и закрыть Записать ВсеНастройки Еще ▾

Представление: ×

Объект: **Документ.СчетНаОплатуПоставщику**

Проверка данных **При создании** Перед записью Пользователи

Текст при создании на сервере:

Рис.20. Вкладка «При создании»

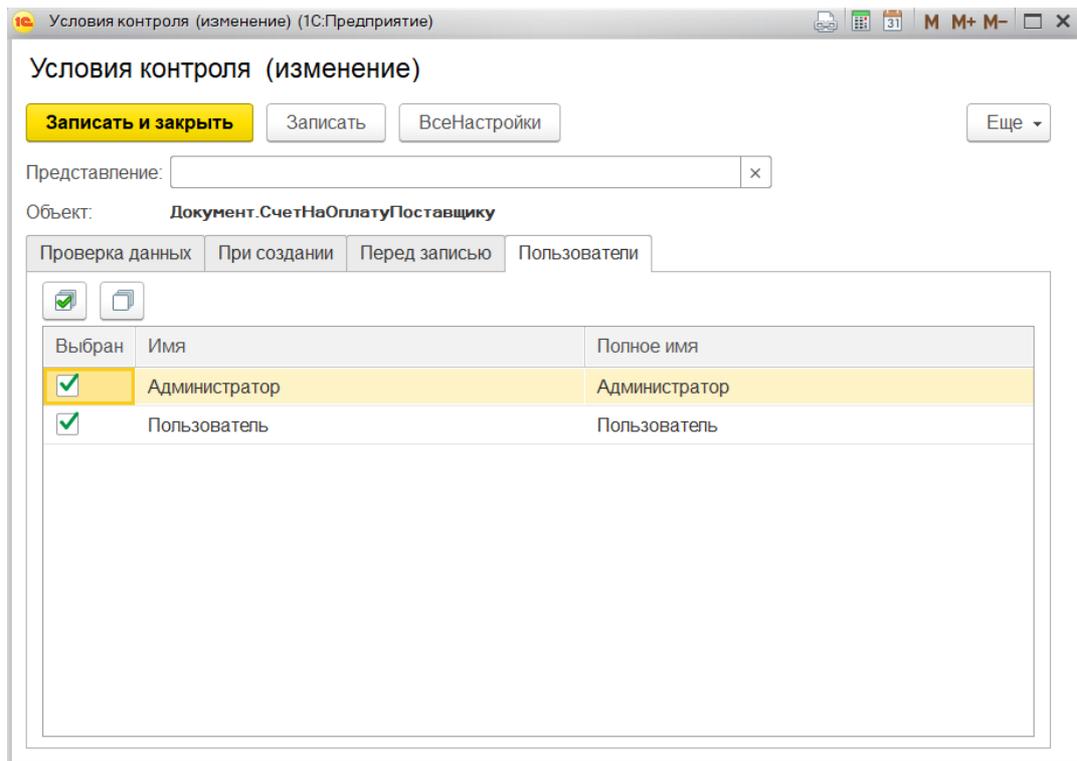


Рис. 21. Вкладка «Пользователи»

При попытке пользователя совершить действия, которые ему запрещены, внизу выводится предупреждающее сообщение (рис.22).

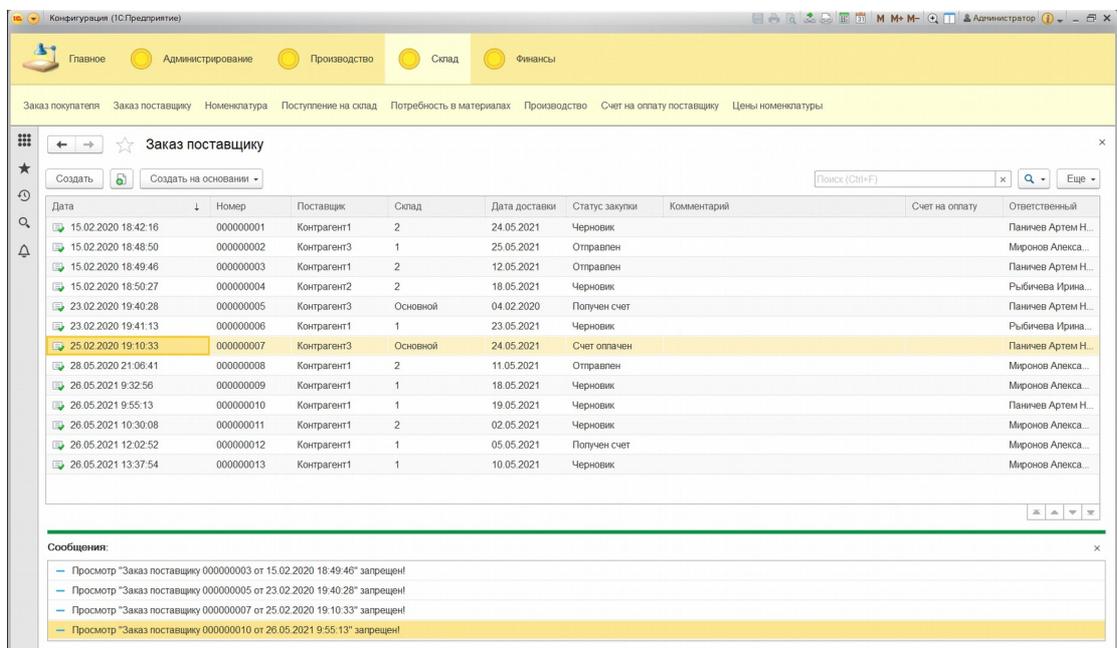


Рис. 22. Сообщение пользователю

1.7 Выбор стратегии тестирования, разработка тестов, программа и методика испытаний

Можно выделить один из наиболее часто используемых методов оценки качества информационной системы, который входит в набор эффективных средств современной системы обеспечения качества программного продукта – это его тестирование.

Тестирование — это процесс оценки системы или ее компонентов с целью выяснить, удовлетворяет ли она указанным требованиям или нет. Другими словами, тестирование — это выполнение системы с целью выявления пробелов, ошибок или отсутствующих требований, противоречащих фактическим требованиям. Под ошибками здесь понимается участок программного кода, выполнение которого при определенных условиях приводит к неожиданному поведению системы (т.е. поведению, не соответствующему требованиям) [3].

Своевременное начало тестирования снижает затраты и время на доработку и создание безошибочного программного обеспечения, которое доставляется клиенту. Однако в жизненном цикле разработки программного обеспечения тестирование можно начинать с этапа сбора требований и продолжать до развертывания программного обеспечения [6].

Это также зависит от используемой модели развития. Например, в модели «Водопад» формальное тестирование проводится на этапе тестирования; но в инкрементальной модели тестирование выполняется в конце каждого приращения / итерации, и все приложение тестируется в конце [3].

1.7.1 Объект и цель испытаний

Объектом испытаний является «Информационная система для контроля и анализа первичных документов» на технологической платформе 1С8.

Целью проводимых испытаний является:

- проверка правильности работы элементов подсистемы;
- проверка соответствия разработанной подсистемы требованиям, обозначенным в техническом задании;
- выявление ошибок, допущенных при проектировании модулей.

1.7.2 Требования к информационному, аппаратно-программному обеспечению и документации

1.7.2.1 Требования к функциональным характеристикам представлены ниже:

Разрабатываемая информационная система должна:

- встраиваться в любую информационную систему на базе платформы 1С (обновить формы в расширении - следует обновить формы расширений из основной конфигурации, чтобы и в расширении и в конфигурации формы были одинаковыми; установка свойств совместимости такими же, как и у основной конфигурации; возможность отключить расширение "на время" и продолжить работу в базовой конфигурации);
- иметь гибкую систему настроек под разные ИБ (при встраивании ИС (расширения) в информационную базу (ИБ) предприятия, предоставлять выбор по каким именно документам будет в дальнейшем происходить контроль и анализ);
- иметь predetermined набор настроек, для дальнейшего анализа с первичной информацией (должно быть предусмотрено хранилище с какими-то базовыми настройками, например: отслеживание на соответствие контрагента и документа в котором он фигурирует);
- своевременно, быстро и четко выявлять расхождения первичной информации от установленных настроек (ИС (расширение) должна иметь обработчик, который оперативно будет анализировать и отслеживать недочеты в указанных документах);

- анализировать такие реквизиты как: табличные части, ссылки на справочники, организации, контрагенты, договоры, номенклатуры и прочее;
 - иметь доступ к источникам ИБ предприятия, для получения информации;
 - подавать сигнал о недочетах (при нахождении различий первичной информации от установленных настроек, подсистема должна подавать сигнал о недочетах в виде сформированного сообщения, в котором будет указано: в каком документе обнаружен недочет, какая информация должна быть заполнена или изменена);
 - иметь интуитивно понятный интерфейс (не должно быть загромождений интерфейса значками, переходами и прочими управляющими элементами. Необходимо создать такие элементы управления, которые сделают работу пользователя простой и удобной, а со временем доведут ее до автоматизма. Интерфейс должен быть создан так, чтобы любую информацию или необходимые функции человек мог находить сразу, что сделает его работу с проектом комфортной.);
 - иметь возможность создавать любое количество настроек контроля изменений реквизитов документов (для одного документа может быть установлено не одно условие контроля, а два, три и т. д. при необходимости);
- иметь настройку пользователей, для которых будет выполняться контроль (при создании условий контроля, должен быть выбор, для каких именно пользователей будет действовать данный контроль).

1.7.2.2 Требования к надежности

Для надежной работы ИС, все функции должны быть проверены, и результаты проверки оформлены протоколом тестирования.

Информационная система должна быть доступна и работоспособна все время.

На каждый неверный ввод данных пользователем система должна реагировать соответствующим сообщением об ошибке.

Система должна обеспечить сохранность информации при наступлении следующих событий:

- отключение питания на рабочем месте;
- отказ аппаратуры сервера (например, накопители на жестких дисках).

1.7.2.3 Условия эксплуатации

Для того чтобы программа работала, пользователю необходимо иметь

- платформу 1С;
- конфигурацию 1С.

Пользовательское приложение должно обладать хорошо проработанным, интуитивно-понятным интерфейсом, не зависящим от разрешения монитора персонального компьютера конечного пользователя.

Программа будет стабильно работать от плюс 5 до плюс 35 °С при относительной влажности 90% и атмосферном давлении 462 мм.рт.ст., поскольку такие условия приблизительно соответствуют условиям эксплуатации современных компьютеров непромышленного исполнения.

Пользователь должен обладать практическими навыками работы с графическим пользовательским интерфейсом операционной системы и платформы 1С8.

1.7.2.4 Требования к составу и параметрам технических средств

Программа «1С:Предприятие» предъявляет следующие системные требования:

64-разрядный сервер 1С: Предприятия:

1. Процессор с архитектурой x86-64;
2. Оперативная память 2048 МБ и выше;
3. Жесткий диск 40 Гб и выше;
4. Устройство чтения компакт-дисков;
5. USB-порт.

1.7.2.5 Требования к информационной и программной совместимости

Программа реализуется на базе технологической платформы 1С8, версии не ниже 8.3.13.

Данный программный продукт может использоваться в операционных системах Windows 7, 8, 10 при наличии установленного программного обеспечения.

1.7.2.6 Требования к программной документации

Программная документация должна содержать следующие документы:

- расчетно-пояснительная записка;
- текст программы;
- спецификация;
- руководство пользователя.

Требования к оформлению представлены в таблице 18.

Таблица 18

Требования к оформлению

Элемент форматирования	Требование к форматированию
1	2
Документ	Печать на отдельных листах формата А4 (210x297 мм); оборотная сторона не заполняется; листы нумеруются. Печать возможна ч/б, или, если требуется, цветная. Файлы предъявляются на компакт-диске: РПЗ с приложениями. Листы и диск в конверте вложены в папку для сшивания дипломных работ [1].
Страницы	Ориентация – книжная; отдельные страницы, при необходимости, альбомные. Поля: верхнее и нижнее – по 2 см, левое – 3 см, правое – 1 см [1].
Абзацы	Межстрочный интервал в абзацах – 1,5, наименование частей, пунктов – двойной [1].
Шрифты	Кегль – 14. В таблицах шрифт 14. Текст программ – 8 [1].

1	2
Рисунки	Подписывается под ним по центру: Рис. X. Название В приложениях: Рис.П1.3. Название [1].
Таблицы	Подписывается: над таблицей, выравнивание по правому: «Таблица X». В следующей строке по центру Название Надписи в шапке (имена столбцов, полей) – по центру [1].
Общие требования к тексту	Красная строкам – 0,75. Выравнивание по ширине, в т.ч. в таблицах. Нумерация страниц. Титульная страница – первая, не нумеруется [1].

1.7.3 Состав, порядок и методы испытаний

Тестирование проводилось на персональном компьютере со следующими характеристиками:

- 1) процессор AMD A4-7210 APU 1.80 ГГц;
- 2) оперативная память 4 Гб;
- 3) монитор с разрешением 1366x768;
- 4) ОС Windows 10.

Что соответствует минимальным системным требованиям, указанным в техническом задании.

Порядок проведения тестирования:

- тестирование модуля формы выбора документов для контроля «Основная форма»;
- тестирование модуля формы ввода/выбора реквизитов документа «Форма ввода»;
- тестирование модуля обработки введенной информации.

Методы и виды тестирования:

- ручной контроль;
- автономное тестирование модулей;
- системное тестирование.

Ручной контроль обычно используют на ранних этапах разработки. Исходными данными для таких проверок являются: техническое задание, спецификации, структурная и функциональная схемы программного продукта и т. д. [6].

Из методов ручного контроля при тестировании модулей используется метод «Проверка за столом». Это проверка исходного текста или сквозные просмотры, выполняемые одним человеком, который читает текст программы, проверяет его по списку и пропускает через программу тестовые данные [6].

Системное тестирование сконцентрировано на поведении тестового объекта как целостной системы или продукта. Системное тестирование способно выявить несовместимость с окружением, некорректные комбинации данных, заведомо непредусмотренные сценарии использования, неудобство для пользователя, отсутствие необходимого пользователям функционала и так далее [3].

Во время системного тестирования тестовое окружение должно быть как можно ближе к предполагаемому эксплуатационному окружению системы для минимизации риска пропуска отказов, связанных с эксплуатационным окружением системы [3].

1.7.4 Результаты проведения испытаний

Результаты проведения испытаний представлены в табл. 19-21.

Таблица 19

Результаты автономного тестирования модулей

Дата	Тестирование проводил	Описание теста	Ожидаемый результат	Результат тестирования
1	2	3	4	5
Модуль формы выбора документов для контроля «Основная форма»				
15.04.21	Разработчик	Вывод списка всех имеющихся документов и справочников системы	Все документы и справочники отображаются в системе контроля	Неудача, справочники не отображаются

1	2	3	4	5
16.04.21	Разработчик	Вывод списка всех имеющихся документов и справочников системы	Все документы и справочники отображаются в системе контроля	Успех
18.04.21	Тестировщик	Создание нового условия контроля	В списке настроек отобразится новое условие контроля	Успех
19.04.21	Тестировщик	Получение нового идентификатора настройки	Автоматическое присваивание имени новому условию контроля	Успех
19.04.21	Тестировщик	Пересохранение настройки	Уже имеющаяся настройка была изменена и сохранена снова	Успех
Модуль формы ввода/выбора реквизитов документа «Форма ввода»				
21.04.21	Разработчик	При создании на сервере	При написании код дополнительной настройки, условие работает корректно	Неудача, условие не работает
23.04.21	Разработчик	При создании на сервере	При написании код дополнительной настройки, условие работает корректно	Успех
25.04.21	Тестировщик	Отображение всех реквизитов в выпадающем списке	Все реквизиты выбранного документа отображаются	Успех

1	2	3	4	5
25.04.21	Тестировщик	Отображение всех значений выбранного реквизита в выпадающем списке	Все значения выбранного реквизита отображаются	Неудача, ничего не отобразилось
27.04.21	Тестировщик	Отображение всех значений выбранного реквизита в выпадающем списке	Все значения выбранного реквизита отображаются	Успех
27.04.21	Разработчик	Выбор пользователей, для которых будет действовать условие	Флажки у пользователей ставятся/убираются, работают исправно	Успех
Модуль обработки введенной информации				
29.04.21	Разработчик	Анализ введенной информации на основе условий контроля	При вводе информации в первичные документы, отличающейся от условий контроля, высветится сообщение	Неудача, сообщение не высвечивается
31.04.21	Разработчик	Анализ введенной информации на основе условий контроля	При вводе информации в первичные документы, отличающейся от условий контроля, высветится сообщение	Успех

1	2	3	4	5
01.05.21	Тестирующий	Обработка данных на основе кода, введенного администратором	При попытке ввода информации, отличной от условия, прописанного в коде, система выдаст сообщение	Успех
03.05.21	Тестирующий	Сохранение документа	После анализа информации на основе условий контроля, несоответствий не выявлено и документ сохранился	Успех

Таблица 20

Результаты системного тестирования

Дата	Тестирование проводил	Описание теста	Ожидаемый результат	Результат тестирования
1	2	3	4	5
05.05.21	Разработчик	Встраивание расширения в любую ИБ на базе платформы	Расширение установилось, и работает	Неудача, при попытке установления расширения в новую ИС, высвечиваются ошибки
10.05.21	Разработчик	Встраивание расширения в любую ИБ на базе платформы	Расширение установилось, и работает	Успех

1	2	3	4	5
11.05.21	Тестировщик	Проверка адаптации под разные ИБ	При открытии списка условий контроля, должны отображаться все документы и справочники ИБ	Неудача, справочники не отображаются
12.05.21	Тестировщик	Проверка адаптации под разные ИБ	При открытии списка условий контроля, должны отображаться все документы и справочники ИБ	Успех
14.05.21	Разработчик	Проверка быстроты отклика ИС на расхождения информации	При введении информации отличной от условий контроля, система сразу выдает сообщение об ошибке	Успех
16.05.21	Тестировщик	Анализ табличных частей документов	При нажатии на поле ввода условий, появляется список соответствующих реквизитов	Успех
18.05.21	Разработчик	Настройка пользователей	При переходе на вкладку «Пользователи» все пользователи ИБ отображаются	Успех
17.05.21	Тестировщик	Удобство интерфейса	Все окна открываются, кнопки работают, логически все просто устроено	Неудача, неудобное расположение кнопок и полей заполнения

1	2	3	4	5
19.05.2 1	Тестировщик	Удобство интерфейса	Все окна открываются, кнопки работают, логически все просто устроено	Успех
05.06.2 1	Тестировщик	Проверка работы системы	Все функции системы работают	Успех

Таблица 21

Результаты тестирования требований к надежности

Дата	Тестирование проводил	Описание теста	Ожидаемый результат	Результат тестирования
20.05.2 1	Разработчик	Требования к надежности	Проверка сохранности информации в ИС при отключении питания или отказ аппаратуры	Успех
22.05.2 1	Тестировщик	Условия эксплуатации	При наличии работоспособной платформы 1С и конфигурации 1С, разрабатываемая ИС работает корректно	Успех
22.05.2 1	Тестировщик	Требования к составу и параметрам технических средств	Работа ИС проверена в условиях указанных в ТЗ	Успех
24.05.2 1	Разработчик	Требования к информационной и программной совместимости	Работа ИС проверена в условиях указанных в ТЗ	Успех

Результаты ручного тестирования (документации)

Дата	Тестирование проводил	Описание теста	Ожидаемый результат	Результат тестирования
10.02.21	Разработчик	Проверка ТЗ (по содержанию, оформлению)	Все оформлено по ГОСТ 19.201-78, функции прописаны подробно	Успех
25.05.21	Разработчик	Проверка РПЗ	Все оформлено по ГОСТам	Неудача, оформление текста не везде соответствует стандарту
28.05.21	Разработчик	Проверка РПЗ	Все оформлено по ГОСТам	Успех
03.06.21	Разработчик	Проверка приложений	Текст программы оформлен по ГОСТ 19.401-78 Спецификации оформлены по ГОСТ 19.202-78 Руководство пользователя по ГОСТ 19.503-79 ГОСТ 19.504-79 ГОСТ 19.505-79	Успех

В результате тестирования информационной системы были обнаружены ошибки, которые были исправлены в ходе дальнейшей разработки и отладки. В завершении тестирования можно сделать вывод о том, что разработанная ИС полностью соответствуют требованиям технического задания (прил.1).

Часть 2. Технико-экономическое обоснование

2.1 Создание иерархической структуры работы, выполнение назначения ресурсов

Для проектирования ВКР были составлены следующие основные задачи (рис.23), а также расписаны даты начала и окончания работы над ними.

	Название задачи	Длительность	Начало	Окончание
1	Выбор предметной области	1 день	Ср 10.02.21	Ср 10.02.21
2	Разработка технического задания	5 дней	Чт 11.02.21	Ср 17.02.21
3	Написание основной части	1 день	Чт 18.02.21	Чт 18.02.21
4	1. Сравнительный анализ отечественных и зарубежных аналогов проектируемой системы	1 день	Пт 19.02.21	Пт 19.02.21
5	2. Выбор технологии, среды и языка программирования	1 день	Пн 22.02.21	Пн 22.02.21
6	3. Анализ процесса обработки информации, выбор структур данных для ее хранения, выбор методов и алгоритмов решения задачи	1 день	Вт 23.02.21	Вт 23.02.21
7	4. Разработка спецификаций проектируемой системы	10 дней	Ср 24.02.21	Вт 09.03.21
12	5. Проектирование системы	7 дней?	Ср 10.03.21	Чт 18.03.21
17	6. Разработка ИС	45 дней	Пт 19.03.21	Чт 20.05.21
21	7. Проектирование интерфейса пользователя	4 дней	Пт 21.05.21	Ср 26.05.21
24	8. Выбор стратегии тестирования, разработка средств, программа и методика испытаний	18 дней	Чт 27.05.21	Пн 21.06.21

Рис. 23. Задачи и даты их проведения

2.2 Выравнивание загрузки ресурсов

Для более подробного плана работы задачи были разбиты на подзадачи, а также присвоены ресурсы необходимые для решения задач (рис. 24).

	Название задачи	Длительность	Начало	Окончание	Приоритет	Названия ресурсов
1	Выбор предметной области	1 день	Ср 10.02.21	Ср 10.02.21		Человеческий труд
2	Разработка технического задания	5 дней	Чт 11.02.21	Ср 17.02.21	1	Человеческий труд; Электроэнергия
3	Написание основной части	1 день	Чт 18.02.21	Чт 18.02.21	2	Человеческий труд; Электроэнергия
4	1. Сравнительный анализ отечественных и зарубежных аналогов проектируемой системы	1 день	Пт 19.02.21	Пт 19.02.21	3	Человеческий труд; Электроэнергия
5	2. Выбор технологии, среды и языка программирования	1 день	Пн 22.02.21	Пн 22.02.21	4	Человеческий труд
6	3. Анализ процесса обработки информации, выбор структур данных для ее хранения, выбор методов и алгоритмов решения задачи	1 день	Вт 23.02.21	Вт 23.02.21	5	Человеческий труд; Электроэнергия
7	4. Разработка спецификаций проектируемой системы	10 дней	Ср 24.02.21	Вт 09.03.21		
8	1) Построение функциональных диаграмм	2 дней	Ср 24.02.21	Чт 25.02.21	6	Человеческий труд; Электроэнергия
9	2) Построение диаграмм потоков данных	3 дней	Пт 26.02.21	Вт 02.03.21	8	Человеческий труд; Электроэнергия
10	3) Проектирование структур данных и построение	3 дней	Ср 03.03.21	Пт 05.03.21	9	Человеческий труд; Электроэнергия
11	4) Построение диаграмм переходов состояний	2 дней	Пн 08.03.21	Вт 09.03.21	10	Человеческий труд; Электроэнергия
12	5. Проектирование системы	7 дней	Ср 10.03.21	Чт 18.03.21		
13						
14	1) Проектирование структуры системы	3 дней	Ср 10.03.21	Пт 12.03.21	11	Человеческий труд; Электроэнергия
15	2) Разработка функциональной схемы	2 дней	Пн 15.03.21	Вт 16.03.21	14	Человеческий труд; Электроэнергия
16	3) Разработка модульной структуры	2 дней	Ср 17.03.21	Чт 18.03.21	15	Человеческий труд; Электроэнергия
17	6. Разработка ИС	45 дней	Пт 19.03.21	Чт 20.05.21		
18	1) Создание базы данных	10 дней	Пт 19.03.21	Чт 01.04.21	16	Человеческий труд; Электроэнергия
19	2) Разработка основ расширения	20 дней	Пт 02.04.21	Чт 29.04.21	18	Человеческий труд; Электроэнергия
20	3) Разработка функций	15 дней	Пт 30.04.21	Чт 20.05.21	19	Человеческий труд; Электроэнергия
21	7. Проектирование интерфейса пользователя	4 дней	Пт 21.05.21	Ср 26.05.21		
22	1) Построение графа диалога	1 день	Пт 21.05.21	Пт 21.05.21	20	Человеческий труд; Электроэнергия
23	2) Разработка форм ввода-вывода информации	3 дней	Пн 24.05.21	Ср 26.05.21	22	Человеческий труд; Электроэнергия
24	8. Выбор стратегии тестирования, разработка средств, программа и методика испытаний	18 дней	Чт 27.05.21	Пн 21.06.21		
25	1) Определение объекта и цели испытаний	3 дней	Чт 27.05.21	Пн 31.05.21	23	Человеческий труд; Электроэнергия
26	2) Формулирование требований к информации	3 дней	Вт 01.06.21	Чт 03.06.21	25	Человеческий труд; Электроэнергия
27	3) Определение состава, порядка и методов исп	3 дней	Пт 04.06.21	Вт 08.06.21	26	Человеческий труд; Электроэнергия
28	4) Формирование наборов тестовых данных	3 дней	Ср 09.06.21	Пт 11.06.21	27	Человеческий труд; Электроэнергия
29	5) Анализ результатов проведения испытаний	3 дней	Пн 14.06.21	Ср 16.06.21	28	Человеческий труд; Электроэнергия
30	Разработка руководства пользователя	3 дней	Чт 17.06.21	Пн 21.06.21	29	Человеческий труд; Электроэнергия

Рис. 24. Подзадачи и ресурсы

2.3 Диаграмма Ганта

Диаграмма Ганта — это инструмент, позволяющий визуализировать и управлять проектами, структурировать их выполнение и видеть общую картину задач, как личных, так и организации.

Диаграмма представлена на рис. 25.

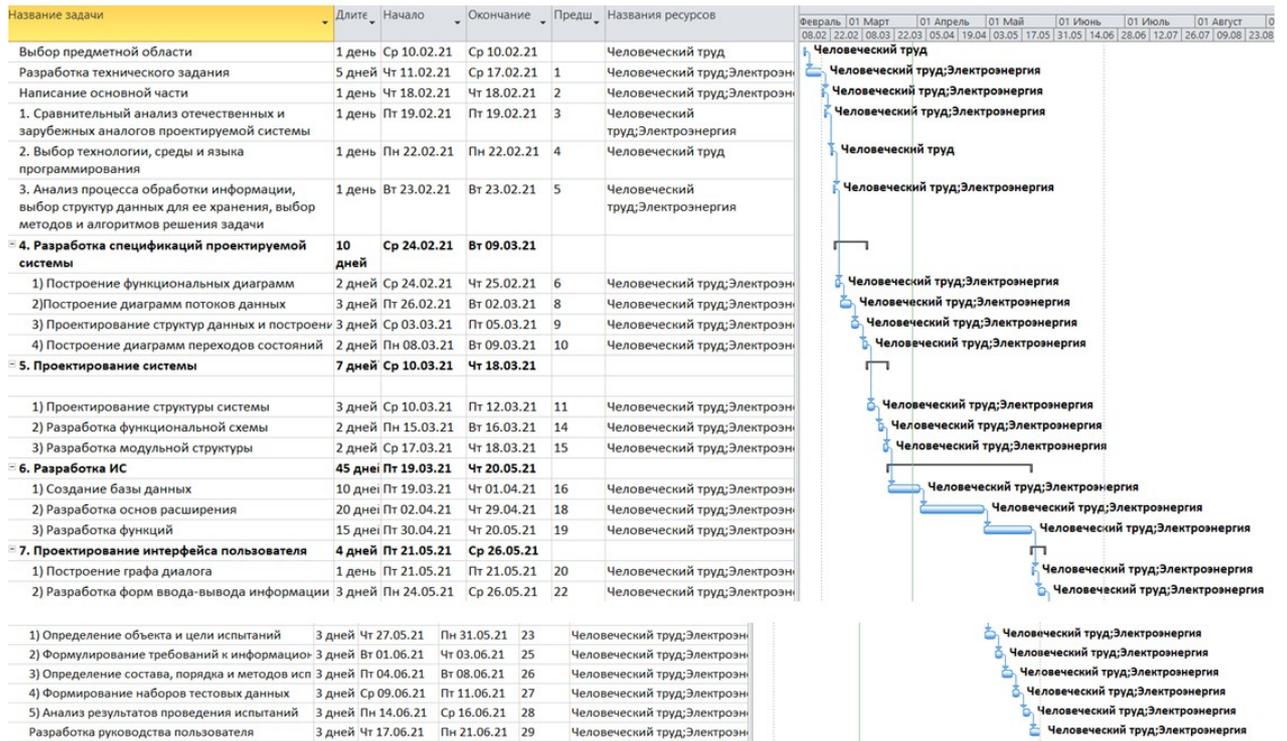


Рис. 25. Диаграмма Ганта

2.4 Критический путь

Критический путь – это одна или несколько задач, влияющие на срок завершения проекта. Это кратчайший путь исполнения проекта, т.е. невозможно выполнить проект ранее, чем будут завершены работы по цепочке задач проекта, лежащих на критическом пути.

Критический путь представлен на рис. 26.

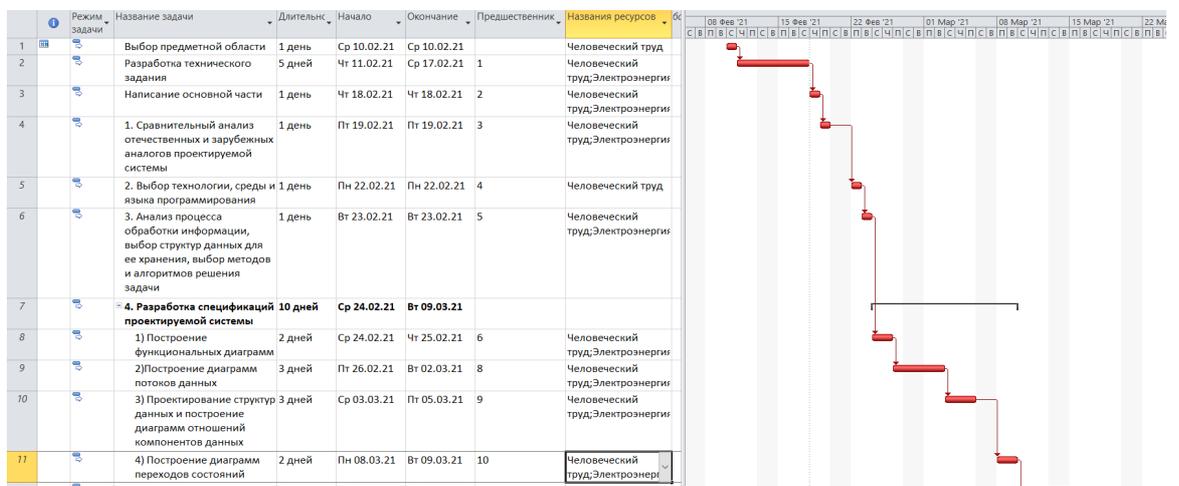


Рис. 26. Критический путь

2.5 Расчет себестоимости программного продукта

Состав разработчиков: программист-дипломник, руководитель ВКР.

Затраты на оплату труда при разработке программного продукта вычисляются по формуле:

$$Z_{тр} = (Z_{общ} + Отч) * T_n \quad - \text{с окладом} \quad (1)$$

где $Z_{общ}$ – общая зарплата работника за час;

$Отч$ – отчисления с зарплаты, %;

T_n – время написания программы.

Заработная плата программиста за час определяется по следующей формуле:

$$Z_{пр} = \frac{Cm_{пр}}{\Phi_{вм}} \quad (2)$$

где $Cm_{пр}$ – ставка программиста;

$\Phi_{вм}$ – фонд рабочего времени в месяц, ч.

Заработная плата дополнительная определяется по следующей формуле:

$$Z_{дон} = \frac{Z_{пр} * H_{дон}}{100} \quad (3)$$

где $Z_{пр}$ – заработная плата программиста;

$H_{дон}$ – норма отчислений на дополнительную зарплату (10%).

Зарплата общая вычисляется по следующей формуле:

$$Z_{общ} = Z_{пр} + Z_{дон} \quad (4)$$

Отчисления на соцстрах, фонд занятости и пенсионный фонд вычисляются по следующей формуле:

$$Отч = O_{cc} + O_{фз} + O_{пф} \quad (5)$$

где O_{cc} – отчисления на соцстрах (0,5% от $З_{общ}$);

$O_{фз}$ – отчисления в фонд занятости (0,5% от $З_{общ}$);

$O_{пф}$ – отчисления в пенсионный фонд (2% от $З_{общ}$).

Ставка начинающего 1С-программиста равна 24 000 р. Ставка руководителя ВКР равна 31700. По формуле (2) рассчитываются общие заработные платы за час программиста-дипломника и руководителя ВКР соответственно:

$$З_{пр} = \frac{24\,000}{160} = 150 \text{ руб.}$$

$$З_{пр} = \frac{31\,700}{160} = 198,13 \text{ руб.}$$

По формуле (3) рассчитываются дополнительные заработные платы:

$$З_{дон} = \frac{150 \cdot 10}{100} = 15 \text{ руб.}$$

$$З_{дон} = \frac{198,13 \cdot 10}{100} = 19,81 \text{ руб.}$$

Зарплаты общие вычисляются по следующей формуле (4):

$$З_{общ} = 150 + 15 = 165 \text{ руб.}$$

$$З_{общ} = 198 + 19,81 = 217,81 \text{ руб.}$$

Отчисления на соцстрах, фонд занятости и пенсионный фонд вычисляются по формуле (5):

$$Отч = 150 \cdot 0,005 + 150 \cdot 0,005 + 150 \cdot 0,02 = 4,50 \text{ руб.}$$

$$Отч = 217,81 * 0,005 + 217,81 * 0,005 + 217,81 * 0,02 = 6,53 \text{ руб.}$$

По формуле (1) рассчитываются затраты на оплату труда при разработке программного продукта:

$$З_{тр} = (165 + 4,50) * 160 = 27\,120 \text{ руб.}$$

$$З_{тр} = (217,81 + 6,53) * 160 = 35\,894,40 \text{ руб.}$$

Данные по заработной плате сведены в табл.22.

Таблица 22

Данные по заработной плате

Должность разработчика	Время работы, мес.	Ст _{пр} , руб.	З _{пр} , руб.	З _{доп} , руб.	З _{общ} , руб.	Отч, руб.	З _{тр} , руб.
Программист-дипломник	4	24 000	150	15	165	4,50	108 480
Руководитель ВКР	4	31 700	198,13	19,81	217,81	6,53	143 576
Итого:		55 700	348,13	34,81	382,81	11,03	252 056

Затраты на использование машинного времени вычисляются по формуле:

$$З_{м.вр} = C_{м.вр} \cdot Вр_{в.т} \quad (6)$$

где $З_{м.вр}$ – затраты на использование машинного времени, руб.;

$C_{м.вр}$ – стоимость одного часа машинного времени, руб./ч;

$Вр_{в.т}$ – время использования вычислительной техники, ч.

Стоимость одного часа машинного времени рассчитывается по формуле:

$$C_{м.вр} = \frac{C_{\kappa}}{C_{сп.к} * K_{р.д} * Bp_c} + C_{тэ} * M_{вс} \quad (7)$$

где C_{κ} – покупная цена компьютера;

$C_{сп.к}$ – срок службы компьютера, год;

$K_{р.д}$ – количество рабочих дней в году;

Bp_c – время работы компьютера в течение суток, ч;

$C_{тэ}$ – стоимость одного кВт*ч электроэнергии, руб;

$M_{вс}$ – мощность вычислительной системы, кВт.

Время использования вычислительной техники рассчитывается по следующей формуле:

$$Bp_{в.т} = K_{д.р} * Bp_c \quad (8)$$

где $Bp_{в.т}$ – время использования вычислительной техники, ч;

$K_{д.р}$ – количество дней разработки программного обеспечения.

Затраты на носители информации принимаются в размере 2 % от цены вычислительной техники и рассчитываются по формуле:

$$З_{н.и} = C_{\kappa} * 2 \quad (9)$$

где C_{κ} – покупная цена компьютера.

Затраты на текущий и профилактический ремонт составляют 4% от цены вычислительной техники и вычисляются по следующей формуле:

$$Z_{рем} = C_k * 4 \quad (10)$$

где C_k – покупная цена компьютера.

Прочие эксплуатационные расходы включают в себя затраты на освещение, отопление, охрану, уборку и текущий ремонт помещений. Они принимаются в размере 10 % от стоимости помещения (или его аренды), где происходит разработка программного продукта и вычисляются по следующей формуле:

$$Z_{пр} = C_{пм} * 10 \quad (11)$$

где $C_{пм}$ – стоимость помещения, руб.

Себестоимость разработки определяется по следующей формуле:

$$C_{п.п} = Z_{тр} + Z_{м.вр} + Z_{н.и} + Z_{рем} + Z_{пр} \quad (12)$$

По формуле (7) вычисляется стоимость одного часа машинного времени:

$$C_{м.вр} = \frac{42000}{10 * 247 * 8} + 4,87 * 0,3 = 3,57 \text{ руб./ч.}$$

Время использования вычислительной техники рассчитывается по формуле (8):

$$Вр_{е.м} = 80 * 5 = 400 \text{ ч.}$$

По формуле (6) вычисляются затраты на использование машинного времени:

$$Z_{м.вр} = 3,57 * 400 = 1428 \text{ руб.}$$

Затраты на носители информации рассчитываются по формуле (9):

$$Z_{н.и} = 42\,000 * 0,02 = 840 \text{ руб.}$$

По формуле (10) определяются затраты на текущий и профилактический ремонт:

$$Z_{рем} = 42\,000 * 0,04 = 1\,680 \text{ руб.}$$

Прочие эксплуатационные расходы вычисляются по формуле (11):

$$Z_{пр} = 12\,900 * 0,1 = 1\,290 \text{ руб.}$$

Себестоимость разработки рассчитывается по формуле (12):

$$C_{н.н} = 252\,056 + 1\,428 + 840 + 1\,680 + 1\,290 = 257\,294 \text{ руб.}$$

2.6 Расчет цены программного продукта

Для определения минимальной цены, ниже которой разработчику будет невыгодно продавать программный продукт, используется следующая формула:

$$Ц_{н.н} = C_{н.н} * (1 + H_{пр}) \quad (13)$$

где $H_{пр}$ – норматив прибыли (20%).

$$Ц_{н.н} = 257\,294 * (1 + 0,2) = 308\,752,80 \text{ руб.}$$

2.7 Расчет экономической эффективности

Расходы потребителя, связанные с эксплуатацией программы, определяются по следующей формуле:

$$P_{э.н} = Bp_{н.н} * C_{м.вр} + \frac{Ц_{н.н}}{C_{сл}} \quad (14)$$

где $Vp_{n.n}$ - объем машинного времени в течение года, необходимый для решения данной задачи с использованием программы, ч;

$C_{м.вр}$ – стоимость одного часа машинного времени, руб./ч;

$C_{сл}$ - срок службы программного продукта, год.

Объем машинного времени в течение года вычисляется по следующей формуле:

$$Vp_{n.n} = K_{д.и} * Vp_c \quad (15)$$

где $K_{д.и}$ – количество дней использования ПО в году ;

Vp_c - количество

Стоимость одного часа машинного времени рассчитывается по формуле (7):

$$C_{м.вр} = \frac{25000}{5 * 247 * 8} + 2,87 * 0,2 = 3,10 \text{ руб./ч.}$$

По формуле (15) объем машинного времени в течение года определяется:

$$Vp_{n.n} = 247 * 8 = 1976 \text{ ч.}$$

Расходы потребителя, связанные с эксплуатацией программы рассчитываются по формуле (14):

$$P_{э.н} = 1976 * 3,10 + \frac{308752,80}{2} = 160502 \text{ руб.}$$

На момент внедрения программного продукта у потребителя все работы выполнялись вручную, поэтому капитальные затраты рассчитываются по формуле:

$$P_{кап} = \frac{Vp_{n.n} * K_{э.м}}{\Phi_{вр}} + Ц_{n.n} \quad (16)$$

где $K_{э.м}$ - капитальные затраты на вычислительную технику, для которой предназначена программа, руб;

$\Phi_{вр}$ - полезный годовой фонд времени работы вычислительной техники, принимается условно 2000 ч. в год;

Капитальные затраты на вычислительную технику рассчитываются по формуле:

$$K_{эвм} = C_{эвм} + P_{п.п} \quad (17)$$

где $C_{эвм}$ - цена вычислительной техники, руб;

$P_{п.п}$ - прочие расходы потребителя, связанные с помещением (отопление, освещение, уборка и т.д.), принимаются в размере 10 % от стоимости помещения потребителя (или его аренды), руб.

$$K_{эвм} = 25000 + 2000 = 27000 \text{ руб.}$$

По формуле (16) рассчитываются капитальные затраты:

$$P_{кап} = \frac{1976 * 27000}{2000} + 308752,80 = 335428,80 \text{ руб.}$$

Для расчета годовой экономии эксплуатационных расходов потребителя вычисляются эксплуатационные затраты потребителя при решении задачи вручную:

$$P_{э.руч} = 1,21 * \PhiЗП * 12 \quad (18)$$

где $\PhiЗП$ - фонд заработной платы персонала, обслуживающего решение задачи вручную, руб.

$$P_{э.руч} = 1,21 * 50000 * 12 = 726000 \text{ руб.}$$

Годовая экономия эксплуатационных расходов у одного потребителя рассчитывается по формуле:

$$\mathcal{E} = P_{\mathcal{E}, \text{рyч}} - P_{\mathcal{E}, \text{н}} \quad (19)$$

$$\mathcal{E} = 726\,000 - 160\,502 = 565\,498 \text{ руб.}$$

Срок окупаемости программного продукта рассчитывается по формуле:

$$T_{\text{ок}} = \frac{P_{\text{кан}}}{\mathcal{E}} \quad (20)$$

$$T_{\text{ок}} = \frac{335\,428,80}{565\,498} = 0,59 \text{ года.}$$

Годовой экономический эффект, получаемый одним потребителем, рассчитывается по формуле:

$$\mathcal{E}\mathcal{E} = \mathcal{E} - E_{\text{н}} * P_{\text{кан}} \quad (21)$$

где $E_{\text{н}}$ - нормативный коэффициент эффективности дополнительных капитальных вложений, равный 0,15.

$$\mathcal{E}\mathcal{E} = 565\,498 - 0,15 * 335\,428,80 = 515\,183,68 \text{ руб.}$$

Заключение

Результатами выполнения выпускной квалификационной работы являются проектирование и разработка информационной системы для контроля и анализа первичных документов, которая представляет собой расширение конфигурации 1С8. Данная работа выполнялась в рамках заказа компании ОАО «Северсталь-инфоком» на создание информационной системы контроля пользовательских ошибок. Программно-информационная система позволит минимизировать затраты рабочего времени на предприятии и сократить сроки финансового закрытия, путем обнаружения ошибок в момент занесения.

В результате проведения испытаний всех модулей было установлено соответствие разработанной подсистемы требованиям, указанных в техническом задании. Были выявлены незначительные отклонения в работе системы, которые в дальнейшем были исправлены.

Были освоены следующие компетенции:

- способность проводить предпроектное обследование объекта проектирования, системный анализ предметной области, их взаимосвязей;
- способность проводить техническое проектирование;
- способность проводить рабочее проектирование;
- способность проводить выбор исходных данных для проектирования;
- способность проводить моделирование процессов и систем;
- способность оценивать надежность и качество функционирования объекта проектирования;
- способность проводить расчет экономической эффективности;
- способность разрабатывать, согласовывать и выпускать все виды проектной документации;
- способность к проектированию базовых и прикладных информационных технологий;

- способность разрабатывать средства реализации информационных технологий (методические, информационные, математические, алгоритмические, технические и программные);
- готовность участвовать в постановке и проведении экспериментальных исследований;
- способность обосновывать правильность выбранной модели, сопоставляя результаты экспериментальных данных и полученных решений;
- способность оформлять полученные рабочие результаты в виде презентаций, научно-технических отчетов, статей и докладов на научно-технических конференциях.

-

- 1.** Ершов Е.В., Виноградова Л.Н. Методика и организация самостоятельной работы: учебно-методическое пособие. – Череповец: ФГБОУ ВПО «Череповецкий государственный университет», 2015 – 208 с.
- 2.** Абрамов Г.В. Проектирование информационных систем [Текст]: учеб. пособие/ Г.В. Абрамов, И.Е. Медведкова, Л.А. Коробова; Воронеж, гос. ун-т инж. технол. – Воронеж: ВГУИТ, 2012. – 172 с.
- 3.** Воробьев А.Ф. Тестирование как ключевой элемент системы обеспечения качества информационных систем. Тип: статья в журнале - научная статья Дальневосточный федеральный университет. – Владивосток, 2014 г. – 248с.
- 4.** Вендров А.М. CASE - технологии. Современные методы и средства проектирования информационных систем. — М.: Финансы и статистика, 1998. – 254с.
- 5.** Громов Ю.Ю., Иванова О.Г., Беляев М.П., Минин Ю.В. Технология программирования: учебное пособие / Ю.Ю. Громов, О.Г. Иванова, М.П. Беляев, Ю.В. Минин. – Тамбов: Изд-во ФГБОУ ВПО «ТГТУ», 2013 – 172 с. – 100 экз.
- 6.** Дворянкин, А. М. Основные методы тестирования программного обеспечения: учеб. пособие / А. М. Дворянкин, А. А. Ерофеев, А. В. Аникин; ВолгГТУ. – Волгоград, 2015. – 120 с.
- 7.** Иванова Г.С., Ничушкина Т.Н. Проектирование программного обеспечения: Учебное пособие. - Московский государственный технический университет, 2002. – 74 с.
- 8.** Иванова Г.С. Технология программирования: Учебник для вузов. - М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. - 320 с.: ил. (Сер. Информатика в техническом университете).

- 9.** Киселев Д.Ю., Киселев Ю.В., Макарьев В.Д. Структурный анализ потоков данных (Data Flow Diagrams – DFD): метод. указания / сост. Д.Ю. Киселев, Ю.В. Киселев, В.Д. Макарьев. – Самара: Изд-во СГАУ, 2014 – 12 с.
- 10.** Кумагина Е.А., Неймарк Е.А. Модели жизненного цикла и технологии проектирования программного обеспечения: Учебно-методическое пособие. - Нижегородский государственный университет, 2016. – 41с.
- 11.** Милованов И. В., Лоскутов В. И. Основы разработки программного обеспечения вычислительных систем: Учебное пособие - Тамбовский государственный технический университет, 2011. – 88с.
- 12.** Радченко М.Г., Хрусталева Е.Ю. 1С: Предприятие 8.3. Практическое пособие разработчика. Примеры и типовые приемы: практическое пособие - М.: ООО «1С-Публишинг», 2013. – 964 с.
- 13.** Сергеев С.Ф., Падерно П.И., Назаренко Н.А. Введение в проектирование интеллектуальных интерфейсов: Учебное пособие. – Санкт-Петербургский государственный университет, 2011. – 108 с.
- 14.** Суркова Н.Е. Методология структурного проектирования информационных систем: Монография / Н.Е. Суркова, А.В. Остроух. Красноярск: Научно-инновационный центр, 2014 190 с.
- 15.** Мещеряков С. В., Иванов В. М. Эффективные технологии создания информационных систем; Политехника - Москва, 2005. - 312 с.
- 16.** Назаров С.В. Архитектура и проектирование программных систем: Монография. / С.В. Назаров. - М.: ИНФРА-М, 2013. - 351 с.
- 17.** Хрусталева Е.Ю. Расширения конфигураций. Адаптация прикладных решений с сохранением поддержки в облаках и на земле. Разработка в системе «1С:Предприятие 8.3». Издание 2 - Библиотека специалиста по внедрению, 2015. - 356с.

18. Клерк. Обзор российских сервисов по обработке первичной документации [Электронный ресурс]. - Режим доступа: <https://www.klerk.ru/buh/articles/486972/> (дата обращения: 15.04.2021).

19. Студопедия. Сцепление модулей [Электронный ресурс]. - Режим доступа: https://studopedia.ru/12_93260_stseplenie-moduley.html (дата обращения: 25.04.2021).

20. Studwood.ru. Описание технического программного средства CA ERwin Data Modeler [Электронный ресурс]. - Режим доступа: https://studwood.ru/1698551/informatika/opisanie_tehnicheskogo_programmnogo_sredstva_erwin_data_modeler (дата обращения: 15.04.2021).

21. Первый бит. Описание и возможности 1С:Предприятие 8 [Электронный ресурс]. - Режим доступа: <https://cherepovec.1cbit.ru/blog/opisanie-i-vozmozhnosti-1s-predpriyatie-8/> (дата обращения: 19.03.2021).

22. Система программ 1С:Предприятие, официальный сайт. Встроенный язык [Электронный ресурс]. - Режим доступа: <https://v8.1c.ru/platforma/vstroennyy-yazyk/> (дата обращения: 19.03.2021).

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий
наименование института (факультета)

Математического и программного обеспечения ЭВМ
наименование кафедры

УТВЕРЖДАЮ
Зав. кафедрой МПО ЭВМ
д.т.н., профессор _____ Ершов Е.В.
« 10 » _____ февраля 2021 г.

Разработка информационной системы контроля и анализа первичных
документов на технологической платформе 1С8

Техническое задание на выпускную квалификационную работу

Листов 7

Руководитель: доцент Ганичева О. Г.
Исполнитель: студент гр. 1ИСб-

Соколова С. С.

2021 год

Ведение

В рамках выпускной квалификационной работы необходимо разработать информационную систему предприятия, которая представляет собой расширение по контролю качества ввода первичных документов в 1С:Бухгалтерии, с целью минимизировать затраты рабочего времени на предприятии «Северсталь-инфоком» и сократить сроки финансового закрытия, путем обнаружения пользовательских ошибок.

Для производственного учета, как на больших, так и на малых предприятиях часто используется программа компании «1С». Данная информационная система будет разрабатываться на базе платформы 1С8.

1 Основания для разработки

Основанием для разработки является заказ от ОАО «Северсталь-инфоком», выданный на кафедре математического и программного обеспечения ЭВМ Института информационных технологий Череповецкого государственного университета.

Задание: Разработка информационной системы контроля и анализа первичных документов на технологической платформе 1С8.

Дата выдачи: 10 февраля 2021 года.

2 Назначение разработки

Разработанная информационная система предназначена для оптимизации работы предприятия ОАО «Северсталь-инфоком», с целью сокращения времени, затрачиваемого на исправление пользовательских ошибок.

3 Требования к программе

3.1 Требования к функциональным характеристикам

Разрабатываемая информационная система должна:

- встраиваться в любую информационную систему на базе платформы 1С (обновить формы в расширении - следует обновить формы расширений из основной конфигурации, чтобы и в расширении и в конфигурации формы были одинаковыми; установка свойств совместимости такими же, как и у основной конфигурации; возможность отключить расширение "на время" и продолжить работу в базовой конфигурации);
- иметь гибкую систему настроек под разные ИБ (при встраивании ИС (расширения) в информационную базу (ИБ) предприятия, предоставлять выбор по каким именно документам будет в дальнейшем происходить контроль и анализ);
- иметь predetermined набор настроек, для дальнейшего анализа с первичной информацией (должно быть предусмотрено хранилище с какими-то базовыми настройками, например: отслеживание на соответствие контрагента и документа в котором он фигурирует);
- своевременно, быстро и четко выявлять расхождения первичной информации от установленных настроек (ИС (расширение) должна иметь обработчик, который оперативно будет анализировать и отслеживать недочеты в указанных документах);
- анализировать такие реквизиты как: табличные части, ссылки на справочники, организации, контрагенты, договоры, номенклатуры и прочее;
- иметь доступ к источникам ИБ предприятия, для получения информации;
- подавать сигнал о недочетах (при нахождении различий первичной информации от установленных настроек, подсистема должна подавать сигнал о недочетах в виде сформированного сообщения, в котором будет указано: в

каком документе обнаружен недочет, какая информация должна быть заполнена или изменена);

- иметь интуитивно понятный интерфейс (не должно быть загромождений интерфейса значками, переходами и прочими управляющими элементами. Необходимо создать такие элементы управления, которые сделают работу пользователя простой и удобной, а со временем доведут ее до автоматизма. Интерфейс должен быть создан так, чтобы любую информацию или необходимые функции человек мог находить сразу, что сделает его работу с проектом комфортной.);

- иметь возможность создавать любое количество настроек контроля изменений реквизитов документов (для одного документа может быть установлено не одно условие контроля, а два, три и т. д. при необходимости);

- иметь настройку пользователей, для которых будет выполняться контроль (при создании условий контроля, должен быть выбор, для каких именно пользователей будет действовать данный контроль).

3.2 Требования к надежности

Для надежной работы ИС, все функции должны быть проверены, и результаты проверки оформлены протоколом тестирования.

Информационная система должна быть доступна и работоспособна все время.

На каждый неверный ввод данных пользователем система должна реагировать соответствующим сообщением об ошибке.

Система должна обеспечить сохранность информации при наступлении следующих событий:

- отключение питания на рабочем месте;
- отказ аппаратуры сервера (например, накопители на жестких дисках).

3.3 Условия эксплуатации

Для того чтобы программа работала, пользователю необходимо иметь

- платформу 1С;
- конфигурацию 1С.

Пользовательское приложение должно обладать хорошо проработанным, интуитивно-понятным интерфейсом, не зависящим от разрешения монитора персонального компьютера конечного пользователя.

Программа будет стабильно работать от плюс 5 до плюс 35 °С при относительной влажности 90% и атмосферном давлении 462 мм.рт.ст., поскольку такие условия приблизительно соответствуют условиям эксплуатации современных компьютеров непромышленного исполнения.

Пользователь должен обладать практическими навыками работы с графическим пользовательским интерфейсом операционной системы и платформы 1С8.

3.4 Требования к составу и параметрам технических средств

Программа 1С: Предприятие предъявляет следующие системные требования:

64-разрядный сервер 1С: Предприятия:

- Процессор с архитектурой x86-64;
- Оперативная память 2048 МБ и выше;
- Жесткий диск 40 Гб и выше;
- Устройство чтения компакт-дисков;
- USB-порт.

3.5 Требования к информационной и программной совместимости

Программа реализуется на базе технологической платформы 1С8, версии не ниже 8.3.13.

Данный программный продукт может использоваться в операционных системах Windows 7, 8, 10 при наличии установленного программного обеспечения.

4 Требования к программной документации

Программная документация должна содержать следующие документы:

- расчетно-пояснительная записка;
- текст программы;
- спецификация;
- руководство пользователя.

Требования к оформлению представлены в таблице П1.1

Таблица П1.1

Требования к оформлению

Элемент форматирования	Требование к форматированию
Документ	Печать на отдельных листах формата А4 (210x297 мм); оборотная сторона не заполняется; листы нумеруются. Печать возможна ч/б, или, если требуется, цветная. Файлы предъявляются на компакт-диске: РПЗ с приложениями. Листы и диск в конверте вложены в папку для сшивания дипломных работ.
Страницы	Ориентация – книжная; отдельные страницы, при необходимости, альбомные. Поля: верхнее и нижнее – по 2 см, левое – 3 см, правое – 1 см.
Абзацы	Межстрочный интервал в абзацах – 1,5, наименование частей, пунктов – двойной.
Шрифты	Кегль – 14. В таблицах шрифт 14. Текст программ – 8.
Рисунки	Подписывается под ним по центру: Рис. X. Название В приложениях: Рис.П1.3. Название
Таблицы	Подписывается: над таблицей, выравнивание по правому: «Таблица X». В следующей строке по центру Название Надписи в шапке (имена столбцов, полей) – по центру.
Общие требования к тексту	Красная строка – 0,75. Выравнивание по ширине, в т.ч. в таблицах. Нумерация страниц. Титульная страница – первая, не нумеруется.

5 Стадии и этапы разработки

Стадии и этапы разработки программного обеспечения содержатся в табл. П1.2.

Таблица П1.2

Стадии и этапы разработки

Наименование этапа разработки	Сроки разработки	Результат выполнения	Отметка о выполнении
1	2	3	4
Разработка технического задания	10.02.2021	Готовое техническое задание	Выполнено
Проведение сравнительного анализа отечественных и зарубежных аналогов проектируемой ИС	16.02.2021	Выявление сильных и слабых сторон аналогов	Выполнено
Выбор технологии, среды и языка программирования	20.02.2021	Выбраны технологии разработки, разработки и языка программирования	Выполнено
Анализ процесса обработки информации, выбор структур данных для ее хранения, выбор методов и алгоритмов решения задачи	27.02.2021	Выбраны структуры данных и алгоритмы решения поставленных задач	Выполнено
Разработка спецификаций проектируемой ИС	15.03.2021	Разработаны спецификации программного обеспечения	Выполнено
Проектирование ИС	30.03.2021	Спроектированное программное обеспечение	Выполнено
Проектирование интерфейса пользователя	10.04.2021	Готовый интерфейс пользователя	Выполнено
Выбор стратегии тестирования, разработка тестов, программа и методика испытаний	15.04.2021	Выбрана стратегия тестирования разработанного программного обеспечения	Выполнено

1	2	3	4
Разработка системы	02.05.2021	Разработана готовая ИС	Выполнено
Тестирование системы	24.05.2021	Система прошла тестирование	Выполнено
Демонстрация системы	26.05..2021	Система продемонстрирована	Выполнено
Доработка модулей ИС	01.06.2021	Модули ИС доработаны	Выполнено
Подготовка к сдаче РПЗ на нормы контроля и дипломному руководителю	04.06.2021	РПЗ подготовлено к проверке	Выполнено
Подготовка к педзащита ВКР	08.06.2021	Предзащита пройдена	Выполнено
Подготовка к защите ВКР	20.06.2021	Все готово	Выполнено

6 Порядок контроля и приемки

Порядок контроля и приема проекта представлен в табл. П1.3.

Таблица П1.3

Порядок контроля и приемки

Наименование контрольного этапа выполнения	Сроки контроля	Результат выполнения	Отметка о приемке результата контрольного этапа
1	2	3	4
Предоставление технического задания	10.02.2021	Техническое задание утверждено	Выполнено
Получение требований к используемым технологиям и языкам программирования	17.02.2021	Согласованы технологии разработки, среды написания программных кодов и языка программирования	Выполнено

1	2	3	4
Анализ процесса обработки информации, выбор структур данных для ее хранения, выбор методов и алгоритмов решения задачи	28.02.2021	Согласованы структуры данных и алгоритмы решения поставленных задач	Выполнено
Предоставление результатов разработки спецификаций проектируемого программного обеспечения	16.03.2021	Предоставлены спецификации программного обеспечения	Выполнено
Демонстрация спроектированной ИС	31.03.2021	Предоставлен прототип программного обеспечения	Выполнено
Демонстрация интерфейса пользователя	11.04.2021	Предоставлен пользовательский интерфейс	Выполнено
Демонстрация разработанных тестов	25.05.2021	Предоставление стратегии тестирования и тестов	Выполнено
Демонстрация ИС	26.05.2021	Продемонстрирована работа ИС	Выполнено
Сдача РПЗ на нормы контроля и дипломному руководителю	05.06.2021	РПЗ сдано на проверки	Выполнено
Предзащита ВКР	09.06.2021	Предзащита пройдена	Выполнено
Защита ВКР	23.06.2021	ВКР защищена	

Текст программы

```

&НаСервере
Процедура ПриСозданииНаСервере(Отказ,
СтандартнаяОбработка)
ЗаполнитьДеревоОбъектов();
КонецПроцедуры

&НаСервере
Процедура ЗаполнитьДеревоОбъектов()
СписокКатегорий = Новый СписокЗначений;
СписокКатегорий.Добавить("Справочник", "Справочники", „Библио
иотекаКартинок.Справочник");
СписокКатегорий.Добавить("Документ", "Документы", „Библиоте
каКартинок.Документ");
ОчищаемоеДерево =
РеквизитФормыВЗначение("ДеревоОбъектов");
ОчищаемоеДерево.Строки.Очистить();
ЗначениеВРеквизитФормы(ОчищаемоеДерево,
"ДеревоОбъектов");
СтрокиГруппы =
ЭтотОбъект.ДеревоОбъектов.ПолучитьЭлементы();
ИмяОсновногоПользователя =
Обработки.ек_ПользовательскиеПроверкиИОбработчики.Полно
еИмяНеуказанногоПользователя();
Для каждого ТекущаяКатегория Из СписокКатегорий Цикл
КартинкаКатегории = ТекущаяКатегория.Картинка;
ИмяКатегории = ТекущаяКатегория.Значение;
ПредставлениеКатегории = ТекущаяКатегория.Представление;
НоваяГруппа = СтрокиГруппы.Добавить();
НоваяГруппа.ИдентификаторОбъекта = ИмяКатегории;
НоваяГруппа.Представление = ПредставлениеКатегории;
НоваяГруппа.Картинка = КартинкаКатегории;
СтрокиОбъекты = НоваяГруппа.ПолучитьЭлементы();
Для каждого ОбъектМетаданных Из
Метаданные[ПредставлениеКатегории] Цикл
Если
СтрНачинаетсяС(ВРег(ОбъектМетаданных.Имя), "УДАЛИТЬ")
Тогда
Продолжить;
КонецЕсли;
Попытка
ЕстьИзмененияРасширениямиКонфигурации =
ОбъектМетаданных.ЕстьИзмененияРасширениямиКонфигураци
и();
Исключение
//Если запускаем на старой платформе и нет возможности
проверить изменение метаданного, то считаем, что изменение
ЕстьИзмененияРасширениямиКонфигурации = Истина;
КонецПопытки;
Если ЕстьИзмененияРасширениямиКонфигурации Тогда
НовыйОбъект = СтрокиОбъекты.Добавить();
НовыйОбъект.ИдентификаторОбъекта =
ОбъектМетаданных.ПолноеИмя();
НовыйОбъект.Представление =
ОбъектМетаданных.Представление();
НовыйОбъект.Картинка = КартинкаКатегории;
ЗаполнитьСписокНастроек(НовыйОбъект,
НовыйОбъект.ИдентификаторОбъекта,
ИмяОсновногоПользователя);
КонецЕсли;
КонецЦикла;
КонецЦикла;
КонецПроцедуры // ЗаполнитьДеревоОбъектов()

&НаСервере
Процедура ЗаполнитьСписокНастроек(знач СтрокаДерева =
Неопределено, знач ИмяОбъекта, ИмяОсновногоПользователя =
Неопределено)
Если ИмяОсновногоПользователя = Неопределено Тогда
ИмяОсновногоПользователя =
Обработки.ек_ПользовательскиеПроверкиИОбработчики.Полно
еИмяНеуказанногоПользователя();
КонецЕсли;
СтруктураВсехНастроек =
Обработки.ек_ПользовательскиеПроверкиИОбработчики.Получ
итьСтруктуруВсехНастроек(ИмяОбъекта, ИмяОсновногоПользов
ателя);
Если СтруктураВсехНастроек.Количество()=0 Тогда
Возврат;
КонецЕсли;
Если СтрокаДерева = Неопределено Тогда
ЭлементыДерева = ДеревоОбъектов.ПолучитьЭлементы();
Для каждого ЭлементГруппа Из ЭлементыДерева Цикл
Если
СтрНачинаетсяС(ИмяОбъекта, ЭлементГруппа.ИдентификаторО
бъекта) Тогда
ЭлементыСтроки = ЭлементГруппа.ПолучитьЭлементы();
Для каждого ЭлементСтрока Из ЭлементыСтроки Цикл
Если ЭлементСтрока.ИдентификаторОбъекта = ИмяОбъекта
Тогда
СтрокаДерева = ЭлементСтрока;
Прервать;
КонецЕсли;
КонецЦикла;
КонецЕсли;
СтрокаДерева.ЕстьНастройки = Истина;
СтрокиНастройки = СтрокаДерева.ПолучитьЭлементы();
СтрокиНастройки.Очистить();
Для каждого КлючНастройка Из СтруктураВсехНастроек Цикл
Настройка = КлючНастройка.Значение;
Если ТипЗнч(Настройка)=Тип("Структура") И
Настройка.Свойство("ИдентификаторНастройки") Тогда
НоваяНастройка = СтрокиНастройки.Добавить();
НоваяНастройка.ИдентификаторОбъекта = ИмяОбъекта;
НоваяНастройка.Представление =
Настройка.ПредставлениеНастройки;
НоваяНастройка.ИдентификаторНастройки =
Настройка.ИдентификаторНастройки;
НоваяНастройка.ПриСозданииНаСервере =
Настройка.Свойство("ПриСозданииНаСервере");
НоваяНастройка.ПередЗаписьюНаСервере =
Настройка.Свойство("ПередЗаписьюНаСервере");
НоваяНастройка.ПроверкаДанных =
Настройка.Свойство("НастройкиКомпоновщика")
КонецЕсли;
КонецЦикла;
КонецПроцедуры // ЗаполнитьСписокНастроек()
;
&НаКлиенте
Процедура ДеревоОбъектовВыбор(Элемент, ВыбраннаяСтрока,
Поле, СтандартнаяОбработка)
ТекущиеДанные = Элементы.ДеревоОбъектов.ТекущиеДанные;
СтандартнаяОбработка = Ложь
ДобавитьОткрытьНастройку(ТекущиеДанные.ИдентификаторОб
ъекта, ТекущиеДанные.ИдентификаторНастройки);
КонецПроцедуры;
&НаКлиенте
Процедура

```

Рис. П2.1 Модуль формы выбора документов для контроля «Основная форма»

П2.1 Продолжение

```

ДобавитьОткрытьНастройку(ИдентификаторОбъекта,ИдентификаторНастройки,Отказ = Ложь)
Если СтрНайти(ИдентификаторОбъекта,".") > 0 Тогда
//Мы на уровне объекта метаданных
ПараметрыФормы = Новый Структура;
ПараметрыФормы.Вставить("ИдентификаторОбъекта",ИдентификаторОбъекта);
Если ЗначениеЗаполнено(ИдентификаторНастройки) Тогда
ИдентификаторНастройки = ИдентификаторНастройки;
Иначе
ИдентификаторНастройки =
ПолучитьНовыйИдентификаторНастройки();
КонецЕсли;
ПараметрыФормы.Вставить("ИдентификаторНастройки",ИдентификаторНастройки);
ОткрытьФорму("Обработка.ек_ПользовательскиеПроверкиИОбработчики.Форма.ФормаВвода",ПараметрыФормы,
ЭтотОбъект.Элементы.ДеревоОбъектов);
Иначе
//Мы на уровне группы
Сообщить("Выберите объект метаданного!");
Отказ = Истина;
КонецЕсли;
КонецПроцедуры // ДобавитьОткрытьНастройку()

```

```

&НаСервереБезКонтекста
Функция ПолучитьНовыйИдентификаторНастройки()
Возврат
Обработки.ек_ПользовательскиеПроверкиИОбработчики.ПолучитьНовыйИдентификаторНастройки();
КонецФункции // ()

```

```

&НаКлиенте
Процедура ДеревоОбъектовОбработкаВыбора(Элемент,
ВыбранноеЗначение, СтандартнаяОбработка)
Если ТипЗнч(ВыбранноеЗначение) = Тип("Строка") Тогда
ЗаполнитьСписокНастроек(Неопределено, ВыбранноеЗначение)
КонецЕсли;
//ТекущиеДанные = Элементы.ДеревоОбъектов.ТекущиеДанные;
//СтруктураНастроек =
ПолучитьСтруктуруНастроекНаСервере(ТекущиеДанные.ИдентификаторОбъекта);
//ТекущиеДанные.ПриСозданииНаСервере =
СтруктураНастроек.Свойство("ПриСозданииНаСервере");
//ТекущиеДанные.ПередЗаписьюНаСервере =
СтруктураНастроек.Свойство("ПередЗаписьюНаСервере");
//ТекущиеДанные.ПроверкаДанных =
СтруктураНастроек.Свойство("ПроверкаДанныхНастройкиКомпозитора");
КонецПроцедуры

```

```

&НаКлиенте
Процедура ДеревоОбъектовПередНачаломДобавления(Элемент,
Отказ, Копирование, Родитель, Группа, Параметр)
ТекущиеДанные = Элементы.ДеревоОбъектов.ТекущиеДанные;
Если Копирование Тогда
Если
ЗначениеЗаполнено(ТекущиеДанные.ИдентификаторОбъекта) И
ЗначениеЗаполнено(ТекущиеДанные.ИдентификаторНастройки)
Тогда
СкопироватьНастройкуНаСервере(ТекущиеДанные.ИдентификаторОбъекта,ТекущиеДанные.ИдентификаторНастройки);
Иначе
Отказ = Истина;
Сообщить("Строку "+ТекущиеДанные.Представление+" скопировать нельзя!");
КонецЕсли;
Иначе

```

```

ДобавитьОткрытьНастройку(ТекущиеДанные.ИдентификаторОбъекта,"",Отказ);
КонецЕсли;
КонецПроцедуры

```

```

&НаСервере
Функция
ПолучитьСтруктуруНастроекНаСервере(ИдентификаторОбъекта
МетаданныхСсылка)
Возврат
Обработки.ек_ПользовательскиеПроверкиИОбработчики.ПолучитьСтруктуруНастроек(ИдентификаторОбъектаМетаданныхСсылка);
КонецФункции

```

```

&НаСервере
Процедура ДобавитьНаСервере()
// Вставить содержимое обработчика.
КонецПроцедуры

```

```

&НаКлиенте
Процедура Добавить(Команда)
ДобавитьНаСервере();
КонецПроцедуры

```

```

&НаКлиенте
Процедура ОбработкаВыбора(ВыбранноеЗначение,
ИсточникВыбора)
//Вставить содержимое обработчика
КонецПроцедуры

```

```

&НаКлиенте
Процедура ДеревоОбъектовПередУдалением(Элемент, Отказ)
ТекущиеДанные = Элементы.ДеревоОбъектов.ТекущиеДанные;
Если
ЗначениеЗаполнено(ТекущиеДанные.ИдентификаторОбъекта) И
ЗначениеЗаполнено(ТекущиеДанные.ИдентификаторНастройки)
Тогда

```

```

УдалитьНастройкуНаСервере(ТекущиеДанные.ИдентификаторОбъекта,ТекущиеДанные.ИдентификаторНастройки);
Иначе
Отказ = Истина;
Сообщить("Строку "+ТекущиеДанные.Представление+" удалить нельзя!");
КонецЕсли;
КонецПроцедуры

```

```

&НаСервереБезКонтекста
Процедура
УдалитьНастройкуНаСервере(ИдентификаторОбъекта,ИдентификаторНастройки)
Обработки.ек_ПользовательскиеПроверкиИОбработчики.УдалитьНастройку(ИдентификаторОбъекта,ИдентификаторНастройки);
КонецПроцедуры // УдалитьНастройкуНаСервере()

```

```

&НаСервереБезКонтекста
Процедура
СкопироватьНастройкуНаСервере(ИдентификаторОбъекта,ИдентификаторНастройки)
Обработки.ек_ПользовательскиеПроверкиИОбработчики.СкопироватьНастройку(ИдентификаторОбъекта,ИдентификаторНастройки);
КонецПроцедуры // УдалитьНастройкуНаСервере()

```

П2.1 Продолжение

```

&НаКлиенте
Процедура
ДеревоОбъектовОбработкаЗаписиНового(НовыйОбъект,
Источник, СтандартнаяОбработка)
// Вставить содержимое обработчика.
КонецПроцедуры

&НаСервере
Процедура ЭкспортНастроекНаСервере(ПолноеИмяФайла)
МассивНастроекДляСохранения = Новый Массив;
ТекстЛога = "";
Для каждого ВыделеннаяНастройка Из
ЭтотОбъект.Элементы.ДеревоОбъектов.ВыделенныеСтроки
Цикл
СтрокаДанных =
ДеревоОбъектов.НайтиПоИдентификатору(ВыделеннаяНастрой
ка);
Если
ЗначениеЗаполнено(СтрокаДанных.ИдентификаторОбъекта) И
ЗначениеЗаполнено(СтрокаДанных.ИдентификаторНастройки)
Тогда
ЗначениеНастройки =
Обработки.ек_ПользовательскиеПроверкиИОбработчики.Получ
итьСтруктуруНастроек(СтрокаДанных.ИдентификаторОбъекта,,
СтрокаДанных.ИдентификаторНастройки);
ЗначениеНастройки.Вставить("ИдентификаторОбъекта",Строка
Данных.ИдентификаторОбъекта);
МассивНастроекДляСохранения.Добавить(ЗначениеНастройки);
ТекстЛога = ТекстЛога + Символы.ПС +
СтрокаДанных.ИдентификаторОбъекта+"."+СтрокаДанных.Пред
ставление;
КонецЕсли;
КонецЦикла;
Если МассивНастроекДляСохранения.Количество() > 0 Тогда
ЗаписьJSON = Новый ЗаписьJSON;
ПараметрыЗаписиJSON = Новый
ПараметрыЗаписиJSON(Символы.Таб);
ЗаписьJSON.ОткрытьФайл(ПолноеИмяФайла,,,ПараметрыЗапис
иJSON);
СериализаторXDTO.ЗаписатьJSON(ЗаписьJSON,
МассивНастроекДляСохранения,НазначениеТипаXML.Явное);
ЗаписьJSON.Закрыть();
Сообщить("В файл "+ПолноеИмяФайла+" сохранено
"+МассивНастроекДляСохранения.Количество()+"
настроек"+ТекстЛога);
Иначе
Сообщить("Выделите настройки для сохранения");
КонецЕсли;
КонецПроцедуры

&НаКлиенте
Процедура СохранитьНастройки(Команда)
ДиалогВыбораФайла = Новый
ДиалогВыбораФайла(РежимДиалогаВыбораФайла.Сохранение);
ДиалогВыбораФайла.Заголовок = НСтр("ru = 'Укажите файл для
сохранения настроек'");
ДиалогВыбораФайла.Фильтр = "Текстовый документ(*.txt)|
*.txt";
//ДиалогВыбораФайла.Каталог = Объект.КаталогКонфигурации;
//ДиалогВыбораФайла.ПроверятьСуществованиеФайла =
Истина;
Если ДиалогВыбораФайла.Выбрать() Тогда
ЭкспортНастроекНаСервере(ДиалогВыбораФайла.ПолноеИмяФай
ла);
КонецЕсли;
КонецПроцедуры

&НаСервере
Процедура ЗагрузитьНастройкиНаСервере(ПолноеИмяФайла)
ЧтениеJSON = Новый ЧтениеJSON;
ЧтениеJSON.ОткрытьФайл(ПолноеИмяФайла);
МассивНастроекДляЗагрузки =
СериализаторXDTO.ПрочитатьJSON(ЧтениеJSON);

```

```

ЧтениеJSON.Закрыть();
ТекстЛога = "Из файла "+ПолноеИмяФайла+" сохранено
"+МассивНастроекДляЗагрузки.Количество()+" настроек";
Для каждого ЗначениеНастройки Из
МассивНастроекДляЗагрузки Цикл
Обработки.ек_ПользовательскиеПроверкиИОбработчики.Сохран
итьСтруктуруНастроек(ЗначениеНастройки,ЗначениеНастройки.
ИдентификаторОбъекта);
ТекстЛога = ТекстЛога + Символы.ПС +
ЗначениеНастройки.ИдентификаторОбъекта+"."+ЗначениеНастр
ойки.ПредставлениеНастройки;
КонецЦикла;
Сообщить(ТекстЛога);
ЗаполнитьДеревоОбъектов();
КонецПроцедуры

&НаКлиенте
Процедура ЗагрузитьНастройки(Команда)
Диалог = Новый
ДиалогВыбораФайла(РежимДиалогаВыбораФайла.Открытие);
Диалог.Заголовок = "Выберите файл с настройками";
Диалог.Фильтр = "Текстовый документ(*.txt)*.txt";
Диалог.ПолноеИмяФайла = "";
Если Диалог.Выбрать() Тогда
ЗагрузитьНастройкиНаСервере(Диалог.ПолноеИмяФайла);
КонецЕсли;
КонецПроцедуры

&НаСервере
Процедура ПересохранитьНастройкиНаСервере()
ПустаяНастройка =
Обработки.ек_ПользовательскиеПроверкиИОбработчики.Получ
итьПустуюСтруктуруНастроекПользователя();
Для каждого Категория Из
ЭтотОбъект.ДеревоОбъектов.ПолучитьЭлементы() Цикл
Для каждого Метаданное Из Категория.ПолучитьЭлементы()
Цикл
СтруктураВсехНастроек =
Обработки.ек_ПользовательскиеПроверкиИОбработчики.Получ
итьСтруктуруВсехНастроек(Метаданное.ИдентификаторОбъект
а);
ЕстьНастройки = Ложь;
Для каждого Настройка Из СтруктураВсехНастроек Цикл
ЕстьНастройки = Истина;
Прервать;
КонецЦикла;
Если ЕстьНастройки Тогда
НоваяСтруктураВсехНастроек = Новый Структура;
Для каждого КлючЗначение Из СтруктураВсехНастроек Цикл
НоваяНастройка =
Обработки.ек_ПользовательскиеПроверкиИОбработчики.Получ
итьПустуюСтруктуруНастроекПользователя();
ЗначениеСохраненнойНастройки = КлючЗначение.Значение;
Если ТипЗнч(ЗначениеСохраненнойНастройки) =
Тип("Структура") Тогда
ЗаполнитьЗначенияСвойств(НоваяНастройка,ЗначениеСохранен
нойНастройки);
//Сбросить некорректные настройки
Для каждого ЗначениеНастройки Из НоваяНастройка Цикл
ИмяРеквизита = ЗначениеНастройки.Ключ;
Если ТипЗнч(НоваяНастройка[ИмяРеквизита]) <>
ТипЗнч(ПустаяНастройка[ИмяРеквизита]) Тогда
НоваяНастройка[ИмяРеквизита] =
ПустаяНастройка[ИмяРеквизита];
КонецЕсли;
КонецЦикла;
НоваяСтруктураВсехНастроек.Вставить(НоваяНастройка.Идент
ификаторНастройки,НоваяНастройка);
КонецЕсли;
КонецЦикла;
Обработки.ек_ПользовательскиеПроверкиИОбработчики.Сохран
итьСтруктуруВсехНастроек(НоваяСтруктураВсехНастроек,Мета
данное.ИдентификаторОбъекта);

```

```
Иначе
Обработки.ек_ПользовательскиеПроверкиИОбработчики.Удалить
СтруктуруВсехНастроек(Метаданное.ИдентификаторОбъекта);
КонецЕсли;
КонецЦикла;
КонецЦикла;
ЗаполнитьДеревоОбъектов();
КонецПроцедуры
```

```
&НаСервере
Процедура ПриСозданииНаСервере(Отказ,
СтандартнаяОбработка)
Если НЕ
Параметры.Свойство("ИдентификаторОбъекта",ЭтотОбъект.ИдентификаторОбъекта) Тогда
Отказ = Истина;
Возврат;
КонецЕсли;
Параметры.Свойство("ИдентификаторНастройки",ЭтотОбъект.ИдентификаторНастройки);
ИмяОсновногоПользователя =
Обработки.ек_ПользовательскиеПроверкиИОбработчики.ПолноеИмяНеуказанногоПользователя();
СтруктураНастроек =
Обработки.ек_ПользовательскиеПроверкиИОбработчики.ПолучитьСтруктуруНастроек(ЭтотОбъект.ИдентификаторОбъекта,ИмяОсновногоПользователя,ЭтотОбъект.ИдентификаторНастройки);
СтруктураНастроек.Свойство("ПриСозданииНаСервере",ЭтотОбъект.ТекстПриСозданииНаСервере);
СтруктураНастроек.Свойство("ПередЗаписьюНаСервере",ЭтотОбъект.ТекстПередЗаписьюНаСервере);
СтруктураНастроек.Свойство("ТекстСообщенияОбОшибке",ЭтотОбъект.ТекстСообщенияОбОшибке);
СтруктураНастроек.Свойство("ИдентификаторНастройки",ЭтотОбъект.ИдентификаторНастройки);
СтруктураНастроек.Свойство("ИспользоватьПриЗаписи",ЭтотОбъект.ИспользоватьПриЗаписи);
```

```
СтруктураНастроек.Свойство("ИспользоватьПриПросмотре",ЭтотОбъект.ИспользоватьПриПросмотре);
СтруктураНастроек.Свойство("ИспользоватьПриИзменении",ЭтотОбъект.ИспользоватьПриИзменении);
СтруктураНастроек.Свойство("ИспользоватьВСписках",ЭтотОбъект.ИспользоватьВСписках);
СтруктураНастроек.Свойство("ПредставлениеНастройки",ЭтотОбъект.ПредставлениеНастройки);
Если НЕ
ЗначениеЗаполнено(ЭтотОбъект.ИдентификаторНастройки)
Тогда
ЭтотОбъект.ИдентификаторНастройки =
Обработки.ек_ПользовательскиеПроверкиИОбработчики.ПолучитьНовыйИдентификаторНастройки();
ЭтотОбъект.Заголовок = ЭтотОбъект.Заголовок + " (создание)";
Иначе
ЭтотОбъект.Заголовок = ЭтотОбъект.Заголовок + " " +
ЭтотОбъект.ПредставлениеНастройки + " (изменение)";
КонецЕсли;
ТекстЗапроса =
Обработки.ек_ПользовательскиеПроверкиИОбработчики.ТекстЗапроса(ЭтотОбъект.ИдентификаторОбъекта);
ИнициализироватьКомпоновщикНастроек();
НастройкиКомпоновщика="";
Если
```

```
&НаКлиенте
Процедура ПересохранитьНастройки(Команда)
ПересохранитьНастройкиНаСервере();
КонецПроцедуры
```

```
&НаКлиенте
Процедура Обновить(Команда)
ЗаполнитьДеревоОбъектов();
КонецПроцедуры
```

```
СтруктураНастроек.Свойство("НастройкиКомпоновщика",НастройкиКомпоновщика) Тогда
КомпоновщикНастроек.ЗагрузитьНастройки(НастройкиКомпоновщика);
КонецЕсли;
МассивВыбранныхПользователей = Новый Массив;
ЕстьМассивВыбранныхПользователей =
СтруктураНастроек.Свойство("МассивВыбранныхПользователей",МассивВыбранныхПользователей);
ВсеПользователиИБ =
ПользователиИнформационнойБазы.ПолучитьПользователей();
Для Каждого ПользовательИБ Из ВсеПользователиИБ Цикл
СтрокаПользователь =
ЭтотОбъект.ТаблицаПользователей.Добавить();
СтрокаПользователь.Имя = ПользовательИБ.Имя;
СтрокаПользователь.ПолноеИмя = ПользовательИБ.ПолноеИмя;
СтрокаПользователь.Выбран = (НЕ
ЕстьМассивВыбранныхПользователей)
ИЛИ
(МассивВыбранныхПользователей.Найти(ПользовательИБ.Имя)
<>Неопределено);
КонецЦикла;
КонецПроцедуры
```

```
&НаСервере
Функция ЗаписатьНаСервере()
СтруктураНастроек = Новый Структура;
ТекстПредставлениеНастройки = "";
Если
ЭтотОбъект.КомпоновщикНастроек.Настройки.Отбор.Элементы.Количество() > 0 Тогда
СтруктураНастроек.Вставить("НастройкиКомпоновщика",ЭтотОбъект.КомпоновщикНастроек.Настройки);
СтруктураНастроек.Вставить("ТекстСообщенияОбОшибке",ЭтотОбъект.ТекстСообщенияОбОшибке);
ТекстПредставлениеНастройки = ТекстПредставлениеНастройки + ЭтотОбъект.КомпоновщикНастроек.Настройки.Отбор;
КонецЕсли;
Если
ЗначениеЗаполнено(ЭтотОбъект.ТекстПриСозданииНаСервере)
Тогда
СтруктураНастроек.Вставить("ПриСозданииНаСервере",ЭтотОбъект.ТекстПриСозданииНаСервере);
ТекстПредставлениеНастройки = ТекстПредставлениеНастройки + " ПриСоздании";
КонецЕсли;
Если
ЗначениеЗаполнено(ЭтотОбъект.ТекстПередЗаписьюНаСервере)
Тогда
СтруктураНастроек.Вставить("ПередЗаписьюНаСервере",ЭтотОбъект.ТекстПередЗаписьюНаСервере);
```

Рис. П2.2. Модуль формы ввода/выбора реквизитов документа «Форма ввода»

П.2.2 Продолжение

```
ТекстПредставлениеНастройки = ТекстПредставлениеНастройки
+ " ПередЗаписью";
```

```
КонецЕсли;
Если НЕ
```

```

ЗначениеЗаполнено(ЭтотОбъект.ПредставлениеНастройки)
Тогда
ЭтотОбъект.ПредставлениеНастройки =
ТекстПредставлениеНастройки;
КонецЕсли;
СтруктураНастроек.Вставить("ПредставлениеНастройки",ЭтотОбъект.ПредставлениеНастройки);
СтруктураНастроек.Вставить("ИдентификаторНастройки",ЭтотОбъект.ИдентификаторНастройки);
СтруктураНастроек.Вставить("ИспользоватьПриЗаписи",ЭтотОбъект.ИспользоватьПриЗаписи);
СтруктураНастроек.Вставить("ИспользоватьПриПросмотре",ЭтотОбъект.ИспользоватьПриПросмотре);
СтруктураНастроек.Вставить("ИспользоватьПриИзменении",ЭтотОбъект.ИспользоватьПриИзменении);
СтруктураНастроек.Вставить("ИспользоватьВСписках",ЭтотОбъект.ИспользоватьВСписках);
ВыбранныеСтроки =
ЭтотОбъект.ТаблицаПользователей.НайтиСтроки(НовыйСтруктура("Выбран",Истина));
МассивВыбранныхПользователей =
ЭтотОбъект.ТаблицаПользователей.Выгрузить(ВыбранныеСтроки,"Имя").ВыгрузитьКолонку("Имя");
СтруктураНастроек.Вставить("МассивВыбранныхПользователей",МассивВыбранныхПользователей);
Обработки.ек_ПользовательскиеПроверкиИОбработчики.СохранитьСтруктуруНастроек(СтруктураНастроек,ЭтотОбъект.ИдентификаторОбъекта);
КонецФункции

&НаКлиенте
Процедура Записать(Команда)
СтруктураНастроек = ЗаписатьНаСервере();
КонецПроцедуры

&НаКлиенте
Процедура ЗаписатьИЗаккрыть(Команда)
СтруктураНастроек = ЗаписатьНаСервере();
ОповеститьОВыборе(ЭтотОбъект.ИдентификаторОбъекта);
КонецПроцедур
#Область СлужебныеПроцедурыИФункции
&НаСервере
Процедура ИнициализироватьКомпоновщикНастроек()
СхемаКомпоновкиДанных =
СхемаКомпоновкиДанных(ТекстЗапроса);
АдресСхемы =
ПоместитьВоВременноеХранилище(СхемаКомпоновкиДанных,УникальныйИдентификатор);
КомпоновщикНастроек.Инициализировать(НовыйИсточникДоступныхНастроекКомпоновкиДанных(АдресСхемы));
КомпоновщикНастроек.ЗагрузитьНастройки(СхемаКомпоновкиДанных.НастройкиПоУмолчанию);
//ЭлементОтбора =
КомпоновщикНастроек.Настройки.Отбор.Элементы.Добавить(Тип("ЭлементОтбораКомпоновкиДанных"));
//ЭлементОтбора.ЛевоеЗначение = НовыйПолеКомпоновкиДанных("Ссылка");
//ЭлементОтбора.ВидСравнения =
ВидСравненияКомпоновкиДанных.НеВСписке;
//ЭлементОтбора.ПравоеЗначение = НовыйСписокЗначений;
УстановитьНастройкуСтруктурыВыводаРезультата(КомпоновщикНастроек.Настройки);
КонецПроцедуры

&НаСервере

```

```

Процедура
УстановитьНастройкуСтруктурыВыводаРезультата(Настройки,ДляИзменения = Ложь)
Настройки.Структура.Очистить();
Настройки.Выбор.Элементы.Очистить();
ГруппировкаКомпоновкиДанных =
Настройки.Структура.Добавить(Тип("ГруппировкаКомпоновкиДанных"));
ГруппировкаКомпоновкиДанных.Выбор.Элементы.Добавить(Тип("АвтоВыборноеПолеКомпоновкиДанных"));
ГруппировкаКомпоновкиДанных.Использование = Истина;
ПолеГруппировки =
ГруппировкаКомпоновкиДанных.ПоляГруппировки.Элементы.Добавить(Тип("ПолеГруппировкиКомпоновкиДанных"));
ПолеГруппировки.Поле = НовыйПолеКомпоновкиДанных("Ссылка");
ПолеГруппировки.Использование = Истина;
ПолеВыбора =
Настройки.Выбор.Элементы.Добавить(Тип("ВыборноеПолеКомпоновкиДанных"));
ПолеВыбора.Поле = НовыйПолеКомпоновкиДанных("Ссылка");
ПолеВыбора.Использование = Истина;
КонецПроцедуры

```

```

&НаСервере
Функция СхемаКомпоновкиДанных(ТекстЗапроса)
Возврат
Обработки.ек_ПользовательскиеПроверкиИОбработчики.СхемаКомпоновкиДанных(ТекстЗапроса);
КонецФункции

```

```

&НаКлиенте
Процедура СнятьФлажкиУВсехПользователей(Команда)
Для каждого ЭлементТаблицы Из
ЭтотОбъект.ТаблицаПользователей Цикл
ЭлементТаблицы.Выбран = Ложь;
КонецЦикла;
КонецПроцедуры

```

```

&НаКлиенте
Процедура УстановитьФлажкиУВсехПользователей(Команда)
Для каждого ЭлементТаблицы Из
ЭтотОбъект.ТаблицаПользователей Цикл
ЭлементТаблицы.Выбран = Истина;
КонецЦикла;
КонецПроцедуры

```

```

&НаКлиенте
Процедура Инструкция(Команда)
ЗапуститьПриложение("http://1c.ekaruk.info/public/359597/");
КонецПроцедуры

```

```

&НаКлиенте
Процедура ВсеНастройки(Команда)
ОткрытьФорму("Обработка.ек_ПользовательскиеПроверкиИОбработчики.Форма.ФормаОсновная");
ЭтотОбъект.Заккрыть();
КонецПроцедуры

```

```

&НаКлиенте
Процедура ПредставлениеНастройкиОчистка(Элемент,СтандартнаяОбработка)
ЭтотОбъект.ПредставлениеНастройки = "";
// Вставить содержимое обработчика.
КонецПроцедуры
#КонецОбласти

```

```

//Если
ТипЗнач(ИдентификаторОбъектаМетаданныхСсылка)=Тип("СправочникСсылка.ИдентификаторыОбъектовМетаданных") Тогда

```

```

Функция ПолучитьСтруктуруНастроек(КлючНастроек="",ИмяПользователя = Неопределено,ИдентификаторНастройки="") Экспорт

```

```

//      КлючНастроек =
ИдентификаторОбъектаМетаданныхСсылка.ПолноеИмя;
//КонецЕсли;
Если НЕ ЗначениеЗаполнено(КлючНастроек) Тогда
ВызватьИсключение "Не указан КлючНастроек";
КонецЕсли;
СтруктураВсехНастроек =
ПолучитьСтруктуруВсехНастроек(КлючНастроек,
ИмяПользователя);
Если ЗначениеЗаполнено(ИдентификаторНастройки) И
СтруктураВсехНастроек.Свойство(ИдентификаторНастройки)
Тогда
СтруктураНастроек =
СтруктураВсехНастроек[ИдентификаторНастройки];
Иначе
Если СтруктураВсехНастроек.Количество()=0 ИЛИ
ЗначениеЗаполнено(ИдентификаторНастройки) Тогда
СтруктураНастроек = Новый
Структура("ИдентификаторНастройки",ИдентификаторНастройк
и);
Иначе
Для каждого Настройка Из СтруктураВсехНастроек Цикл
СтруктураНастроек = Настройка.Значение;
Прервать;
КонецЦикла;
КонецЕсли;
КонецЕсли;
Возврат СтруктураНастроек;
КонецФункции //
ПолучитьСтруктуруДополнительныхСведений()

Функция ПолучитьСтруктуруВсехНастроек(КлючНастроек="",
ИмяПользователя = Неопределено) Экспорт
//Если НЕ ЗначениеЗаполнено(КлючНастроек) Тогда
///^ВызватьИсключение "Не удалось определить
КлючНастроек";
//Возврат Новый Структура();
//КонецЕсли;
СтруктураВсехНастроек =
ХранилищеОбщихНастроек.Загрузить("ек_ПользовательскиеПр
оверкиИОбработчики",КлючНастроек,,ИмяПользователя);

Если СтруктураВсехНастроек = Неопределено ИЛИ
ТипЗнч(СтруктураВсехНастроек) <> Тип("Структура") Тогда
СтруктураВсехНастроек = Новый Структура();
КонецЕсли;
Возврат СтруктураВсехНастроек;
КонецФункции //
ПолучитьСтруктуруДополнительныхСведений()

Процедура СохранитьСтруктуруНастроек(СтруктураНастроек,
КлючНастроек="") Экспорт
Если НЕ ЗначениеЗаполнено(КлючНастроек) Тогда
ВызватьИсключение "Не удалось определить КлючНастроек";
КонецЕсли;
ИмяОсновногоПользователя =
Обработки.ек_ПользовательскиеПроверкиИОбработчики.Полно
еИмяНеуказанногоПользователя();
СтруктураВсехНастроек =
ПолучитьСтруктуруВсехНастроек(КлючНастроек,
ИмяОсновногоПользователя);
СтруктураВсехНастроек.Вставить(СтруктураНастроек.Идентиф
икаторНастройки,СтруктураНастроек);
СохранитьСтруктуруВсехНастроек(СтруктураВсехНастроек,Кл
ючНастроек);
КонецПроцедуры

Процедура
СохранитьСтруктуруВсехНастроек(СтруктураВсехНастроек,Кл
ючНастроек) Экспорт
ИмяОсновногоПользователя =
Обработки.ек_ПользовательскиеПроверкиИОбработчики.Полно
еИмяНеуказанногоПользователя();
ИмяНеуказанногоПользователя();
ХранилищеОбщихНастроек.Сохранить("ек_ПользовательскиеПр
оверкиИОбработчики",КлючНастроек,СтруктураВсехНастроек,,
ИмяОсновногоПользователя);
ПерезаполнитьСтруктуруНастроекПользователей(КлючНастроек
,СтруктураВсехНастроек)
КонецПроцедуры

Процедура УдалитьСтруктуруВсехНастроек(КлючНастроек)
Экспорт
ХранилищеОбщихНастроек.Удалить("ек_ПользовательскиеПров
еркиИОбработчики",КлючНастроек,Неопределено);
КонецПроцедуры

Процедура
УдалитьНастройку(КлючНастроек,ИдентификаторНастройки)
Экспорт
//Если
ТипЗнч(ИдентификаторОбъектаМетаданныхСсылка)=Тип("Спра
вочникСсылка.ИдентификаторыОб
ъектовМетаданных") Тогда
//КлючНастроек =
ИдентификаторОбъектаМетаданныхСсылка.ПолноеИмя;
//КонецЕсли;
Если НЕ ЗначениеЗаполнено(КлючНастроек) Тогда
ВызватьИсключение "Не удалось определить КлючНастроек";
КонецЕсли;
ИмяОсновногоПользователя =
Обработки.ек_ПользовательскиеПроверкиИОбработчики.Полно
еИмяНеуказанногоПользователя();
СтруктураВсехНастроек =
ПолучитьСтруктуруВсехНастроек(КлючНастроек,
ИмяОсновногоПользователя);
СтруктураВсехНастроек.Удалить(ИдентификаторНастройки);
ХранилищеОбщихНастроек.Сохранить("ек_ПользовательскиеПр
оверкиИОбработчики",КлючНастроек,СтруктураВсехНастроек,,
ИмяОсновногоПользователя);
ПерезаполнитьСтруктуруНастроекПользователей(КлючНастроек
,СтруктураВсехНастроек)
КонецПроцедуры

Процедура
СкопироватьНастройку(КлючНастроек,ИдентификаторНастройк
и) Экспорт
Если НЕ ЗначениеЗаполнено(КлючНастроек) Тогда
ВызватьИсключение "Не указан КлючНастроек";
КонецЕсли;
ИмяОсновногоПользователя =
Обработки.ек_ПользовательскиеПроверкиИОбработчики.Полно
еИмяНеуказанногоПользователя();
СтруктураВсехНастроек =
ПолучитьСтруктуруВсехНастроек(КлючНастроек,
ИмяОсновногоПользователя);
КопируемаяНастройка =
СтруктураВсехНастроек[ИдентификаторНастройки];
НоваяНастройка =
ПолучитьПустуюСтруктуруНастроекПользователя();
ЗаполнитьЗначенияСвойств(НоваяНастройка,КопируемаяНастро
йка);
ИдентификаторНастройки =
ПолучитьНовыйИдентификаторНастройки();
НоваяНастройка.ИдентификаторНастройки =
ИдентификаторНастройки;
СтруктураВсехНастроек.Вставить(НоваяНастройка.Идентифика
торНастройки,НоваяНастройка);

```

Рис. П2.3 Модуль обработки введенной информации

```
ХранилищеОбщихНастроек.Сохранить("ек_ПользовательскиеПроверкиИОбработчики",КлючНастроек,СтруктураВсехНастроек,,ИмяОсновногоПользователя);
ПерезаполнитьСтруктуруНастроекПользователей(КлючНастроек,СтруктураВсехНастроек)
КонецПроцедуры
```

```
Процедура
ПерезаполнитьСтруктуруНастроекПользователей(КлючНастроек,СтруктураВсехНастроек)
ВсеПользователиИБ =
ПользователиИнформационнойБазы.ПолучитьПользователей();
Для Каждого ПользовательИБ Из ВсеПользователиИБ Цикл
СтруктураВсехНастроекПользователя = Новый Структура;
Для каждого КлючСтруктураНастроек Из
СтруктураВсехНастроек Цикл
СтруктураНастроекПользователя =
КлючСтруктураНастроек.Значение;
МассивВыбранныхПользователей = Новый Массив;
ЕстьМассивВыбранныхПользователей =
СтруктураНастроекПользователя.Свойство("МассивВыбранныхПользователей",МассивВыбранныхПользователей);
Если ЕстьМассивВыбранныхПользователей И
МассивВыбранныхПользователей.Найти(ПользовательИБ.Имя)<
>Неопределено Тогда
СтруктураВсехНастроекПользователя.Вставить(СтруктураНастроекПользователя.ИдентификаторНастройки,СтруктураНастроекПользователя);
КонецЕсли;
КонецЦикла;
ХранилищеОбщихНастроек.Сохранить("ек_ПользовательскиеПроверкиИОбработчики",КлючНастроек,СтруктураВсехНастроекПользователя,,ПользовательИБ.Имя);
КонецЦикла;
КонецПроцедуры
```

```
Функция ПолучитьПустуюСтруктуруНастроекПользователя() Экспорт
СтруктураНастроек = Новый Структура;
СтруктураНастроек.Вставить("НастройкиКомпоновщика",Новый НастройкиКомпоновкиДанных);
СтруктураНастроек.Вставить("ТекстСообщенияОбОшибке","");
СтруктураНастроек.Вставить("ПриСозданииНаСервере","");
СтруктураНастроек.Вставить("ПередЗаписьюНаСервере","");
СтруктураНастроек.Вставить("ПредставлениеНастройки","");
СтруктураНастроек.Вставить("ИдентификаторНастройки","");
СтруктураНастроек.Вставить("ИспользоватьПриЗаписи",Ложь);
СтруктураНастроек.Вставить("ИспользоватьПриПросмотре",Ложь);
СтруктураНастроек.Вставить("ИспользоватьПриИзменении",Ложь);
СтруктураНастроек.Вставить("ИспользоватьВСписках",Ложь);
СтруктураНастроек.Вставить("ЗапрещатьФормированиеОтчета",Ложь);
СтруктураНастроек.Вставить("МассивВыбранныхПользователей",Новый Массив);
Возврат СтруктураНастроек;
КонецФункции //
ПолучитьПустуюСтруктуруНастроекПользователя()
```

```
Функция ПолучитьНовыйИдентификаторНастройки() Экспорт
Возврат "_" + СтрЗаменить(Строка(Новый УникальныйИдентификатор),"-","");
КонецФункции // ПолучитьНовыйИдентификаторНастройки()
```

```
#Область ОбработкаСобытийФорм
Процедура ПриСозданииНаСервере(Форма, Отказ,
СтандартнаяОбработка, ИмяОсновногоОбъекта="Объект")
Экспорт
```

```
КонецЕсли;
НастройкиКомпоновщика="";
ТекстСообщенияОбОшибке = "";
Если
СтруктураНастроек.Свойство("НастройкиКомпоновщика",НастройкиКомпоновщика) Тогда
```

```
КлючНастройки = ПолучитьКлючНастройкиПоФорме(Форма);
СтруктураВсехНастроек =
Обработки.ек_ПользовательскиеПроверкиИОбработчики.ПолучитьСтруктуруВсехНастроек(КлючНастройки,Неопределено);
СсылкаНаОбъект = Форма[ИмяОсновногоОбъекта].Ссылка;
Для каждого КлючНастройка Из СтруктураВсехНастроек Цикл
СтруктураНастроек = КлючНастройка.Значение;
ИсполняемыйКод = "";
Если
СтруктураНастроек.Свойство("ПриСозданииНаСервере",ИсполняемыйКод) Тогда
Попытка
Выполнить(ИсполняемыйКод);
Исключение
ВызватьИсключение;
КонецПопытки;
КонецЕсли;
Если НЕ ЗначениеЗаполнено(СсылкаНаОбъект) Тогда
Продолжить;
КонецЕсли;
НастройкиКомпоновщика="";
ТекстСообщенияОбОшибке = "";
Если
СтруктураНастроек.Свойство("НастройкиКомпоновщика",НастройкиКомпоновщика) Тогда
Если
СтруктураНастроек.Свойство("ИспользоватьПриПросмотре")
И СтруктураНастроек.ИспользоватьПриПросмотре
И НЕ
СсылкаСоответствуетНастройкамКомпоновки(КлючНастройки,СсылкаНаОбъект,НастройкиКомпоновщика) Тогда
Отказ = Истина;
ТекстСообщенияОбОшибке = "";
Если НЕ
(СтруктураНастроек.Свойство("ТекстСообщенияОбОшибке",ТекстСообщенияОбОшибке) И
ЗначениеЗаполнено(ТекстСообщенияОбОшибке)) Тогда
ТекстСообщенияОбОшибке = "Просмотр
"""+СсылкаНаОбъект+"""" запрещен!";
КонецЕсли;
Сообщить(ТекстСообщенияОбОшибке);
КонецЕсли;
КонецЕсли;
КонецЦикла;
КонецПроцедуры
```

```
Процедура ПередЗаписьюНаСервере(Форма, Отказ,
ТекущийОбъект, ПараметрыЗаписи,
ИмяОсновногоОбъекта="Объект") Экспорт
КлючНастройки = ПолучитьКлючНастройкиПоФорме(Форма);
СтруктураВсехНастроек =
Обработки.ек_ПользовательскиеПроверкиИОбработчики.ПолучитьСтруктуруВсехНастроек(КлючНастройки,Неопределено);
СсылкаНаОбъект = ТекущийОбъект.Ссылка;
Для каждого КлючНастройка Из СтруктураВсехНастроек Цикл
СтруктураНастроек = КлючНастройка.Значение;
// СтруктураНастроек =
ПолучитьСтруктуруНастроек(Неопределено, КлючНастройки);
ИсполняемыйКод = "";
Если
СтруктураНастроек.Свойство("ПередЗаписьюНаСервере",ИсполняемыйКод) Тогда
Попытка
Выполнить(ИсполняемыйКод);
Исключение
ВызватьИсключение;
КонецПопытки;
КонецЕсли;
Если НЕ ЗначениеЗаполнено(СсылкаНаОбъект) Тогда
Продолжить;
```

П.2.3 Продолжение

```
Если
СтруктураНастроек.Свойство("ИспользоватьПриИзменении")
И СтруктураНастроек.ИспользоватьПриИзменении
И НЕ
СсылкаСоответствуетНастройкамКомпоновки(КлючНастройки,СсылкаНаОбъект,НастройкиКомпоновщика) Тогда
```

```
Отказ = Истина;
ТекстСообщенияОбОшибке = "";
Если НЕ
(СтруктураНастроек.Свойство("ТекстСообщенияОбОшибке",ТекстСообщенияОбОшибке) И
ЗначениеЗаполнено(ТекстСообщенияОбОшибке)) Тогда
ТекстСообщенияОбОшибке = "Изменение
"""+СсылкаНаОбъект+"""" запрещено!";
КонецЕсли;
Сообщить(ТекстСообщенияОбОшибке);
КонецЕсли;
КонецЕсли;
КонецЦикла;
КонецПроцедуры
```

```
Процедура ПриЗаписиНаСервере(Форма, Отказ,
ТекущийОбъект, ПараметрыЗаписи) Экспорт
КлючНастройки = ПолучитьКлючНастройкиПоФорме(Форма);
СтруктураВсехНастроек =
Обработки.ек_ПользовательскиеПроверкиИОбработчики.ПолучитьСтруктуруВсехНастроек(КлючНастройки,Неопределено);
Для каждого КлючНастройка Из СтруктураВсехНастроек Цикл
СтруктураНастроек = КлючНастройка.Значение;
НастройкиКомпоновщика="";
Если
СтруктураНастроек.Свойство("НастройкиКомпоновщика",НастройкаКомпоновщика) Тогда
Если СтруктураНастроек.Свойство("ИспользоватьПриЗаписи")
И СтруктураНастроек.ИспользоватьПриЗаписи
И НЕ
СсылкаСоответствуетНастройкамКомпоновки(КлючНастройки,ТекущийОбъект.Ссылка,НастройкаКомпоновщика) Тогда
Отказ = Истина;
ТекстСообщенияОбОшибке = "";
Если НЕ
(СтруктураНастроек.Свойство("ТекстСообщенияОбОшибке",ТекстСообщенияОбОшибке) И
ЗначениеЗаполнено(ТекстСообщенияОбОшибке)) Тогда
ТекстСообщенияОбОшибке = "Введенные данные не
соответствуют условию, указанному в настройке
""Пользовательские проверки и обработки""";
КонецЕсли;
Сообщить(ТекстСообщенияОбОшибке);
КонецЕсли;
КонецЕсли;
КонецЦикла;
КонецПроцедуры
```

```
Функция
СсылкаСоответствуетНастройкамКомпоновки(КлючНастройки,СсылкаНаЭлемент,НастройкаКомпоновщика)
ЭлементОтбора =
НастройкаКомпоновщика.Отбор.Элементы.Добавить(Тип("ЭлементОтбораКомпоновкиДанных"));
ЭлементОтбора.ЛевоеЗначение = Новый
ПолеКомпоновкиДанных("Ссылка");
ЭлементОтбора.ВидСравнения =
ВидСравненияКомпоновкиДанных.Равно;
ЭлементОтбора.ПравоеЗначение = СсылкаНаЭлемент;
Результат = Новый ТаблицаЗначений;
//ТекстЗапроса = "ВЫБРАТЬ Таблица.Ссылка ИЗ
"+КлючНастройки+" КАК Таблица";
ТекстЗапроса = ТекстЗапроса(КлючНастройки);
СхемаКомпоновкиДанных =
СхемаКомпоновкиДанных(ТекстЗапроса);
КомпоновщикНастроекКомпоновкиДанных = Новый
КомпоновщикНастроекКомпоновкиДанных;
```

```
Процедура
ПрименитьУсловиеОтбораКСписку(ДинамическийСписок)
Экспорт
КлючНастройки = ДинамическийСписок.ОсновнаяТаблица;
СтруктураВсехНастроек =
Обработки.ек_ПользовательскиеПроверкиИОбработчики.ПолучитьСтруктуруВсехНастроек(КлючНастройки,Неопределено);
Для каждого КлючНастройка Из СтруктураВсехНастроек Цикл
СтруктураНастроек = КлючНастройка.Значение;
```

```
АдресСхемы =
ПоместитьВоВременноеХранилище(СхемаКомпоновкиДанных,
Новый УникальныйИдентификатор);
КомпоновщикНастроекКомпоновкиДанных.Инициализировать(Новый
ИсточникДоступныхНастроекКомпоновкиДанных(АдресСхемы)
);
КомпоновщикНастроекКомпоновкиДанных.ЗагрузитьНастройки(НастройкиКомпоновщика);
КомпоновщикМакета = Новый
КомпоновщикМакетаКомпоновкиДанных;
//Попытка
МакетКомпоновкиДанных =
КомпоновщикМакета.Выполнить(СхемаКомпоновкиДанных,
КомпоновщикНастроекКомпоновкиДанных.Настройки, , ,
Тип("ГенераторМакетаКомпоновкиДанныхДляКоллекцииЗначений"));
//Исключение
//Возврат Ложь;
//КонецПопытки;
ПроцессорКомпоновкиДанных = Новый
ПроцессорКомпоновкиДанных;
ПроцессорКомпоновкиДанных.Инициализировать(МакетКомпоновкиДанных);
ПроцессорВывода = Новый
ПроцессорВыводаРезультатаКомпоновкиДанныхВКоллекциюЗначений;
Результат =
ПроцессорВывода.Вывести(ПроцессорКомпоновкиДанных);
Возврат Результат.Количество() <> 0;
КонецФункции // СсылкаСоответствуетСхемеКомпоновки()
Функция ПолучитьКлючНастройкиПоФорме(Форма) Экспорт
Возврат
Сред(Форма.ИмяФормы,1,СтрНайти(Форма.ИмяФормы,".",НаправлениеПоиска.СНачала,1,2)-1);
КонецФункции // ПолучитьКлючНастройкиПоФорме()
```

```
&НаСервере
Процедура
ПрименитьУсловиеОтбораКоВсемДинамическимСпискамСпискам(Форма) Экспорт
МассивРеквизитовФормы = Форма.ПолучитьРеквизиты();
ТипДинамическийСписок = Новый
ОписаниеТипов("ДинамическийСписок");
Для каждого РеквизитФормы Из МассивРеквизитовФормы Цикл
```

```
Если РеквизитФормы.ТипЗначения = ТипДинамическийСписок
Тогда
ОсновнаяТаблица =
Форма[РеквизитФормы.Имя].ОсновнаяТаблица;
Если СтрНачинаетсяС(ОсновнаяТаблица,"ЖурналДокументов.")
ТогдаОбработки.ек_ПользовательскиеПроверкиИОбработчики.ПрименитьУсловиеОтбораКСпискуЖурнала(Форма[РеквизитФормы.Имя]);
ИначеЕсли СтрНачинаетсяС(ОсновнаяТаблица,"Документ.")
ИЛИ СтрНачинаетсяС(ОсновнаяТаблица,"Справочник.")
ИЛИ
СтрНачинаетсяС(ОсновнаяТаблица,"ПланВидовХарактеристик.")
Тогда
Обработки.ек_ПользовательскиеПроверкиИОбработчики.ПрименитьУсловиеОтбораКСписку(Форма[РеквизитФормы.Имя]);
КонецЕсли;
КонецЕсли;
КонецЦикла;
КонецПроцедуры
```

П2.3 Продолжение

```
НастройкаКомпоновщика="";
Если
СтруктураНастроек.Свойство("НастройкиКомпоновщика",НастройкаКомпоновщика) Тогда
Если СтруктураНастроек.Свойство("ИспользоватьВСписках")
И СтруктураНастроек.ИспользоватьВСписках Тогда
СкопироватьУсловияОтбора(НастройкаКомпоновщика.Отбор.Элементы,ДинамическийСписок.Отбор.Элементы);
КонецЕсли;
```

КонецЕсли;
КонецЦикла;
КонецПроцедуры

Процедура
ПрименитьУсловиеОтбораКСпискуЖурнала(ДинамическийСписок) Экспорт
КлючНастройки = ДинамическийСписок.ОсновнаяТаблица;
ИмяЖурнала =
СтрЗаменить(КлючНастройки,"ЖурналДокументов.", "");
КоллекцияДокументов =
Метаданные.ЖурналыДокументов[ИмяЖурнала].РегистрируемыеДокументы;
НоваяГруппаОтбораОбщая =
ДинамическийСписок.Отбор.Элементы.Добавить(Тип("ГруппаЭлементовОтбораКомпоновкиДанных"));
НоваяГруппаОтбораОбщая.ТипГруппы =
ТипГруппыЭлементовОтбораКомпоновкиДанных.ГруппаИли;
Для каждого МетаданноеДокумент Из КоллекцияДокументов Цикл
ИмяТипаДанных =
"ДокументСсылка."+МетаданноеДокумент.Имя;
КлючНастройки = МетаданноеДокумент.ПолноеИмя();
СтруктураВсехНастроек =
Обработки.ек_ПользовательскиеПроверкиИОбработчики.ПолучитьСтруктуруВсехНастроек(КлючНастройки,Неопределено);
ЕстьУсловиеОтбора = Ложь;
Для каждого КлючНастройка Из СтруктураВсехНастроек Цикл
СтруктураНастроек = КлючНастройка.Значение;
Если СтруктураНастроек.Свойство("ИспользоватьВСписках")
И СтруктураНастроек.ИспользоватьВСписках Тогда
ЕстьУсловиеОтбора = Истина;
КонецЕсли;
КонецЦикла;
Если ЕстьУсловиеОтбора Тогда
НоваяГруппаОтбора =
НоваяГруппаОтбораОбщая.Элементы.Добавить(Тип("ГруппаЭлементовОтбораКомпоновкиДанных"));

НоваяГруппаОтбора.ТипГруппы =
ТипГруппыЭлементовОтбораКомпоновкиДанных.ГруппаИ;
ДобавитьУсловиеНаТип(Тип(ИмяТипаДанных),НоваяГруппаОтбора.Элементы);
Для каждого КлючНастройка Из СтруктураВсехНастроек Цикл
СтруктураНастроек = КлючНастройка.Значение;
НастройкиКомпоновщика="";
Если
СтруктураНастроек.Свойство("НастройкиКомпоновщика",НастройкиКомпоновщика) Тогда
Если СтруктураНастроек.Свойство("ИспользоватьВСписках")
И СтруктураНастроек.ИспользоватьВСписках Тогда

СкопироватьУсловияОтбора(НастройкиКомпоновщика.Отбор.Элементы,НоваяГруппаОтбора.Элементы);
КонецЕсли;
КонецЕсли;
КонецЦикла;
Иначе
ДобавитьУсловиеНаТип(Тип(ИмяТипаДанных),НоваяГруппаОтбораОбщая.Элементы);
КонецЕсли;

Для Каждого Реквизит Из
ОбъектМетаданных.СтандартныеРеквизиты Цикл
Если Не ПустаяСтрока(ТекстЗапроса) Тогда
ТекстЗапроса = ТекстЗапроса + "," + Символы.ПС;
КонецЕсли;
ТекстЗапроса = ТекстЗапроса + Реквизит.Имя;
КонецЦикла;
Для Каждого ТабличнаяЧасть Из
ОбъектМетаданных.ТабличныеЧасти Цикл
ИмяТабличнойЧасти = ТабличнаяЧасть.Имя;
ТекстЗапроса = ТекстЗапроса + "," + Символы.ПС +
ИмяТабличнойЧасти + " (";

КонецЦикла;
КонецПроцедуры

Процедура
ДобавитьУсловиеНаТип(ТипДанных,ОтборСписокЭлементы)
ЭлементОтбора =
ОтборСписокЭлементы.Добавить(Тип("ЭлементОтбораКомпоновкиДанных"));
ЭлементОтбора.ЛевоеЗначение = Новый
ПолеКомпоновкиДанных("Тип");
ЭлементОтбора.ВидСравнения =
ВидСравненияКомпоновкиДанных.Равно;
ЭлементОтбора.ПравоеЗначение = ТипДанных;
КонецПроцедуры

Процедура
СкопироватьУсловияОтбора(ОтборИсточникЭлементы,ОтборСписокЭлементы)
Для каждого УсловиеОтбора Из ОтборИсточникЭлементы Цикл
Если ТипЗнч(УсловиеОтбора) =
Тип("ГруппаЭлементовОтбораКомпоновкиДанных") Тогда
НоваяГруппа =
ОтборСписокЭлементы.Добавить(Тип("ГруппаЭлементовОтбораКомпоновкиДанных"));
НоваяГруппа.ТипГруппы = УсловиеОтбора.ТипГруппы;

НоваяГруппа.Представление = УсловиеОтбора.Представление;
СкопироватьУсловияОтбора(УсловиеОтбора.Элементы,НоваяГруппа.Элементы);
Иначе
ЭлементОтбора =
ОтборСписокЭлементы.Добавить(Тип("ЭлементОтбораКомпоновкиДанных"));
ПутьКПолюКомпоновки = ?
(УсловиеОтбора.ЛевоеЗначение="Ссылка","Ссылка","Ссылка."+
УсловиеОтбора.ЛевоеЗначение);
ЭлементОтбора.ЛевоеЗначение = Новый
ПолеКомпоновкиДанных(ПутьКПолюКомпоновки);
ЭлементОтбора.ВидСравнения = УсловиеОтбора.ВидСравнения;
ЭлементОтбора.ПравоеЗначение =
УсловиеОтбора.ПравоеЗначение;
ЭлементОтбора.Представление =
УсловиеОтбора.Представление;
КонецЕсли;
КонецЦикла;
КонецПроцедуры
#КонецОбласти

#Если Сервер Или ТолстыйКлиентОбычноеПриложение Или
ВнешнееСоединение Тогда
#Область СлужебныеПроцедурыИФункции
// Для внутреннего использования.
Функция ТекстЗапроса(ИмяОбъекта) Экспорт
ОбъектМетаданных =
Метаданные.НайтиПоПолномуИмени(ИмяОбъекта);
ТекстЗапроса = "";
Для Каждого Реквизит Из ОбъектМетаданных.Реквизиты Цикл
Если Не ПустаяСтрока(ТекстЗапроса) Тогда
ТекстЗапроса = ТекстЗапроса + "," + Символы.ПС;
КонецЕсли;
ТекстЗапроса = ТекстЗапроса + Реквизит.Имя;
КонецЦикла;

П2.3 Продолжение

СтрокаРеквизитов = "НомерСтроки";
РеквизитыТабличнойЧасти = ТабличнаяЧасть.Реквизиты;
Для Каждого Реквизит Из РеквизитыТабличнойЧасти Цикл
Если Не ПустаяСтрока(СтрокаРеквизитов) Тогда
СтрокаРеквизитов = СтрокаРеквизитов + "," + Символы.ПС;
КонецЕсли;

СтрокаРеквизитов = СтрокаРеквизитов + Реквизит.Имя;
КонецЦикла;
ТекстЗапроса = ТекстЗапроса + СтрокаРеквизитов + ")";
КонецЦикла;
ТекстЗапроса = "ВЫБРАТЬ " + ТекстЗапроса + Символы.ПС + "

```

|ИЗ
|      "+ ОбъектМетаданных.ПолноеИмя());

Возврат ТекстЗапроса;
КонецФункции

// Для внутреннего использования.
Функция СхемаКомпоновкиДанных(ТекстЗапроса) Экспорт
СхемаКомпоновкиДанных = Новый СхемаКомпоновкиДанных;
ИсточникДанных =
СхемаКомпоновкиДанных.ИсточникиДанных.Добавить();
ИсточникДанных.Имя = "ИсточникДанных1";
ИсточникДанных.ТипИсточникаДанных = "local";
НаборДанных =
СхемаКомпоновкиДанных.НаборыДанных.Добавить(Тип("Набо
рДанныхЗапросСхемыКомпоновкиДанных"));
НаборДанных.ИсточникДанных = "ИсточникДанных1";
НаборДанных.АвтоЗаполнениеДоступныхПолей = Истина;
НаборДанных.Запрос = ТекстЗапроса;
НаборДанных.Имя = "НаборДанных1";
Возврат СхемаКомпоновкиДанных;
КонецФункции

////////////////////////////////////
// Процедуры и функции из базовой функциональности для
обеспечения автономности
// Возвращает Истина, если подсистема существует.
//
// Параметры:
// ПолноеИмяПодсистемы - Строка. Полное имя объекта
метаданных подсистема без слов "Подсистема.".
//      Например:
"СтандартныеПодсистемы.БазоваяФункциональность".
//
// Пример вызова необязательной подсистемы:
//
// Если
ОбщегоНазначения.ПодсистемаСуществует("СтандартныеПодси
стемы.УправлениеДоступом") Тогда
//      МодульУправлениеДоступом =
ОбщегоНазначения.ОбщийМодуль("УправлениеДоступом");
//      МодульУправлениеДоступом.<Имя метода>();
//      КонецЕсли;
//
// Возвращаемое значение:
// Булево.
//
Функция ПодсистемаСуществует(ПолноеИмяПодсистемы)
Экспорт
ИменаПодсистем = ИменаПодсистем();
Возврат ИменаПодсистем.Получить(ПолноеИмяПодсистемы) <>
Неопределено;
КонецФункции
// Возвращает ссылку на общий модуль по имени.

```

```

//
// Параметры:
// Имя - Строка - имя общего модуля, например:
//      "ОбщегоНазначения",
//      "ОбщегоНазначенияКлиент".
//
// Возвращаемое значение:
// ОбщийМодуль.
//
Функция ОбщийМодуль(Имя) Экспорт
Если Метаданные.ОбщиеМодули.Найти(Имя) <> Неопределено
Тогда
Модуль = Вычислить(Имя);
Иначе
Модуль = Неопределено;
КонецЕсли;
Если ТипЗнч(Модуль) <> Тип("ОбщийМодуль") Тогда
ВызватьИсключение СтрШаблон(НСтр("ru = 'Общий модуль
'%"1'" не найден."), Имя);
КонецЕсли;
Возврат Модуль;
КонецФункции

// Возвращает соответствие имен подсистем и значения Истина;
Функция ИменаПодсистем() Экспорт
Возврат Новый
ФиксированноеСоответствие(ИменаПодчиненныхПодсистем(Ме
таданные));
КонецФункции

Функция
ИменаПодчиненныхПодсистем(РодительскаяПодсистема)
Имена = Новый Соответствие;
Для Каждого ТекущаяПодсистема Из
РодительскаяПодсистема.Подсистемы Цикл
Имена.Вставить(ТекущаяПодсистема.Имя, Истина);

ИменаПодчиненных =
ИменаПодчиненныхПодсистем(ТекущаяПодсистема);
Для каждого ИмяПодчиненной Из ИменаПодчиненных Цикл
Имена.Вставить(ТекущаяПодсистема.Имя + "." +
ИмяПодчиненной.Ключ, Истина);
КонецЦикла;
КонецЦикла;
Возврат Имена;
КонецФункции

// Возвращает полное имя неуказанного пользователя для
отображения в интерфейсах.
Функция ПолноеИмяНеуказанногоПользователя() Экспорт
Возврат НСтр("ru = '<Не указан>');
КонецФункции
#КонецОбласти
#КонецЕсли

```

Спецификация

Спецификация на программное изделие представлена в табл. ПЗ.1.

Таблица ПЗ.1

Спецификация

Обозначение	Наименование	Примечание
Документация		
РПЗ_Соколова.docx	Разработка информационной системы контроля и анализа первичных документов на технологической платформе 1С8	Расчетно-пояснительная записка с приложениями в виде: технического задания, текста программы и руководства пользователя
Компоненты		
Monitoring_and_analysis.cfe	Файл конфигурации (расширения) «Контроль и анализ первичных документов»	
Test.cd	Файл информационной базы для тестовой работы	

Руководство пользователя

- Общие сведения о программе

Информационная система предназначена для предприятия ОАО «Северсталь-инфоком».

Данная ИС разрабатывалась, как расширение конфигурации, поэтому для ее работы необходимо подключить ее к ИБ.

- Установка расширения

Установка расширения выполняется из меню «Все функции» - «Стандартные» - «Управление расширениями конфигурации» (рис. П4.1)

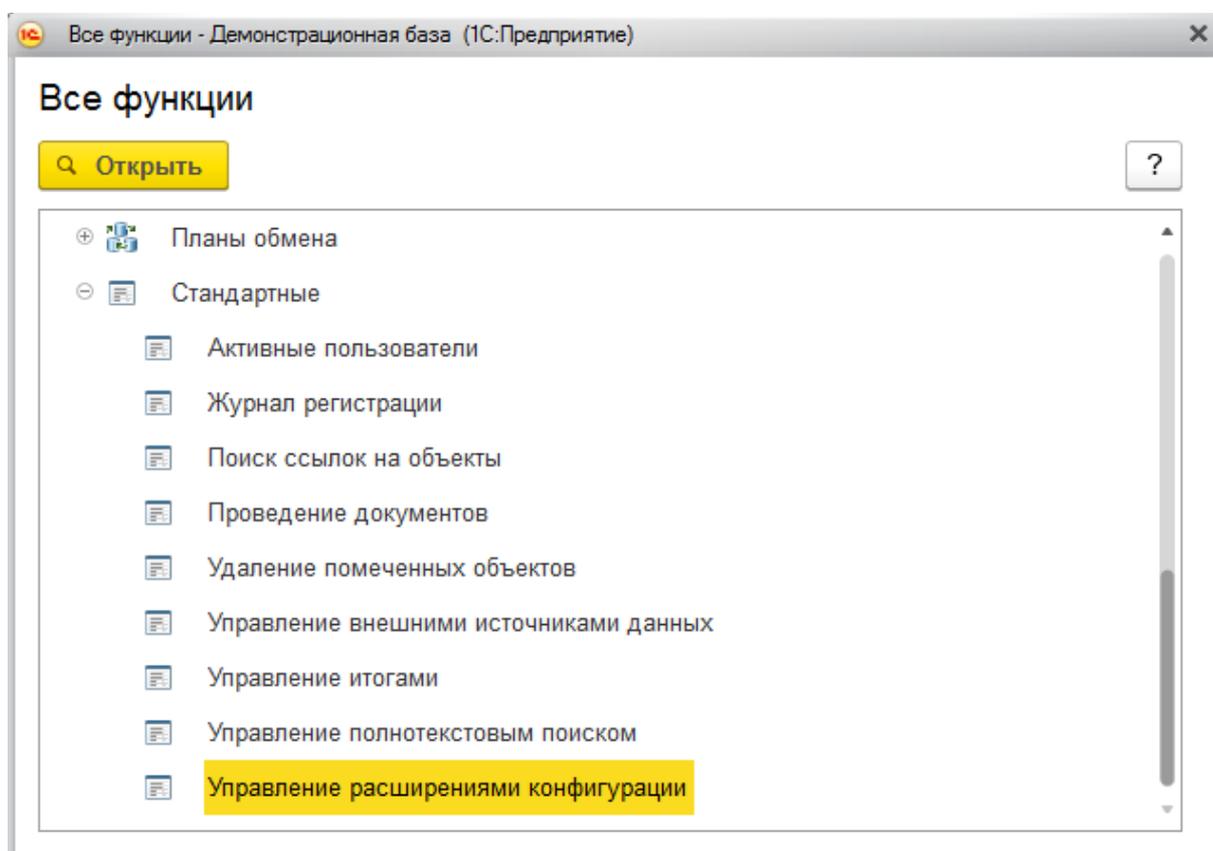


Рис. П4.1. Установка расширения

После нажатия кнопки «Добавить» необходимо выбрать на диске файл с расширением (рис. П4.2).

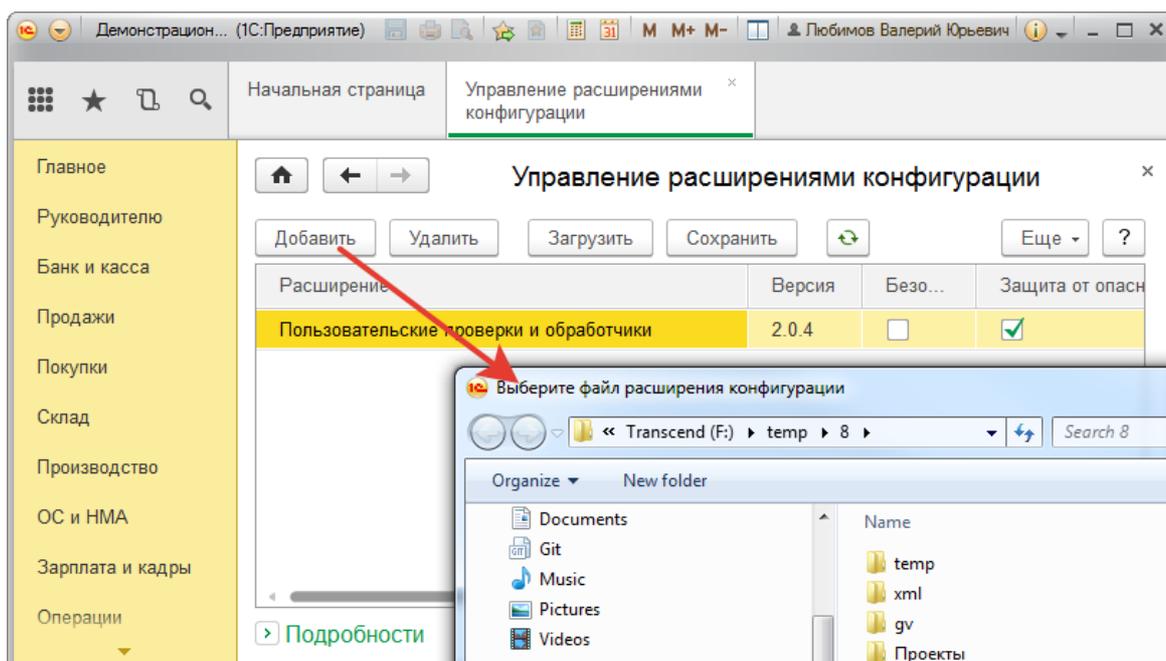


Рис. П4.2. Выбор файла расширения

После установки расширения и перезапуска конфигурации появится пункт меню «Администрирование» - «Сервис» - «Пользовательские проверки и обработчики» (рис. П4.3).

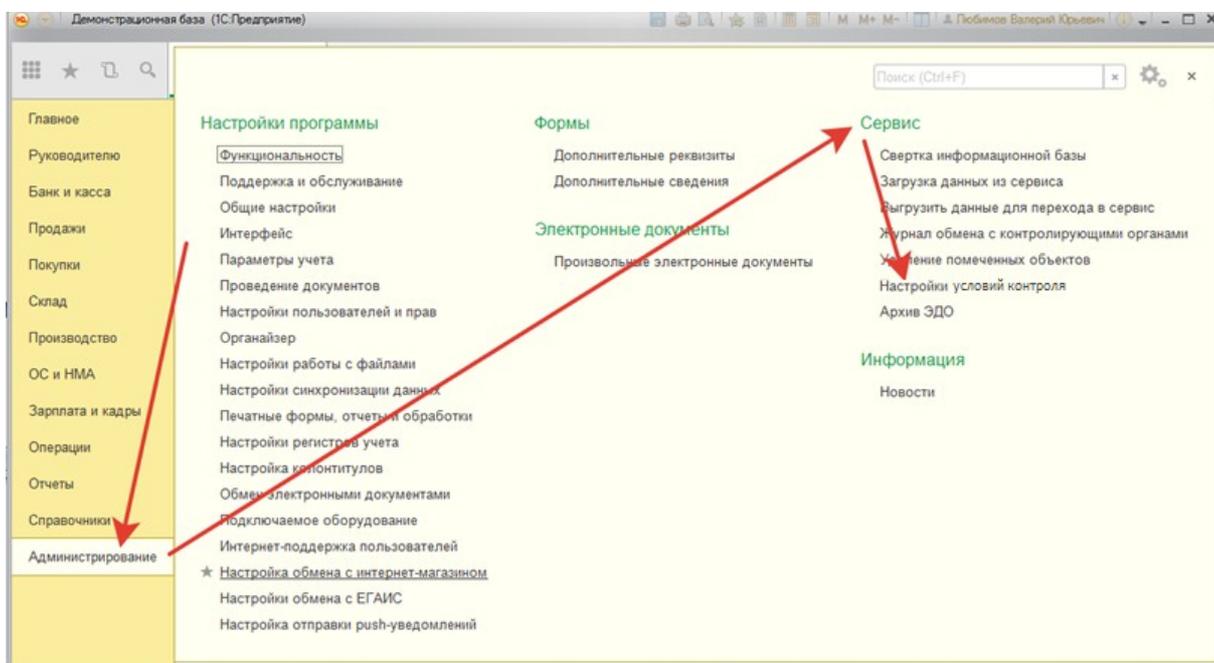


Рис. П4.3. Открытие «Настройки условий контроля»

- Настройка условий контроля

Настройка условий контроля при вводе данных и контроля доступа выполняется из пункта меню «Администрирование» - «Сервис» - «Настройка условий контроля».

При выборе данного пункта меню открывается окно настройки дополнительных проверок со списком всех документов и справочников конфигурации (рис. П4.4).

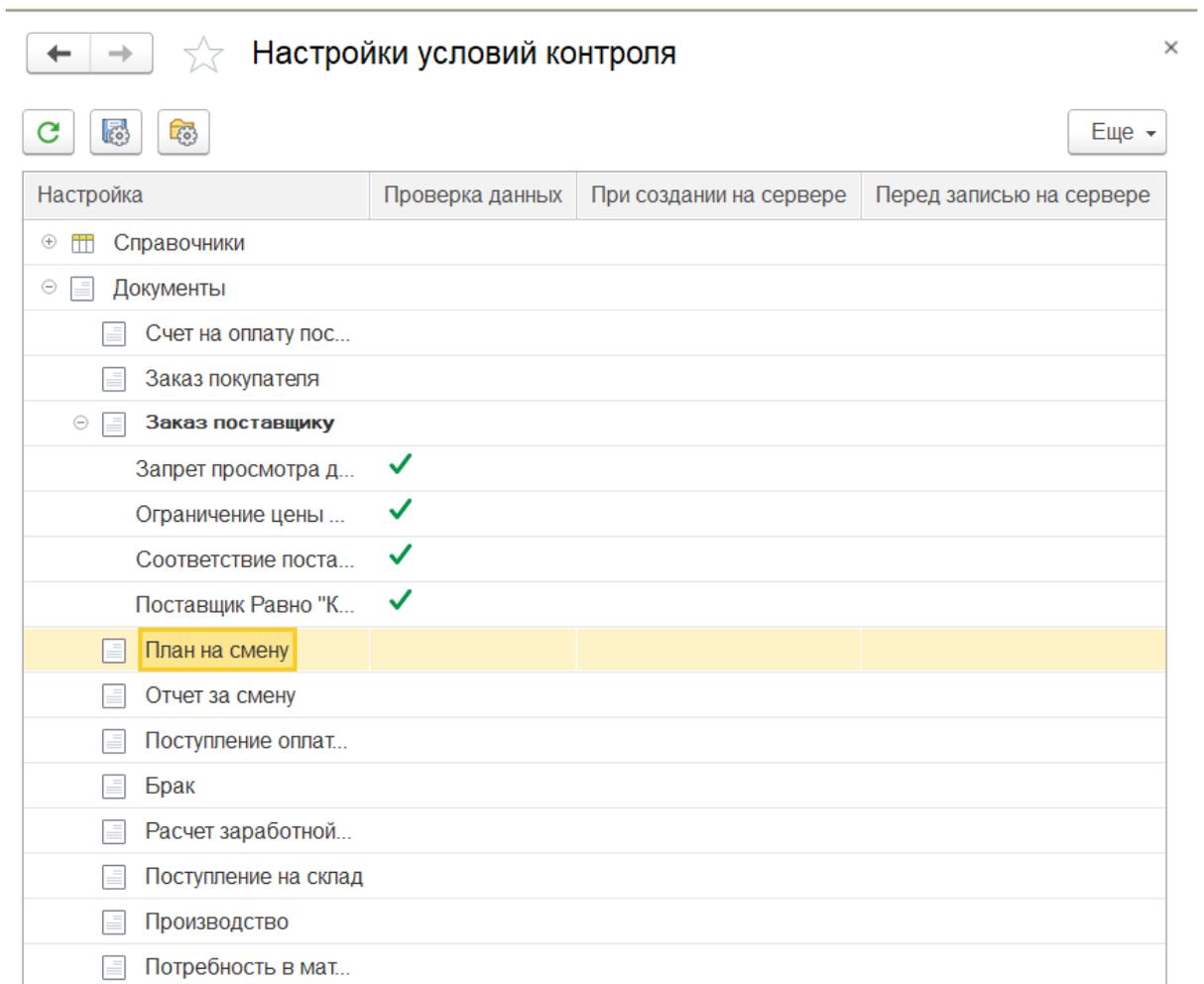


Рис. П4.4. Отображение всех документов для контроля

При выборе конкретного документа или справочника открывается форма настройки.

На основной закладке указывается условие проверки (рис. П4.5), которому должны соответствовать вводимые пользователем данные.

Также указывается режим использования ограничения:

Запрещать запись - запрещать запись элементов, не соответствующих условию;

Запрещать изменение - запрещать изменение элементов, не соответствующих условию;

Запрещать просмотр - запрещать просмотр элементов, не соответствующих условию;

Скрывать в списках - скрывать в списках элементы, не соответствующие условию.

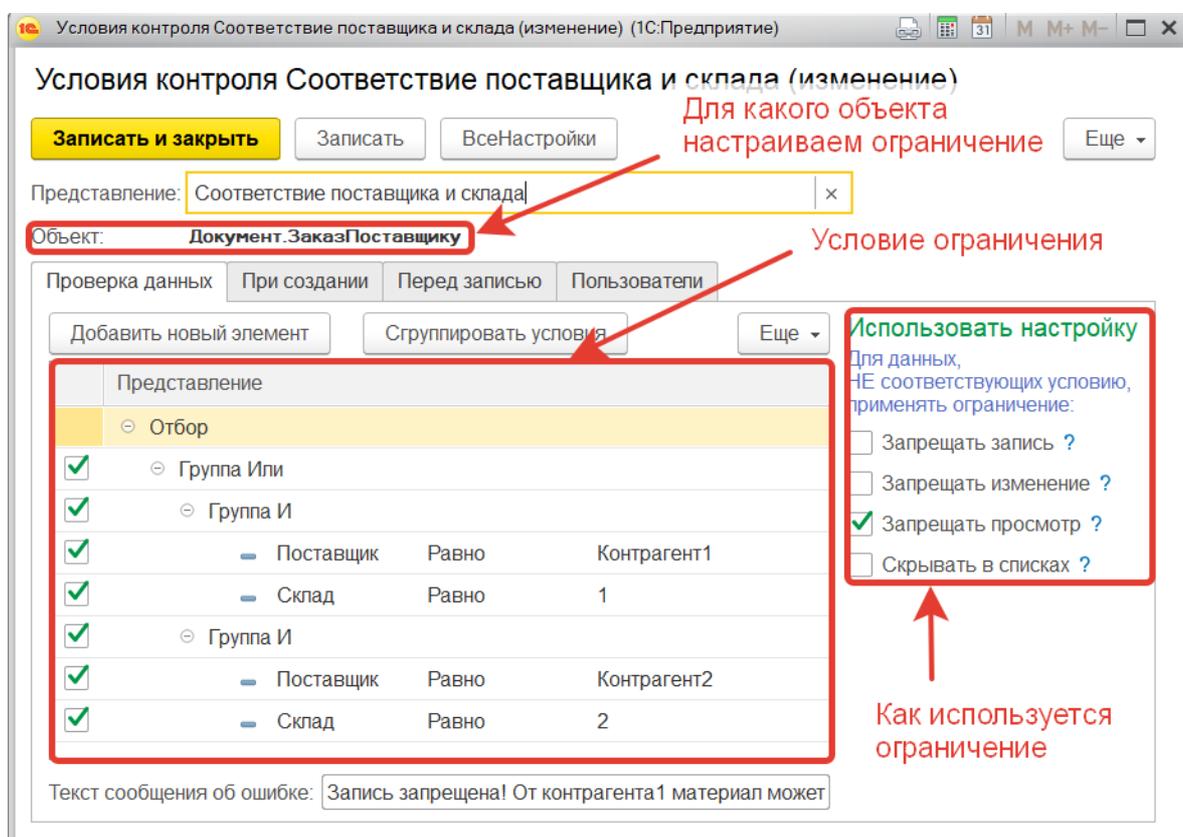


Рис. П4.5. Установление условий контроля

На закладке пользователи (рис. П4.6) указан список пользователей, для которых действует введенное ограничение.

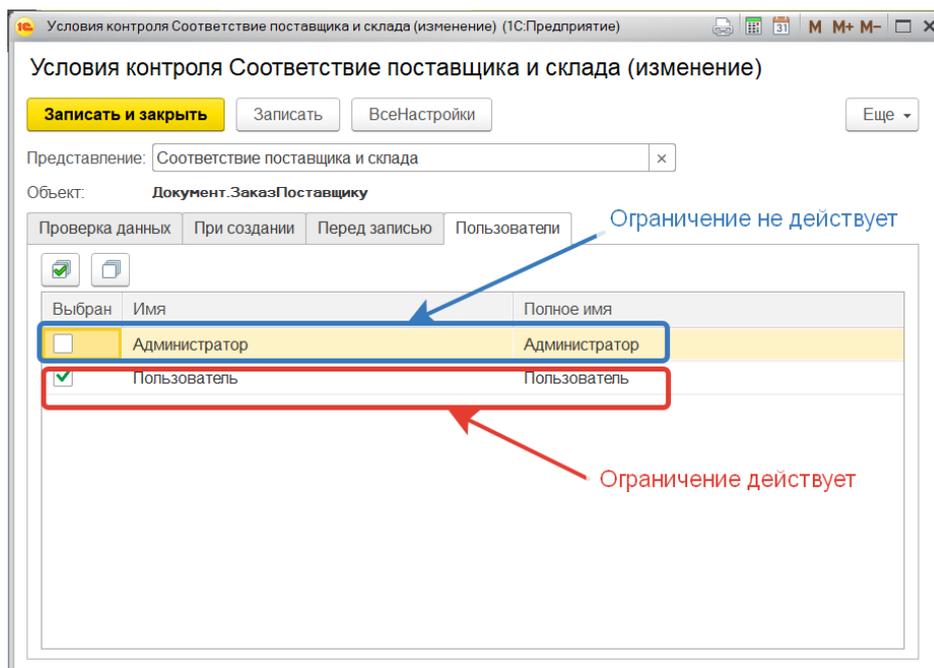


Рис. П4.6. Выбор пользователей

1) Ограничение «Запрет просмотра»

Рассмотрим настройку запрета пользователю просмотра документов реализации всем складам, кроме склада 1 (рис. П4.7).

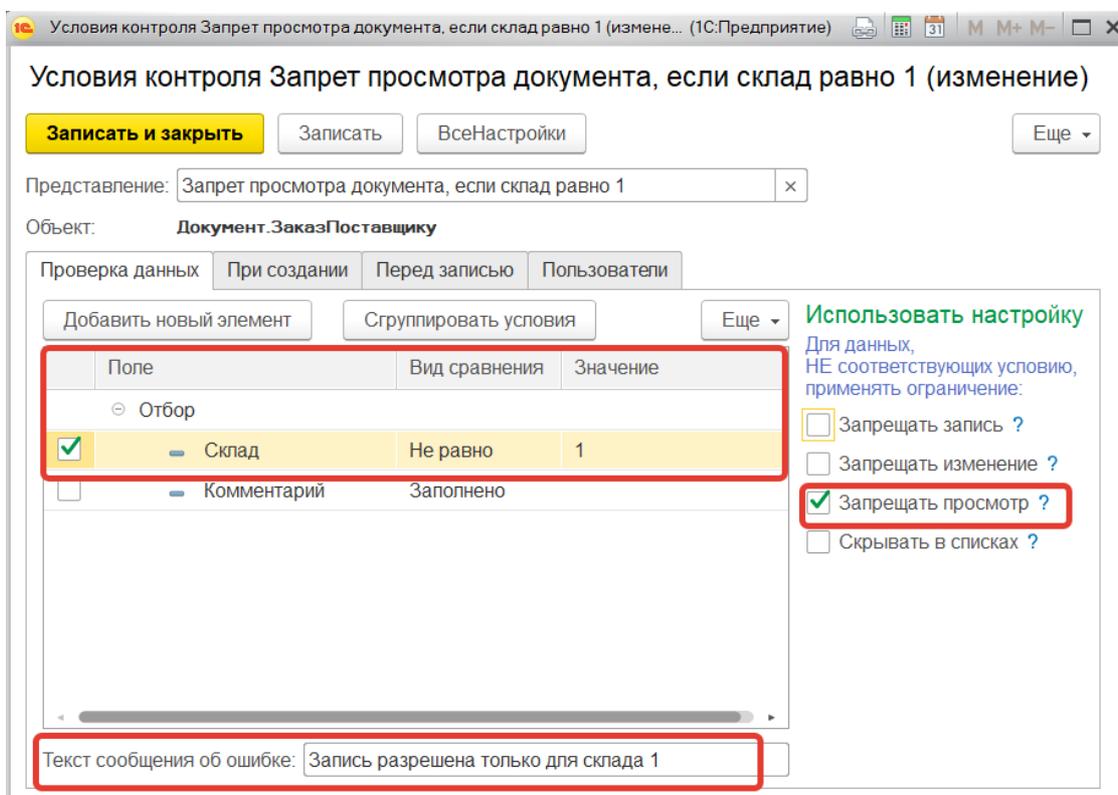


Рис. П4.7. Запрет просмотра

При попытке просмотра документа отгрузки другого склада, пользователю выдаться сообщение о том, что просмотр таких документов ему запрещен (рис. П4.8).

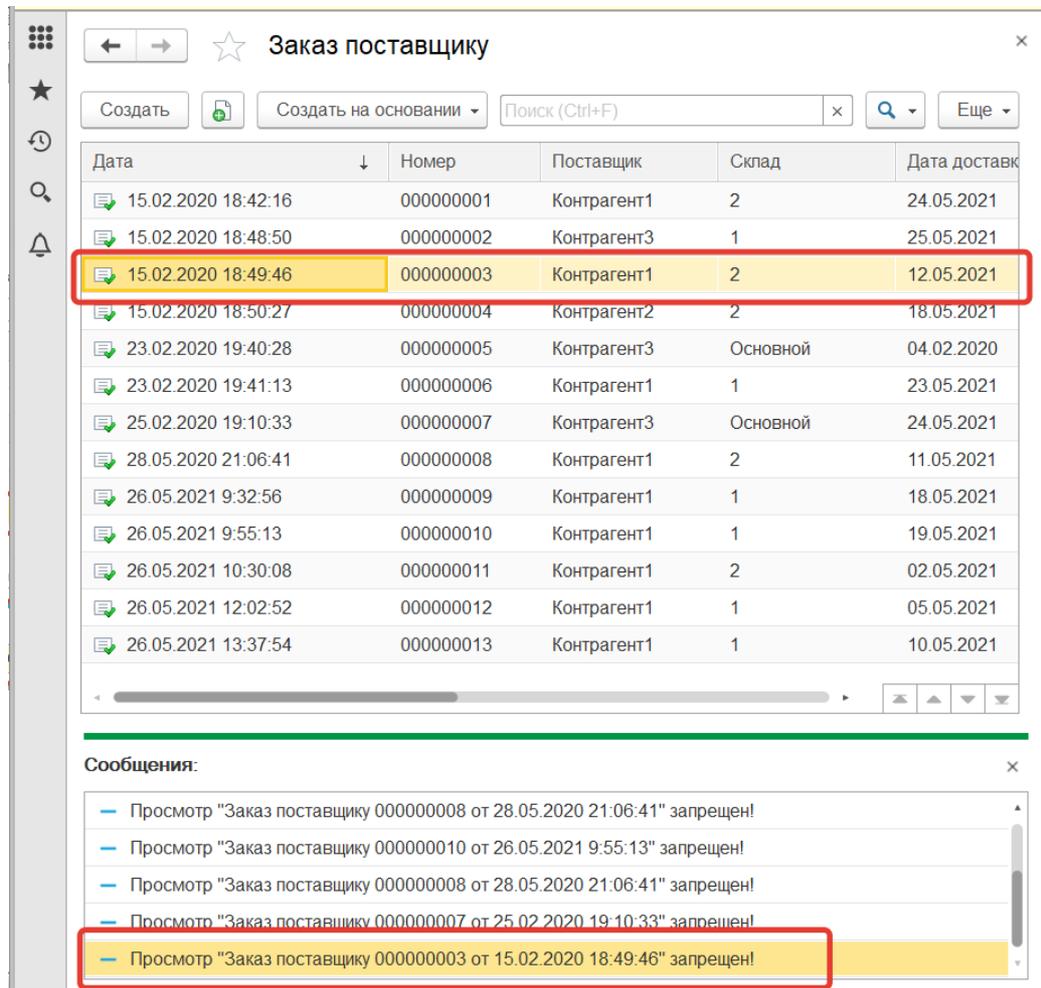


Рис. П4.8. Сообщение о запрете просмотра

2) Ограничение «При создании»

На закладке «При создании» (рис. П4.9) в произвольной форме можно указать любые действия, которые необходимо выполнить при создании формы. Это могут быть:

1. Проверка доступа к данным;
2. Начальное заполнение значениями по умолчанию для новых документов;

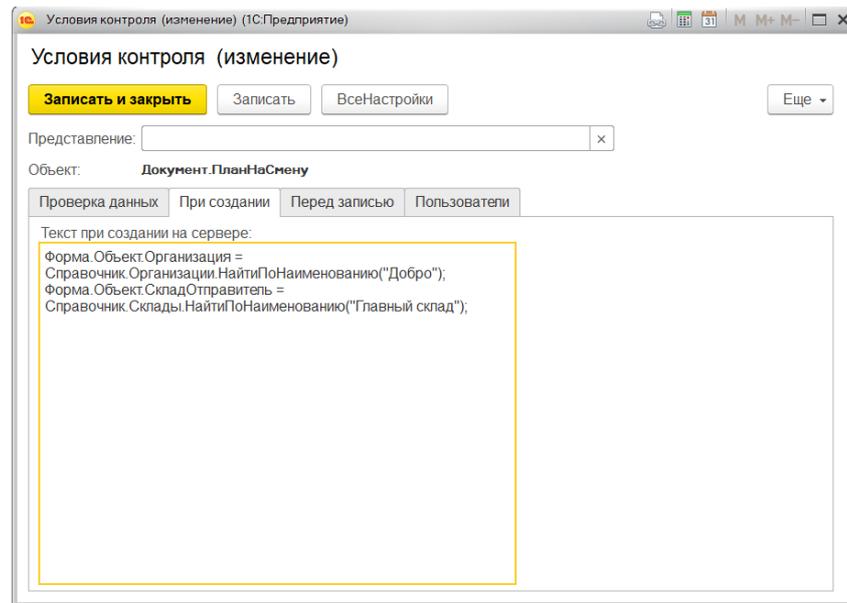


Рис. П4.9. Дополнительные настройки при создании

3) Ограничение «Перед записью»

На закладке «Перед записью» (рис. П4.10) в произвольной форме можно указать любые действия, которые необходимо выполнить перед записью объекта.

Это могут быть:

1. Проверка корректности данных;
2. Дозаполнение данных по связанным полям;

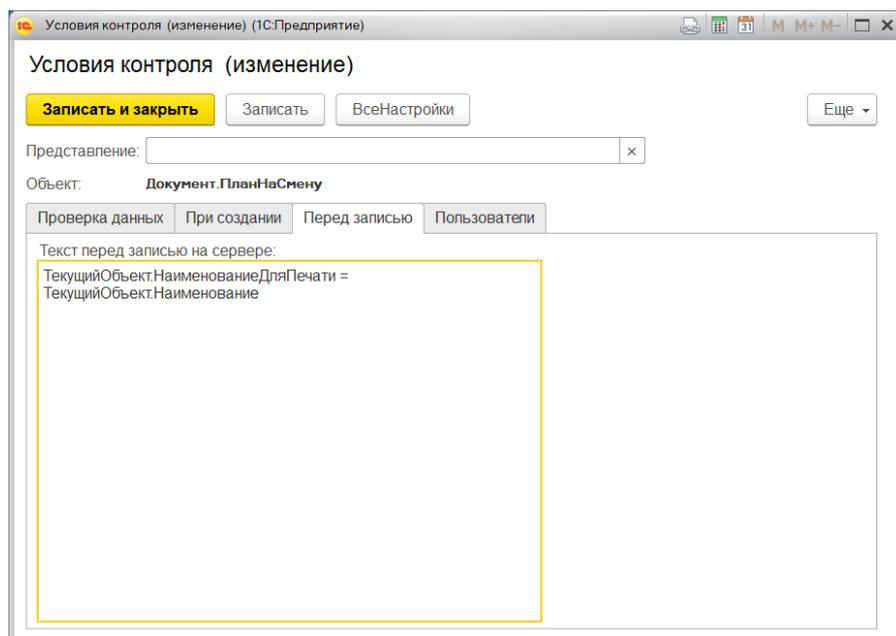


Рис. П4.10. Дополнительные настройки перед записью