

НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ФИНАНСОВО-
ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ «СИНЕРГИЯ»

Факультет электронного обучения

Направление

09.03.03

(код)

Кафедра

ЭО

(аббревиатура)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

на тему «АВТОМАТИЗАЦИЯ ПРИЕМА И ОБРАБОТКИ ЗАЯВОК ОТДЕЛОМ
ТЕХПОДДЕРЖКИ ООО "КОМПЬЮТЕРНЫЕ
БИЗНЕС СИСТЕМЫ"»

Обучающийся **Моховиков Семен Николаевич**

(Ф.И.О. полностью)

подпись

Руководитель **Петухова Мария Владиславовна**

(Ф.И.О. полностью)

подпись

Москва 2019

Содержание	
Введение	4
1 Аналитическая часть	6
1.1 Техничко-экономическая характеристика предметной области и предприятия. Анализ деятельности «КАК ЕСТЬ»	6
1.1.1 Характеристика предприятия и его деятельности	6
1.1.2 Организационная структура управления предприятием	8
1.1.3 Программная и техническая архитектура ИС предприятия	12
1.2 Характеристика комплекса задач, задачи и обоснование необходимости автоматизации	15
1.2.1 Выбор комплекса задач автоматизации и характеристика существующих бизнес процессов	15
1.2.2 Определение места проектируемой задачи в комплексе задач и ее описание	18
1.2.3 Обоснование необходимости использования вычислительной техники для решения задачи	22
1.2.4 Анализ системы обеспечения информационной безопасности и защиты информации	29
1.3 Анализ существующих разработок и выбор стратегии автоматизации «КАК ДОЛЖНО БЫТЬ»	31
1.3.1 Анализ существующих разработок для автоматизации задачи ..	31
1.3.2 Выбор и обоснование стратегии автоматизации задачи	39
1.3.3 Выбор и обоснование способа приобретения ИС для автоматизации задачи	40
1.4 Обоснование проектных решений	41
1.4.1 Обоснование проектных решений по информационному обеспечению	41
1.4.2 Обоснование проектных решений по программному обеспечению	41
1.4.3 Обоснование проектных решений по техническому обеспечению	44
2 Проектная часть	47

2.1	Разработка проекта автоматизации	47
2.1.1	Этапы жизненного цикла проекта автоматизации	47
2.1.2	Ожидаемые риски на этапах жизненного цикла и их описание .	50
2.1.3	Организационно-правовые и программно-аппаратные средства обеспечения информационной безопасности и защиты информации	51
2.2	Информационное обеспечение задачи	53
2.2.1	Информационная модель и её описание.....	53
2.2.2	Характеристика нормативно-справочной, входной и оперативной информации	55
2.2.3	Характеристика результатной информации.....	57
2.3	Программное обеспечение задачи	62
2.3.1	Общие положения (дерево функций и сценарий диалога)	62
2.3.2	Характеристика базы данных	71
2.3.3	Структурная схема пакета (дерево вызова программных модулей)	75
2.3.4	Описание программных модулей.....	87
2.4	Контрольный пример реализации проекта и его описание.....	90
3	Обоснование экономической эффективности проекта.....	95
3.1	Выбор и обоснование методики расчёта экономической эффективности.....	95
3.2	Расчёт показателей экономической эффективности проекта	100
	Заключение.....	108
	Список использованных источников.....	110
	Приложение 1. Текст программных модулей.....	113

Введение

Целью данного проекта является разработка и внедрение автоматизированной информационной системы приема и обработки заявок, поступивших в службу технической поддержки эксплуатации информационных ресурсов и ИТ-систем предприятия ООО «Компьютерные Бизнес Системы» от пользователей ИТ-инфраструктуры предприятия

Необходимость проекта обусловлена тем, что в текущих экономических условиях эффективная деятельность любого коммерческого предприятия возможна при условии автоматизации всех сторон деятельности предприятия. Особенно это актуально для такой области как обеспечение бесперебойной работы всех информационных систем крупного холдинга.

Среди всех процессов в работе службы технической поддержки выделяется процесс приема и обработки поступивших в службу заявок, касающихся решения проблем, возникших у сотрудников предприятия в ходе эксплуатации ИТ-ресурсов. Успешное решение этого процесса является залогом успешной автоматизации всего комплекса задач предприятия.

Из анализа деятельности службы технической поддержки выявляется много недостатков построения существующей системы организации учета и контроля выполнения поставленных перед службой задач. Прежде всего, это касается того, что хранилища данных представляют собой либо бумажные хранилища, либо разрозненные файлы, поиск информации в которых может быть проведен только вручную, что является долгим и трудоемким процессом. Также неудобно построено составление отчетов, подготовка реестров документов.

Анализ поставленной задачи приводит к выводу, что ее решение традиционными средствами учета с использованием бумажных документов и разрозненных файлов невозможно. Следовательно, стратегия решения задачи – это автоматизация всех процессов, относящихся к задаче постановки задач исполнителям, а также контроль их выполнения.

Объект исследования – служба технической поддержки отдела ИТ ООО «Компьютерные Бизнес Системы»

Предмет исследования – предметная область работы службы технической поддержки в части прохождения всех этапов приема и дальнейшего сопровождения

поступающих заявок.

Задачи исследования:

- Исследование существующего процесса приема и прохождения заявки пользователя для устранения проблемы;
- Выявление основных недостатков существующей системы приема и сопровождения заявок;
- Обоснование необходимости разработки информационной системы, автоматизирующей процесс приема и сопровождение заявки;
- Формализация задачи, постановка задачи на разработку программного продукта;
- Разработка программного обеспечения решения задачи;
- Тестирование разработанного программного продукта, устранение ошибок и недостатков;
- Оценка экономической эффективности проекта.

Результатом работы будет являться готовая система, которую можно внедрить в службу технической поддержки для автоматизации процесса приема и сопровождения поступающих заявок.

Теоретико-методической основой выполнения ВКР являются современные подходы и разработки в области создания автоматизированных информационных систем. Эти разработки были взяты за основу данного проекта.

1 Аналитическая часть

1.1 Техничко-экономическая характеристика предметной области и предприятия. Анализ деятельности «КАК ЕСТЬ»

1.1.1 Характеристика предприятия и его деятельности

Разрабатываемая автоматизированная информационная система (АИС) создается для службы технической поддержки (далее Службы) ООО «Компьютерные Бизнес Системы», работающего в сфере информационных технологий. Основным результатом деятельности такого предприятия является разработка информационных систем, внедрение разработанных систем, выполнение каких-либо услуг по поддержанию работоспособности внедренных систем, сопровождение систем.

ООО «Компьютерные Бизнес Системы» предоставляет своим клиентам широкий спектр услуг в области информационных технологий:

- Системная интеграция;
- Разработка автоматизированных информационных систем
- Продажа программного обеспечения
- Поставка оргтехники и расходных материалов;
- Сервисное обслуживание компьютерной и оргтехники;
- Разработка сетевых проектов и их внедрение на территории клиентов;
- Продажа и внедрение комплексных систем безопасности.

В компании работают квалифицированные специалисты, обладающие большим опытом и многолетней практикой разработки проектов в области ИТ-технологий. Отработанное за много лет сотрудничества командное взаимодействие, понимание каждым сотрудником своей роли в проекте, профессионализм и нацеленность на единый конечный результат, позволяют компании держать высокую планку качества предоставляемых клиентам услуг.

Внутреннюю организацию работы современного предприятия сложно представить без использования корпоративной информационной системы (КИС), автоматизирующей все сферы деятельности ООО «Компьютерные Бизнес Системы». В качестве такой системы в ООО «Компьютерные Бизнес Системы» используется SAP R/3.

Система управления предприятием SAP R/3 – это система класса ERP, которая охватывает все участки финансового и управленческого учета, управления персоналом, оперативной деятельности и сервисных служб компании. Она обеспечивает полную функциональность, необходимую для реализации информационных сервисов самообслуживания, аналитики.

Отличительной особенностью такой КИС является сложность как самой системы, так и среды, в которой она функционирует. Система включает в себя большое количество разнообразных модулей самого разного направления и на сегодняшний день обеспечивает работу почти трехсот пользователей.

Эксплуатация такой большой и сложной КИС невозможна без эффективной работы службы технической поддержки пользователей, являющейся подразделением отдела ИТ компании. Регулярно у пользователей КИС возникает множество проблем, начиная от подключения к сети и поломки периферийных устройств до ошибок в логике работы какого-то из модулей КИС.

В Таблице 1.1 приведены сведения об эффективности деятельности предприятия в динамике последних трех лет.

Таблица 1.1. Показатели экономической эффективности деятельности

Показатели	2016г.	2017 г.	2018 г.
Выручка от реализации, тыс. руб.	2400	2500	2700
Себестоимость продукции, тыс. руб.	1605	1680	1895
Валовой доход, тыс. руб.	795	820	805
Управленческие расходы, тыс. руб.	50	52	55
Коммерческие расходы, тыс. руб.	30	35	40
Заработная плата сотрудников	225	240	255
Прибыль/убыток, тыс. руб.	490	493	455
Внереализационные доходы, тыс. руб.	0	0	0
Внереализационные расходы, тыс. руб.	10	12	15
Прибыль до налогообложения, тыс. руб.	480	481	440
Чистая прибыль, тыс. руб.	307,2	307,84	281,60
Рентабельность продаж, %	12,8	12,31	10,42

Из анализа таблицы видно, что объем валовой реализации ежегодно увеличивается. Однако одновременно растет себестоимость выпускаемой

продукции, управленческие, коммерческие и внереализационные расходы. Незначительными темпами, но растет заработная плата сотрудников.

Однако, не смотря на тенденцию к уменьшению, рентабельность остается на достаточно высоком уровне, прибыль остается положительной. В 2019 году руководство предприятия планирует провести ряд организационных мероприятий, направленных на улучшение ситуации.

1.1.2 Организационная структура управления предприятием

Сферы деятельности ООО «Компьютерные Бизнес Системы» определяют его структуру, представленную на Рисунке 1.1.

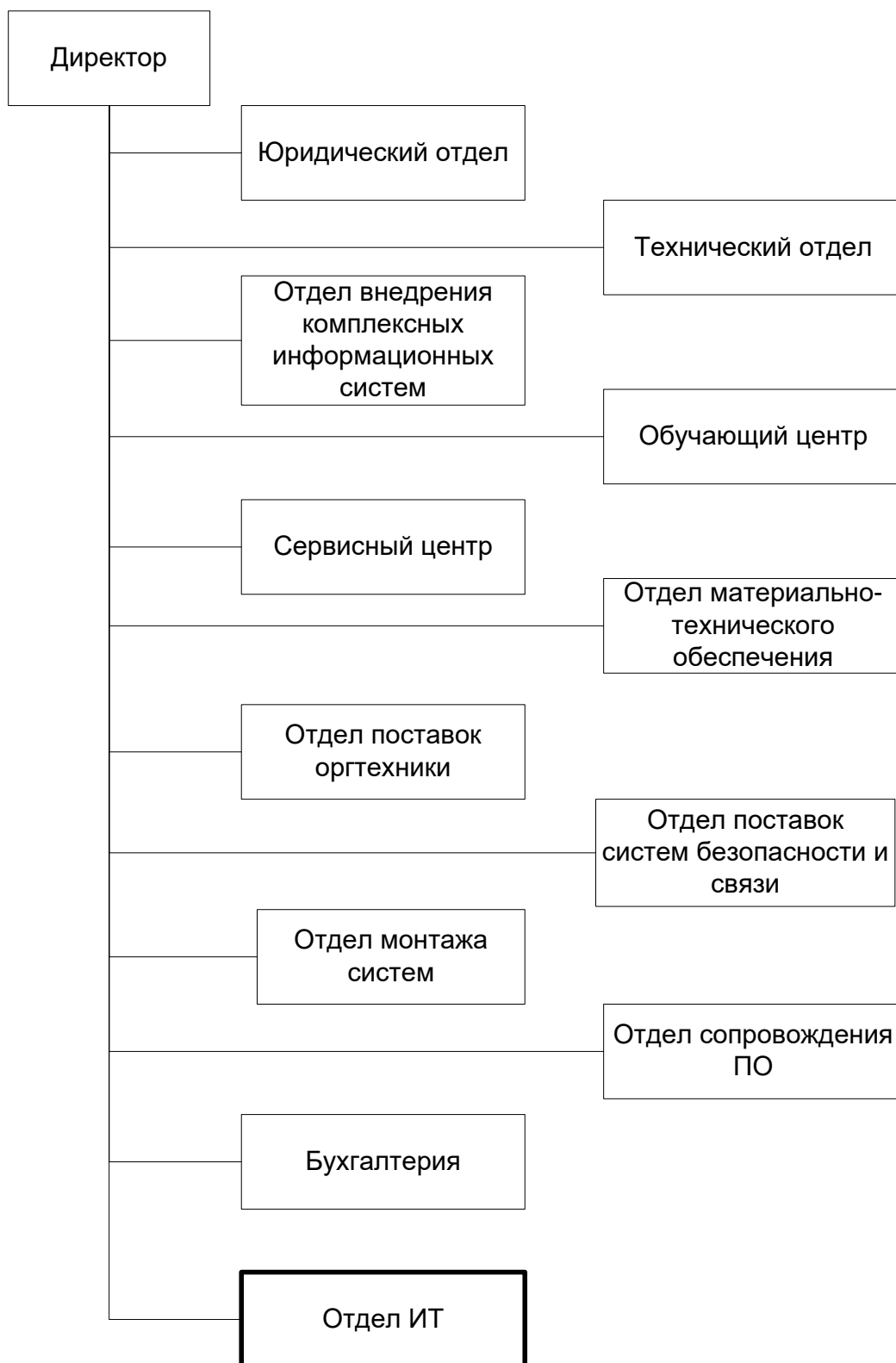


Рисунок 1.1. Структура предприятия

В работе рассматривается работа отдела ИТ и его функции. Бесперебойная и эффективная работа КИС, обеспечивающая работу всех остальных подразделений компании, является основной обязанностью и главным направлением деятельности

отдела ИТ, одним из подразделений которого является служба технической поддержки. Структура отдела ИТ приведена на Рисунке 1.2.

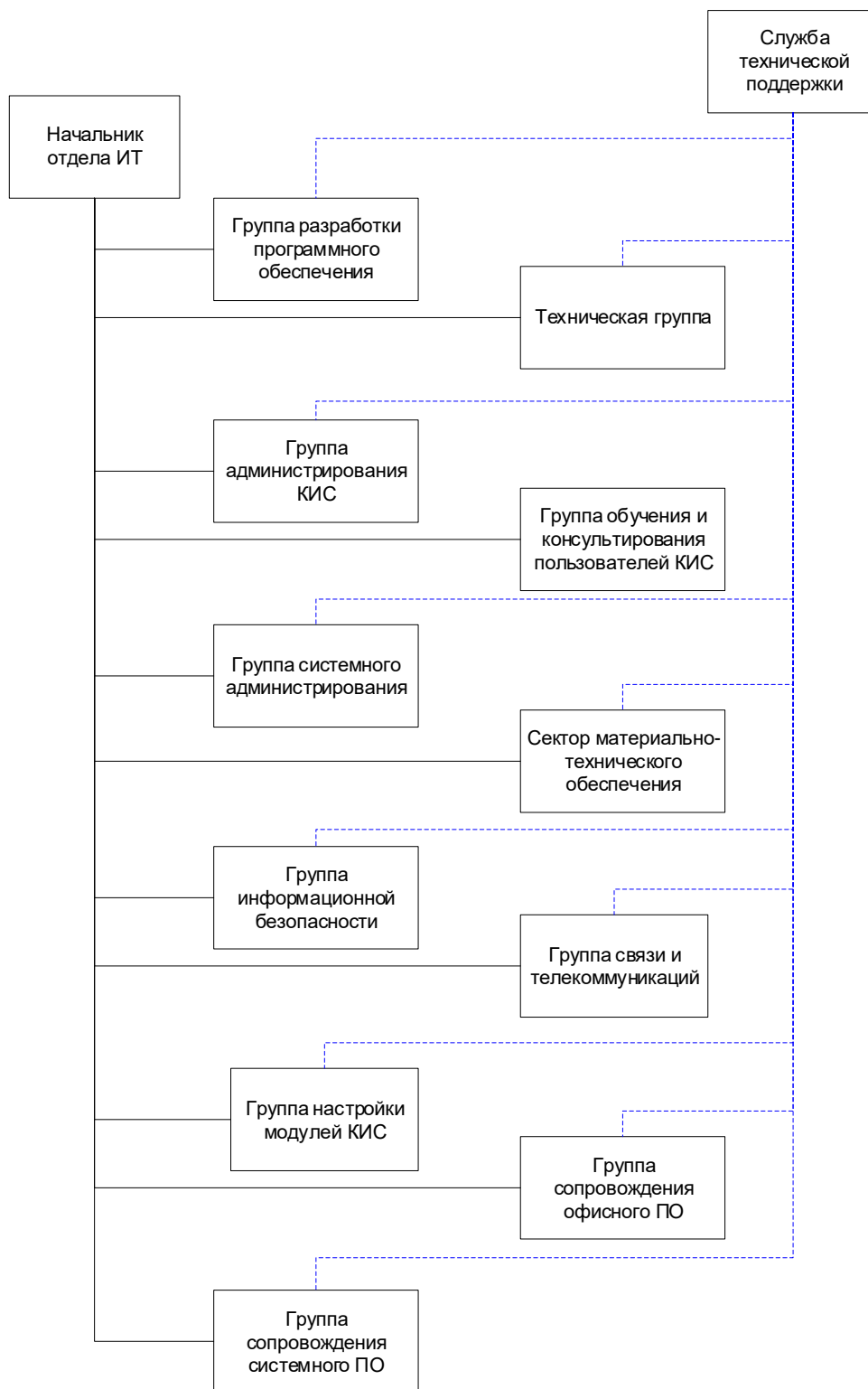


Рисунок 1.2. Структура отдела ИТ

Как видно из рисунка, отдел ИТ имеет сложную структуру, состоит из большого количества подразделений слаженная и профессиональная работа которых призвана организовать бесперебойную и эффективную работу КИС.

Служба технической поддержки пользователей занимает особое место в структуре отдела ИТ. Основная задача Службы – прием заявок пользователей ИТ-инфраструктуры компании на устранение проблем, возникших в ходе ее эксплуатации, в том числе в ходе работы в КИС. Эти проблемы могут иметь совершенно разный характер – от простой замены мышки, до исправления неверных проводок в модуле бухгалтерского учета. В связи с этим, Служба взаимодействует абсолютно со всеми остальными подразделениями отдела ИТ – Служба должна устранить любую проблему, возникшую у пользователя.

Особенностью организационного построения отдела ИТ является то, что администратор Службы имеет право постановки задач сотрудникам любого подразделения отдела ИТ, минуя начальника отдела ИТ. Поставленные Службой задачи для всех сотрудников отдела ИТ являются приоритетными, так как обеспечение бесперебойной и корректной работы КИС является важнейшей функцией отдела ИТ. Можно сказать, что сотрудники всех подразделений отдела ИТ являются одновременно сотрудниками Службы и обязаны исполнять задачи, поставленные начальником или администратором Службы.

Сама Служба имеет простую структуру, представленную на Рисунке 1.3.



Рисунок 1.3. Структура службы технической поддержки

Простота структуры Службы вызвана тем, что непосредственно решением проблем пользователей, связанных с работой КИС, других офисных программных систем, общесистемного программного обеспечения, технических средств занимаются не сотрудники Службы, а сотрудники подразделений отдела ИТ. Сотрудники Службы лишь фиксируют заявки, поступающие от пользователей, распределяют их по исполнителям (сотрудникам отдела ИТ), отслеживают их исполнение, назначают новые задачи, в случае необходимости, доводят до пользователя промежуточные и конечные результаты работы над его заявкой, составляют отчетность по принятым и обработанным заявкам.

1.1.3 Программная и техническая архитектура ИС предприятия

Основой ИТ-архитектуры ООО «Компьютерные Бизнес Системы» является мощный сервер «HP ProLiant DL785 G5». Данный сервер является современным техническим решением, и служит для обеспечения доступа пользователей к общим данным, принтерам, файлам, а также к другим общим сервисам. Сервер «HP ProLiant DL785 G5» – это масштабируемый, мощный сервер с 8 процессорами «Quad-Core AMDR Opteron». Данный сервер является хорошим ценовым предложением из предлагаемых на рынке 8- процессорных серверов. Этот сервер выполняет в организации ООО «Компьютерные Бизнес Системы» функции контроллера домена, сервера электронной почты, контроля доступа в сеть «Интернет», «файл-сервера» и сервера баз данных (БД). Для его бесперебойного питания используется ИБП «APC Smart-UPS RT 1000VA 230V».

Основой построения сети «Ethernet» ООО «Компьютерные Бизнес Системы» является управляемый коммутатор «Fast Ethernet» уровня 3 «D_Link DES-3552P». Коммутатор обеспечивает безопасное подключение конечных пользователей к сети крупных предприятий и предприятий малого и среднего бизнеса (SMB).

Сеть «Ethernet» в ООО «Компьютерные Бизнес Системы» построена в виде «многоуровневой» сети, т.е. часть пользователей подключается к серверу непосредственно, а часть через коммутаторы, установленные в отделах. Как правило это небольшие 8- или 16 –портовые коммутаторы «D-Link» типа «Des 1008A» и «DES 1016».

Для печати, копирования и сканирования документов в ООО «Компьютерные Бизнес Системы» используется сетевое многофункциональное устройство (МФУ) «Xerox Phaser 3635 MFP/S». МФУ установлено в приемной ООО «Компьютерные Бизнес Системы», подключен через сетевой порт к локальной сети предприятия и используется всеми сотрудниками совместно.

На рабочих местах сотрудников установлены персональные компьютеры (ПК) архитектуры «x86» или «x86-64». Типовой ПК сотрудников имеет следующие характеристики:

- Процессор – «Intel I3»;
- Оперативная память: 3 Gb;
- Жесткий диск 500 Гб.

Техническая архитектура ИС изображена на Рисунке 1.4.

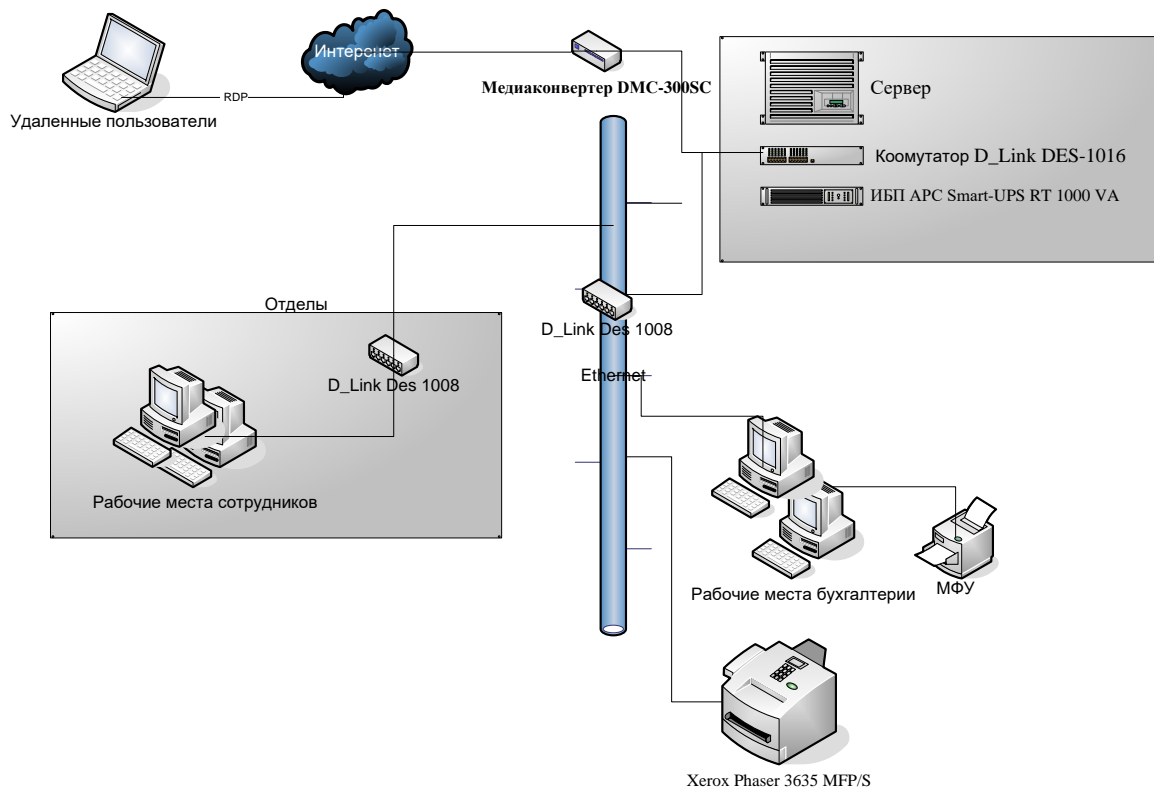


Рисунок 1.4. Техническая архитектура ИС предприятия

На предприятии ООО «Компьютерные Бизнес Системы» остановлены и успешно функционируют следующие ПП.

«1С:Бухгалтерия 8.3 версия для SQL». Данный ПП является системой комплексной автоматизации деятельности бухгалтерии [4]. Возможностей этой

системы достаточно для того, чтобы вести бухгалтерский учет предприятия. В этой системе ведется также расчет заработной платы сотрудников. «1С:Бухгалтерия 8.3» для хранения использует БД «Microsoft SQL Server», установленную на отдельном сервере БД, функционирующем под управлением «Windows Server 2012 Standard Edition». У системы «1С:Бухгалтерия 8.3» отсутствует серверная часть и пользователи обращаются к БД непосредственно со своих рабочих станций или через терминальное подключение [4].

Кроме специального в ООО «Компьютерные Бизнес Системы» используется и общесистемное программное обеспечение (ПО).

В качестве системы антивирусной защиты используется «Symantec Endpoint Security». Система установлена на сервере, а на рабочих местах пользователей установлена клиентская часть.

«Консультант Плюс» – правовая БД используется не только сотрудниками юридического отдела, но и другими сотрудниками.

На рабочих станциях пользователей установлена операционная система (ОС) «Windows версии 7». Эта ОС позволяет подключаться к сети и использовать все ее ресурсы. На всех компьютерах установлено следующее специальное ПО:

«Symantec Endpoint Security» клиентская часть, позволяющая обеспечить защиту компьютера от вирусов;

«Microsoft Office 2016» – ПП для создания и ведения электронных документов;

«Mozilla Firefox 53» – удобный «Интернет-браузер»;

«Outlook Express 2016» – быстрый почтовый клиент.

Программная архитектура ИС изображена на Рисунке 1.5.

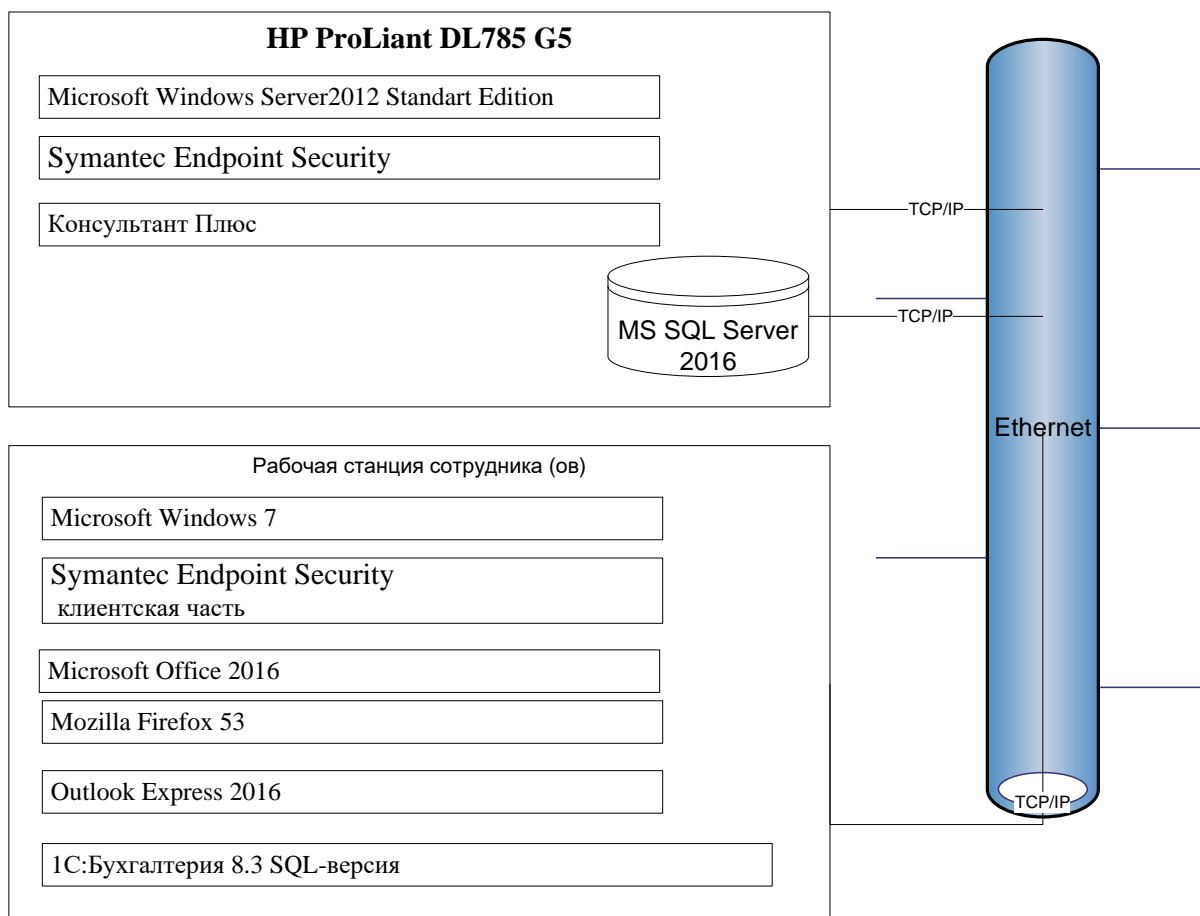


Рисунок 1.5. Программная архитектура ИС предприятия

1.2 Характеристика комплекса задач, задачи и обоснование необходимости автоматизации

1.2.1 Выбор комплекса задач автоматизации и характеристика существующих бизнес процессов

Предприятие работает по многим направлениям, поэтому имеется широкий спектр применений для автоматизации деятельности:

- Бухгалтерский учет;
- Ведение договоров с клиентами;
- Ведение взаиморасчетов с клиентами;
- Учета основных средств и товарно-материальных ценностей;
- Учет разрабатываемых и реализуемых проектов;
- Учет взаимоотношений с поставщиками;
- Учет оргтехники и расходных материалов;

- Техническая поддержка пользователей ИТ – инфраструктуры предприятия.

Техническая поддержка пользователей реализуется в основном путем приема и последующей обработки заявок пользователей на устранение проблем, возникающих при эксплуатации ИТ-ресурсов предприятия.

Процесс приема и обработки заявок пользователей, поступающих в Службу, является многовариантным, сложным и растянутым во времени. В процессе участвуют сотрудники различных подразделений отдела ИТ. Организовать взаимодействие всех подразделений, обмен данными по пользователю и объекту заявки, обмен документами бывает достаточно сложно и проблематично.

При прохождении заявки возникают различные варианты, проблемы и потребности, представленные на Рисунке 1.6.

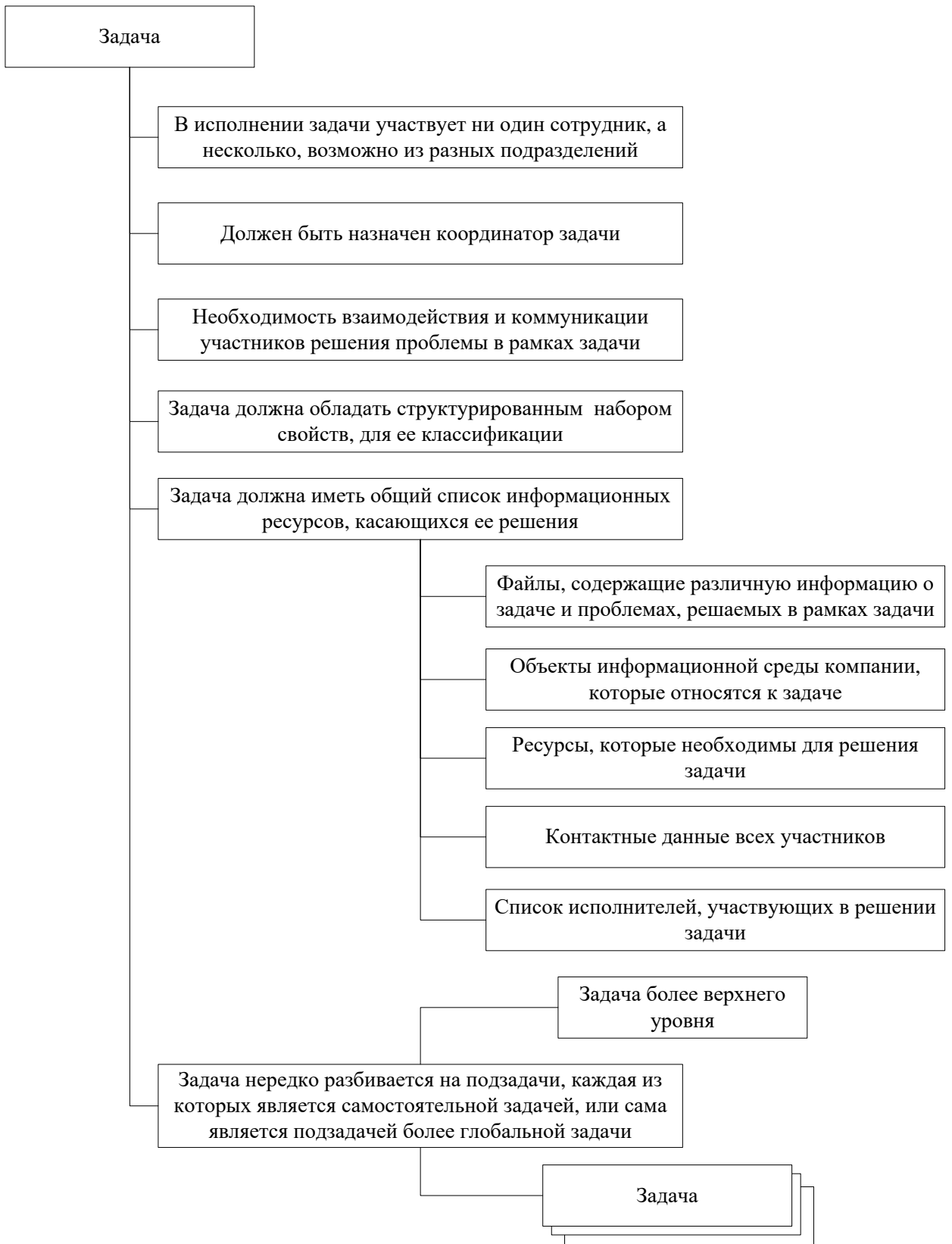


Рисунок 1.6. Структура реальной заявки

Сотрудники отдела ИТ имеют лишь один единственный инструмент для того, чтобы отразить сложный и многовариантный процесс решения проблемы, указанный в заявке – это ведение записей в карточке заявки. Весь ход процесса

решения проблемы записывается вручную в карточке и в последствии сдается в Службу, после чего сотрудник Службы кратко фиксирует в журнале ход исполнения и результат исполнения заявки.

1.2.2 Определение места проектируемой задачи в комплексе задач и ее описание

Для более наглядной реализации проектируемой задачи ДП были разработаны «IDEF-диаграммы» процессов работы предприятия, показанные на рисунках раздела.

Первая диаграмма представляет из себя модель системы в целом, без детализации. Общая модель приведена на Рисунке 1.7.

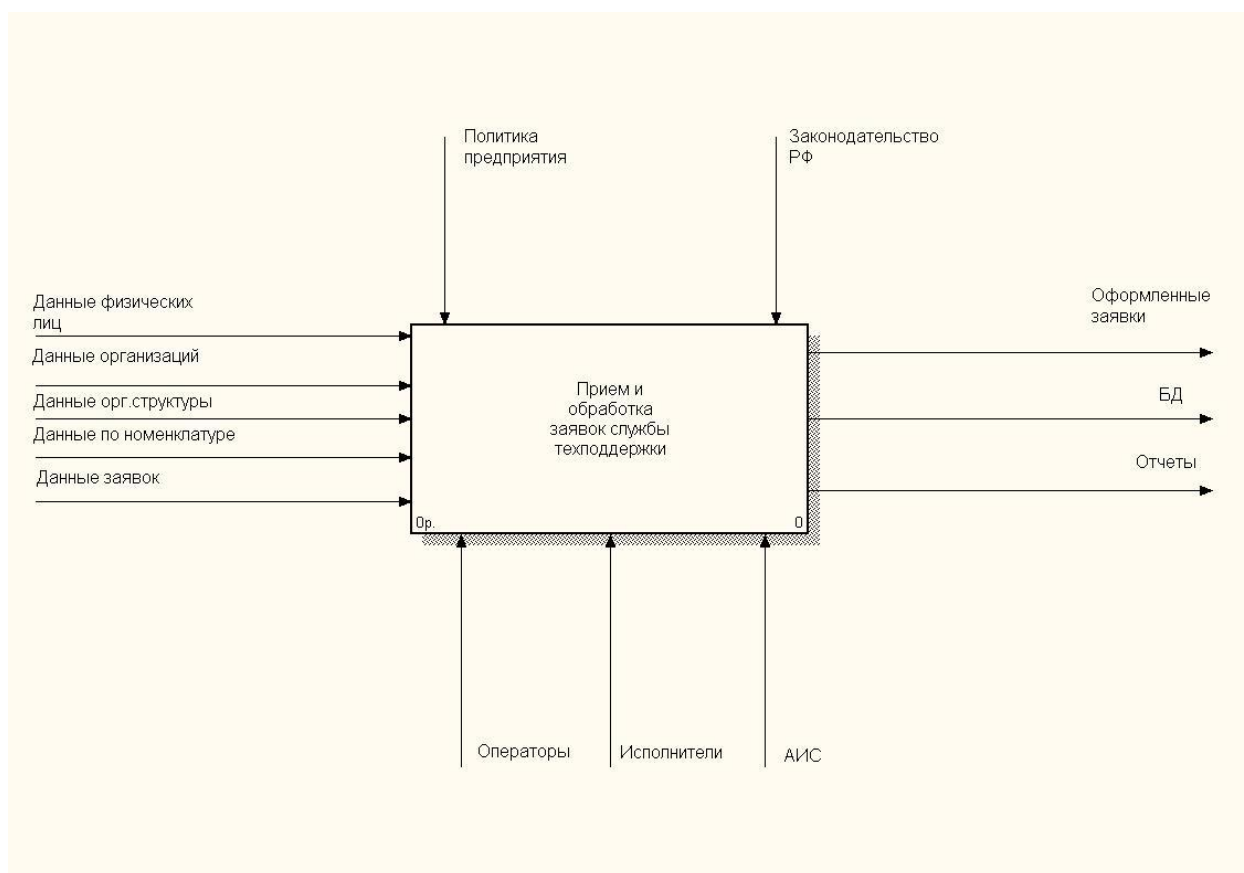


Рисунок 1.7. Общая модель системы

Далее проведена декомпозиция модели на подзадачи. В системе выделены следующие подзадачи:

- Ведение справочников;
- Ведение заявок;
- Оформление отчетов.

Декомпозиция задачи на подзадачи приведена на Рисунке 1.8.

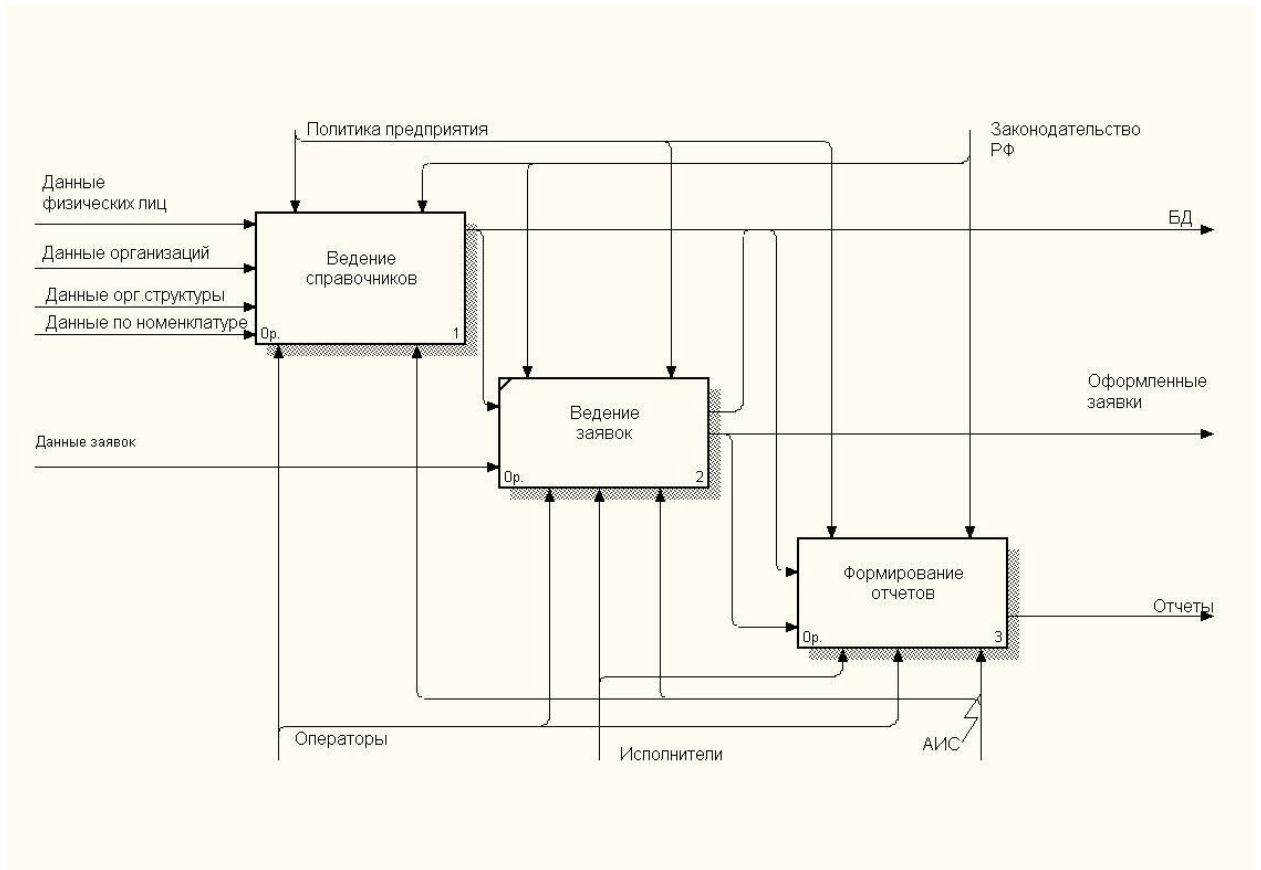


Рисунок 1.8. Декомпозиция первого уровня

Далее каждая из подзадач была детализирована.

Детализация задачи «Ведение справочников» приведена на Рисунке 1.9.

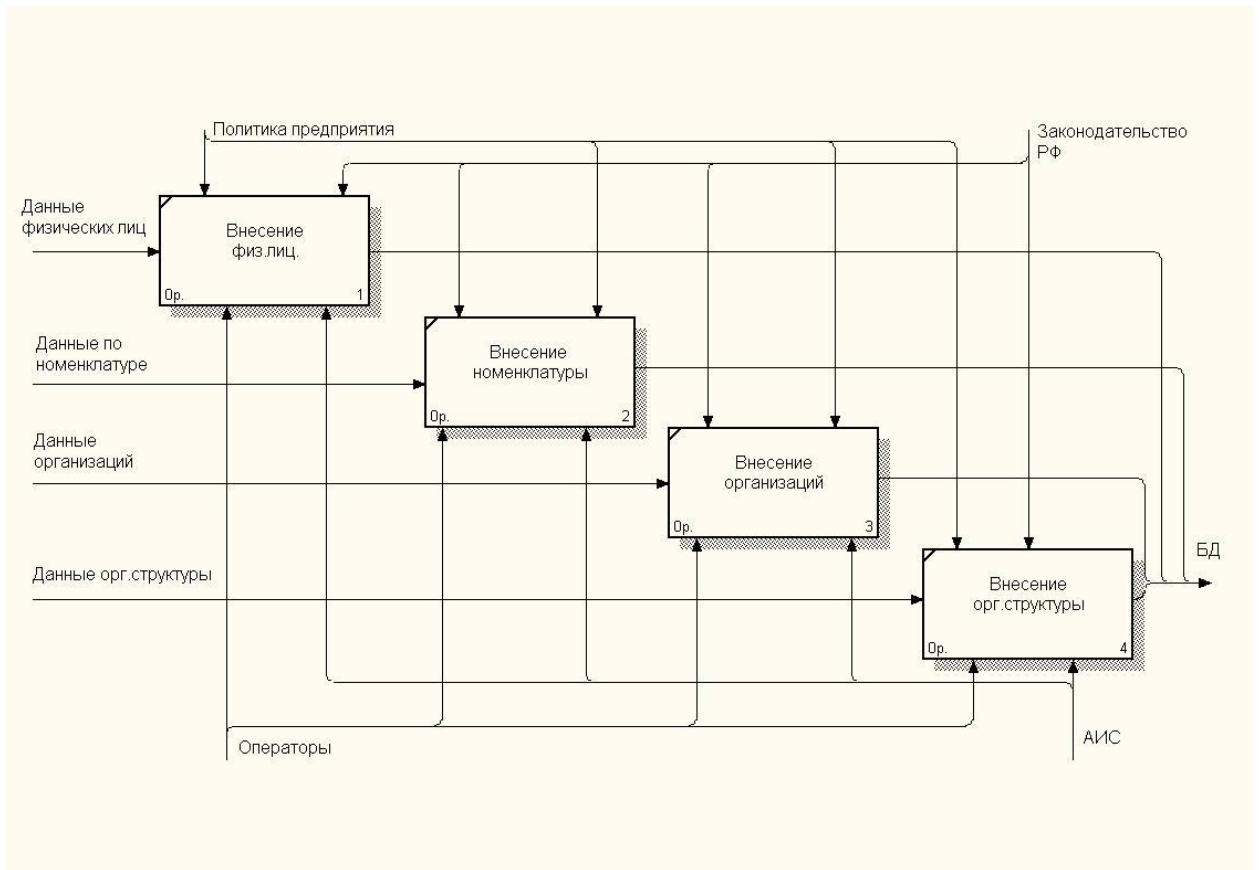


Рисунок 1.9. Декомпозиция подзадачи «Ведение справочников»
 Детализация задачи «Ведение заявок» приведена на Рисунке 1.10.

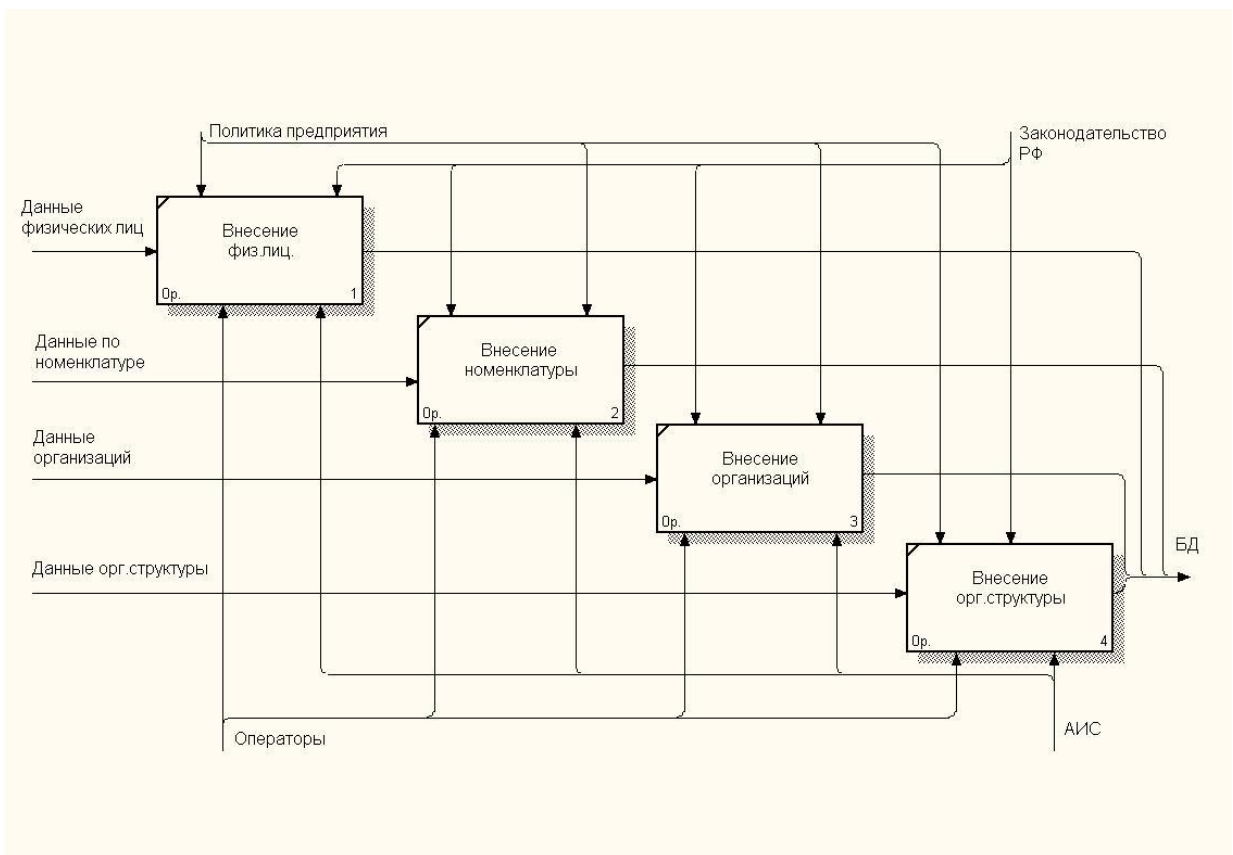


Рисунок 1.10. Декомпозиция задачи «Ведение заявок»

Детализация задачи «Формирование отчетов» приведена на Рисунке 1.11.

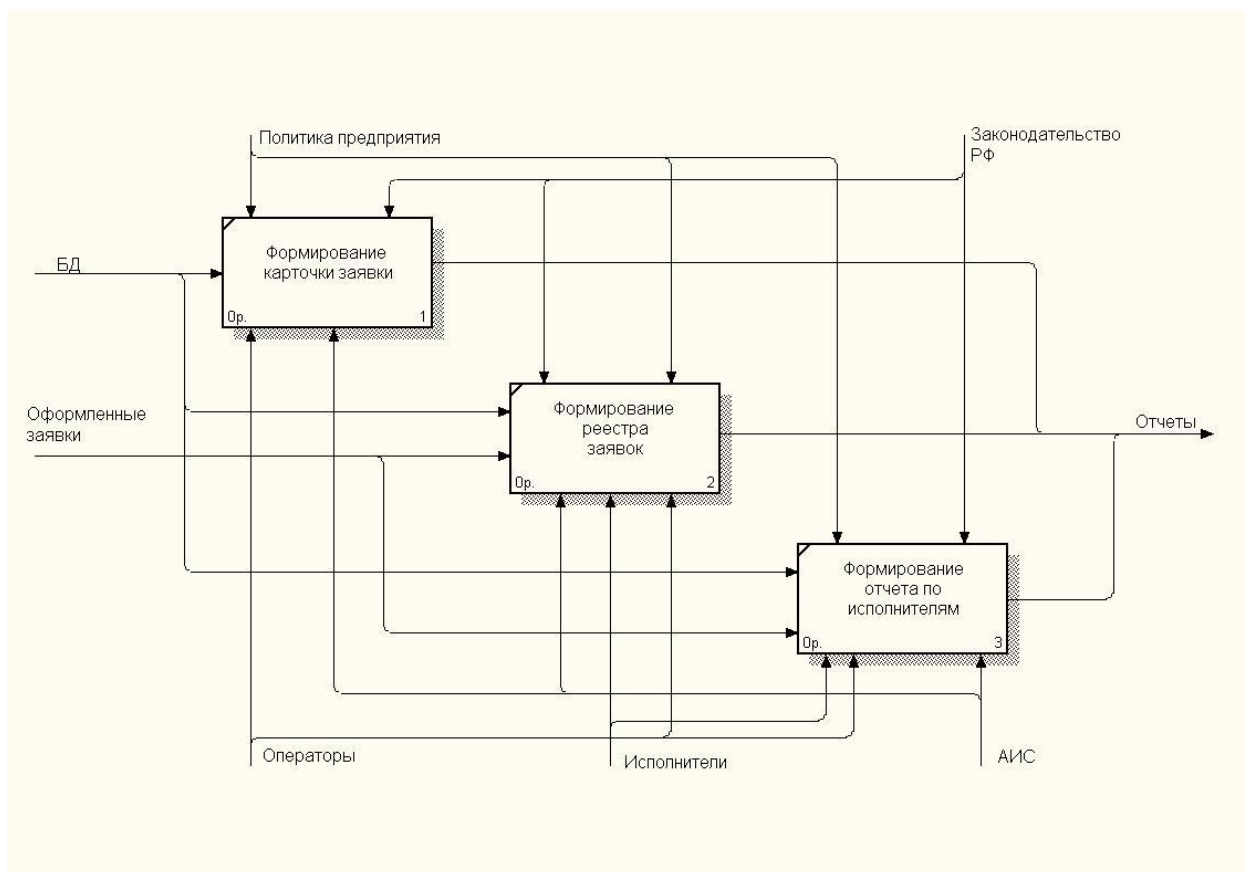


Рисунок 1.11. Декомпозиция задачи «Формирование отчетов»

Благодаря универсальности платформы 1С разрабатываемую систему можно будет запускать как обычное приложение в локальной сети организации и как web-приложение, запускаемое через интернет. В последнем случае приложение публикуется на web-сервере организации и все зарегистрированные пользователи получают к нему доступ через интернет. В любом случае приложение работает с единой базой данных. В случае web-приложения пользователи сами могут заходить в приложение и оставлять там заявки, просматривать историю своих заявок, отслеживать состояние своих заявок. Так как система 1С не предполагает работу неавторизованного пользователя, пользователь должен быть зарегистрирован в системе. Зарегистрировать пользователя может как сотрудник Службы, выдав ему логин и пароль, так и сам пользователь при первом входе в систему.

1.2.3 Обоснование необходимости использования вычислительной техники для решения задачи

Степень удовлетворенности пользователей работой Службы зависит от организации процесса принятия заявок от пользователей и их последующей обработки и сопровождения.

Самым большим недостатком существующего процесса обработки заявки пользователя является длительность этого процесса, что в свою очередь связано со сложностью и многовариантностью ее прохождения, участия большого количества специалистов различных подразделений отдела ИТ в рассмотрении заявки, проблемами, возникающими при их взаимодействии.

В п. 1.2 описан процесс фиксации и отработки заявки в идеальном, гипотетическом варианте. По такому варианту обрабатываются лишь небольшая часть заявок, которые можно назвать «простыми». Но большинство заявок, поступающих в Службу «простыми» вовсе не являются. Это связано со сложностью используемых в компании программных и технических средств построения информационной среды и, прежде всего, со сложностью используемой КИС.

Основной проблемой существующей системы приема и обработки заявок является разрозненность хранения данных по отдельным папкам и файлам на сервере Службы, а также хранение части данных в бумажном виде. Это приводит к неоднозначности данных, возможности их потери, ошибках в расчетах, начиная от банальных опечаток, заканчивая потерей целых кусков данных, большой объем ручного ввода данных, что также приводит к ошибкам, невозможность составления аналитических отчетов [11]. Существующий вариант приема и обработки заявок имеет, конечно, целый ряд недостатков, представленных на Рисунке 1.12.

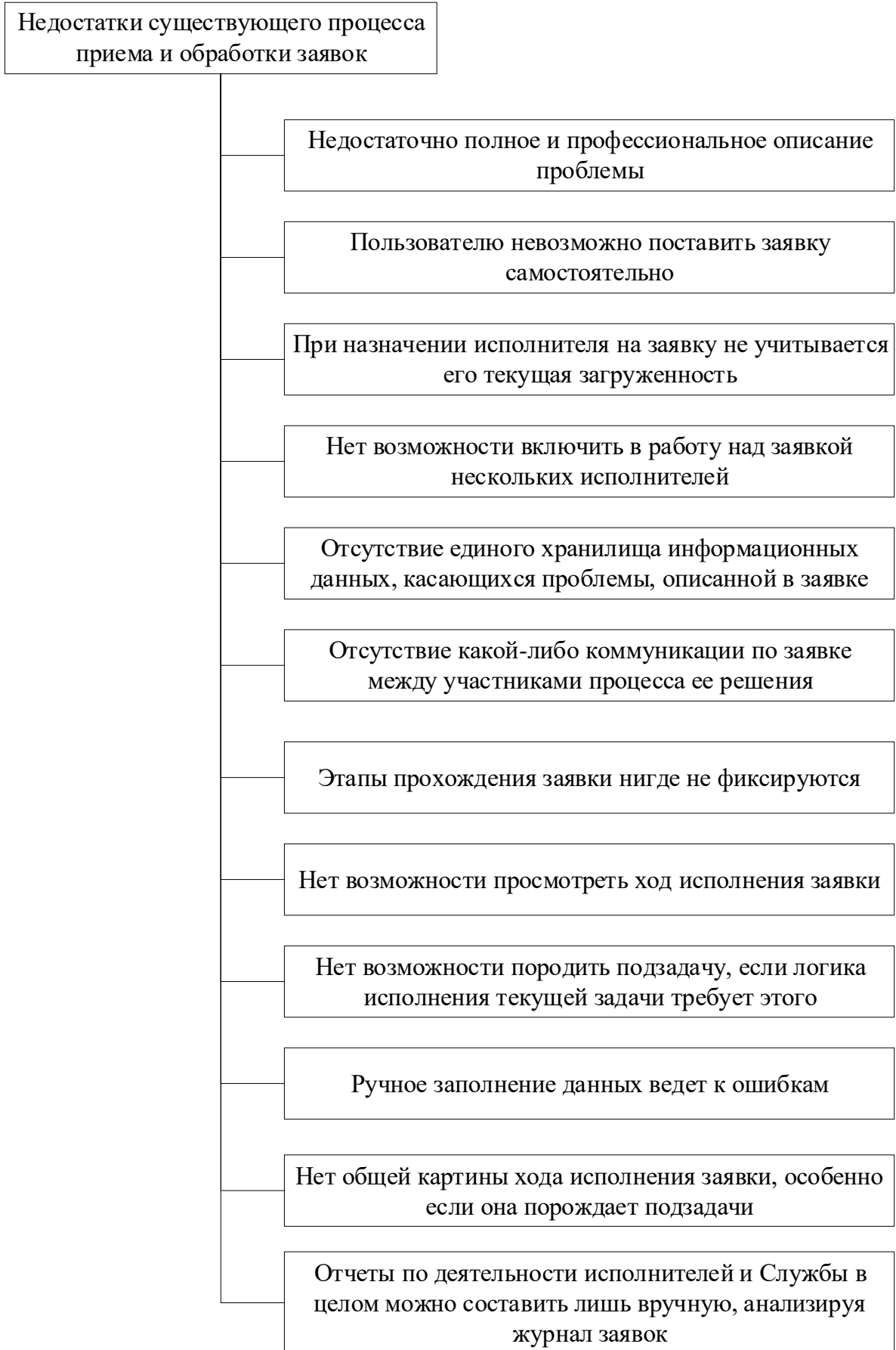


Рисунок 1.12. Недостатки существующего процесса приема и обработки заявок

Эффективно организовать процесс приема и обработки заявок можно путем разработки и внедрения в Службе автоматизированной информационной системы учета заявок и их сопровождения на протяжении всего жизненного цикла заявки. Преимущества автоматизированного варианта работы приведены на Рисунке 1.13.

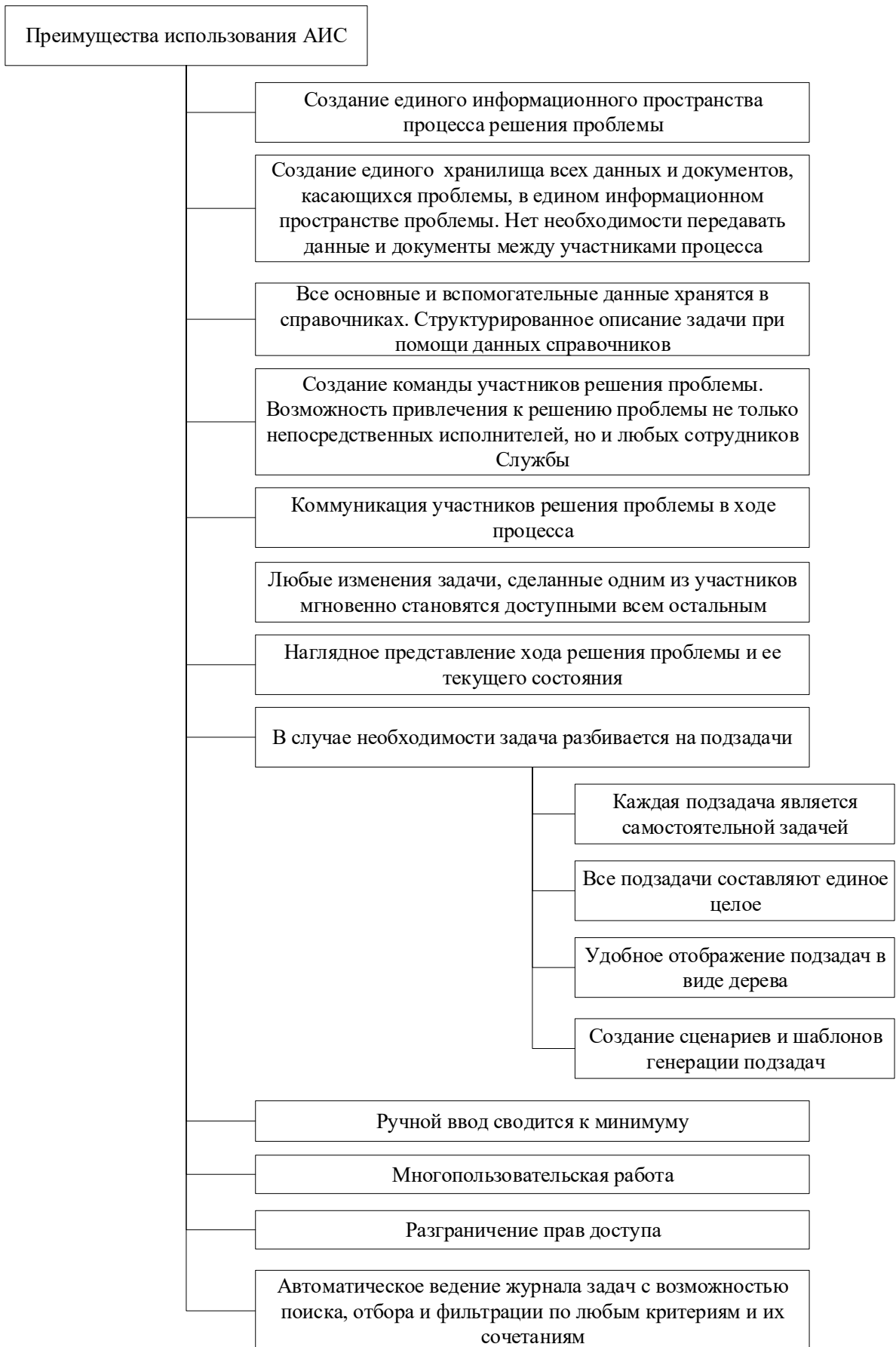


Рисунок 1.13. Преимущества автоматизированного варианта приема и обработки заявок

Кроме общего времени, затрачиваемого на операции документооборота в рамках процесса согласования заявки можно привести ряд показателей сравнения систем бумажного и электронного документооборота.

Таблица 1.2. Сравнение основных временных показателей документооборота процесса решения заявки

Операция	Время при бумажном документообороте	Время при электронном документообороте
Поиск документа по номеру	В зависимости от местонахождения документа от 10 минут до бесконечности	Не более 0,5 минуты
Поиск документа по другим реквизитам	В зависимости от сложности поиска от 15 минут до бесконечности	Не более 1 минуты
Составление списка документов по одному исполнителю за период	В зависимости от длины периода от 20 минут до нескольких часов	Не более 2 минут не зависимо от периода
Составление списка просроченных документов по всем ответственным исполнителям	В зависимости от количества исполнителей от 1 часа до нескольких часов	Не более 3 минут не зависимо от количества исполнителей
Доставка исполнителю извещения о назначении его ответственным по документу	В зависимости от присутствия исполнителя на рабочем месте от 30 минут до нескольких часов	Не более 0,1 минуты сразу после назначения
Доставка всем ответственным исполнителям извещения о неисполненных ими документах за период	В зависимости от глубины периода и количества исполнителей – от нескольких часов до нескольких дней	Не более 0,1 минуты после составления списка неисполненных документов
Составление отчета о документообороте за период	В зависимости от глубины периода и количества документов – от нескольких часов до нескольких дней	Не более 1 минуты не зависимо от глубины периода и количества документов

Эффект от внедрения разрабатываемой системы будет состоять в экономии времени сотрудников и повышении эффективности их работы.

Еще одним преимуществом применения разработанной АИС является безопасность хранения данных. При хранении данных на традиционных бумажных носителях исчезновение любого бумажного документа было критичным. Такое

исчезновение могло случиться как в результате действий злоумышленников, так и в результате действия чрезвычайных обстоятельств. При хранении данных в электронном виде существующие современные методы обеспечения безопасности хранения данных и их дублирования делают такое хранение полностью надежным.

Схема документооборота приведена на Рисунке 1.14.

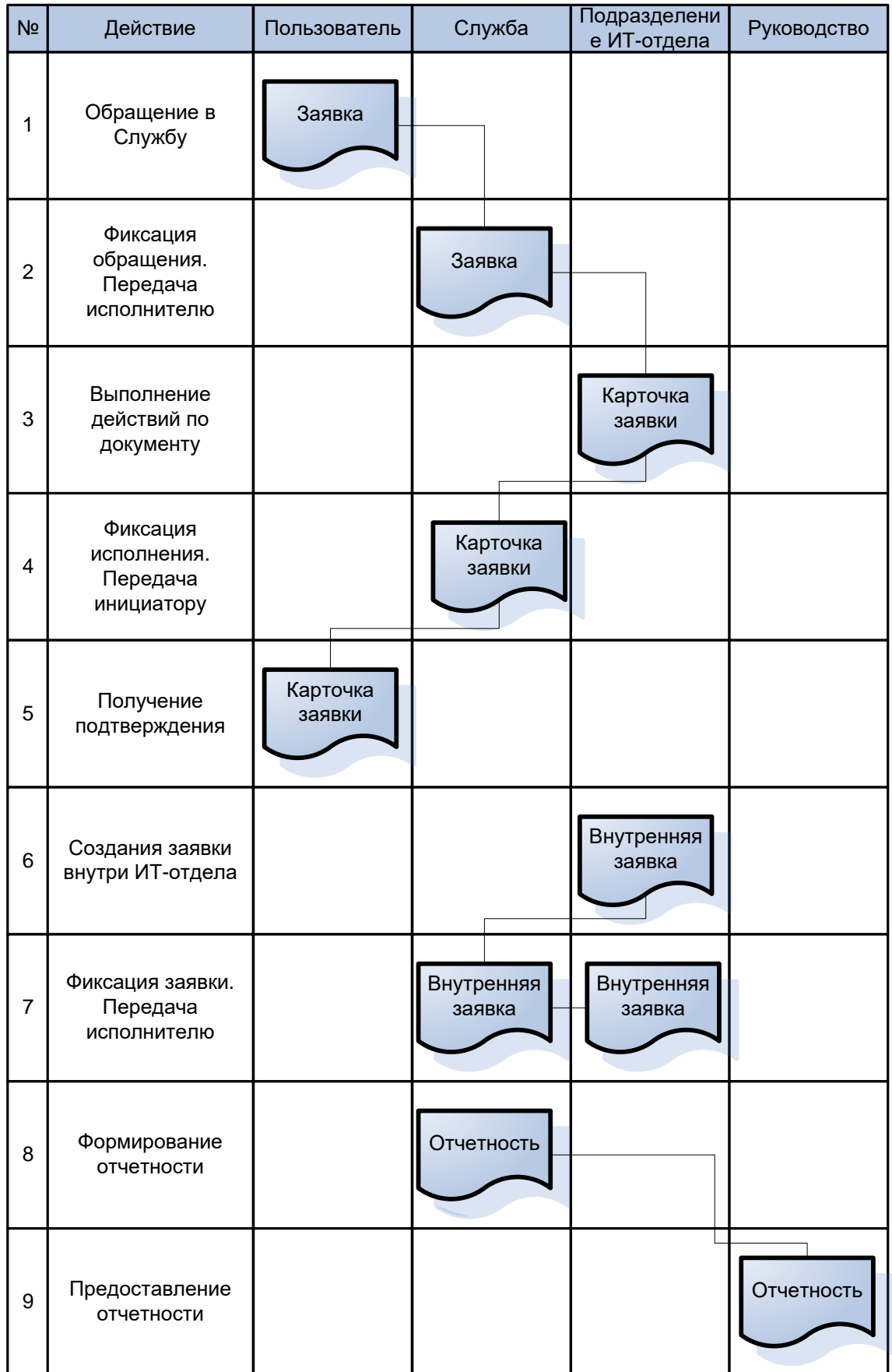


Рисунок 1.14. Схема документооборота

1.2.4 Анализ системы обеспечения информационной безопасности и защиты информации

При использовании компьютеризированного варианта документооборота уровень обеспечения информационной безопасности хранения данных повышается во много раз. Сама система документооборота, основанного на компьютерной системе, заставляет пользователя быть более ответственным и внимательным в этом вопросе.

С точки зрения технической реализации защита данных осуществляется при помощи иерархической системы паролей для доступа к ресурсам АИС. В первую очередь, это пароль авторизации пользователя при входе в ОС его компьютера. Эта авторизация открывает пользователю доступ к ресурсам его компьютера и к документам, находящимся в нем. При этом политика безопасности предприятия должна быть построена т.о., чтобы пользователь не был полновластным «хозяином» на своем ПК и не мог, например, установить «вирусное» ПО или ПО по копированию данных. Ограничение прав немного усложняет работу операторов, но при этом обеспечивает гарантию защищенности данных. Необходимо всегда искать баланс между удобством работы оператора и безопасностью хранения корпоративных данных.

При авторизации пользователя на своем ПК, он получает доступ не только к ресурсам своего ПК, но и к ресурсам локальной сети всего предприятия. Это происходит в том случае, если пользователь входит на ПК как сетевой пользователь. При этом вопрос разграничения прав пользователей нужно рассматривать еще более тщательно. Установить права сетевого пользователя нужно так, чтобы обеспечить ему возможность свободно работать со своими документами, но при этом запретить доступ к документам, прав на работу с которыми у него нет. В этом случае одновременно решается задача защиты данных от несанкционированного доступа и от непреднамеренной их порчи.

Правом распределения прав пользователей по доступу к данным обладает системный администратор. Именно он разграничивает права доступа пользователей к документам и приложениям, как в сети, так и на локальных компьютерах.

Вторым уровнем защиты важной информации является парольная защита доступа непосредственно в АИС, реализованную в «1С:Предприятие 8.3».

Платформа «1С:Предприятие 8.3» имеет в своем составе механизм ведения пользователей и назначения им прав и ролей доступа к данным. При помощи этого механизма можно гибко настроить права доступа пользователей только к необходимым им данным.

Третьим уровнем защиты информации является пароль авторизации доступа к серверу «MS SQL Server» при использовании «клиент-серверного» варианта работы «1С:Предприятие 8.3». В данном варианте работы данные защищены не только системой разграничения прав доступа «1С:Предприятие 8.3» но и СУБД «MS SQL Server», что повышает уровень безопасности.

Правом распределения прав доступа пользователей на уровне «1С:Предприятие 8.3» лежит на администраторе системы. Совместно с системным администратором они настраивают и доступ к СУБД «MS SQL Server».

Важным фактором, повышающим уровень информационной безопасности при внедрении АИС в ООО «Компьютерные Бизнес Системы» является возможность «спасти» всю БД документов на электронном носителе при возникновении неполадок. В дальнейшем эта копия БД может быть восстановлена и работа с продолжена.

Кроме организации защиты информации на уровне паролей организация не пренебрегает и физическими методами защиты данных. Сервер компьютерной сети предприятия, «1С:Предприятие 8.3» и сервера БД «MS SQL Server» расположены в отдельном помещении (серверной), в котором кроме особых условий, необходимых для работы серверов (температура, вентиляция, ...) созданы и специальные условия безопасности, исключающие проникновение посторонних лиц.

Основным документом, определяющим правила и методы обеспечения информационной безопасности в организации, является «Положение о политике безопасности в ООО «Компьютерные Бизнес Системы».

Основными инструментами обеспечения информационной безопасности являются инструменты ОС Windows на серверной части и клиентских машинах - контроль учетных записей пользователей, Защитник Windows и брандмауэр

Контроль учетных записей пользователей уменьшает риски использования записи администратора для выполнения повседневных задач, требуя согласия пользователя для выполнения действий, влияющих на всю систему.

Защитник Windows — это антишпионская программа, которая непрерывно отслеживает настройки системы, чтобы предотвратить установку известных шпионских программ и оповестить пользователя о подозрительных действиях.

Двухсторонний брандмауэр, отслеживает входящий и исходящий трафик и полностью поддерживающий протокол IPv6. Он обеспечивает профили многоадресного доступа, что дает защиту для всех сетей, к которым пользователь подключен одновременно.

1.3 Анализ существующих разработок и выбор стратегии автоматизации «КАК ДОЛЖНО БЫТЬ»

1.3.1 Анализ существующих разработок для автоматизации задачи

На текущий момент на рынке существует ряд готовых программных решений выполняющих задачу автоматизации работы служб технической поддержки пользователей. Рассмотрим некоторые из этих программных решений.

Open Ticket Request System (Рисунок 1.15) — бесплатная система приема и обработки заявок, написанная на языке perl с поддержкой различных СУБД, в том числе свободных. OTRS обладает рядом преимуществ:

- Систему работает как на windows так и на linux;
- Интеграция с LDAP;
- Возможность создавать очереди заявок;
- Личный кабинет для пользователей, где видна вся история обращений;
- Гибкая настройка прав пользователей;
- Эскалация обращений;
- Простой редактор шаблонов;
- Улучшенный поиск;
- Русская локализация;
- Гибкая система отчетов;
- Возможна интеграция с уже имеющимися БД клиентов и сотрудников.

Система легко расширяется через дополнительные модули: календарь, база знаний/FAQ, файловый менеджер, ITSM и другие.

Многие компании используют OTRS в качестве helpdesk-системы по двум причинам: она бесплатная и отлично подходит для интеграции с веб-сайтом и электронной почтой.

Для функционирования системы необходимы: WEB-Сервер (можно использовать Apache 2), сервер баз данных (можно использовать MySQL 5.0 или выше), рабочая среда Perl. Для клиентского приложения требуется любой браузер с поддержкой JavaScript.

Настройка системы кропотлива и сложна требует глубоких знаний и хороших специалистов. Доработка каких-либо модулей требует знаний языка программирования Perl. Посредник внедрения системы, компания Radiant System предлагает услуги внедрения стоимостью 160 000 рублей за 10 дней работы.

Техническая поддержка присутствует на английском языке. Поддержка на русском языке – платная (от компании Radiant System)

Система является открытой, то есть возможна самостоятельная разработка модулей и в целом доработка системы. API-интерфейс также можно скачать на официальном сайте.

Данная система хорошо поддается масштабируемости. Система может обрабатывать большое количество одновременно работающих запросов и пользователей. Возможно использование кластерных конфигураций.

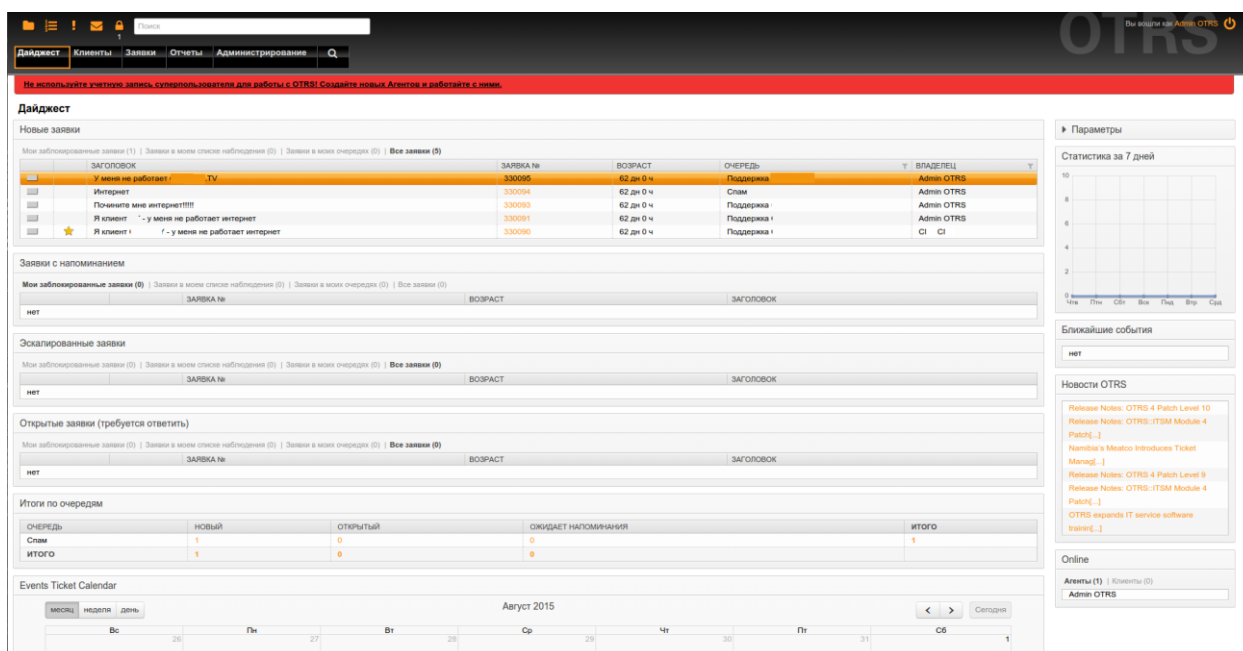


Рисунок 1.15. Внешний вид OTRS

Система **Trellis Desk** – еще одна типичная система класса service desk .

Для работы системы необходимо наличие следующих компонентов: PHP 4.3 или выше, MySQL 4.1 или выше, веб-хост с поддержкой рассылки входящей электронной почты, библиотека GD2 Library – для динамической работы с изображениями.

Система остро нуждается в PHP-программисте в штате компании. Если потребуется русификация клиентской части, интеграция с Active Directory, доработка ролей, доработка кодировок то найм программиста выйдет дешевле, чем внедрение коммерческой версии системы.

Техническая поддержка от разработчиков отсутствует вовсе. Имеется peer-to-peer поддержка на форуме системы.

На официальном сайте системы можно найти данные о внушительном количестве успешных внедрений системы в самых разных компаниях. В русскоязычном секторе интернета можно найти примеры внедрения системы в России в небольших компаниях.

Имеется тесная интеграция с электронной почтой, а также возможность импорта/экспорта шаблонов графического интерфейса и возможность экспортировать созданную в системе локализацию, либо импортировать готовую. Импорт/экспорт CSV отсутствует.

API-интерфейс отсутствует. Код программы является открытым для изменений. Система представляет собой «Коробочное» веб-приложение. Мобильное приложение отсутствует.

Интерфейс системы Trellis Desk представлен на Рисунке 1.16.

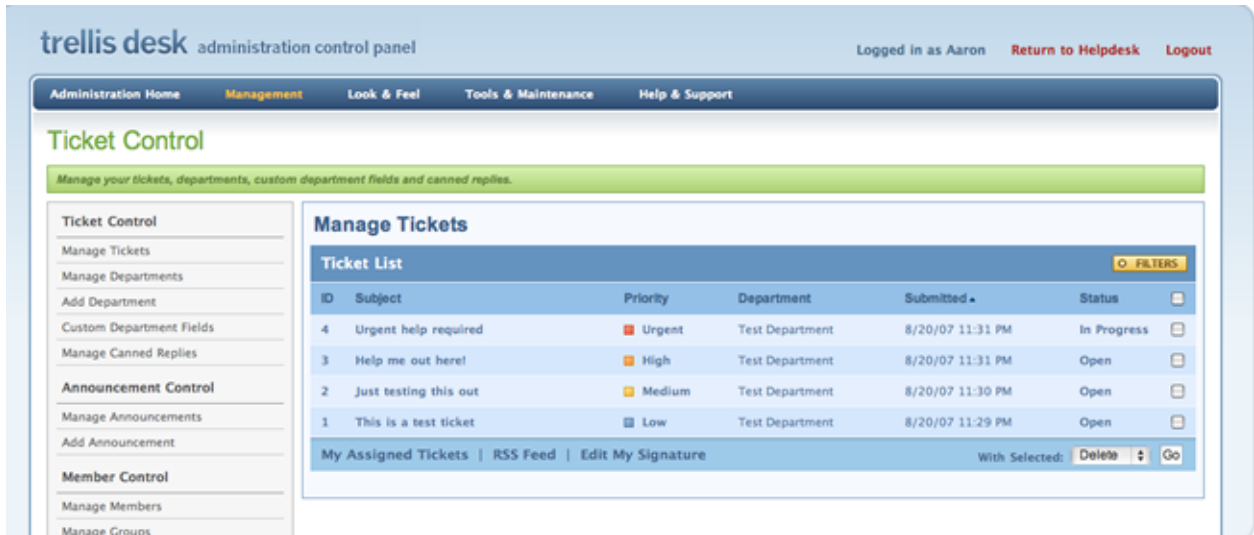


Рисунок 1.16. Интерфейс Trellis Desk

В системе имеются мощные средства администрирования. Администратор может создавать пользователей, создавать группы пользователей и назначать определённые права для групп.

Так же в системе можно вести базу знаний по итогам отработанных проблем. Авторы могут создавать коллекции написанных статей, содержащих существующие решения типовых проблем. При создании пользователем инцидента система может автоматически отображать статьи базы знаний, которые могут оказать помощь в решении возникшей проблемы.

Система **Service Desk Lite**. Компания-разработчик позиционирует эту систему, как полностью бесплатную. Но на самом деле Бесплатной является лишь версия Lite, которая подойдет лишь небольшой группе разработчиков до трех человек для трекинга ошибок и багов. Именно столько пользователей поддерживает бесплатная версия системы.

Система является «коробочным» Windows-приложением. Поэтому бесплатные операционные системы использовать не получится. Внедрить систему можно за два-три дня. Присутствует поддержка по email, но только англоязычная.

Данных по внедрению системы в России очень мало, практические примеры отсутствуют. Русскоязычная версия тоже отсутствует, а активность сообщества весьма слаба.

Отсутствуют интеграция с другими системами. API-интерфейс также отсутствует, поэтому разработка собственных инструментов интеграции невозможна, а код программы закрыт. Нет инструментов экспорта и импорта данных. Мобильного приложения тоже нет.

В системе реализован простейший учет активов и справочник брендов.

Проблемы клиентов регистрируются в разделе «Complaint Management». В инциденте указывается продукт, с которым возникли неполадки, его бренд и пользователь. Также выбирается контракт на обслуживание. Затем можно назначить инцидент на исполнение, а затем закрыть его.

В сравнении с аналогами функциональные возможности системы наиболее примитивны. Из положительных моментов можно отметить удобный отчет по организации, в котором в наглядной форме отображены все клиенты, продукты, сотрудники, активы, бренды, контракты, список жалоб. Отчет представлен на Рисунке 1.17.



Рисунок 1.17. Отчет по организации в Service Desk Lite

Система **OSTicket**. Система представлена в двух вариантах. Первый – загружаемое и устанавливаемое приложение (бесплатно). Второй – работа в облаке за ежемесячную оплату. По функционалу системы абсолютно одинаковы.

Для работы бесплатной версии компании необходим собственный web-сервер с установленными компонентами: PHP 5.3, MariaDB 5.5, PHP manager for IIS.

С установкой системы справится любой системный администратор. Требования к конфигурации сервера и рабочих станций минимальны.

В некоторых случаях система может потребовать доработок. Если в штате есть программист, разбирающийся в PHP, то он справится с этой задачей.

Для бесплатной версии техническая поддержка не предоставляется. Но можно пользоваться официальным форумом для решения возникших вопросов. Имеется база знаний, в которой описано множество моментов по работе с системой.

Большое количество предприятий уже используют данную систему. OSTicket устанавливается и интегрируется очень просто. Установку системы можно выполнить за 10 дней.

Сообщество пользователей системы является очень активным, в том числе на просторах рунета. Имеется русскоязычная группа в социальной сети «ВКонтакте». В системе имеется поддержка русского языка.

Интеграция с какими-либо другими продуктами отсутствует, но, при наличии хорошего специалиста, можно сделать интеграцию с WordPress, Active Directory, Joomla.

Имеется API-интерфейс. Данный API позволяет реализовывать собственные HTML-формы. Код системы полностью открыт и доступен в свободном доступе.

Бесплатная версия распространяется в виде «Коробочного» решения. Установщик скачивается с официального сайта. Система реализована в виде WEB-приложения. Официальное мобильное приложение отсутствует, но имеется множество оптимизаций шаблона под мобильные системы, а также неофициальные мобильные приложения под Android и IOS, как платные, так и бесплатные. Реализована возможность импортировать и экспортировать csv-файлы.

Интерфейс системы OSTicket представлен на Рисунке 1.18.

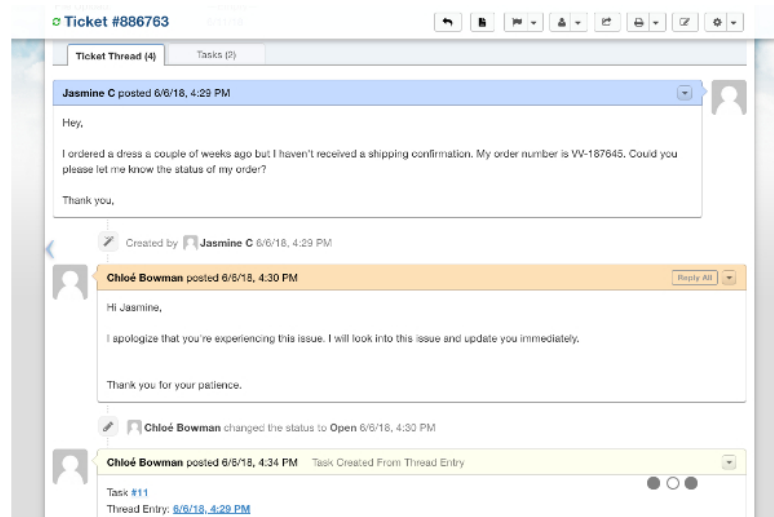


Рисунок 1.18. Интерфейс пользователя OSTicket

Для того чтобы сделать вывод было сформировано несколько сравнительных таблиц.

Таблица 1.3. Затраты на внедрение

	Лицензия	Сопровождение	Тех. средства	Ср. балл
OTRS ITSM 5S	10	5	10	8,3
Trellis Desk	10	6	10	8,7
SD Lite	10	8	10	9,3
OSTicket	10	7	10	9

Таблица 1.4. Сложность внедрения

	Поддержка	Опытвнедр.	Рус. Яз.	Скорость внедрения	Сообщество	Ср. балл
OTRS ITSM 5S	7	10	7	8	10	8,4
Trellis Desk	3	10	6	8	8	7
SD Lite	4	5	0	10	4	4,6
OSTicket	3	10	10	8	10	8,2

Таблица 1.5. Используемые технологии

	OTRS	Trellis Desk	SD Lite	OSTicket
Интеграция	6	4	0	4
API	10	5	0	10
Масштабируемость	5	5	0	5
Класс системы	5	5	5	5

Импорт/Экспорт	8	5	0	8
Интерфейс	10	10	5	10
Мобильное прил.	0	0	0	2
Средний балл	6	5	2	7

Таблица 1.6. Функциональность

	OTRS	Trellis Desk	SD Lite	OSTicket
Инциденты	7	5	2	4
Запросы	7	5	2	5
Активы	7	0	2	0
Конфигурации	7	0	2	0
Знания	6	6	0	6
Безопасность	8	6	0	6
Обслуживание	7	0	0	6
Каталог услуг	7	0	0	0
Средний балл	7	2,75	1	3,38

Таблица 1.7. Итоговая таблица

	OTRS	Trellis Desk	SD Lite	OSTicket
Затраты	8,3	8,7	9,3	9
Внедрение	8,4	7	4,6	8,2
Технологии	6	5	2	7
Функционал	7	2,75	1	3,38
ITIL	7	5	2	5
Удобство	6	7	0	7
Средний балл	7,1	6	3,2	6,6

В целом, общие затраты на внедрение рассмотренных систем практически схожи между собой. Какие-то системы обойдутся чуть дороже в момент внедрения, другие обойдутся дороже в процессе сопровождения, но в целом по суммарным затратам системы практически идентичны. По критерию «сложность внедрения» можно выделить двух явных лидеров – это системы OTRS ITSM 5S и OSTicket. По функционалу они тоже лидеры.

Рассмотрев несколько из наиболее популярных и мощных готовых решений для автоматизации деятельности службы технической поддержки предприятия, мы тем не менее делаем вывод о необходимости разработки собственного программного

продукта, который будет являться конфигурацией для системы «1С:Предприятие 8.3» и будет реализовывать именно ту функциональность, которая нам нужна.

1.3.2 Выбор и обоснование стратегии автоматизации задачи

При анализе работы службы технической поддержки ООО «Компьютерные Бизнес Системы» обнаруживается много недостатков организации учета заявок. Архив заявок представляет собой «бумажные» хранилища, поиск информации в которых производится «вручную», что является очень сложным и время затратным процессом.

Существует четыре варианта стратегии автоматизации:

- хаотичная;
- по участкам;
- по направлениям;
- полная;

Хаотичная автоматизация состоит из набора автоматизированных участков, не связанных друг с другом. Такая автоматизация определяется оперативными задачами и обычно не отражается в стратегических планах компании.

Автоматизация по участкам представляет собой процесс автоматизации отдельных функциональных участков, например, отдел бухгалтерии, коммерческий отдел и т.д.

Автоматизация по направлениям. Отличается от автоматизации по участкам тем, что предполагает участие всех функциональных подразделений, деятельность которых связана с направлением автоматизации.

Полная автоматизация предполагает автоматизацию абсолютно всех бизнес-процессов компании.

Стратегия автоматизации, выбранная в исследовании – по направлению, так как автоматизируется одно направление деятельности предприятия – прием и сопровождение заявок, поступающих в службу технической поддержки. Такой подход подразумевает участие в процессе всех отделов, функционирование которых связано с работой службы технической поддержки, а это практически все предприятие.

Анализ задачи приводит к пониманию, что ее решение средствами учета с использованием «бумажных документов» труднореализуемо. Следовательно, стратегия решения – это автоматизация всех процессов, относящихся к учету и сопровождению заявок, поступающих в службу технической поддержки.

1.3.3 Выбор и обоснование способа приобретения ИС для автоматизации задачи

С позиции информационного менеджмента, создание и внедрение автоматизированной информационной системы планирования может быть выполнено различными способами:

- покупка готовой системы;
- создание проектной команды и разработка системы средствами и трудовыми ресурсами компании;
- заказ системы у компании разработчика.

Необходимо рассмотреть различные варианты способов приобретения. При покупке готовой системы можно приобрести довольно дешевую ИС, но невозможно подобрать программный продукт, который будет на 100% удовлетворять требования компании. Перестройка функционала ИС стоит недешево, поэтому такой вариант не рассматривается.

Если заказывать систему у компании разработчика, то технология работы компании станет известной компании разработчика, а это полностью противоречит политике информационной безопасности компании.

Самостоятельная разработка ИС полностью соответствует политике информационной безопасности компании ООО «Компьютерные Бизнес Системы», а также позволяет гибко изменять запрограммированный механизм ИС, т.е. адаптировать её под новые технические и программные платформы.

Требуется разработать собственную АИС учета взаимоотношений с клиентами ООО «Компьютерные Бизнес Системы» и прежде всего выбрать систему (платформу) для разработки собственной АИС.

1.4 Обоснование проектных решений

1.4.1 Обоснование проектных решений по информационному обеспечению

Основными входными данными разрабатываемой системы являются заявки пользователей ИТ-инфраструктуры компании, поступающие в Службу. Эти заявки не имеют строго определенной формы, могут передаваться письменно, по электронной почте, либо с использованием web-клиента через подключение к системе в любом браузере. Т.о. либо пользователь вводит информацию о заявке самостоятельно, либо сотрудник Службы вводит заявку в систему в виде электронного документа «Заявка».

Выходными документами системы являются те же заявки, но уже в электронном виде с отметками исполнителей об исполнении (или невозможности исполнения), историей работы над заявкой, перепиской по заявке, прикрепленными к ней в ходе исполнения файлами, созданными в случае необходимости подзадачами. Журнал заявок, отфильтрованный и отсортированный в различных вариантах, также является результатом работы системы. Электронные заявки не имеют печатной формы в виду отсутствия необходимости в ней.

Экранные формы всех объектов разрабатываемой системы будут построены по технологии «Управляемые формы», предоставляемой разработчику платформой «1С:Предприятие 8.3». Отличительной особенностью этой технологии является декларативное, а не «по пикселям» размещение элементов управления на форме и особенный, узнаваемый интерфейс.

В разрабатываемой системе не будут использоваться классификаторы.

Информационная база разрабатываемой системы будет построена как интегрированная база данных на основе СУБД MS SQL Server 2013.

1.4.2 Обоснование проектных решений по программному обеспечению

Разрабатываемая система предназначена для работы в операционной системе Windows. Программа должна работать в любой операционной системе семейства Windows, а также на серверных вариантах этой операционной системы. Выбор операционной системы объясняется развитым удобным графическим пользовательским интерфейсом данного семейства операционных систем, а также ее

популярностью на российском рынке. Многие фирмы-производители программного обеспечения, в том числе систем разработки приложений, также ориентированы на это семейство операционных систем, что привело к наличию на рынке большого количества систем программирования, позволяющих разрабатывать программное обеспечение для Windows. Эта операционная система имеет очень большой набор утилит и служебных программ, предназначенных для администрирования, управления и диагностики системы, а также прикладных приложений. Кроме того, она поддерживает обширный программный интерфейс для написания собственных приложений для работы в этой операционной среде.

Так как разрабатываемая система является конфигурацией для 1С:Предприятие 8.3 то данная система становится центром системы автоматизации приема и обработки заявок.

Система разрабатывается для клиент-серверного варианта работы, работающего в локальной сети предприятия. Выделение в локальной сети сервера становится обязательным. На этом сервере должна быть обязательно установлена серверная операционная система. На клиентских машинах ничего кроме операционной системы и системы «1С:Предприятие 8.3» устанавливать не нужно [2].

В случае клиент-серверного варианта работы выделение в локальной сети сервера становится обязательным. На этом сервере должна быть обязательно установлена серверная операционная система и сервер системы «1С:Предприятие 8». Кроме этого, нужен еще один сервер – баз данных. Физически это может быть тот же самый сервер, на котором функционирует сервер системы, а может быть и отдельная машина. В любом случае на нем должен быть установлен сервер базы данных.

В качестве операционной системы, функционирующей на клиентских машинах в случае клиент-серверного варианта работы целесообразно выбрать операционную систему Windows.

В качестве серверной операционной системы, функционирующей на серверах системы 1С:Предприятие и на сервере баз данных (в случае выделения его в отдельное рабочее место) целесообразно использовать Windows Server 2013 Standard

Edition. В качестве сервера баз данных целесообразно использовать MS SQL Server 2015 Standard Edition.

В последнее время все большую популярность приобретают Unix-ориентированные операционные системы, в частности Linux. Серверная часть системы 1С:Предприятие 8.3 также способна функционировать в этой операционной системе, которая с легкостью может заменить на сервере Windows 2013 Server. В этом случае в качестве серверной СУБД целесообразно использовать не MS SQL Server, а PostgreSQL, которая является достойной свободной альтернативой коммерческому СУБД.

Пользователи КИС и сотрудники Службы могут подключаться к системе, запущенной в режиме веб-клиента, по сети Интернет через браузеры, установленные на их персональных или мобильных компьютерах. В этом случае взаимодействие пользователей с системой осуществляется по сети Интернет через любой браузер. Для того чтобы приложение могло работать в режиме веб-клиента оно должно быть опубликовано на каком-либо web-сервере, к которому можно обратиться из сети Интернет по IP-адресу. В качестве программного web-сервера целесообразно использовать web-сервер Apache, а в качестве физического сервера – тот же самый сервер, на котором размещается информационная база приложения. Таким образом, программное обеспечение системы дополняется web-сервером Apache. Необходимо обеспечить круглосуточное функционирование сервера и свободный доступ к нему из сети Интернет. Сервер должен иметь статический IP-адрес.

Схема взаимодействия информационного и программного обеспечения в варианте web-клиента приведена на Рисунке 1.19.

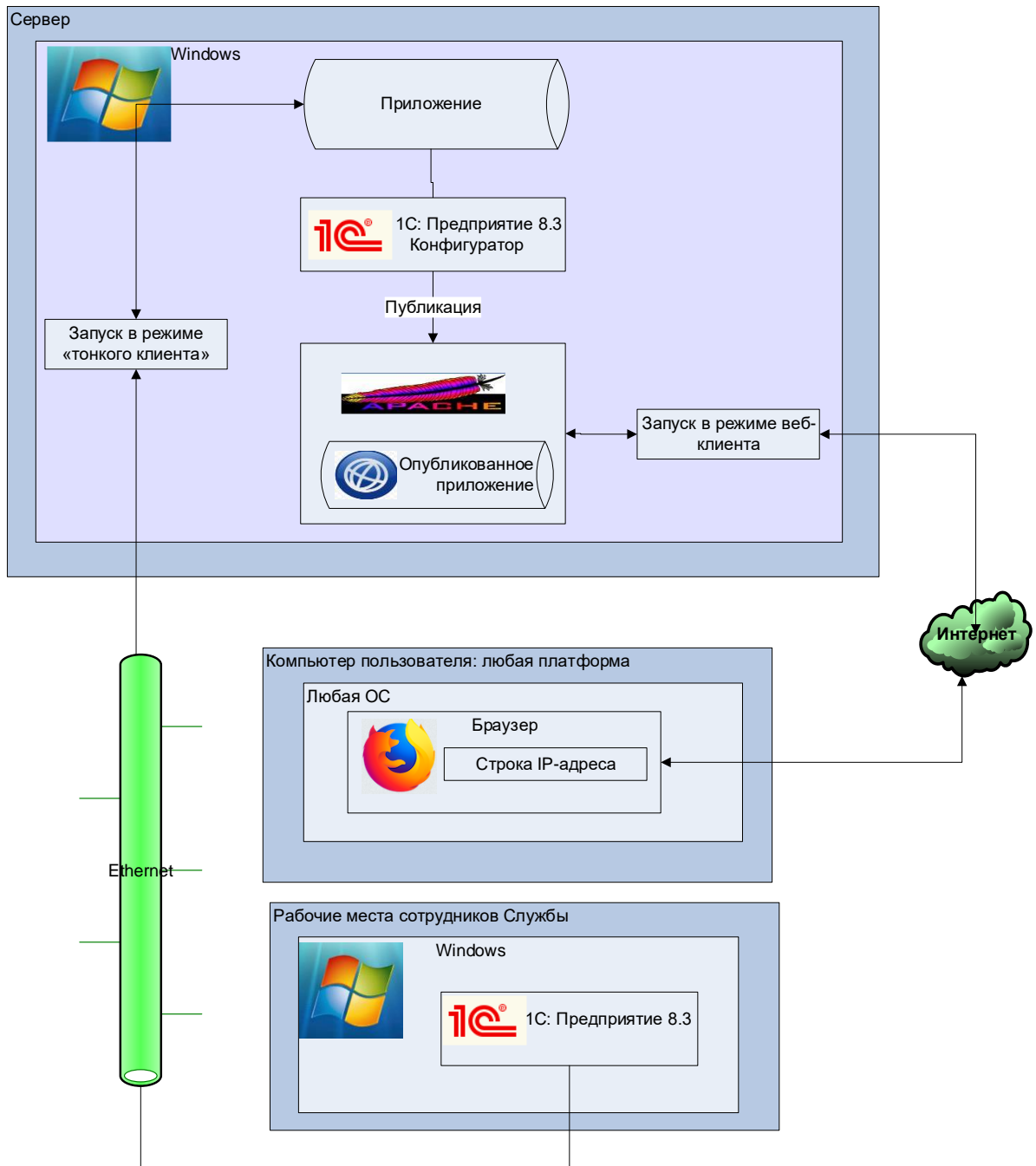


Рисунок 1.19 - Архитектура разрабатываемой системы в варианте web-клиента

1.4.3 Обоснование проектных решений по техническому обеспечению

Так как разрабатываемая система является конфигурацией для системы «1С:Предприятие 8.3» то данная система становится центром информационно-программного комплекса автоматизации работы службы технической поддержки и определяет требования к составу и структуре технического обеспечения системы [7].

Техническое обеспечение разрабатываемого проекта делится на две части: серверную и клиентскую.

На серверной части экземпляр системы «1С: Предприятие 8.3» и встроенная в платформу ««1С: Предприятие 8.3» СУБД. Аппаратное обеспечение сервера должно отвечать следующим характеристикам:

- Процессор Intel Pentium Dual Core 2,4 ГГц и выше;
- Оперативная память в размере 2 Гб;
- Жесткий диск не менее 500 Гб;
- Клавиатура, мышь, монитор.

В настоящее время на рабочих местах пользователей уже установлены компьютеры для работы других информационных систем. Все компьютеры имеют примерно одинаковые характеристики:

- процессор – Intel I3;
- оперативная память: 1024 Мб;
- SVGA 256 Мб;
- 24-х CD-ROM;
- жесткий диск, обеспечивающий размещение операционной системы, системы «1С:Предприятие 8.3» и хранение информационной базы, для этих целей достаточно 320 Гб.

Для печати отчетов и выходных документов используются принтеры, совместимые с компьютерами вышеперечисленной комплектации.

Все компьютеры, задействованные в системе, связаны между собой и с сервером по локальной сети, построенной на основе протокола TCP/IP. Аппаратное обеспечение локальной сети построено на базе коммутатора D_Link DES-1016, установленного в серверные и нескольких коммутаторы D_Link DES -1008, установленных на этажах предприятия.

Реализация данного дипломного проекта предусматривает максимальное применение уже используемых технических средств и адаптацию проектных решений к используемому техническому обеспечению. Это диктуется соображениями экономической целесообразности: ограничениями, накладываемыми текущим финансированием, выделяемым на обновление вычислительной и организационной техники отделов предприятия, техническим состоянием вычислительной техники, уровнем подготовки пользователей системы и

соответствием современным информационным технологиям. Схема технического обеспечения системы приведена на Рисунке 1.20.

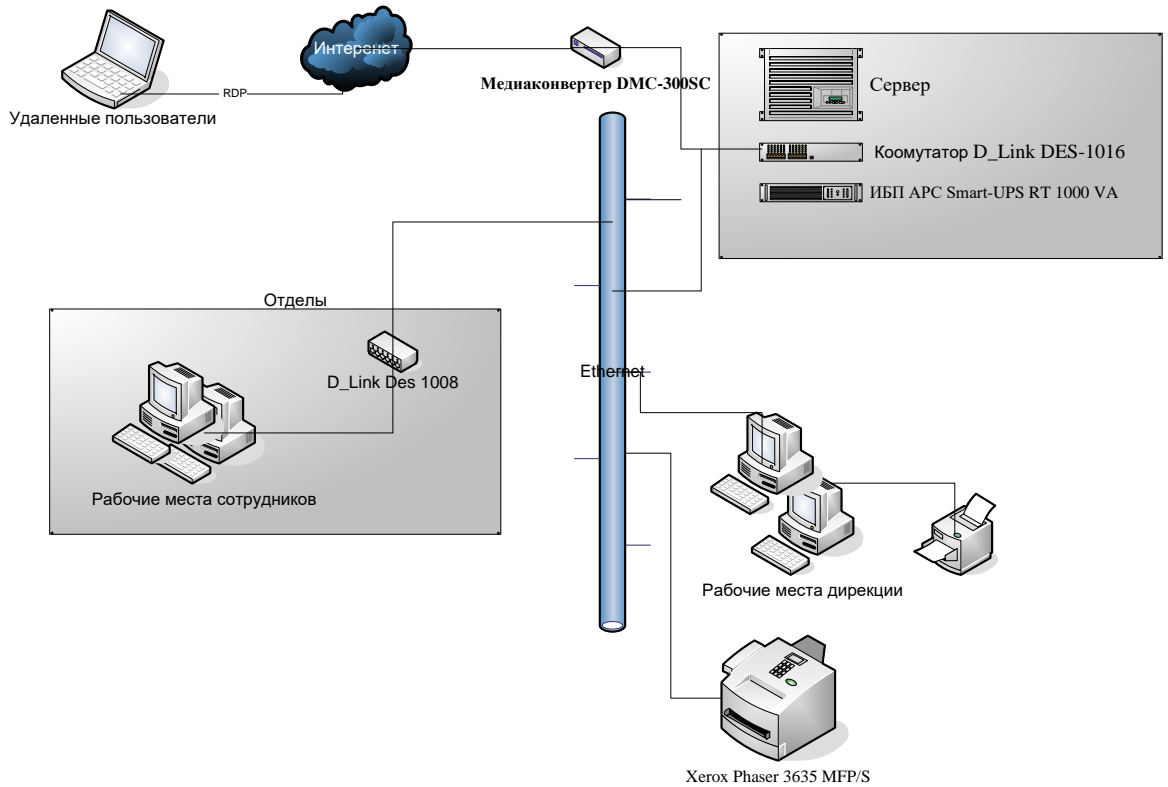


Рисунок 1.20. Схема технического обеспечения системы

2 Проектная часть

2.1 Разработка проекта автоматизации

2.1.1 Этапы жизненного цикла проекта автоматизации

Жизненный цикл (ЖЦ) разработанного ПО любой АИС является непрерывным процессом сопровождения, который начинается с принятия решения о необходимости разработки ПО и заканчивающийся при выводе разработанной ИС из эксплуатации.

Для построения и анализа ЖЦ ПО нередко используются различные модели. Под моделью ЖЦ ПО подразумевается специальная структура, задающая последовательность выполнения процессов и определяющая взаимосвязи между ними на протяжении всего ЖЦ. Наибольшее распространение в настоящее время получили следующие виды моделей: «каскадная», «модель с промежуточным контролем» и «спиральная модель».

Виды моделей каскадная и модель с промежуточным контролем включают этапы ЖЦ ПО:

- этап анализа;
- этап проектирования;
- этап реализации;
- этап внедрения;
- этап сопровождения.

Каскадная модель ЖЦ предполагает строго последовательное выполнение вышеперечисленных этапов. Достоинства каскадной модели: формирование на каждом из этапов законченного комплекта документации по этапу и возможность четкого планирования сроков этапа и затрат материальных и финансовых ресурсов. Недостаток данной модели: несоответствие реальному процессу разработки и создания ПО, который обычно не укладывается в такую жесткую схему и требует периодического возврата к предыдущим этапам для пересмотра или уточнения принятых решений.

Модель с промежуточным контролем более приближает ЖЦ к реальному процессу разработки и эксплуатации ПО. В противовес каскадной модели, она допускает возвращение из любого этапа ЖЦ на любой из предыдущих этапов для

выполнения необходимой корректировки. При этом создаются условия для разработки более надежного ПО, но вместе с тем увеличивается длительность и сложность процесса разработки.

Последняя модель из списка - спиральная позволяет устранить недостатки предыдущих. Основное внимание в ней уделяется на работы, выполняемые на начальных этапах: этапах анализа и проектирования. На этих этапах реализуемость проектируемых решений проверяется с помощью разработки специальных прототипов.

При применении спиральной схемы разработки ПО неполное завершение работ на каком-либо очередном этапе позволяет беспрепятственно переходить на последующий этап. Незавершенная работа или задача на предыдущем этапе может выполняться на следующем витке работы спирали. Т.о., обеспечивается возможность создать некоторый незавершенный, но работоспособный вариант разрабатываемого ПО для последующего уточнения требований.

В связи с небольшой сложностью предстоящей разработки будет использована каскадная модель ЖЦ разработки и сопровождения ПО. Использование этой модели оправдано, так как позволит создавать на каждом этапе законченный комплект необходимой документации и планировать сроки завершения всего проекта целиком и соответствующие затраты на каждом этапе и в целом по проекту.

Каскадная модель предполагает исключительно последовательную реализацию этапов ЖЦ. Достоинства каскадной модели: формирование на каждом этапе законченного комплекта документации по разрабатываемому ПО и возможность планирования сроков работ и затрат. Недостаток: некоторое несоответствие реальному процессу написания ПО, который обычно не вписывается в предписанную жесткую схему и требует частого возврата к предыдущим этапам для уточнения принятых решений.

На этапе анализа необходимо собрать информацию по документообороту предприятия.

На этапе проектирования происходит проектирование БД и структуры разрабатываемой АИС.

На этапе реализации программисты создают спроектированную АИС.

На этапе внедрения происходит развертывание технических и программных средств и инсталляция разработанной АИС.

На этапе эксплуатации конечные пользователи работают с АИС и ведут с ее помощью электронный документооборот.

Существует целый ряд стандартов, регламентирующих ЖЦ ИС, а в некоторых случаях и процессы разработки.

Среди известных стандартов можно выделить: «ГОСТ 34», «ISO 12207», «ISO 15288», «MSF», «RUP», «COBIT», «Oracle CDM», «XP».

В ВКР будет использован стандарт «ISO 15288». Использование данного стандарта позволит формировать на каждом этапе конечный комплект документации и планировать сроки завершения каждого этапа и проекта целиком и соответствующие затраты. Стандарт легко применим для сопровождения любого рода и класса систем, но его основное предназначение – сопровождение процесса разработки именно компьютеризированных систем.

В стандарте «ISO»/«IEC 15288» предусмотрены следующие этапы разработки систем:

- формирование концепции - анализ необходимых потребностей, выбор концепции разработки и проектных решений;
- разработка - проектирование будущей ИС;
- реализация – непосредственно изготовление ИС;
- эксплуатация - ввод ИС в эксплуатацию и ее использование;
- поддержка - обеспечение функционирования ИС в процессе работы;
- снятие с эксплуатации - прекращение использования ИС, ее демонтаж и архивирование.

На этапе «Формирование концепции» необходимо собрать информацию по деятельности предприятия. Сотрудники предприятия должны собрать и проанализировать данные необходимые им в программе по автоматизации документооборота. Это данные по номенклатуре, контрагентам, видам документам и их движениям.

На этапе «Разработка» по результатам представленной информации происходит проектирование программистами БД и структуры программы.

На этапе «Реализация» программисты создают БД: создают все необходимые справочники документы, регистры. Затем производят настройку главного меню (ГМ) и меню всех элементов АИС.

Этап «Эксплуатация» включает в себя следующие работы:

- Развертывание сервера БД, установка на нем ОС, а именно «Windows Server 2008» или более поздней версии.
- Развертывание рабочих ПК, установка на них ОС «Windows 10» и инсталляция клиентской части АИС.

Стратегия внедрения, выбранная на данном этапе – пилотный проект. Предполагается внедрение системы в учреждении, ввод всех необходимых исходных данных. Далее предполагается фиксировать в АИС документы по движениям ТМЦ. В процессе исполнения пилотного проекта АИС пройдет обкатку, возможно, будут выявлены недостатки, после устранения которых можно будет разворачивать АИС в остальных подразделениях организации.

После окончания процесса развертывания технических, информационных и программных компонентов необходимо провести окончательное тестирование АИС и убедиться в работоспособности всех модулей АИС.

На стадии эксплуатации нужно выполнить обучение операторов работе с АИС.

На первой стадии эксплуатации ИС нужно внести все первичные данные:

- данные по номенклатуре предоставляемых услуг и реализуемых ТМЦ;
- данные пациентов и контрагентов;
- данные по учреждению;
- и т.д.

Этапы «Поддержка» и «Снятие с эксплуатации» в данной работе не рассматриваются.

2.1.2 Ожидаемые риски на этапах жизненного цикла и их описание

На этапе анализа обязательно необходимо точно определить все виды информации, которая может поступать в разрабатываемую АИС документооборота. Если на этом этапе будет пропущена какая-либо информация, то не будет реализована возможность хранения и обработки её в АИС. Для уменьшения риска

упущения какой-либо информации необходима перекрестная проверка между различными подразделениями учреждения, участниками разрабатываемой АИС.

На этапе проектирования нужно детально изучить сделанный анализ информации и транслировать его на структуры БД и программные модули.

На этапе реализации необходимо предотвратить возможность пользователя совершать ошибочные действия, которые могут повлечь крах АИС или ввод некорректных данных. Для уменьшения такого риска необходимо осуществлять тестирование весьма большим количеством пользователей.

При внедрении нужно проверить наличие необходимого ПО и лицензий к нему.

На этапе эксплуатации нужно обеспечить правильное обучение персонала и для уменьшения риска провести контрольную проверку по изученной АИС.

2.1.3 Организационно-правовые и программно-аппаратные средства обеспечения информационной безопасности и защиты информации

При использовании любого ПП необходимо следовать нескольким важным правилам:

- использовать только лицензионное ПО;
- следить за выходом пакетов обновлений «общесистемного» и «специального» ПО и вовремя устанавливать их;
- использовать только проверенное ПО, приобретенное у официальных поставщиков.

Для обеспечения информационной безопасности необходимо разграничить доступ пользователей к данным на нескольких уровнях:

- на уровне ОС ограничить доступ пользователя к сетевым ресурсам, в зависимости от его прав;
- ограничить доступ пользователей к ресурсам сети «Интернет»;
- настроить права пользователя в зависимости от должностной инструкции сотрудника.

В разрабатываемой системе права пользователя на редактирование документов «Заявка» будут определяться его положением в иерархическом справочнике «Сотрудники». Полные права на редактирование документа есть

только у автора документа – штатного сотрудника Службы, который создал этот документ «Задача». Права на редактирование «чужих» документов сотрудником, который в данный момент авторизовался в системе, определяются местоположением текущего сотрудника в иерархии Службы. Так для документов, авторами которых являются сотрудники, стоящие в иерархии ниже текущего (его подчиненные), у текущего сотрудника есть полные права на редактирование документа (в том числе на установку признаков исполнения), но только при условии, что у него установлено свойство «Право руководителя». А для документов, авторами которых являются сотрудники, стоящие в иерархии выше текущего (его начальники) у данного сотрудника есть только право проставить отметку об исполнении своей части задачи в списке исполнителей «чужого» документа «Задача». Такая гибкая настройка прав позволяет сотрудникам редактировать документы только в пределах своей компетенции.

Для предотвращения потери и порчи информации необходимо также установить и настроить антивирусную систему, с постоянным обновлением баз данных вирусов, а также установить и настроить систему «Firewall» для закрытия угроз из внешней сети, например «Интернет».

Что касается политики безопасности, то все организационные меры защиты информации, которые существовали в учреждении до внедрения новой АИС, продолжают свое существование.

К внешним угрозам можно отнести взлом БД, вирусные атаки, шпионское ПО, физическая порча оборудования, а также угрозы, поступающие из внешней среды организации.

Для обеспечения информационной безопасности и защиты информации от внешних угроз, в учреждении уже была установлена антивирусная система «Symantec Endpoint Security», которая предусматривает защиту по всем факторам программных внешних угроз. Данной антивирусной программы достаточно для обеспечения безопасности такого уровня, который необходим для данного предприятия. К аппаратным средствам защиты информации от внешних угроз после внедрения АИС относятся те же средства, что и до внедрения, а именно: блок «бесперебойного» питания сервера, который предохранит не только от потери информации, но и от порчи оборудования

К защите информации также стоит отнести резервное копирование данных, которое должно проводиться администратором раз в месяц.

В разрабатываемой АИС предусмотрена защита от удаления информации, на случай невнимательности пользователей. Прежде чем удалить информацию из БД, АИС выдает сообщение о подтверждении удаления, и только в случае согласия пользователя, происходит удаление записи.

2.2 Информационное обеспечение задачи

2.2.1 Информационная модель и её описание

В результате проектирования была разработана информационная модель системы, которая представляет из себя модель организации работы системы и схематично поясняет, на основании каких входных документов, а также нормативно-справочной информации происходит функционирование системы и получение выходных данных, т.е. процесс преобразования данных в информационной системе. Схема представлена на Рисунке 2.1.

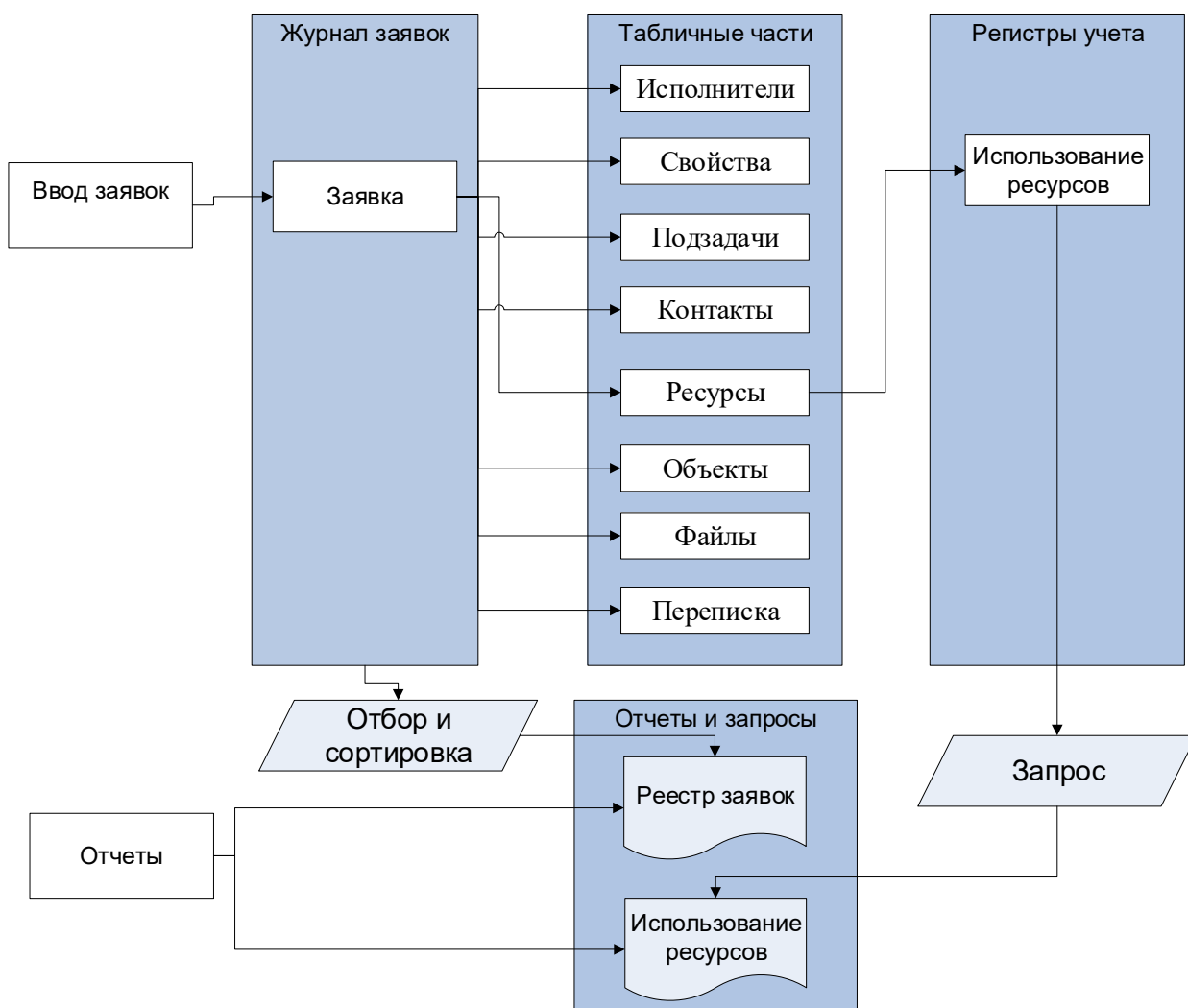
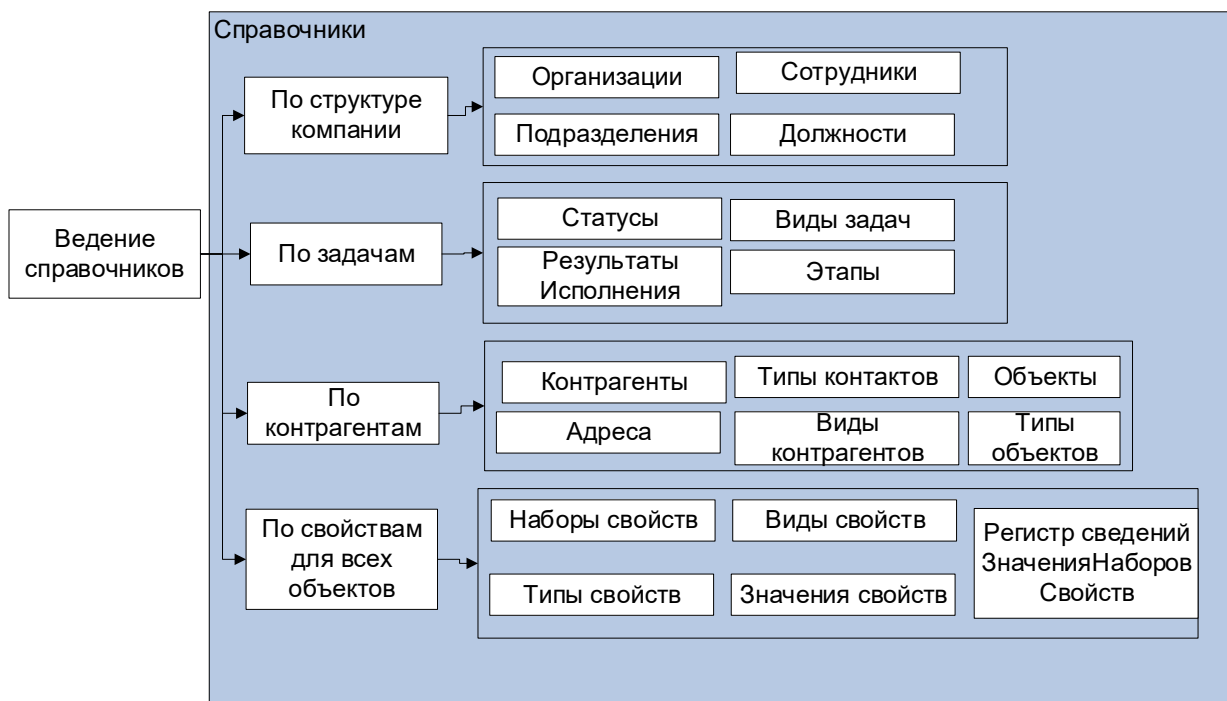


Рисунок 2.1. Информационная модель

2.2.2 Характеристика нормативно-справочной, входной и оперативной информации

Основной компонентой немашинного информационного обеспечения является система документации. Документ – определенная совокупность сведений, используемая при решении экономических задач, расположенная на материальном носителе в соответствии с установленной формой. Первичные документы предназначены для отражения процессов в материальной сфере и поставляют всю постоянную и оперативную информацию, необходимую для решения экономических задач. Вся первичная информация находится на бумажных носителях [10].

Под входной информацией понимается вся информация, необходимая для решения задачи и расположенная на различных носителях: первичных документах, машинных носителях, в памяти персонального компьютера.

От рациональной организации входной информации предприятия, способов сбора, регистрации, передачи, хранения и обработки информации, ее состава и своевременного получения зависят оперативность и эффективность управления.

Входной информацией для разрабатываемой в ВКР автоматизированной системы являются условно-постоянные данные, хранящиеся в справочниках и регистрах сведений, представленных в Таблице 2.1:

Таблица 2.1. Данные по справочникам

Название справочника	Ответственный	Средний объём	Частота актуализации	Средний объём актуализации
«Организации»	Администратор системы	1	низкая	1
«Подразделения»	Администратор системы	1	низкая	1
«Сотрудники»	Администратор системы	2	низкая	2
«Должности»	Администратор системы	3	средняя	3
«Статусы»	Оператор	50	высокая	100
«Результаты исполнения»	Оператор	70	высокая	100
«Виды задач»	Администратор системы	1	низкая	1

«Этапы»	Администратор системы	1	низкая	1
«Контрагенты»	Оператор	80	высокая	100
«Адреса»	Оператор	80	высокая	100
«Типы контактов»	Администратор системы	1	низкая	1
«Виды контактов»	Администратор системы	1	низкая	1
«Объекты»	Оператор	50	высокая	100
«Типы объектов»	Администратор системы	1	низкая	
«Наборы свойств»	Оператор	20	средняя	100
«Виды свойств»	Администратор системы	5	низкая	1
«Типы свойств»	Администратор системы	5	низкая	1
«Значения свойств»	Оператор	50	высокая	100
«Значения наборов свойств»	Оператор	50	высокая	100

В автоматизированном рабочем месте службы технической поддержки, выполняющей функции системы обработки заявок, присутствует только один документ «Задача», который благодаря различным реквизитам может иметь различные смысловые нагрузки. В документе присутствуют табличные части: «Исполнители», «Свойства», «Контакты», «Подзадачи», «Ресурсы», «Объекты», «Файлы», «Переписка». Табличная часть «Исполнители» содержит данные по исполнителям текущей задачи, проценту их участия, дате ознакомления исполнителя с задачей, дате исполнения задачи. Табличная часть «Контакты» содержит данные по контактам для данной задачи. Табличная часть «Подзадачи» содержит данные по подзадачам, которые необходимо выполнить для выполнения текущей задачи, т.е. табличная часть содержит ссылки на другие документы «Задача». Табличная часть «Ресурсы» содержит список номенклатуры ресурсов, которые понадобятся для выполнения данной задачи. Табличная часть «Файлы» содержит в себе список дополнительной информации по задаче, которая хранится в виде отдельных файлов. Табличная часть «Переписка» хранит в себе сообщения пользователей-участников данной задачи друг другу, представляя собой диалог участников.

Частота возникновения документов зависит от объема задач службы и может варьироваться от нескольких в неделю до нескольких десятков в день (см. Таблицу 2.2)

Таблица 2.2. Данные по документам

Название документа	Ответственный	Средний объём	Частота актуализации	Средний объём актуализации
Задача	Сотрудник Службы	10	высокая	300

При проведении документа «Задача» он выполняет движение по регистру накопления «Использование ресурсов», в котором накапливаются данные по перечню наименований и количеству ресурсов, требуемых для выполнения заявок, если в табличной части «Ресурсы» документа «Задача» введены какие-то данные. Частота движений по этому регистру зависит от количества документов «Задача», в которых эта табличная часть заполняется (см. Таблицу 2.3).

Таблица 2.3. Данные по регистрам накопления

Название регистра	Ответственный	Средний объём	Частота актуализации	Средний объём актуализации
Использование ресурсов	Сотрудник Службы	10	высокая	150

2.2.3 Характеристика результатной информации

Результатная информация – это информация, прошедшая через блоки обработки АИС и поступающая на ее выход. В разрабатываемой АИС это журнал обработанных заявок пользователей на решение проблем, возникающих в ходе эксплуатации информационных ресурсов предприятия.

Благодаря широким возможностям платформы «1С:Предприятие 8.3» по настройке журнала, он становится основным инструментом работы с заявками. Возможности настройки настолько широки, что позволяют любым категориям пользователей настроить журнал под любые задачи, возникающие у них.

Одной из важных сервисных возможностей является поиск отбор и сортировка данных в журнале заявок (Рисунок 2.2)

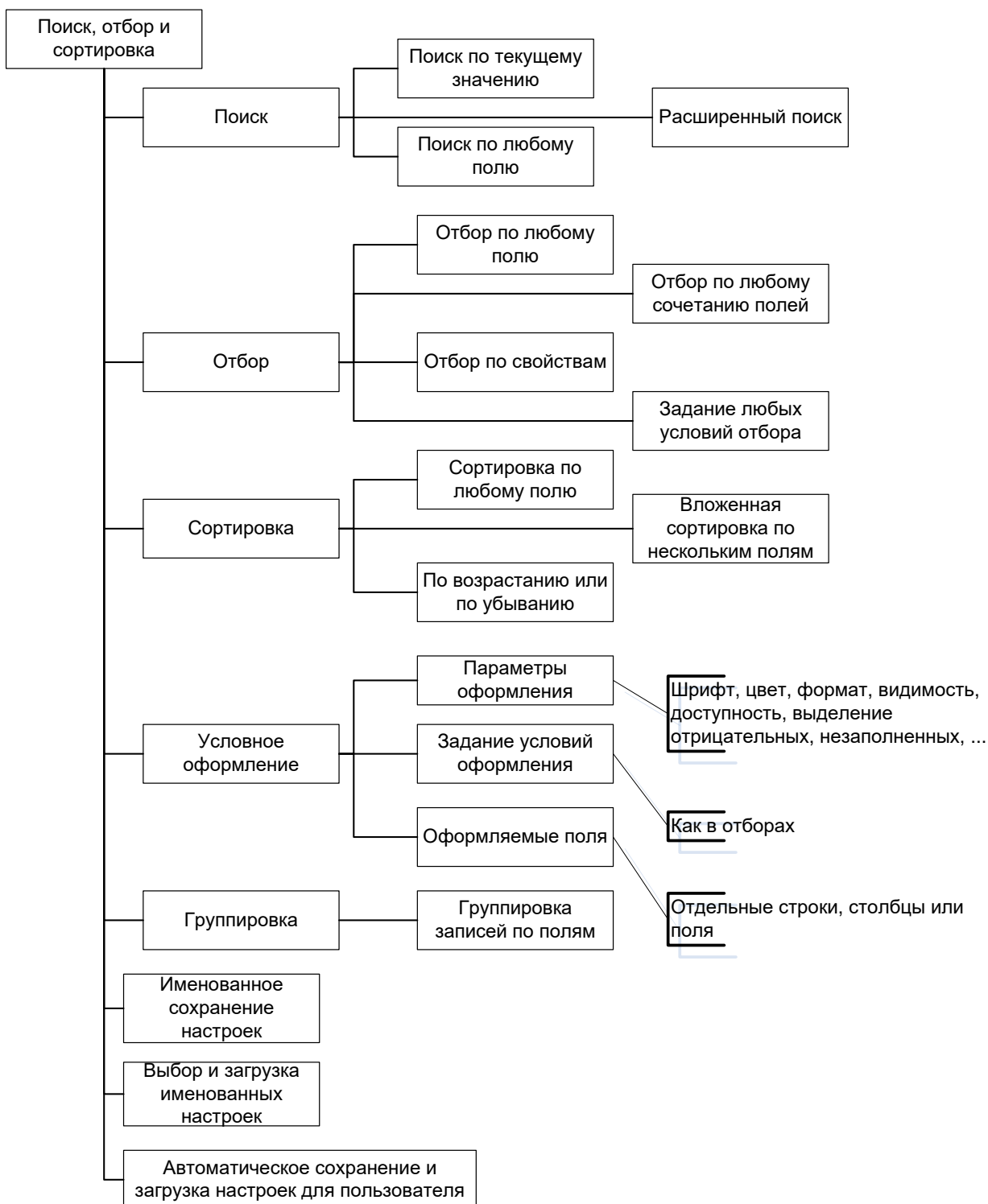


Рисунок 2.2. Возможности поиска, отбора и сортировки журнала заявок

Так, рядовой сотрудник может настроить журнал для поиска неисполненных задач, адресованных непосредственно ему (в которых он назначен исполнителем или ответственным). Это достигается установкой отбора, представленной на Рисунке 2.3.

Добавить новый элемент		Сгруппировать условия		Удалить	↑ ↓	Еще ▾
	Поле	Вид сравн...	Значение			
⊖ Отбор						
<input checked="" type="checkbox"/>	⊖ Группа Или					
<input checked="" type="checkbox"/>	Исполнители.Испол...	Равно	Смирнов Семен Валерьевич			
<input checked="" type="checkbox"/>	Ответственный	Равно	Смирнов Семен Валерьевич			
<input checked="" type="checkbox"/>	Проведен	Равно	Нет			

Рисунок 2.3. Настройка отбора неисполненных задач «по себе»

Результатом такого отбора будет список задач, адресованных данному сотруднику и ожидающих выполнения (Рисунок 2.4). Открыв любую задачу, сотрудник получает полную информацию по ней и своему заданию по данной задаче, если же этой информации недостаточно – он может перейти на задачу более высокого уровня и получить необходимую информацию там.

← →		☆ Задачи					Поиск (Ctrl+F)
Задача: создать на основании		Создать	↔	🔍	📄	📁	
Исполнители.Исполнитель:	<input checked="" type="checkbox"/>	Смирнов Семен Валерьевич	Проведен:	<input checked="" type="checkbox"/>	Нет		
Ответственный:	<input checked="" type="checkbox"/>	Смирнов Семен Валерьевич					
Номер	Дата	↓	Наименование	Статус			
ЦО-000005	19.04.18		Терминальный сервер бухгалтерии тормозит	Ожидает выполнения			
ЦО-000006	19.04.18		Не доходят письма по электронке	Новая			

Рисунок 2.4. Результат отбора неисполненных задач «по себе»

У начальника Службы может быть другая задача: вывести список неисполненных и просроченных задач, сгруппировав его по ответственным исполнителям. При этом красным цветом выделять задачи, находящиеся в статусе «Ожидает выполнения», а синим цветом и увеличенным шрифтом выделять задачи, находящиеся в статусе «Мешает выполнению других». Это достигается настройкой отбора (Рисунок 2.5), условного оформления (Рисунок 2.6) и группировок (Рисунок 2.7).

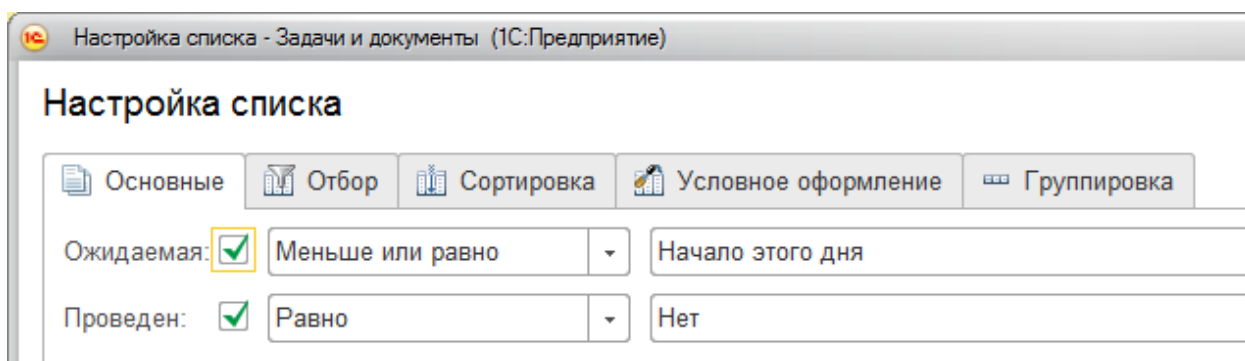


Рисунок 2.5. Настройка отбора

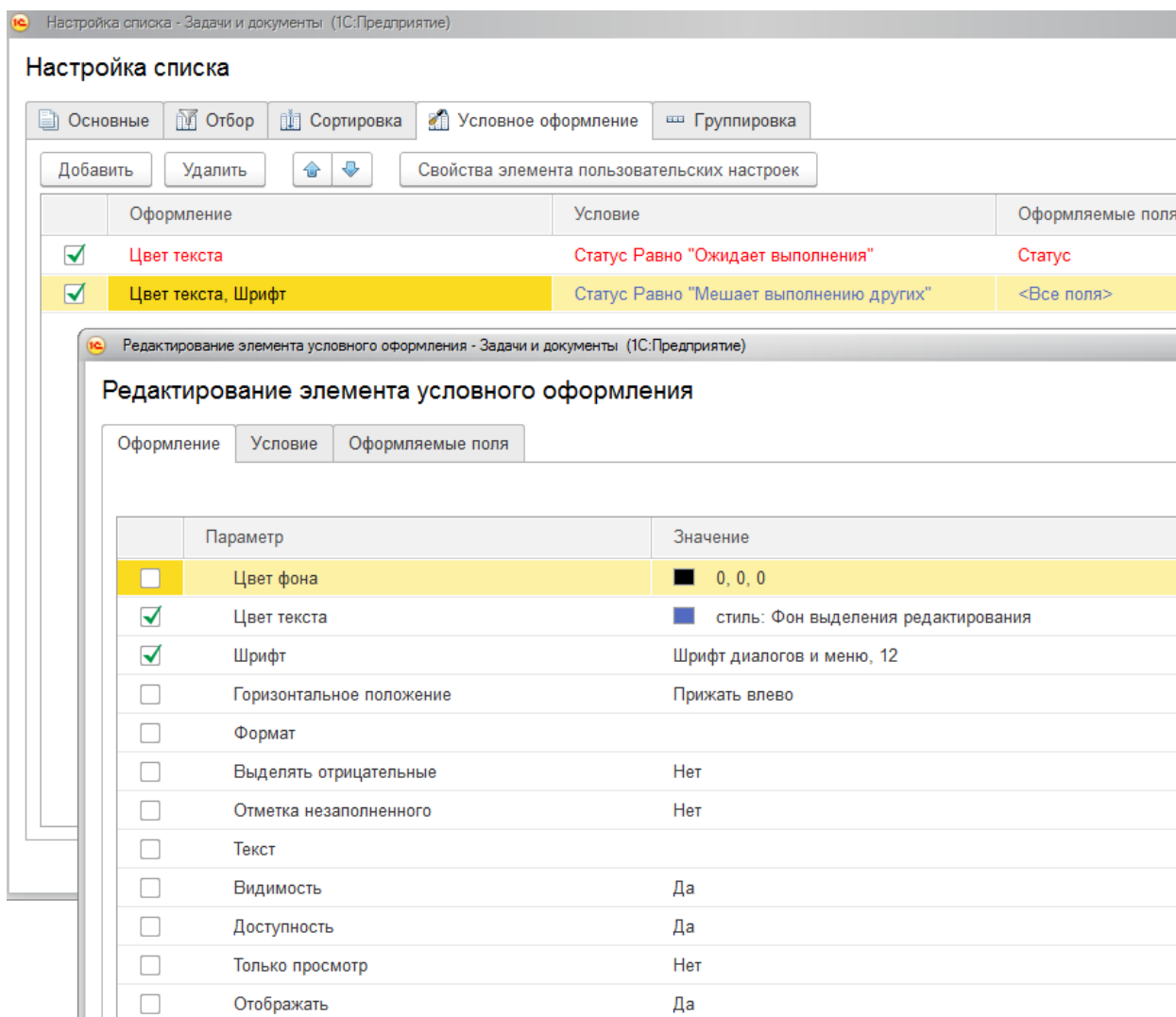


Рисунок 2.6. Настройка условного оформления

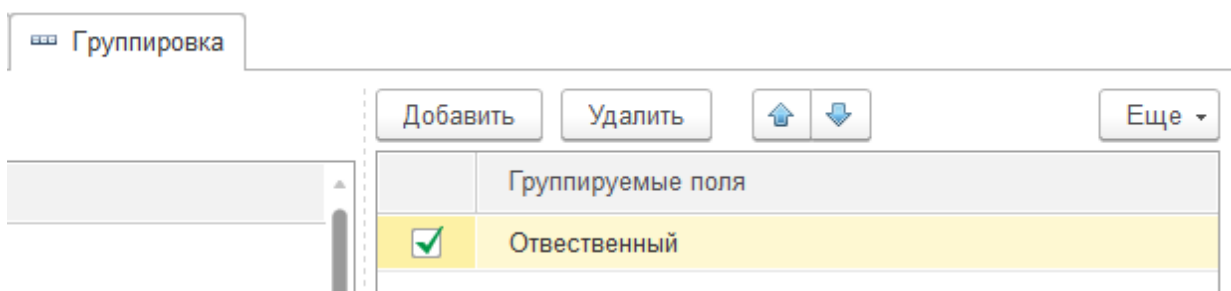


Рисунок 2.7. Настройка группировки

Результат применения данных настроек к списку задач представлена на Рисунке 2.8.

Номер	Дата ↓	Наименование	Статус	Ожидает...
Администратор Службы тех поддержки				
ЦО-000002	19.04.18	Модуль расчета отпускных	Новая	19.04.18
Багратионов Илья Борисович				
ЦО-000001	19.04.18	Мышка на компе бухгалтера по взаиморасчет...	Новая	19.04.18
Баранов Виктор Сергеевич				
ЦО-000003	19.04.18	Расчет ранее использованного отпуска - не ра...	Новая	19.04.18
Едалов Сергей Борисович				
ЦО-000004	19.04.18	Устранить программную ошибку в модуле рас...	Новая	19.04.18
Коршунов Михаил Сергеевич				
ЦО-000007	19.04.18	Купить мышку	Мешает выполне...	19.04.18
Смирнов Семен Валерьевич				
ЦО-000005	19.04.18	Терминальный сервер бухгалтерии тормозит	Ожидает выполнения	19.04.18
ЦО-000006	19.04.18	Не доходят письма по электронке	Новая	19.04.18

Рисунок 2.8. Результат отбора неисполненных и просроченных задач

Таким образом, журнал задач предоставляет широкие возможности по настройке и отбору списка задач по любым условиям, превращая его в основной инструмент работы с задачами для всех категорий сотрудников – руководителей и администраторов Службы, непосредственных исполнителей, пользователей, поставивших проблемы.

Помимо отбора и сортировки важной сервисной функцией является настройка внешнего вида формы и выводимых в нее элементов. Эта настройка позволяет настроить форму под любые потребности пользователя (Рисунок 2.9).

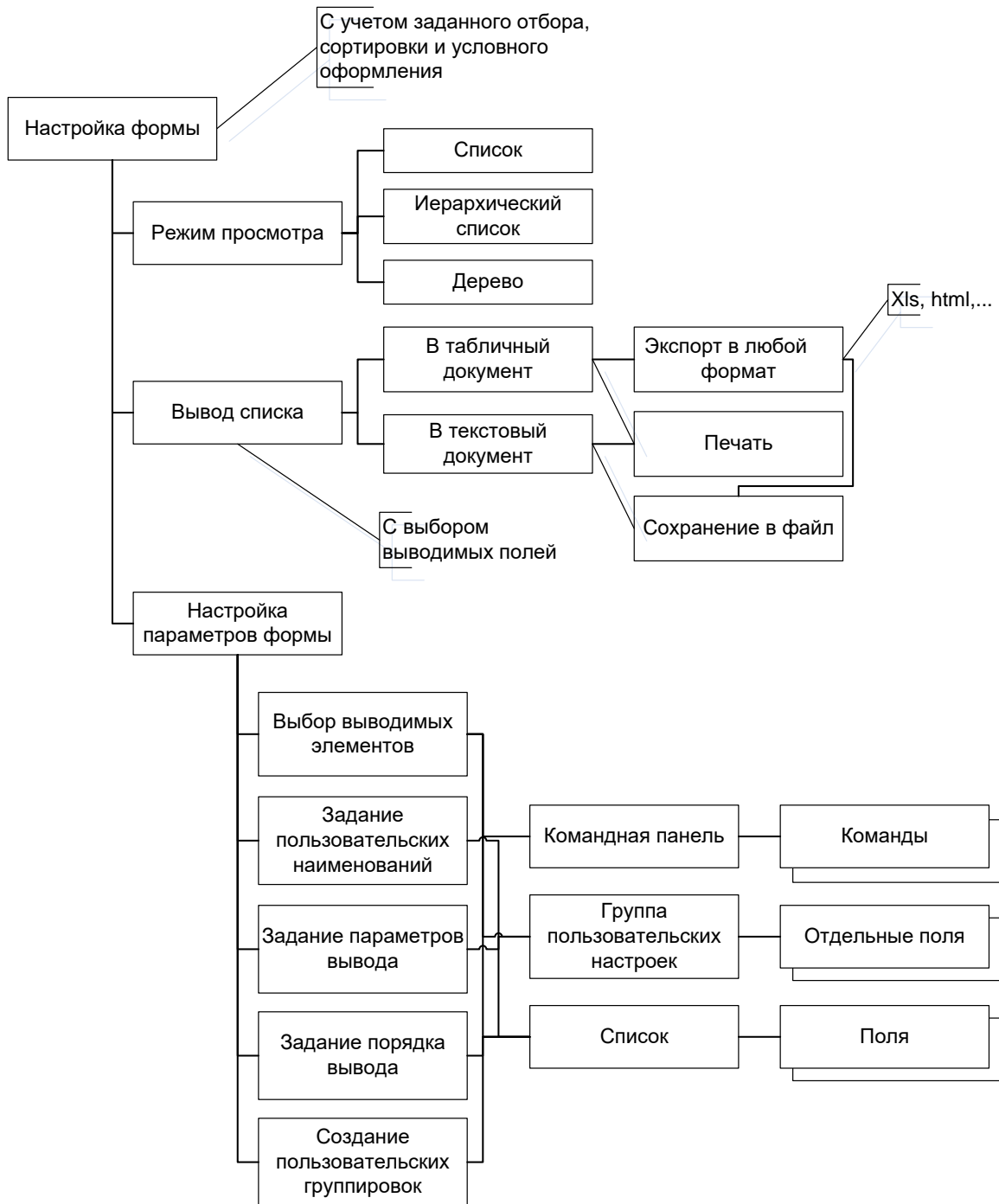


Рисунок 2.9. Функции настройки формы

2.3 Программное обеспечение задачи

2.3.1 Общие положения (дерево функций и сценарий диалога)

Общая функциональная структура системы представлена на Рисунке 2.10.



Рисунок 2.10. Функциональная структура системы

Подсистема ведения справочников предназначена для ввода в систему условно-постоянной информации, которая понадобится на следующих этапах работы с системой.

Подсистема фиксации заявок должна предоставлять возможность ввода в формализованном виде заявок пользователей, назначения им реквизитов, в том числе списочного характера, максимально полно описывающих заявки.

Подсистема автоматической генерации подзадач по сопровождению заявки должна позволять на основании шаблонов, задающих различные сценарии прохождения этапов решения типовой проблемы сгенерировать документы «Задача», адресовав их конкретным сотрудникам (или подразделениям) Службы (назначив их исполнителями).

Подсистема сопровождения задач должна предоставлять возможность сопровождения задачи по всему ее жизненному циклу.

Подсистема выполнения задач позволяет разделить задачи на полностью выполненные (закрытые) и находящиеся в работе, на каком-либо из этапов.

Подсистема разграничения прав пользователей должна предоставлять гибкую возможность настройки прав пользователей на ввод и редактирование задач и их параметров в зависимости от положения пользователя в иерархии Службы.

Работа в режиме «Управляемое приложение» изменяет прежде всего всю концепцию построения пользовательского интерфейса. Интерфейс «Управляемого приложения» кардинально отличается от интерфейса «обычного» приложения, созданного даже в той же версии платформы «1С:Предприятие 8.3». Новая концепция пользовательского интерфейса ориентирована на максимально комфортную и особенно эффективную работу и соответствует современным тенденциям [5].

Опишем основные возможности новой концепции интерфейса, которые мы использовали при разработке нашей конфигурации:

- Основное окно и панели;
- Командный интерфейс;
- Управляемые формы.

Основное окно представляет доступ ко всем элементам программы. Функции, необходимые для удобной навигации по прикладному решению, реализованы в нескольких вспомогательных панелях: разделов, функций текущего раздела, инструментов, избранного, истории и др. Разработчик прикладного решения может задать некоторый стандартный состав и расположение этих панелей в соответствии с назначением и особенностями приложения, но пользователь может по своему усмотрению менять расположение панелей, скрывать и раскрывать панели и настраивать свое рабочее пространство максимально комфортно и эффективно. На Рисунке 2.11. представлен один из вариантов настройки и расположения панелей, сконструированный пользователем по своему вкусу.

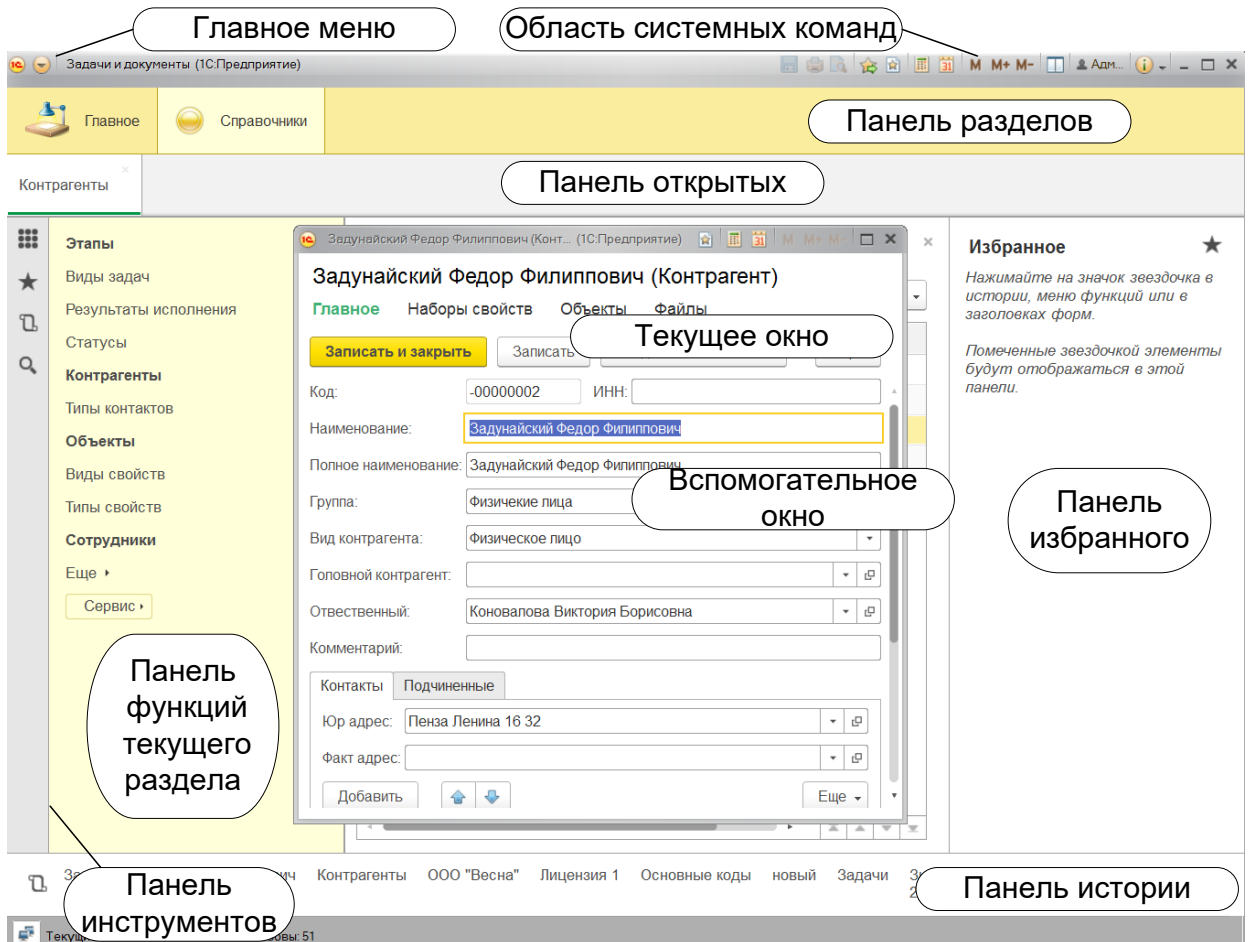


Рисунок 2.11. Панели основного окна

Командный интерфейс – это основное средство декларативного построения интерфейса пользователя и основное средство навигации пользователя по функциональности конфигурации. Командный интерфейс пронизывает все уровни построения интерфейса приложения: от подсистем до конкретных форм. Командный интерфейс строится платформой автоматически на основе декларативных включений разработчиком команд в тот или иной элемент командного интерфейса [6].

Командный интерфейс основного окна и разделов строится на основе подсистем, заданных в конфигураторе. Разработчик должен создать в конфигурации иерархию подсистем, отражающую для пользователя структуру функциональности прикладного решения. На основе этой информации при помощи редактора командного интерфейса конфигурации строится панель разделов основного окна (Рисунок 2.12). Все прикладные объекты конфигурации (справочники, документы, отчеты и т. д.) привязываются разработчиком к той или иной подсистеме или

одновременно к нескольким подсистемам [17]. На основе этой привязки платформа автоматически строит командный интерфейс разделов (Рисунок 2.13).

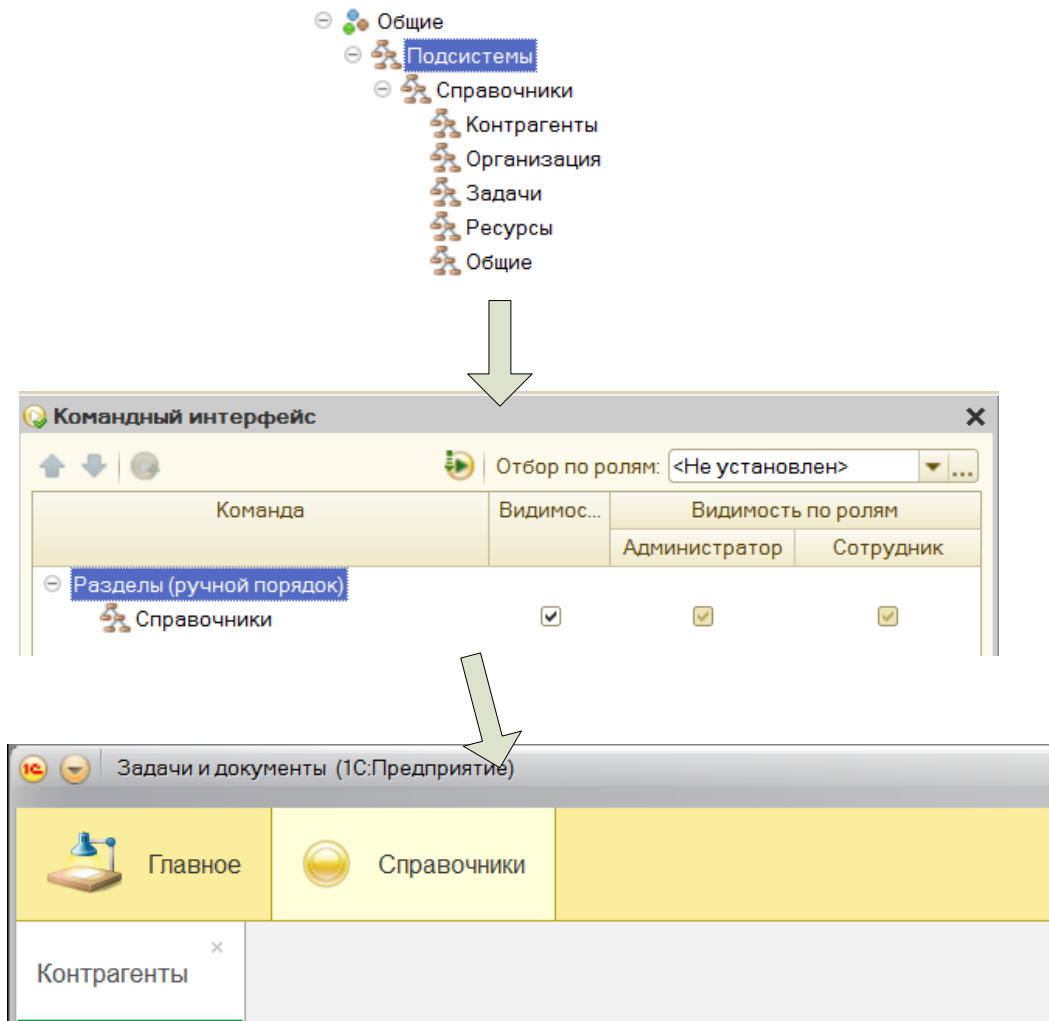


Рисунок 2.12. Создание разделов на основе подсистем

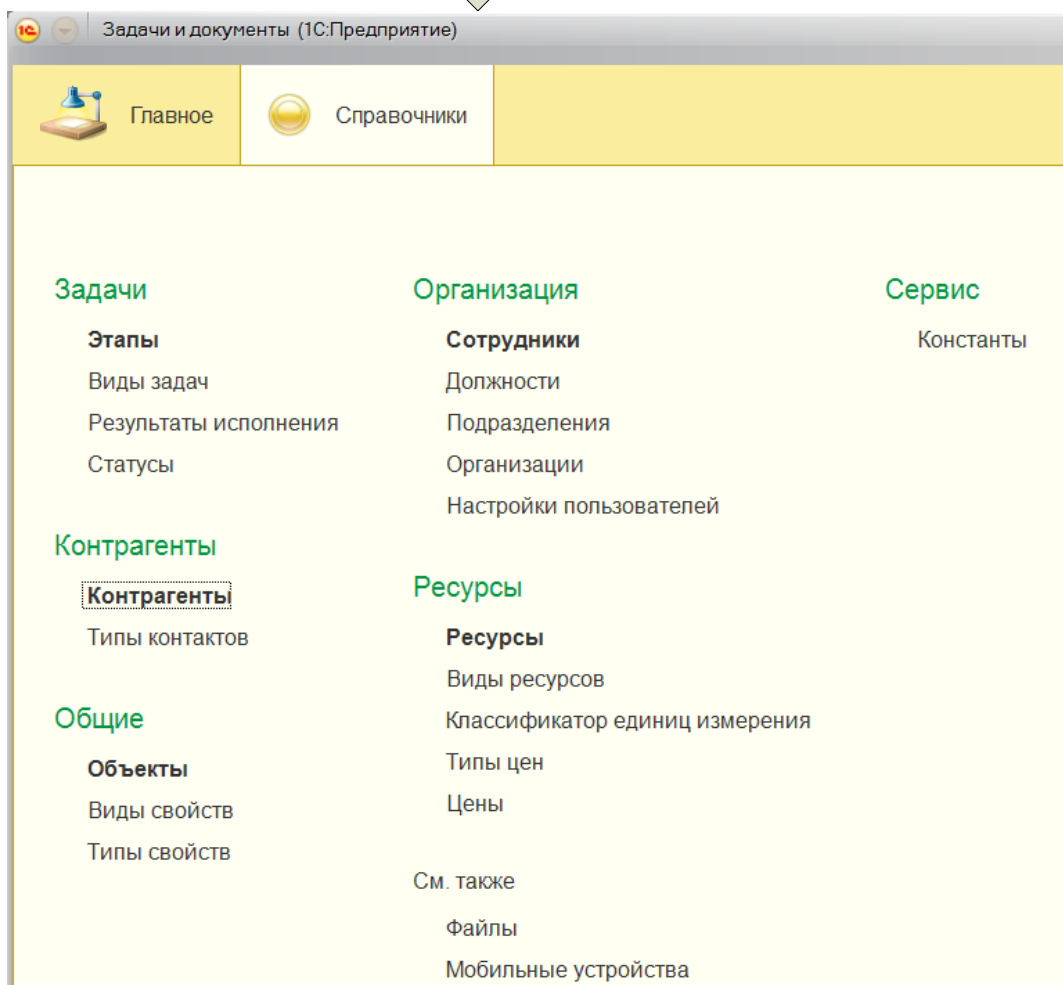
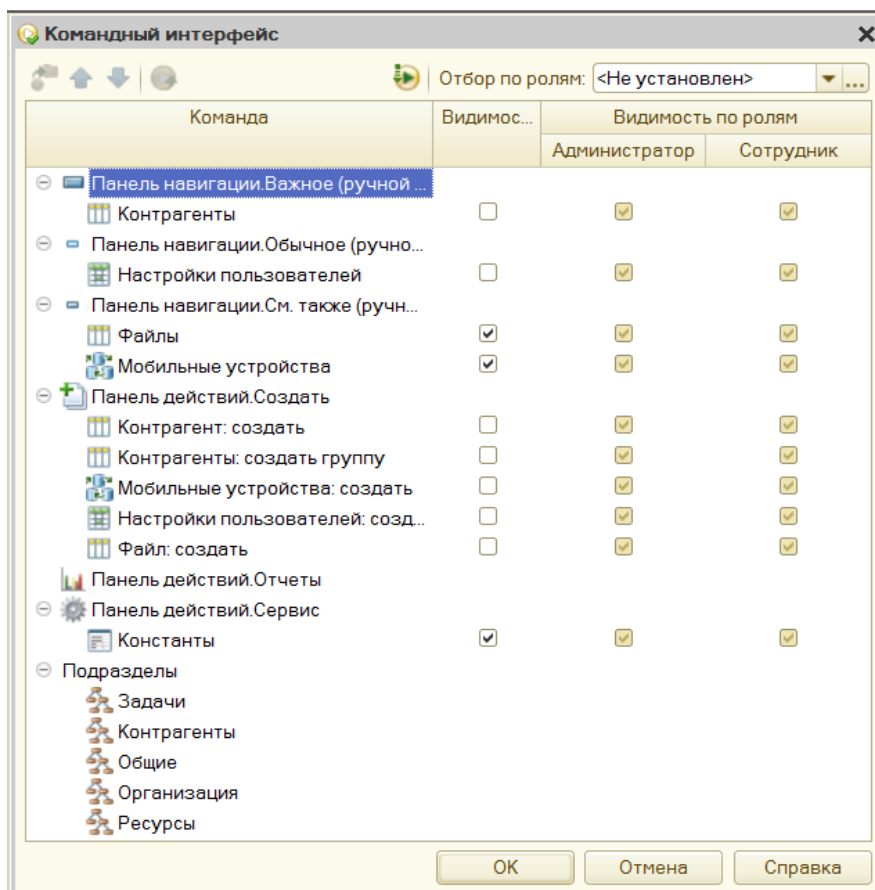


Рисунок 2.13. Создание командного интерфейса раздела (подсистемы)

Командный интерфейс форм задается также декларативно. Платформа автоматически включает в командный интерфейс формы прикладного объекта все связанные объекты (подчиненные справочники, регистры, по которым проводится движение и т.д.). Разработчик при помощи редактора командного интерфейса формы выбирает – какие команды включить в командный интерфейс.

В управляемом приложении используются формы, построенные на новой технологии. Они называются **Управляемые формы**. Основные отличия управляемых форм для разработчика:

- Декларативное, а не «по пикселям» описание структуры. Конкретное размещение элементов выполняется системой автоматически при отображении формы.
- Вся функциональность формы описывается в виде реквизитов и команд. Реквизиты – это данные, с которыми работает форма, а команды – выполняемые действия.
- Форма выполняется и на сервере, и на клиенте.
- В контексте клиента, недоступны практически все прикладные типы, и соответственно невозможно изменить данные в информационной базе.
- Для каждого метода или переменной формы обязательно должна быть указана директива компиляции, определяющая, место выполнения (клиент или сервер) и доступ к контексту формы.

Форма документа «Задача» предназначена для работы с одним выбранным документом. На форме содержится большое количество интерфейсных элементов, отображающих реквизиты шапки и табличных частей документа. Так как документ имеет много табличных частей, то отобразить их на одной форме будет проблематично. Для решения этой проблемы используются закладки: реквизиты шапки документа отображаются на общем поле документа, а каждая табличная часть – на отдельной закладке. Это позволяет разгрузить большой объем информации, содержащийся в табличных частях документа по визуальным разным закладкам и создать пользователю комфортный режим работы с документом, при этом основные параметры документа остаются видны на всех закладках. Закладки формы документа «Задача» представлены на Рисунке 2.14.

The screenshot shows a software interface for a 'Task' form. At the top, there is a toolbar with buttons: 'Провести и закрыть' (highlighted in yellow), 'Записать и закрыть', 'Задача: создать на основании', 'Записать', 'Закрыть', and 'Еще'. Below the toolbar are several input fields: 'Номер:', 'Дата:', 'Ожидаемая:', 'Дата выполнения:', 'Головная', and 'Выполнена'. The main area contains a large 'Наименование:' field, followed by 'Статус:', 'Автор:', 'Ответственный:', and 'Вид задачи:' on the left, and 'Фирма:', 'Контрагент:', 'Объем:', 'Ед изм:', and 'Тип цен:' on the right. At the bottom, there is a tabbed interface with the following tabs: 'Переписка' (highlighted with a blue border), 'Свойства', 'Дополнительно', 'Исполнители', 'Подзадачи', 'Контакты', 'Ресурсы', 'Объекты', and 'Файлы'.

Рисунок 2.14. Закладки формы документа «Задача»

Каждая закладка предназначена для работы со своей категорией данных, каждая из которых представляет из себя данные табличного вида. Все интерфейсные элементы, отражающие эти табличные данные имеют примерно одинаковый вид и функционал: табличное поле, в котором отображаются строки соответствующей табличной части и командная панель, позволяющая работать с этой табличной частью. Однако ввиду специфичности каждого набора данных, на каждой закладке предусмотрены свои интерфейсные элементы, а в программном модуле формы – свои процедуры и функции, обеспечивающие специфику отображения и редактирования данных именно этой табличной части.

Вся функциональность формы реализуется ее командами. Это могут быть стандартные команды формы, команды, созданные разработчиком, а также глобальные команды из командного интерфейса всей конфигурации. Кроме того, реализована возможность создания параметризуемых команд, которые будут открывать другие формы с учетом конкретных данных текущей формы [19]. На Рисунке 2.15 представлен процесс размещения в командной панели дополнительной команды в дополнение к стандартным.

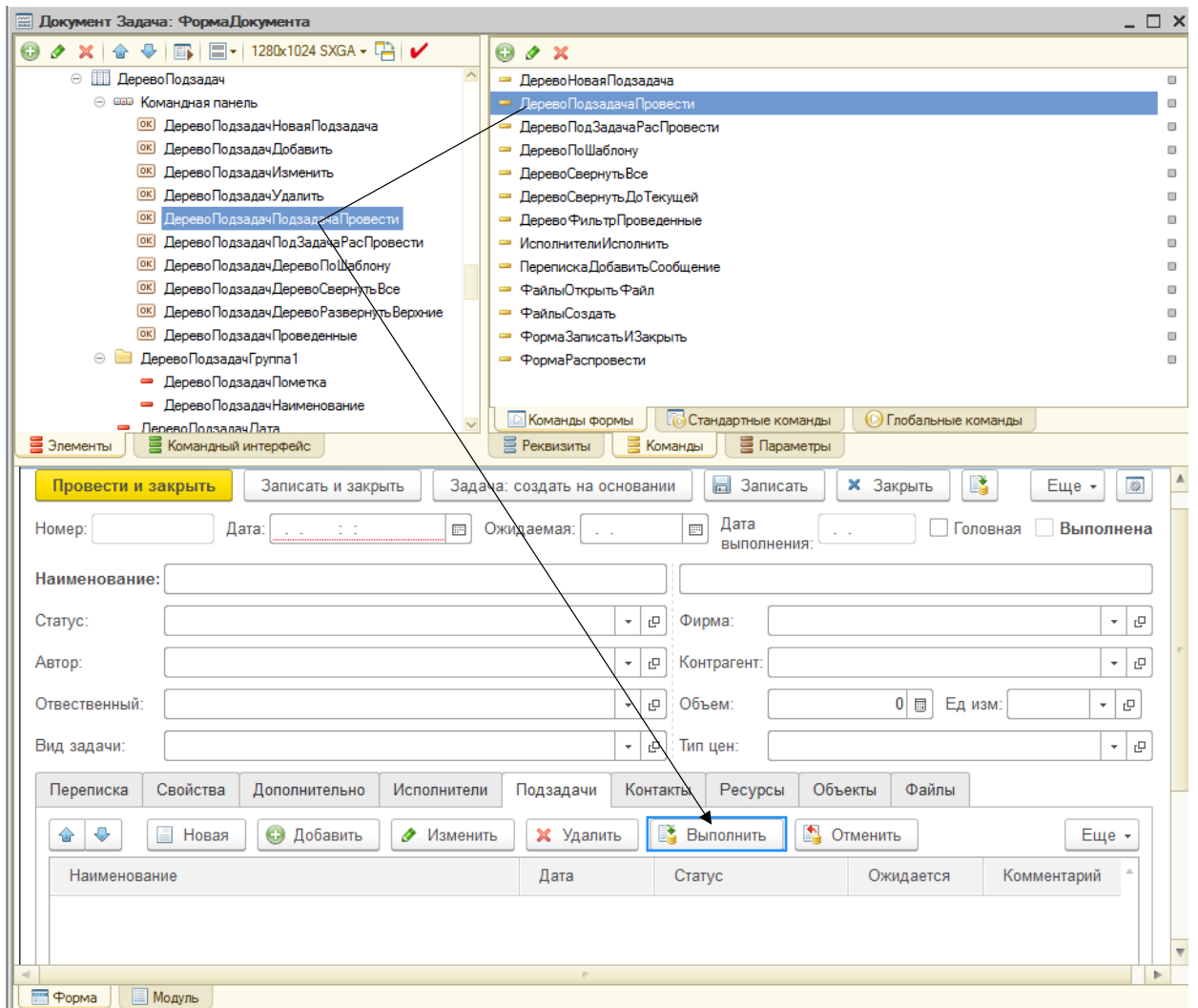


Рисунок 2.15. Размещение команд на форме

В результате мы получили форму документа «Задача», представленную на Рисунке 2.16.

← → ☆ **Задача ЦО-000002 от 19.04.2018 5:38:57**

Главное Использование ресурсов

Провести и закрыть Записать и закрыть Задача: создать на основании Записать Закрыть Показать в списке

Номер: ЦО-000002 Дата: 19.04.2018 5:38:57 Ожидаемая: 19.04.2018 Дата выполнения: . . . Главная Выполнена

Наименование: Модуль расчета отпускных Не работает расчет за использованный ранее отпуск

Статус: Новая Фирма: ООО "Континент"

Автор: Ивашкина Антонина Сергеевна Контрагент: Ивашкина Антонина Сергеевна

Отвественный: Администратор Службы тех поддержки Объем: 0,000 Ед изм: . . .

Вид задачи: Настройка модуля КИС

Переписка Свойства Дополнительно Исполнители Подзадачи Контакты Ресурсы Объекты Файлы

Новая Добавить Изменить Удалить Выполнить Отменить По шаблону Все

Наименование	Дата	Статус	Ожидается
ЭТА ЗАДАЧА			
ЦО-000003 Расчет ранее использованного отпуска - не работает	19.04.2018	Новая	19.04.2018
ЦО-000004 Устранить программную ошибку в модуле расчета отпускных	19.04.2018	Новая	19.04.2018

Рисунок 2.16. Результирующая форма документа «Задача»

2.3.2 Характеристика базы данных

Опишем структуру прикладных объектов конфигурации разрабатываемой системы в виде схем данных [19]. Из названия реквизитов объектов понятно назначение этих реквизитов. Типы данных реквизитов указаны в таблице для каждого объекта. Связи между объектами указаны на рисунках стрелками. Для каждого объекта приводится взаимосвязь с другими объектами. Структура справочников приведена на Рисунке 2.17.

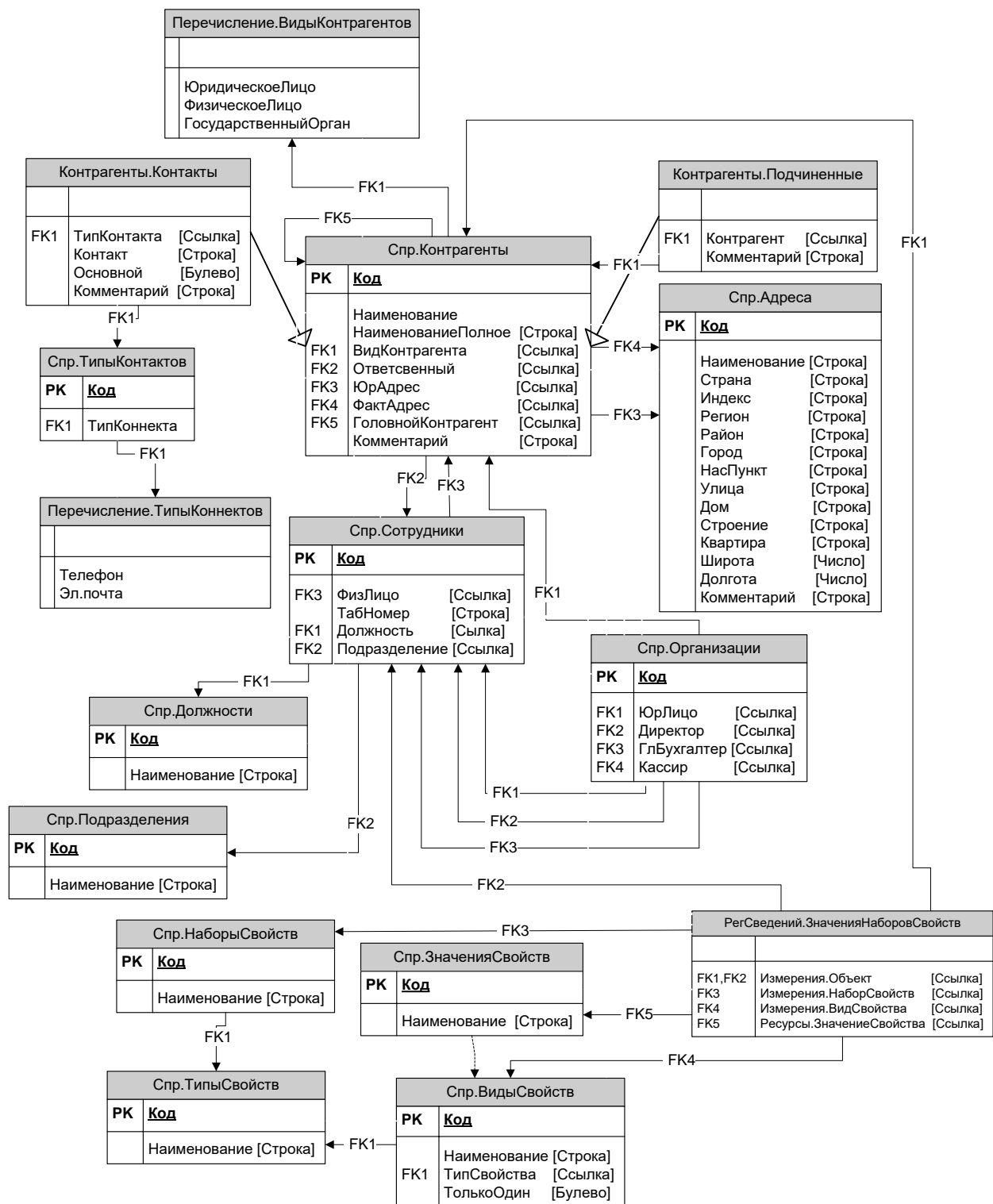


Рисунок 2.17. Структура справочников

Ключевые моменты организации справочников приведены на Рисунке 2.18.

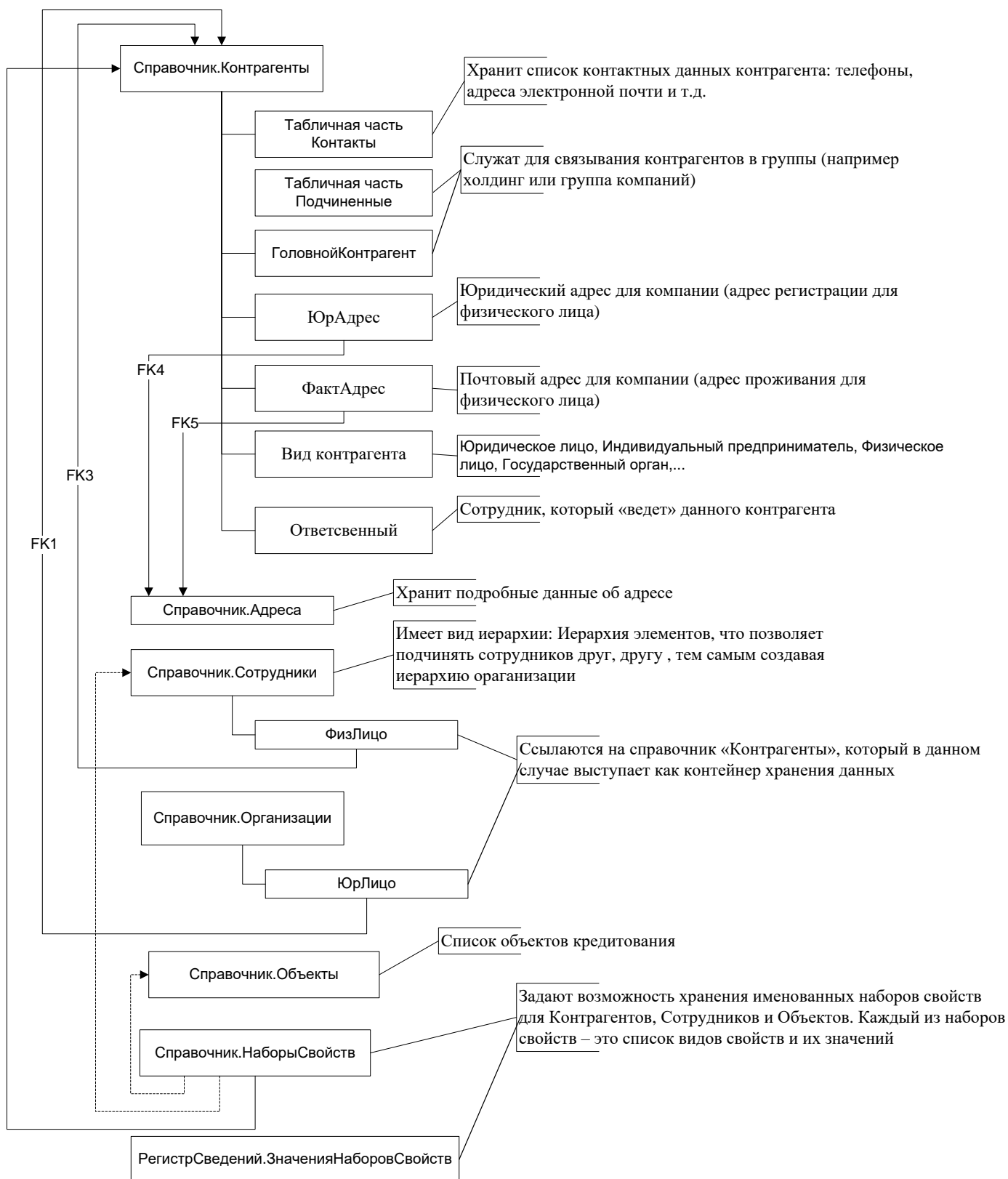


Рисунок 2.18. Ключевые моменты организации справочников

Документ «Задача» является центральным документом системы. Его структура приведена на Рисунке 2.19.

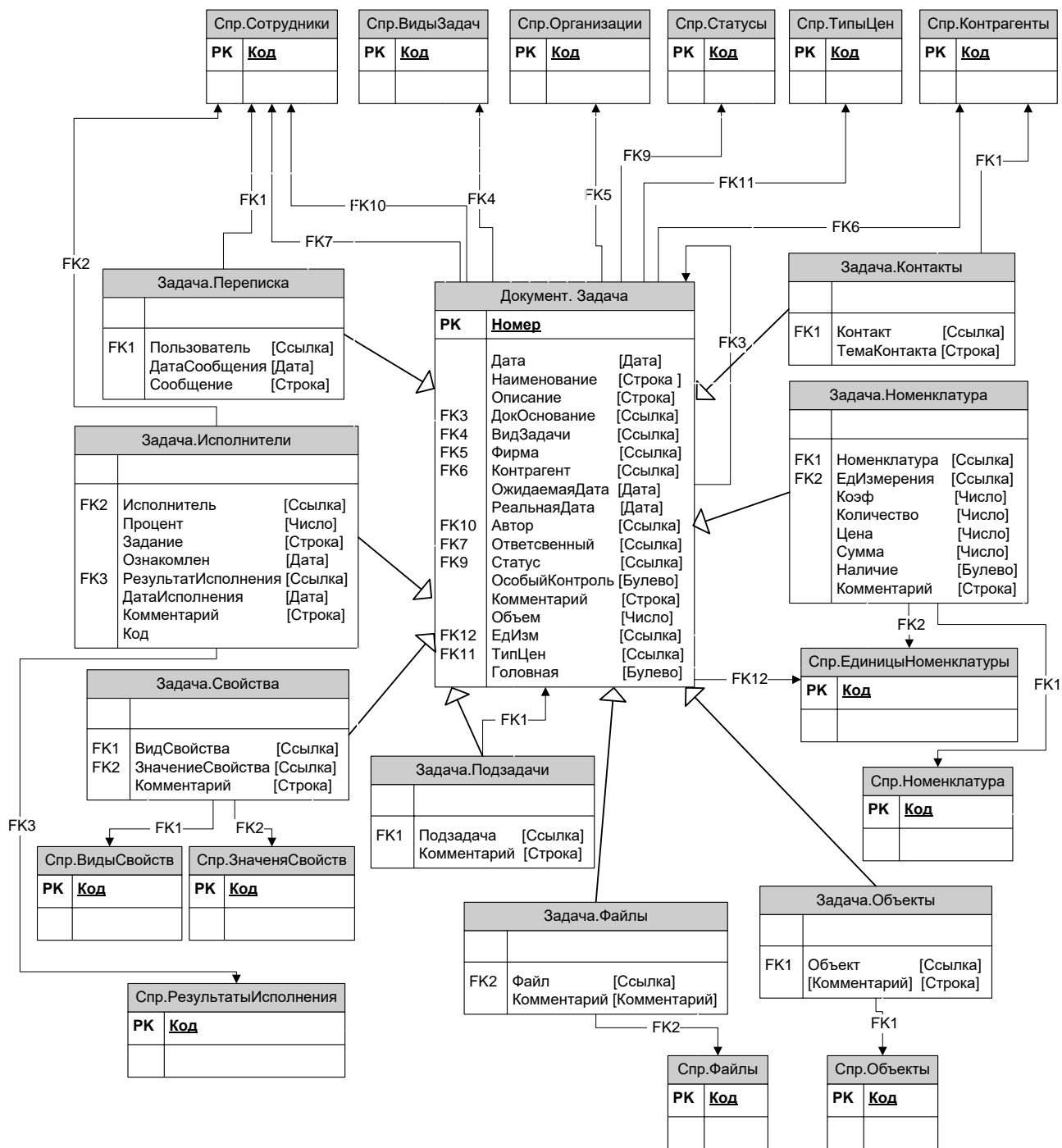


Рисунок 2.19. Структура документа «Задача»

Ключевые моменты структуры документа «Задача» приведены на Рисунке 2.20.

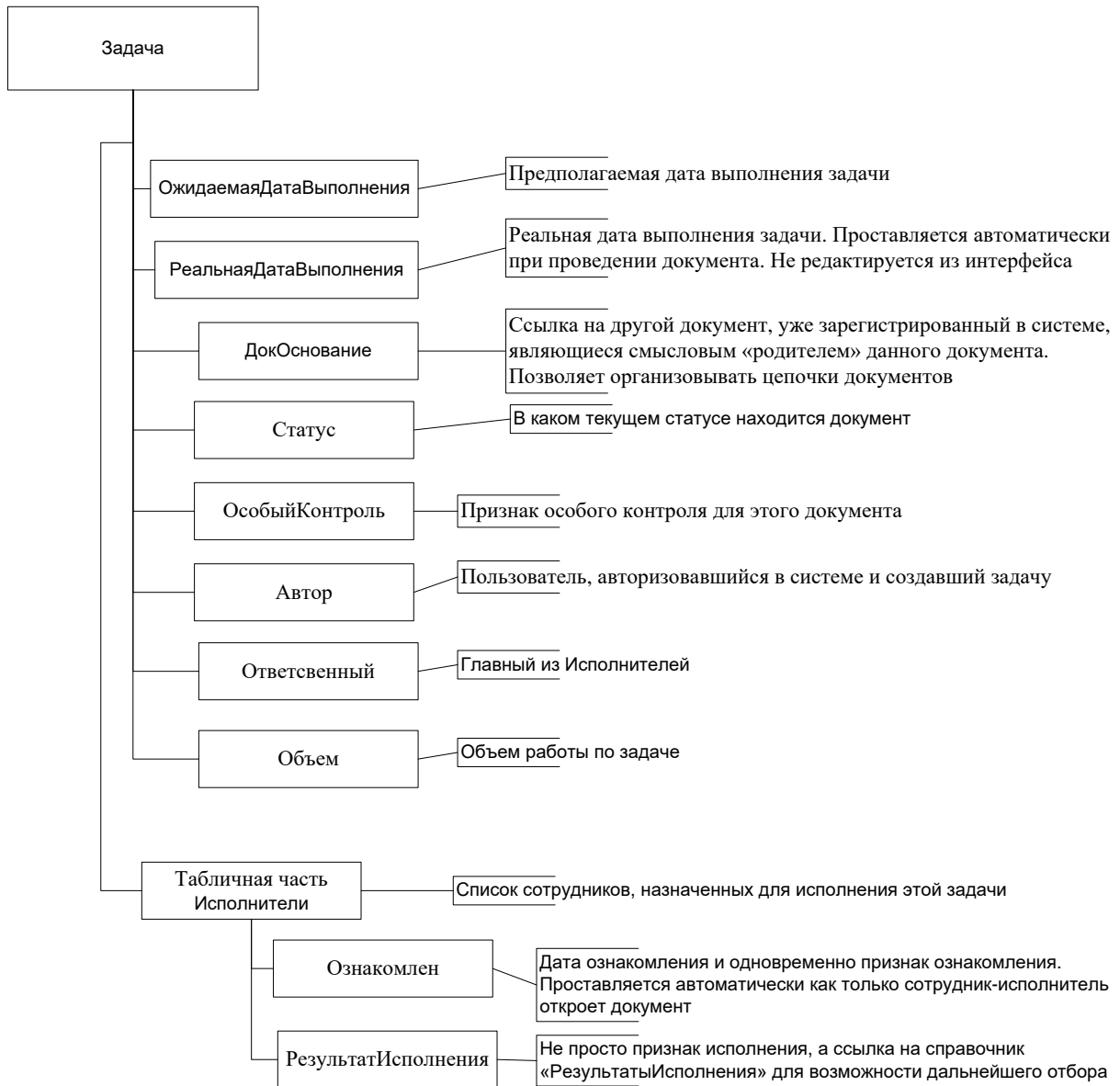


Рисунок 2.20. Ключевые моменты документа «Задача»

2.3.3 Структурная схема пакета (дерево вызова программных модулей)

Общая функциональная структура системы представлена на Рисунке 2.10. Опишем подробно функции каждой подсистемы.

Подсистема ведения справочников предназначена для ввода в систему условно-постоянной информации, которая понадобится на следующих этапах работы с системой.

Структура подсистемы ведения справочников приведена на Рисунке 2.21.



Рисунок 2.21. Структура подсистемы ведения справочников

Подсистема фиксации заявок должна предоставлять возможность ввода в формализованном виде заявок пользователей, назначения им реквизитов, в том числе списочного характера, максимально полно описывающих заявки. Подсистема фиксации заявок представлена документом «Задача», который является формальным представлением реальной заявки, поступившей в Службу. Этим же документом в систему вводятся или автоматически генерируются задачи сотрудникам по сопровождению заявки, если ее решение требует решения подзадач. Документ «Задача» должен позволять максимально полно описать реальную проблему, возникшую у пользователя ИТ-инфраструктуры предприятия, которую он формально представляет. Для этого разработана структура этого документа, представленная на Рисунке 2.22.



Рисунок 2.22. Структура документа «Задача»

Подсистема автоматической генерации подзадач по сопровождению заявки должна позволять на основании шаблонов, задающих различные сценарии прохождения этапов решения типовой проблемы сгенерировать документы «Задача», адресовав их конкретным сотрудникам (или подразделениям) Службы (назначив их исполнителями).

Совокупность документов «Задача» порожденных автоматически или вручную к одной заявке представляет собой древовидный список работ по ее решению. Структура подсистемы приведена на Рисунке 2.23.

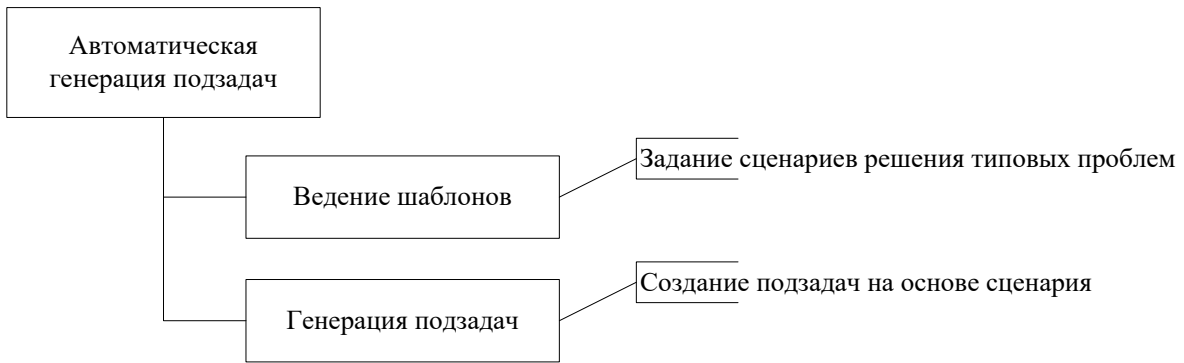


Рисунок 2.23. Структура подсистемы генерации подзадач

Подсистема сопровождения задач должна предоставлять возможность отслеживания задачи по всему ее жизненному циклу. Ввод в систему формального представления задачи, будь то заявка пользователя или один из этапов ее решения – это только начало ее жизненного цикла. В реальной жизни Службы редко возникают задачи, которые решаются за один день и не требуют каких-либо дополнительных действий. Большинство задач требуют выполнения определенных действий от сотрудников в процессе своего жизненного цикла. Для этого система должна предусматривать следующие возможности, представленные на Рисунке 2.24.

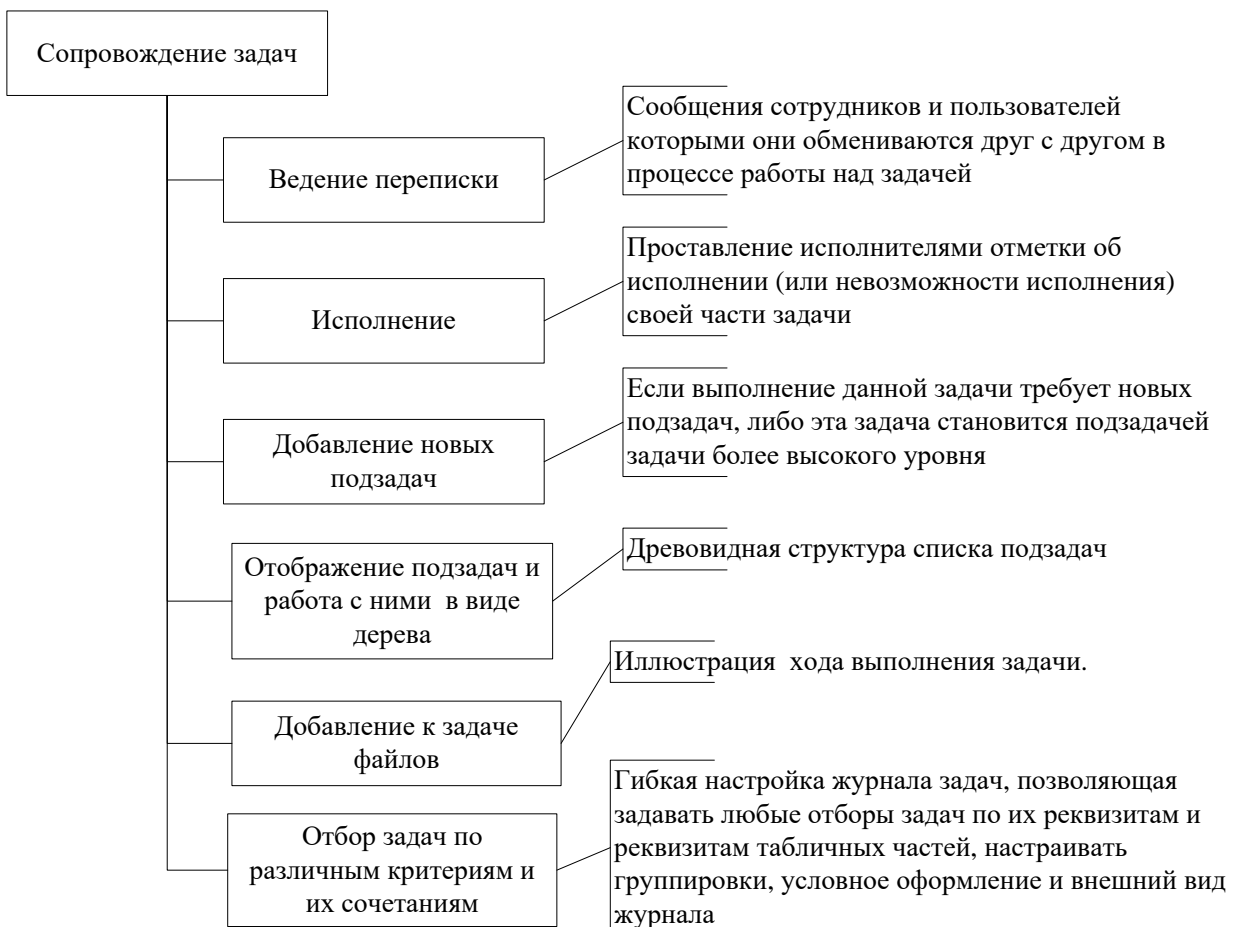


Рисунок 2.24. Функции подсистемы сопровождения задач

Подсистема выполнения задач позволяет разделить задачи на полностью выполненные (закрытые) и находящиеся в работе, на каком-либо из этапов. Исполнение задачи заключается в проставлении исполнителями признака исполнения частей задачи, их касающихся. Список исполнителей – это список сотрудников Службы, которые назначены исполнителями по данной задаче, исходя из темы задачи. Исполнителей назначает администратор Службы (или его начальник), либо их список генерируется автоматически, если сама задача сгенерирована по шаблону. Исполнителей может быть несколько – по одному для каждой из частей задачи. Процедура исполнения задачи представлена на Рисунке 2.25.

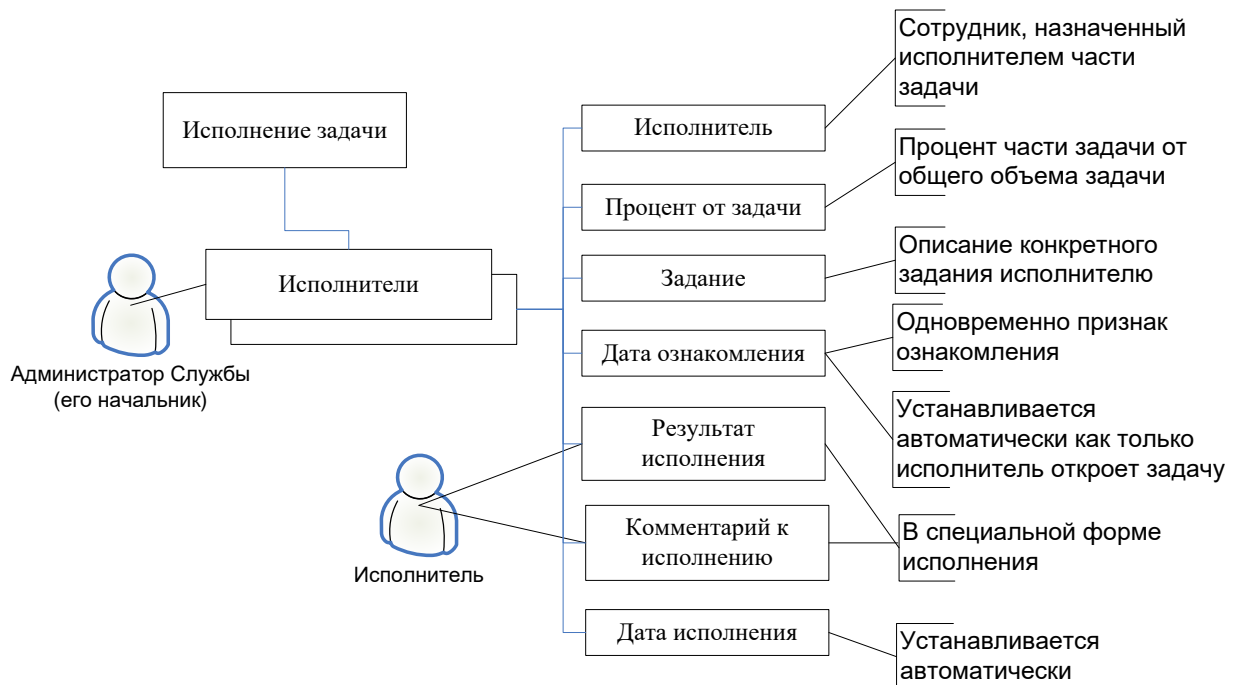


Рисунок 2.25. Исполнение задачи

Особое место в подсистеме сопровождения задач занимает работа с **древовидной структурой подзадач**. В дереве должны быть реализованы все операции с подзадачами любого уровня вложенности как с обычной задачей. Работа с деревом подзадач представлена на Рисунке 2.26.

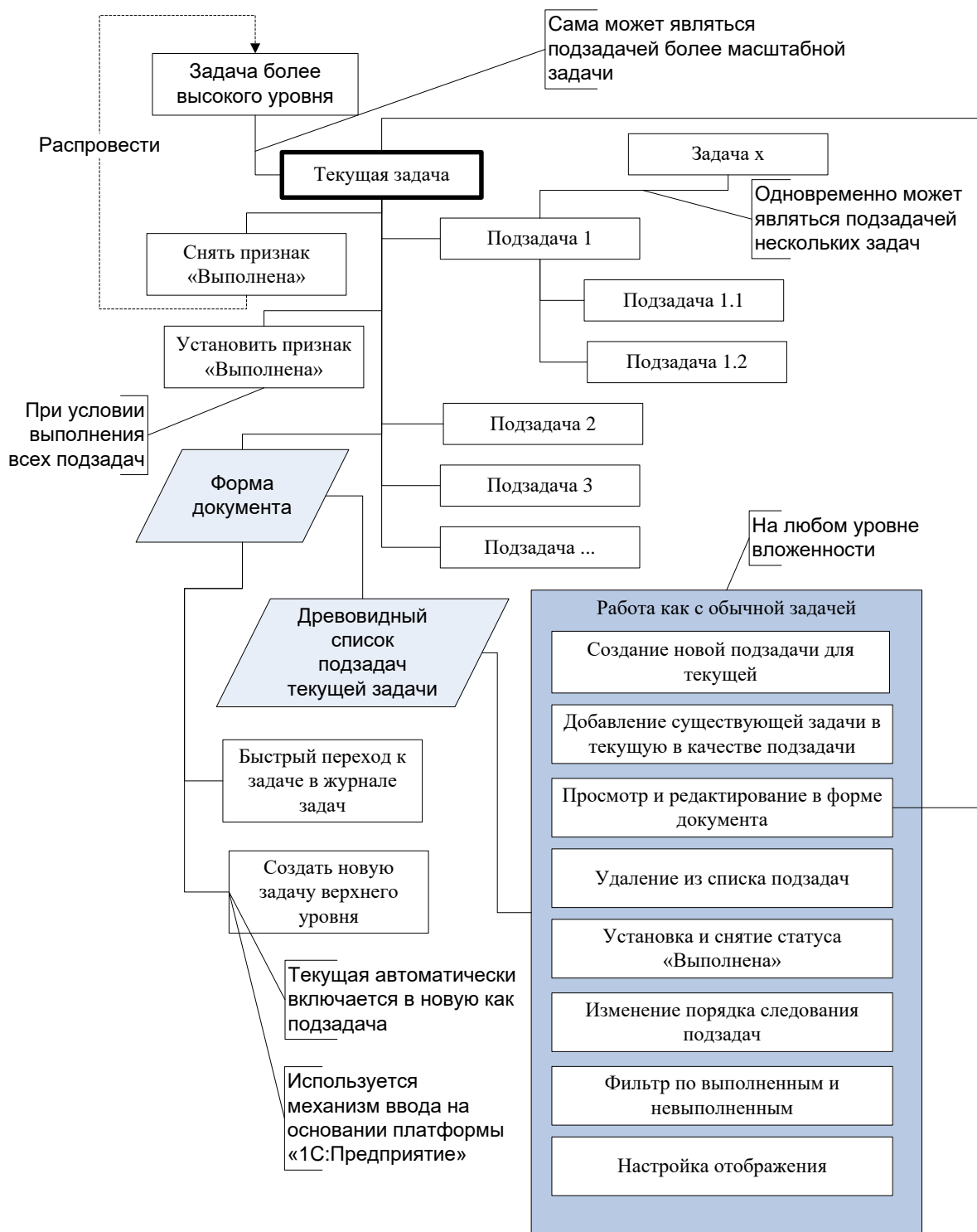


Рисунок 2.26. Работа с деревом подзадач

При выполнении задачи она переводится в состояние «Выполнена», после чего ее редактирование становится невозможным. Важным реквизитом документа «Задача» является «Статус». Поэтапное ведение задач заключается в переходе их из одного статуса в другой. Механизм выполнения задач представлен на Рисунке 2.27.



Рисунок 2.27. Механизм выполнения задачи

Подсистема разграничения прав пользователей должна предоставлять гибкую возможность настройки прав пользователей на ввод и редактирование задач и их параметров в зависимости от положения пользователя в иерархии Службы. Одновременно с системой могут работать несколько человек – сотрудники Службы и сами пользователи. Гибкая настройка прав, проиллюстрированная на Рисунке 2.28, позволяет сотрудникам редактировать документы только в пределах своей компетенции.

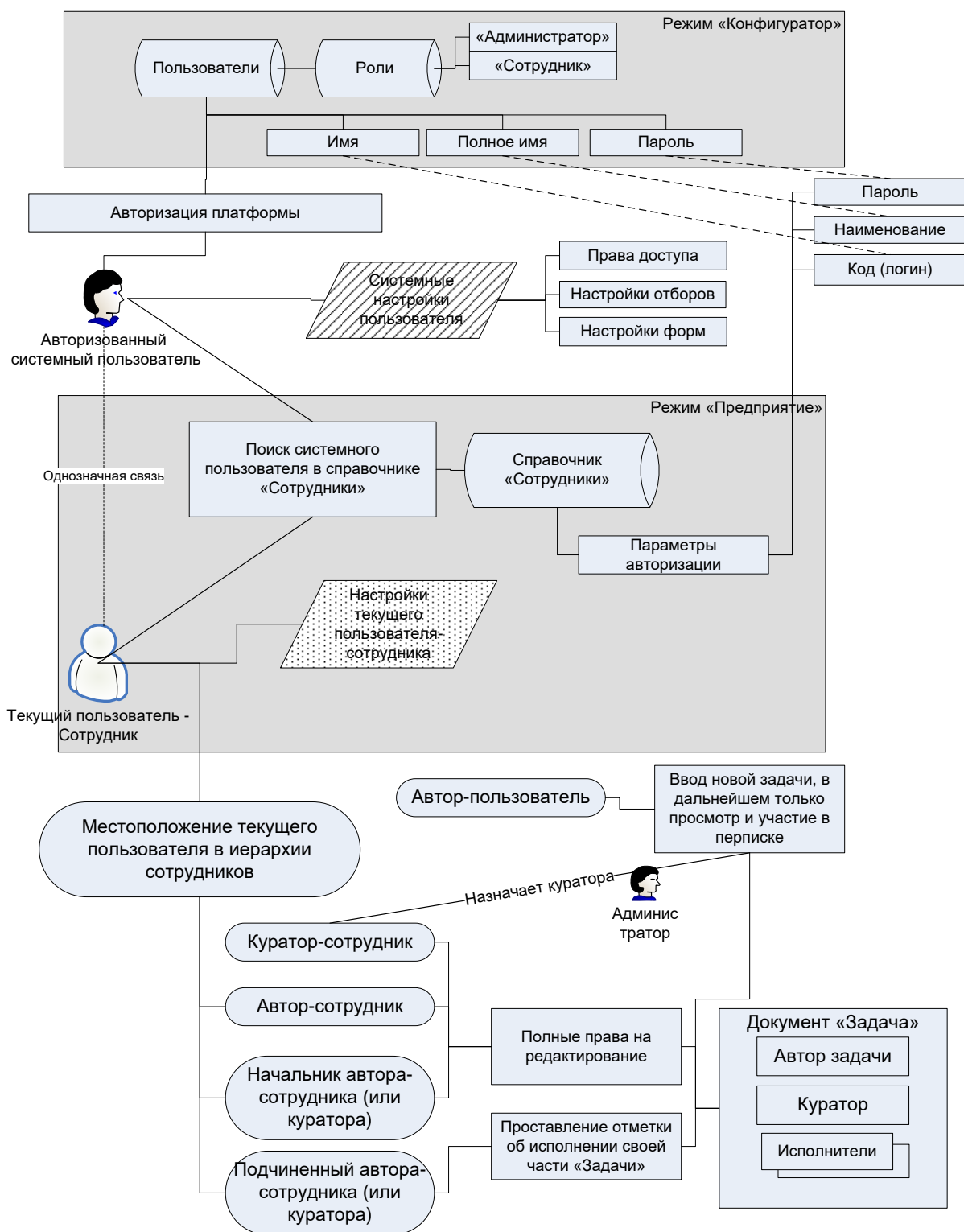


Рисунок 2.28. Настройка прав пользователей

Список пользователей системы может формироваться двумя способами: администратором Службы в «ручном» режиме (Рисунок 2.29) и самостоятельно пользователем КИС в автоматическом режиме при регистрации нового пользователя (Рисунок 2.30). В последнем случае предполагается что пользователь зашел в систему в том числе и через web.

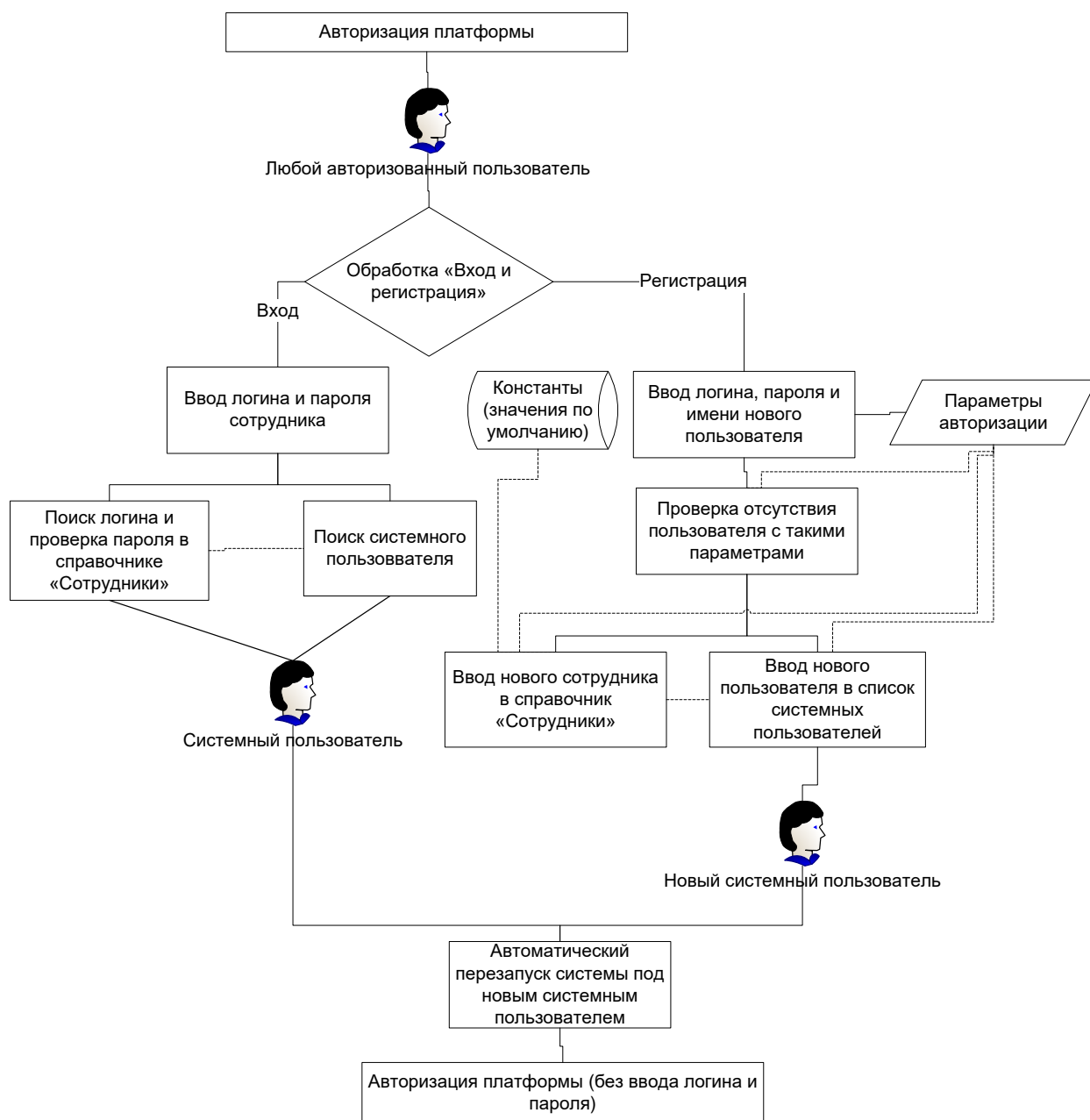


Рисунок 2.30. Ввод нового пользователя через обработку «Вход и регистрация»

Система не предполагает возможность входа неавторизованного пользователя. Новый пользователь, который хочет самостоятельно зарегистрироваться в системе (обычно это пользователь, зашедший через web) должен авторизоваться под системным пользователем «Гость», который обязательно должен быть создан администратором в списке системных пользователей с правами «Гость» в момент ввода системы в эксплуатацию. Пользователь «Гость» обладает правами аналогичными правам «Пользователь», кроме оформления заявки. Кроме всего прочего ему доступна и обработка «Вход и

регистрация», запустив которую он может зарегистрировать нового персонифицированного пользователя (себя) и далее авторизовываться уже под ним.

Пользователю, зарегистрировавшемуся самостоятельно (равно как и зарегистрированному администратором в «ручном» режиме) доступна только функция ввода новой задачи - оставить заявку администратору Службы на выполнение каких-то работ, задать вопрос или написать претензию, а также просмотр «своих» ранее введенных заявок. При просмотре новой или ранее созданной заявки пользователь может просматривать все ее параметры и ход выполнения.

Так как система устанавливает однозначную связь между системными пользователями, созданными в модуле «Конфигуратор» и сотрудниками (записями справочника «Сотрудники»), то получается что при регистрации нового пользователя он автоматически становится сотрудником Службы. Для штатных сотрудников это верно, а пользователи КИС становятся псевдо-сотрудниками и заносятся в специальную папку «Пользователи» справочника «Сотрудники».

Создаваемое приложение будет работать в режиме «тонкого клиента» и в режиме веб-клиента, но разработчику нет надобности ориентироваться на какой-то конкретный вариант. Это еще один плюс платформы «1С:Предприятие 8.3»: не нужно задумываться о том в каком варианте будет запускаться разрабатываемое приложение – платформа обо всем позаботится сама. Но это произойдет только в том случае если приложение разрабатывается по технологии «Управляемое приложение».

Эта технология разработки отличается от обычного приложения. В первую очередь, это кардинально новый подход к созданию интерфейса пользователя. В основном это относится к командному интерфейсу, имеющего декларативный принцип построения. Также полностью изменился процесс разработки форм, интерфейс которых лишь декларируется, а создается платформой «на лету» в момент вызова формы [17].

Пользователи КИС и сотрудники Службы могут работают с приложением в режиме веб-клиента через сеть Интернет. В отличие от "обычных" приложений веб-клиент не нужно устанавливать на компьютер и у него нет исполняемого файла.

Веб-клиент выполняется не в среде операционной системы компьютера, а в среде интернет-браузера (Mozilla Firefox, Windows Internet Explorer, Google Chrome или Safari). Достаточно всего лишь запустить браузер, ввести адрес веб-сервера, на котором опубликовано разработанное приложение, – и веб-клиент сам "придет" к нему на компьютер и начнет работать.

Веб-клиент применяет технологии XMLHttpRequest и DHTML. Программные модули, разработанные в решении, автоматически компилируются из языка платформы «1С:Предприятия 8» и непосредственно исполняются на стороне веб-клиента (Рисунок 2.31).

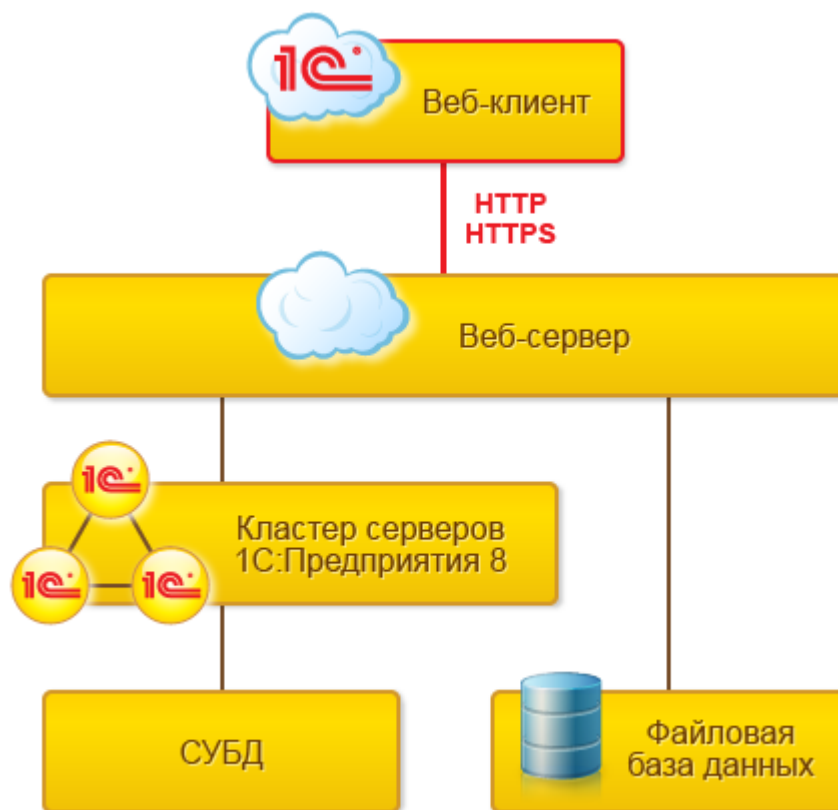


Рисунок 2.31. Схема работы веб-клиента

Прикладные решения не требуют какой-либо доработки для того, чтобы работать с ними через Интернет. И тонкий, и веб-клиент самостоятельно обеспечивают функционирование интерфейса 1С:Предприятия 8 на компьютере пользователя. Подключение клиентских приложений может выполняться как по протоколу http, так и по протоколу https, поддерживающему шифрование передаваемых данных.

2.3.4 Описание программных модулей

Для хранения условно-постоянных данных разрабатываемой системы целесообразно использовать прикладной объект «Справочники» платформы. Для каждой категории информации необходимо создать отдельный справочник, определить и задать для него параметры, представленные закладками формы настройки прикладного объекта «Справочник».

Состав справочников разрабатываемой системы, как он выглядит в модуле «Конфигуратор», приведен на Рисунке 2.32.

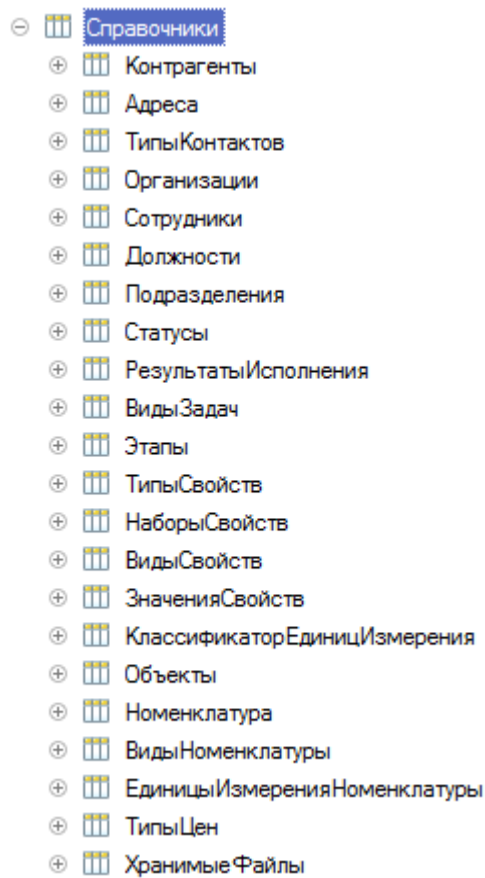


Рисунок 2.32. Список справочников системы

Для каждого справочника необходимо задать структуру – т.е. состав реквизитов справочника и состав его табличных частей, для каждой из которых необходимо создать свою структуру реквизитов. В реквизитах целесообразно хранить свойства объектов информации. На Рисунке 2.33 представлены структура реквизитов и табличных частей справочника «Контрагенты».

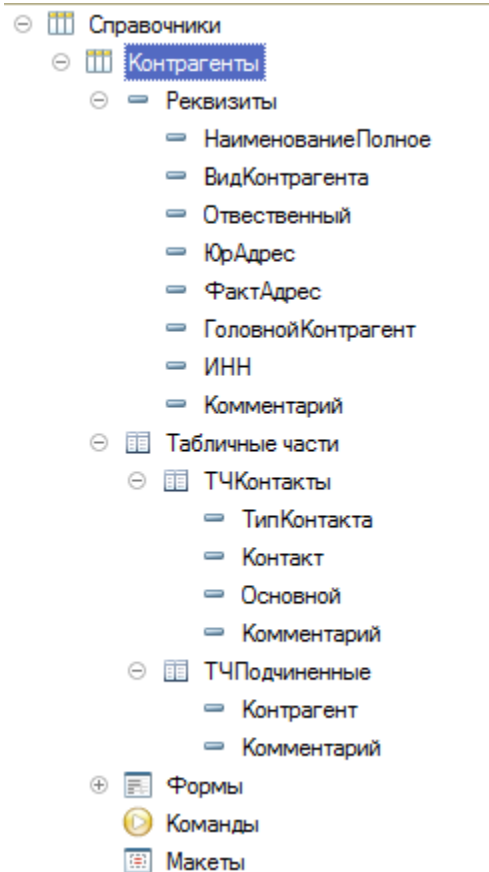


Рисунок 2.33. Структура справочника «Контрагенты»

Центральным объектом разрабатываемой конфигурации должен стать документ «Задача». Документ «Задача» — это формальное представление реальной заявки, поступившей в Службу или задачи по одному из этапов ее сопровождения. Реквизиты шапки и табличных частей документа полностью отображают параметры заявки, состав экранных форм позволяет создать интерфейс пользователя, в программных модулях экранных форм задается логика поведения документа и взаимодействия с пользователем, факт проведения документа отражает факт фиксации полного выполнения задачи [18]. Структура документа «Задача» приведена на Рисунке 2.34.

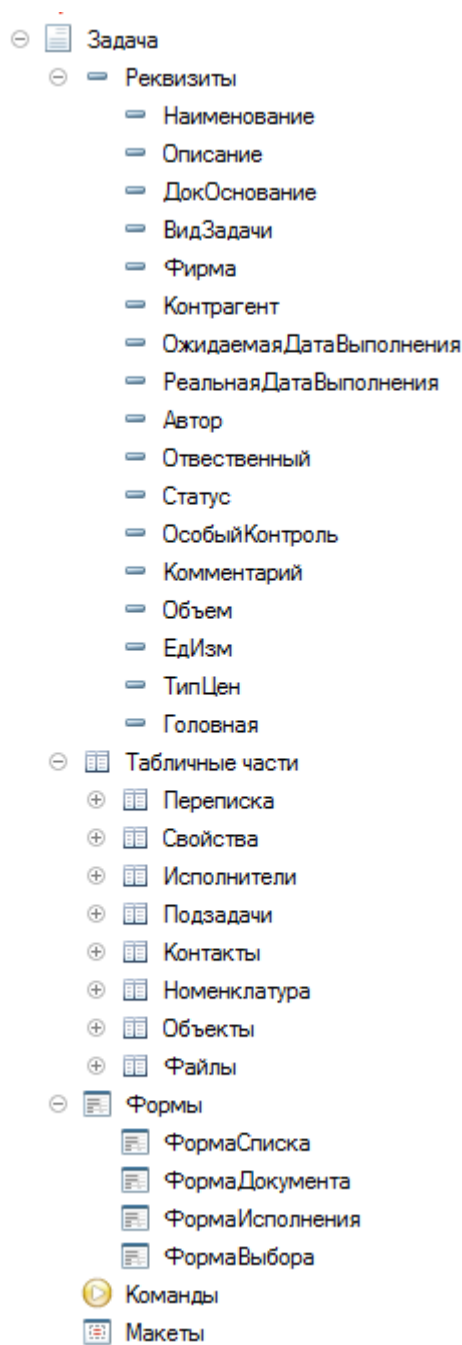


Рисунок 2.34. Структура документа «Задача»

При такой сложной структуре на первое место выходит организация грамотного и удобного интерфейса этого документа, который, во-первых, предоставит полноценный интерфейс работы с этим документом, во-вторых, обеспечит логику работы: оградит от недопустимых действий, автоматически заполнит необходимые поля и сделает нужные действия, представит данные в удобном виде.

2.4 Контрольный пример реализации проекта и его описание

При запуске программы пользователь видит перед собой основное окно и панель разделов. Все функции программы сгруппированы по разделам, соответствующим подсистемам конфигурации. На Рисунке 2.35 представлена панель разделов и меню функций текущего раздела «Справочники».

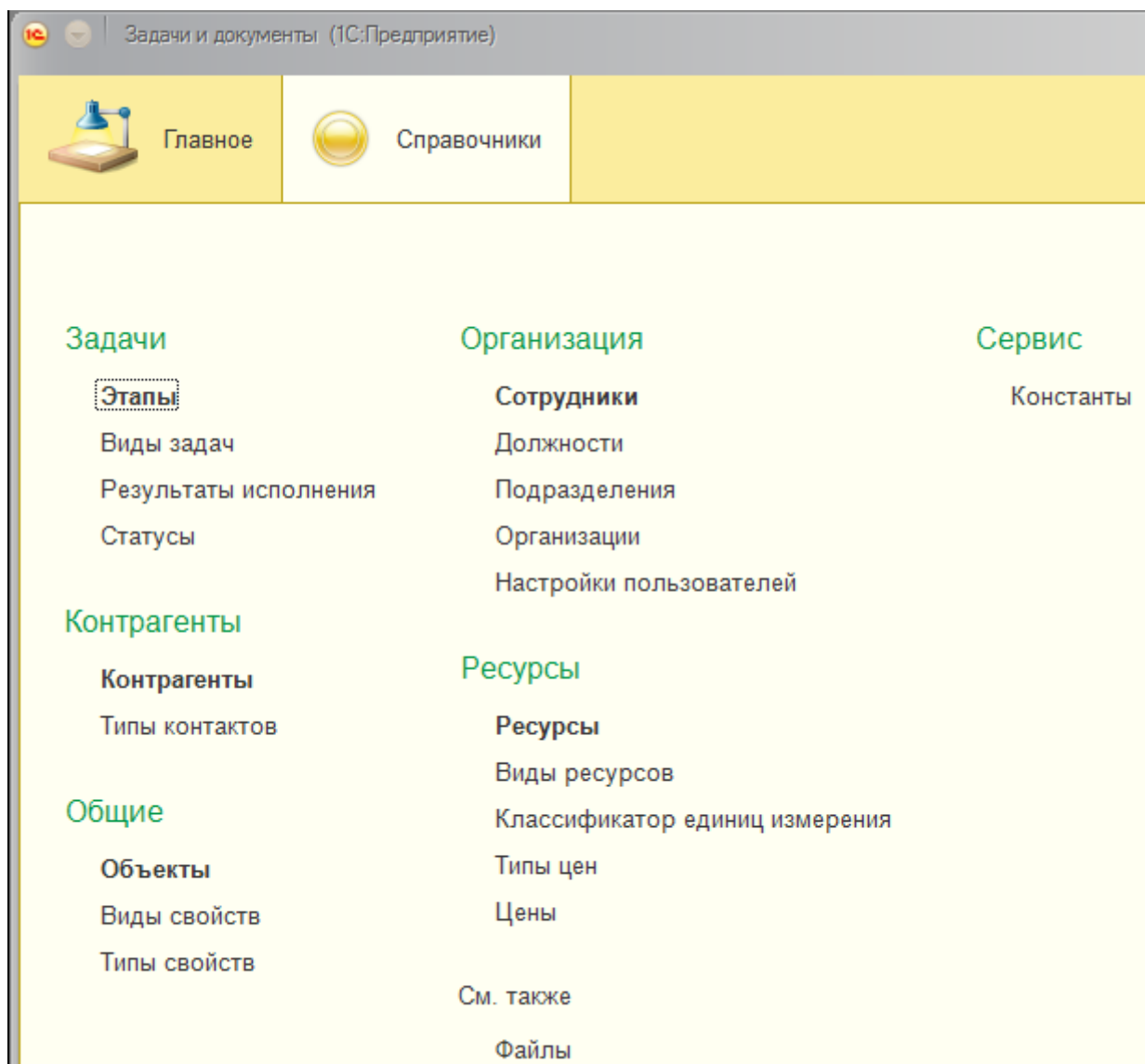


Рисунок 2.35. Панель разделов и меню функций раздела «Справочники»

Работу с системой целесообразно начать с заполнения справочников, заполненные значения которых удобно будет подставлять в качестве параметров документов. Среди справочников наибольшее значение имеют справочник «Контрагенты», в котором хранятся сведения о клиентах предприятия компании

Справочник открывается при вызове команды «Контрагенты» меню функций раздела «Справочники». Справочник имеет иерархическую структуру. Для

просмотра и редактирования данных конкретного контрагента необходимо двойным щелчком открыть окно этого контрагента (Рисунок 2.36).

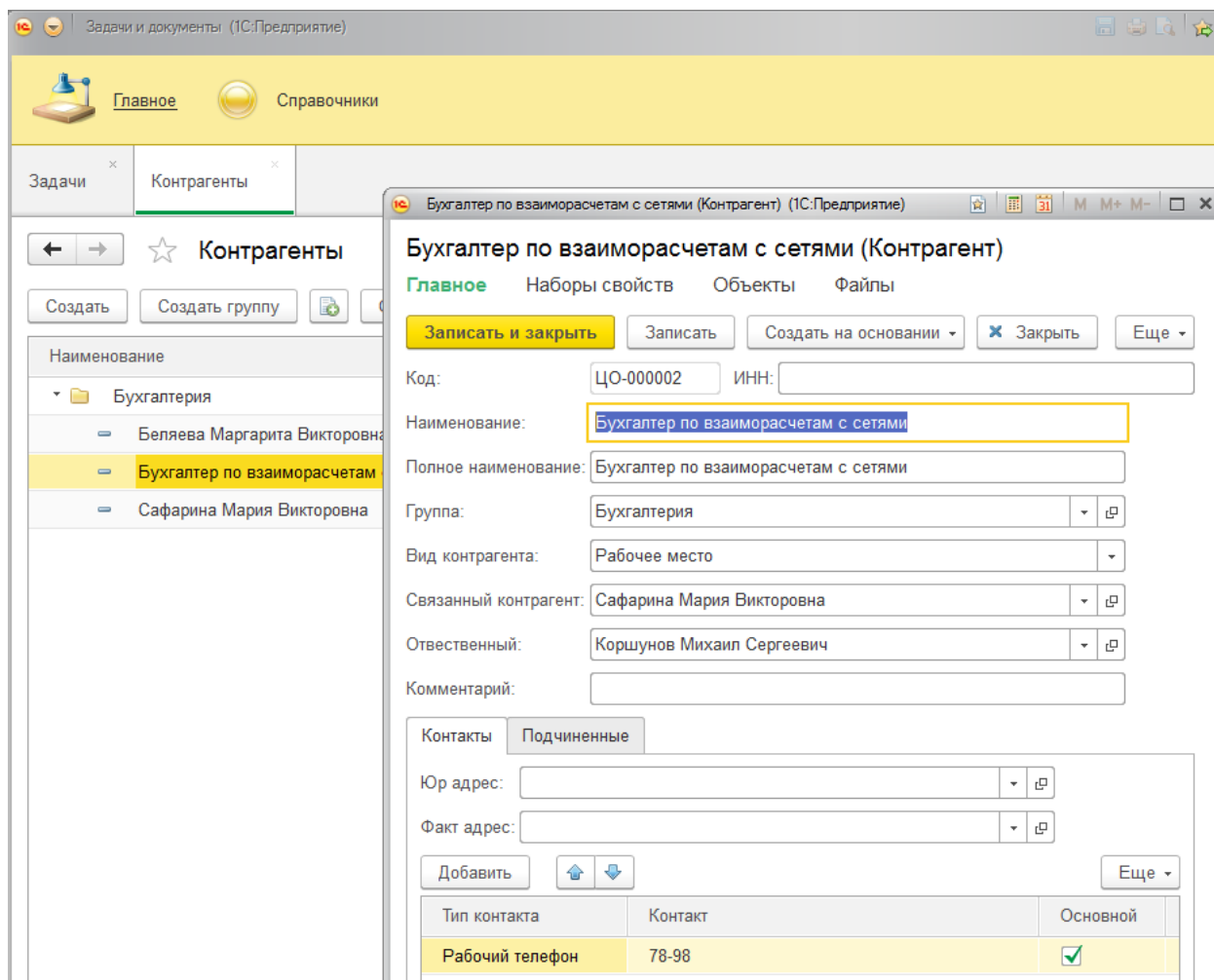


Рисунок 2.36. Справочник «Контрагенты»

При помощи командного интерфейса можно перейти к просмотру и редактированию связанных данных контрагента.

Остальные справочники редактируются аналогично, и мы не будем на них подробно останавливаться.

Наибольший интерес при работе с программой представляет из себя режим работы со списком задач по сопровождению заявок клиентов. Как правило, контрагент звонит в Службу и описывает проблему, возникшую у него в ходе эксплуатации ресурсов. Сотрудником Службы оформляется задача со слов пользователя, назначается на нее ответственный исполнитель.

Задачу в системе может сгенерировать и сам пользователь, если он самостоятельно зарегистрировался в системе или администратором Службы ему дан доступ, и он авторизовался в ней. В этом случае автором задачи является

пользователь, а ответственным исполнителем – администратор Службы (Рисунок 2.37)

Рисунок 2.37. Постановка задачи

Решение задачи назначается конкретному исполнителю по профилю задачи.

Цель исполнения – решение данной задачи (Рисунок 2.38).

N	Исполнитель	Процент	Ознако...	Исполнение
1	Логинев Сергей Валерьевич	100,00		
	Модуль расчета отпускных			Комментарий

Рисунок 2.38. Назначение исполнителя на задачу

После того как исполнитель задачи выполнил все действия по задаче, он должен проставить отметку об ее исполнении. Для этого на закладке «Исполнители» необходимо нажать кнопку «Исполнить» и внести параметры исполнения (Рисунок 2.39).

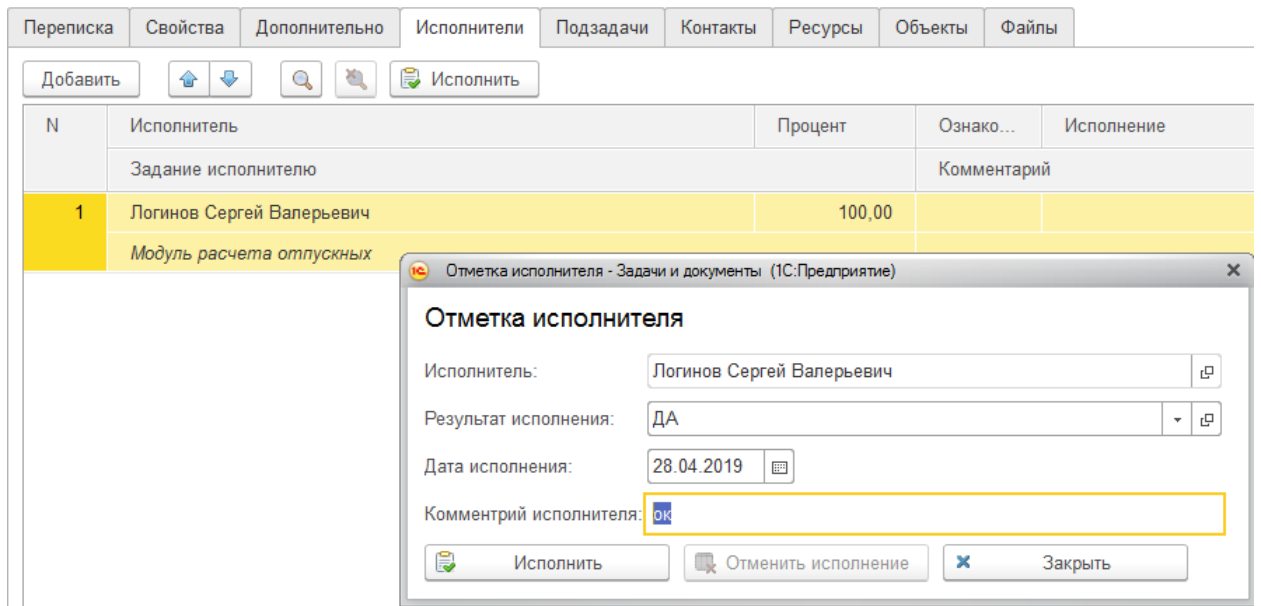


Рисунок 2.39. Отметка исполнения задачи

Если решение текущей задачи требует решения других задач, то их можно сгенерировать на основании данной задачи. Сгенерированные задачи могут в свою очередь иметь подзадачи. Образуется древовидная структура задач, которую можно видеть на закладке «Подзадачи» текущей задачи (Рисунок 2.40).

Наименование	Дата	Статус	Ожидается
ЭТА ЗАДАЧА			
ЦО-000003 Расчет ранее использованного отпуска - не работает	28.04.2019	Новая	30.04.2019
ЦО-000004 Устранить программную ошибку в модуле расчета отпускных	28.04.2019	Новая	03.05.2019

Рисунок 2.40. Закладка «Подзадачи» текущей задачи

Дальнейшая работа по устранению проблемы, указанной в задаче и возможно каких-то еще действий по этой задаче производится из головной задачи. На закладке «Подзадачи» хорошо видна структура и ход выполнения каждой задачи. Список команд данной закладки предоставляет пользователю все функции по управлению задачами любого уровня вложенности.

Войдя в любую задачу из дерева задач можно подробной с ней ознакомиться. Закладки «Свойства», «Дополнительно», «Контакты», «Объекты», «Файлы» являются общими для всех подзадач процесса, хотя каждая конкретная задача может быть дополнена своими специфическими записями на любой из закладок.

Работа с задачей «сверху» очень удобна для руководителя Службы для просмотра хода выполнения задачи и поиска узких мест и проблем ее выполнения. Однако, в повседневной работе сотрудники предприятия не работают с одной

задачей, они работают со списком задач, поступающим к ним на выполнение, отрабатывают их, возможно порождают новые подзадачи, которые в свою очередь требуют отработки и переходят к следующим задачам.

3 Обоснование экономической эффективности проекта

3.1 Выбор и обоснование методики расчёта экономической эффективности

Создание новых программных решений, а также их последующее внедрение для автоматизации деятельности коммерческих предприятий требует больших материальных, трудовых и временных ресурсов. Для реализации проектов требуется привлекать значительные материальные и финансовые средства. Эти средства, безусловно, должны дать соответствующую отдачу. Вот из-за чего возникает необходимость определения экономической эффективности (ЭЭ) процесса разработки и внедрения АИС.

Обоснование и доказательство ЭЭ (или ее неэффективности) нового разрабатываемого варианта компьютерной обработки данных позволяет определить необходимость и целесообразность затрат на разработку и внедрение новой АИС на конкретном объекте; определить срок окупаемости затрат на разработку АИС и сравнить его с установленными нормами; определить степень влияния и эффекта от внедрения новых разрабатываемых продуктов на успешную деятельность предприятия, на котором происходит внедрение.

Существует несколько вариантов определения ЭЭ проектов:

- Сравнение способов создания различных систем обработки данных по рассматриваемой задаче;
- Сравнение способов организации сбора, обработки, передачи и выдачи данных с точки зрения технологического процесса;
- Сравнение способов создания информационной базы рассматриваемой задачи;
- Сравнение существующих методов и инструментов для проектирования систем обработки данных;
- Сравнение различных технологий компьютерной обработки данных.

Разработка рассматриваемого проекта предполагает, в первую очередь, перестройку системы обработки данных с ручного на компьютеризированный способ. В итоге был выбран первый вариант расчета ЭЭ.

Понятие «Экономическая эффективность» является оценкой результативности разрабатываемой АИС, определяемую отношением между затратами и получаемым от внедрения системы эффектом.

ЭЭ «ИТ - проекта» определяется показателями трудовой и стоимостной групп, дающие возможность измерить эффект от внедрения предлагаемого проекта компьютерной обработки данных относительно базового, ручного варианта.

ЭЭ проекта суммируется из следующих составляющих:

- Косвенного эффекта, учитывающего, например, увеличением прибыли компании, привлечением более значительного числа покупателей, уменьшение уровня брака в процессе производства, уменьшение количества претензий, получаемых от покупателей, уменьшение затрат на полуфабрикаты, сырье и материалы, уменьшение сумм пени, штрафов, неустоек и т. д.

- Прямого эффекта, который характеризуется снижением стоимостных и трудовых затрат.

К группе трудовых затрат (ТЗ) относятся такие как:

1. Абсолютное снижение ТЗ (ΔT)

$$\Delta T = T_0 - T_1, \text{ где} \quad (1)$$

T_0 – показатель ТЗ на обработку данных по базовому (ручному) варианту;

T_1 – показатель ТЗ на обработку данных по новому варианту;

2. Коэффициент относительного уменьшения показателей ТЗ (K_T)

$$K_T = \frac{\Delta T}{T_0} \times 100\%, \quad (2)$$

3. Индекс снижения показателей группы ТЗ, или увеличение производительности труда (Y_T)

$$Y_T = \frac{T_0}{T_1}, \quad (3)$$

Показатели стоимостных затрат (СЗ) рассчитываются по аналогичным формулам:

4. Абсолютное снижение СЗ (ΔC)

$$\Delta C = C_0 - C_1, \text{ где} \quad (4)$$

C_0 - СЗ на обработку данных по базовому (ручному) варианту;

C_1 - СЗ на обработку информации по новому варианту;

5. Коэффициент относительного уменьшения СЗ (K_C)

$$K = \frac{\Delta C}{C_0} \times 100\%, \quad (5)$$

6. Индекс снижения СЗ (Y_C)

$$Y_C = \frac{C_0}{C_1}, \quad (6)$$

Помимо основных показателей логично определить срок окупаемости проекта (T_{OK}):

$$T_{OK} = \frac{K_N}{\Delta C}, \text{ где} \quad (7)$$

K_N - дополнительные расходы на разработку проекта (затраты на его проектирование и внедрение).

Рассмотрим показатели, которые рассчитываются при ручном варианте обработки данных:

1. Трудоемкость обработки документа № I (T_{O_i}):

$$T_{O_i} = \frac{0,45Qz_i + 1,8Qn_i + 1,125Qc_i + 3,71Qy_i + 4,05Qq_i}{3600} \quad (8)$$

2. ТЗ в варианте ручной обработке (T_O):

$$T_O = \sum_i T_{O_i} \times N \quad (9)$$

3. СЗ в варианте ручной обработке (C_O):

$$C_O = T_O \times p \times (1 + K_g), \text{ где} \quad (10)$$

$Qz_i, Qn_i, Qc_i, Qy_i, Qq_i$ - объем символов в i -ой операции;

N – количество операций, вводимых вручную;

p – тарифная часовая ставка сотрудника;

K_g – коэффициент доп. заработной платы, отчисления на социальное страхование, накладные и косвенные расходы. $K_g = 0,53 \div 0,85$

Рассмотрим показатели, которые рассчитываются при компьютерном варианте обработки данных:

1. Трудоемкость одной операции:

$$T_j = \frac{Q_j}{H_j} \quad (\text{кроме ПК}) \quad (11)$$

$$T_{jЭВМ} = t_{ме} \quad (\text{ПК}) \quad (12), \text{ где:}$$

T_j , $T_{jЭВМ}$ - трудоемкость операции № j и обработки на ПК соответственно;

Q_j - количество работы на операции № j ;

H_j - норма выработки на операции № j ;

$t_{ме}$ - длительность решения задачи на компьютере.

2. Заработная плата участвующих в процессе операторов:

$$C_{зj} = P_j * T_j \quad (\text{кроме ПК}) \quad (13)$$

3. Прочие и накладные расходы:

$$C_{gj} = C_{зj} * K_{gj} \quad (\text{кроме ПК}) \quad (14)$$

4. Стоимость амортизации машин и оборудования:

$$C_{aj} = a_j * T_j \quad (\text{кроме ПК}) \quad (15)$$

5. Операционные и прочие расходы:

$$C_j = C_{зj} + C_{gj} + C_{aj} \quad (\text{кроме ПК}) \quad (16)$$

$$C_{iЭВМ} = C_{MB} * t_{MB}, \text{ где:} \quad (17)$$

$C_{зj}$, C_{gj} , C_{aj} - заработная плата оператора, накладные и прочие расходы и амортизация на операции № j ;

P_j - тарифная ставка данной операции;

$K_{gj} = 0,6 \div 0,8$ - коэффициент дополнительной зар.платы, косвенные, накладные и прочие расходы, отчисления на социальное страхование;

a_j - часовая сумма амортизации;

C_j , $C_{jЭВМ}$ - общие СЗ на операции № j и при обработке на ПК соответственно;

$C_{МВ}$ - цена машинного часа ПК;

6. ТЗ в варианте компьютерной обработки (T_1):

$$T_1 = \sum_j T_j \quad (\text{с учетом ПК}) \quad (18)$$

7. СЗ в варианте компьютерной обработки (C_1):

$$C_1 = \sum_j C_j \quad (19)$$

Показатель «Годовой экономический эффект» от внедрения АИС предназначен для сравнения разных направлений капитальных вложений и считается по формуле:

$$\mathcal{E} = П - K_{П} \times E_n \quad (20),$$

где:

\mathcal{E} – годовой экономический эффект;

$П$ – экономия за год (годовой прирост прибыли), руб.;

$K_{П}$ – капитальные затраты единовременные, руб.;

E_n – коэффициент нормативный эффективности капитальных вложений. Данный коэффициент представляет из себя минимально возможную норму эффективности капитальных вложений, ниже которой проект нецелесообразен. Значение E_n обычно принимается равным 0,15.

К единовременным затратам ($K_{П}$) относят затраты на создание, программирование и тестирование решаемого комплекса задач, а также затраты на его внедрение на конкретном предприятии.

Рассчитываемый коэффициент ЭЭ капитальных затрат (E_P) представляет собой отношение экономии за год (или прироста прибыли за год) к капитальным затратам на разработку и внедрение проекта:

$$E_P = \frac{\Delta C}{K_{П}} = \frac{1}{T_{ок}}, \quad (21)$$

3.2 Расчёт показателей экономической эффективности проекта

Расчет конкретных показателей ЭЭ разрабатываемого проекта представлен в таблицах данного раздела. В таблице (Таблица 3.1) представлен расчет ТЗ в варианте ручной обработки документов. Все расчеты выполнялись при следующих заданных условиях:

Каждая операция выполняется только одним сотрудником;

Тарифная ставка работы сотрудника составляет в среднем 500 руб в час;

Цена одного часа работы компьютера составляет 5 руб./час.

Таблица 3.1.

Расчет ТЗ при ручном проведении операции в базовом ручном варианте

№ п/ п	Наименование операции	Кол-во операц й в год	В одной операции				
			Знаки записи	Знаки сложен ия	Знаки умноже ния	Знаки делени я	Сравнен ие, округлен ие
1	Приход от поставщика	1000	3284				
2	Заявки подразделений	1750	3284				
3	Возврат поставщику	625	587				
4	Перемещение	10750	426				
5	Заказ поставщику	700	498				
6	Списание	12	104058				
7	Внесение ТМЦ	25	104058				
8	Внесение контрагентов	12	183750				
9	Отчет по остаткам	12	200167				
10	Отчет по контролю списка ТМЦ	60	7621				
11	Отчет по каждодневному	600	351				

<i>№ n/ n</i>	<i>Наименование операции</i>	<i>Кол-во операц й в год</i>	В одной операции				
			<i>Знаки записи</i>	<i>Знаки сложен ия</i>	<i>Знаки умноже ния</i>	<i>Знаки делени я</i>	<i>Сравнен ие, округлен ие</i>
	списку журнала документов						

Трудоемкость выполнения одной операции по формуле (8) составит:

$$T_{01} = 0,453 \text{ (человек/час);}$$

$$T_{02} = 0,453 \text{ (человек/час);}$$

$$T_{03} = 0,073 \text{ (человек/час);}$$

$$T_{04} = 0,053 \text{ (человек/час);}$$

$$T_{05} = 0,064 \text{ (человек/час);}$$

$$T_{06} = 16,478 \text{ (человек/час);}$$

$$T_{07} = 13,320 \text{ (человек/час);}$$

$$T_{08} = 22,969 \text{ (человек/час);}$$

$$T_{09} = 25,021 \text{ (человек/час);}$$

$$T_{10} = 0,953 \text{ (человек/час);}$$

$$T_{11} = 0,044 \text{ (человек/час);}$$

Общие расходы за год по статье «трудовые затраты» в варианте ручной обработки данных, определенные по формуле (9) будут:

$$T_{0P} = 3109,87 \text{ (человек/час);}$$

Общие затраты за год по статье «стоимостные затраты» в варианте ручной обработки данных, по формуле (10) будут:

$$C_{0P} = 3109,871 \times 500 \times (1 + 0,7) = 2643390,35 \text{ (руб.);}$$

В Таблице 3.2 приведен расчет затрат при вводе операций при помощи компьютера в базовом ручном варианте. Все расчеты выполнены по формулам (11) – (19).

Расчет операционных затрат при компьютерной обработке операций в базовом (ручном) варианте

Показатель	Операция					Итого
	Проектирование, регистрация, контроль	Ввод исходных данных	Обработка данных	Выдача данных	Контроль	
Оборудование		ПК	ПК	ПК		-
Единица измерения	Операция	Знак	Знак	Знак	Операция	-
Объем, Q	15796	608515	6154	60000 5	15796	-
Среднечасовая норма выработки, Н (час ⁻¹)	6	2000	2000	12000	12	-
Трудоемкость, T _{ОМ} (час)	2632,67	304,26	3,07	50	1316,33	4306,33
Среднечасовая тарифная ставка, Р (руб.)	50	50	50	50	50	-
Заработная плата оператора, С _з (руб.)	131633,50	15213,0 0	153,50	2500,00	65816,50	-
Прочие расходы, С _г (руб.)	92143,45	10649,1 0	107,45	1750,00	46071,55	-
Стоимость амортизации оборудования, С _а (руб.)	-	1000,95	15,35	250,00	6581,65	-
Операционные расходы, С _{ОМ} (руб.)	223776,95	27383,4 0	276,30	4500,00	118469,7 0	374406,35

Таким образом, расходы по статье «трудовые затраты» за год в базовом (ручном) варианте составят:

$$T_0 = T_{Op} + T_{Om} = 3109,87 + 4306,33 = 7416,20 \text{ (человек/час);}$$

Общие СЗ за год в результате расчета в базовом (ручном) варианте составят:

$$C_0 = C_{0p} + C_{0m} = 2643390,3 + 374406,35 = 3017796,70 \text{ (руб.)};$$

В Таблице 3.3 приведен расчет операционных затрат при компьютерной обработке операций в предлагаемом варианте. Расчеты проведены по формулам (11) – (19).

Таблица 3.3.

Расчет операционных затрат при компьютерной обработке операций в новом варианте

<i>Показатель</i>	<i>Операция</i>					Итого
	<i>Проектирование, регистрация, контроль</i>	<i>Ввод данных</i>	<i>Обработка</i>	<i>Выдача данных</i>	<i>Визуальный контроль</i>	
Оборудование		ПК	ПК	ПК		-
Единица измерения	Документ	Символ	Символ	Символ	Документ	-
Объем, Q	15796	296935	317734	600005	761	-
Среднечасовая норма выработки, Н (час ⁻¹)	6	2000	200000	20000	12	-
Трудоемкость, T _{ом} (час)	2632,67	148,47	1,59	30,00	63,42	2876,15
Среднечасовая тарифная ставка, Р (руб.)	50	50	50	50	50	-
Заработная плата оператора, С _з (руб.)	131633,50	7423,50	79,50	1500,00	3171,00	-
Прочие расходы, С _г (руб.)	92143,45	5196,45	55,65	1050,00	2219,70	-
Стоимость амортизации оборудования, С _а (руб.)	-	742,35	7,95	150,00	317,10	-
Операционные расходы, С _{ом} (руб.)	223776,95	13362,30	143,10	2700,00	5707,80	245690,15

Расчет затрат непосредственно на проектирование разрабатываемой АИС представлен в Таблице 3.4.

Таблица 3.4.

Расчет затрат на проектирование АИС

<i>№ n/n</i>	<i>Наименование затрат</i>	<i>Сумма, руб.</i>
1	Лицензия на использование системы разработки	28000,00
2	Стоимость оборудования (сервер базы данных, сервер, приложений, каналы связи)	715000,00
3	Фонд премий сотрудникам, участвующим в проекте	100000,00

Ежегодные расходы по статье «трудовые затраты» в новом варианте составляют:

$$T_1 = 2876,15 \text{ (человек/час);}$$

Ежегодные расходы по статье «стоимостные затраты» в новом варианте составляют:

$$C_1 = 245690,15 \text{ (руб.);}$$

На основании рассчитанных данных по формулам (1) – (6) рассчитаны ТЗ и СЗ, показанные в Таблице 3.5.

Таблица 3.5.

Показатели от внедрения новой АИС

	Затраты		Абсолютное изменение затрат	Коэффициент изменения затрат	Индекс изменения затрат
	Базовый вариант	Проектный вариант			
Трудовая мощность	T_0 (чел./час)	T_1 (чел./час)	$\Delta T = T_0 - T_1$ (чел./час)	$K_T = \Delta T / T_0 \times 100\%$	$Y_T = T_0 / T_1$
	7416,20	2876,15	4540,05	61,22%	2,579
Стоимость	C_0 (руб.)	C_1 (руб.)	$\Delta C = C_0 - C_1$ (руб.)	$K_C = \Delta C / C_0 * 100\%$	$Y_C = C_0 / C_1$
	3017796,7	245690,15	2772106,55	91,85	12,28

Срок окупаемости расходов на внедрение разрабатываемой АИС, рассчитывается по формуле (7) и составляет:

$$T_{OK} = \frac{K_N}{\Delta C} = \frac{843000 \text{ (руб.)}}{393055,23 \text{ (руб./год)}} \approx 2 \text{ года } 2 \text{ месяца}$$

Коэффициент ЭЭ капитальных затрат, рассчитанный по формуле (21) равен:

$$E_p = 1/2.2 = 0.45$$

Годовой экономический эффект был рассчитан по формуле (20) и составил:

$$\mathcal{E} = \Pi - K_{II} \times E_{II} = 393055,23 - 0,15 \times 843000 = 350905,23(\text{руб.})$$

На основании рассчитанных данных построены сравнительные диаграммы, представленные на рисунках (Рисунок 3.1 – Рисунок 3.4).

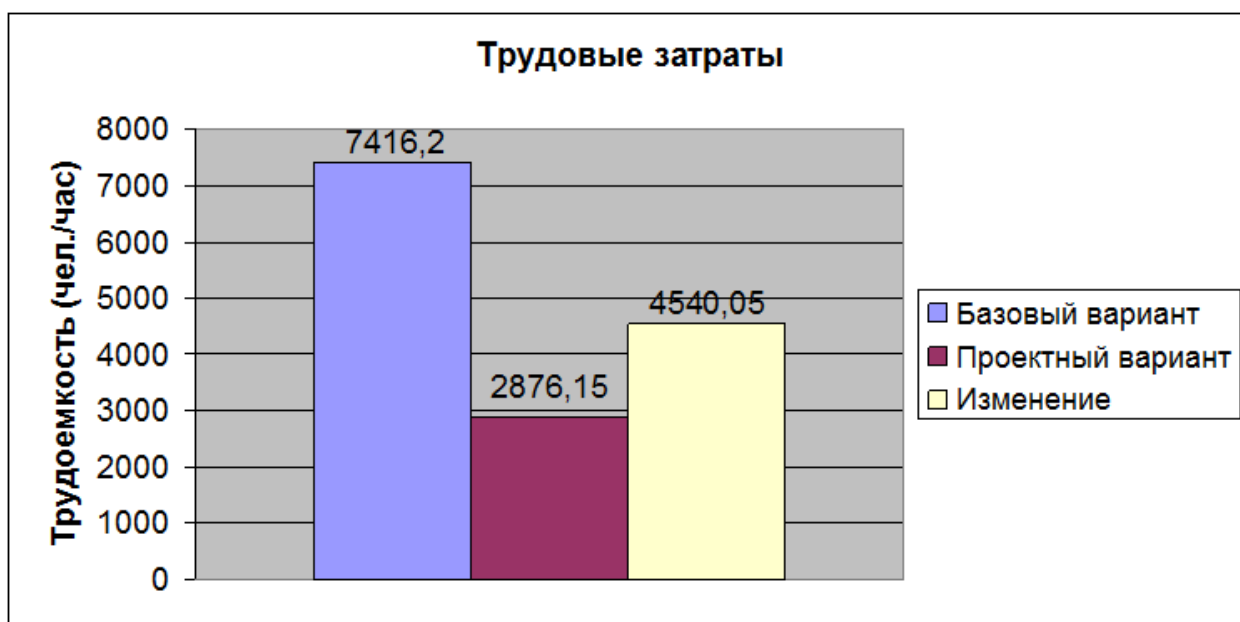


Рисунок 3.1. Диаграмма сравнения изменения ТЗ

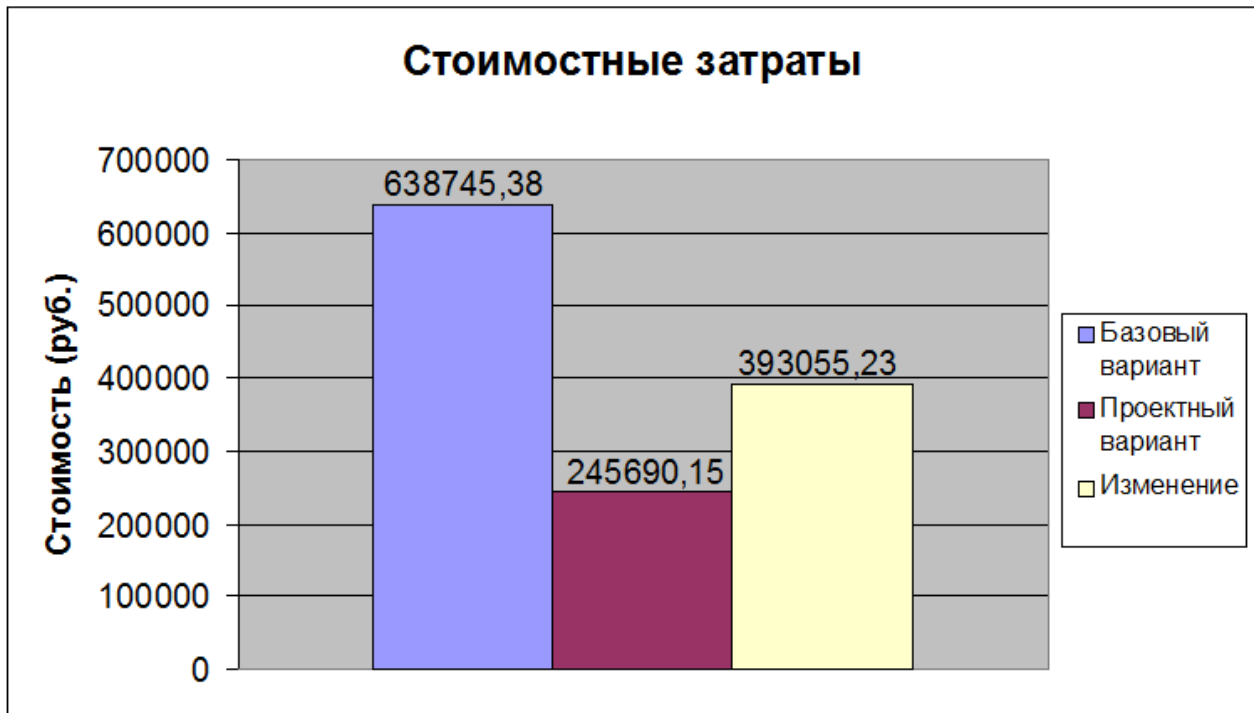


Рисунок 3.2. Диаграмма сравнения изменения СЗ

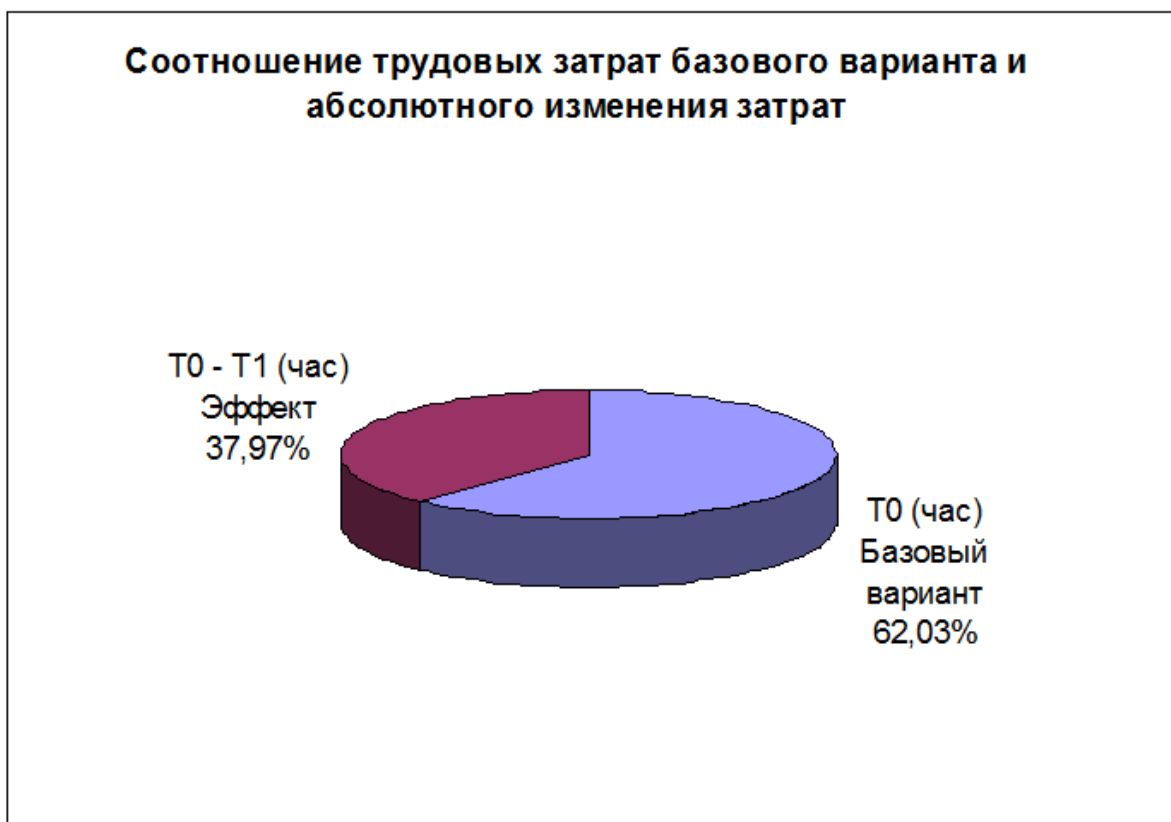


Рисунок 3.3. Соотношение ТЗ базового варианта и абсолютного изменения затрат

Соотношение стоимостных затрат базового варианта и абсолютного изменения затрат

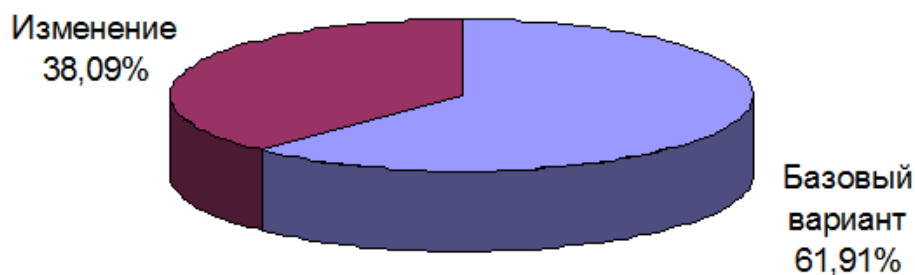


Рисунок 3.4. Соотношение СЗ базового варианта и абсолютного изменения затрат

Следует упомянуть, что кроме прямого эффекта от внедрения разрабатываемой АИС, присутствует также и косвенный эффект. Он выражается в том, что с точки зрения сотрудников работа с АИС становится в разы проще и понятнее, вследствие чего повышается эффективность работы сотрудников и снижается вероятность допущения ими ошибок.

Заключение

В результате выполнения исследования были изучены особенности деятельности Службы и основные протекающие в ней процессы. Особое внимание уделено деятельности Службы по приему заявок на устранение проблем, связанных с эксплуатацией ИТ-инфраструктуры компании, как основному направлению ее деятельности. Изучены основные виды проблем и особенности работы Службы.

Процесс прохождения заявки пользователя по решению какой-либо проблемы всесторонне изучен и проанализирован. Определены его этапы и участники, выявлены основные проблемы и недостатки.

Сделан вывод о том, что эффективность эксплуатации ИТ–инфраструктуры зависит от организации процесса решения и устранения проблем, возникающих в ходе работы с ИТ-ресурсами. В значительной мере повысить эту эффективность можно путем внедрения в Службе автоматизированной информационной системы приема и обработки заявок и сопровождения процесса их решения на протяжении всего жизненного цикла заявки. Был сделан вывод о необходимости разработки и внедрения такой системы.

Была поставлена основная задача разрабатываемой системы – автоматизация работы Службы в части учета заявок пользователей и их сопровождения. Определены требования к системе, разработана ее функциональная структура и функции каждой подсистемы. Был сделан вывод о необходимости разработки системы собственными силами.

В качестве среды разработки системы выбрана платформа «1С:Предприятие 8.3», как наиболее подходящая для целей системы. Рассмотрены основные особенности архитектуры этой системы, ее функциональность и механизмы, при помощи которых строятся прикладные решения. Найдены способы использования основных механизмов системы «1С:Предприятие 8.3» для целей дипломного проекта.

В среде разработки «1С:Предприятие 8.3» была реализована конфигурация, выполняющая поставленную задачу. Были созданы объекты конфигурации, определена их структура, реквизиты, экранные формы, написаны программные модули, реализующие заданную функциональность этих объектов. Особое внимание

уделено разработке структуры объектов информационной базы: справочников, регистров сведений, документов и регистров накопления.

Проведенное тестирование разработанной конфигурации показывает, что она полностью удовлетворяет поставленной задаче, работает стабильно и без ошибок, обладает большой функциональностью и удобным интерфейсом. Разработанную конфигурацию можно внедрять в Службе.

Внедрение разработанной системы в Службе должно резко увеличить эффективность процесса решения заявок пользователей, уменьшить время их прохождения, повысить производительность труда сотрудников.

Список использованных источников

1. ГОСТ Р ИСО/МЭК 12207-99 «Информационная технология. Процессы жизненного цикла программных средств» (принят и введен в действие постановлением Госстандарта РФ от 23.12.1999 №675-ст).
2. ГОСТ Р ИСО/МЭК 12207-99, Руководящий документ, Госстандарт России, Москва, 2014;
3. ГОСТ Р ИСО/МЭК 15271-2002 «Информационная технология. Руководство по применению ГОСТ Р ИСО/МЭК 12207-99 (Процессы жизненного цикла программных средств)» (принят и введен в действие постановлением Госстандарта РФ от 05.06 2012 №227-ст).
4. 101 совет начинающим разработчикам в системе "1С:Предприятие 8". Хрусталева Е. Ю. 214 с. 1С-Публишинг, 2014 г.
5. 1С:Предприятие 8.2. Практическое пособие разработчика. Примеры и типовые приемы (+диск). М.Г. Радченко, Е.Ю. Хрусталева. 968 с. 1С-Публишинг, 2014 г.
6. 1С:Предприятие 8.2. Коротко о главном. Новые возможности версии 8.2. Радченко М.Г. 690 с. 1С-Публишинг, 2014 г.
7. 1С:Предприятие 8.2. Практическое пособие разработчика. Примеры и типовые приемы. Радченко М.Г., Хрусталева Е.Ю. 620 с. 1С-Публишинг, 2012 г.
8. Арутюнов В.В. Теория экономических информационных систем: Московская финансово-промышленная академия. М., 2015. – 192 с.
9. Архитектура и работа с данными "1С:Предприятия 8.2". Серия «1С:Профессиональная разработка». Радченко М. Г., Хрусталева Е. Ю. 650 с. 1С-Публишинг, 2013 г.
10. Балдин К.В., Уткин В.Б. Информационные системы в экономике. М.:Издательско-торговая корпорация «Дашков и К», 2018. – 395 с.
11. Вендров А.М. CASE технологии Современные методы и средства проектирования информационных систем М.: Финансы и статистика, 2014. - 176 с.: ил.;
12. Вендров А.М. Проектирование программного обеспечения экономических информационных систем. – М.: Финансы и статистика, 2000.-187с.

13. Гагарина Л.Г., Киселев Д.В., Федотова Е.Л. Разработка и эксплуатация автоматизированных информационных систем / под ред. проф. Л.Г. Гагариной. – Москва: ИД «Форум»: Инфра-М, 2017 – 384 с.
14. Диго С.М. Проектирование и использование баз данных Учебник. М.: Финансы и статистика. 2013 г;
15. Ефимов Е.Н., Патрушина С.М., Панферова Л.Ф., Хашиева Л.И. Информационные системы в экономике. – Москва: ИКЦ «МарТ», 2014. – 352 с.
16. Информационные системы в экономике: Учебник для студ. высш. учеб. заведений / В.Б. Уткин, К.В. Балдин. — М.: Издательский центр «Академия», 2014. — 288 с.
17. Калянов Г.Н. CASE-технологии. Консалтинг при автоматизации предприятий.-М.: СИНТЕГ, 2015.-276с.
18. Котлер Ф. Маркетинг менеджмент / Пер. с англ. под ред. Л.А. Волковой, Ю.Н. Каптунеревского. – СПб.: Питер, 2012. – 756с.р
19. Липаев В.В. Программная инженерия. Методологические основы. М.: ТЕИС, 2016. – 608 с.
20. Максимов Н.В., Попов И.И., Голицына О.Л. Информационные системы. Учебное пособие. Форум, 2019 г. - 496 с.
21. Матищев А.Н. Эффективность рекламы. – М.: Издательство «Финпресс», 2012. – 416с.
22. Методология функционального моделирования IDEF0, Руководящий документ, Госстандарт России.;
23. Основы построения баз данных под ред. А.Д. Хомоненко Санкт-Петербург, 2014;
24. Петров В.Н. Информационные системы. – СПб.: Питер, 2003. – 688 с.
25. Проектирование экономических информационных систем: Учебник/Смирнова Г.Н., Сорокин А.А., Тельнов Ю.Ф.; Под ред. Тельнова Ю.Ф. – М.: Финансы и статистика, 2013. - 512 с.
26. Профессиональная разработка в системе «1С:Предприятие 8» (+DVD-ROM). Издание 2. Ажеронок В.А., Габец А.П., Гончаров Д.И., Козырев Д.В.,

- Кухлевский Д.С., Островерх А.В., Радченко М.Г., Хрусталева Е.Ю. Под редакцией М. Г. Радченко. 870 с. 1С-Паблишинг, 2013 г.
27. Разработка управляемого интерфейса (+CD). Серия "1С:Профессиональная разработка". Ажеронок В.А., Островерх А. В., Радченко М. Г., Хрусталева Е. Ю. 740 с. 1С-Паблишинг, 2015 г.
28. Реализация прикладных задач в системе "1С:Предприятие 8.2" (+CD). Серия "1С:Профессиональная разработка". Габец А.П., Козырев Д.В., Кухлевский Д.С., Хрусталева Е.Ю. 830 с. 1С-Паблишинг, 2014 г.
29. Репин В.В., Елиферов В.Г. Процессный подход к управлению. Моделирование бизнес-процессов. – М.: РИА «Стандарты и качество», 2014. – 408 с., илл. – (Серия «Практический менеджмент»)
30. Решение специальных прикладных задач в "1С:Предприятии 8.2". Серия "1С:Профессиональная разработка". Гончаров Д. И., Хрусталева Е. Ю. 935 с. 1С-Паблишинг, 2015 г.
31. 1С:Предприятие 8. Система программ [Электронный ресурс]. - <http://v8.1c.ru/>
32. IntraService. Универсальная Service Desk система [Электронный ресурс]. - <https://intraservice.ru/>
33. Новости и технологии торговли [Электронный ресурс]. – Режим доступа: <http://www.torgrus.com>
34. Программирование для всех [Электронный ресурс]. – Режим доступа: <http://www.realcoding.net/>

Приложение 1. Текст программных модулей

Модуль документа «Задача»

```
// префикс номера документа
Процедура ПриУстановкеНовогоНомера(СтандартнаяОбработка, Префикс)
    Префикс = Обмен.ПолучитьПрефиксНомера();
КонецПроцедуры

// при вводе нового документа
Процедура ОбработкаЗаполнения(Основание, Значение)

    ОбщийСрв.ПриВводеНовойЗадачи(ЭтотОбъект); // заполнить все реквизиты
по-умолчанию для новой задачи

    // если вводится на основании Контрагента
    Если ТипЗнч(Основание) = Тип("СправочникСсылка.Контрагенты") Тогда
        Контрагент = Основание; // основание - это
контрагент
        Ответственный = Основание.Ответственный; // ответственный по задаче -
ответственный по контрагенту
        Исполнитель = Исполнители.Добавить(); // первый исполнитель -
Исполнитель.Исполнитель = Основание.Ответственный;
        Исполнитель.Процент = 100;
    КонецЕсли;

    // если вводится на основании другой задачи
    Если ТипЗнч(Основание) = Тип("ДокументСсылка.Задача") Тогда
        ОбщийСрв.ПриВводеНовойЗадачи(ЭтотОбъект, Основание);
        ДокОснование = Документы.Задача.ПустаяСсылка();
    КонецЕсли;

    Если ТипЗнч(Основание) = Тип("Структура") Тогда //если вводлится новый
из списка с установленным отбором
        // то данные заполняются автоматически из отбора "Заполнять из
данных заполнения"
    КонецЕсли

КонецПроцедуры

// если копирование - то переопределить логику копирования
Процедура ПриКопировании(ОбъектКопирования)

    Автор = ПараметрыСеанса.глПользователь; // автор - текущий
пользователь
    Ответственный = ПараметрыСеанса.глПользователь; // автор - текущий
пользователь
    Статус = Справочники.Статусы.Новая; // новая
задача - статус Новая
```

```

    ДокОснование = ОбъектКопирования.Ссылка; // Док основание - та задача
из котрой скопировали
    Переписка.Очистить();
    Исполнители.Очистить();
    Подзадачи.Очистить();
    Номенклатура.Очистить();

```

КонецПроцедуры

```

// перед удалением документа
Процедура ПередУдалением(Отказ)

```

```

    // проверка на закрытый период
    Если ОбщийСрв.ВЗакрытомПериоде(ЭтотОбъект) Тогда
        Сообщить(ОбщийСрв.ПредставлениеЗадачи(ЭтотОбъект) +
"Находится в закрытом периоде. Удаление невозможно");
        Отказ = Истина;
        Возврат;
    КонецЕсли;

```

```

    // проверка Проведена
    Если (ЭтотОбъект.Проведен И Константы.ЗапретПроведенных.Получить()) И
Не ПараметрыСеанса.глЗаписьВедетИсполнитель Тогда
        Сообщить(ОбщийСрв.ПредставлениеЗадачи(ЭтотОбъект) + "Уже
выполнена. Удаление невозможно");
        Отказ = Истина;
        Возврат;
    КонецЕсли;

```

```

    // проверка прав
    Если Не ОбщийСрв.ПроверкаПрав(Автор,
ПараметрыСеанса.глПользователь) Тогда
        Сообщить(ОбщийСрв.ПредставлениеЗадачи(ЭтотОбъект) + "Нет прав
удалять чужой документ");
        Отказ = Истина;
        Возврат;
    КонецЕсли;

```

```

    // удалить вхождение этого документа в табличную часть Подзадачи любых
остальных документов
    ТЗРодителей = СформироватьСписокРодителей(ЭтотОбъект.Ссылка); //
сформировать ТЗ всех документов, в которые входит удаляемая
    Для Каждого СтрокаРодитель Из ТЗРодителей Цикл // для каждого родителя
        УдалитьПодзадачу(СтрокаРодитель.Ссылка, ЭтотОбъект.Ссылка); //
удалить текущую(удаляемую) задачу из родителя
    КонецЦикла

```

КонецПроцедуры

// удалить ЗадачаРебенок из табличной части Подзадачи у ЗадачаРодитель
Процедура УдалитьПодзадачу(ЗадачаРодитель, ЗадачаРебенок)

```
Если ЗадачаРодитель.Пустая() Тогда
    Возврат;
КонецЕсли;
ЗадачаРодительОбъект = ЗадачаРодитель.ПолучитьОбъект();
```

```
СтруктураОтбора = Новый Структура("Подзадача", ЗадачаРебенок);
МассивУдаляемыхСтрок
ЗадачаРодительОбъект.Подзадачи.НайтиСтроки(СтруктураОтбора);
Для Каждого ЭлементМассива Из МассивУдаляемыхСтрок Цикл
    ЗадачаРодительОбъект.Подзадачи.Удалить(ЭлементМассива);
КонецЦикла;
```

```
ЗадачаРодительОбъект.Записать();
КонецПроцедуры
```

// сформироват ТЗ документов, в которые ПодЗадачаРебенок входит в таб часть
Подзадачи (в явном виде, не через рекурсию)
Функция СформироватьСписокРодителей(ПодЗадачаРебенок)

```
Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
|     Задачи.Ссылка
|ИЗ
|     Документ.Задача.Подзадачи КАК ЗадачаПодзадачи
|         ЛЕВОЕ СОЕДИНЕНИЕ Документ.Задача КАК Задачи
|         ПО ЗадачаПодзадачи.Ссылка = Задачи.Ссылка
|ГДЕ
|     ЗадачаПодзадачи.Подзадача = &ПодзадачаРебенок
|";
```

```
Запрос.УстановитьПараметр("ПодзадачаРебенок", ПодзадачаРебенок);
```

```
РезультатЗапроса = Запрос.Выполнить();
ТЗ = РезультатЗапроса.Выгрузить();
```

```
Возврат ТЗ;
```

КонецФункции

// провести все непроведенные подзадачи из ЗадачаОбъект и провести ее саму
Процедура ВыполнитьВсеПодзадачиПринудительно(ЗадачаОбъект)

```

Для Каждого ТекСтрокаПодзадача Из ЗадачаОбъект.Подзадачи Цикл // по
всем строкам таб. части Подзадачи
    Если ТекСтрокаПодзадача.Подзадача.Проведен <> Истина Тогда //
если не была проведена

```

```

    ВыполнитьВсеПодзадачиПринудительно(ТекСтрокаПодзадача.Подзадача.По
лучитьОбъект()); // провести все ее подзадачи

```

```

    ТекСтрокаПодзадача.Подзадача.ПолучитьОбъект().Записать(РежимЗаписиД
окумента.Проведение, РежимПроведенияДокумента.Неоперативный); // провести ее
саму

```

```

        Сообщить("Выполнена подзадача " +
ОбщийСрв.ПредставлениеЗадачи(ТекСтрокаПодзадача.Подзадача));
        КонецЕсли;
    КонецЦикла;

```

КонецПроцедуры

```

// проверить, есть ли среди ТЧ Исполнители записи без отметки исполнения
Функция ПроверитьИсполнение()

```

```

    Исполнено = Истина;
    Для Каждого ТекСтрокаИсполнитель Из Исполнители Цикл
        Если ТекСтрокаИсполнитель.ДатаИсполнения =
Дата(01,01,0001,00,00,00) Тогда
            Сообщить("Задача не может быть выполнена, так как
исполнитель \" + ТекСтрокаИсполнитель.Исполнитель +
            "\ не проставил отметку исполнения");
            Исполнено = Ложь;
        КонецЕсли;
        Если Не ТекСтрокаИсполнитель.РезультатИсполнения.Пустая() Тогда
            Если Не ТекСтрокаИсполнитель.РезультатИсполнения.Успех
Тогда
                Сообщить("Задача не может быть выполнена, так как
исполнитель \" + ТекСтрокаИсполнитель.Исполнитель +
                "\ проставил отрицательную отметку
исполнения");
            Исполнено = Ложь;
        КонецЕсли;
    КонецЕсли;
КонецЦикла;

```

```

    Возврат Исполнено;
КонецФункции

```

```

// действия при проведении документа
Функция ПриПроведении(Отказ)

```

```

// проверка того, что все исполнители проставили отметку об исполнении
Если Константы.ПроверитьИсполнениеПриВыполнении.Получить() Тогда //
общая константа
    Если
        ОбщийСрв.ПолучитьКонстанту(ТекущаяДата(),
ОбщийСрв.ПолучитьГлПользователь(), "ПроверитьИсполнениеПриВыполнении")
Тогда // настройка пользователя
        Если Не ПроверитьИсполнение() Тогда
            Отказ = Истина;
            Возврат Ложь;
        КонецЕсли;
    КонецЕсли;
КонецЕсли;

// проверка того, что все подзадачи проведены
ЕстьНеВыполненныеПодзадачи = Ложь;
Для Каждого ТекСтрокаПодзадача Из Подзадачи Цикл
    Если ТекСтрокаПодзадача.Подзадача.Проведен <> Истина Тогда
        Сообщить(ОбщийСрв.ПредставлениеЗадачи(ЭтотОбъект) + "
имеет невыполненные подзадачи");
        ЕстьНеВыполненныеПодзадачи = Истина;
    КонецЕсли;
КонецЦикла;

Если ЕстьНеВыполненныеПодзадачи = Истина Тогда
    Если (Константы.ПринудительноВыполнятьПодзадачи.Получить()) И
// общая константа
        (ОбщийСрв.ПолучитьКонстанту(ТекущаяДата(),
ОбщийСрв.ПолучитьГлПользователь(), "ПринудительноВыполнятьПодзадачи"))
Тогда // настройка пользователя
        Сообщить("Подзадачи будут выполнены
принудительно...");
        ВыполнитьВсеПодзадачиПринудительно(ЭтотОбъект); //
автоматически провести все подзадачи данной задачи
        Иначе // нельзя автоматически выполнять подзадачи
            Сообщить(ОбщийСрв.ПредставлениеЗадачи(ЭтотОбъект) + " не
может быть выполнена");
            Отказ = Истина;
            Возврат Ложь;
        КонецЕсли;
    КонецЕсли;

    Возврат Истина
КонецФункции

// обработка проведения
Процедура ОбработкаПроведения(Отказ, РежимПроведения)

    Если Не ПриПроведении(Отказ) Тогда // проверить проведение подзадач

```

```

    Возврат;
КонецЕсли;

// регистр ИспользованиеНоменклатуры
Движение.ИспользованиеНоменклатуры.Записывать = Истина;
Для Каждого ТекСтрокаНоменклатура Из Номенклатура Цикл
    Если ТекСтрокаНоменклатура.Наличие Тогда // если в ТЧ стоит
признак наличия
        Продолжить;
    КонецЕсли;
    Если
        ТекСтрокаНоменклатура.Номенклатура.ВидНоменклатуры.Списывается Тогда // НЕ
если номенклатуру не надо списывать
        Продолжить;
    КонецЕсли;
    Движение = Движения.ИспользованиеНоменклатуры.Добавить();
    Движение.Период = Дата;
    Движение.Номенклатура =
ТекСтрокаНоменклатура.Номенклатура;
    Движение.Задача = ЭтотОбъект.Ссылка;
    Движение.Количество = ТекСтрокаНоменклатура.Количество;
    Движение.Сумма = ТекСтрокаНоменклатура.Сумма;
КонецЦикла;

КонецПроцедуры

// обработка распределения
Процедура ОбработкаУдаленияПроведения(Отказ)

    Если Не ОтменитьПроведениеРодителей(ЭтотОбъект.Ссылка) Тогда //
отменить проведение задач-родителей
        Отказ = Истина;
    Возврат;
КонецЕсли;

КонецПроцедуры

// отменить проведение всех родителей ДокСсылкаРебенок
Функция ОтменитьПроведениеРодителей(ДокСсылкаРебенок)

    МассивОтменяемых =
ОбщийСрв.СформироватьСписокОтменяемых(ДокСсылкаРебенок); //
сформировать массив распроеднимых документов
    Если МассивОтменяемых.Количество() Тогда // если есть хоть один -
сообщить
        Сообщить ("Отмена выполнения " +
ОбщийСрв.ПредставлениеЗадачи(ЭтотОбъект)
        + " приводит к отмене выполнения задач-родителей...");

```

КонецЕсли;

Для Каждого СтрокаОтменяемый Из МассивОтменяемых Цикл // для каждого документа из распроектируемых

Если ОбщийСрв.ВЗакрытомПериоде(СтрокаОтменяемый.Значение) Тогда // если находится в закрытом периоде

Сообщить("Отмена выполнения " +
ОбщийСрв.ПредставлениеЗадачи(ДокСсылкаРебенок)

+ " приводит к отмене выполнения " +
ОбщийСрв.ПредставлениеЗадачи(СтрокаОтменяемый.Значение)

+ ", находящейся в закрытом периоде. Отмена не может быть выполнена");

Возврат Ложь;

КонецЕсли;

Если

ОбщийСрв.ПроверкаПрав(СтрокаОтменяемый.Значение.Автор,
ПараметрыСеанса.глПользователь) Тогда // проверка прав

Сообщить("Отмена выполнения " +
ОбщийСрв.ПредставлениеЗадачи(ДокСсылкаРебенок)

+ " приводит к отмене выполнения " +
ОбщийСрв.ПредставлениеЗадачи(СтрокаОтменяемый.Значение)

+ ", на редактирование которого у Вас нет прав. Отмена не может быть выполнена");

Возврат Ложь;

КонецЕсли;

КонецЦикла;

// распроектировать каждый

Для Каждого СтрокаОтменяемый Из МассивОтменяемых Цикл

Если СтрокаОтменяемый.Значение.ПолучитьОбъект().Проведен Тогда

СтрокаОтменяемый.Значение.ПолучитьОбъект().Записать(РежимЗаписиДокумента.ОтменаПроведения);

Сообщить("Отменено выполнение родительской " +
ОбщийСрв.ПредставлениеЗадачи(СтрокаОтменяемый.Значение));

КонецЕсли;

КонецЦикла;

Возврат Истина;

КонецФункции

// перед любой записью (включая проведение и распроектирование) - проверка Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения)

Головная = ЭтоГоловная(); // установить признак - головная или нет

Если ОбщийСрв.ВЗакрытомПериоде(ЭтотОбъект) Тогда // проверка на закрытый период

```

Сообщить(ОбщийСрв.ПредставлениеЗадачи(ЭтотОбъект)
"Находится в закрытом периоде. Редактирование невозможно");
Отказ = Истина;
Возврат;
КонецЕсли;

// проверка того что проведен (тогда редактирование запрещено)
Если Не (РежимЗаписи = РежимЗаписиДокумента.Проведение Или
РежимЗаписи = РежимЗаписиДокумента.ОтменаПроведения) Тогда
    Если (ЭтотОбъект.Проведен И
Константы.ЗапретПроведенных.Получить()) И Не
ПараметрыСеанса.глЗаписьВедетИсполнитель
    И Не ПараметрыСеанса.ИдетОбменДанными Тогда
        Сообщить(ОбщийСрв.ПредставлениеЗадачи(ЭтотОбъект)
"Уже выполнена. Редактирование невозможно");
Отказ = Истина;
Возврат;
КонецЕсли;
КонецЕсли;

// проверка прав
Если Не ОбщийСрв.ПроверкаПрав(Автор,
ПараметрыСеанса.глПользователь) Тогда
    Сообщить(ОбщийСрв.ПредставлениеЗадачи(ЭтотОбъект) + "Нет прав
редактировать чужой документ");
Отказ = Истина;
Возврат;
КонецЕсли;

Если РежимЗаписи = РежимЗаписиДокумента.Проведение Тогда
    РеальнаяДатаВыполнения = ТекущаяДата(); // дата реального
выполнения - дата проведения
    Если Константы.УстанавливатьЗавершенаУспешно.Получить() =
Истина Тогда
        Статус = Справочники.Статусы.ЗавершенаУспешно; // статус -
успешно
        КонецЕсли;
    КонецЕсли;

Если РежимЗаписи = РежимЗаписиДокумента.ОтменаПроведения Тогда
    РеальнаяДатаВыполнения = 0; // если распрвели - значит нет
реальной даты завершения
    Статус = Справочники.Статусы.ПустаяСсылка(); // и нет статуса
КонецЕсли;

КонецПроцедуры

```


// проверка на то, что данная задача не входит ни в одну другую в качестве подзадачи
 Функция ЭтоГоловная()

```

Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
|     ЗадачаПодзадачи.Подзадача
|ИЗ
|     Документ.Задача.Подзадачи КАК ЗадачаПодзадачи
|     ГДЕ ЗадачаПодзадачи.Подзадача     = &ИскомаяЗадача
|";

Запрос.УстановитьПараметр("ИскомаяЗадача", ЭтотОбъект.Ссылка);
Результат = Запрос.Выполнить();

Возврат (Результат.Выбрать().Количество() = 0);

```

КонецФункции

Форма документа «Задача»

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

```

// ЗАКЛАДКА ПОДЗАДАЧИ
Если Параметры.Свойство("ЭтоНовый") Тогда // если из вызывающей формы
был передан признак что это новый документ
    Если Параметры.ЭтоНовый Тогда // значит нажали кнопку Новая
подзадача в тч Подзадачи родителя
        Основание = Параметры.Родитель;
        Если Не Основание.Пустая() Тогда
            ОбщийСрв.ПриВводеНовойЗадачи(Объект, Основание,
Ложь); // перезаполнить реквизиты и тч новой на основании родителя

                // но не добавлять в тч Подзадачи
                // далее см. в ПослеЗаписи этой задачи - добавить эту
новую в тч Подзадачи родителя
                // далее см. в ОбработкаОповещения родителя -
спозиционироваться на вновь добавленной задче в дереве
        КонецЕсли;
    КонецЕсли;
КонецЕсли;
ФильтрПроведенные = 2; // первоначальное значение фильтра "Все"
ЗаполнитьДеревоПодзадач(); // первоначально создать дерево подзадач

```

КонецПроцедуры

&НаКлиенте

Процедура ПриОткрытии(Отказ)

```
// если нет прав - то форма только просмотр
Если Не ОбщийСрв.ПроверкаПрав(Объект.Автор,
ОбщийСрв.ПолучитьГлПользователь()) Тогда
    ЭтаФорма.ТолькоПросмотр = Истина;
КонецЕсли;
```

```
// если проведена и запрет проведенных
Если Объект.Проведен И
ОбщийСрв.обПолучитьКонстанту("ЗапретПроведенных") Тогда
    ЭтаФорма.ТолькоПросмотр = Истина;
КонецЕсли;
```

// ЗАКЛАДКА ИСПОЛНИТЕЛИ

```
// если пользователь является исполнителем этого документа и он открыл
документ - то считается что он с ним ознакомлен -
// надо проставить отметку об ознакомлении
ИсполнительОзнакомлен = Ложь;
Для Каждого СтрокаИсполнитель Из Объект.Исполнители Цикл
    Если (СтрокаИсполнитель.Исполнитель =
ОбщийСрв.ПолучитьГлПользователь()) И (СтрокаИсполнитель.Ознакомлен =
глПустаяДата) Тогда
        СтрокаИсполнитель.Ознакомлен = ТекущаяДата(); //
простановка даты ознакомления - и есть признак ознакомления
        ИсполнительОзнакомлен = Истина;
        ОбщийСрв.ЗаписьВедетИсполнитель(Истина); // признак того
что дальнейшая запись будет вестись исполнителем
        Прервать;
    КонецЕсли;
КонецЦикла;
Если ИсполнительОзнакомлен = Истина Тогда // если была простановка даты
ознакомления
    П = Новый Структура;
    П.Вставить("РежимЗаписи", РежимЗаписиДокумента.Запись);
    Записать(П);
КонецЕсли;
```

// ЗАКЛАДКА ПОДЗАДАЧИ

```
ВывестиПанельПодзадач(); // первоначальное заполнение дерева подзадач
// РодительскаяЗадача =
ОбщийСрв.НайтиРодительскуюЗадачу(Объект.Ссылка);
```

КонецПроцедуры

```
// реакция на команду формы Записать и закрыть
&НаКлиенте
```

Процедура ФормаЗаписатьИЗаккрыть(Команда)

Записать();

Заккрыть();

КонецПроцедуры

// реакция на команду формы Распрровести

&НаКлиенте

Процедура ФормаРаспрровести(Команда)

П = Новый Структура;

П.Вставить("РежимЗаписи", РежимЗаписиДокумента.ОтменаПроведения);

Записать(П);

КонецПроцедуры

// реакция на окончание выбора в других формах

&НаКлиенте

Процедура ОбработкаВыбора(ВыбранноеЗначение, ИсточникВыбора)

// результат пришел из формы Исполнение исполнителем

Если ИсточникВыбора.ИмяФормы =

"Документ.Задача.Форма.ФормаИсполнения" Тогда

ЭтотОбъект.Элементы.Исполнители.ТекущиеДанные.РезультатИсполнения
= ВыбранноеЗначение.Получить(0).Значение;

ЭтотОбъект.Элементы.Исполнители.ТекущиеДанные.ДатаИсполнения

= ВыбранноеЗначение.Получить(1).Значение;

ЭтотОбъект.Элементы.Исполнители.ТекущиеДанные.Комментарий

= ВыбранноеЗначение.Получить(2).Значение;

ОбщийСрв.ЗаписьВедетИсполнитель(Истина); // флаг того, что
записывать можно, даже если нет прав

П = Новый Структура;

П.Вставить("РежимЗаписи", РежимЗаписиДокумента.Запись);

ЭтотОбъект.Записать(П);

КонецЕсли;

// результат пришел из формы выбора справочника Этапы

Если ИсточникВыбора.ИмяФормы = "Справочник.Этапы.ФормаВыбора"

Тогда

ЗаполнитьДеревоПоШаблону(ВыбранноеЗначение);

КонецЕсли;

КонецПроцедуры

// проверить возможность записи документа из формы

&НаКлиенте

Процедура ПередЗаписью(Отказ, ПараметрыЗаписи)

Если ПараметрыЗаписи.РежимЗаписи = РежимЗаписиДокумента.Проведение
Тогда

Отказ = Не ПроверкаВозможностиПроведения());
КонецЕсли;

Если ПараметрыЗаписи.РежимЗаписи =
РежимЗаписиДокумента.ОтменаПроведения Тогда
Отказ = Не ПроверкаВозможностиРасПроведения());
КонецЕсли
КонецПроцедуры

&НаКлиенте

Функция ПроверкаВозможностиРасПроведения(ЭтаЗадача = Неопределено)

Если ЭтаЗадача = Неопределено Тогда
ЭтаЗадача = Объект.Ссылка;
КонецЕсли;

// задать пользователю вопрос
//МассивОтменяемых =
ОбщийСрв.СформироватьСписокОтменяемых(ЭтаЗадача);
//Если МассивОтменяемых.Количество() Тогда
// Если Вопрос("Отмена выполнения задачи " +
ОбщийСрв.ПредставлениеЗадачи(ЭтаЗадача)
// + " приведет к отмене выполнения задач-родителей. Все
равно отменить?", РежимДиалогаВопрос.ДаНет) = КодВозвратаДиалога.Нет Тогда

// Возврат Ложь;
// КонецЕсли;
//КонецЕсли;

Возврат Истина;

КонецФункции

Функция ПроверкаВозможностиПроведения(ЭтаЗадача = Неопределено)

Если ЭтаЗадача = Неопределено Тогда
ЭтаЗадача = Объект.Ссылка;
КонецЕсли;

// задать пользователю вопрос

Возврат Истина;

КонецФункции

// ЗАКЛАДКА СВОЙСТВА

&НаКлиенте

Процедура СвойстваВидСвойстваОбработкаВыбора(Элемент, ВыбранноеЗначение, СтандартнаяОбработка)

// проверка соответствия типа выбранного свойства

Если

Не

СвойстваВидСвойстваОбработкаВыбораНаСервере(ВыбранноеЗначение) Тогда

Сообщить("Тип свойства выбранного вида свойств не относится к задачам");

СтандартнаяОбработка = Ложь;

Возврат;

КонецЕсли;

КонецПроцедуры

&НаСервереБезКонтекста

Функция СвойстваВидСвойстваОбработкаВыбораНаСервере(ВыбВидСвойства)

Если ВыбВидСвойства.ПолучитьОбъект().ТипСвойства.ТипОбъекта ◇
Перечисления.ТипыОбъектов.Задача Тогда

Возврат Ложь;

КонецЕсли;

Возврат Истина;

КонецФункции

//-----

// ЗАКЛАДКА ПЕРЕПИСКА

// выход из поля ввода текста сообщения - это команда на добавление сообщения

&НаКлиенте

Процедура НовоеСообщениеОкончаниеВводаТекста(Элемент, Текст, ДанныеВыбора, ПараметрыПолученияДанных, СтандартнаяОбработка)

Если ПустаяСтрока(Текст) Тогда

Возврат;

КонецЕсли;

ДобавитьНовоеСообщение(Текст);

КонецПроцедуры

// программное добавление сообщения

&НаКлиенте

Процедура ДобавитьНовоеСообщение(ТекстСообщения)

Если Объект.Ссылка.Пустая() Тогда

Сообщить("Нужно сначала записать новую задачу");

Возврат;
КонецЕсли;

НоваяСтрока = Объект.Переписка.Добавить();
НоваяСтрока.Пользователь = ОбщийСрв.ПолучитьГлПользователь();
НоваяСтрока.ДатаСообщения = ТекущаяДата();
НоваяСтрока.Сообщение = ТекстСообщения;

ОбщийСрв.ЗаписьВедетИсполнитель(Истина); // флаг того, что записывать
можно, даже если нет прав
Записать();

КонецПроцедуры

//-----
// ЗАКЛАДКА ИСПОЛНИТЕЛИ

// по команде Исполнить
&НаКлиенте
Процедура ИсполнителиИсполнить(Команда)

Если
ОбщийСрв.ПроверкаПрав(ЭтотОбъект.Элементы.Исполнители.ТекущиеДанные.Ис
полнитель, ОбщийСрв.ПолучитьГлПользователь()) Тогда
 П = Новый Структура;
 П.Вставить("Исполнитель",
ЭтотОбъект.Элементы.Исполнители.ТекущиеДанные.Исполнитель);
 П.Вставить("РезультатИсполнения",
ЭтотОбъект.Элементы.Исполнители.ТекущиеДанные.РезультатИсполнения);
 П.Вставить("ДатаИсполнения",
ЭтотОбъект.Элементы.Исполнители.ТекущиеДанные.ДатаИсполнения);

 П.Вставить("Комментарий",
ЭтотОбъект.Элементы.Исполнители.ТекущиеДанные.Комментарий);

 ФормаИсполнение =
ПолучитьФорму("Документ.Задача.Форма.ФормаИсполнения", П, ЭтотОбъект);
 ФормаИсполнение.ОткрытьМодально();
 // дальше см. в ОбработкаВыбора формы
Иначе
 Сообщить("У вас нет прав для проставления отметки об исполнении");
КонецЕсли;

КонецПроцедуры

// перед записью в ТЧ Исполнители - не должно быть пустых и проверка процентов
&НаКлиенте

Процедура ИсполнителиПередОкончаниемРедактирования(Элемент, НоваяСтрока, ОтменаРедактирования, Отказ)

Если Элемент.ТекущиеДанные.Исполнитель.Пустая() И Не (НоваяСтрока и ОтменаРедактирования) Тогда

Сообщить("Исполнитель должен быть обязательно выбран");

Отказ = Истина;

КонецЕсли;

Процентов = 0;

Для Каждого СтрокаИсполнитель Из Объект.Исполнители Цикл

Процентов = Процентов + СтрокаИсполнитель.Процент;

КонецЦикла;

Если Процентов > 100 Тогда

Сообщить("Суммарный процент исполнения не может превышать 100%");

Отказ = Истина;

КонецЕсли;

КонецПроцедуры

// выбор нового исполнителя - проверка что такого еще не было

&НаКлиенте

Процедура ИсполнителиИсполнительОбработкаВыбора(Элемент, ВыбранноеЗначение, СтандартнаяОбработка)

Если УжеЕстьВТаблице(ВыбранноеЗначение) Тогда

СтандартнаяОбработка = Ложь;

Сообщить("Этот исполнитель уже выбран");

КонецЕсли;

КонецПроцедуры

&НаСервере

Функция УжеЕстьВТаблице(Выбор)

МассивНайденныхСтрок = Объект.Исполнители.НайтиСтроки(Новый Структура("Исполнитель",Выбор));

Возврат МассивНайденныхСтрок.Количество();

КонецФункции

//ввод новой строки в ТЧ Исполнители - запрет копирования (не может быть двух одинаковых строк)

&НаКлиенте

Процедура ИсполнителиПередНачаломДобавления(Элемент, Отказ, Копирование, Родитель, Группа, Параметр)

Если Копирование = Истина Тогда // запрет копирования

Отказ = Истина;

КонецЕсли;
КонецПроцедуры

//ввод новой строки в ТЧ Исполнители - автоматически рассчитать процент
&НаКлиенте

Процедура ИсполнителиПриНачалеРедактирования(Элемент, НоваяСтрока,
Копирование)

Если НоваяСтрока = Истина Тогда
Процентов = 0;
Для Каждого СтрокаИсполнитель Из Объект.Исполнители Цикл
Процентов = Процентов + СтрокаИсполнитель.Процент;
КонецЦикла;
Если Процентов > 100 Тогда
Процентов = 100;
КонецЕсли;
СтрокаИсполнитель = Элемент.ТекущиеДанные;
СтрокаИсполнитель.Процент = 100-Процентов;
СтрокаИсполнитель.Задание = Объект.Наименование;
КонецЕсли;

КонецПроцедуры

//-----

// ЗАКЛАДКА ПОДЗАДАЧИ
&НаКлиенте

Процедура ВывестиПанельПодзадач() // первоначальное заполнение дерева
подзадач

ЭтотОбъект.Элементы.ДеревоПодзадач.НачальноеОтображениеДерева =
НачальноеОтображениеДерева.РаскрыватьВерхнийУровень;
КонецПроцедуры

// заполнить дерево подзадач
&НаСервере

Процедура ЗаполнитьДеревоПодзадач()

ЭтаФорма.Прочитать(); // перечитать данные формы
ДеревоОбъект = РеквизитФормыВЗначение("ДеревоПодзадач");
ДеревоОбъект.Строки.Очистить(); // удалить все подзадачи

// сформировать корень - текущая задача

Голова = ДеревоОбъект.Строки.Добавить();

Голова.ПодЗадача = Объект.Ссылка;

Голова.Выполнена = Объект.Проведен;

Голова.Наименование = "ЭТА ЗАДАЧА";

Голова.Пометка = УстановитьПометку(Объект); //

отметка исполнения


```

// по всем подзадачам табличной части
Для Каждого СтрокаПодЗадача Из Объект.Подзадачи Цикл
    ЗаполнитьПодЗадачу(СтрокаПодЗадача, Голова); // вызвать для корня и
далее для каждой строки дерева
    КонецЦикла;

```

```

ЗначениеВРеквизитФормы(ДеревоОбъект, "ДеревоПодзадач");

```

КонецПроцедуры

```

// рекурсивное заполнение одной подзадачи

```

```

&НаСервере

```

```

Процедура ЗаполнитьПодЗадачу(СтрокаПодЗадача, Узел)

```

```

    // фильтр по проведенным и не проведенным
    Если (ФильтрПроведенные = 0) Или (ФильтрПроведенные = 1) Тогда
        Если Не Число(СтрокаПодЗадача.ПодЗадача.Проведен) =
ФильтрПроведенные Тогда
            Возврат;
        КонецЕсли;
    КонецЕсли;
    // // фильтр по текущему значению
    //Если ФильтрЗначение <> Неопределено Тогда
    //    Если ФильтрКолонка = "Наименование" Тогда
    //        Если Найти(СтрокаПодЗадача.ПодЗадача.Наименование,
ФильтрЗначение) = 0 Тогда
            //            Возврат;
            //            КонецЕсли;
            //        ИначеЕсли ФильтрКолонка = "Комментарий" Тогда
            //            Если Найти(СтрокаПодЗадача.Комментарий, ФильтрЗначение) =
0 Тогда
                //                Возврат;
                //                КонецЕсли;
                //            ИначеЕсли ФильтрКолонка = "Описание" Тогда
                    //                Если Найти(СтрокаПодЗадача.ПодЗадача.КраткоеОписание,
ФильтрЗначение) = 0 Тогда
                        //                    Возврат;
                        //                    КонецЕсли;
                        //                ИначеЕсли ФильтрКолонка = "Статус" Тогда
                        //                    Если СтрокаПодЗадача.ПодЗадача.Статус <> ФильтрЗначение
Тогда
                            //                            Возврат;
                            //                            КонецЕсли;
                            //                        ИначеЕсли ФильтрКолонка = "ДатаПостановки" Тогда
                            //                            Если НачалоДня(СтрокаПодЗадача.ПодЗадача.Дата) <>
НачалоДня(ФильтрЗначение) Тогда

```

```

//          Возврат;
//          КонецЕсли;
//          ИначеЕсли ФилтpКолонка = "ДатаОжидаемая" Тогда
//          Если НачалоДня(СтрокаПодЗадача.ПодЗадача.ДатаОкончания)
<> НачалоДня(ФилтpЗначение) Тогда
//          Возврат;
//          КонецЕсли;
//          КонецЕсли;
//КонецЕсли;

// создать новый узел дерева
НовыйУзел = Узел.Строки.Добавить();
НовыйУзел.ПодЗадача = СтрокаПодЗадача.ПодЗадача;
НовыйУзел.Выполнена =
СтрокаПодЗадача.ПодЗадача.Проведен;
НовыйУзел.НомерВТабЧасти = СтрокаПодЗадача.НомерСтроки;
НовыйУзел.Наименование =
Строка(СтрокаПодЗадача.ПодЗадача.Номер) + " " +
СтрокаПодЗадача.ПодЗадача.Наименование;
НовыйУзел.Статус = СтрокаПодЗадача.ПодЗадача.Статус;

НовыйУзел.Дата =
СтрокаПодЗадача.ПодЗадача.Дата;
НовыйУзел.ДатаОжидаемая =
СтрокаПодЗадача.ПодЗадача.ОжидаемаяДатаВыполнения;
НовыйУзел.Комментарий =
СтрокаПодЗадача.Комментарий;
НовыйУзел.Пометка =
УстановитьПометку(СтрокаПодЗадача.ПодЗадача);

//НовыйУзел.Картинка = БиблиотекаКартинок.ДебетКредит;

// теперь рекурсивно для каждого поддрева
Для Каждого СтрокаПодЗадача Из СтрокаПодЗадача.ПодЗадача.Подзадачи
Цикл
    ЗаполнитьПодЗадачу(СтрокаПодЗадача, НовыйУзел);
КонецЦикла;

КонецПроцедуры // ЗаполнитьПодЗадачу

// признак выполнения задачи
Функция УстановитьПометку(Задача)

    Возврат Число(Задача.Проведен);
//Если Задача.Проведен Тогда
//    Возврат 1;
//КонецЕсли;
//

```

```

//Для Каждого СтрокаПодЗадача Из Задача.Подзадачи Цикл
//    Если НЕ Задача.Проведен Тогда
//        Возврат 2;
//    КонецЕсли;
//КонецЦикла;
//
//Возврат 0;
КонецФункции

//пройти по массиву Путь (хранит ссылки на подзадачи) начиная с головы и вернуть
ИД последнего узла
&НаКлиенте
Функция СпозиционироватьсяВДеревеПодзадач(ВДереве, Путь, Й)

    Если Путь.Количество() = 0 Тогда
        Возврат 0;
    КонецЕсли;
    Ключ = Путь[Й];
    Для Каждого ТекущаяСтрока Из Вдереве.ПолучитьЭлементы() Цикл
        Если ТекущаяСтрока.Подзадача = Ключ Тогда
            Если Й = 0 Тогда
                Возврат ТекущаяСтрока.ПолучитьИдентификатор();
            Иначе
                Возврат СпозиционироватьсяВДеревеПодзадач(ТекущаяСтрока, Путь,
Й-1);
            КонецЕсли;
        КонецЕсли;
    КонецЦикла;

КонецФункции

// перехват выбора (двойной щелчок) - открыть текущую подзадачу
&НаКлиенте
Процедура ДеревоПодзадачВыбор(Элемент,      ВыбраннаяСтрока,      Поле,
СтандартнаяОбработка)

    Если Не ((Поле.Имя = "ДеревоПодзадачКомментарий") Или (Поле.Имя =
"ДеревоПодзадачПодзадача")) Тогда // эти поля - для собственного редактирования
        ОткрытьПодзадачу();
    КонецЕсли;

КонецПроцедуры

// перехват стандартной команды Добавить - добавить в дерево существующую
задачу
// открыть форму выбора и после выбора добавить выбранную задачу в дерево
&НаКлиенте

```

Процедура ДеревоПодзадачПередНачаломДобавления(Элемент, Отказ,
Копирование, Родитель, Группа, Параметр)

```
Отказ = Истина;
Если Не Копирование Тогда // копирование невозможно
    ДобавитьСуществующуюПодзадачу();
КонецЕсли;
КонецПроцедуры
```

```
// перехват стандартной команды Изменить - открыть подзадачу
&НаКлиенте
Процедура ДеревоПодзадачПередНачаломИзменения(Элемент, Отказ)
```

```
    Если Не ((Элемент.ТекущийЭлемент.Имя = "ДеревоПодзадачКомментарий")
Или (Элемент.ТекущийЭлемент.Имя = "ДеревоПодзадачПодзадача")) Тогда // эти
поля - для собственного редактирования
        Отказ = Истина;
        ОткрытьПодзадачу();
    КонецЕсли;
```

КонецПроцедуры

```
// перехват стандартной команды Удалить
&НаКлиенте
Процедура ДеревоПодзадачПередУдалением(Элемент, Отказ)
```

```
    УзелРодитель =
Элементы.ДеревоПодзадач.ТекущиеДанные.ПолучитьРодителя(); // получить узел-
родитель
    Если УзелРодитель = Неопределено Тогда // если нет родителя - то значит
пытаются удалить корень
        Отказ = Истина;
        Возврат;
    КонецЕсли;
```

```
    ПутьВДереве = ПолучитьПутьВДереве(УзелРодитель); //запомнить путь до
текущего узла
    УдалитьПодзадачуНаСервере(УзелРодитель.Подзадача,
Элементы.ДеревоПодзадач.ТекущиеДанные.ПодЗадача); // удалить из таб части
ЗаполнитьДеревоПодзадач(); // перестроить дерево подзадач
Элементы.ДеревоПодзадач.ТекущаяСтрока =
СпозиционироватьсяВДеревеПодзадач(ДеревоПодзадач, ПутьВДереве,
ПутьВДереве.Количество()-1); // спозиционироваться на добавленной задаче
```

```
    Оповестить("Дерево", Новый Структура,ЭтотОбъект); // всем формам послать
оповещение о проведенном изменении - пусть они перстаивают свое дерево
```

```
Отказ = Истина; // уже все сделано
```

КонецПроцедуры

// перехват стандартного Изменения (изменение последовательность строк или комментарий)

&НаКлиенте

Процедура ДеревоПодзадачПриИзменении(Элемент)

СохранитьДеревоНаСервере();

П = Новый Структура;

Оповестить("Дерево", П, ЭтотОбъект); // оповестить об изменении, сделанном в этом форме, все остальные формы - пусть они перстаривают свое дерево

КонецПроцедуры

// по команде НоваяПодзадача

&НаКлиенте

Процедура ДеревоНоваяПодзадача(Команда)

ЭтаЗадача = Элементы.ДеревоПодзадач.ТекущиеДанные.ПодЗадача;

//// вариант программного создания и записи нового документа

//НоваяПодЗадача = СоздатьПодзадачуНаСервере(); // создать новый документ и записать его

//ПутьВДереве

ПолучитьПутьВДереве(Элементы.ДеревоПодзадач.ТекущиеДанные, НоваяПодЗадача); //сформировать массив подзадач от текущего узла до корня

//ДобавитьСуществующуюПодзадачуНаСервере(ЭтаЗадача, НоваяПодЗадача); // добавить новую задачу в объект (в таб. часть)

//ЭтаФорма.Прочитать(); // перечитать данные формы

//ЗаполнитьДеревоПодзадач(); // перстроить дерево подзадач

//СпозиционироватьсяВДеревеПодзадач(ДеревоПодзадач, ПутьВДереве, ПутьВДереве.Количество()-1); // спозиционироваться на добавленной задаче

//П = Новый Структура;

//П.Вставить("Ключ", НоваяПодЗадача);

//ОткрытьФорму("Документ.Задача.ФормаОбъекта", П); // открыть отдельную форму редактирования новой задачи

Если ОбщийСрв.ВЗакрытомПериоде(ЭтаЗадача) Тогда

Сообщить(ОбщийСрв.ПредставлениеЗадачи(ЭтаЗадача) + "Находится в закрытом периоде. Редактирование невозможно");

Возврат;

КонецЕсли;

Если

ОбщийСрв.ПроверкаПрав(ОбщийСрв.ПолучитьРеквизитЗадачи(ЭтаЗадача, "Автор"), ОбщийСрв.ПолучитьГлПользователь()) Тогда

Не

Сообщить(ОбщийСрв.ПредставлениеЗадачи(ЭтаЗадача) + "Нет прав редактировать чужой документ");

Возврат;

КонецЕсли;

Если (ОбщийСрв.ПолучитьРеквизитЗадачи(ЭтаЗадача, "Проведен")) И
(ОбщийСрв.обПолучитьКонстанту("ЗапретПроведенных")) Тогда
Сообщить(ОбщийСрв.ПредставлениеЗадачи(ЭтаЗадача) + "Уже
выполнена. Редактирование невозможно");

Возврат;

КонецЕсли;

// вариант открытия формы нового пустого документа

П = Новый Структура;

П.Вставить("ЭтоНовый", Истина); // признак того что форма открывается для
нового документа

П.Вставить("Родитель", ЭтаЗадача);

ОткрытьФорму("Документ.Задача.ФормаОбъекта", П, ЭтотОбъект); //
открыть отдельную форму редактирования новой задачи

// далее см. в ПриСозданииНаСервере - заполнение реквизитов и таб. частей
нового документа на основании этого

// далее см. в ПослеЗаписи - запись новой задачи в тч Подзадачи родительской
задачи

// далее см. в ОбработкаОповещения - в родительской задаче в дереве
спозиционироваться на новой задаче

КонецПроцедуры

&НаКлиенте

Процедура ЗаполнитьДеревоПоШаблону(Шаблон)

ЭтаЗадача = Элементы.ДеревоПодзадач.ТекущиеДанные.ПодЗадача;

Если ОбщийСрв.ВЗакрытомПериоде(ЭтаЗадача) Тогда
Сообщить(ОбщийСрв.ПредставлениеЗадачи(ЭтаЗадача) + "Находится
в закрытом периоде. Редактирование невозможно");

Возврат;

КонецЕсли;

Если Не
ОбщийСрв.ПроверкаПрав(ОбщийСрв.ПолучитьРеквизитЗадачи(ЭтаЗадача,
"Автор"), ОбщийСрв.ПолучитьГлПользователь()) Тогда

Сообщить(ОбщийСрв.ПредставлениеЗадачи(ЭтаЗадача) + "Нет прав
редактировать чужой документ");

Возврат;

КонецЕсли;

Если (ОбщийСрв.ПолучитьРеквизитЗадачи(ЭтаЗадача, "Проведен")) И
(ОбщийСрв.обПолучитьКонстанту("ЗапретПроведенных")) Тогда

Сообщить(ОбщийСрв.ПредставлениеЗадачи(ЭтаЗадача) + "Уже
выполнена. Редактирование невозможно");

Возврат;

КонецЕсли;

НоваяПодЗадача = СоздатьПодзадачуНаСервере(Шаблон, ЭтаЗадача); //
создать новый документ и записать его

ДобавитьСуществующуюПодзадачуНаСервере(ЭтаЗадача, НоваяПодЗадача);
// добавить новую задачу в объект (в таб. часть)

ПутьВДереве =
ПолучитьПутьВДереве(Элементы.ДеревоПодзадач.ТекущиеДанные,
НоваяПодЗадача); //сформировать массив подзадач от текущего узла до корня

ЗаполнитьПоШаблонуРекурсивно(Шаблон, НоваяПодЗадача);

ЭтаФорма.Прочитать(); // перечитать данные формы

ЗаполнитьДеревоПодзадач(); // перестроить дерево подзадач

Элементы.ДеревоПодзадач.ТекущаяСтрока =

СпозиционироватьсяВДеревеПодзадач(ДеревоПодзадач, ПутьВДереве,
ПутьВДереве.Количество()-1); // спозиционироваться на добавленной задаче

КонецПроцедуры

&НаСервере

Процедура ЗаполнитьПоШаблонуРекурсивно(Шаблон, Основание)

ВыборкаСправочника = Справочники.Этапы.Выбрать(Шаблон);

Пока ВыборкаСправочника.Следующий() Цикл

НоваяПодЗадача =

СоздатьПодзадачуНаСервере(ВыборкаСправочника, Основание); // создать новый
документ и записать его

ДобавитьСуществующуюПодзадачуНаСервере(Основание,
НоваяПодЗадача); // добавить новую задачу в объект (в таб. часть)

ЗаполнитьПоШаблонуРекурсивно(ВыборкаСправочника.Ссылка,
НоваяПодЗадача);

КонецЦикла;

КонецПроцедуры

// открыть текущую подзадачу в дереве в отдельной форме. Вызывается из Выбор и
по команде Открыть

&НаКлиенте

Процедура ОткрытьПодзадачу()

ПодЗадача =

ЭтотОбъект.Элементы.ДеревоПодзадач.ТекущиеДанные.Подзадача;

Если ПодЗадача = Неопределено Тогда

Возврат;

```

КонецЕсли;
Если ПодЗадача.Пустая() = Истина Тогда
    Возврат;
КонецЕсли;

```

```
//ОткрытьЗначение(ПодЗадача);
```

```

П = Новый Структура;
П.Вставить("Ключ", ПодЗадача);

```

```
ОткрытьФорму("Документ.Задача.ФормаОбъекта", П); // открыть отдельную форму редактирования
```

```
КонецПроцедуры
```

```

// открыть форму выбора списка задач
&НаКлиенте
Процедура ДобавитьСуществующуюПодзадачу()

```

```

    ПодЗадача =
    ЭтотОбъект.Элементы.ДеревоПодзадач.ТекущиеДанные.Подзадача;

```

```

    П = Новый Структура;
    П.Вставить("ПараметрТекущаяСтрока", ПодЗадача);
    ОткрытьФорму("Документ.Задача.ФормаВыбора", П,
    ЭтотОбъект.Элементы.ДеревоПодзадач);

```

```
// результат выбора см. в ОбработкаВыбора элемента формы ДеревоПодзадач
```

```
КонецПроцедуры
```

```

// во внешней форме выбора завершили выбор существующей задачи - добавить ее в
дерево подзадач

```

```

&НаКлиенте
Процедура ДеревоПодзадачОбработкаВыбора(Элемент, ВыбранноеЗначение,
СтандартнаяОбработка)

```

```

    Если Не Элемент.Имя = "ДеревоПодзадач" Тогда // если выбор был не из
дерева

```

```

        Возврат;
    КонецЕсли;

```

```

    Если ВыбранноеЗначение = Неопределено Тогда // если ничего не было
выбрано

```

```

        Возврат;
    КонецЕсли;

```



```

Если Элементы.ДеревоПодзадач.ТекущиеДанные.Подзадача =
ВыбранноеЗначение Тогда // заикливание!
    Сообщить("Нельзя вставлять задачу саму в себя!");
    Возврат;
КонецЕсли;

```

```

Если ЯвляетсяПредком(ВыбранноеЗначение,
Элементы.ДеревоПодзадач.ТекущиеДанные.Подзадача) Тогда // провка на предка
    Сообщить("Добавление приведет к заикливанию!");
    Возврат;
КонецЕсли;

```

```

Если УжеЕстьВСпискеПодзадач(Элементы.ДеревоПодзадач.ТекущиеДанные,
ВыбранноеЗначение) Тогда // провка на то, что уже есть
    Сообщить("Выбранная подзадача уже является подзадачей данной
задачи!");
    Возврат;
КонецЕсли;

```

```

//ЭтаЗадача = Объект.Ссылка; // если в эту задачу
//Если Элементы.ДеревоПодзадач.ТекущиеДанные.Подзадача <> ЭтаЗадача
Тогда // если в какую-то подзадачу в дереве
// ЭтаЗадача = Элементы.ДеревоПодзадач.ТекущиеДанные.ПодЗадача;
//КонецЕсли;

```

```

ЭтаЗадача = Элементы.ДеревоПодзадач.ТекущиеДанные.ПодЗадача;

```

```

ПутьВДереве =
ПолучитьПутьВДереве(Элементы.ДеревоПодзадач.ТекущиеДанные,
ВыбранноеЗначение); //запомнить путь в дереве до нового выбранного значения
ДобавитьСуществующуюПодзадачуНаСервере(ЭтаЗадача,
ВыбранноеЗначение); // добавить выбранную задачу в объект
ЗаполнитьДеревоПодзадач(); // перстроить дерево подзадач
Элементы.ДеревоПодзадач.ТекущаяСтрока =
СпозиционироватьсяВДеревеПодзадач(ДеревоПодзадач, ПутьВДереве,
ПутьВДереве.Количество()-1); // спозиционироваться на добавленной задаче

```

```

Оповестить("Дерево", Новый Структура, ЭтотОбъект); // оповестить все
формы об изменении - пусть они переставляют свое дерево

```

```

КонецПроцедуры

```

```

//Проверка - является ли ДобавляемаяЗадача предком текущего узла (в который
добавляют) на каком-либо уровне
&НаКлиенте
Функция ЯвляетсяПредком(ДобавляемаяЗадача, ВЗадачу)

```

```

СЗ = Новый СписокЗначений; // СЗ всех предков
ОбщийСрв.СформироватьСписокПредков(СЗ, ВЗадачу); // сформировать
список всех предков

```

```

Если СЗ.НайтиПоЗначению(ДобавляемаяЗадача) = Неопределено Тогда //
если входит в СЗ - значит на каком-то уровне является предком

```

```

    Возврат Ложь;

```

```

КонецЕсли;

```

```

    Возврат Истина;

```

```

КонецФункции

```

```

// проверка - задача ВыбранноеЗначение уже есть в списке подзадач ВеткаДерева на
одном уровне

```

```

&НаКлиенте

```

```

Функция УжеЕстьВСпискеПодзадач(ВеткаДерева, ВыбранноеЗначение)

```

```

    Для Каждого Строка Из ВеткаДерева.ПолучитьЭлементы() Цикл

```

```

        Если Строка.Подзадача = ВыбранноеЗначение Тогда

```

```

            Возврат Истина;

```

```

        КонецЕсли;

```

```

    КонецЦикла;

```

```

    Возврат Ложь;

```

```

КонецФункции

```

```

//сформировать массив подзадач от текущего узла до корня

```

```

&НаКлиенте

```

```

Функция ПолучитьПутьВДереве(УзелНачало, ВыбранноеЗначение = Неопределено
)

```

```

    ПутьВДереве = Новый Массив;

```

```

    Если Не ВыбранноеЗначение = Неопределено Тогда // включить
ВыбранноеЗначение в массив или нет

```

```

        ПутьВДереве.Добавить(ВыбранноеЗначение);

```

```

    КонецЕсли;

```

```

    Если УзелНачало = Неопределено Тогда

```

```

        Возврат ПутьВДереве;

```

```

    КонецЕсли;

```

```

    // рекурсивно для каждого уровня

```

```

    ПолучитьПутьВДеревеРекурсивно(УзелНачало.ПолучитьИдентификатор(),
ПутьВДереве);

```

```

    Возврат ПутьВДереве;

```

КонецФункции

//сформировать массив подзадач от текущего узла до корня
&НаСервере

Процедура ПолучитьПутьВДеревеРекурсивно(А, ПутьВДереве)

СтрокаДереваФормы = ДеревоПодзадач.НайтиПоИдентификатору(А);
ПутьВДереве.Добавить(СтрокаДереваФормы.ПодЗадача);

Если СтрокаДереваФормы.Подзадача = Объект.Ссылка Тогда // если это
корень - то закончить рекурсию

Возврат;
КонецЕсли;

Родитель = СтрокаДереваФормы.ПолучитьРодителя();

Если Родитель = Неопределено Тогда
Возврат;

КонецЕсли;

ПолучитьПутьВДеревеРекурсивно(СтрокаДереваФормы.ПолучитьРодителя()).Полу
читьИдентификатор(), ПутьВДереве)

КонецПроцедуры

// в объект ВКакуюЗадачу добавить ЧтоДобавлять

&НаСервере

Процедура ДобавитьСуществующуюПодзадачуНаСервере(ВКакуюЗадачу,
ЧтоДобавлять)

ВКакуюЗадачуОбъект = ВКакуюЗадачу.ПолучитьОбъект();
ЧтоДобавлятьОбъект = ЧтоДобавлять.ПолучитьОбъект();

НоваяСтрокаПодзадача = ВКакуюЗадачуОбъект.Подзадачи.Добавить();
НоваяСтрокаПодзадача.ПодЗадача = ЧтоДобавлять;
ВКакуюЗадачуОбъект.Записать(); // просто записать

// если добавляемая была головная - то больше она не головная

Если ЧтоДобавлять.Головная Тогда

ЧтоДобавлятьОбъект.Головная = Ложь;

ОбщийСрв.ЗаписьВедетИсполнитель(Истина);

ЧтоДобавлятьОбъект.Записать();

ОбщийСрв.ЗаписьВедетИсполнитель(Ложь);

КонецЕсли;

Если (ВКакуюЗадачу.Проведен) И (ЧтоДобавлять.Проведен = Ложь) Тогда //
если эта задача была проведена, а новая нет - то распроводим эту

```
Сообщить("Выполнение задачи-родителя " +
ОбщийСрв.ПредставлениеЗадачи(ВКакуюЗадачу) + " будет отменено");
```

```
ВКакуюЗадачуОбъект.Записать(РежимЗаписиДокумента.ОтменаПроведения,
РежимПроведенияДокумента.Неоперативный);
КонецЕсли;
```

КонецПроцедуры

```
// удалить Подзадача из объекта Задача (в БД)
&НаСервере
Процедура УдалитьПодзадачуНаСервере(Задача, Подзадача)
```

```
ЗадачаОбъект = Задача.ПолучитьОбъект();
```

```
УдаляемаяСтрока = ЗадачаОбъект.Подзадачи.Найти(Подзадача);
Если Не УдаляемаяСтрока = Неопределено Тогда
    ЗадачаОбъект.Подзадачи.Удалить(УдаляемаяСтрока);
    ЗадачаОбъект.Записать();
КонецЕсли;
```

КонецПроцедуры

```
&НаСервере
Процедура СохранитьДеревоНаСервере()
```

```
Для Каждого Строка Из ДеревоПодзадач.ПолучитьЭлементы() Цикл
    СохранитьСтрокуДерева(Строка); // для каждой строки
КонецЦикла;
```

КонецПроцедуры

```
// сохранить строку дерева (со всеми ее подстроками)
&НаСервере
Процедура СохранитьСтрокуДерева(СтрокаДерева)
```

```
Задача = СтрокаДерева.ПодЗадача;
ЗадачаОбъект = Задача.ПолучитьОбъект();
ЗадачаОбъект.Подзадачи.Очистить(); // очистить таб. часть подзадачи
Для Каждого Строка Из СтрокаДерева.ПолучитьЭлементы() Цикл
    Новая = ЗадачаОбъект.Подзадачи.Добавить(); // добавить из дерева
каждую строку в таб часть
    Новая.Подзадача = Строка.Подзадача;
    Новая.Комментарий = Строка.Комментарий;
    СохранитьСтрокуДерева(Строка); // рекурсивно для каждой строки
КонецЦикла;
ЗадачаОбъект.Записать();
КонецПроцедуры
```

// создать новую задачу (для варианта создания новой подзадачи программно, а не в пустой форме)
 &НаСервере
 Функция СоздатьПодзадачуНаСервере(Шаблон = Неопределено, Основание = Неопределено)

// создать новый документ
 НоваяПодЗадача = Документы.Задача.СоздатьДокумент();
 НоваяПодЗадача.Дата = ТекущаяДата();
 НоваяПодЗадача.УстановитьНовыйНомер();

// заполнить реквизиты по-умолчанию и из основания
 ОбщийСрв.ПриВводеНовойЗадачи(НоваяПодЗадача, Основание, Ложь);

Если Не Шаблон = Неопределено Тогда // если шаблон задан, то часть реквизитов - из шаблона

НоваяПодЗадача.Наименование = Шаблон.Наименование;
 Ответственный = Справочники.Сотрудники.ПустаяСсылка();
 Если Не Шаблон.ОтветственныйСотрудник.Пустая() Тогда
 Ответственный = Шаблон.ОтветственныйСотрудник;

Иначе

Ответственный =

ОбщийСрв.ОтветственныйПоПодразделению(Шаблон.ОтветственноеПодразделение);

КонецЕсли;

Если Не Ответственный.Пустая() Тогда

НоваяПодЗадача.Ответственный = Ответственный; //

Ответственный

НоваяСтрИсполнители =

НоваяПодЗадача.Исполнители.Добавить();// он же - первый исполнитель

НоваяСтрИсполнители.Исполнитель = Ответственный;

НоваяСтрИсполнители.Процент = 100;

НоваяСтрИсполнители.Задание = Шаблон.Наименование;

КонецЕсли;

Если Не Шаблон.Статус.Пустая() Тогда

НоваяПодЗадача.Статус = Шаблон.Статус;

КонецЕсли;

Если Не Шаблон.Срок = 0 Тогда

НоваяПодЗадача.ОжидаемаяДатаВыполнения = ТекущаяДата() +

24*60*60*Шаблон.Срок;

КонецЕсли;

Если Не Шаблон.Объем = 0 Тогда

НоваяПодЗадача.Объем = Шаблон.Объем;

КонецЕсли;

Если Не Шаблон.ЕдИзм.Пустая() Тогда

НоваяПодЗадача.ЕдИзм = Шаблон.ЕдИзм;

КонецЕсли;
 Если Не ПустаяСтрока(Шаблон.Комментарий) Тогда
 НоваяПодЗадача.Комментарий = Шаблон.Комментарий;
 КонецЕсли;
 Если Не Шаблон.ВидЗадачи.Пустая() Тогда
 НоваяПодЗадача.ВидЗадачи = Шаблон.ВидЗадачи;

КонецЕсли;
 КонецЕсли;

НоваяПодЗадача.Записать();

Возврат НоваяПодЗадача.Ссылка;
 КонецФункции

// при любом изменении (в т.ч. проведении) оповестить другие открытые формы
 &НаКлиенте
 Процедура ПослеЗаписи(ПараметрыЗаписи)

 П = Новый Структура;
 Если Параметры.Свойство("ЭтоНовый") Тогда // если из вызывающей формы
 был передан признак что это новый документ (кнопка Новая в Подзадачи)
 Если Параметры.ЭтоНовый Тогда // то в родителя добавить эту задачу

 ДобавитьСуществующуюПодзадачуНаСервере(Параметры.Родитель,
 Объект.Ссылка); // добавить новую задачу в объект (в таб. часть)

 П.Вставить("НоваяПодЗадача", Объект.Ссылка); // и передаем
 новую задачу в ОбработкаОповещения
 П.Вставить("ЭтоНовый", Параметры.ЭтоНовый);
 КонецЕсли;
 КонецЕсли;

 Оповестить("Дерево", П, ЭтотОбъект); // оповестить об изменении, сделанном
 в этом форме, все остальные формы - пусть они перстаривают свое дерево
 // далее см. в ОбработкаОповещения
 КонецПроцедуры

// обработка оповещения о любых изменениях, проведенных в других формах
 &НаКлиенте
 Процедура ОбработкаОповещения(ИмяСобытия, Параметр, Источник)

 Если ИмяСобытия <> "Дерево" Тогда // не наш случай
 Возврат;
 КонецЕсли;

//Если Источник = ЭтотОбъект Тогда // если источник изменения - сама эта форма

// Возврат;

//КонецЕсли;

ПутьВДереве =
ПолучитьПутьВДереве(Элементы.ДеревоПодзадач.ТекущиеДанные); // запомни путь до текущего элемента

Если Параметр.Свойство("ЭтоНовый") Тогда // если была передана структура
Если Параметр.ЭтоНовый Тогда // если оповещение передала форма, в которой был создан новый документ

Если Источник.ВладелецФормы = ЭтотОбъект Тогда // если та форма вызывалась из этой

НоваяПодЗадача = Параметр.НоваяПодзадача; // ссылка на измененную подзадачу

ПутьВДереве =
ПолучитьПутьВДереве(Элементы.ДеревоПодзадач.ТекущиеДанные, НоваяПодЗадача); //сформировать массив подзадач от добавленной подзадачи до корня

КонецЕсли;

КонецЕсли;

КонецЕсли;

// в любом случае - перестроить дерево

ЗаполнитьДеревоПодзадач(); // перестроить дерево подзадач

Элементы.ДеревоПодзадач.ТекущаяСтрока =

СпозиционироватьсяВДеревеПодзадач(ДеревоПодзадач, ПутьВДереве, ПутьВДереве.Количество()-1); // спозиционироваться

КонецПроцедуры

// по команде ДеревоПодзадачаПровести и ДеревоПодзадачаРасПровести &НаКлиенте

Процедура ДеревоПодзадачаПровести(Команда)

Если Элементы.ДеревоПодзадач.ТекущиеДанные = Неопределено Тогда

Возврат;

КонецЕсли;

ЭтаЗадача = Элементы.ДеревоПодзадач.ТекущиеДанные.ПодЗадача; // какую подзадачу в дереве проводим

Если ЭтаЗадача.Пустая() Тогда

Возврат;

КонецЕсли;

Если Команда.Имя = "ДеревоПодзадачаПровести" Тогда

```

Если Не ПроверкаВозможностиПроведения(ЭтаЗадача) Тогда
    Возврат;
КонецЕсли;
ПровестиПодЗадачуНаСервере(ЭтаЗадача);
КонецЕсли;

```

```

Если Команда.Имя = "ДеревоПодЗадачаРасПровести" Тогда
    Если Не ПроверкаВозможностиРасПроведения(ЭтаЗадача) Тогда
        Возврат;
    КонецЕсли;
    РасПровестиПодЗадачуНаСервере(ЭтаЗадача);
КонецЕсли;

```

```

ПутьВДереве =
ПолучитьПутьВДереве(Элементы.ДеревоПодзадач.ТекущиеДанные); // запомниь
путь до текущего элемента
    ЗаполнитьДеревоПодзадач(); // перестроить дерево подзадач
    Элементы.ДеревоПодзадач.ТекущаяСтрока =
СпозиционироватьсяВДеревеПодзадач(ДеревоПодзадач, ПутьВДереве,
ПутьВДереве.Количество()-1); // спозиционироваться

```

```

Оповестить("Дерево", Новый Структура, ЭтотОбъект); // оповестить об
изменении, сделанном в этом форме, все остальные формы - пусть они перстаривают
свое дерево

```

```

ОтобразитьИзменениеДанных(Объект.Ссылка,
ВидИзмененияДанных.Изменение);

```

```

КонецПроцедуры

```

```

// провести ПодЗадача
&НаСервере
Процедура ПровестиПодЗадачуНаСервере(ПодЗадача)

```

```

    ПодЗадачаОбъект = ПодЗадача.ПолучитьОбъект();
    ПодЗадачаОбъект.Записать(РежимЗаписиДокумента.Проведение,
РежимПроведенияДокумента.Неоперативный);
КонецПроцедуры

```

```

// распровести ПодЗадача
&НаСервере
Процедура РасПровестиПодЗадачуНаСервере(ПодЗадача)

```

```

    ПодЗадачаОбъект = ПодЗадача.ПолучитьОбъект();
    ПодЗадачаОбъект.Записать(РежимЗаписиДокумента.ОтменаПроведения,
РежимПроведенияДокумента.Неоперативный);
КонецПроцедуры

```

```

// по команде ДеревоСвернутьВсе - свернуть все узлы, кроме самого верхнего

```


&НаКлиенте

Процедура ДеревоСвернутьВсе(Команда)

Для Каждого СтрокаДерева Из ДеревоПодзадач.ПолучитьЭлементы() Цикл
 Для Каждого Строка1 Из СтрокаДерева.ПолучитьЭлементы() Цикл

Элементы.ДеревоПодзадач.Свернуть(Строка1.ПолучитьИдентификатор());
 КонецЦикла;

КонецЦикла;

КонецПроцедуры

// по команде ДеревоСвернутьДоТекущей - свернуть все узлы, но развернуть путь до
 текущего узла

&НаКлиенте

Процедура ДеревоСвернутьДоТекущей(Команда)

ПутьВДереве

ПолучитьПутьВДереве(Элементы.ДеревоПодзадач.ТекущиеДанные); // запомни
 путь до текущего элемента

Для Каждого СтрокаДерева Из ДеревоПодзадач.ПолучитьЭлементы() Цикл
 Для Каждого Строка1 Из СтрокаДерева.ПолучитьЭлементы() Цикл

Элементы.ДеревоПодзадач.Свернуть(Строка1.ПолучитьИдентификатор());
 КонецЦикла;

КонецЦикла;

ЗаполнитьДеревоПодзадач(); // перестроить дерево подзадач

Элементы.ДеревоПодзадач.ТекущаяСтрока

СпозиционироватьсяВДеревеПодзадач(ДеревоПодзадач, ПутьВДереве,
 ПутьВДереве.Количество()-1); // спозиционироваться

КонецПроцедуры

// по команде ДеревоФильтрПроведенные - отобразить в дереве только
 Проведенные/Непроведенные/Все

&НаКлиенте

Процедура ДеревоФильтрПроведенные(Команда)

ФильтрПроведенные = ФильтрПроведенные - 1; // при каждом входе в
 команду - уменьшить значеней фильтра на 1

Если ФильтрПроведенные = -1 Тогда

ФильтрПроведенные = 2; // по кругу

КонецЕсли;

// заголовок кнопки

Если ФильтрПроведенные = 2 Тогда

Элементы.ДеревоПодзадачПроведенные.Заголовок = "Все";

ИначеЕсли ФильтрПроведенные = 1 Тогда
 Элементы.ДеревоПодзадачПроведенные.Заголовок = "Выполненные";

ИначеЕсли ФильтрПроведенные = 0 Тогда
 Элементы.ДеревоПодзадачПроведенные.Заголовок = "Не
 выполненные";
 КонецЕсли;

ЗаполнитьДеревоПодзадач(); // перестроить дерево с отбором,
 удовлетворяющем фильтру

КонецПроцедуры

&НаКлиенте

Процедура ДеревоПоШаблону(Команда)

ОткрытьФорму("Справочник.Этапы.ФормаВыбора", ,ЭтаФорма);
 // Продолжение см. в ОбработкаВыбора формы

КонецПроцедуры

//-----

// ЗАКЛАДКА РЕСУРСЫ

// при выборе/изменении номенклатуры

&НаКлиенте

Процедура НоменклатураНоменклатураПриИзменении(Элемент)

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Количество = 1;

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф = 1;
 ОсновнаяЕдиницаОтгрузки =

ОбщийСрв.ПолучитьОсновнуюЕдиницуОтгрузкиНоменклатуры(ЭтотОбъект.Элем
 енты.Номенклатура.ТекущиеДанные.Номенклатура);

Если Не ОсновнаяЕдиницаОтгрузки.Пустая() Тогда

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.ЕдИзмерения =
 ОсновнаяЕдиницаОтгрузки;

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф =
 ОбщийСрв.ПолучитьКоэффициент(ЭтотОбъект.Элементы.Номенклатура.Текущие
 Данные.ЕдИзмерения);

КонецЕсли;

ОбщийСрв.РассчитатьСуммуСтроки(Объект.Дата,
 ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Номенклатура,

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Количество,

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф,

Объект.ТипЦен,

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Цена,

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Сумма);

КонецПроцедуры

&НаКлиенте

Процедура НоменклатураКоличествоПриИзменении(Элемент)

ОбщийСрв.РассчитатьСуммуСтроки(Объект.Дата,
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Номенклатура,

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Количество,

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф,

Объект.ТипЦен,

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Цена,

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Сумма);

КонецПроцедуры

&НаКлиенте

Процедура НоменклатураЕдИзмеренияПриИзменении(Элемент)

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф =
ОбщийСрв.ПолучитьКоэффициент(ЭтотОбъект.Элементы.Номенклатура.Текущие
Данные.ЕдИзмерения);

ОбщийСрв.РассчитатьСуммуСтроки(Объект.Дата,
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Номенклатура,

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Количество,

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф,

Объект.ТипЦен,

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Цена,

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Сумма);

КонецПроцедуры

// при выборе номенклатуры - проверка резерва

&НаКлиенте

Процедура НоменклатураНоменклатураОбработкаВыбора(Элемент,
ВыбранноеЗначение, СтандартнаяОбработка)

Если Зарезервировано(ВыбранноеЗначение) Тогда
СтандартнаяОбработка = Ложь;

КонецЕсли;

КонецПроцедуры

// проверка не зарезервирована ли ВыбНоменклатура в других задачах

&НаСервере

Функция Зарезервировано(ВыбНоменклатура)

Если Не ВыбНоменклатура.ВидНоменклатуры.Резервируется Тогда
Возврат Ложь;

КонецЕсли;

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ЗадачаНоменклатура.Номенклатура,

| Задача.Ссылка КАК СамаЗадача

|ИЗ

| Документ.Задача.Номенклатура КАК ЗадачаНоменклатура

| ЛЕВОЕ СОЕДИНЕНИЕ Документ.Задача КАК Задача

| ПО ЗадачаНоменклатура.Ссылка = Задача.Ссылка

|ГДЕ

| ЗадачаНоменклатура.Номенклатура = &ВыбНоменклатура

| И Задача.Проведен = Ложь

| И ((&ДатаНачалаРезерва МЕЖДУ Задача.Дата И

Задача.ОжидаемаяДатаВыполнения)

| ИЛИ (&ДатаОкончанияРезерва МЕЖДУ Задача.Дата И

Задача.ОжидаемаяДатаВыполнения)

| ИЛИ (Задача.Дата МЕЖДУ &ДатаНачалаРезерва И

&ДатаОкончанияРезерва)

| ИЛИ (Задача.ОжидаемаяДатаВыполнения МЕЖДУ

&ДатаНачалаРезерва И &ДатаОкончанияРезерва)

|)

!";

Запрос.УстановитьПараметр("ВыбНоменклатура",
ВыбНоменклатура);

```

Запрос.УстановитьПараметр("ДатаНачалаРезерва",
НачалоДня(Объект.Дата));
Запрос.УстановитьПараметр("ДатаОкончанияРезерва",
КонецДня(Объект.ОжидаемаяДатаВыполнения));

```

```

Результат = Запрос.Выполнить();
Выборка = Результат.Выбрать();
Пока Выборка.Следующий() Цикл
    Сообщить(ВыбНоменклатура.Наименование + " зарезервировано "+
ОбщийСрв.ПредставлениеЗадачи(Выборка.СамаЗадача));
    Возврат Истина
КонецЦикла;

Возврат Ложь;

```

КонецФункции

```

// при изменени даты документа проверить - нет ли зарезервированных ресурсов
&НаКлиенте
Процедура ДатаПриИзменении(Элемент)

```

```

    Для Каждого ТекСтрокаНоменклатура Из Объект.Номенклатура Цикл
        Зарезервировано(ТекСтрокаНоменклатура.Номенклатура);
    КонецЦикла;
КонецПроцедуры

```

```

//-----
// ЗАКЛАДКА ФАЙЛЫ

```

```

// по команде Создать новый файл
&НаКлиенте
Процедура ФайлыСоздать(Команда)

```

```

    П = Новый Структура;
    П.Вставить("Документ", Объект.Ссылка);
    ОткрытьФорму("Справочник.ХранимыеФайлы.ФормаОбъекта", П,
ЭтотОбъект.Элементы.Файлы);

```

```

    // результат выбора см. в ОбработкаВыбора элемента формы Файлы (таблица)
КонецПроцедуры

```

```

&НаКлиенте
Процедура ФайлыОбработкаВыбора(Элемент, ВыбранноеЗначение,
СтандартнаяОбработка)

```

```

    НоваяСтрока = Объект.Файлы.Добавить();

```

НоваяСтрока.Файл = ВыбранноеЗначение.Получить(0).Значение;
НоваяСтрока.Комментарий = ВыбранноеЗначение.Получить(0).Значение;
КонецПроцедуры

&НаКлиенте

Процедура ОткрытьФайл(Команда)

ХранимыйФайл = ЭтотОбъект.Элементы.Файлы.ТекущиеДанные.Файл;

ОткрытьХранимыйФайл(ХранимыйФайл);

КонецПроцедуры

&НаКлиенте

Процедура ОткрытьХранимыйФайл(ХранимыйФайл)

Если ХранимыйФайл = Неопределено Тогда

Возврат;

КонецЕсли;

Файл = Новый Файл(ОбщийСрв.ПолучитьИмяФайла(ХранимыйФайл));

ИмяФайла = ПолучитьИмяВременногоФайла(Файл.Расширение);

Адрес = ПолучитьНавигационнуюСсылку(ХранимыйФайл, "ДанныеФайла");

ПолучитьФайл(Адрес, ИмяФайла, Ложь);

ЗапуститьПриложение(ИмяФайла);

КонецПроцедуры