

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ
«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК

Базовая кафедра автоматизации бизнес-процессов
(на платформе 1С:Предприятие)

РЕКОМЕНДОВАНО К ЗАЩИТЕ
В ГЭК И ПРОВЕРЕНО НА ОБЪЕМ
ЗАИМСТВОВАНИЯ

Заведующий кафедрой

Д.т.н., профессор

_____ *А.Г. Ивашко*

_____ 2019 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(бакалаврская работа)

АВТОМАТИЗАЦИЯ БИЗНЕС-ПРОЦЕССОВ В КОМПАНИИ «РЕМТЕХ» С
ИСПОЛЬЗОВАНИЕМ ПРОГРАММНОГО ПРОДУКТА «УПРАВЛЕНИЕ
СЕРВИСНЫМ ЦЕНТРОМ»

09.03.03 Прикладная информатика

Выполнил работу
Студент 4 курса
очной формы обучения

(Подпись)

Шенгелия
Давид
Юзаевич

Руководитель работы
к.т.н., доцент

(Подпись)

Григорьева
Инна
Ивановна

Руководитель от предприятия
руководитель отдела разработки

(Подпись)

Усманов
Вадим
Фильсунович

Содержание

Введение.....	5
Глава 1. Деятельность в ООО «РемТех»	6
1.1. Описание схем метаданных объектов, используемых в доработках	6
1.2. Проблематика	23
1.3. Цель и задачи.....	24
Глава 2. Проектирование программного обеспечения.....	25
2.1 Архитектура.....	25
2.1.1. Интеграция 1С:Рарус-Софтфон с 1С: Управление сервисным центром.....	25
2.1.1.1. Сравнение 1С:Рарус-Софтфон с другими решениями	25
2.1.1.2. Общая архитектура.....	26
2.1.1.2.1. Доработки в системе.....	27
2.1.1.2.1.1. 1С: Рарус-Софтфон (клиент и сервер).....	27
2.1.1.2.1.2. Конфигурация 1С: СофтФон для конфигурации «1С: Управление сервисным центром»	28
2.1.1.2.1.3. Конфигурация «1С: Управление сервисным центром»	28
2.1.2. Публикации заказ-нарядов и назначения мастеров	31
2.1.2.1. Доработки в системе.....	34
2.1.3. Табель рабочего времени на месяц.....	37
2.2. Бизнес-процессы	38
2.2.1. Бизнес-процесс работы со звонками и телефонией	38

2.2.1.1. Бизнес-процесс работы со звонками и телефонией до интеграции с 1С:Рарус-Софтфон.....	38
2.2.1.2. Бизнес-процесс работы со звонками и телефонией после интеграции с 1С:Рарус-Софтфон.....	41
2.2.1.3. Сравнение эффективности бизнес-процессов «до» и «после»	42
2.2.2. Бизнес-процесс публикации заказ-нарядов и назначения на них мастеров	43
2.2.2.1. Бизнес-процесс публикации заказ-нарядов и назначения на них мастеров до появления обработки	43
2.2.2.2. Бизнес-процесс публикации заказ-нарядов и назначения на них мастеров после появления обработки	45
2.2.2.3. Сравнение эффективности бизнес-процессов «до» и «после»	48
2.3. Алгоритмы	49
2.3.1. Алгоритм работы интегрированной телефонии	49
2.3.2. Алгоритм публикации и назначения мастера в рамках одного заказ-наряда	52
Глава 3. Реализация.....	55
3.1. Реализация архитектуры интеграции 1С:Рарус-Софтфон с 1С: Управление сервисным центром	55
3.1.1. Реализация в части 1С:Рарус-Софтфон.....	55
3.1.2. Реализация в модуле Asterisk	55
3.1.3. Реализация в конфигурации 1С: Управление сервисным центром	61
3.2. Реализация архитектуры процесса публикации заказ-нарядов и назначения мастеров	66

3.3. Аналитика	67
3.3.1. Аналитика по публикации и назначению сотрудника на заказ-наряд	67
3.3.2. Аналитика по интеграции телефонии в конфигурацию «1С:Управление сервисным центром»	68
3.3.3. Апробация.....	69
Заключение	70
Список литературы	71
Приложение А. Интеграция 1С:Рарус-Софтфон с 1С: Управление сервисным центром.....	72
Приложение Б. Публикации заказ-нарядов и назначения мастеров.....	81
Приложение В. Табель рабочего времени на месяц	100

Введение

Предприятие «РемТех» занимается ремонтом бытовой техники, а также её подключением и настройкой, помимо этого предприятие оказывает услуги ремонта компьютеров, планшетов и смартфонов. Для автоматизации процессов формирования заказов на ремонт техники и отслеживания каждого из них диспетчером предприятие ведет свою деятельность в конфигурации «1С: Управление сервисным центром», которая включает в себя бизнес-процессы учета рабочего времени сотрудника, CRM для звонков, управления кадрами для расчета заработной платы, учета стационарных и выездных ремонтных работ и т.д.

При поступлении звонка от клиента диспетчер вручную создает заказ-наряд и описывает детали заказа. Проблема заключается в том, что диспетчер не всегда может оперативно найти и назначить мастера на поступивший заказ-наряд и/или весьма поверхностно описать его, что в свою очередь ведет к низкой эффективности бизнес-процесса ввиду неполной детализации по заказу, и как следствие, понижает производительность бизнес-процесса. Также этому способствует тот момент, что учет рабочего времени сотрудника в конфигурации «1С: Управление сервисным центром» ведется в упрощенном виде. Под этим подразумевается то, что у сотрудника всего два состояния: либо он работает, либо он заблокирован до определенного периода времени. Упрощенный вид учёта не учитывает то, что сотрудник может быть на больничном или быть в отпуске и не учитывает многие другие стандартные ситуации с сотрудником, что также влияет на эффективность бизнес-процесса формирования заказ-наряда, т.к. сотрудник может быть на больничном, но при этом в системе будет указано, что он «работает», что не является корректным.

Глава 1. Деятельность в ООО «РемТех»

1.1. Описание схем метаданных объектов, используемых в доработках

Для того, чтобы построить схемы метаданных, используемых для дальнейших доработок объектов, был использован инструмент «Схема данных» из среды разработки под названием «1С: Enterprise Development Tools» [3].

На рис. 1 показана связь между историей звонков и контрагентами, пользователями и т.д (см. табл. 1-4 для более детального описания сущностей).

Таблица 1. Реквизиты документа «ТелефонныйЗвонок»

Название реквизита	Тип	Описание
АбонентКакСвязаться	Строка	Контактная информация
АбонентКонтакт	Справочник.КонтактныеЛица, Справочник.ФизическиеЛица, Справочник.Контрагенты	Ссылка на контактное лицо
АбонентПредставление	Строка	Контакт
Автор	Справочнк.Пользователи	Пользователь, зарегистрировавший звонок
Важность	Перечисление.ВариантыВажностиВзаимодействия	Отметка о важности письма
Входящий	Булево	Тип звонка
Описание	Строка	Описание телефонного звонка
Ответственный	Справочник.Пользователи	Ответственный за обработку звонка
Рассмотрено	Булево	Признак, указывающий на то, что необходимые действия завершены.
РассмотретьПосле	Дата	Плановые дата и время работы по звонку
Предмет	Документ.ТелефонныйЗвонок	Ссылка на предмет обсуждения.
Тема	Строка	Тема письма
сфпСоMagicID	Строка	Идентификатор контакта в системе СоMagic
сфпДлительностьЗвонка	Число	Длительность звонка в секундах
сфпИдентификаторЗаписи	Строка	Идентификатор записи телефонного разговора

сфпИдентификаторЗвонка а	Число	Идентификатор звонка
сфпСостояниеЗвонка	Перечисление.сфпСостоянияЗвонков	Состояние звонка
Комментарий	Строка	Произвольный текст

Таблица 2. Реквизиты справочника «Контрагенты»

Название реквизита	Тип	Описание
БанковскийСчетПоУмолчанию	Справочник.БанковскиеСчета	Банковский счет по умолчанию, для подстановки в документы.
ВидГосударственногоОргана	Перечисление.ВидыГосударственныхОрганов	
ВидКонтрагента	Перечисление.ВидыКонтрагентов	
ДатаРождения	Дата	Дата рождения физического лица
ДокументУдостоверяющийЛичность	Строка	Документ, удостоверяющий личность
СЦентр_Перевозчик	Булево	Если установлен, то контрагент является перевозчиком
СЦентр_СервисныйЦентр	Булево	Если установлен, то контрагент является сервисным центром
АдресЭП	Строка	Адрес электронной почты
ИНН	Строка	Идентификационный номер контрагента, для отражения в печатных формах документов
КонтактноеЛицо	Справочник.КонтактныеЛица	Физическое лицо, с помощью которого

		осуществляется связь между контрагентом и предприятием
КПП	Строка	Код постановки на учет, для отражения в печатных формах документов. Указывается только для юридического лица
КодПоОКПО	Строка	Код по Общероссийскому классификатору предприятий и организаций, для отражения в печатных формах документов
НомерТелефона	Строка	Номер телефона для поиска при вводе по строке (служебный реквизит)
Покупатель	Булево	Если установлен, то контрагент является покупателем
Поставщик	Булево	Если установлен, то контрагент является поставщиком
Пол	Перечисление.ПолФизическогоЛица	Пол контрагента
РегистрационныйНомер	Строка	
СтранаРегистрации	Справочник.СтраныМира	
ФИО	Строка	
ГоловнойКонтрагент	Справочник.Контрагенты	
сфпСоMagicID	Строка	Идентификатор контакта в системе СоMagic

Таблица 3. Реквизиты справочника «Структурные единицы»

Название реквизита	Тип	Описание
ТипСтруктурнойЕдиниц ы	Перечисление.ТипыСтру ктурныхЕдиниц	Тип структурной единицы: склад, производство, управленческое подразделение
ОрдерныйСклад	Булево	Признак ордерного склада. Флаг следует установить, если на складе будет использоваться ордерная схема
Организация	Справочник.Организации	Организация, к которой относится структурная единица. Реквизит используется только при обмене данными с конфигурацией бухгалтерия предприятия
МОЛ	Справочник.Физические Лица	Материально-ответственное лицо
РозничныйВидЦен	Справочник.ВидыЦен	Вид цен при продаже в розницу
СчетУчетаВРознице	ПланСчетов.Управленчес кий	Счет учета в рознице
ГоловнаяОрганизация	Справочник.Организации	
КПП	Строка	
НаименованиеПолное	Строка	Полное официальное наименование организации для вывода в печатные формы документов

Таблица 4. Реквизиты справочника «Пользователи»

Название реквизита	Тип	Описание
Недействителен	Булево	Пользователь больше не работает в программе, но сведения о нем сохранены
Подразделение	ОпределяемыйТип.Подразделение	Подразделение, в котором работает пользователь
ФизическоеЛицо	ОпределяемыйТип.ФизическоеЛицо	Физическое лицо, с которым связан пользователь
Комментарий	Строка	Произвольная строка
Служебный	Булево	Неразделенный или разделенный служебный пользователь, права к которому устанавливаются непосредственно и программно
Подготовлен	Булево	Пользователь подготовлен ответственным за список пользователей
ИдентификаторПользователяИБ	УникальныйИдентификатор	Уникальный идентификатор пользователя информационной базы
д_Организация	Справочник.Организации	Организация, которой принадлежит пользователь

На рис. 2 показана связь между историей звонков и контактными лицами, физическими лицами и организациями (см. табл. 5-10 для более детального описания сущностей).

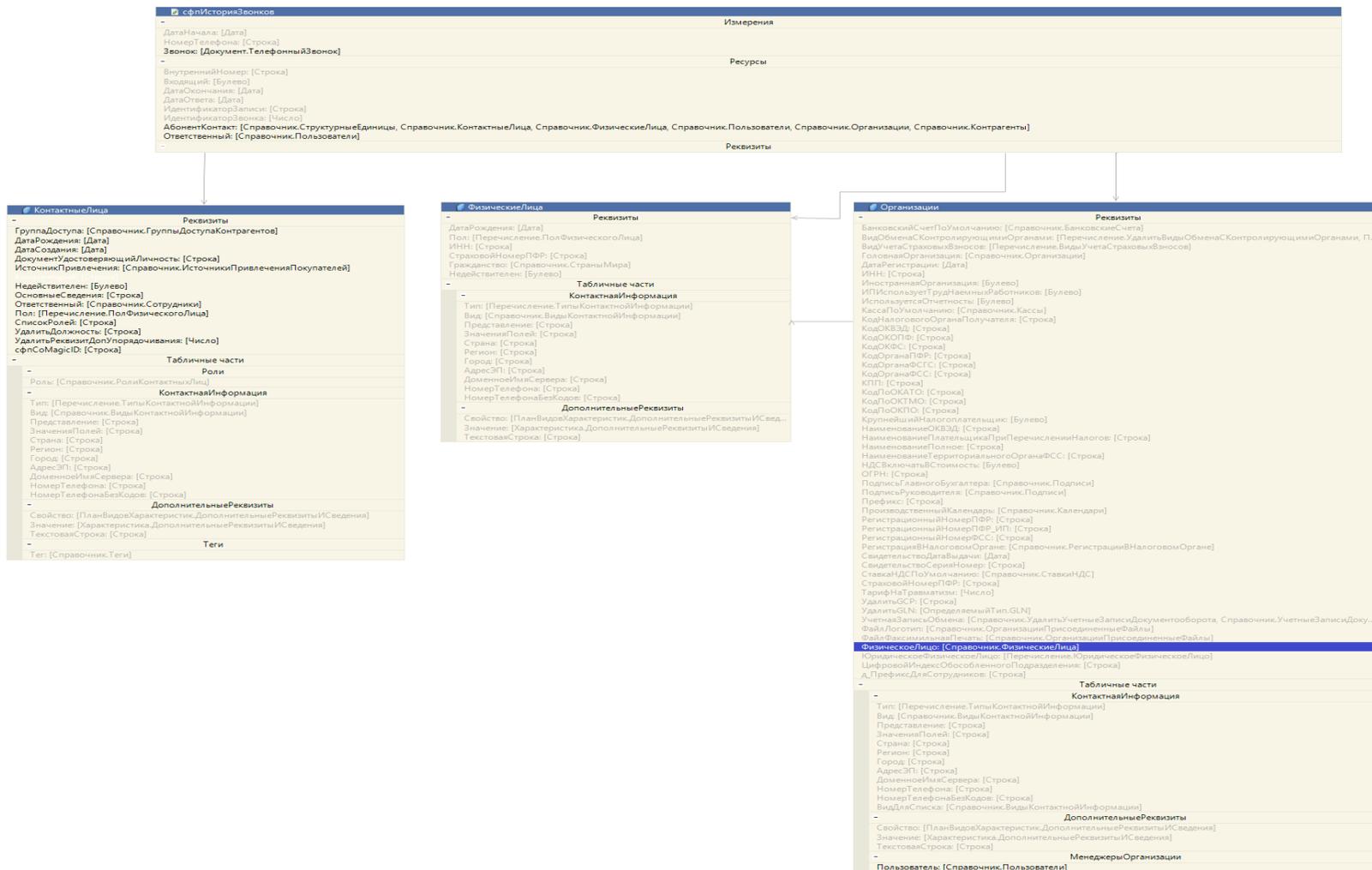


Рис. 2. ER-диаграмма связей истории звонков с контактными лицами, физическими лицами и организациями

Таблица 5. Реквизиты справочника «Контактные лица»

Название реквизита	Тип	Описание
ГруппаДоступа	Справочник.ГруппыДоступа	Группа доступа, к которой отнесен контакт
ДатаРождения	Дата	
ДатаСоздания	Дата	Дата создания контактного лица
ДокументУдостоверяющийЛичность	Строка	Документ, удостоверяющий личность
ИсточникПривлечения	Справочник.ИсточникиПривлеченияПокупателей	
Комментарий	Строка	Дополнительная информация о взаимодействии с контактным лицом
ОсновныеСведения	Строка	
Ответственный	Справочник.Сотрудники	Сотрудник, поддерживающий связь с контактным лицом
Пол	Перечисление.ПолФизическогоЛица	
УдалитьДолжность	Строка	Должность с визитки, по классификации контрагента
сфпCoMagicID	Строка	Идентификатор контакта в системе CoMagic

Таблица 6. Реквизиты справочника «Организации»

Название реквизита	Тип	Описание
БанковскийСчетПоУмолчанию	Справочник.БанковскиеСчета	Банковский счет организации по умолчанию, для подстановки в документы
ВидУчетаСтраховыхВзносов	Перечисление.ВидыУчетаСтраховыхВзносов	В случае, если одновременно используются УСН и ЕНВД необходимо выбрать, при каком налогообложении при расчете налога будут учитываться уплаченные страховые взносы
ГоловнаяОрганизация	Справочник.Организации	
ДатаРегистрации	Дата	Дата государственной регистрации
ИНН	Строка	Идентификационный номер налогоплательщика
ИностраннаяОрганизация	Булево	
ИПИспользуетТрудНаемныхРаботников	Булево	
КодНалоговогоОрганаПолучателя	Строка	Заполняется только для крупнейших налогоплательщиков
КПП	Строка	Код причины постановки на учет
КрупнейшийНалогоплательщик	Булево	
ПодписьГлавногоБухгалтера	Справочник.Подписи	
ПодписьРуководителя	Справочник.Подписи	
ПроизводственныйКалендарь	Справочник.Календари	Основной производственный календарь

Регистрация В Налоговом Органе	Справочник. Регистрации В Налоговом Органе	
Ставка НДС По Умолчанию	Справочник. Ставки НДС	Ставка НДС организации по умолчанию, для подстановки в документы
Физическое Лицо	Справочник. Физическое Лицо	Физическое лицо, если организация является физическим лицом
Юридическое Физическое Лицо	Перечисление. Юридическое Физическое Лицо	Признак, является ли организация юридическим или физическим лицом

Таблица 7. Реквизиты справочника «Физические лица»

Название реквизита	Тип	Описание
Дата Рождения	Дата	Дата рождения физического лица
Пол	Перечисление. Пол Физического Лица	Пол физического лица
ИНН	Строка	Индивидуальный номер налогоплательщика сотрудника
Страховой Номер ПФР	Строка	Страховой номер сотрудника в пенсионном фонде
Гражданство	Справочник. Страны Мира	Гражданином какой страны является сотрудник
Недействителен	Булево	Физическое лицо не используется больше в программе, но сведения о нем сохранены

Таблица 8. Измерения регистра сведений «сфпИсторияЗвонков»

Название реквизита	Тип	Описание
ДатаНачала	Дата	Дата начала звонка
НомерТелефона	Строка	Номер телефона
Звонок	Документ.ТелефонныйЗв онок	Документ «Телефонный звонок»

Таблица 9. Ресурсы регистра сведений «сфпИсторияЗвонков»

Название реквизита	Тип	Описание
ВнутреннийНомер	Строка	Внутренний номер пользователя
Входящий	Булево	Тип звонка
ДатаОкончания	Дата	Дата окончания звонка
ДатаОтвета	Дата	Дата ответа на звонок, если дата пустая, то звонок считается пропущенным
ИдентификаторЗаписи	Строка	Идентификатор записи телефонного разговора
ИдентификаторЗвонка	Число	Идентификатор звонка
АбонентКонтакт	Справочник.КонтактныеЛица, Справочник.ФизическиеЛица, Справочник.Пользователи, Справочник.Контрагенты	Ссылка на контактное лицо

Таблица 10. Реквизиты табличной части «Контактная информация»

Название реквизита	Тип	Описание
Тип	Перечисление.ТипыКонтактнойИнформации	Тип контактной информации (телефон, адрес и т.п.)
Вид	Справочник.ВидыКонтактнойИнформации	Вид контактной информации
Представление	Строка	Представление контактной информации для отображения в формах
ЗначенияПолей	Строка	Служебное поле, для хранения контактной информации
Страна	Строка	Страна (заполняется для адреса)
Регион	Строка	Регион (заполняется для адреса)
Город	Строка	Город (заполняется для адреса)
ДоменноеИмяСервера	Строка	Доменное имя сервера электронной почты или веб-страницы
НомерТелефона	Строка	Полный номер телефона
НомерТелефонаБезКодов	Строка	Номер телефона без кодов и добавочного номера

На рис. 3. показана структура документа «Заказ-наряд», который является ключевым элементом, используемым в результате доработок (см. табл. 11 для описания структуры).

_ЗаказНарядСервисн...	
+	Реквизиты
-	Табличные части
-	Запчасти
	Номенклатура: [Справочник...]
	ТипНоменклатурыЗапас: [Б...]
	Характеристика: [Справочни...]
	Партия: [Справочник.Парти...]
	Количество: [Число]
	Резерв: [Число]
	ЕдиницаИзмерения: [Справо...]
	Цена: [Число]
	Сумма: [Число]
	ПроцентСкидкиНаценки: [Ч...]
	СтавкаНДС: [Справочник.Ст...]
	СуммаНДС: [Число]
	Всего: [Число]
	Содержание: [Строка]
	КлючСвязи: [Число]
-	КонтактнаяИнформация
	Тип: [Перечисление.ТипыКо...]
	Вид: [Справочник.ВидыКонт...]
	Представление: [Строка]
	ЗначенияПолей: [Строка]
	Страна: [Строка]
	Регион: [Строка]
	Город: [Строка]
	АдресЭП: [Строка]
	ДоменноеИмяСервера: [Стр...]
	НомерТелефона: [Строка]
	НомерТелефонаБезКодов: [...]
-	ДополнительныеРеквизиты
	Свойство: [ПланВидовХарак...]
	Значение: [Характеристика...]
	ТекстоваяСтрока: [Строка]
-	ВидыРабот
	ВидРабот: [Справочник._Вид...]
-	РемонтируемыеИзделия
	ВидИзделия: [Справочник._В...]
	Модель: [Справочник._Моде...]
	Бренд: [Справочник._Бренды]
	СерийныйНомер: [Строка]
	ОписаниеНеисправности: [С...]
	МеханическиеПовреждения:...
	Комплектация: [Строка]
	КлючСвязи: [Число]
	ПечатьНомерГарантии: [Стр...]
	ПечатьМестоПриобретения:...
	ПечатьДатаВыпускаИзделия...
	ПечатьДатаПриобретения: [...]

Рис. 3. Структура документа «Заказ-наряд»

Таблица 11. Реквизиты документа «Заказ-наряд»

Название реквизита	Тип	Описание
Организация	СправочникСсылка.Организации	Организация, к которой относится сотрудник
Контрагент	СправочникСсылка.Контрагенты	Клиент
Договор	СправочникСсылка.ДоговорыКонтрагентов	Договор контрагента, в рамках которого оформляется документ
Подразделение	СправочникСсылка.СтруктурныеЕдиницы	Ответственное подразделение
СостояниеЗаказа	СправочникСсылка._СостояниеЗаказовНарядов	Состояние заказа
Сотрудник	СправочникСсылка.Сотрудники	Сотрудник, работающий с заказом
СтоимостьРемонта	Число	Стоимость ремонта
СтоимостьВызова	Число	Стоимость вызова
ТипРемонта	ПеречислениеСсылка._ТипыРемонта	Описание одного из типов ремонта
МеханическиеПовреждения	Строка	Описание механических повреждений
д_ДоляСотрудника	Число	Доля мастера (%)
д_ДоляОрганизации	Число	Процент по городу
д_ДоляПоВидуТехники	Число	Процент по типу техники
д_КорректировкаДолиСотрудника	Число	Корректировка доли сотрудника
ДатаВизитаМастера	Дата	Дата визита мастера
ВремяНачалаВизитаМастера	Дата (время)	Время начала визита мастера
ВремяОкончанияВизитаМастера	Дата (время)	Время окончания визита мастера
ОписаниеНеисправности	СправочникСсылка._ШаблоныЗаполнения	Описание одного из видов неисправности

На рис. 4 показана структура документа «Табель на месяц», который основывается на регистре сведений «Отклонение от графиков работы ресурсов» (см. табл. 12-15 для более детального описания сущностей).



Рис. 4. Структура документа «Табель на месяц»

Таблица 12. Реквизиты документа «Табель на месяц»

Название реквизита	Тип	Описание
Год	Число	Год проведения табеля
ПериодРегистрации	Дата	Дата проведения табеля
Организация	СправочникСсылка.Орган изации	Организация, в которой находятся сотрудники

Таблица 13. Реквизиты табличной части документа «Табель на месяц»

Название реквизита	Тип	Описание
Сотрудники	СправочникСсылка.Сотру дники	Сотрудники, который находится в вышестоящей организации
День (1...31)	Число	Значение в табеле на (1...31) день месяца

Таблица 14. Измерения регистра сведений «Отклонения от графиков работы
ресурсов»

Название реквизита	Тип	Описание
Сотрудники	СправочникСсылка.Ключе выеРесурсы, СправочникСсылка.Сотру дники	Сотрудники, который находится в вышестоящей организации и другие ключевые ресурсы
Год	Число	Год, за который заполняется табель
День	Дата	Дата проведения табеля
ВремяНачала	Дата и время	Время начала работы сотрудника
ВремяОкончания	Дата и время	Время окончания работы сотрудника

Таблица 15. Ресурсы и реквизиты регистра сведений «Отклонения от графиков работы ресурсов»

Название реквизита	Тип	Описание
ЧасыРаботы	Число	Количество часов работы на день работы сотрудника
ЧасовПерерывов	Число	Количество часов, когда сотрудник отдыхал в течение дня
НеРабочийДень	Булево	Работает ли сотрудник в этот день или нет

1.2. Проблематика

Проблематика имеющейся в конфигурации «1С:Управление сервисным центром» телефонии заключается в том, что она имеет весьма ограниченный функционал. Его ограниченность заключается в отсутствии многопользовательской работы телефонии в рамках одного общего сеанса, в котором может работать несколько пользователей одновременно, а также в весьма посредственном хранении данных, полученных из звонка через телефонию (диспетчер может оставить только комментарий во время звонка, больше ничем полезным он не может дополнить документ телефонного звонка, ни вручную, ни автоматически).

Если рассматривать имеющийся принцип работы с заказ-нарядами с точки зрения диспетчера, то очень заметен такой недостаток как ручной поиск и назначение мастера, причем в рамках заказ-наряда отсутствует фильтрация по городу организации, в котором планируется ремонт. Это вызывает в свою очередь дополнительную проверку сотрудника со стороны диспетчера, чтобы убедиться в том, что выбранный им мастер находится именно в том городе, что и заказчик, запросивший ремонт или диагностику прибора.

1.3. Цель и задачи

Исходя из вышеописанных проблем была поставлена цель повышения эффективности бизнес-процессов, за счёт более оперативного поиска и назначения мастера, а также за счёт фиксации более детальной информации по заказам.

Для достижения этой цели поставлены следующие задачи:

- интеграция конфигурации «1С: Управление сервисным центром» с конфигурацией «1С: Парус-Софтфон» для фиксации звонков по заказу с детализацией и для звонков контрагентам;
- автоматизация процесса публикации заказ-нарядов и назначения на них сотрудников с последующим оповещением в программном приложении «РемТех» на Android;
- создание и использование табеля рабочего времени за месяц для более детального учета рабочего времени сотрудника.

Если выполнение второго пункта не предусмотрено ни типовой конфигурацией, ни интеграцией с какой-либо другой конфигурацией ввиду узкой специализации задачи, достигающей этой цели, то для выполнения первого пункта помимо вышеописанной интеграции есть другие решения.

Для того, чтобы выполнить вышеуказанные задачи, необходимо предварительно спроектировать то, как будут реализованы доработки и в чём они будут эффективнее уже имеющихся решений в конфигурации.

Глава 2. Проектирование программного обеспечения

2.1 Архитектура

2.1.1. Интеграция 1С:Рарус-Софтфон с 1С: Управление сервисным центром

2.1.1.1. Сравнение 1С:Рарус-Софтфон с другими решениями

Рассмотрим существующие на данный момент альтернативные решения более подробно:

1. Система звонков в конфигурации «1С: Управление сервисным центром»

В типовой конфигурации «1С: Управление сервисным центром» предусмотрена возможность звонков клиентам. Однако эта система имеет один большой недостаток: отсутствие фиксации записи звонка и его последующей привязки к клиенту, что ведет к проблеме низкой детализации по заказу.

2. Интеграция конфигурации «1С: Управление сервисным центром» с конфигурацией «МИКО-Телефония»

Конфигурация «МИКО-Телефония» имеет гораздо больший функционал по сравнению с предыдущей альтернативой, а именно:

- поддержка Asterisk;
- возможность расширенной аналитики звонков [1];
- удобное взаимодействие для облачных решений 1С.

Однако есть недостатки по сравнению с конфигурацией «1С: Рарус-Софтфон», а именно:

- отсутствие штатных средств для взаимодействия с телефонией для 1С CRM;
- ограниченное число пользователей в Asterisk (до 50 пользователей);

- отсутствие поддержки аппаратного IP АТС (например, АТОЛ);
- недостаточная отказоустойчивость относительно Парус-Софтфона.

Поэтому данное решение также не подходит, т.к. конфигурация «1С: Управление сервисным центром» включает в себя использование CRM и требует количества пользователей больше 50.

3. Использование других софтфонов (например, X-Lite)

Софтфон X-Lite весьма хорошо справляется с локальной фиксацией входящих и исходящих звонков, однако у него также есть свои недостатки, а именно:

- поддержка только протокола SIP (ТAPI-протокол отсутствует) [2];
- невозможность корректной интеграции с конфигурациями 1С по причине того, что к нему нельзя подключиться через СОМ-соединение.

После рассмотрения и анализа вышеописанных решений интеграции «1С: Управление сервисным центром» с софтфоном было принято решение в качестве конфигурации для софтфона использовать «1С: Парус-Софтфон».

2.1.1.2. Общая архитектура

При звонке из конфигурации «1С:Управление сервисным центром» (из документа «Событие», из справочника «Контрагенты» и т.д.) происходит обращение к конфигурации 1С:Софтфон (интеграция с телефонией), а именно к клиентскому и серверному модулю, методы которых приводят к вызову клиент-серверного модуля. После этого с помощью предварительной инициализации в основной конфигурации открывается клиентская панель 1С:Парус-Софтфон, которая принимает настройки, сформированные в серверном модуле 1С:Парус-Софтфон. Далее на сервере происходит коммутация АТС через систему Asterisk с заранее

прописанной в ней маршрутизацией. После этого через API серверная часть 1С:Рарус-Софтфона получает данные о звонке из АТС и отображает эти параметры в клиентской части 1С:Рарус-Софтфон.

После того, как клиент принял звонок, на стороне Asterisk звонок обрабатывается через один из основных доступных кодеков - G.729 или GSM. После звонка в Asterisk фиксируется запись в формате .gsm [5]. Также доступна запись звонка в формате .wav для того, чтобы в 1С:Рарус-Софтфон её можно было прослушать. Софтфон получает список звонков из заранее созданной сетевой папки, в которой они хранятся.

По окончании звонка в конфигурации «1С:Управление сервисным центром» создается документ «Телефонный звонок» со следующими данными:

- номер телефона;
- имя абонента;
- ID звонившего;
- ID звонка;
- время звонка;
- ссылка на файл с аудиозаписью (если есть);

и т.д.

Визуально архитектура описана на рис. 5.

2.1.1.2.1. Доработки в системе

Красным цветом помечены те модули, в которых были произведены доработки для корректной работы интегрированной системы.

Опишем их более подробно.

2.1.1.2.1.1. 1С: Рарус-Софтфон (клиент и сервер)

- список линий и модулей АТС – настройка SIP-линий и модулей АТС, появившихся в результате соединения с модулем Asterisk и

последующее распределение рабочего списка линий для каждого пользователя;

- мониторинг звонков через АТС – настройка отображения информации по всем входящим и исходящим вызовам, а также непринятым звонкам;
- учетные записи для работы с клиент-серверной частью Рарус-Софтфон – настройка учетных записей и привязка необходимых линий АТС для каждой из них на сервере для последующего взаимодействия с клиентской частью Рарус-Софтфон;
- Настройки работы сервера (порт для подключения, префиксы АТС, внутренние номера, история звонков в SQL Server и т.д.) – основные настройки в серверной части Рарус-Софтфон.

2.1.1.2.1.2. Конфигурация IC: СофтФон для конфигурации «IC: Управление сервисным центром»

- модуль общего назначения;
- клиент-серверный модуль общего назначения;
- Софтфон-Про (клиент) – изменена настройка отображения формы телефонного звонка при активации события открытия соответствующей формы;
- СофтФон-Про (сервер) – добавлена возможность получить контактное лицо по номеру телефона.

2.1.1.2.1.3. Конфигурация «IC: Управление сервисным центром»

- заказ-наряды – добавлена возможность звонка в документе «Заказ-наряд» клиенту или сотруднику, а также добавлено отображение истории звонков по клиенту и сотруднику с аудиозаписями;
- покупатели – добавлена возможность звонка в справочнике «Покупатели» клиенту с последующей записью в список

телефонных звонков по клиенту и возможностью просмотра истории звонков;

- контрагенты – добавлена возможность звонка в справочнике «Контрагенты» контрагенту с последующей записью в список телефонных звонков по клиенту и возможностью просмотра истории звонков;
- взаимодействие с клиентами.

Более подробно изменения каждого из вышеуказанных модулей описаны в листинге в приложении А.

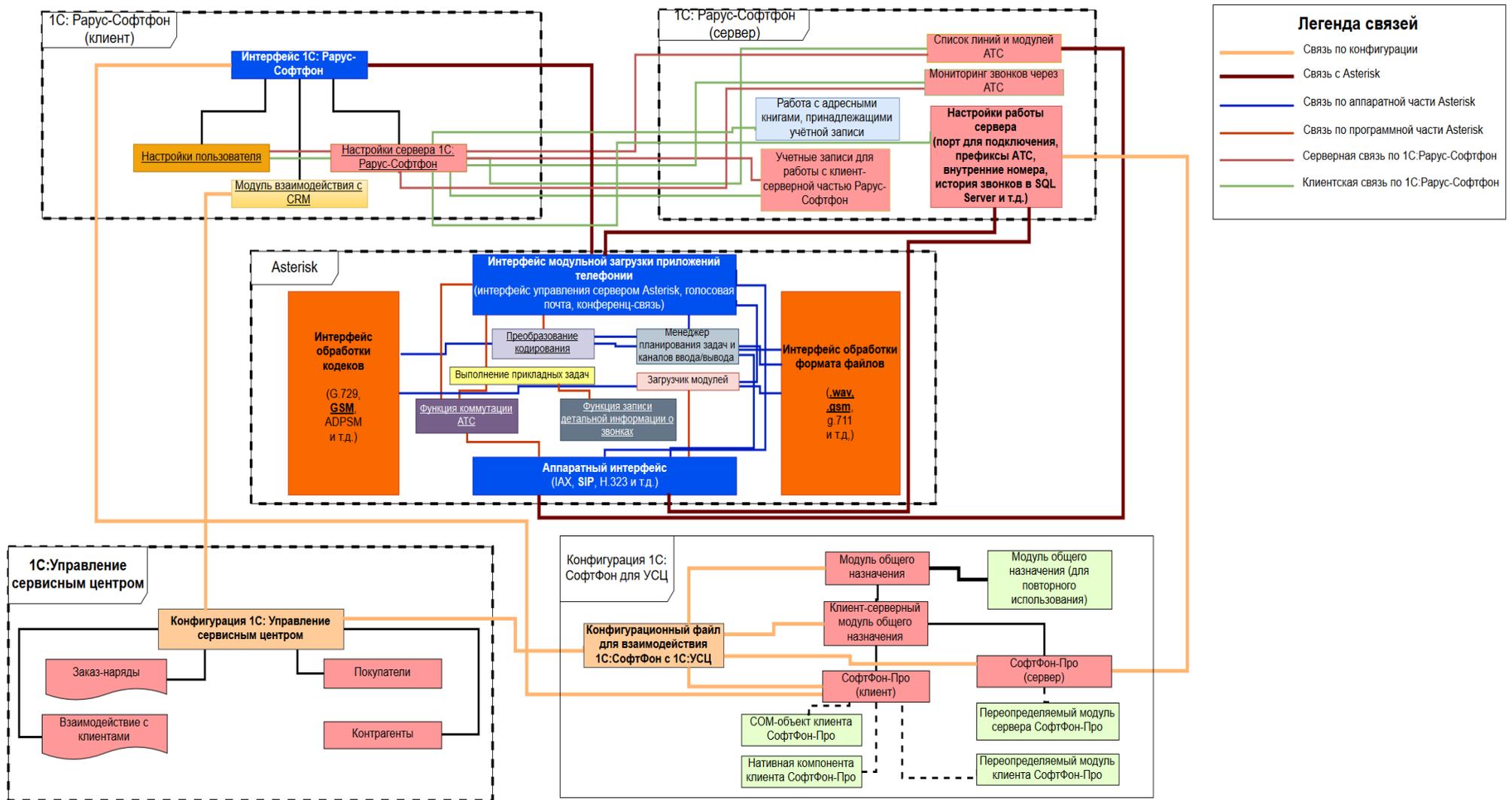


Рис. 5. Визуализация интеграции 1C:Парус-Софтфон с 1C: Управление сервисным центром

2.1.2. Публикации заказ-нарядов и назначения мастеров

В левой части формы обработки выводятся непроведённые заказ-наряды без мастера (см. рис. 6). В правой части на основании выбранного ранга и уровня подбираются мастера, которые:

- соответствуют нужному рангу и уровню, а именно:
 - попадают под следующие условия при формировании рангов:
 - 1 ранг: средний чек от 3501 до 10000 р. при минимальном количестве заказов в размере 20 шт. и процент отказов диагностик менее 30%;
 - 2 ранг: средний чек от 3501 до 10000 р. при минимальном количестве заказов в размере 15 шт. и процент отказов диагностик менее 30%;
 - 3 ранг: средний чек от 3501 до 10000 р. при минимальном количестве заказов в размере 20 шт. и процент отказов диагностик от 30% до 45%;
 - 4 ранг: средний чек от 3501 до 10000 р. при минимальном количестве заказов в размере 10-15 шт. или средний чек от 2000 до 3500 р. при минимальном количестве заказов в размере 15 шт. и процент отказов диагностик менее 30%;
 - 5 ранг: средний чек от 3501 до 10000 р. при минимальном количестве заказов в размере 5 шт. или средний чек от 2000 до 3500 р. при минимальном количестве заказов в размере 10 шт. и процент отказов диагностик от 30% до 50%;
 - 6 ранг: средний чек от 3501 до 10000 р. при минимальном количестве заказов в размере 5 шт. или средний чек от 2000 до 3500 р. при минимальном количестве заказов в размере 10 шт. и процент отказов диагностик менее 30%.

- соответствуют максимально доступному уровню ремонта и ранга:
 - 1 ранг: 1 уровень;
 - 2 ранг: 1 (реже - 2) уровень;
 - 3 ранг: 2 уровень;
 - 4 ранг: 2 уровень;
 - 5 ранг: 3 (в очень редких случаях - 2) уровень;
 - 6 ранг: 3 уровень
- способны выполнять тот тип работ, который указан в заказе.

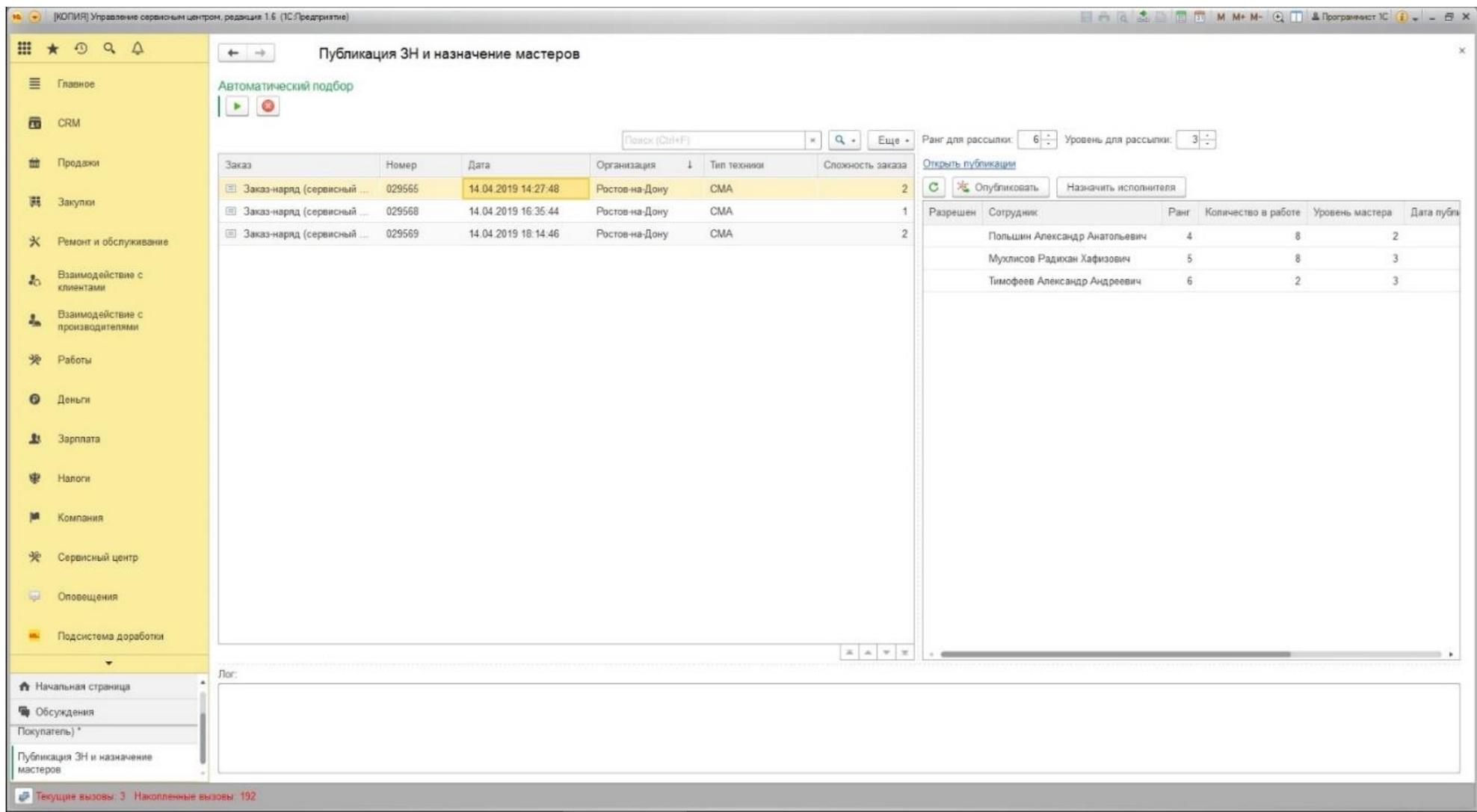


Рис. 6. Форма обработки публикации заказ-наряда и назначения мастеров

После этого при нажатии кнопки обработка подбирает наиболее доступного (у которого наибольший запас доступных работ) из них и у которого и производит следующие действия:

- проводит ранее непроведенный заказ-наряд со статусом «Принят в ремонт» (пример. см. на рис. 7);
- через Android API вызывает метод оповещения выбранного мастера в виде push-уведомления;
- переходит к следующему непроведенному заказу.

Визуально архитектура публикации заказ-нарядов и назначения мастеров описана на рис. 8.

2.1.2.1. Доработки в системе

Красным цветом помечены те взаимодействия, в которых были внесены изменения для корректной работы обработки. Опишем их более подробно.

- список непроведенных заказ-нарядов – сформированный запросом список заказ-нарядов, в которых не указаны сотрудники для дальнейших работ по ним;
- список доступных (разрешенных) мастеров – сформированный запросом список тех мастеров, которые относятся к тому городу, в котором нужно выполнить заказ-наряд, имеют уровень ремонта не ниже того, что соответствует типу ремонта из заказ-наряда и имеют текущее количество работ, не превышающее максимальное для ранга количество работ;
- ранг и уровень для рассылки – минимальные ранг и уровень в качестве входного параметра для обработки;
- обработка публикации заказ-нарядов и назначения мастеров – процесс работы обработки показан в главе «Бизнес-процессы».

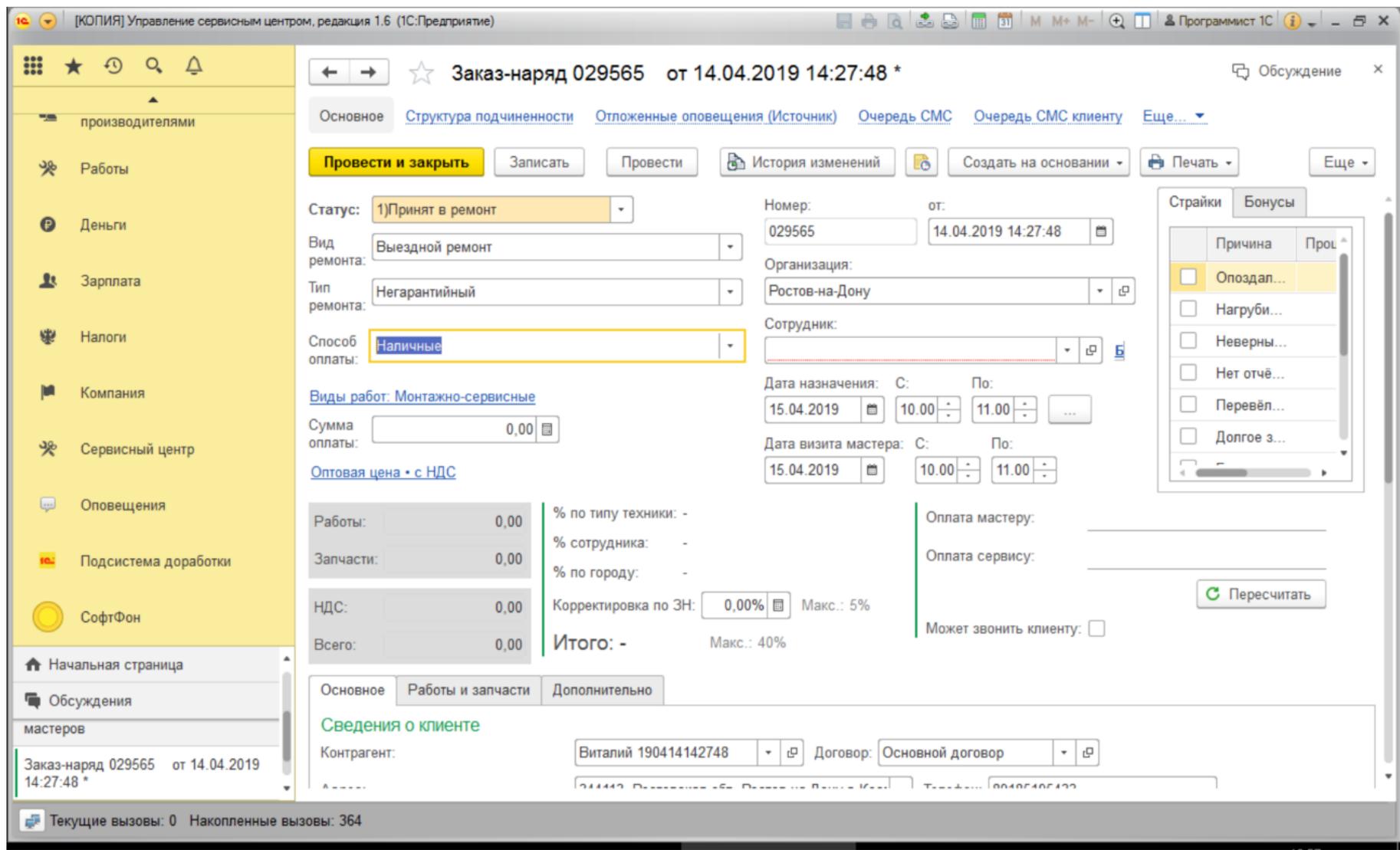


Рис. 7. Заказ-наряд со статусом «Принят в ремонт»

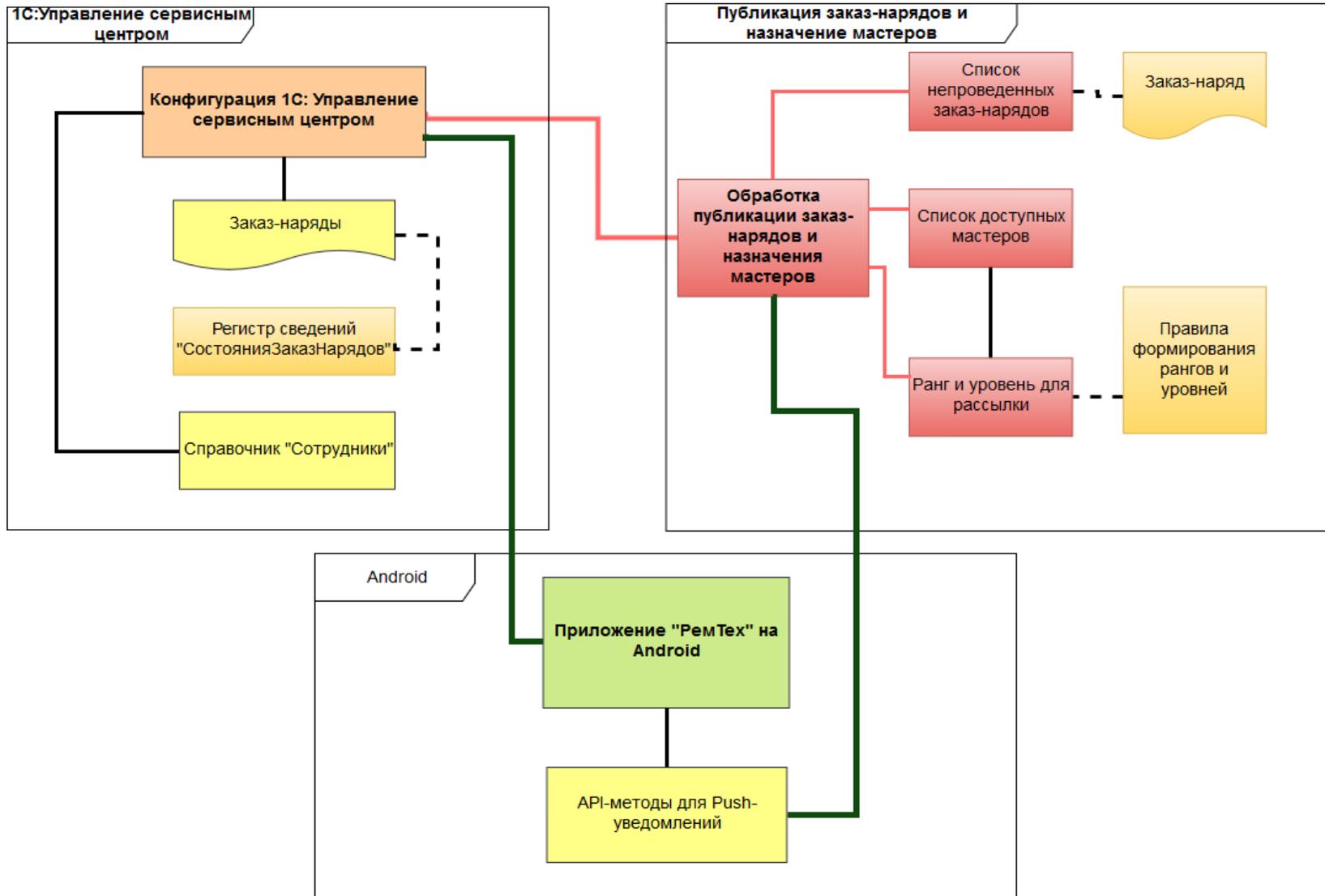


Рис. 8. Визуализация публикации заказ-наряда и назначения мастеров

2.1.3. Табель рабочего времени на месяц

Табель рабочего времени в компании РемТех предназначен для того, чтобы для каждого филиала компании можно было учесть время работы и его характер (например, полный день/полдня/больничный) каждого работника на каждый проведенный по табелю месяц в разрезе рабочего дня.

Он основан на регистре сведений «Отклонения от графиков работы ресурсов», который содержит в себе количество часов работы по сотруднику на каждый день месяца. В компании количество часов работы представляет собой следующую структуру (показана в табл. 16):

Таблица 16. Количество часов в табеле на день и соответствующий ему статус

Значение	Статус
0	сотрудник не трудоустроен
1	сотрудник полностью заблокирован в системе
2	сотрудник заблокирован для получения заказов
3	у сотрудника выходной день
4	сотрудник на больничном
5	сотрудник работает полдня
6	у сотрудника прогул по невыясненной причине
7	сотрудник опоздал на работу
8	сотрудник работает

Данный табель формируется и проводится ежемесячно. Корректировки в табеле вносятся диспетчером.

В дальнейшем этот табель используется как в документе «Заказ-наряд» для того, чтобы проверить то, можно ли назначить мастера на созданный заказ-наряд (если на день создания заказ-наряда у мастера стоит значение 5, 7 или 8, то мастера можно рассматривать на назначение, иначе - нет), так и в обработке для публикации заказ-нарядов и назначения на них мастеров по той же причине.

Подробная реализация показана в приложении В.

Визуально архитектура описана на рис. 9.

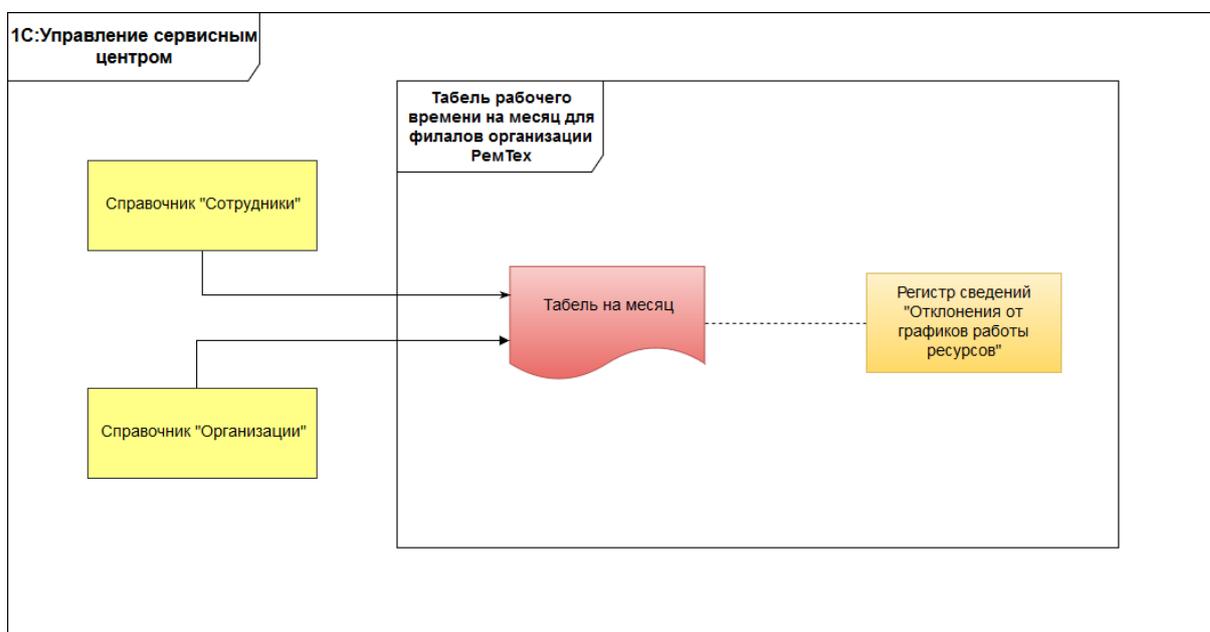


Рис. 9. Визуализация архитектуры формирования табеля рабочего времени на месяц для филиалов организации РемТех

2.2. Бизнес-процессы

2.2.1. Бизнес-процесс работы со звонками и телефонией

2.2.1.1. Бизнес-процесс работы со звонками и телефонией до интеграции с 1С:Рарус-Софтфон

Диспетчер может совершить звонок только из двух элементов конфигурации: справочник «Контрагенты» для уточнения у клиента деталей и статуса оплаты по ремонту и справочник «Организации» для звонка мастеру, принадлежащему той или иной организации. В первом

случае для диспетчера открывается упрощенная форма телефонного звонка, в котором пользователь может только оставить комментарий по деталям заказа. По окончании звонка диспетчер проводит документ «Телефонный звонок» с клиентом и комментарием по заказу. Ограничение заключается в том, что в данном справочнике доступно только одно поле с контактными данными (номер телефона, имя и фамилия контрагента). Во втором случае открывается аналогичная форма, но без возможности добавления комментария по звонку.

На рис. 10. показан текущий бизнес-процесс работы со звонками в системе.

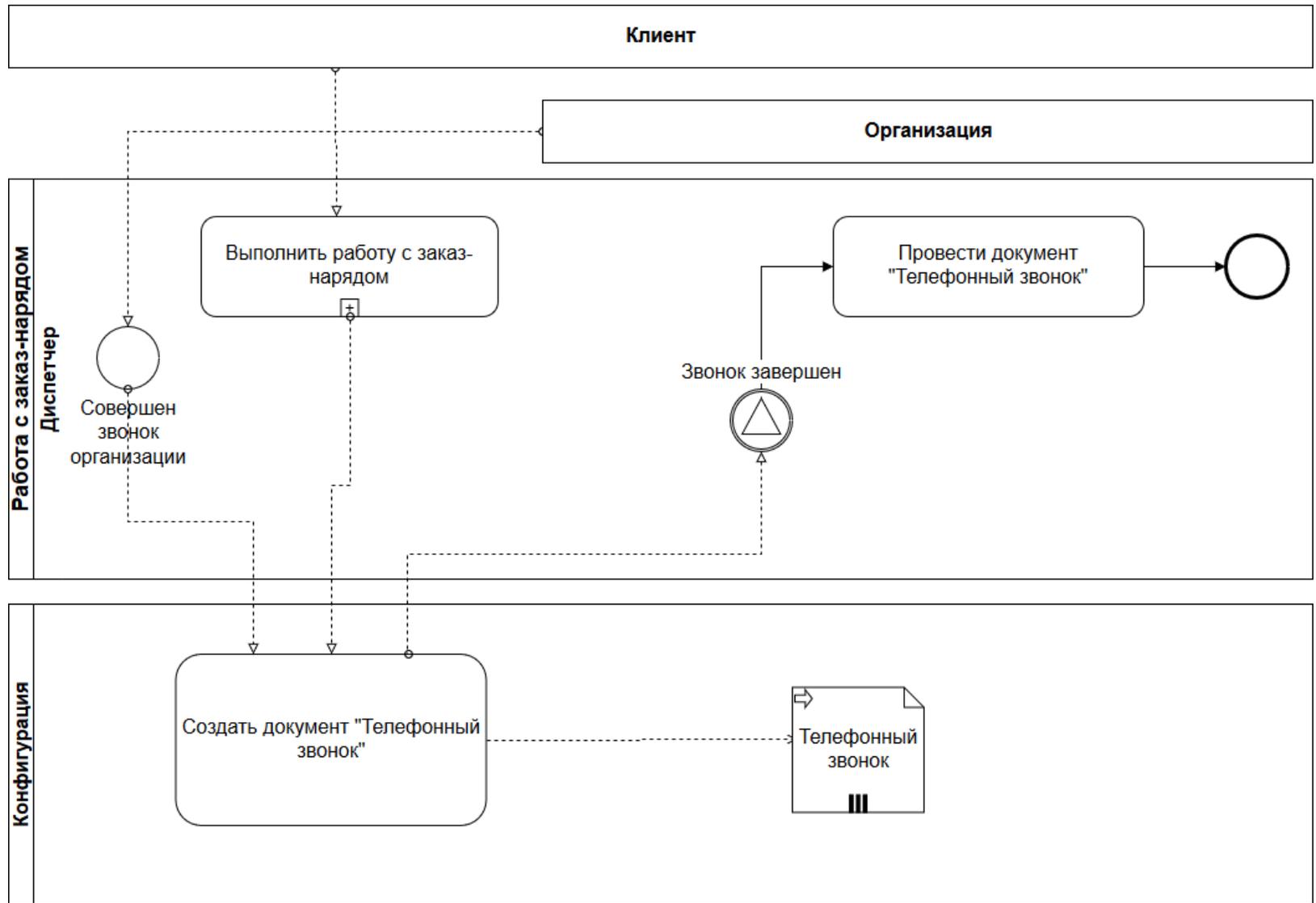


Рис. 10. Процесс работы со звонками в системе

2.2.1.2. Бизнес-процесс работы со звонками и телефонией после интеграции с 1С:Рарус-Софтфон

После применения интеграции «1С:Рарус-Софтфон» с конфигурацией «1С:Управление сервисным центром» диспетчер может звонить не только из справочника «Контрагенты» и справочника «Организации», но также и из документа «Заказ-наряды» для того, чтобы диспетчер мог с легкостью в выбранном заказе-наряде сразу позвонить клиенту, минуя переход в справочник или же позвонить мастеру, который указан в заказе-наряде (телефон определяется путем поиска мастера в справочнике «Сотрудники» с помощью запроса). Также диспетчер может позвонить и из справочника «Контрагенты» с возможностью выбора одного из нескольких контактных номеров. После принятия звонка создается документ «Телефонный звонок» с ссылкой на того, кому звоним, а также с имеющимися у него номерами телефона в табличной части «Как связаться». По окончании звонка за этим документом закрепляется соответствующая аудиозапись для того, чтобы диспетчер мог вспомнить ранее обговоренные детали заказа-наряда и его текущего статуса. Также диспетчер может звонить из конфигурации с помощью панели управления Софтфон (при условии, если эта панель включена в общих настройках Софтфона). В таком случае также создается документ «Телефонный звонок», однако, если номера нет ни в списке контактных лиц, ни в списке сотрудников, то в поле абонент будет выведено значение «!!!Не определен». При этом всё остальное (в том числе и запись звонка) все равно будет зафиксировано.

Бизнес-процесс работы со звонками и телефонией после интеграции с 1С:Рарус-Софтфон показан на рис. 11.

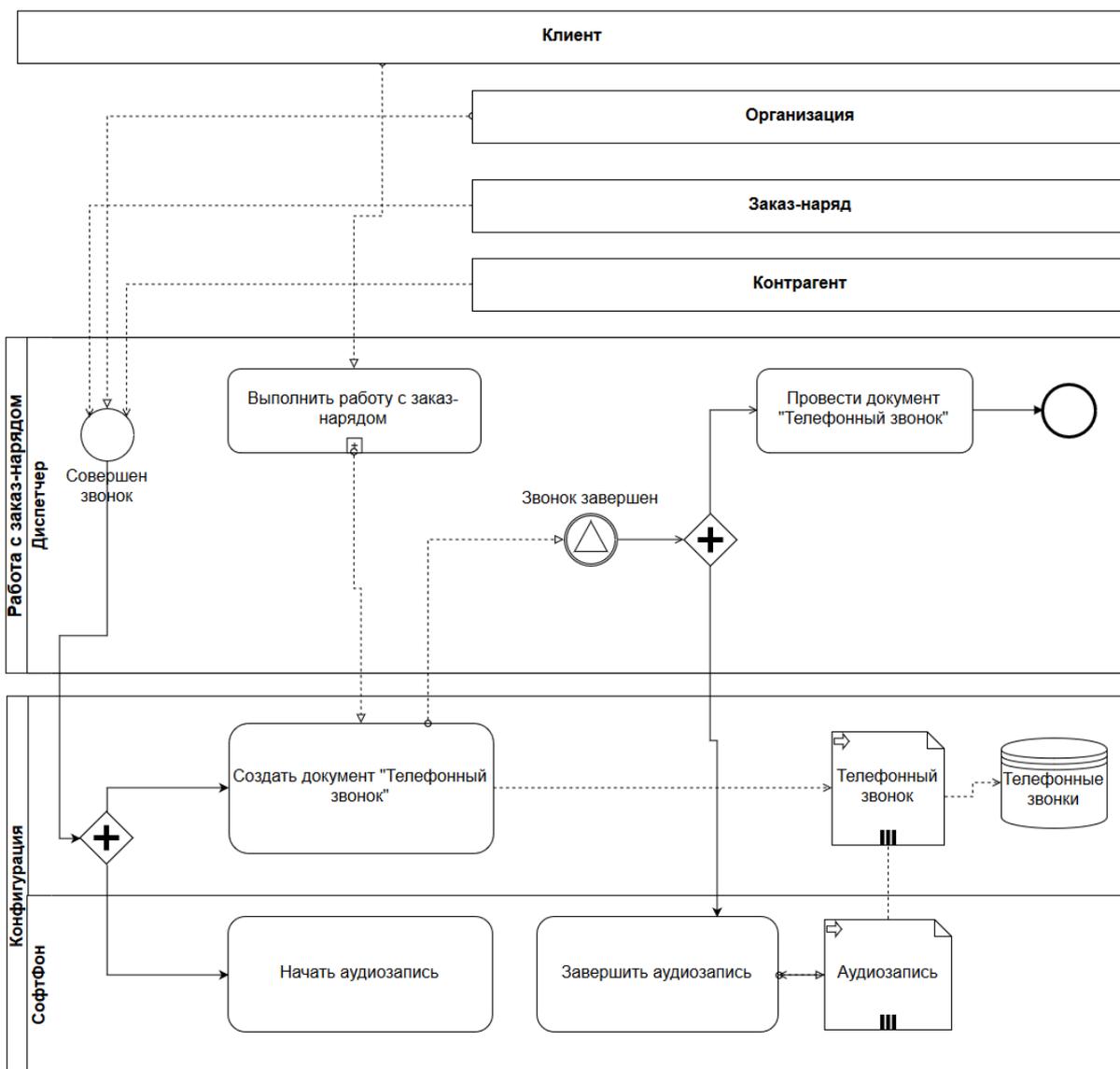


Рис. 11. Бизнес-процесс работы со звонками и телефонией после интеграции с 1С:Рарус-Софтфон

2.2.1.3. Сравнение эффективности бизнес-процессов «до» и «после»

В версии бизнес-процесса «до», показанной на рис. 10, в системе можно совершить звонок только из справочника «Организации» или из справочника «Контрагенты», при этом у диспетчера для фиксации телефонного звонка параллельно должен быть открыт заказ-наряд для того, чтобы во время звонка контрагенту или мастеру из организации в документе «Телефонный звонок» фиксировать детали по заказ-наряду в поле комментария.

В версии бизнес-процесса «после», показанной на рис. 11, диспетчер может помимо стандартного звонка из справочника «Организации» или из справочника «Контрагенты» без проблем совершить звонок клиенту или мастеру из организации прямо из заказ-наряда. Также по окончании звонка за документом «Телефонный звонок» фиксируется запись по вызываемому субъекту из Софтфона, что положительно влияет на детализацию информации о заказ-наряде. Записанный телефонный звонок можно в дальнейшем увидеть в истории телефонных звонков как из заказ-наряда, так и из справочника «Организации» или из справочника «Контрагенты».

Таким образом, можно утверждать, что реализация версии «после» является более широко применимой и более детализированной в конфигурации «1С:Управление сервисным центром» по сравнению с версией «до», показанной на рис.10.

2.2.2. Бизнес-процесс публикации заказ-нарядов и назначения на них мастеров

2.2.2.1. Бизнес-процесс публикации заказ-нарядов и назначения на них мастеров до появления обработки

Клиент звонит диспетчеру и описывает проблему, от которой зависит сложность заказа и минимальный доступный ранг мастера, который сможет принять этот заказ-наряд. Диспетчер должен вручную найти мастера, который удовлетворяет вышеописанным условиям, а также доступный для работы (т.е. текущее количество работ у мастера не должно превышать максимальное для ранга количество работ). После того, как диспетчер согласовал детали заказа с клиентом и подобрал подходящего мастера, диспетчер фиксирует заказ-наряд и проводит его. В противном случае заказ-наряд остается непроведённым до того момента, пока диспетчер не найдет нужного и доступного мастера для заказ-наряда.

Бизнес-процесс работы с заказ-нарядом до появления обработки показан на рис. 12.

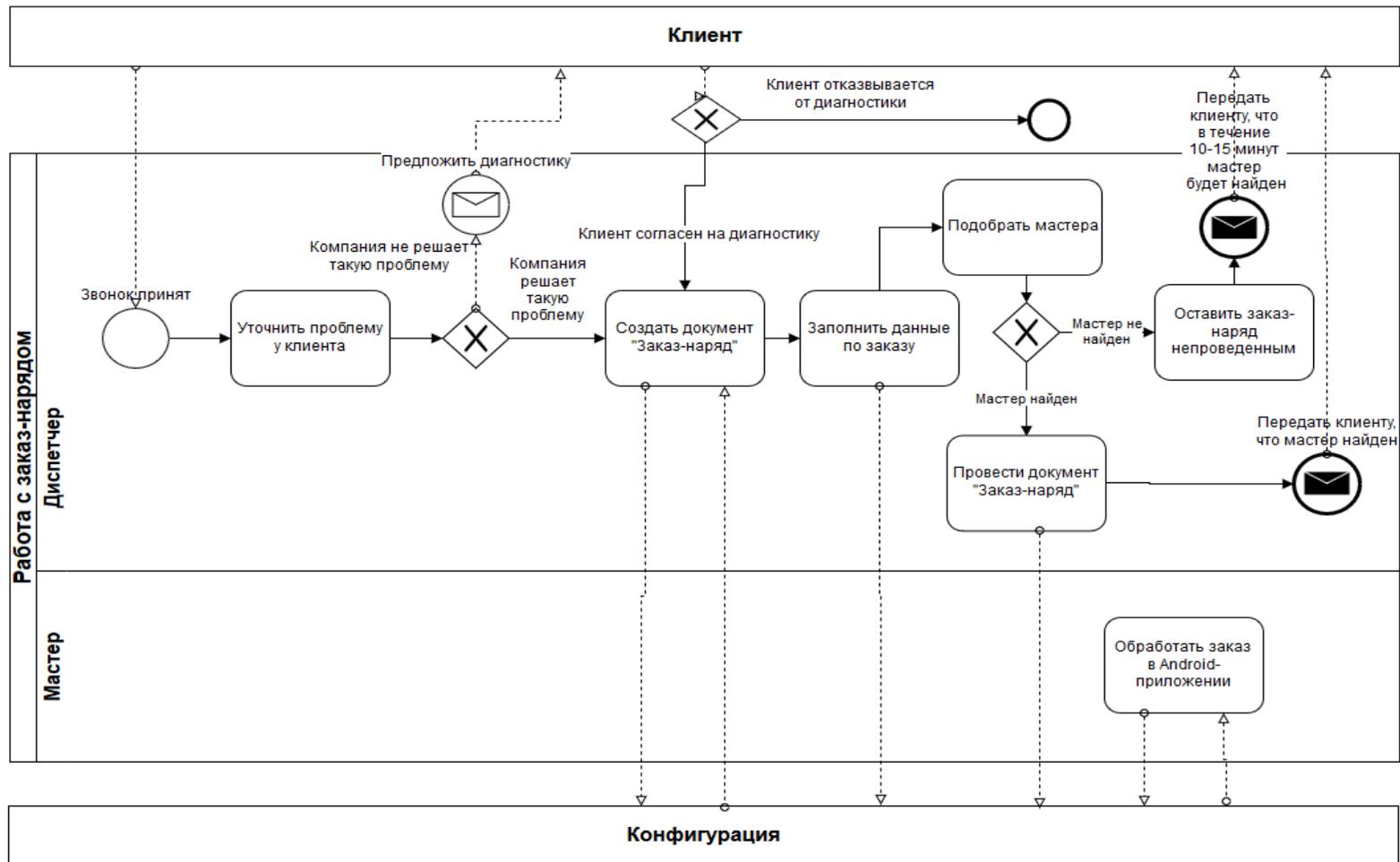


Рис. 12. Бизнес-процесс работы с заказ-нарядом до обработки

2.2.2.2. Бизнес-процесс публикации заказ-нарядов и назначения на них мастеров после появления обработки

После создания и установки обработки в конфигурацию в разделе «Администрирование» диспетчеру достаточно просто создать заказ-наряд с описанием проблемы, после чего обработка с периодичностью в каждые 10 минут будет автоматически подбирать мастеров для непроведённых заказ-нарядов согласно их рангу и уровню сложности ремонта, а также расположению заказа (например, если заказ-наряд на ремонт поступил из Тюмени, то обработка будет рассматривать только тюменских мастеров). В случае, если за это время по какому-либо из заказ-нарядов так и не был найден мастер, обработчик подберет из всех мастеров того, чей ранг и минимальный уровень сложности ремонт наиболее близки к тому, что указаны в заказ-наряде. После этого данному мастеру придет push-уведомление о возможности поработать по желанию сверх максимального количества работ за доплату. В свою очередь, мастер сам решает, хочет ли он получить надбавку за лишнюю работу или отказаться и остаться с тем же максимальным количеством работ.

Бизнес-процесс работы с заказ-нарядом после появления обработки показан на рис. 13.

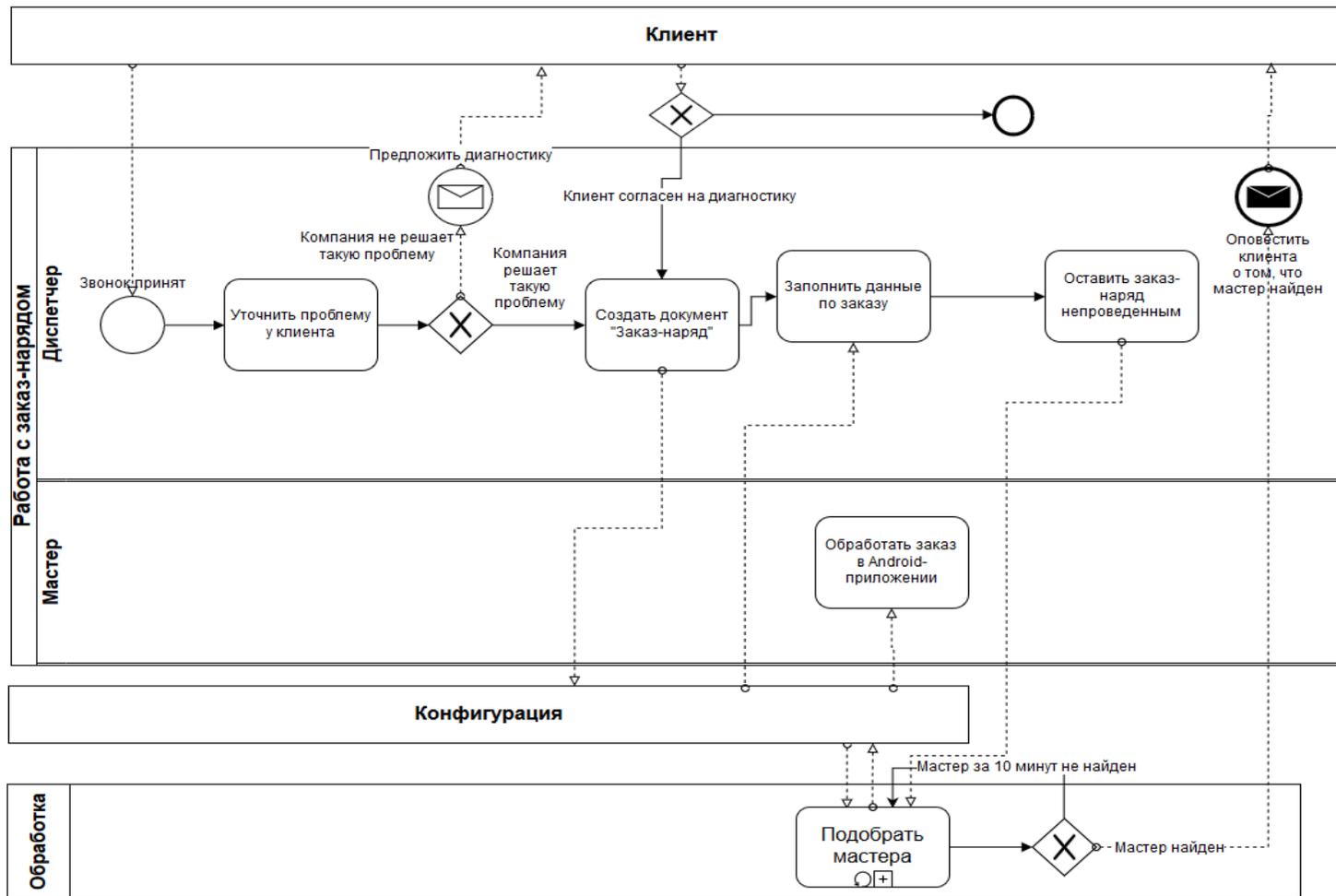


Рис. 13. Бизнес-процесс работы с заказ-нарядом после обработки

На рис. 14 показан циклический (т.к. он запускается с периодичностью в каждые 10 минут и циклично проходит по всем непроведенным заказ-нарядам) подпроцесс «Подобрать мастера», в котором более детально показан принцип работы обработчика.

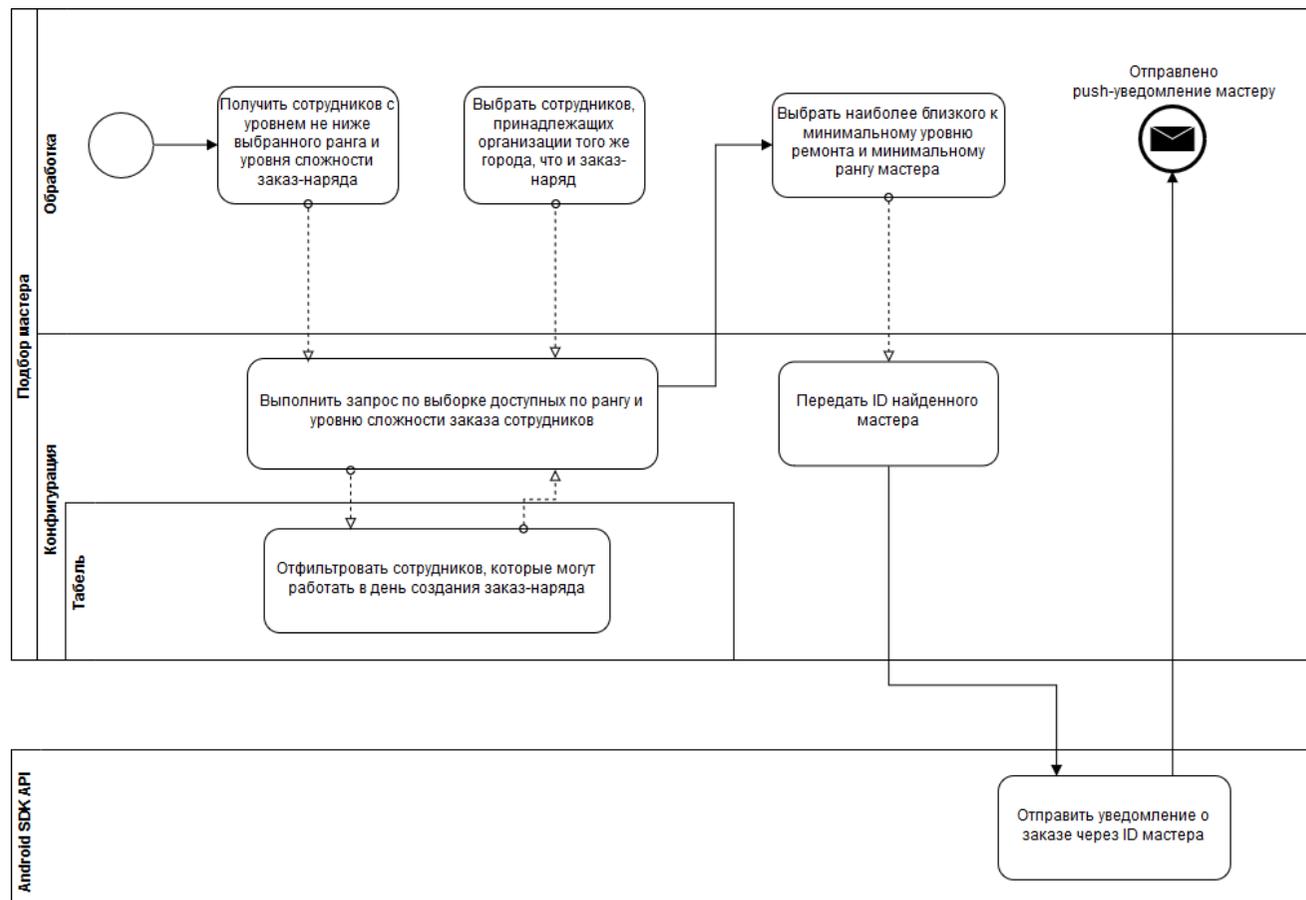


Рис. 14. Подпроцесс «Подобрать мастера»

2.2.2.3. Сравнение эффективности бизнес-процессов «до» и «после»

В версии бизнес-процесса «до», показанном на рис. 12, у диспетчера 5 основных задач (одна из них разделена на две вариации – если мастер найден или не найден соответственно), причем задача «Подбор мастера» имеет весьма неопределенное количество времени: от нескольких секунд до нескольких часов, что ведет к неопределенному времени работы всего бизнес-процесса.

В версии бизнес-процесса «после», показанном на рис. 13, задача подбора мастера полностью переходит обработчику, тем самым снимая задачу его поиска в списке сотрудников для диспетчера. Такой подход сокращает время выполнения задачи до 10 минут в худшем случае (данный показатель был вычислен с помощью инструмента 1С «замер производительности» [7]). Помимо этого, обработка затрагивает не только текущий заказ-наряд, но также и все остальные непроведенные заказ-наряды.

Таким образом, можно утверждать, что реализация версии «после» является более оптимизированной «1С:Управление сервисным центром» по сравнению с версией «до», показанной на рис. 12 по причине того, что основная задача (задача поиска и назначения мастера) была автоматизирована за счёт обработчика и распространена сразу на несколько заказ-нарядов.

2.3. Алгоритмы

2.3.1. Алгоритм работы интегрированной телефонии

Независимо от типа звонка (исходящий/входящий), через COM-соединение клиентской части Рарус-Софтфон заполняются параметры звонка (см. структуру звонка на табл. 17), после чего идет подключение к клиентской и серверной части Рарус-Софтфон. После этого клиентская часть Рарус-Софтфона подключается к серверу с системой аудиозаписи CLON для фиксации звонков через серверную часть Рарус-Софтфона, которая в свою очередь получает структуру настроек для доступа к серверу аудиозаписи. После этого клиентская часть Рарус-Софтфона применяет настройки из серверной панели (префиксы, код города и т.д.). Далее из клиентской части Рарус-Софтфона вызывается форма звонка и в конфигурации пользователю сообщается о том, что происходит вызов звонка. Если звонок принят, то пользователю в конфигурации сообщается о создании документа, после чего создается документ телефонного звонка. По завершении звонка происходит обращение к серверу аудиозаписей CLON через серверную панель Рарус-Софтфон. После этого к документу телефонного звонка прикрепляется аудиозапись.

На рис. 14. показана работа алгоритма в виде блок-схемы.

Таблица 17. Структура звонка [8]

Переменная	Тип	Описание
hCall	Число	Идентификатор линии.
LineName	Строка	Имя линии.
LineType	Строка	Тип линии.
CallerID	Строка	Номер звонящего.
CallerInfoName	Строка	Представление звонящего.
CalledId	Строка	Номер принимающего звонок.
CalledInfoName	Строка	Представление принимающего звонок.
State	Число	Состояние звонка.
Origin	Число	Направление звонка. (0 – Неопределенно, 1 – Внутренний исходящий, 2 – Внутренний входящий, 3 – Внешний входящий, 4 – Недоступно, 5 – Конференция, 6 – Входящий)
DopInfo	Строка	Дополнительная информация о звонке.
AvailableActions	Число	Доступные действия со звонком. (aa_Drop, aa_Answer, aa_Hold, aa_UnHold и т.д.)
AppValue	Произвольный	Произвольная, служебная информация, хранимая приложением.
ImageData	Строка	Картинка контакта в формате Base64.
ContactID	Строка	Идентификатор контакта во внешней учетной системе.

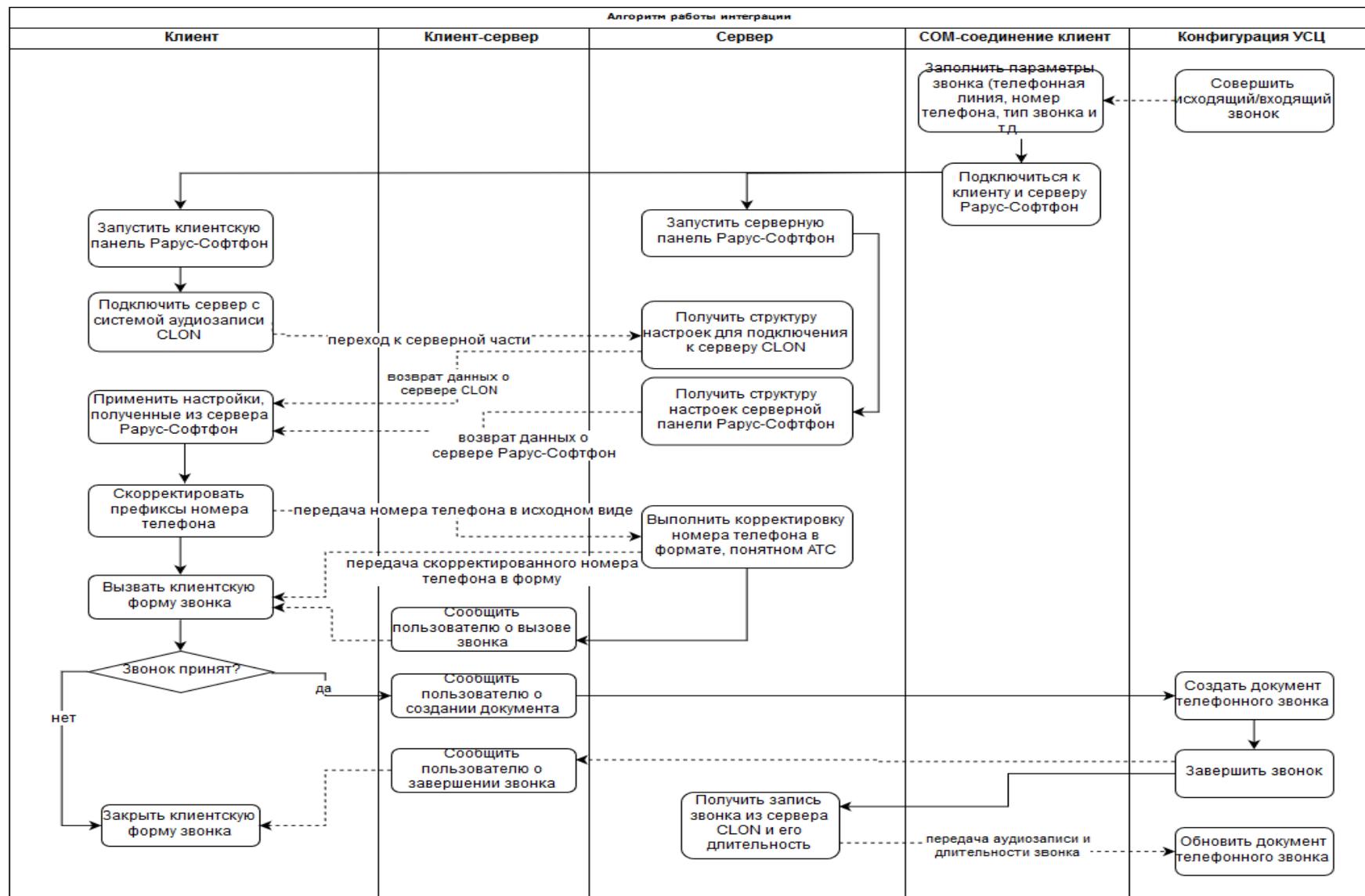


Рис. 14. Блок-схема алгоритма работы с телефонией Рарус-Софтфон.

2.3.2. Алгоритм публикации и назначения мастера в рамках одного заказ-наряда

При запуске обработки с входным рангом и уровнем сложности ремонта идет полный циклический проход по рангам и подцикл по всем уровням сложности ремонта, в каждом проходе происходит подбор сотрудника к заказу согласно рангу и уровню сложности ремонта. Подбор сотрудника к заказу основан на получении выборки сотрудников из запроса, учитывающего средний чек сотрудника, количество его работ, а также ранг (не ниже ранга в проходе) и уровень сложности выполняемого заказа (не ниже уровня в проходе) (см. пример структуры из табл. 18). Далее, если сотрудники в проходе найдены, происходит цикл по списку сотрудников, в котором проверяется, превышает ли текущее количество заказов у мастера максимальное для его ранга количество заказов. Если не превышает, то можно публиковать заказ-наряд мастеру и отправлять оповещение мастеру в Android-приложение (в противном случае - переход к следующему мастеру). Далее, мастер решает, хочет ли он выполнять заказ-наряд или нет. Если хочет, тогда в конфигурации происходит назначение заказ-наряда на мастера и его последующая фиксация (в противном случае - переход к следующему мастеру). После того, как был назначен мастер, у заказ-наряда меняется статус с розового цвета на зеленый в списке заказ-нарядов.

Далее происходит переход к следующему заказ-наряду. Публикация происходит с периодичностью раз в минуту, а назначение мастера - раз в 10 минут.

Блок-схема работы алгоритма показана на рис. 15.

Таблица 18. Соответствие среднего чека, процента отказов от диагностики и количества заказов рангу и максимальному уровню сложности.

сред.чек >=	сред.чек <	%отказ. от диаг> =	%отказ. от диаг<	min.кол заказов	max.кол заказов	Ранг	max.ур.сло жности
4 508	99 999	0	30	7	40	1	1
4 507	99 999	0	30	7	35	2	1
4 007	4 508	0	30	7	35	3	1
4 506	99 999	0	30	7	30	3	1
4 507	99 999	30	35	7	35	3	1
3 505	4 006	0	30	7	30	4	1
4 006	4 507	0	30	7	30	4	1
4 007	4 508	30	35	7	35	4	1
4 505	99 999	0	30	7	25	4	1
4 506	99 999	30	35	7	30	4	1
4 507	99 999	35	40	7	35	4	1
3 004	3 505	0	30	6	20	5	1
3 504	4 005	0	30	6	25	5	1
3 505	4 006	30	35	6	30	5	1
4 005	4 506	0	30	6	25	5	1
4 006	4 507	30	35	6	30	5	1
4 007	4 508	35	40	6	35	5	1
4 504	99 999	0	30	6	20	5	1
4 505	99 999	30	35	6	25	5	1
4 506	99 999	35	40	6	30	5	1
3 003	3 504	0	30	5	15	6	2
3 004	3 505	30	35	5	20	6	2
3 503	4 004	0	30	5	20	6	2
3 504	4 005	30	35	5	25	6	2
3 505	4 006	35	40	5	30	6	2
4 004	4 505	0	30	5	20	6	2
4 005	4 506	30	35	5	25	6	2
4 006	4 507	35	40	5	30	6	3
4 503	99 999	0	30	5	15	6	3
4 504	99 999	30	35	5	20	6	3
4 505	99 999	35	40	5	25	6	3

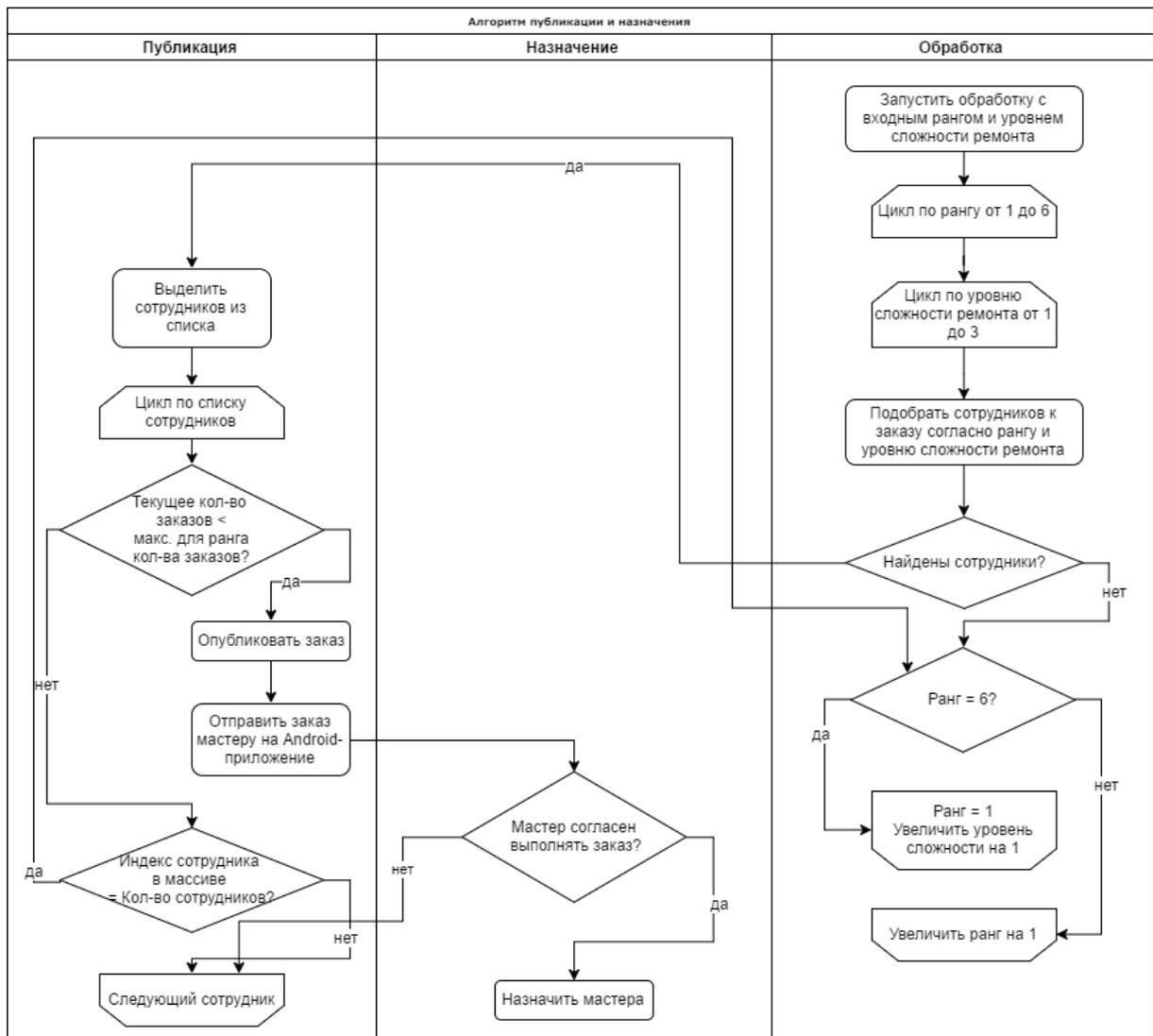


Рис.15. Блок-схема алгоритма публикации и назначения мастера на заказ-наряд.

После проектирования архитектуры, построения бизнес-процессов до и после интеграции конфигураций для телефонии и последующих доработок по ней и по назначению мастеров со сравнением и описания алгоритмов работы можно приступить к реализации вышеописанных проектных решений.

Глава 3. Реализация

3.1. Реализация архитектуры интеграции 1С:Рарус-Софтфон с 1С: Управление сервисным центром

3.1.1. Реализация в части 1С:Рарус-Софтфон

Для того, чтобы обеспечить работоспособность приема/вызова звонков, а также их прослушивания в конфигурации, нужно предварительно произвести настройку на серверной части и последующее подключение к ней на клиентской части Софтфона.

В первую очередь в настройках сервера прописываем порт для подключения к серверу Софтфона, а также прописываем настройку префиксов АТС. Эти настройки в дальнейшем будут использоваться как клиентской частью приложения Софтфон, так и конфигурацией 1С через СОМ-объект (настройки см. на рис. 16).

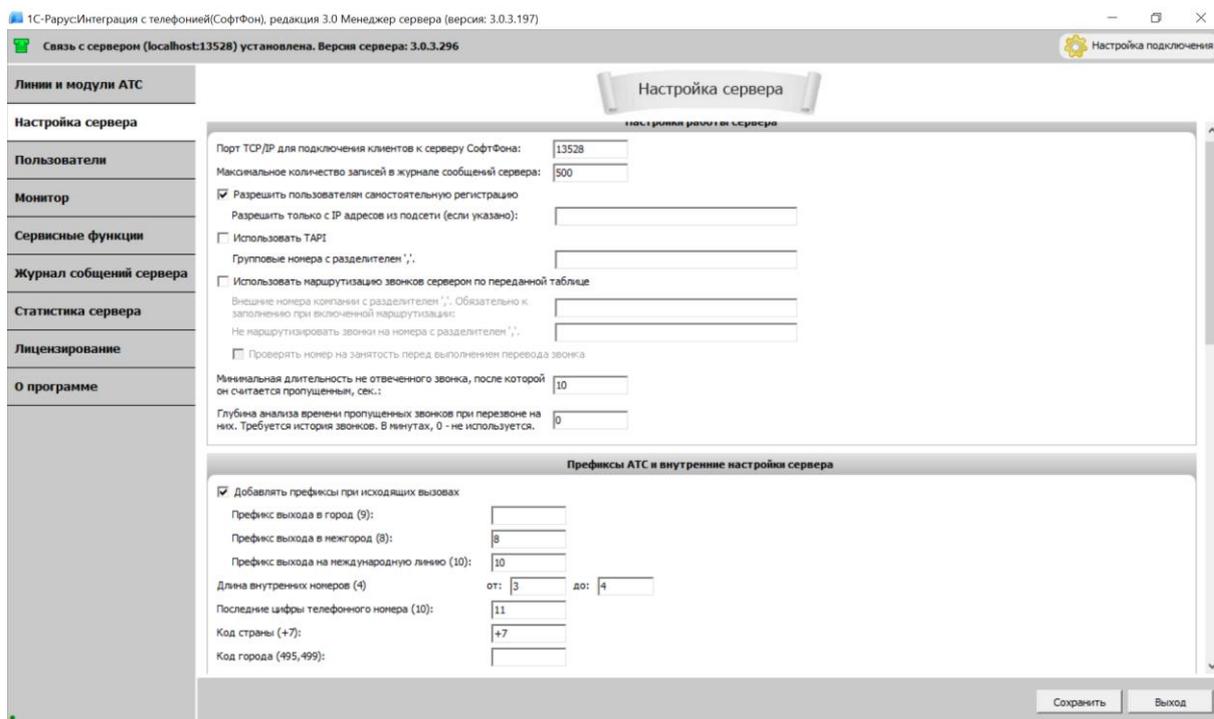


Рис. 16. Настройка сервера в менеджере сервера 1С-Рарус СофтФон

3.1.2. Реализация в модуле Asterisk

Помимо этого, на сервере Софтфона доступны настройки Asterisk (серверное ПО для VoIP-телефонии) для подключения к нему и

последующему доступу к записанным разговорам в формате .gsm или .wav, а именно: IP-адрес для доступа к Asterisk, порт, путь к сетевой папке с записями (при условии, что поставлена галочка «Использовать запись разговоров»), логин/пароль к Asterisk и т.д (настройки см. на рис. 17).

Наименование настройки	Значение
Представление модуля в Менеджере	Asterisk модуль
Префикс к имени линий	
Имя сервера Asterisk или его IP адрес	10.5.0.254
Порт сервера Asterisk	5038
Логин менеджера Asterisk	██████████
Пароль менеджера Asterisk	██
Контекст для исходящих звонков	
Запрос контекста и описания линий	<input checked="" type="checkbox"/>
Доп. параметр в команде Originate	
Использовать запись разговоров	<input checked="" type="checkbox"/>
Использовать команду "Monitor" вместо "MixMonitor"	<input type="checkbox"/>
Использовать FTP для получения записей разговоров	<input type="checkbox"/>

Рис. 17. Настройка Asterisk-модуля в менеджере сервера 1С-Рарус СофтФон

Более полная настройка Asterisk (маршрутизация, составление полного отчёта звонков, формирование тех или иных протоколов) доступна через веб-интерфейс FreePBX [6] (главное меню с отслеживанием по количеству пользователей и отказоустойчивости системы показано на рис. 18).

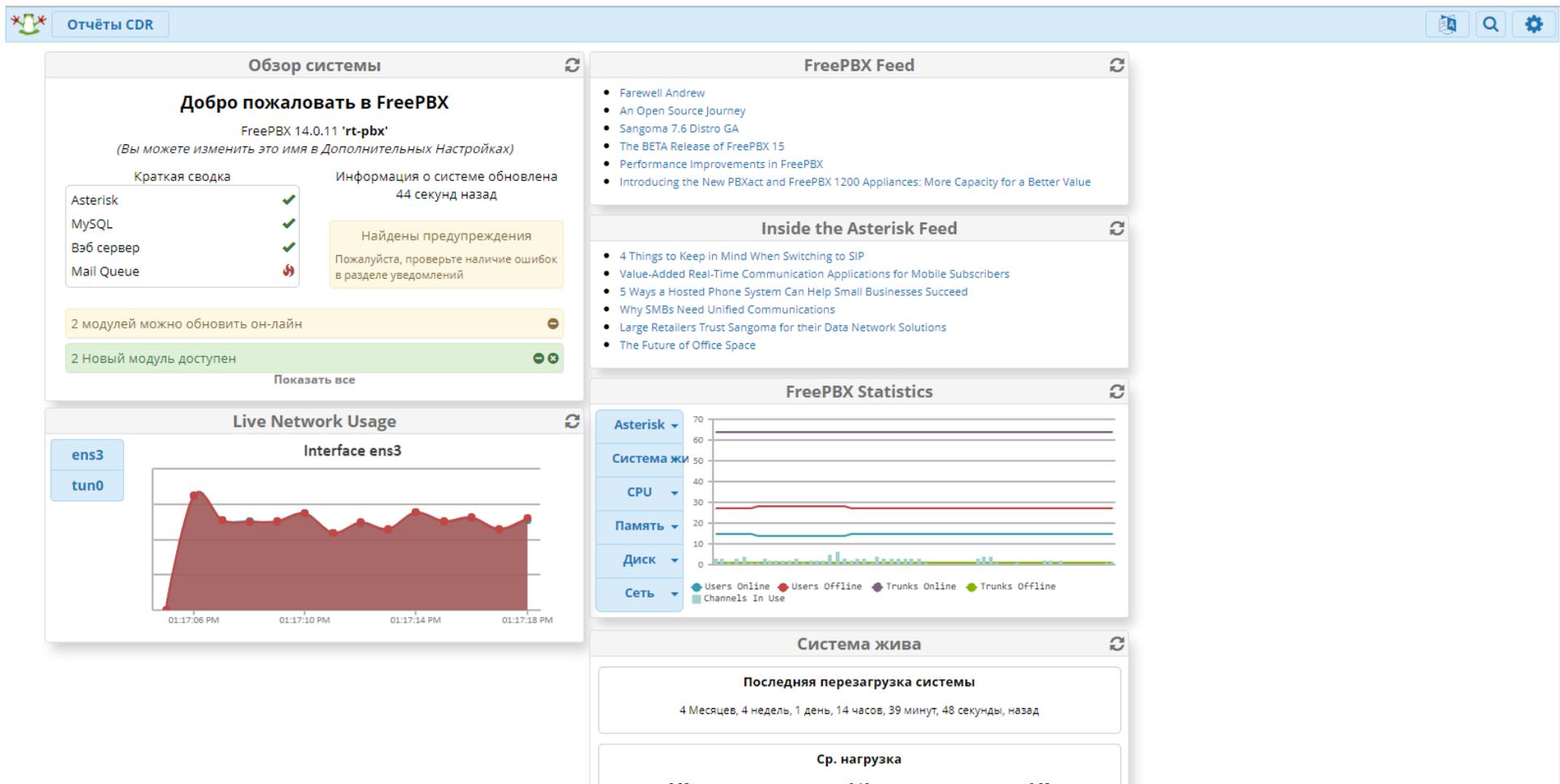


Рис. 18. Главное меню FreePBX

С помощью FreePBX можно сформировать список звонков за определенный период с отбором по различным параметрам (номер CallerID, имя CallerID, входящий номер и т.д.) (показано на рис. 19).

Отчёты CDR

Отчеты CDR

Отчет по звонкам

Сортировать по

- Дата звонка
- Номер CallerID
- Имя CallerID
- Номер исх. CallerID
- Входящий номер
- Куда звонил
- Имя назначения CallerID
- Пользовательское поле
- Код Учетки
- Длительность
- Статус звонка

Search Conditions

Между: 01 Апрель 2019 00:00 и: 31 Май 2019 23:59

Не: Начинается с: Содержит: Оканчивается на: Равно:

Между: и: Секунд

Любой статус Не:

Группировать по: День

Искать

Extra Options

Список звонков

Report Type: CSV File

Суммарный график

Result Limit: 100



FreePBX is a registered trademark of Sangoma Technologies Inc. FreePBX 14.0.11 is licensed under the GPL Copyright© 2007-2019



Рис. 19. Меню формирования отчета по списку звонков

В результате формирования отчета получается детализация звонков в том числе с описанием статуса каждого звонка и его идентификатору (см. рис. 20, личные данные скрыты).

Отчет по звонкам - Результат поиска 100 Звонки												
Дата звонка	Запись	Идентификатор	Имя звонящего	Исходящий CallerID	Входящий номер	Прил.	Куда звонил	Статус звонка	Длительность	Пользовательское поле	Учетка	
Tue, 21 May 2019 15:16						Playback	660	ANSWERED	00:06			
Tue, 21 May 2019 15:15						Dial	540	NO ANSWER	00:02			
Tue, 21 May 2019 15:15						Dial	540	NO ANSWER	00:02			
Tue, 21 May 2019 15:15						Dial	540	NO ANSWER	00:02			
Tue, 21 May 2019 15:15						Dial	540	NO ANSWER	00:02			
Tue, 21 May 2019 15:14						Dial	540	ANSWERED	02:20			
Tue, 21 May 2019 15:08						Dial	510	NO ANSWER	00:05			
Tue, 21 May 2019 15:08						Dial	510	ANSWERED	01:03			
Tue, 21 May 2019 15:06						Dial	012	NO ANSWER	00:03			
Tue, 21 May 2019 15:06						Dial	012	NO ANSWER	00:03			
Tue, 21 May 2019 15:06						Dial	012	ANSWERED	01:37			
Tue, 21 May 2019 15:06						Dial	012	NO ANSWER	00:03			
Tue, 21 May 2019 15:06						Dial	012	NO ANSWER	00:03			
Tue, 21 May 2019 15:06						Dial	012	NO ANSWER	00:03			
Tue, 21 May 2019 15:06						Dial	012	NO ANSWER	00:03			
Tue, 21 May 2019 15:06						Dial	1	NO ANSWER	00:20			
Tue, 21 May 2019 15:06						Dial	1	NO ANSWER	00:20			
Tue, 21 May 2019 15:05						Dial	501	NO ANSWER	00:03			
Tue, 21 May 2019 15:05						Dial	501	NO ANSWER	00:03			
Tue, 21 May 2019 15:05						Dial	501	NO ANSWER	00:03			
Tue, 21 May 2019 15:05						Dial	501	ANSWERED	00:52			
Tue, 21 May 2019 15:05						Dial	501	NO ANSWER	00:17			
Tue, 21 May 2019 15:05						Dial	83452399495	ANSWERED	02:19			
Tue, 21 May 2019 15:01						Dial	89125730001	ANSWERED	01:17			
Tue, 21 May 2019 15:01						Dial	+73432535885	ANSWERED	00:14			
Tue, 21 May 2019 15:00						BackGround	s [ivr-3]	ANSWERED	00:09			
Tue, 21 May 2019 14:57						Dial	73452434464	ANSWERED	01:08			
Tue, 21 May 2019 14:56						Dial	540	NO ANSWER	00:04			
Tue, 21 May 2019 14:56						Dial	540	NO ANSWER	00:04			
Tue, 21 May 2019 14:56						Dial	540	ANSWERED	02:54			
Дата звонка						Прил.	Куда звонил	Статус звонка	Длительность	Пользовательское поле	Учетка	
Tue, 21 May 2019 14:56						Dial	540	NO ANSWER	00:04			
Tue, 21 May 2019 14:56						Dial	540	NO ANSWER	00:18			
Tue, 21 May 2019 14:56						Dial	89823124513	ANSWERED	02:25			
Tue, 21 May 2019 14:55						Dial	89048032844	ANSWERED	02:37			
Tue, 21 May 2019 14:55						Dial	89222642602	ANSWERED	01:53			
Tue, 21 May 2019 14:55						Dial	1	BUSY	00:02			
Tue, 21 May 2019 14:55						Dial	1	ANSWERED	00:35			
Tue, 21 May 2019 14:53						Dial	89058208826	BUSY	00:33			

Рис. 20. Сформированный отчет со списком звонков

По нажатию на идентификатор можно получить еще более детальное описание списка звонков (см. рис. 14, личные данные скрыты).

Отчёты CDR

Статус звонка: Любой статус Не:

Сначала более ранние Группировать по: День Искать

Журнал вызовов - Результат поиска 22 События

Время	Событие	CNAM	CNUM	АНИ	Входящий номер	АМА	расш	контекст	Прил.	канал	Польз. тип	Экстрасобытие
Sun, 12 May 2019 16:43	CHAN_START					DEFAULT	9634523810	from-trunk		SIP/Beeline-9068883432-00034bfe		
Sun, 12 May 2019 16:43	APP_START					DEFAULT	recordcheck	sub-record-check	MixMonitor	SIP/Beeline-9068883432-00034bfe		
Sun, 12 May 2019 16:43	APP_END					DEFAULT	recordcheck	sub-record-check	MixMonitor	SIP/Beeline-9068883432-00034bfe		
Sun, 12 May 2019 16:43	ANSWER	Tmn_-79995477469				DEFAULT	s	ivr-10	Answer	SIP/Beeline-9068883432-00034bfe		
Sun, 12 May 2019 16:43	CHAN_START	221				DEFAULT	s	from-internal		SIP/221-00034bff		
Sun, 12 May 2019 16:43	CHAN_START	222				DEFAULT	s	from-internal		SIP/222-00034c00		
Sun, 12 May 2019 16:43	CHAN_START	224				DEFAULT	s	from-internal		SIP/224-00034c01		
Sun, 12 May 2019 16:43	CHAN_START	225				DEFAULT	s	from-internal		SIP/225-00034c02		
Sun, 12 May 2019 16:43	CHAN_START	226				DEFAULT	s	from-internal		SIP/226-00034c03		
Sun, 12 May 2019 16:43	HANGUP	221				DEFAULT	501	from-internal	AppDial	SIP/221-00034bff		
Sun, 12 May 2019 16:43	CHAN_END	221				DEFAULT	501	from-internal	AppDial	SIP/221-00034bff		
Sun, 12 May 2019 16:43	HANGUP	222				DEFAULT	501	from-internal	AppDial	SIP/222-00034c00		
Sun, 12 May 2019 16:43	CHAN_END	222				DEFAULT	501	from-internal	AppDial	SIP/222-00034c00		
Sun, 12 May 2019 16:43	HANGUP	224				DEFAULT	501	from-internal	AppDial	SIP/224-00034c01		
Sun, 12 May 2019 16:43	CHAN_END	224				DEFAULT	501	from-internal	AppDial	SIP/224-00034c01		
Sun, 12 May 2019 16:43	HANGUP	225				DEFAULT	501	from-internal	AppDial	SIP/225-00034c02		
Sun, 12 May 2019 16:43	CHAN_END	225				DEFAULT	501	from-internal	AppDial	SIP/225-00034c02		
Sun, 12 May 2019 16:43	HANGUP	226				DEFAULT	501	from-internal	AppDial	SIP/226-00034c03		
Sun, 12 May 2019 16:43	CHAN_END	226				DEFAULT	501	from-internal	AppDial	SIP/226-00034c03		
Sun, 12 May 2019 16:43	HANGUP	Tmn_-79995477469				DEFAULT	h	ext-group		SIP/Beeline-9068883432-00034bfe		
Sun, 12 May 2019 16:43	CHAN_END	Tmn_-79995477469				DEFAULT	h	ext-group		SIP/Beeline-9068883432-00034bfe		
Sun, 12 May 2019 16:43	LINKEDID_END	Tmn_-79995477469				DEFAULT	h	ext-group		SIP/Beeline-9068883432-00034bfe		

Детализация связанных звонков

Дата звонка	Запись	Идентификатор	Имя звонящего	Исходящий CallerID	Входящий номер	Прил.	Куда звонил	Статус звонка	Длительность	Пользовательское поле	Учетка
Sun, 12 May 2019 16:43		1557661421.222077				Dial	501	NO ANSWER	00:16		
Sun, 12 May 2019 16:43		1557661421.222077				Dial	501	NO ANSWER	00:02		
Sun, 12 May 2019 16:43		1557661421.222077				Dial	501	NO ANSWER	00:02		
Sun, 12 May 2019 16:43		1557661421.222077				Dial	501	NO ANSWER	00:02		
Sun, 12 May 2019 16:43		1557661421.222077				Dial	501	NO ANSWER	00:02		

FreePBX is a registered trademark of Sangoma Technologies Inc. FreePBX 14.0.11 is licensed under the GPL Copyright© 2007-2019

SANGOMA

Рис. 21. Список звонков по идентификатору

3.1.3. Реализация в конфигурации 1С: Управление сервисным центром

После успешной настройки и подключения к 1С-Рарус Софтфон можно приступить к настройкам Софтфона в конфигурации 1С: Управление сервисным центром.

В меню Софтфона в конфигурации 1С: Управление сервисным центром переходим в персональные настройки конфигурации (см. рис. 22). В разделе «Настройка действий при звонке» устанавливаем значение «Событие» в действиях при входящем и исходящем звонке. Это означает, что после принятия входящего или исходящего звонка будет открыт документ «Событие» с данными, полученными как из сервера с помощью Asterisk (например, поля CallerID, CallerName, PhoneNumber и т.д.), так и из справочника или документа в конфигурации в случае с исходящим звонком или регистра сведений по контрагентам и контактными лицам в случае с входящим звонком (см. рис. 23 и рис. 24 с показом звонка и созданного документа «Событие» соответственно).

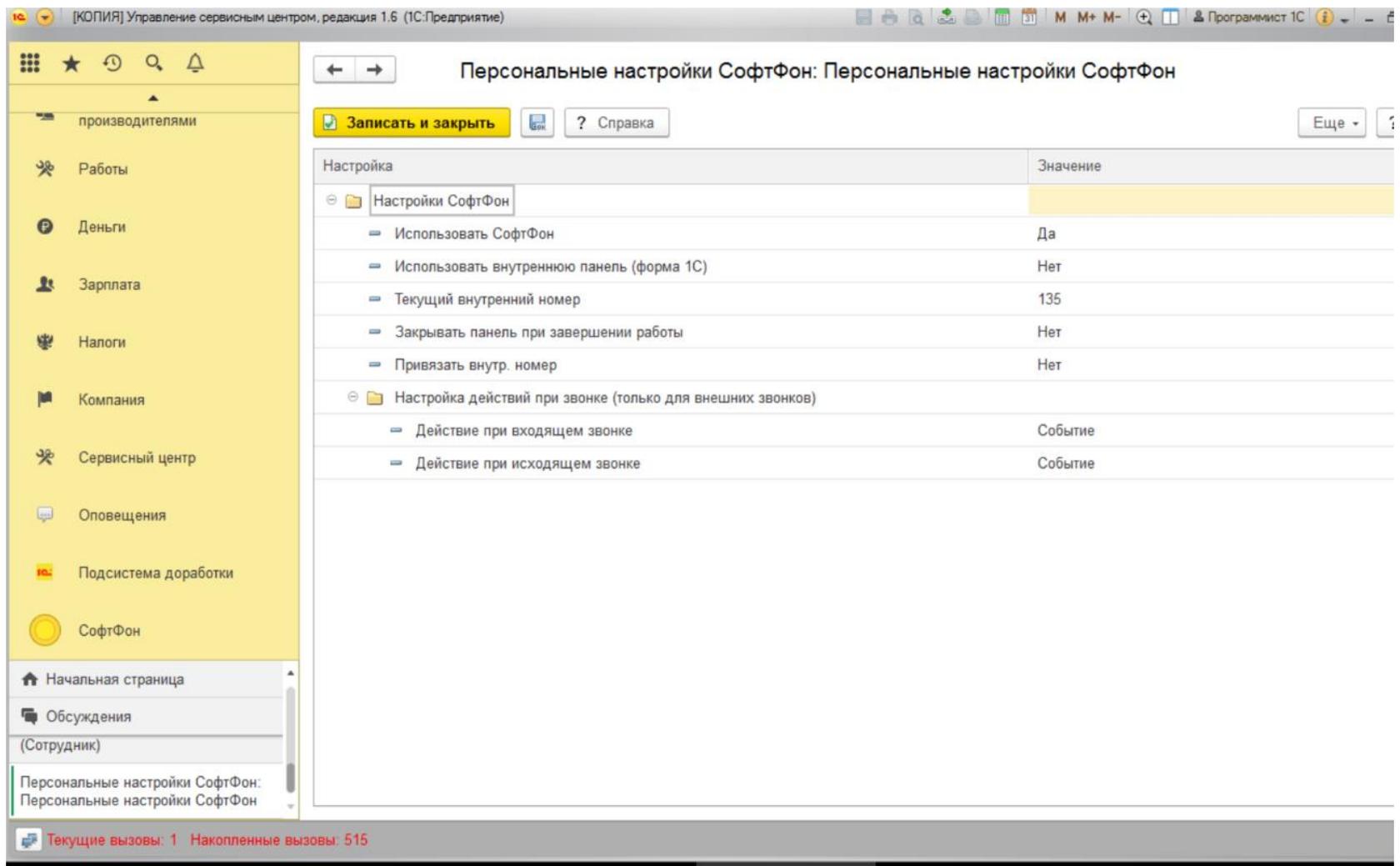


Рис. 22. Персональные настройки 1С:Софтфон в конфигурации «1С:Управление сервисным центром»

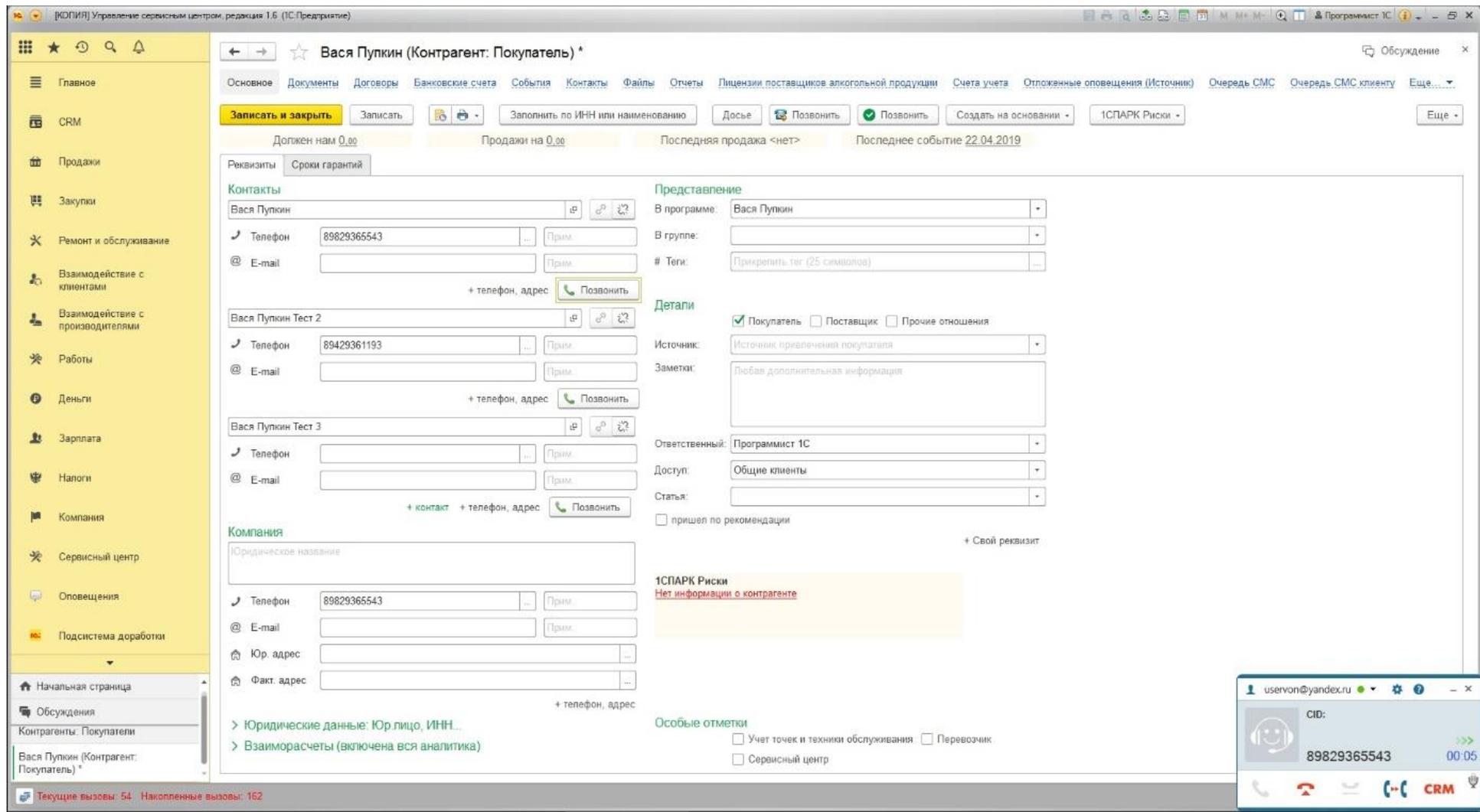


Рис. 23. Совершение звонка из элемента справочника «Контрагенты»

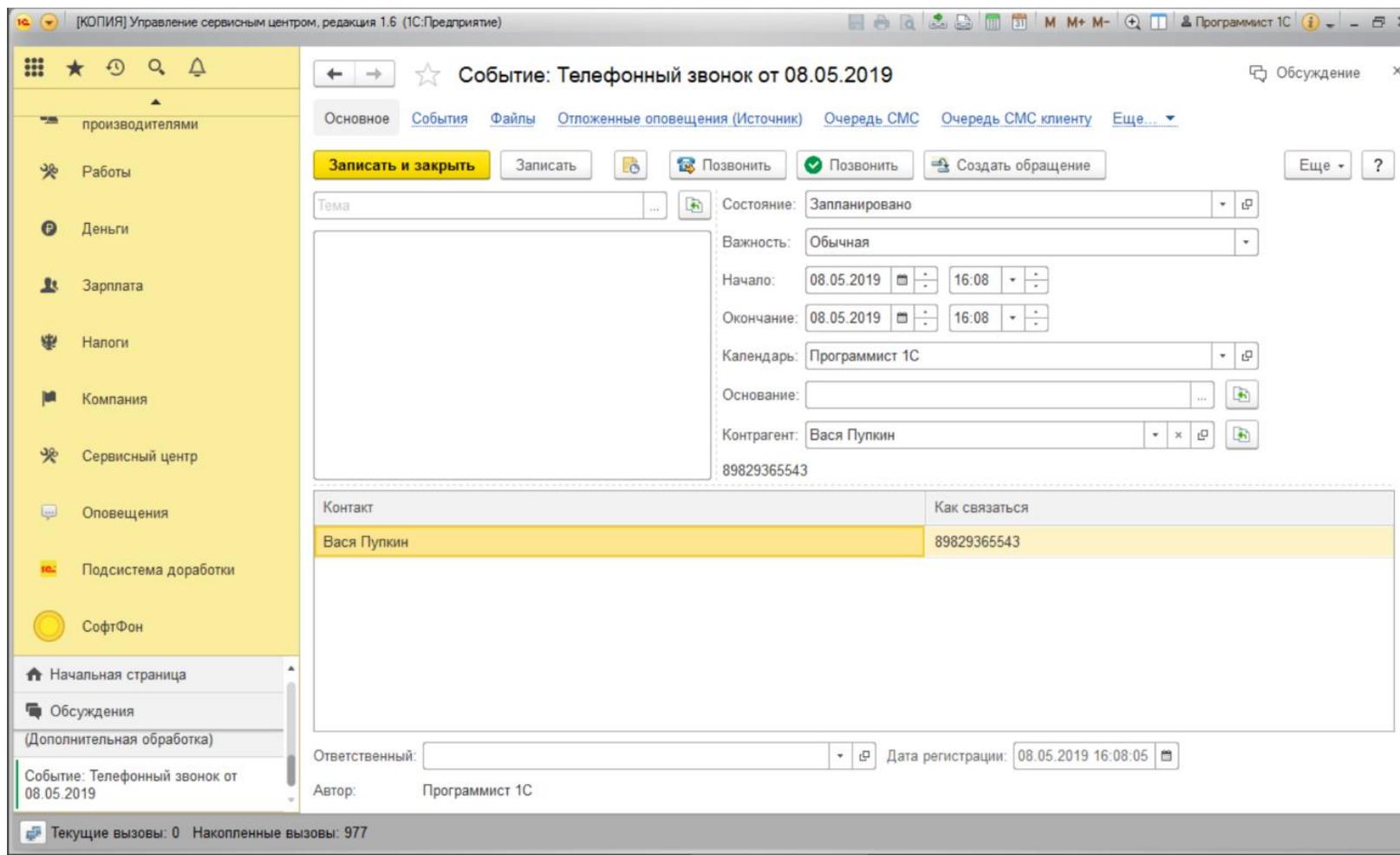


Рис. 24. Создание документа «Событие» с телефонным звонком

По контрагенту можно получить список документов «Событие» с зафиксированными в них звонками (см. рис. 25).

[КОПИЯ] Управление сервисным центром, редакция 1.6 (ТС:Предприятие)

Вася Пупкин (Контрагент: Покупатель)

Основное | Документы | Договоры | Банковские счета | **События** | Контакты | Файлы | Отчеты | Еще...

События

Создать | Позвонить | Позвонить | Поиск (Ctrl+F) | Еще

Период: за все время

Входящие | Исходящие | **Все**

Ответственный

Событие

Состояние

Важность

Учетная запись

Контактная информация

Телефон: 89829365543

Вася Пупкин

Телефон: 89829365543

Вася Пупкин Тестовый

Телефон: 89429361193

[Настроить учетные записи](#)

Событие	Начало	↓	Тема	Состояние	Важность
	Окончание				
☎ Телефонный звонок	22.04.19 13:36			Запланировано	Обычная
	22.04.19 14:04		Вася Пупкин <89829365543>; Вася Пупкин <89829365543>		
☎ Телефонный звонок	07.05.19 13:55			Запланировано	Обычная
	07.05.19 13:55		Вася Пупкин <89829365543>; Вася Пупкин <89829365543>		
☎ Телефонный звонок	07.05.19 13:55			Запланировано	Обычная
	07.05.19 13:55		Вася Пупкин <89829365543>; Вася Пупкин <89829365543>		
☎ Телефонный звонок	07.05.19 15:00			Запланировано	Обычная
	07.05.19 15:00		Вася Пупкин <89829365543>; Вася Пупкин <89829365543>		
☎ Телефонный звонок	07.05.19 16:02			Запланировано	Обычная
	07.05.19 16:02		Вася Пупкин <89829365543>; Вася Пупкин <89829365543>		
☎ Телефонный звонок	08.05.19 16:08			Запланировано	Обычная
	08.05.19 16:08		Вася Пупкин <89829365543>; Вася Пупкин <89829365543>		
☎ Телефонный звонок	08.05.19 16:08			Запланировано	Обычная
	08.05.19 16:08		Вася Пупкин <89829365543>; Вася Пупкин <89829365543>		

Текущие вызовы: 0 | Накопленные вызовы: 979

свернуть

Рис. 25. Список документов «Событие» с звонками по контрагенту

3.2. Реализация архитектуры процесса публикации заказ-нарядов и назначения мастеров

Для корректной периодической работы вышеописанной обработки необходимо в подразделе «Дополнительные отчеты и обработки» раздела «Администрирование» необходимо задать корректное расписание запуска обработки. В результате проверок на примере копии рабочей базы в качестве источника было принято решение запускать команду «Назначение исполнителей на заказы» каждые 10 минут, а также запускать команду «Рассылка заказ-нарядов мастерам» каждую минуту (см. рис. 26).

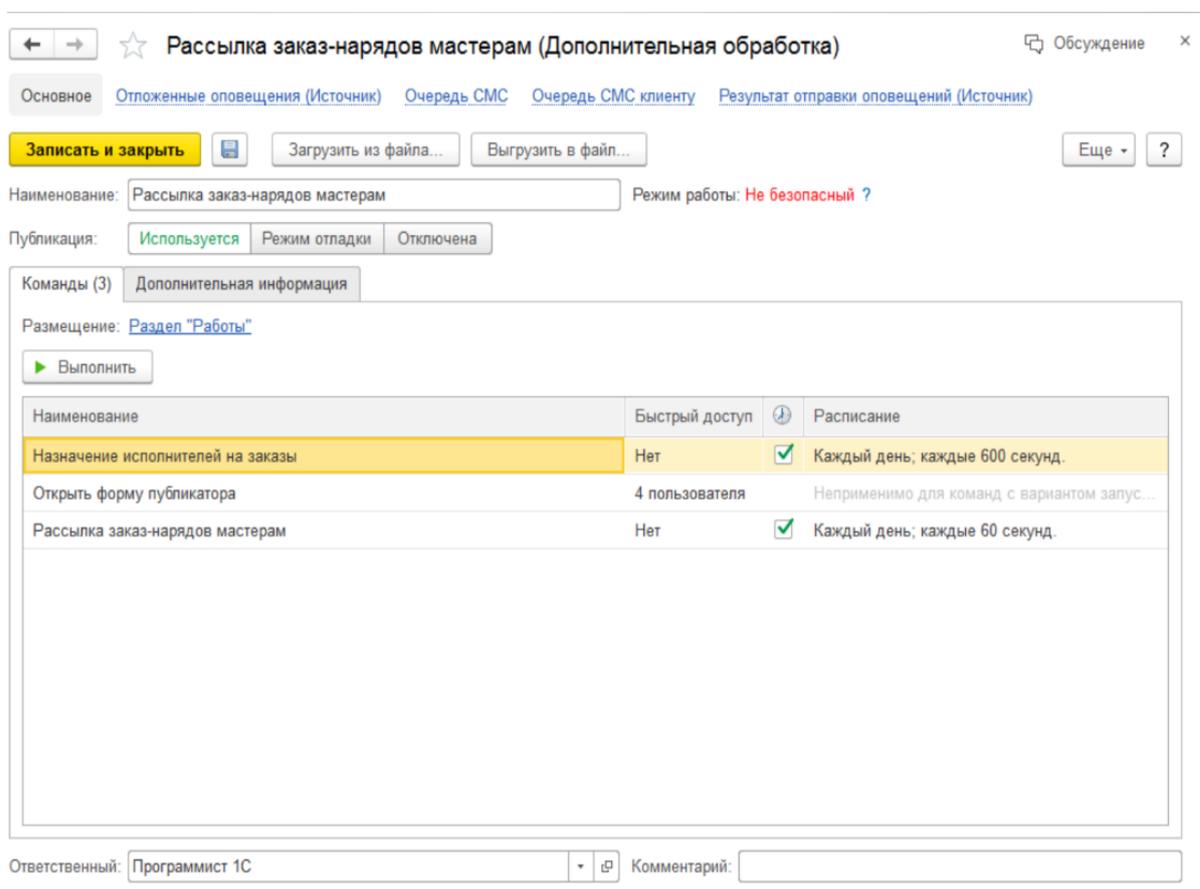


Рис. 26. Настройки расписания вызова команд обработки «Рассылка заказ-нарядов мастерам»

Более подробно изменения каждого из вышеуказанных модулей описаны в листинге в приложении Б.

3.3. Аналитика

3.3.1. Аналитика по публикации и назначению сотрудника на заказ-наряд

Для того, чтобы оценить, насколько эффективнее стала работа обновленного бизнес-процесса публикации и назначения сотрудника, было принято решение провести тестирование на свежей копии рабочей базы с небольшими изменениями в процессе публикации сотрудника, а именно отключение Android-оповещений мастеру для исключения случаев ложного назначения сотрудника на заказ-наряд на период тестирования. Суть тестирования заключается в следующем: в начальной версии бизнес-процесса производится замер секундомером времени ручного назначения сотрудника на 10 последних непроведенных заказ-нарядах, а в обновленной версии - замер времени работы обработки с теми же 10 последними непроведенными заказ-нарядами. Результаты сравнения показаны в табл. 19.

Таблица 19. Сравнение времени поиска и назначения сотрудников на 10 заказ-нарядов вручную и с помощью обработки.

Порядков ый № заказ- наряда	Город	Ручной вариант, сек	Автоматич еский вариант, сек	Относительная разность во времени
1	Уфа	35,5	26,075	27%
2	Краснодар	204,276	87,232	57%
3	Пермь	115,62	83,566	28%
4	Тюмень	326,308	163,542	50%
5	Екатеринбург	260,68	129,2	50%
6	Ростов-На- Дону	100,035	60,645	39%
7	Тюмень	167,787	89,095	47%
8	Ростов-На- Дону	25,42	17,36	32%
9	Уфа	83,52	65,94	21%
10	Челябинск	37,17	23,7	36%
Итог		1356,316	746,355	45%

Если рассмотреть таблицу сравнения времени, то можно заметить, что в одних заказ-нарядах прирост не превышает 30%, а в других превышает 40%. Это обусловлено тем, что в первом случае в городах (например, Уфа и Пермь), в котором происходит поиск мастера, не так много мастеров, в то время как в другом случае в городах (например, Тюмень и Краснодар) мастеров довольно много.

В результате сравнения было обнаружено, что время поиска и назначения мастера на заказ-наряд в общем было уменьшено на 45%.

При этом нужно учитывать, что время ручного процесса поиска в данном эксперименте взято без учета перехода между заказ-нарядами, т.к. диспетчер работает не только над поиском и назначением сотрудников, в то время как версия с обработкой работает постоянно в фоне с периодичностью раз в 10 минут для публикации и раз в минуту для назначения сотрудников. Если брать в учет переход между заказ-нарядами, то время будет нелинейно увеличиваться ввиду человеческого фактора.

3.3.2. Аналитика по интеграции телефонии в конфигурацию «1С:Управление сервисным центром»

Для оценки эффективности внедрения конфигурации телефонии «1С:Рарус-Софтфон» в основную конфигурацию «1С:Управление сервисным центром» необходимо провести предварительную проверку тестировщиками на копии рабочей базы. После успешной проверки работоспособности и отказоустойчивости конфигурации можно проводить тестирование на рабочей базе с несколькими одновременно работающими пользователями с последующим ведением статистики звонков за рабочий день для оценки активности.

В результате тестирования и собранной за 5 рабочих дней статистики было обнаружено следующее (см. табл. 20):

Таблица 20. Оценка загруженности линий телефонии после внедрения конфигурации телефонии в основную.

Время работы (по МСК)	Одновременное кол-во пользователей в конфигурации	Кол-во звонков за период	Кол-во занятых линий	% загруженности линий
09:00-10:00	17	13	10	59%
10:00-12:00	30	49	20	67%
12:00-14:00	29	68	28	97%
14:00-16:00	26	54	23	88%
16:00-18:00	26	41	20	77%
Итого	25,6	225	20,2	77%

Исходя из вышеописанной загруженности линий за рабочий день, можно сделать вывод, что, несмотря на оптимальную среднюю загруженность линий (ниже 80%), рекомендуется в период времени с 12:00 по 16:00 активировать дополнительные учетные записи телефонии для разгрузки линий телефонии с целью уменьшения задержек во время разговора.

Если рассматривать активность фиксируемых звонков в конфигурации, то в результате опытного сравнения замечен внушительный рост кол-ва документов с телефонными звонками (40-50 звонков до внедрения против 180-200 звонков после внедрения) и их детализация.

3.3.3. Апробация

В результате аналитики и реализации вышеописанных решений были проведены следующие действия:

1. проверка работоспособности и отказоустойчивости на копии рабочей базы тестировщиками;
2. опытная эксплуатация 30 пользователями одновременно на рабочей базе.

В качестве подтверждения к отчёту прилагается акт о внедрении.

Заключение

В результате работы были реализованы следующие задачи:

- выполнена интеграция конфигурации «1С: Управление сервисным центром» с конфигурацией «1С: Парус-Софтфон» для фиксации звонков по заказу с детализацией и для звонков контрагентам;
- автоматизирован процесс публикации заказ-нарядов и назначения на них сотрудников с последующим оповещением в программном приложении «РемТех» на Android;
- разработан и создан табель рабочего времени за месяц для более детального учета рабочего времени сотрудника;
- была проведена аналитика по вышеописанным решениям с последующей их апробацией.

За счет реализованных механизмов интеграции и реализации различных структур и обработок были перестроены бизнес-процессы телефонии и назначения заказ-нарядов, что позволило диспетчерам сократить время на исполнения своих бизнес-процессов и исключить фактор ручной ошибки.

Список литературы

1. Панель телефонии в «МИКО: Телефония»
URL: <https://telefon.miko.ru/products/panel-telefonii/>
2. Описание работы X-Lite
URL: <https://www.counterpath.com/x-lite/>
3. 1С:Enterprise Development Tools: руководство разработчика
URL: <https://its.1c.ru/db/edt16doc>
4. Описание методологии BPMN
URL: <https://www.omg.org/spec/BPMN/2.0/About-BPMN/>
5. Asterisk - архитектура, протоколы, кодеки
URL: <http://omoled.ru/publications/view/28>
6. FreePBX - User Control Panel (UCP)
URL: <https://asterisk-pbx.ru/wiki/freepbx/freepbx>
7. Использование замера производительности для оптимизации приложений в 1С:Предприятии 8
URL: <https://its.1c.ru/db/metod8dev#content:1553:hdoc>
8. 1С-Рарус: Интеграция с телефонией (СофтФон) – руководство разработчика
URL: <https://telefon.1crm.ru/help/3300/>

Приложение А. Интеграция 1С:Рарус-Софтфон с 1С: Управление сервисным центром

```
////////////////////////////////////
// СЛУЖЕБНЫЕ ПРОЦЕДУРЫ И ФУНКЦИИ
// Данная функция возвращает имя абонента, если он есть в регистре сведений контактных лиц.
Функция сфпПолучитьКонтактноеЛицоПоНомеруТелефона(НомерТелефона) Экспорт
    Запрос = Новый Запрос;
    Запрос.Текст = «ВЫБРАТЬ
        СправСсылка,                |   д_КонтрагентыИКонтактныеЛица.СправСсылка                КАК
        НомерТелефона              |   д_КонтрагентыИКонтактныеЛица.НомерТелефона            КАК
                                   |ИЗ
        д_КонтрагентыИКонтактныеЛица |   РегистрСведений.д_КонтрагентыИКонтактныеЛица        КАК
                                   |ГДЕ
                                   |   д_КонтрагентыИКонтактныеЛица.НомерТелефона        =
        &НомерТелефона»;
    Запрос.УстановитьПараметр(«НомерТелефона», НомерТелефона);
    Выборка = Запрос.Выполнить().Выгрузить();
    Если Выборка.Количество() = 0 Тогда
        Возврат Справочники.КонтактныеЛица.ПустаяСсылка();
    Иначе
        Возврат Выборка[0].СправСсылка;
    КонецЕсли;
КонецФункции
////////////////////////////////////
// ПРОЦЕДУРЫ И ФУНКЦИИ РАБОТЫ С НОМЕРОМ ТЕЛЕФОНА

// Функция возвращает номер телефона в котором убраны все символы, кроме цифр
//
// Параметры:
//     НомерТелефона - Строка      - Номер телефона
//
// Возвращаемое значение:
//     Строка - Очищенный номер телефона
//
Функция сфпУбратьИзНомераТелефонаВсеПрефиксы(НомерТелефона) Экспорт
```

```

ПозицияРазделителя = Найти(НомерТелефона, «@»);
Если ПозицияРазделителя > 0 Тогда
    НомерТелефона = Лев(НомерТелефона, ПозицияРазделителя - 1);
КонецЕсли;

СтрокаЦифр = «0123456789»;

ОчищенныйНомер = «»;
ДлинаНомера = СтрДлина(НомерТелефона);
Для НомерСимвола = 1 По ДлинаНомера Цикл
    ТекущийСимвол = Сред(НомерТелефона, НомерСимвола, 1);
    Если Найти(СтрокаЦифр, ТекущийСимвол) > 0 Тогда
        ОчищенныйНомер = ОчищенныйНомер + ТекущийСимвол;
    КонецЕсли;
КонецЦикла;

// Убираем из начала полученного внутреннего номера лидирующие нули, если такие есть
сфПараметрыСервера = сфСофтФонПроСервер_.сфПараметрыСервера();
МаксимальнаяДлинаВнутреннихНомеров =
сфПараметрыСервера.МаксимальнаяДлинаВнутреннихНомеров;
Если Лев(ОчищенныйНомер, 3) = «000» И СтрДлина(ОчищенныйНомер) <=
МаксимальнаяДлинаВнутреннихНомеров + 3 И СтрДлина(ОчищенныйНомер) < 7 Тогда
    ОчищенныйНомер = Сред(ОчищенныйНомер, 4);

ИначеЕсли Лев(ОчищенныйНомер, 2) = «00» И СтрДлина(ОчищенныйНомер) <=
МаксимальнаяДлинаВнутреннихНомеров + 2 И СтрДлина(ОчищенныйНомер) < 7 Тогда
    ОчищенныйНомер = Сред(ОчищенныйНомер, 3);

ИначеЕсли Лев(ОчищенныйНомер, 1) = «0» И СтрДлина(ОчищенныйНомер) <=
МаксимальнаяДлинаВнутреннихНомеров + 1 И СтрДлина(ОчищенныйНомер) < 7 Тогда
    ОчищенныйНомер = Сред(ОчищенныйНомер, 2);
КонецЕсли;

Возврат ОчищенныйНомер;

КонецФункции // сфУбратьИзНомераТелефонаВсеПрефиксы()
////////////////////
// ЗАПИСЬ ТЕЛЕФОННЫХ ПЕРЕГОВОРОВ

```

```

// Функция проверяет использование записи переговоров
//
// Параметры:
//     Нет.
//
// Возвращаемое значение:
//     Булево - Признак использования СофтФона
//
Функция сфИспользоватьЗаписьПереговоров() Экспорт
    Возврат Константы.сфИспользоватьЗаписьПереговоров.Получить();
КонецФункции // ИспользоватьЗаписьПереговоров(

// Функция проверяет использование Спрут7
//
// Параметры:
//     Нет.
//
// Возвращаемое значение:
//     Булево - Признак использования СофтФона
//
Функция сфИспользоватьСпрут7() Экспорт
    Возврат Константы.сфИспользоватьСпрут7.Получить();
КонецФункции // ИспользоватьСпрут7(

// Функция проверяет использование CLON
//
// Параметры:
//     Нет.
//
// Возвращаемое значение:
//     Булево - Признак использования СофтФона
//
Функция сфИспользоватьCLON() Экспорт
    Возврат Константы.сфИспользоватьCLON.Получить();
КонецФункции // ИспользоватьCLON(

```

```

// Функция возвращает массив пользователей, телефонные переговоры которых могут
прослушиваться переданным пользователем

//
// Параметры:
// Пользователь - СправочникСсылка - Пользователь, который контролирует телефонные
переговоры

//
// Возвращаемое значение:
// Массив - Массив прослушиваемых пользователей

//
Функция сфПолучитьМассивПрослушиваемыхПользователей(Пользователь) Экспорт
    МассивПользователей = Новый Массив;

    Если НЕ ЗначениеЗаполнено(Пользователь) Тогда
        Возврат МассивПользователей;

    КонецЕсли;

    Запрос = Новый Запрос;
    Запрос.УстановитьПараметр(«Пользователь», Пользователь);
    СписокПользователейГрупп = Новый СписокЗначений;
    ИспользоватьГруппыПользователей = сфИспользоватьГруппыПользователей();
    Если ИспользоватьГруппыПользователей Тогда
        Запрос.Текст = «ВЫБРАТЬ РАЗРЕШЕННЫЕ РАЗЛИЧНЫЕ
            |
            | сфДоступныеДляПрослушиванияПользователи.КонтролируемыйПользователь КАК Группа
            | ИЗ
            |
            | РегистрСведений.сфДоступныеДляПрослушиванияПользователи КАК
сфДоступныеДляПрослушиванияПользователи
            | ГДЕ
            |
            | сфДоступныеДляПрослушиванияПользователи.Пользователь =
&Пользователь
            |
            | И сфДоступныеДляПрослушиванияПользователи.ЭтоГруппа»;
        СписокГрупп = Новый СписокЗначений;

        СписокГрупп.ЗагрузитьЗначения(Запрос.Выполнить().Выгрузить().ВыгрузитьКолонку(«Групп
а»));

        Если
СписокГрупп.НайтиПоЗначению(Справочники.ГруппыПользователей.ВсеПользователи) = Неопределено
Тогда
            Запрос.УстановитьПараметр(«СписокГрупп», СписокГрупп);
            Запрос.Текст = «ВЫБРАТЬ РАЗРЕШЕННЫЕ РАЗЛИЧНЫЕ
                |
                | ГруппыПользователейСостав.Пользователь КАК
Пользователь

```

```

        | ИЗ
        | Справочник.ГруппыПользователей.Состав КАК
ГруппыПользователейСостав
        | ГДЕ
        | ГруппыПользователейСостав.Ссылка В
ИЕРАРХИИ(&СписокГрупп));

```

```

СписокПользователейГрупп.ЗагрузитьЗначения(Запрос.Выполнить()).Выгрузить().ВыгрузитьК
олонку(«Пользователь»));

```

```

Иначе

```

```

    Запрос.Текст = «ВЫБРАТЬ РАЗРЕШЕННЫЕ РАЗЛИЧНЫЕ

```

```

        | Пользователи.Ссылка КАК Пользователь

```

```

        | ИЗ

```

```

        | Справочник.Пользователи КАК Пользователи»;

```

```

    Выборка = Запрос.Выполнить().Выбрать();

```

```

    Пока Выборка.Следующий() Цикл

```

```

        МассивПользователей.Добавить(Выборка.Пользователь);

```

```

    КонецЦикла;

```

```

    Если МассивПользователей.Найти(Пользователь) = Неопределено Тогда

```

```

        МассивПользователей.Добавить(Пользователь);

```

```

    КонецЕсли;

```

```

    Возврат МассивПользователей;

```

```

КонецЕсли;

```

```

КонецЕсли;

```

```

Запрос.Текст = «ВЫБРАТЬ РАЗРЕШЕННЫЕ РАЗЛИЧНЫЕ

```

```

    |
сфДоступныеДляПрослушиванияПользователи.КонтролируемыйПользователь КАК Пользователь

```

```

    | ИЗ

```

```

    | РегистрСведений.сфДоступныеДляПрослушиванияПользователи КАК
сфДоступныеДляПрослушиванияПользователи

```

```

    | ГДЕ

```

```

    | сфДоступныеДляПрослушиванияПользователи.Пользователь =
&Пользователь

```

```

    | И НЕ сфДоступныеДляПрослушиванияПользователи.ЭтоГруппа»;

```

```

    Выборка = Запрос.Выполнить().Выбрать();

```

```

    Пока Выборка.Следующий() Цикл

```

```

        МассивПользователей.Добавить(Выборка.Пользователь);

```

```

    КонецЦикла;

```

```

    Если ИспользоватьГруппыПользователей Тогда

```

```

        Для Каждого ПользовательГруппы Из СписокПользователейГрупп Цикл

```

```

        Если МассивПользователей.Найти(ПользовательГруппы.Значение) =
Неопределено Тогда
            МассивПользователей.Добавить(ПользовательГруппы.Значение);
        КонецЕсли;
    КонецЦикла;
КонецЕсли;
Если МассивПользователей.Найти(Пользователь) = Неопределено Тогда
    МассивПользователей.Добавить(Пользователь);
КонецЕсли;
Возврат МассивПользователей;
КонецФункции // сфпПолучитьМассивПрослушиваемыхПользователей()

```

```

////////////////////////////////////

```

```

// МАСТЕР НАСТРОЙКИ IP-ATC AGAT UX

```

```

// Функция возвращает массив пользователей, для использования в мастере настроек

```

```

//

```

```

// Параметры:

```

```

// Нет.

```

```

//

```

```

// Возвращаемое значение:

```

```

// Массив - Массив пользователей

```

```

//

```

```

Функция сфпПолучитьМассивПользователей() Экспорт

```

```

Запрос = Новый Запрос;

```

```

    Запрос.Текст = «ВЫБРАТЬ РАЗРЕШЕННЫЕ

```

```

        | Пользователи.Наименование КАК Наименование

```

```

        | ИЗ

```

```

        | Справочник.Пользователи КАК Пользователи

```

```

        |

```

```

        | УПОРЯДОЧИТЬ ПО

```

```

        | Наименование»;

```

```

    Возврат Запрос.Выполнить().Выгрузить().ВыгрузитьКолонку(«Наименование»);

```

```

КонецФункции // сфпПолучитьМассивПользователей()

```

```

сфпПроКлиент:

```

```

// Функция разбирает переданную структуру номера телефона и возвращает строку, которую
// будет набирать АТС
//
// Параметры:
// СтруктураНомера - Структура - Структура телефонного номера
//
// Возвращаемое значение:
// Строка - Строка для набора АТС
//
Функция сфпПреобразоватьНомерСУчетомПрефиксов(СтруктураНомера)
    сфпПараметрыСервера = сфпСофтФонПроСервер_.сфпПараметрыСервера();
    Если ПустаяСтрока(СтруктураНомера.НомерТелефона) Тогда
        Если ПустаяСтрока(СтруктураНомера.Добавочный) Тогда
            // Номера нет.
            РезультатНомер = «»;
        Иначе
            // Внутренний номер.
            РезультатНомер = СтруктураНомера.Добавочный;
        КонецЕсли;
        // проверяем длину номеру вместе с длиной кода города, т.к. в ряде компаний
        // внутренний
        // номер может быть достаточно длинным, до 9 символов
        // у внутренних же номеров код города отсутствует
        ИначеЕсли
            СтрДлина(СтруктураНомера.НомерТелефона)+СтрДлина(СтруктураНомера.КодГорода) >
            сфпПараметрыСервера.МаксимальнаяДлинаВнутреннихНомеров Тогда
                КодСтраныНомера = СокрЛП(СтрЗаменить(СтруктураНомера.КодСтраны, «+»,
                ««));
                КодСтраныНомера = ?(КодСтраныНомера = «8», «7», КодСтраныНомера);
                КодСтраныПользователя = СокрЛП(СтрЗаменить(сфпПараметрыСервера.КодСтраны,
                «+», ««));
                КодСтраныПользователя = ?(КодСтраныПользователя = «8», «7»,
                КодСтраныПользователя);
                Если ПустаяСтрока(КодСтраныНомера) Тогда
                    // Номер нашей страны.
                    РезультатНомер = СтруктураНомера.КодГорода +
                    СтруктураНомера.НомерТелефона;
                ИначеЕсли ПустаяСтрока(КодСтраныПользователя) Тогда
                    // Номер нашей страны.
                    РезультатНомер = СтруктураНомера.КодГорода +
                    СтруктураНомера.НомерТелефона;
                ИначеЕсли (КодСтраныНомера = КодСтраныПользователя) Тогда
                    // Номер нашей страны.

```

```

        РезультатНомер = СтруктураНомера.КодГорода +
СтруктураНомера.НомерТелефона;
        Иначе
            // Номер другой страны.
            РезультатНомер = СтруктураНомера.КодСтраны + СтруктураНомера.КодГорода
+ СтруктураНомера.НомерТелефона;
        КонецЕсли;
    Иначе
        // Внутренний номер.
        РезультатНомер = СтруктураНомера.НомерТелефона;
    КонецЕсли;
    Возврат РезультатНомер
КонецФункции // ПреобразоватьНомерСУчетомПрефиксов()

// Процедура выполняет звонок через панель СофтФон
//
// Параметры:
//     ДополнительныеПараметры - Структура - Структура дополнительных параметров
//
Процедура сфпВыполнитьЗвонок(ДополнительныеПараметры)
    Если ПустаяСтрока(ДополнительныеПараметры.Телефон) Тогда Возврат; КонецЕсли;
    ContactID = «»;
    ContactName = «»;
    ContactDopInfo = «»;
    ContactImage = «»;
    Если ЗначениеЗаполнено(ДополнительныеПараметры.СсылкаНаОбъект) Тогда
        ContactName =
сфпСофтФонПроСервер_.сфпПолучитьНаименованиеКонтакта(ДополнительныеПараметры.СсылкаНаОбъект
);
        Если НЕ (ТипЗнч(ДополнительныеПараметры.СсылкаНаОбъект) = Тип(«Строка»))
Тогда
            ContactID =
сфпСофтФонПроСервер_.сфпПолучитьИдентификаторКонтакта(ДополнительныеПараметры.СсылкаНаОбъект
т);
            ВладелецКонтакта =
сфпСофтФонПроСервер_.сфпПолучитьВладельцаКонтакта(ДополнительныеПараметры.СсылкаНаОбъект);
            Если ЗначениеЗаполнено(ВладелецКонтакта) Тогда
                ContactDopInfo =
сфпСофтФонПроСервер_.сфпПолучитьНаименованиеКонтакта(ВладелецКонтакта);
            Иначе
                ContactDopInfo = «»;
            КонецЕсли;
        КонецЕсли;
    КонецЕсли;

```

```

                ContactImage =
сфСофтФонПроСервер_.сфПолучитьАватарКонтакта(ДополнительныеПараметры.СсылкаНаОбъект,
Истина);

                КонецЕсли;

            КонецЕсли;

            Если ДополнительныеПараметры.Свойство(«МеждународныйЗвонок») и
ДополнительныеПараметры.МеждународныйЗвонок Тогда

                ДополнительныеПараметры.Телефон =
СтрЗаменить(ДополнительныеПараметры.Телефон, «+», ««);

                КонецЕсли;

            Попытка

                Результат = сфПанельУправления.MakeCall(««, ДополнительныеПараметры.Телефон,
ContactID, ContactName, ContactDopInfo, ContactImage);

                сфДанныеЗаполнения = ДополнительныеПараметры.ДанныеЗаполнения;

            Исключение

                сфДанныеЗаполнения = Неопределено;

                Состояние(НСтр(«ru=Не удалось создать звонок'«));

            КонецПопытки;

КонецПроцедуры // сфВыполнитьЗвонок()

// Процедура - обработчик ответа на вопрос о совершении международного звонка
//
// Параметры:
//     ОтветНаВопрос           - КодВозвратаДиалога - Ответ на вопрос
//     ДополнительныеПараметры - Структура           - Структура
дополнительных параметров
//
Процедура сфВыполнитьМеждународныйЗвонок(ОтветНаВопрос, ДополнительныеПараметры) Экспорт
    Если ОтветНаВопрос = КодВозвратаДиалога.Да Тогда
        ДополнительныеПараметры.Вставить(«МеждународныйЗвонок», Истина);
        сфВыполнитьЗвонок(ДополнительныеПараметры);
    КонецЕсли;

КонецПроцедуры // ВыполнитьМеждународныйЗвонок()

// Процедура выполняет звонок по номеру телефона
//
// Параметры:
//     Телефон           - Строка           - Номер телефона
//     Объект           - СправочникСсылка - Контакт
//     ДанныеЗаполнения - Структура           - Данные для заполнения
создаваемого телефонного звонка
//

```

```

Процедура сфпПозвонить(Телефон, Объект = Неопределено, ДанныеЗаполнения = Неопределено)
Экспорт

    Если ПустаяСтрока(Телефон) Тогда
        ПоказатьПредупреждение(, НСтр(«ru = 'Не выбран номер телефона!'»), 5);

        Возврат;
    ИначеЕсли НЕ сфпПроверитьДоступностьСофтФон(Истина) Тогда
        Возврат;
    ИначеЕсли Телефон = сфпСофтФонПроСервер_.сфпТекущийВнутреннийНомер() Тогда
        Возврат;
    КонецЕсли;

    сфпЗаполнитьПрефиксыИНастройки();
    ContactID = «»;
    ContactName = «»;
    ContactDopInfo = «»;
    ContactImage = «»;
    СтруктураНомера =
сфпСофтФонПроСервер_.сфпЗаполнитьСтруктуруПолейПоПредставлениюТелефон(Телефон);
    Телефон = сфпПреобразоватьНомерСУчетомПрефиксов(СтруктураНомера);
    ДополнительныеПараметры = Новый Структура;
    ДополнительныеПараметры.Вставить(«Телефон», Телефон);
    ДополнительныеПараметры.Вставить(«СсылкаНаОбъект», Объект);
    ДополнительныеПараметры.Вставить(«ДанныеЗаполнения», ДанныеЗаполнения);
    Если СтрДлина(Телефон) > 10 Тогда
        ОповещениеОтвета = Новый
ОписаниеОповещения(«сфпВыполнитьМеждународныйЗвонок», сфпСофтФонПроКлиент_,
ДополнительныеПараметры);
        ТекстЗаголовка = НСтр(«ru='ВНИМАНИЕ!!! Выполняется звонок в другую страну'«);
        ТекстВопроса = НСтр(«ru='ВОЗМОЖНО!!! Телефонный номер набран неверно» +
Символы.ПС + «Подтвердите выполнение международного звонка'«);
        СписокКнопок = Новый СписокЗначений;
        СписокКнопок.Добавить(КодВозвратаДиалога.Да, НСтр(«ru='Набрать
номер'«));
        СписокКнопок.Добавить(КодВозвратаДиалога.Нет, НСтр(«ru='Отменить'«));
        ПоказатьВопрос(ОповещениеОтвета, ТекстВопроса, СписокКнопок, 30,
КодВозвратаДиалога.Нет, ТекстЗаголовка, КодВозвратаДиалога.Нет);
    Иначе
        сфпВыполнитьЗвонок(ДополнительныеПараметры);
    КонецЕсли;
КонецПроцедуры // сфпПозвонить()

```

Приложение Б. Публикации заказ-нарядов и назначения мастеров

Функция СведенияОВнешнейОбработке() Экспорт

```
Синоним = «Рассылка заказ-нарядов мастерам»;  
РегистрационныеДанные = Новый Структура();  
РегистрационныеДанные.Вставить(«Вид», «ДополнительнаяОбработка»);  
РегистрационныеДанные.Вставить(«Наименование», Синоним);  
РегистрационныеДанные.Вставить(«Версия», «1.0»);  
РегистрационныеДанные.Вставить(«БезопасныйРежим», Ложь);  
РегистрационныеДанные.Вставить(«Информация», «Обработка «+Синоним»);  
РегистрационныеДанные.Вставить(«ВерсияБСП»,»1.2.1.4»);  
ТаблицаКоманд = ПолучитьТаблицуКоманд();
```

```
// Добавим команду в таблицу
```

```
ДобавитьКоманду(ТаблицаКоманд, «Рассылка заказ-нарядов мастерам»,  
«СформироватьРекомендацию», «ВызовСерверногоМетода», Истина, );
```

```
ДобавитьКоманду(ТаблицаКоманд, «Назначение исполнителей на заказы»,  
«ВыполнитьНазначение», «ВызовСерверногоМетода», Истина, );
```

```
ДобавитьКоманду(ТаблицаКоманд, «Открыть форму публикатора», «ПоказатьСтатусРаботы»,  
«ОткрытиеФормы», Истина, );
```

```
// Сохраним таблицу команд в параметры регистрации обработки
```

```
РегистрационныеДанные.Вставить(«Команды», ТаблицаКоманд);
```

```
Возврат РегистрационныеДанные;
```

КонецФункции

Функция ПолучитьТаблицуКоманд()

```
// Создадим пустую таблицу команд и колонки в ней
```

```
Команды = Новый ТаблицаЗначений;
```

```
//как будет выглядеть описание печатной формы для пользователя
```

```
Команды.Колонки.Добавить(«Представление», Новый ОписаниеТипов(«Строка»));
```

```
// Имя нашего макета, что бы могли отличить вызванную команду в обработке печати
```

```
Команды.Колонки.Добавить(«Идентификатор», Новый ОписаниеТипов(«Строка»));
```

```
// Тут задается, как должна вызваться команда обработки
```

```
// Возможные варианты:
```

// - ОткрытиеФормы - в этом случае в колонке идентификатор должно быть указано имя формы, которое должна будет открыть система

// - ВызовКлиентскогоМетода - вызвать клиентскую экспортную процедуру из модуля формы обработки

// - ВызовСерверногоМетода - вызвать серверную экспортную процедуру из модуля объекта обработки

Команды.Колонки.Добавить(«Использование», Новый ОписаниеТипов(«Строка»));

// Следующий параметр указывает, необходимо ли показывать оповещение при начале и завершению работы обработки. Не имеет смысла при открытии формы

Команды.Колонки.Добавить(«ПоказыватьОповещение», Новый ОписаниеТипов(«Булево»));

// Для печатной формы должен содержать строку ПечатьMXL

Команды.Колонки.Добавить(«Модификатор», Новый ОписаниеТипов(«Строка»));

Возврат Команды;

КонецФункции

Процедура ДобавитьКоманду(ТаблицаКоманд, Представление, Идентификатор, Использование, ПоказыватьОповещение = Ложь, Модификатор)

// Добавляем команду в таблицу команд по переданному описанию.

// Параметры и их значения можно посмотреть в функции ПолучитьТаблицуКоманд

НоваяКоманда = ТаблицаКоманд.Добавить();

НоваяКоманда.Представление = Представление;

НоваяКоманда.Идентификатор = Идентификатор;

НоваяКоманда.Использование = Использование;

НоваяКоманда.ПоказыватьОповещение = ПоказыватьОповещение;

НоваяКоманда.Модификатор = Модификатор;

КонецПроцедуры

Процедура ВыполнитьКоманду(ИдентификаторКоманды, ПараметрыВыполненияКоманды) Экспорт

Если ИдентификаторКоманды=«СформироватьРекомендацию» тогда

 ФормироватьРекомендацию();

ИначеЕсли ИдентификаторКоманды = «ВыполнитьНазначение» Тогда

 ВыполнитьНазначение();

КонецЕсли;

КонецПроцедуры

//Подбирает сотрудников и публикует заказы по рекомендации

Процедура ФормироватьРекомендацию() Экспорт

ПубликацииЗаказов = НовыеЗаказыИПубликации();

Для Каждого Запись Из ПубликацииЗаказов Цикл

РангРассылки = Запись.Ранг;

МинимальныйУровень = Запись.Уровень;

Заказ = Запись.Заказ;

Если Ложь Тогда Заказ = Документы._ЗаказНарядСервисногоЦентра.ПустаяСсылка();
КонецЕсли;

//Цикл, чтобы если никого не подобрали, сразу переходили к следующему
рангу/уровню

//Пока РангРассылки < 7 И МинимальныйУровень < 4 Цикл

Пока РангРассылки < 10 И МинимальныйУровень < 4 Цикл

РангРассылки = РангРассылки + 1;

МинимальныйУровень = ?(МинимальныйУровень = 0,
Заказ.ОписаниеНеисправности.д_Сложность, МинимальныйУровень);

//доработка 22.05.2019

//Если РангРассылки = 7 Тогда

Если РангРассылки = 10 Тогда

РангРассылки = 1;

МинимальныйУровень = МинимальныйУровень + 1;

КонецЕсли;

Если МинимальныйУровень = 4 Тогда Продолжить; КонецЕсли;

Подобранные = ПодобратьСотрудниковКЗаказу(Заказ, РангРассылки,
МинимальныйУровень);

Если Подобранные.Количество() = 0 Тогда Продолжить; КонецЕсли;

```

ВыделенныеСотрудники = ВыделитьСотрудниковИзСтатистики(Подобранные);
Сотрудник = РекомендоватьСотрудника(Заказ, Подобранные);

Опубликованные = ПолучитьПубликации(Заказ);
Опубликованные = ОбщегоНазначения.ВыгрузитьКолонку(Опубликованные,
«Сотрудник»);
ОбщегоНазначенияКлиентСервер.ДополнитьМассив(ВыделенныеСотрудники,
Опубликованные, Истина);
ОпубликоватьЗаказ(Заказ, ВыделенныеСотрудники, РангРассылки,
МинимальныйУровень);
//Останавливаем цикл если у нас есть подходящие мастера
//Если мы сразу нашли масетра, цикл сработает как «Если Тогда» т.е.
один раз

Прервать;
КонецЦикла;

КонецЦикла;

КонецПроцедуры

//Подбирает сотрудников и назначает заказы по рекомендации
Процедура ВыполнитьНазначение() Экспорт

ПубликацииЗаказов = НовыеЗаказыИПубликации();

Для Каждого Запись Из ПубликацииЗаказов Цикл

Ранг = Запись.Ранг;
МинимальныйУровень = Запись.Уровень;
Заказ = Запись.Заказ;

//доработка 22.05.2019(

// Если условие выполнено, то пора назначать
ПораНазначать = Ранг = 9 И МинимальныйУровень = 3;
//ПораНазначать = Ранг = 6 И МинимальныйУровень = 3;

```

```

//)Доработка 22.05.2019
Если Не ПораНазначать Тогда Продолжить; КонецЕсли;

//V1
//Подобранные = ПодобратьСотрудниковКЗаказу(Заказ, Ранг, МинимальныйУровень);
//ВыделенныеСотрудники = ВыделитьСотрудниковИзСтатистики(Подобранные);
//Сотрудник = РекомендоватьСотрудника(Заказ, Подобранные);
//Сообщить(«Рекомендован « + Сотрудник);
//НазначитьСотрудникаНаЗН(Запись.Заказ, Сотрудник);

//V2
ИтерУровень = Заказ.ОписаниеНеисправности.д_Сложность;
//Если заказ 1 сложности тогда идем вниз по уровням, иначе вверх до 1 и
останавливаемся.
Направление = ?(Заказ.ОписаниеНеисправности.д_Сложность = 1, 1, -1);
МастераСоВсехУровней = ПодобратьСотрудниковКЗаказу(Заказ, Ранг, 3);

//V3 (доработка ШенгелияДЮ 24.05.2019)
Разрешенные = ОтборНаСервере(МастераСоВсехУровней,
Заказ.ОписаниеНеисправности.д_Сложность);
Если Разрешенные.Количество() > 0 Тогда
    Разрешенные.Сортировать(«Ранг, УровеньМастера»);
    НазначитьСотрудникаНаЗН(Заказ, Разрешенные[0].Сотрудник);
КонецЕсли;

//Пока ИтерУровень < 4 И ИтерУровень > 0 Цикл
//    ПодходящиеПоУровню = МастераСоВсехУровней.НайтиСтроки(Новый
Структура(«УровеньМастера, Разрешен», ИтерУровень, Истина));
//    Если ПодходящиеПоУровню.Количество() = 0 Тогда
//        ИтерУровень = ИтерУровень + Направление;
//        Продолжить;
//    КонецЕсли;
//    НазначитьСотрудникаНаЗН(Заказ, ПодходящиеПоУровню[0].Сотрудник);
//    Прервать;
//КонецЦикла;

КонецЦикла;

```

КонецПроцедуры

//Возвращает публикации по заказу

Функция ПолучитьПубликации(Заказ)

```
Запрос = Новый Запрос();
Запрос.Текст = «ВЫБРАТЬ
    |     д_ПубликацииЗаказов.Заказ КАК Заказ,
    |     д_ПубликацииЗаказов.Сотрудник КАК Сотрудник,
    |     д_ПубликацииЗаказов.ДатаПубликации КАК ДатаПубликации,
    |     д_ПубликацииЗаказов.ДатаПриема КАК ДатаПриема,
    |     д_ПубликацииЗаказов.Ранг КАК Ранг,
    |     д_ПубликацииЗаказов.Уровень КАК Уровень
| ИЗ
    |     РегистрСведений.д_ПубликацииЗаказов КАК д_ПубликацииЗаказов
| ГДЕ
    |     д_ПубликацииЗаказов.Заказ = &Заказ»;
Запрос.УстановитьПараметр(«Заказ», Заказ);
Возврат Запрос.Выполнить().Выгрузить();
```

КонецФункции

// Получаем заказы, у которых не указан сотрудник.

// Для каждого заказа берем ранг, которому он был опубликован

//

// Возвращаемое значение:

// ТаблицаЗначений - Таблица с колонками: Заказ, Ранг

//

Функция НовыеЗаказыИПубликации() Экспорт

```
Запрос = Новый Запрос();
Запрос.Текст =
«ВЫБРАТЬ
    |     _ЗаказНарядСервисногоЦентра.Ссылка КАК Заказ,
    |     МАКСИМУМ(ЕСТЬNULL(д_ПубликацииЗаказов.Ранг, 0)) КАК Ранг,
    |     МАКСИМУМ(ЕСТЬNULL(д_ПубликацииЗаказов.Уровень, 0)) КАК Уровень
| ИЗ
```

```

|      Документ._ЗаказНарядСервисногоЦентра КАК _ЗаказНарядСервисногоЦентра
|
|      ЛЕВОЕ      СОЕДИНЕНИЕ      РегистрСведений.д_ПубликацииЗаказов      КАК
д_ПубликацииЗаказов
|
|      ПО _ЗаказНарядСервисногоЦентра.Ссылка = д_ПубликацииЗаказов.Заказ
|ГДЕ
|
|      _ЗаказНарядСервисногоЦентра.Сотрудник      =
ЗНАЧЕНИЕ(Справочник.Сотрудники.ПустаяСсылка)
|
|      И НЕ _ЗаказНарядСервисногоЦентра.ПометкаУдаления
|
|      И 1=1
|СГРУППИРОВАТЬ ПО
|
|      _ЗаказНарядСервисногоЦентра.Ссылка»;
//ТЕСТ
//Запрос.Текст      =      СтрЗаменить(Запрос.Текст,      «1=1»,
«_ЗаказНарядСервисногоЦентра.Организация.Наименование = «!Администрация»«»);
      Возврат Запрос.Выполнить().Выгрузить();
КонецФункции

// Для указанного заказа выбирает лучшего сотрудника из таблицы со статистикой
//
// Параметры:
// Заказ      -      ДокументСсылка._ЗаказНарядСервисногоЦентра -
Заказ для подбора
// СтатистикаСотрудников      -      ТаблицаЗначений - Таблица со статистикой сотрудников
// РангРассылки      -      Число - Пока не используется
//
// Возвращаемое значение:
//      - СправочникСсылка.Сотрудники
//
функция РекомендоватьСотрудника(Заказ, СтатистикаСотрудников, РангРассылки = 0) Экспорт

      Если Не ЗначениеЗаполнено(Заказ) Или СтатистикаСотрудников.Количество() = 0 Тогда
Возврат Справочники.Сотрудники.ПустаяСсылка(); КонецЕсли;

      Разрешенные = СтатистикаСотрудников.СкопироватьКолонки();

      Если СтатистикаСотрудников.Количество() <> 0 Тогда

//Рекомендуем только тех сотрудников, кому можно выдать заказ

```

```

РазрешенныеСтроки = СтатистикаСотрудников.НайтиСтроки(Новый
Структура(«Разрешен», Истина));

Для Каждого Строка Из РазрешенныеСтроки Цикл
    Стр = Разрешенные.Добавить();
    ЗаполнитьЗначенияСвойств(Стр, Строка);
КонецЦикла;

Если Разрешенные.Количество() > 0 Тогда
    Возврат Разрешенные[0].Сотрудник;
КонецЕсли;

КонецЕсли;

Возврат Справочники.Сотрудники.ПустаяСсылка();

КонецФункции

// Подбирает сотрудников с рангом выше (число больше - ранг ниже) чем указанный в параметре
// «РангПодбора».
// Ранг определяется прямо в запросе.
// Если логика присвоения ранга меняется, запрос необходимо изменить!
//
// Параметры:
// Заказ - ДокументСсылка._ЗаказНарядСервисногоЦентра - Заказ для подбора
// РангПодбора - Число - Будут подобраны сотрудники с рангом больше этого числа.
// (Число меньше - ранг больше)
//
// Возвращаемое значение:
// ТаблицаЗначений - Таблица с сотрудниками и их статистикой
//
Функция ПодобратьСотрудниковКЗаказу(Заказ, РангПодбора, УровеньМастера) Экспорт

    Результат = Новый Структура(«ВсеЗаказы, СуммаЗакрытыхРабот, КоличествоЗакрытых,
    КоличествоВыполнено, КоличествоДиагностик,
    |КоличествоОтказов, КоличествоВРаботе,
    ГарантийныйРемонтСвой, ГарантийныйРемонтЧужой, ЗабралТехникуВСчетДиагностики,
    |ПроцентОтказов, ПроцентДиагностик, СреднийЧек,
    РангМастера, МаксКолЗаказов, МаксУровеньСложностиЗаказов, КолСтрокВРезультате»);

```

```

Запрос = Новый Запрос;
Запрос.Текст =
#Область Запрос
«ВЫБРАТЬ
|      СУММА(ВЫБОР
|
|      КОГДА  СЦентр_СреднийЧекОбороты.КоличествоЗаказовОборот  ЕСТЬ
NULL
|
|      ТОГДА  0
|
|      ИНАЧЕ  СЦентр_СреднийЧекОбороты.КоличествоЗаказовОборот
|
|      КОНЕЦ) КАК  ВсегоЗаказов,
|
|      СУММА(ВЫБОР
|
|      КОГДА  СЦентр_СреднийЧекОбороты.Регистратор.Закрыт
|
|      И  (СЦентр_СреднийЧекОбороты.СостояниеЗаказа  =
&СтатусВыполнено
|
|      ИЛИ
СЦентр_СреднийЧекОбороты.СостояниеЗаказа = &СтатусДиагностика
|
|      ИЛИ
СЦентр_СреднийЧекОбороты.СостояниеЗаказа = &СтатусОтказ)
|
|      ТОГДА  СЦентр_СреднийЧекОбороты.СтоимостьРаботОборот
|
|      ИНАЧЕ  0
|
|      КОНЕЦ) КАК  СуммаЗакрытыхРабот,
|
|      СУММА(ВЫБОР
|
|      КОГДА  СЦентр_СреднийЧекОбороты.Регистратор.Закрыт
|
|      И  (СЦентр_СреднийЧекОбороты.СостояниеЗаказа  =
&СтатусВыполнено
|
|      ИЛИ
СЦентр_СреднийЧекОбороты.СостояниеЗаказа = &СтатусДиагностика
|
|      ИЛИ
СЦентр_СреднийЧекОбороты.СостояниеЗаказа = &СтатусОтказ)
|
|      ТОГДА  1
|
|      ИНАЧЕ  0
|
|      КОНЕЦ) КАК  КоличествоЗакрытых,
|
|      СУММА(ВЫБОР
|
|      КОГДА
&СтатусВыполнено
|
|      И  СЦентр_СреднийЧекОбороты.Регистратор.Закрыт
|
|      ТОГДА  1
|
|      ИНАЧЕ  0
|
|      КОНЕЦ) КАК  КоличествоВыполнено,
|
|      СУММА(ВЫБОР

```

```

|
|          КОГДА          СЦентр_СреднийЧекОбороты.СостояниеЗаказа          =
&СтатусДиагностика
|
|          И СЦентр_СреднийЧекОбороты.Регистратор.Закрыт
|
|          ТОГДА 1
|
|          ИНАЧЕ 0
|
|          КОНЕЦ) КАК КоличествоДиагностик,
|
|          СУММА(ВЫБОР
|
|          КОГДА СЦентр_СреднийЧекОбороты.СостояниеЗаказа = &СтатусОтказ
|
|          И СЦентр_СреднийЧекОбороты.Регистратор.Закрыт
|
|          ТОГДА 1
|
|          ИНАЧЕ 0
|
|          КОНЕЦ) КАК КоличествоОтказов,
|
|          СУММА(ВЫБОР
|
|          КОГДА НЕ СЦентр_СреднийЧекОбороты.Регистратор.Закрыт
|
|          И НЕ(СЦентр_СреднийЧекОбороты.СостояниеЗаказа =
&СтатусГарантийныйРемонтЧужой
|
|          ИЛИ
СЦентр_СреднийЧекОбороты.СостояниеЗаказа = &СтатусГарантийныйРемонтСвой
|
|          ИЛИ
СЦентр_СреднийЧекОбороты.СостояниеЗаказа = &СтатусРассрочкаПлатежа)
|
|          ТОГДА 1
|
|          ИНАЧЕ 0
|
|          КОНЕЦ) КАК КоличествоВРаботе,
|
|          СУММА(ВЫБОР
|
|          КОГДА          СЦентр_СреднийЧекОбороты.СостояниеЗаказа          =
&СтатусГарантийныйРемонтСвой
|
|          И СЦентр_СреднийЧекОбороты.Регистратор.Закрыт
|
|          ТОГДА 1
|
|          ИНАЧЕ 0
|
|          КОНЕЦ) КАК ГарантийныйРемонтСвой,
|
|          СУММА(ВЫБОР
|
|          КОГДА          СЦентр_СреднийЧекОбороты.СостояниеЗаказа          =
&СтатусГарантийныйРемонтЧужой
|
|          И СЦентр_СреднийЧекОбороты.Регистратор.Закрыт
|
|          ТОГДА 1
|
|          ИНАЧЕ 0
|
|          КОНЕЦ) КАК ГарантийныйРемонтЧужой,
|
|          СУММА(ВЫБОР
|
|          КОГДА          СЦентр_СреднийЧекОбороты.СостояниеЗаказа          =
&СтатусЗабралТехникуВСчетДиагностики

```

```

|
|                                     И СЦентр_СреднийЧекОбороты.Регистратор.Закрит
|
|                                     ТОГДА 1
|
|                                     ИНАЧЕ 0
|
|                                     КОНЕЦ) КАК ЗабралТехникуВСчетДиагностики,
|
|                                     Сотрудники.Ссылка КАК Сотрудник,
|
|                                     Сотрудники.Владелец КАК Организация,
|
|                                     Сотрудникд_Профили.Профиль КАК Профиль
|ПОМЕСТИТЬ ВремТаб
|
|ИЗ
|
|                                     РегистрСведений.ОтклоненияОтГрафиковРаботыРесурсов
|
|                                     ОтклоненияОтГрафиковРаботыРесурсов
|
|                                     КАК
|
|                                     ВНУТРЕННЕЕ СОЕДИНЕНИЕ Справочник.Сотрудники КАК Сотрудники
|
|                                     ВНУТРЕННЕЕ СОЕДИНЕНИЕ Справочник.Сотрудники.д_Профили
|
|                                     КАК
|
|                                     Сотрудникд_Профили
|
|                                     ЛЕВОЕ
|
|                                     СОЕДИНЕНИЕ
|
|                                     РегистрНакопления.СЦентр_СреднийЧек.Обороты(, , Регистратор, ) КАК СЦентр_СреднийЧекОбороты
|
|                                     ПО
|
|                                     Сотрудникд_Профили.Профиль
|
|                                     =
|
|                                     СЦентр_СреднийЧекОбороты.ВидИзделия.Родитель
|
|                                     И
|
|                                     Сотрудникд_Профили.Ссылка
|
|                                     =
|
|                                     СЦентр_СреднийЧекОбороты.Сотрудник
|
|                                     ПО (Сотрудникд_Профили.Ссылка = Сотрудники.Ссылка)
|
|                                     ПО
|
|                                     ОтклоненияОтГрафиковРаботыРесурсов.РесурсПредприятия
|
|                                     =
|
|                                     Сотрудники.Ссылка
|
|ГДЕ
|
|                                     Сотрудники.д_ОсновнойТипТехники
|
|                                     ЗНАЧЕНИЕ(Справочник._ВидыИзделий.ПустаяСсылка)
|
|                                     <>
|
|                                     И НЕ Сотрудники.Недействителен
|
|                                     И НЕ Сотрудники.ПометкаУдаления
|
|                                     И ОтклоненияОтГрафиковРаботыРесурсов.День = &ТекущаяДата
|
|                                     И ОтклоненияОтГрафиковРаботыРесурсов.ЧасыРаботы В (5, 7, 8)
|
|СГРУППИРОВАТЬ ПО
|
|                                     Сотрудники.Ссылка,
|
|                                     Сотрудники.Владелец,
|
|                                     Сотрудникд_Профили.Профиль
|;
|
|
|////////////////////////////////////
|
|ВЫБРАТЬ
|
|                                     ВремТаб.ВсегоЗаказов КАК ВсегоЗаказов,

```

```

|   ВремТаб.СуммаЗакрытыхРабот КАК СуммаЗакрытыхРабот,
|   ВремТаб.КоличествоЗакрытых КАК КоличествоЗакрытых,
|   ВремТаб.КоличествоВыполнено КАК КоличествоВыполнено,
|   ВремТаб.КоличествоДиагностик КАК КоличествоДиагностик,
|   ВремТаб.КоличествоОтказов КАК КоличествоОтказов,
|   ЕСТЬNULL(ВремТаб.КоличествоВРаботе, 0) КАК КоличествоВРаботе,
|   ВремТаб.ГарантийныйРемонтСвой КАК ГарантийныйРемонтСвой,
|   ВремТаб.ГарантийныйРемонтЧужой КАК ГарантийныйРемонтЧужой,
|   ВремТаб.ЗабралТехникуВСчетДиагностики КАК ЗабралТехникуВСчетДиагностики,
|   ЕСТЬNULL(ВЫБОР
|
|           КОГДА ВремТаб.КоличествоЗакрытых = 0
|
|               ТОГДА 0
|
|           ИНАЧЕ ВремТаб.КоличествоОтказов / ВремТаб.КоличествоЗакрытых *
100
|
|           КОНЕЦ, 0) КАК ПроцентОтказов,
|   ЕСТЬNULL(ВЫБОР
|
|           КОГДА ВремТаб.КоличествоЗакрытых = 0
|
|               ТОГДА 0
|
|           ИНАЧЕ ВремТаб.КоличествоДиагностик / ВремТаб.КоличествоЗакрытых
* 100
|
|           КОНЕЦ, 0) КАК ПроцентДиагностик,
|   ВЫБОР
|
|           КОГДА ВремТаб.КоличествоЗакрытых = 0
|
|               ТОГДА 0
|
|           ИНАЧЕ ВремТаб.СуммаЗакрытыхРабот / ВремТаб.КоличествоЗакрытых
|   КОНЕЦ КАК СреднийЧек,
|   ВремТаб.Организация КАК Организация,
|   ВремТаб.Сотрудник КАК Сотрудник,
|   ВремТаб.Профиль КАК Профиль
|ПОМЕСТИТЬ сПроцентомОтказов
|ИЗ
|   ВремТаб КАК ВремТаб
|;
|
|////////////////////////////////////
|ВЫБРАТЬ
|
|   МИНИМУМ(д_НастройкаРангов.Ранг) КАК Ранг,
|   сПроцентамиОтказов.Организация КАК Организация,

```

```

|      сПроцентамиОтказов.Сотрудник КАК Сотрудник,
|
|      сПроцентамиОтказов.КоличествоВРаботе КАК КоличествоВРаботе,
|
|      сПроцентамиОтказов.Профиль КАК Профиль,
|
|      МИНИМУМ(д_НастройкаРангов.МаксУровеньСложностиЗаказов)           КАК
МаксУровеньСложностиЗаказов
|ПОМЕСТИТЬ РангиСотрудников
|
|ИЗ
|
|      сПроцентомОтказов КАК сПроцентамиОтказов
|
|      ВНУТРЕННЕЕ      СОЕДИНЕНИЕ      РегистрСведений.д_НастройкаРангов      КАК
д_НастройкаРангов
|
|      ПО сПроцентамиОтказов.СреднийЧек >= д_НастройкаРангов.СреднийЧекОт
|
|      И                  сПроцентамиОтказов.СреднийЧек      <
д_НастройкаРангов.СреднийЧекДо
|
|      И                  (сПроцентамиОтказов.ПроцентДиагностик      +
сПроцентамиОтказов.ПроцентОтказов >= д_НастройкаРангов.ПроцентОтказовДиагностикОт)
|
|      И                  (сПроцентамиОтказов.ПроцентДиагностик      +
сПроцентамиОтказов.ПроцентОтказов < д_НастройкаРангов.ПроцентОтказовДиагностикДо)
|
|      И                  сПроцентамиОтказов.КоличествоЗакрытых      >=
д_НастройкаРангов.МинКоличествоЗаказов
|
|      И                  сПроцентамиОтказов.Организация      =
д_НастройкаРангов.Организация
|
|      И сПроцентамиОтказов.Профиль = д_НастройкаРангов.ТипТехники
|
|СГРУППИРОВАТЬ ПО
|
|      сПроцентамиОтказов.Организация,
|
|      сПроцентамиОтказов.Сотрудник,
|
|      сПроцентамиОтказов.КоличествоВРаботе,
|
|      сПроцентамиОтказов.Профиль
|;
|
|////////////////////////////////////
|ВЫБРАТЬ
|
|      МИНИМУМ(д_НастройкаРангов.Ранг) КАК Ранг,
|
|      сПроцентамиОтказов.Организация КАК Организация,
|
|      сПроцентамиОтказов.Сотрудник КАК Сотрудник,
|
|      сПроцентамиОтказов.КоличествоВРаботе КАК КоличествоВРаботе,
|
|      сПроцентамиОтказов.Профиль КАК Профиль
|ПОМЕСТИТЬ РангиСотрудниковРазреш
|
|ИЗ
|
|      сПроцентомОтказов КАК сПроцентамиОтказов

```

```

|          ВНУТРЕННЕЕ      СОЕДИНЕНИЕ      РегистрСведений.д_НастройкаРангов      КАК
д_НастройкаРангов

|          ПО сПроцентамиОтказов.СреднийЧек >= д_НастройкаРангов.СреднийЧекОт
|          И                  сПроцентамиОтказов.СреднийЧек      <
д_НастройкаРангов.СреднийЧекДо

|          И          (сПроцентамиОтказов.ПроцентДиагностик      +
сПроцентамиОтказов.ПроцентОтказов >= д_НастройкаРангов.ПроцентОтказовДиагностикОт)

|          И          (сПроцентамиОтказов.ПроцентДиагностик      +
сПроцентамиОтказов.ПроцентОтказов < д_НастройкаРангов.ПроцентОтказовДиагностикДо)

|          И          сПроцентамиОтказов.КоличествоЗакрытых      >=
д_НастройкаРангов.МинКоличествоЗаказов

|          И          сПроцентамиОтказов.Организация      =
д_НастройкаРангов.Организация

|          И сПроцентамиОтказов.Профиль = д_НастройкаРангов.ТипТехники

|          И          сПроцентамиОтказов.КоличествоВРаботе      <
д_НастройкаРангов.МаксКолЗаказов

|
|СГРУППИРОВАТЬ ПО
|
|    сПроцентамиОтказов.Организация,
|
|    сПроцентамиОтказов.Сотрудник,
|
|    сПроцентамиОтказов.КоличествоВРаботе,
|
|    сПроцентамиОтказов.Профиль
|;
|
|////////////////////////////////////
|ВЫБРАТЬ
|
|    _ЗаказНарядСервисногоЦентра.Ссылка КАК Ссылка,
|
|    РангиСотрудников.Ранг КАК Ранг,
|
|    РангиСотрудников.Сотрудник КАК Сотрудник,
|
|    РангиСотрудников.КоличествоВРаботе КАК КоличествоВРаботе,
|
|    РангиСотрудниковРазреш.Сотрудник ЕСТЬ НЕ NULL КАК Разрешен,
|
|    РангиСотрудников.Профиль КАК Профиль,
|
|    РангиСотрудников.Организация КАК Организация,
|
|    _ЗаказНарядСервисногоЦентра.ОписаниеНеисправности.д_Сложность      КАК
СложностьНеисправности,
|
|    РангиСотрудников.МаксУровеньСложностиЗаказов КАК УровеньМастера
|ИЗ
|
|    Документ._ЗаказНарядСервисногоЦентра КАК _ЗаказНарядСервисногоЦентра
|
|          ВНУТРЕННЕЕ СОЕДИНЕНИЕ РангиСотрудников КАК РангиСотрудников
|
|          ЛЕВОЕ      СОЕДИНЕНИЕ      РангиСотрудниковРазреш      КАК
РангиСотрудниковРазреш

```

```

|                ПО                РангиСотрудников.Сотрудник                =
РангиСотрудниковРазреш.Сотрудник

|                И                РангиСотрудников.Профиль                =
РангиСотрудниковРазреш.Профиль

|                И                РангиСотрудников.Организация                =
РангиСотрудниковРазреш.Организация

|                И РангиСотрудников.Ранг = РангиСотрудниковРазреш.Ранг
|                _ЗаказНарядСервисногоЦентра.Организация                =
РангиСотрудников.Организация

|                И                _ЗаказНарядСервисногоЦентра.ВидИзделия.Родитель                =
РангиСотрудников.Профиль

```

|ГДЕ

| _ЗаказНарядСервисногоЦентра.Ссылка = &ПодбираемыйЗаказ

| И РангиСотрудников.Ранг <= &МинимальныйРанг

| И РангиСотрудников.МаксУровеньСложностиЗаказов <= &МинимальныйУровеньМастера

|

|УПОРЯДОЧИТЬ ПО

| Ранг,

| КоличествоВРаботе,

| УровеньМастера»;

#КонецОбласти

Запрос.УстановитьПараметр(«ТекущаяДата», НачалоДня(ТекущаяДата()));

Запрос.УстановитьПараметр(«Организация», Заказ.Организация);

Запрос.УстановитьПараметр(«ВидИзделия», Заказ.ВидИзделия);

Запрос.УстановитьПараметр(«СложностьРемонта»,
Заказ.ОписаниеНеисправности.д_Сложность);

Запрос.УстановитьПараметр(«ПодбираемыйЗаказ», Заказ);

Запрос.УстановитьПараметр(«МинимальныйРанг», РангПодбора);

Запрос.УстановитьПараметр(«МинимальныйУровеньМастера», УровеньМастера);

Запрос.УстановитьПараметр(«СтатусВыполнено»,
Справочники._СостояниеЗаказовНарядов.РемонтЗавершен);

Запрос.УстановитьПараметр(«СтатусГарантийныйРемонтСвой»,
Справочники._СостояниеЗаказовНарядов.ГарантийныйРемонтСвой);

Запрос.УстановитьПараметр(«СтатусГарантийныйРемонтЧужой»,
Справочники._СостояниеЗаказовНарядов.ГарантийныйРемонтЧужой);

Запрос.УстановитьПараметр(«СтатусДиагностика»,
Справочники._СостояниеЗаказовНарядов.Диагностика);

Запрос.УстановитьПараметр(«СтатусЗабралТехникуВСчетДиагностики»,
Справочники._СостояниеЗаказовНарядов.ЗабралТехникуВСчетДиагностики);

```
Запрос.УстановитьПараметр(«СтатусОтказ»,  
Справочники._СостояниеЗаказовНарядов.ОтказОтРабот);
```

```
Запрос.УстановитьПараметр(«СтатусРассрочкаПлатежа»,  
Справочники._СостояниеЗаказовНарядов.РассрочкаПлатежа);
```

```
Запрос.УстановитьПараметр(«СтатусСозвонитьсяСКлиентом»,  
Справочники._СостояниеЗаказовНарядов.НайтиПоКоду(«00000009»));
```

```
Возврат Запрос.Выполнить().Выгрузить();
```

КонецФункции

```
//Публикует заказ указанным сотрудникам, и записывает факт публикации
```

```
//по рангу указанному в параметре «Ранг»
```

```
Процедура ОпубликоватьЗаказ(Заказ, ЦелевыеСотрудники, Ранг, Уровень) Экспорт
```

```
Если ЦелевыеСотрудники.Количество() = 0 Тогда Возврат; КонецЕсли;
```

```
НаборЗаписей = РегистрыСведений.д_ПубликацииЗаказов.СоздатьНаборЗаписей();
```

```
НаборЗаписей.Отбор.Заказ.Установить(Заказ);
```

```
Для Каждого Сотрудник Из ЦелевыеСотрудники Цикл
```

```
    СтрокаНабора = НаборЗаписей.Добавить();
```

```
    СтрокаНабора.Заказ = Заказ;
```

```
    СтрокаНабора.Сотрудник = Сотрудник;
```

```
    СтрокаНабора.ДатаПубликации = ТекущаяДата();
```

```
    СтрокаНабора.Ранг = Ранг;
```

```
    СтрокаНабора.Уровень = Уровень;
```

```
КонецЦикла;
```

```
НаборЗаписей.Записать(Истина);
```

КонецПроцедуры

```
//Возвращает массив сотрудников из колонки «Сотрудник» таблицы с сотрудниками.
```

```
Функция ВыделитьСотрудниковИзСтатистики(СтатистикаПоМастерам)
```

```
    Мастера = Новый Массив;
```

```
Если Не ЗначениеЗаполнено(СтатистикаПоМастерам) Тогда Возврат Мастера; КонецЕсли;
Для Каждого Строка Из СтатистикаПоМастерам Цикл
    Мастера.Добавить(Строка.Сотрудник);
КонецЦикла;
Возврат Мастера;
```

КонецФункции

```
//Выполняет функцию модуля МобильноеПриложениеМастера УстановитьЗаказКакПринятый
//Возвращает результат назначения (Булево)
```

Функция НазначитьСотрудникаНаЗН(Заказ, Сотрудник) Экспорт

```
Если Не ЗначениеЗаполнено(Заказ) И Не ЗначениеЗаполнено(Сотрудник) Тогда Возврат Ложь;
КонецЕсли;
```

```
Описание = «»;
```

```
Успешно =
МобильноеПриложениеМастераСервер.УстановитьЗаказКакПринятый(Сотрудник.УникальныйИдентификатор(),
Заказ.УникальныйИдентификатор(), Описание);
```

```
Если Не Успешно Тогда Сообщить(Описание); КонецЕсли;
```

```
Сообщить(«На заказ №» + Строка(Заказ.Номер) + « установлен мастер « +
Строка(Сотрудник));
```

```
Возврат Успешно;
```

КонецФункции

```
//Получает последний ранг, которому был опубликован заказ.
```

Функция ПолучитьПоследнийРангПолучатель(Заказ) Экспорт

```
Запрос = Новый Запрос();
```

```
Запрос.Текст =
```

```
«ВЫБРАТЬ
```

```
|     д_ПубликацииЗаказов.Заказ КАК Заказ,
```

```
|     МАКСИМУМ(д_ПубликацииЗаказов.Ранг) КАК Ранг
```

```
|ИЗ
```

```
|     РегистрСведений.д_ПубликацииЗаказов КАК д_ПубликацииЗаказов
```

```
|ГДЕ
```

```
|     д_ПубликацииЗаказов.Заказ = &Заказ
```

```
|
```

```
|СГРУППИРОВАТЬ ПО
```

```
|     д_ПубликацииЗаказов.Заказ»;
```

```
Запрос.УстановитьПараметр(«Заказ», Заказ);
```

```
Выборка = Запрос.Выполнить().Выбрать();
```

```

Если Выборка.Следующий() Тогда
    Возврат Выборка.Ранг;
КонецЕсли;
Возврат 0;
КонецФункции

//Возвращает сотрудника принявшего заказ, дата записывается в параметр ДатаПриема.
//Приемка определяется по регистру д_ПубликацииЗаказов, колонка ДатаПриема.
Функция ПолучитьСотрудникаПринявшегоЗаказ(Заказ, ДатаПриема)

    Сотрудник = Справочники.Сотрудники.ПустаяСсылка();
    Запрос = Новый Запрос();
    Запрос.Текст = «ВЫБРАТЬ
        |     д_ПубликацииЗаказовСрезПоследних.Сотрудник КАК Сотрудник,
        |     д_ПубликацииЗаказовСрезПоследних.Заказ КАК Заказ,
        |     д_ПубликацииЗаказовСрезПоследних.ДатаПриема КАК ДатаПриема
        | ИЗ
        |     РегистрСведений.д_ПубликацииЗаказов.СрезПоследних(, Заказ =
&Заказ) КАК д_ПубликацииЗаказовСрезПоследних
        | ГДЕ
        |     д_ПубликацииЗаказовСрезПоследних.ДатаПриема <> ДАТАВРЕМЯ(1, 1,
1)»;

    Запрос.УстановитьПараметр(«Заказ», Заказ);
    Выборка = Запрос.Выполнить().Выбрать();
    Если Выборка.Следующий() Тогда
        ДатаПриема = Выборка.ДатаПриема;
        Сотрудник = Выборка.Сотрудник;
    КонецЕсли;
    Возврат Сотрудник;
КонецФункции

//Добавляет строку к накапливаемой строке логa.
//Данная функция не записывает лог на диск, только задает формат для каждой добавляемой строки.
Процедура ДобавитьКЛог(Строка)
    Лог = Формат(ТекущаяДата(), «ДФ=DT») + Строка + Символы.ПС + Символы.ПС + Лог;
КонецПроцедуры

```

Функция ОтборНаСервере(ТабЗначений, УровеньСложностиРемонта)

Построитель = Новый ПостроительЗапроса;

Построитель.ИсточникДанных = Новый ОписаниеИсточникаДанных(ТабЗначений);

тОтбор = Построитель.Отбор;

ИмяОтбора = «Разрешен»;

тОтбор.Добавить(ИмяОтбора);

тОтбор[ИмяОтбора].ВидСравнения = ВидСравнения.Равно;

тОтбор[ИмяОтбора].Значение = Истина;

тОтбор[ИмяОтбора].Использование = Истина;

ИмяОтбора = «УровеньМастера»;

тОтбор.Добавить(ИмяОтбора);

Если УровеньСложностиРемонта > 1 Тогда

 тОтбор[ИмяОтбора].ВидСравнения = ВидСравнения.МеньшеИлиРавно;

Иначе

 тОтбор[ИмяОтбора].ВидСравнения = ВидСравнения.БольшеИлиРавно;

КонецЕсли;

тОтбор[ИмяОтбора].Значение = УровеньСложностиРемонта;

тОтбор[ИмяОтбора].Использование = Истина;

Построитель.Выполнить();

Возврат Построитель.Результат.Выгрузить();

КонецФункции

Приложение В. Табель рабочего времени на месяц

&НаСервере

Процедура ЗаполнитьНаСервере(Год, Месяц)

МассСотрудников = Новый Массив;

МассСотрудников = ПолучитьСписокНаСервере();

Организация = Объект.Организация;

Стр = Стр;

Год = Год(Объект.ПериодРегистрации);

```

Месяц = Месяц(Объект.ПериодРегистрации);
СегоднДень = День(ТекущаяДата());

День = 1;
Дата1 = Дата(Год, Месяц(Объект.ПериодРегистрации), День);

Дата2 = КонецМесяца(Дата1);
Запрос = Новый Запрос;
//Доработка АТЭ ШенгелияДЮ 2019.04.19
//В запросе добавлена проверка на список сотрудников из табеля.
//Теперь запрос выводит только тех сотрудников, которых нет в таблице.
Запрос.Текст =
«ВЫБРАТЬ
|      влож1.Сотрудник КАК Сотрудник,
|      ПриминУволВТекМес.Период КАК Период,
|      ПриминУволВТекМес.ЗанимаемыхСтавок КАК ЗанимаемыхСтавок
|ПОМЕСТИТЬ ВременнаяТаблицаОсн
|ИЗ
|      (ВЫБРАТЬ
|          Влож.Сотрудник КАК Сотрудник
|      ИЗ
|          (ВЫБРАТЬ
|              СотрудникиСрезПоследних.Сотрудник КАК Сотрудник
|          ИЗ
|              РегистрСведений.Сотрудники.СрезПоследних(&Дата1, Организация =
&Организация) КАК СотрудникиСрезПоследних
|          ГДЕ
|              СотрудникиСрезПоследних.ЗанимаемыхСтавок > 0
|      ОБЪЕДИНИТЬ ВСЕ
|      ВЫБРАТЬ
|          Сотрудники.Сотрудник
|      ИЗ
|          РегистрСведений.Сотрудники КАК Сотрудники
|      ГДЕ
|          Сотрудники.ЗанимаемыхСтавок > 0
|      И Сотрудники.Период МЕЖДУ &Дата1 И &Дата2
|      И Сотрудники.Организация = &Организация) КАК Влож

```

```

|
| СГРУППИРОВАТЬ ПО
|     Влож.Сотрудник) КАК влож1
|     ЛЕВОЕ СОЕДИНЕНИЕ (ВЫБРАТЬ
|         Сотрудники.Период КАК Период,
|         Сотрудники.ЗанимаемыхСтавок КАК ЗанимаемыхСтавок,
|         Сотрудники.Сотрудник КАК Сотрудник
|     ИЗ
|         РегистрСведений.Сотрудники КАК Сотрудники
|     ГДЕ
|         Сотрудники.Период МЕЖДУ &дата1 И &дата2) КАК ПринимУволВТекМес
|     ПО влож1.Сотрудник = ПринимУволВТекМес.Сотрудник
|;
|
| //////////////////////////////////////
| ВЫБРАТЬ
|     д_ТабельНаМесяцТабЧасть.Сотрудники КАК Сотрудники
| ПОМЕСТИТЬ ВременнаяТаблицаТабЧасть
| ИЗ
|     Документ.д_ТабельНаМесяц.ТабЧасть КАК д_ТабельНаМесяцТабЧасть
| ГДЕ
|     д_ТабельНаМесяцТабЧасть.Ссылка = &Ссылка
|;
|
| //////////////////////////////////////
| ВЫБРАТЬ
|     ВременнаяТаблицаОсн.Сотрудник КАК Сотрудник,
|     ВременнаяТаблицаОсн.Период КАК Период,
|     ВременнаяТаблицаОсн.ЗанимаемыхСтавок КАК ЗанимаемыхСтавок
| ИЗ
|     ВременнаяТаблицаОсн КАК ВременнаяТаблицаОсн
|     ЛЕВОЕ СОЕДИНЕНИЕ ВременнаяТаблицаТабЧасть КАК ВременнаяТаблицаТабЧасть
|     ПО ВременнаяТаблицаОсн.Сотрудник = ВременнаяТаблицаТабЧасть.Сотрудники
| ГДЕ
|     ВременнаяТаблицаТабЧасть.Сотрудники ЕСТЬ NULL
| ИТОГИ
|     КОЛИЧЕСТВО(РАЗЛИЧНЫЕ Период)
| ПО
|     Сотрудник»;

```

```
Запрос.УстановитьПараметр(«Дата1», Дата1);
Запрос.УстановитьПараметр(«Дата2», Дата2);
Запрос.УстановитьПараметр(«Организация», Организация);
Запрос.УстановитьПараметр(«Ссылка», Объект.Ссылка);
РезультатЗапроса = Запрос.Выполнить();

ВыборкаСотр = РезультатЗапроса.Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам);
```

Пока ВыборкаСотр.Следующий() Цикл

```
Сотр = ВыборкаСотр.Сотрудник;
День = 1;
Дата1 = Дата(Год, Месяц, День);
Дата2 = КонецМесяца(Дата1);
КолДней = Окр((Дата2 - Дата1) / (60 * 60 * 24));
```

```
НовСтр = Объект.ТабЧасть.Добавить();
```

Если ВыборкаСотр.Период = 0 Тогда

```
НовСтр.Сотрудники = Сотр;
```

Для т=1 по КолДней Цикл

```
пер = «День» + т;
```

```
//Доработка АТЭ ШенгелияДЮ 2019.04.19
```

```
//Теперь значение 8 добавляется только со дня формирования
```

табеля.

```
Если Сегодня <= т Тогда
```

```
НовСтр[пер] = 8;
```

```
КонецЕсли;
```

```
КонецЦикла
```

Иначе

```
ВыборкаДетальныеЗаписи = ВыборкаСотр.Выбрать();
```

```
Уволен = 0;
```

```
Устроен = 0;
```

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Если ВыборкаДетальныеЗаписи.ЗанимаемыхСтавок > 0 Тогда

Устроен = День(ВыборкаДетальныеЗаписи.Период);

УстроенДата = ВыборкаДетальныеЗаписи.Период;

ИначеЕсли ВыборкаДетальныеЗаписи.ЗанимаемыхСтавок = 0 Тогда

Уволен = День(ВыборкаДетальныеЗаписи.Период);

УволенДата = ВыборкаДетальныеЗаписи.Период;

КонецЕсли;

//ОбщегоНазначенияКлиентСервер.СообщитьПользователю(«Сотр
«+ВыборкаСотр.Сотрудник+» Устроен «+Устроен + « Уволен «+Уволен»);

КонецЦикла;

Если Устроен > Уволен Тогда

НовСтр.Сотрудники = Сотр;

Для к = Устроен по День(КонецМесяца(Объект.ПериодРегистрации))

Цикл

пер = «День» + к;

Если Сегодня <= к Тогда

НовСтр[пер] = 8;

КонецЕсли;

КонецЦикла;

ИначеЕсли Устроен < Уволен Тогда

НовСтр.Сотрудники = Сотр;

Если Устроен = 0 Тогда Устроен = 1 КонецЕсли;

Для к = Устроен по Уволен Цикл

пер = «День» + к;

```
        Если Сегодня <= к Тогда
            НовСтр[пер] = 8;
        КонечЕсли;
    КонечЦикла;
КонечЕсли;
КонечЕсли;
КонечЦикла;
```

КонечПроцедуры

&НаКлиенте

Процедура Заполнить(Команда)

```
    Месяц = Месяц(Объект.ПериодРегистрации);
```

```
    Год = Год(Объект.ПериодРегистрации);
```

```
    Если ЭтаФорма.Модифицированность Тогда
```

```
        Режим = РежимДиалогаВопрос.ДаНетОтмена;
```

```
        Текст = «ги = «Желаете продолжить заполнение списка?»»;»
```

```
            + « en = «Do you want to continue filling list?»»;»;
```

```
        Ответ = Вопрос(НСтр(Текст), Режим, 0);
```

```
        Если Ответ = КодВозвратаДиалога.Да Тогда
```

```
            ЗаполнитьНаСервере(Год, Месяц);
```

```
        КонечЕсли;
```

```
    Иначе
```

```
        ЗаполнитьНаСервере(Год, Месяц);
```

```
        ЭтаФорма.Модифицированность = Истина;
```

```
    КонечЕсли;
```

```
//Объект.ТабЧасть.Очистить();
```

```
КонечПроцедуры
```

&НаСервере

Функция ПолучитьСписокНаСервере()

```
////////////////////////////////////
```

```
Организация = Объект.Организация;
```

```
СписокСотрудников = Новый Массив;
```

```

Запрос = Новый Запрос;
Запрос.Текст =
«ВЫБРАТЬ
|      СотрудникиСрезПоследних.Сотрудник КАК Сотрудник
|ИЗ
|      РегистрСведений.Сотрудники.СрезПоследних(&Дата, Организация = &Организация)
КАК СотрудникиСрезПоследних
|ГДЕ
|      СотрудникиСрезПоследних.ЗанимаемыхСтавок > 0»;

Запрос.УстановитьПараметр(«Дата», Объект.ПериодРегистрации);
Запрос.УстановитьПараметр(«Организация», Организация);
РезультатЗапроса = Запрос.Выполнить();

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
    СписокСотрудников.Добавить(ВыборкаДетальныеЗаписи.Сотрудник);
КонецЦикла;

Возврат СписокСотрудников;

```

КонецФункции

&НаСервере

Процедура РаспределениеДока()

```

//Распроведение Документа
Ссылка = Объект;
Запрос = Новый Запрос;
Запрос.Текст =
«ВЫБРАТЬ
|      ОтклоненияОтГрафиковРаботыРесурсов.РесурсПредприятия КАК РесурсПредприятия,
|      ОтклоненияОтГрафиковРаботыРесурсов.Год КАК Год,
|      ОтклоненияОтГрафиковРаботыРесурсов.День КАК День,
|      ОтклоненияОтГрафиковРаботыРесурсов.ВремяНачала КАК ВремяНачала,
|      ОтклоненияОтГрафиковРаботыРесурсов.ВремяОкончания КАК ВремяОкончания,
|      ОтклоненияОтГрафиковРаботыРесурсов.ЧасыРаботы КАК ЧасыРаботы,
|      ОтклоненияОтГрафиковРаботыРесурсов.ЧасовПерерывов КАК ЧасовПерерывов,

```

```

|      ОтклоненияОтГрафиковРаботыРесурсов.НеРабочийДень КАК НеРабочийДень
|ИЗ
|      Документ.д_ТабельНаМесяц.ТабЧасть КАК Бар_ТабельНпМесяцТабЧасть
|      ВНУТРЕННЕЕ СОЕДИНЕНИЕ
РегистрСведений.ОтклоненияОтГрафиковРаботыРесурсов КАК ОтклоненияОтГрафиковРаботыРесурсов
|      ПО Бар_ТабельНпМесяцТабЧасть.Сотрудники =
ОтклоненияОтГрафиковРаботыРесурсов.РесурсПредприятия
|ГДЕ
|      Бар_ТабельНпМесяцТабЧасть.Ссылка = &Ссылка»;

```

Запрос.УстановитьПараметр(«Ссылка», Ссылка);

РезультатЗапроса = Запрос.Выполнить();

ВыборкаДетальныеЗаписи= РезультатЗапроса.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

```

      М =
РегистрыСведений.ОтклоненияОтГрафиковРаботыРесурсов.СоздатьМенеджерЗаписи();

```

ЗаполнитьЗначенияСвойств(м, ВыборкаДетальныеЗаписи);

м.Прочитать();

// ОТБОР

```

НаборЗаписей
РегистрыСведений.ОтклоненияОтГрафиковРаботыРесурсов.СоздатьНаборЗаписей(); =

```

СотрУд = ВыборкаДетальныеЗаписи.РесурсПредприятия;

Год = ВыборкаДетальныеЗаписи.Год;

День = ВыборкаДетальныеЗаписи.День;

НаборЗаписей.Отбор.РесурсПредприятия.Установить(СотрУд);

НаборЗаписей.Отбор.Год.Установить(Год);

НаборЗаписей.Отбор.День.Установить(День);

// НОВАЯ ЗАПИСЬ

НоваяЗапись = НаборЗаписей.Добавить();

НоваяЗапись.РесурсПредприятия =СотрУд;

НоваяЗапись.Год = Год;

```
НоваяЗапись.ВремяНачала = Дата(2018,2,1);
НоваяЗапись.ВремяОкончания = Дата(2018,2,1);

НоваяЗапись.День = День;

НоваяЗапись.ЧасыРаботы = 0;
НоваяЗапись.ЧасовПерерывов = Дата(2018,2,1);
НаборЗаписей.Записать(Истина);
```

КонецЦикла;

НаборЗаписей.Записать(Истина);

КонецПроцедуры

&НаКлиенте

Процедура РаспровестиДокумент(Команда)

Если НЕ Объект.Проведен Тогда

Сообщить(«Нельзя отменить проведения у непроведенного документа!»);

Возврат;

КонецЕсли;

РаспределениеДока();

КонецПроцедуры

&НаКлиенте

Процедура ПериодРегистрацииНачалоВыбора(Элемент, ДанныеВыбора, СтандартнаяОбработка)

СтандартнаяОбработка = Ложь;

//пусть используют только регулировку

//ДатаКалендаряПриОткрытии = ?(ЗначениеЗаполнено(Объект.ПериодРегистрации),
Объект.ПериодРегистрации, УправлениеНебольшойФирмойПовтИсп.ПолучитьТекущуюДатаСеанса());

//ОткрытьФорму(«ОбщаяФорма.ФормаКалендаря»,
УправлениеНебольшойФирмойКлиент.ПолучитьПараметрыОткрытияФормыКалендаря(ДатаКалендаряПриОтк
рытии), ЭтаФорма);

КонецПроцедуры

&НаКлиенте

Процедура ПериодРегистрацииРегулирование(Элемент, Направление, СтандартнаяОбработка)

УправлениеНебольшойФирмойКлиент.ПриРегулированииПериодаРегистрации(ЭтаФорма,
Направление);

УправлениеНебольшойФирмойКлиент.ПриИзмененииПериодаРегистрации(ЭтаФорма);

//Доработка АТЭ 2019.04.03 Басс дни недели в заголовках(

Год = Год(Объект.ПериодРегистрации);

Месяц = Месяц(Объект.ПериодРегистрации);

СегоднДень = День(ТекущаяДата());

СегоднМесяц = Месяц(ТекущаяДата());

СегоднГод = Год(ТекущаяДата());

СтрокаТабличнойЧасти = Элементы.ТабЧасть.ТекущиеДанные;

Дата1 = НачалоМесяца(Объект.ПериодРегистрации);

Дата2 = КонецМесяца(Объект.ПериодРегистрации);

КолДней = Окр((Дата2 - Дата1) / (60 * 60 * 24));

Для т = 1 по КолДней Цикл

ДеньНедели = ДеньНедели(Дата(Год, Месяц, т));

Если ДеньНедели = 1 Тогда ДеньНедели = «пн»

ИначеЕсли ДеньНедели = 2 Тогда ДеньНедели = «вт»

ИначеЕсли ДеньНедели = 3 Тогда ДеньНедели = «ср»

ИначеЕсли ДеньНедели = 4 Тогда ДеньНедели = «чт»

ИначеЕсли ДеньНедели = 5 Тогда ДеньНедели = «пт»

ИначеЕсли ДеньНедели = 6 Тогда ДеньНедели = «сб»

ИначеЕсли ДеньНедели = 7 Тогда ДеньНедели = «вс»

КонецЕсли;

пер = «ТабЧастьДень» + т;

тТекст = Строка(т);

Элементы[пер].Заголовок = тТекст + « « + ДеньНедели;

Если Объект.ПериодРегистрации = НачалоМесяца(ТекущаяДата()) Тогда

Если СегоднДень > т Тогда

Элементы[пер].Доступность = Ложь;

Элементы[пер].ЦветФона = Новый Цвет(150,150,150);

Иначе

Элементы[пер].Доступность = Истина;

```

        Элементы[пер].ЦветФона = WEBЦвета.Белый;
    КонецЕсли;
ИначеЕсли Объект.ПериодРегистрации < НачалоМесяца(ТекущаяДата()) Тогда
    Элементы[пер].ЦветФона = WEBЦвета.Белый;
    Элементы[пер].Доступность = Ложь;
Иначе
    Элементы[пер].ЦветФона = WEBЦвета.Белый;
    Элементы[пер].Доступность = Истина;
КонецЕсли;

```

КонецЦикла;

КолДней = КолДней+1;

Если День(Дата2) > КолДней Тогда

```

    //Для т = КолДней по 31 Цикл
    //    Элементы[пер].Заголовок = ««;
    //    Элементы[пер].Доступность = Ложь;
    //    Элементы[пер].ЦветФона = Новый Цвет(150,150,150);
    //КонецЦикла;

```

КонецЕсли;

Объект.Год = Год;

//)

КонецПроцедуры

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

Если Объект.Ссылка.Пустая() Тогда

```

    Объект.ПериодРегистрации = НачалоМесяца(ТекущаяДата());

```

КонецЕсли;

```

    ОтображениеПериодаРегистрации = Формат(Объект.ПериодРегистрации, «ДФ='ММММ уууу'«);

```

КонецПроцедуры

&НаКлиенте

Процедура ТабЧастьСотрудникиНачалоВыбора(Элемент, ДанныеВыбора, СтандартнаяОбработка)

СтандартнаяОбработка = Ложь;

ПараметрыОткрытияФормы = Новый Структура;

СтруктураОтбора = Новый Структура;

СтруктураОтбора.Вставить(«Ссылка», АдресСпискаПодобранныхСотрудников());

ПараметрыОткрытияФормы.Вставить(«МножественныйВыбор», Ложь);

ПараметрыОткрытияФормы.Вставить(«ЗакрыватьПриВыборе», Истина);

ПараметрыОткрытияФормы.Вставить(«Отбор», СтруктураОтбора);

ОткрытьФорму(«Справочник.Сотрудники.Форма.ФормаВыбора», ПараметрыОткрытияФормы,
Элементы.ТабЧастьСотрудники);

КонецПроцедуры

&НаСервере

Функция АдресСпискаПодобранныхСотрудников()

Спис = Новый СписокЗначений;

Организация = Объект.Организация;

Год = Год(Объект.ПериодРегистрации);

Месяц = Месяц(объект.ПериодРегистрации);

День = 1;

Дата = Дата(Год,Месяц,День);

Запрос = Новый Запрос;

Запрос.Текст =

«ВЫБРАТЬ

| Влож.Сотрудник КАК Сотрудник

|ИЗ

| (ВЫБРАТЬ

| СотрудникиСрезПоследних.Сотрудник КАК Сотрудник

| ИЗ

| РегистрСведений.Сотрудники.СрезПоследних(&Дата1, Организация =
&Организация) КАК СотрудникиСрезПоследних

| ГДЕ

| СотрудникиСрезПоследних.ЗанимаемыхСтавок > 0

|

| ОБЪЕДИНИТЬ ВСЕ

|

| ВЫБРАТЬ

| Сотрудники.Сотрудник

| ИЗ

| РегистрСведений.Сотрудники КАК Сотрудники

```

|       ГДЕ
|
|           Сотрудники.ЗанимаемыхСтавок > 0
|
|           И Сотрудники.Период МЕЖДУ &Дата1 И &Дата2
|
|           И Сотрудники.Организация = &Организация) КАК Влож
|
|СГРУППИРОВАТЬ ПО
|
|       Влож.Сотрудник»;

```

```

Запрос.УстановитьПараметр(«Дата1», Дата);
Запрос.УстановитьПараметр(«Дата2», КонецМесяца(Дата));
Запрос.УстановитьПараметр(«Организация», Организация);
РезультатЗапроса = Запрос.Выполнить();

```

```

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

```

```

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
    Спис.Добавить(ВыборкаДетальныеЗаписи.Сотрудник);
КонецЦикла;

```

```

Возврат Спис;

```

КонецФункции

&НаКлиенте

Процедура ПриОткрытии(Отказ)

```

Если Объект.ПериодРегистрации < НачалоМесяца(Объект.ПериодРегистрации) Тогда

```

```

    ЭтаФорма.ТолькоПросмотр = Истина;

```

```

Иначе

```

```

    Дата1 = НачалоМесяца(Объект.ПериодРегистрации);

```

```

    Дата2 = КонецМесяца(Дата1);

```

```

    КолДней = Окp((Дата2 - Дата1) / (60 * 60 * 24));

```

```

    Сегодня = День(ТекущаяДата());

```

```

Для т = 1 по КолДней Цикл

```

```

    пер = «ТабЧастьДень» + т;

```

```

Если Сегодня > т Тогда

```

```

    Элементы[пер].Доступность = Ложь;

```

```

    Элементы[пер].ЦветФона = Новый Цвет(150,150,150);

```

```

Иначе

```

```

        Элементы[пер].Доступность = Истина;
    КонецЕсли;
КонецЦикла;
КонецЕсли;
    ПрименитьУсловноеФорматирование();
КонецПроцедуры

&НаСервере
// Устанавливает условное форматирование.
//
// Параметры:
// ЦветТекста - Цвет - формат цвета текста;
// ПутьКДанным - Строка - путь к данным на форме;
// ПолеКомп - Строка - поле компоновки на форме;
// Значение - Произвольный - значение для отбора;
// Сравнение - ВидСравнения - вид сравнения.
//
//
Процедура УстановитьНастройкиУсловногоФормирования(ЦветТекста, ПутьКДанным, ПолеКомп,
Значение, Сравнение)
    ЭлементОформления = УсловноеОформление.Элементы.Добавить();
    ЭлементОформления.Использование = Истина;
    ЭлементОформления.Оформление.УстановитьЗначениеПараметра(«ЦветТекста», ЦветТекста);

    ЭлементУсловия = ЭлементОформления.Отбор.Элементы.Добавить(Тип(«ЭлементОтбораКомпоновкиДанных»));
    ЭлементУсловия.ЛевоеЗначение = Новый ПолеКомпоновкиДанных(«Объект.» + ПутьКДанным);
    ЭлементУсловия.ПравоеЗначение = Значение;
    ЭлементУсловия.ВидСравнения = Сравнение;
    ЭлементУсловия.Использование = Истина;

    ОформляемоеПоле = ЭлементОформления.Поля.Элементы.Добавить();
    ОформляемоеПоле.Поле = Новый ПолеКомпоновкиДанных(ПолеКомп);
КонецПроцедуры

&НаКлиенте
Процедура ЗаписатьТабельВыделенногоСотрудника(Команда)
    Дата1 = НачалоМесяца(Объект.ПериодРегистрации);
    Дата2 = КонецМесяца(Дата1);
    КолДней = Окр((Дата2 - Дата1) / (60 * 60 * 24));

```

```

СегоднДень = День(ТекущаяДата());
ТекСтрока = Элементы.ТабЧасть.ТекущаяСтрока;
НовСтр = Объект.ТабЧасть[ТекСтрока];
Для т=1 по КолДней Цикл
    пер =«День» + т;
        Если СегоднДень <= т Тогда
            НовСтр[пер] = 8;
        КонецЕсли;
    КонецЦикла
КонецПроцедуры

&НаСервере
Процедура ПрименитьУсловноеФорматирование()
    Если Объект.ПериодРегистрации < НачалоМесяца(Объект.ПериодРегистрации) Тогда
        ЭтаФорма.ТолькоПросмотр = Истина;
    Иначе
        Дата1 = НачалоМесяца(Объект.ПериодРегистрации);
        Дата2 = КонецМесяца(Дата1);
        КолДней = Окр((Дата2 - Дата1) / (60 * 60 * 24));
        СегоднДень = День(ТекущаяДата());

        Для т = 1 по КолДней Цикл
            пер = «ТабЧастьДень» + т;
            ЭлементДеньПутьКДанным = СтрЗаменить(Элементы[пер].Имя, «ТабЧасть»,
«ТабЧасть.»);

            //Доработка АТЭ ШенгелияДЮ 2019.04.18
            // Применяем условное форматирование для каждого дня в таблице.
            // В данном случае: если значение на день в таблице равно 3, то окрашиваем
ячейку в красный цвет.

            УстановитьНастройкиУсловногоФормирования(Новый          Цвет(255,0,0),
ЭлементДеньПутьКДанным, пер,

        3, ВидСравненияКомпоновкиДанных.Равно);
            КонецЦикла;
        КонецЕсли;
    КонецПроцедуры

```