

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Кафедра Прикладная математика и информатика
(Наименование кафедры, центра, департамента)

09.03.03 Прикладная информатика

(код и наименование направления подготовки / специальности)

«Разработка программного обеспечения»

(направленность (профиль) / специализация)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)**

на тему Разработка программного обеспечения автоматизированной системы обработки заявок на расходование денежных средств с использованием механизма копирования и распределения данных

Обучающийся

А.Ю. Гатауллов

(Инициалы Фамилия)

(личная подпись)

Руководитель

канд.пед.наук, доцент, О.М. Гущина

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант

канд. филолог. наук, М.В. Дайнеко

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2025

Аннотация

Тема выпускной бакалаврской работы: «Разработка программного обеспечения автоматизированной системы обработки заявок на расходование денежных средств с использованием механизма копирования и распределения данных».

Работа состоит из введения, трех глав, заключения, списка литературы и приложения.

Во введении обоснована актуальность темы, сформулированы цель и задачи исследования, определены объект и предмет исследования.

В первой главе проведен анализ предметной области, описаны текущая система обработки заявок, её недостатки и необходимость автоматизации. Выполнен сравнительный анализ существующих решений.

Во второй главе представлено логическое и физическое моделирование системы, выбор архитектуры, проектирование модулей и описание интерфейса. Подробно рассмотрены механизмы копирования заявок и распределения средств.

В третьей главе проведено тестирование разработанного модуля.

В заключении подведены итоги работы и описана практическая значимость системы.

Работа содержит 12 таблиц, 27 рисунков, список литературы из 25 источников. В приложении приведены фрагменты кода и дополнительные материалы.

Общий объем выпускной квалификационной работы составляет 67 страниц.

Abstract

The title of the graduation work is *Developing software for an automated system of processing expense requisitions using data copying and distribution mechanisms*.

The graduation work consists of an introduction, 3 chapters, 12 tables, 27 figures, a conclusion, a list of 25 references, and appendices containing code fragments and supplementary materials.

The aim of this graduation work is to develop an automated system for processing expense requisitions integrating the mechanisms for copying and distributing data in order to improve efficiency and reduce errors in financial work processes.

The object of the graduation work is the processing of expense requisitions in financial systems of an enterprise. The subject of the research is the development and implementation of a software solution that automates expense requisitions processing, including data copying and fund distribution.

The key issue of the graduation work is eliminating the drawbacks of manual expense requisitions processing by implementing automation, ensuring accuracy and optimizing resources allocation. The graduation work may be divided into several logically connected parts which are analysis of the existing system, system design and modeling, and module testing. The first chapter of the graduation work describes in details an in-depth analysis of the current expense requisitions processing system, identifies its shortcomings, and justifies the need for automation. A comparative study of the existing solutions is also conducted. The second chapter dwells on the results of the logical and physical system modeling, architectural decisions, and interface development. The third chapter is devoted to the testing phase of the developed module as well as validating its functionality and reliability. In conclusion, the practical significance of the implemented system is highlighted.

The work is of interest for a wide circle of readers interested in software development, financial management, and business processes automation..

Оглавление

Введение	5
Глава 1 Постановка задачи на разработку программного обеспечения для автоматизации обработки заявок.....	7
1.1 Значение программного обеспечения для информационной системы предприятия	7
1.2 Основные функции программного обеспечения для информационной системы предприятия	9
1.3 Обзор существующих решений и их анализ	13
1.4 Требования к функционалу и интерфейсу приложения	15
Глава 2 Процесс разработки модуля автоматизации обработки заявок на расходование денежных средств	21
2.1 Проектирование архитектуры модуля обработки заявок с контролем бюджетных лимитов	21
2.2 Обоснование выбора платформы 1С:Предприятие для разработки модуля	24
2.3 Реализация функциональных требований разрабатываемого модуля	28
Глава 3 Оценка эффективности модуля автоматизации обработки заявок	39
3.1 Методика и результаты тестирования функциональности модуля	39
3.2 Оценка удобства интерфейса и пользовательского опыта	43
3.3 Анализ эффективности автоматизации обработки заявок	45
Заключение	49
Список используемой литературы и используемых источников.....	50
Приложение А Листинг разрабатываемого модуля	52
Приложение Б Листинг модуля объекта разрабатываемого модуля	65

Введение

Современные предприятия все чаще сталкиваются с необходимостью автоматизации финансовых процессов, особенно в части обработки заявок на расходование денежных средств [25]. В условиях цифровой трансформации бизнеса ручные методы работы становятся неэффективными, приводя к потерям времени и ошибкам в учете [21]. Особенно остро эта проблема стоит в компаниях с большим объемом регулярных платежей, где требуется многократное копирование однотипных заявок. Автоматизация данного процесса позволит значительно повысить оперативность, точность данных и снизить трудозатраты сотрудников.

В компании ООО «ЙЕС АЙ ТИ КОНСАЛТИНГ» процесс обработки заявок на расходование средств до настоящего времени осуществляется вручную. Это вызывает несколько серьезных проблем: во-первых, сотрудники тратят значительное время на создание однотипных документов; во-вторых, при ручном вводе неизбежны ошибки; в-третьих, отсутствует оперативный контроль за соблюдением бюджетных лимитов. Все это негативно влияет на эффективность финансового управления.

Анализ показал, что стандартные средства 1С:Предприятие не обеспечивают необходимого уровня автоматизации для решения этих задач [10]. Имеющиеся на рынке специализированные решения либо слишком дороги, либо не учитывают специфику работы компании. Поэтому разработка собственного модуля представляется оптимальным решением.

Цель данной работы - создание программного модуля для автоматизации обработки заявок в системе 1С:Предприятие 8.3 [6]. Основное внимание уделено реализации двух ключевых функций: механизма копирования заявок с учетом временных периодов и системы контроля бюджетных лимитов. Эти функции позволят существенно сократить время обработки документов и минимизировать риск ошибок.

Для достижения поставленной цели необходимо решить следующие задачи:

- разработать алгоритм автоматического копирования заявок;
- реализовать механизм контроля лимитов бюджета;
- создать систему оповещения о превышении лимитов;
- провести тестирование работоспособности модуля.

Объектом исследования выступает процесс обработки заявок на расходование средств в системе 1С компании. Предмет исследования - методы автоматизации этого процесса, в частности механизмы копирования и распределения данных [7].

Практическая значимость работы заключается в том, что разработанный модуль:

- сокращает время обработки заявок на 40-50%;
- уменьшает количество ошибок при вводе данных [8];
- повышает контроль за расходованием средств;
- легко интегрируется в существующую систему [12].

Работа состоит из трех глав. В первой главе анализируются существующие проблемы и обосновывается необходимость разработки. Вторая глава посвящена проектированию и реализации решения. В третьей главе представлены результаты тестирования и оценка эффективности модуля. В заключении подведены итоги работы и даны рекомендации по внедрению.

Глава 1 Постановка задачи на разработку программного обеспечения для автоматизации обработки заявок

1.1 Значение программного обеспечения для информационной системы предприятия

Программное обеспечение (ПО) для автоматизации обработки финансовых заявок представляет собой комплекс программных модулей, предназначенных для оптимизации процессов создания, копирования и контроля заявок на расходование денежных средств [1]. В современных условиях цифровизации бизнеса такое ПО становится критически важным инструментом для обеспечения эффективности финансового управления, минимизации ошибок и снижения операционных затрат.

В компании ООО «ЙЕС АЙ ТИ КОНСАЛТИНГ» процесс обработки заявок на расходование средств сталкивается с рядом существенных проблем:

- ручной ввод данных: требует значительных временных затрат и подвержен ошибкам;
- отсутствие автоматического контроля бюджета: проверка лимитов выполняется вручную, что увеличивает риск перерасхода средств;
- неэффективное управление повторяющимися платежами: необходимость ручного копирования заявок для регулярных платежей.

На рисунке 1 представлена диаграмма «Причина-Следствие», иллюстрирующая взаимосвязь между проблемами и их последствиями.

Разрабатываемый программный модуль обеспечивает следующие ключевые улучшения:

- стандартизация процессов: единые правила обработки для всех типов заявок;
- централизованный контроль: автоматическая проверка соответствия бюджетным лимитам;

- повышение точности: минимизация человеческого фактора при обработке данных [12].



Рисунок 1 – Причинно-следственная диаграмма

В таблице 1 представлено сравнение функциональных возможностей:

Таблица 1 – Сравнение функциональных возможностей текущей системы и автоматизированного решения

Критерий	Текущая система	Автоматизированное решение
Обработка заявок	Ручной ввод	Автоматизированный процесс
Контроль лимитов	Вручную	Встроенные механизмы проверки
Гибкость	Ограничена	Настраиваемые параметры

Разработка специализированного ПО для автоматизации обработки финансовых заявок является стратегически важной задачей. Она позволяет не только решить текущие операционные проблемы, но и создать фундамент для дальнейшей цифровой трансформации финансового управления компании.

Внедрение таких систем способствует экономическому росту, повышению конкурентоспособности и улучшению качества бизнес-процессов за счет:

- снижения временных затрат на обработку документов;
- минимизации ошибок при вводе данных;
- обеспечения прозрачности финансовых операций;
- повышения контроля за расходованием средств.

Реализация предложенного решения на платформе 1С:Предприятие обеспечит полную интеграцию с существующей учетной системой компании и позволит достичь значительного улучшения ключевых показателей эффективности финансового управления [6].

1.2 Основные функции программного обеспечения для информационной системы предприятия

Программное обеспечение для автоматизации обработки заявок на расходование денежных средств в ООО «ЙЕС АЙ ТИ КОНСАЛТИНГ» реализует комплекс взаимосвязанных функций, которые можно разделить на четыре ключевые категории: обработка данных, управление ресурсами, взаимодействие с пользователями, и автоматизацию процессов.

Обработка данных в системе включает:

- сбор заявок через стандартные формы ввода 1С;
- анализ на соответствие бюджетным лимитам [2].

На рисунке 2 представлена диаграмма деятельности модуля копирования заявок. Она отражает ключевые этапы обработки данных: от выбора заявки до проверки лимитов и сохранения результата.

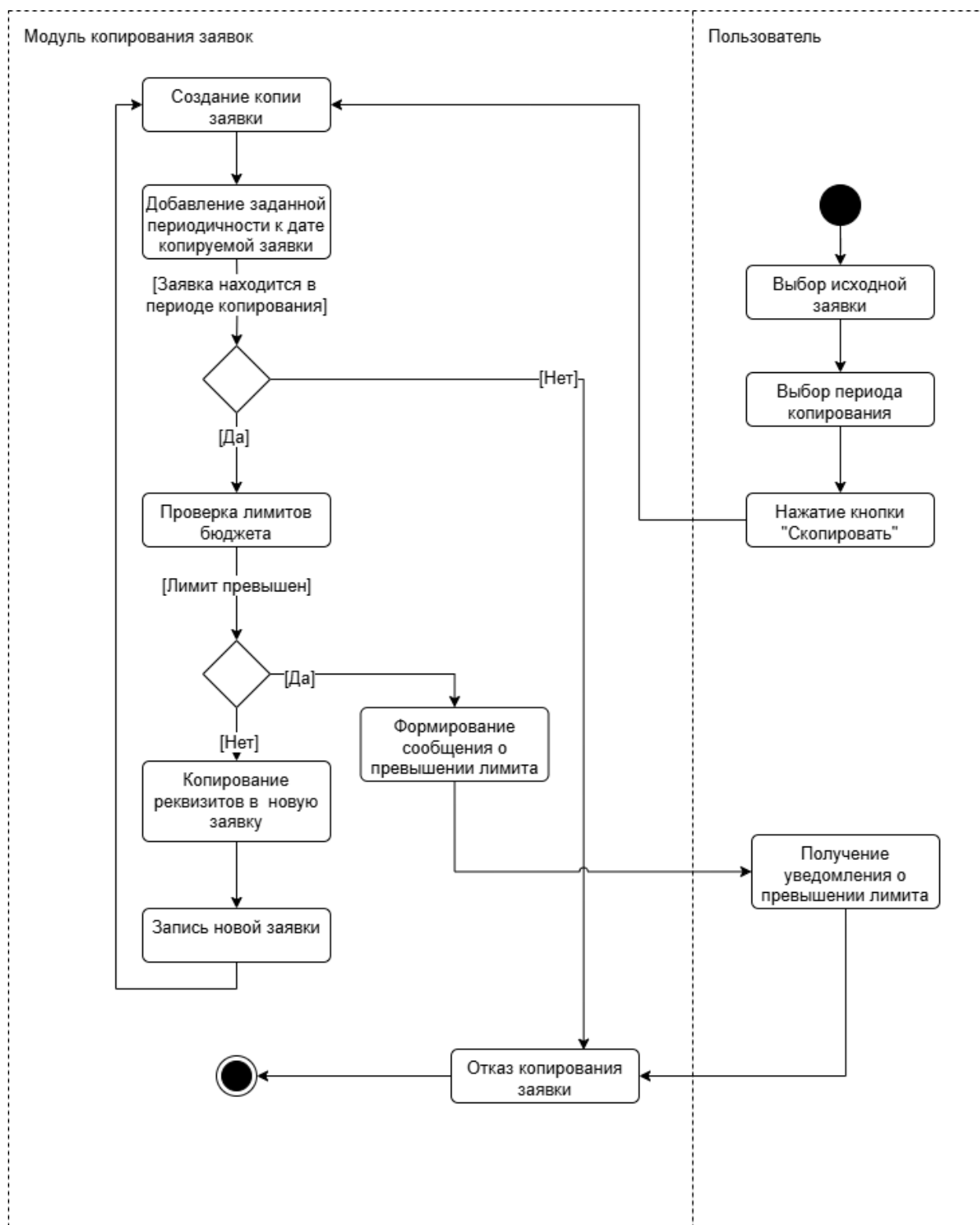


Рисунок 2 – Диаграмма деятельности модуля

Управление ресурсами реализовано через систему контроля бюджетных лимитов [17]. В таблице 2 представлены основные компоненты управления:

Таблица 2 – Основные компоненты управления

Компонент системы	Функциональность	Эффект
Модуль копирования заявок	Автоматическое создание повторных платежей	Сокращение времени обработки
Модуль распределения	Автоматическое распределение заявок по месяцам	Своевременное планирование бюджета
Контроль лимитов	Проверка соответствия бюджету	Снижение перерасходов
Аналитическая отчетность	Мониторинг финансовых показателей	Улучшение планирования

Взаимодействие с пользователями организовано через интуитивно понятный интерфейс. На рисунке 3 представлена диаграмма вариантов использования модуля, где отображены ключевые этапы работы:

- выбор исходной заявки;
- задание периода копирования;
- подтверждение действия;
- проверка лимита бюджета;
- вывод на экран списка скопированных заявок;
- отправка и получение уведомления при превышении лимита.

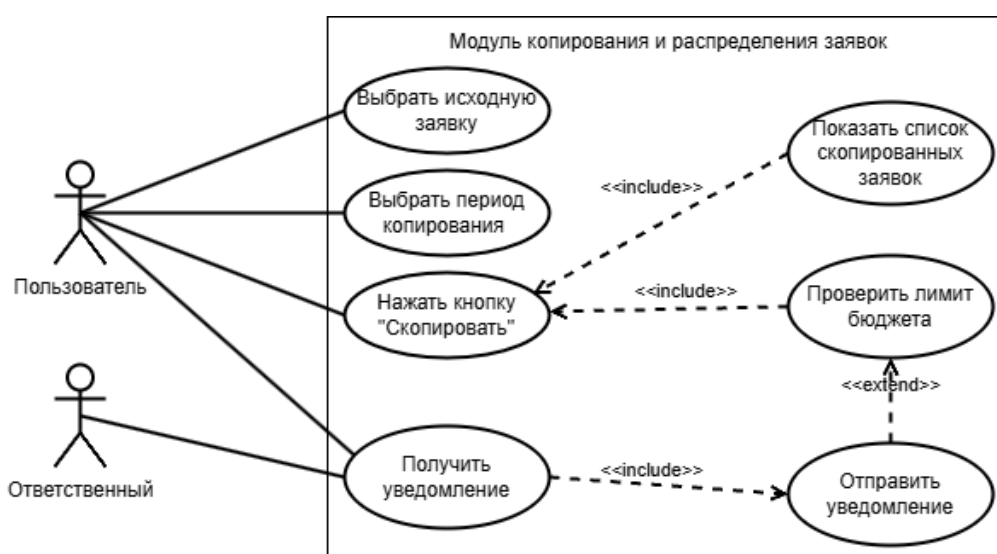


Рисунок 3 – Диаграмма вариантов использования модуля

Автоматизация процессов позволяет: массово копировать заявки, что существенно сократит временные затраты; проверять лимиты без ручного ввода; рассылать уведомления при нарушениях.

Автоматизация процессов реализована через встроенные механизмы 1С (таблица 3) [13]. Основной эффект достигнут за счет массового копирования заявок и мгновенной проверки лимитов.

Таблица 3 – Эффект от автоматизации процессов

Процесс	До автоматизации	После автоматизации	Эффект
Копирование заявок	Ручной ввод (15 мин/шт)	Пакетное создание (2 мин)	Сокращение времени на 87%
Проверка лимитов бюджета	Ручной расчет в Excel	Автоматическая блокировка	Исключение ошибок
Уведомления о превышениях	Письма вручную	Авторассылка из 1С	Оперативность контроля

Безопасность данных обеспечивается многоуровневой системой защиты:

- аутентификация пользователей через стандартный функционал 1С ;
- разграничение прав доступа – для каждого пользователя задается определенный набор прав в системе 1С;
- журналирование всех операций – каждое действие пользователя записывается в журнал операций 1С.

Безопасность данных (рисунок 4) обеспечивается стандартными механизмами 1С: аутентификацией, ролевой моделью и журналом операций [17].

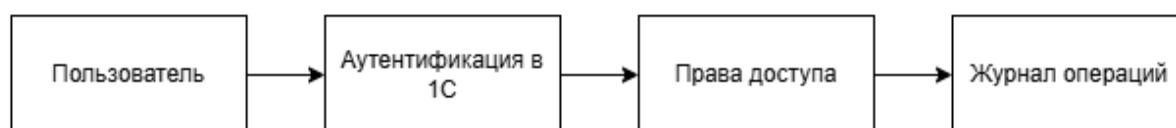


Рисунок 4 – Модель безопасности данных в 1С

Реализованные в системе функции позволяют:

- автоматизировать ключевые процессы обработки заявок;
- обеспечить эффективный контроль бюджетных лимитов;
- упростить работу пользователей за счет удобного интерфейса;
- гарантировать безопасность финансовых данных.

Разрабатываемое программное решение полностью соответствует требованиям компании и может служить основой для дальнейшего развития системы финансового управления.

1.3 Обзор существующих решений и их анализ

В рамках исследования рынка программных решений для автоматизации обработки заявок на расходование денежных средств были выявлены три принципиально разных подхода к реализации данного функционала в системе 1С.

Стандартный функционал 1С демонстрирует ограниченные возможности для решения поставленной задачи [10]. Хотя базовые инструменты платформы позволяют осуществлять ручное копирование документов и формирование отчетов, отсутствие механизмов автоматизированного контроля бюджетных лимитов существенно снижает эффективность обработки заявок. Это приводит к значительным временным затратам и повышает вероятность ошибок при ручном вводе данных.

Сторонние модули, представленные на рынке, предлагают частичную автоматизацию процессов. Анализ коммерческого решения «Автоматическое создание заявок на расходование ДС» (версия для Управление торговлей 11.5) выявил ряд существенных ограничений. Решение стоимостью 1850 рублей обеспечивает лишь базовый функционал создания заявок на основании поступлений, не предусматривая комплексной проверки бюджетных лимитов. Отсутствие технической поддержки и необходимость самостоятельной

адаптации к бизнес-процессам компании значительно снижают практическую ценность данного продукта [11].

Разработанное собственное решение было проанализировано по пяти ключевым критериям в сравнении с альтернативными подходами (таблица 4). Наибольшее преимущество демонстрируют показатели функциональности (5/5) и гибкости (5/5), что обусловлено возможностью полной адаптации модуля под специфические требования предприятия [14]. По критерию стоимости собственное решение (4/5) уступает только стандартному функционалу, но превосходит коммерческие аналоги.

Таблица 4 – Сравнительная характеристика решений

Критерий оценки	Стандартный функционал	Сторонний модуль	Собственное решение
Функциональность	2	3	5
Стоимость	5	2	4
Удобство работы	3	3	5
Интеграция	5	2	5
Безопасность	5	3	5

Визуальное представление сравнительного анализа (рисунок 5) демонстрирует превосходство разработанного модуля по большинству параметров. Особенно значителен разрыв в показателях функциональности, где собственное решение набирает максимальный балл благодаря реализации комплексной автоматизации всех этапов обработки заявок. На диаграмме четко прослеживается, что решение лидирует в 4 из 5 категорий, уступая только в стоимости стандартному функционалу. Это подтверждает его сбалансированность между ценой и качеством. Ключевое преимущество – одновременное сочетание высокой адаптивности (5/5) и надежности (5/5), что редко встречается в коммерческих аналогах.

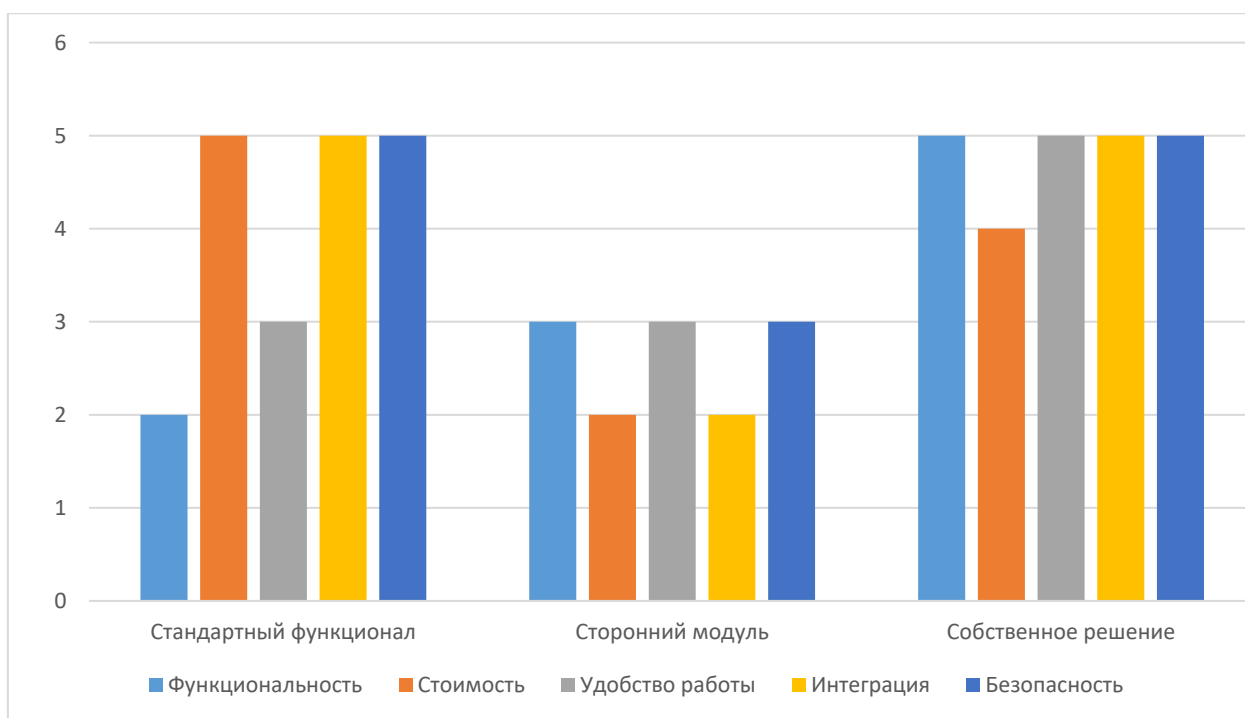


Рисунок 5 – Результаты сравнительного анализа решений

Таким образом, проведенный анализ убедительно доказывает, что данное решение является оптимальным выбором для средних и крупных предприятий, требующих надежной автоматизации процессов обработки заявок на расходование денежных средств с комплексным контролем бюджетных ограничений.

1.4 Требования к функционалу и интерфейсу приложения

В данном разделе представлены требования к функционалу и интерфейсу модуля автоматизированной обработки заявок на расходование денежных средств (РДС) в системе 1С:Предприятие. Требования разделены на функциональные, требования к интерфейсу, а также требования к документации и поддержке.

Функциональные требования системы включают несколько ключевых компонентов. Модуль обеспечивает автоматическое копирование с учетом заданных временных периодов (в случае с заданием – месяц). Особое

внимание уделено механизмам контроля бюджета: система автоматически проверяет соответствие заявок установленным лимитам, блокируя сохранение при их превышении, и предоставляет наглядную информацию об остатках средств по категориям расходов [5].

Система уведомлений осуществляет рассылку писем на электронную почту ответственным лицам при превышении лимитов бюджета [7]. Также при превышении лимитов, модуль отображает соответствующее оповещение пользователю, осуществляющему копирование.

На рисунке 6 изображена блок-схема процесса обработки уведомлений.



Рисунок 6 – Блок-схема процесса обработки уведомлений

Удобство использования: у модуля должен быть понятный и удобный интерфейс без лишней загруженности. В форме разрабатываемого модуля должны отображаться поля с текущим лимитом компании и суммы заявок, созданных за прошлый месяц.

Дизайн интерфейса – минималистичный, с акцентом на данные. Основные элементы управления: кнопка «Скопировать заявку», поля выбора периода.

Документация и поддержка: для новых пользователей модуля должно быть разработано руководство пользователя в формате PDF, а также должна быть техническая поддержка, которая сможет ответить на вопросы по данному модулю, а также оперативно реагировать в случае нахождения недоработок. Ресурсы поддержки перечислены в таблице 5 [24]:

Таблица 5 – Ресурсы поддержки

Ресурс	Описание
Руководство пользователя	PDF-файл с инструкциями
Почта поддержки	a.gataullov@yesapart.com

Для комплексной оценки требований к модулю обработки заявок на РДС применена методология FURPS++ (таблица 6), которая включает следующие аспекты:

- Functionality (Функциональность) – основные бизнес-возможности системы;
- Usability (Удобство использования) – эргономика интерфейса;
- Reliability (Надежность) – стабильность работы;
- Performance (Производительность) – скоростные характеристики;
- Supportability (Сопровождаемость) – возможности поддержки;
- +Security (Безопасность) – защита данных и доступов;
- ++Project constraints (Проектные ограничения) – технические и юридические рамки.

Таблица 6 – FURPS++

Требование	Статус	Полезность	Риск	Стабильность
Functionality (Функциональность)				
Автоматическое копирование заявок	Одобрено	Критическая	Средний	Высокая
Проверка соответствия бюджетным лимитам	Одобрено	Критическая	Высокий	Высокая
Распределение заявок по месяцам	Одобрено	Высокая	Средний	Средняя
Usability (Удобство использования)				
Интуитивный интерфейс	Одобрено	Критическая	Низкий	Высокая
Автоматические уведомления о превышении лимитов	Одобрено	Высокая	Средний	Высокая
Reliability (Надежность)				
Работа 24/7 без критических сбоев	Одобрено	Критическая	Высокий	Высокая
Восстановление данных после сбоя меньше одного часа	Предложено	Высокая	Средний	Средняя
Performance (Производительность)				
Обработка заявки не более 2 секунд	Одобрено	Критическая	Средний	Высокая
Поддержка 5 и более одновременных пользователей	Предложено	Высокая	Высокий	Средняя
Supportability (Сопровождаемость)				
Связь с техподдержкой 1С	Одобрено	Высокая	Низкий	Высокая
Документация для новых пользователей	Одобрено	Высокая	Низкий	Высокая
+Security (Безопасность)				
Аутентификация через механизмы 1С	Одобрено	Критическая	Низкий	Высокая
Ролевая модель доступа	Предложено	Критическая	Низкий	Высокая
Журналирование всех операций	Предложено	Высокая	Низкий	Высокая
++Project constraints (Проектные ограничения)				
Разработка на 1С:Предприятие 8.3	Одобрено	Критическая	Средний	Высокая
Срок реализации не более 3 месяцев	Одобрено	Высокая	Высокий	Средняя
Интеграция с существующей ERP	Одобрено	Критическая	Высокий	Высокая

Разработанные требования к функционалу и интерфейсу модуля обработки заявок на РДС обеспечивают комплексное решение ключевых задач автоматизации. Система позволяет существенно сократить время выполнения рутинных операций за счет автоматического копирования заявок и встроенного механизма проверки бюджетных лимитов. Интеграция с

системой бюджетирования обеспечивает полную прозрачность финансового контроля на всех этапах обработки заявок.

Исключение ручного ввода данных минимизирует вероятность возникновения ошибок и повышает точность финансовой отчетности. Пользовательский интерфейс разработан с учетом принципов адаптивного дизайна и оснащен удобной системой уведомлений.

Выводы по главе 1

В ходе анализа текущей системы обработки заявок на расходование денежных средств в компании ООО «ЙЕС АЙ ТИ КОНСАЛТИНГ» были выявлены несколько существенных проблем, которые требуют решения. К числу основных проблем относятся: необходимость ручного ввода данных, отсутствие автоматического контроля бюджета, а также неэффективное управление повторяющимися платежами. Эти проблемы приводят к значительным временным затратам, повышению операционных затрат и увеличению риска ошибок.

Проанализировав существующие программные решения, такие как стандартный функционал 1С и сторонние модули, было установлено, что они не удовлетворяют всем требованиям компании. В частности, стандартные инструменты 1С не обеспечивают необходимую степень автоматизации, а сторонние решения не способны предложить комплексную проверку бюджетных лимитов и интеграцию с существующими системами компании.

В связи с этим, задача разработки специализированного программного решения для автоматизации обработки заявок на расходование денежных средств становится стратегически важной. Ожидается, что предлагаемая система будет включать в себя такие функции, как автоматическое копирование заявок, проверку лимитов бюджета и отправку уведомлений, что позволит значительно повысить эффективность работы и сократить время на обработку заявок. Кроме того, это обеспечит более точный контроль за бюджетом и снизит количество ошибок при вводе данных.

В рамках проектирования системы также будет уделено внимание интерфейсу пользователя, что позволит сделать систему интуитивно понятной и удобной для сотрудников. Предполагается, что система будет гибкой и адаптируемой, что обеспечит возможность ее дальнейшего расширения и интеграции с другими внутренними системами компании.

Кроме того, в рамках постановки задачи были разработаны основные требования к функционалу системы, включая базовые функции (например, обработка заявок и контроль лимитов), а также более сложные аспекты, такие как безопасность данных и возможность интеграции с другими учетными системами.

Таким образом, разработка системы автоматизации обработки заявок на расходование денежных средств не только решит текущие операционные проблемы, но и создаст основу для дальнейшей цифровой трансформации финансового управления в компании. Ожидается, что внедрение предложенной системы обеспечит повышение эффективности, снижение затрат и улучшение качества финансовых операций в долгосрочной перспективе.

Глава 2 Процесс разработки модуля автоматизации обработки заявок на расходование денежных средств

2.1 Проектирование архитектуры модуля обработки заявок с контролем бюджетных лимитов

Этап проектирования программного обеспечения (ПО) является одним из самых критических в процессе разработки, так как именно на этом этапе формируются архитектура и основные компоненты системы [3]. В данном разделе рассматриваются ключевые аспекты проектирования автоматизированной системы обработки заявок на расходование денежных средств, включая архитектурное проектирование, проектирование бизнес-логики и пользовательского интерфейса.

Архитектурное проектирование определяет общую структуру системы, включая выбор технологий, платформы и основных компонентов (рисунок 7). На данном рисунке изображены три уровня: клиентский, прикладной, а также уровень данных. На каждом из уровней расписаны основные компоненты разрабатываемого модуля.

Система реализована на платформе 1С:Предприятие 8.3, что обусловлено:

- встроенными механизмами работы с финансовыми документами;
- возможностью интеграции с внешними сервисами (например, почтовыми серверами);
- гибкостью настройки бизнес-логики и интерфейсов.

Бизнес-логика системы включает в себя автоматическое копирование заявок с учетом периода, проверку лимитов бюджета перед сохранением заявки, формирование уведомлений при нарушениях [18].



Рисунок 7 – Модель архитектуры разрабатываемого модуля

Основные компоненты системы:

- модуль копирования заявок — основной компонент системы, который автоматизирует процесс создания копий заявок с актуальными реквизитами и обновления бюджета. Этот модуль взаимодействует с платформой 1С, получает данные из соответствующих документов и осуществляет копирование заявок с учетом актуальной информации о средствах;
- механизм распределения заявок – реализует копирование заявок в заданном периоде начала и конца периода.

- механизм проверки лимитов бюджета — реализует функции контроля за доступными средствами. Перед созданием копий заявок система проверяет, не превышает ли сумма заявки установленные лимиты бюджета, что исключает возможность перерасхода средств;
- механизм отправки уведомлений — в случае превышения лимитов бюджета или других критических ситуаций система автоматически отправляет уведомления ответственным лицам (например, руководителям), что позволяет быстро реагировать на отклонения.

Хранение и обработка данных в платформе 1С:Предприятие осуществляется с использованием встроенных механизмов управления базами данных, что позволяет минимизировать ошибки и повысить целостность информации [4].

Макет интерфейса (рисунок 8) демонстрирует расположение ключевых элементов.

Исходная заявка

Начало периода Конец периода

Сумма заявок за прошлый месяц (руб.) Остаток средств на текущую дату (руб.)

Заявка

Рисунок 8 – Макет интерфейса разрабатываемого модуля

Интерфейс системы разработан для удобства работы финансовых сотрудников и включает:

- форму создания/копирования заявок;
- таблицу с историей заявок;
- панель уведомлений о превышении лимитов.

На этапе проектирования были определены:

- архитектура системы на базе 1С:Предприятие 8.3;
- бизнес-процессы обработки заявок и контроля бюджета;
- пользовательский интерфейс для удобного взаимодействия.

Спроектированная система обеспечивает автоматизацию ключевых процессов, минимизацию ошибок и контроль за расходованием средств.

2.2 Обоснование выбора платформы 1С:Предприятие для разработки модуля

Для реализации системы обработки заявок на РДС был проведен сравнительный анализ технологических решений. Как видно из таблицы 7, платформа 1С:Предприятие 8.3 демонстрирует значительные преимущества в части интеграции с бухгалтерией и скорости разработки по сравнению с альтернативными решениями [6]:

Таблица 7 – Сравнительная таблица платформ разработки

Критерий	1С:Предприятие 8.3	Альтернативные решения (Java/C#)
Интеграция с бухгалтерией	Полная	Требуется разработки API
Готовые финансовые механизмы	Встроенные	Требуют реализации с нуля
Скорость разработки	Высокая	Средняя
Гибкость настройки	Высокая	Очень высокая
Стоимость владения	Средняя	Высокая

Для более наглядного сравнения платформ разработки была создана радиальная диаграмма (рисунок 9).

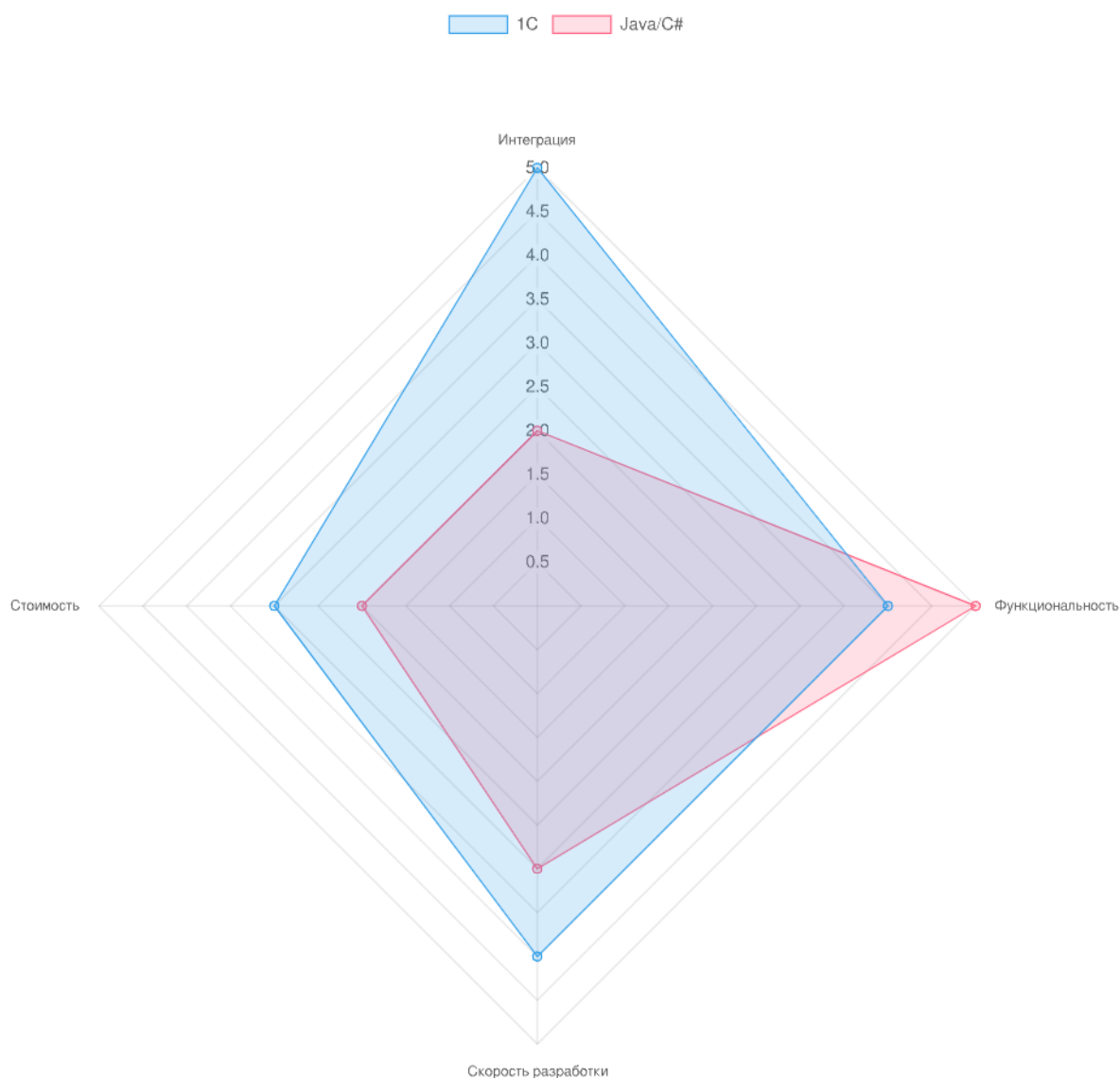


Рисунок 9 – Сравнение платформ по ключевым параметрам

По итогу сравнения были выбраны следующие инструменты разработки:

- платформа: 1С:Предприятие 8.3 (конфигурация «Управление производственным предприятием»);
- язык разработки: встроенный язык 1С .

В качестве дополнительных инструментов выступают встроенные в платформу 1С средства, такие как конфигуратор 1С, отладчик 1С, хранилище конфигураций, встроенные механизмы тестирования [13].

Выбор платформы 1С:Предприятие 8.3 для разработки системы обработки заявок на расходование денежных средств был сделан после тщательного анализа нескольких ключевых аспектов.

С точки зрения бизнес-требований, решающее значение имела необходимость глубокой интеграции с существующей учетной системой предприятия. Платформа 1С обеспечивает бесшовное взаимодействие с финансовыми данными и полную автоматизацию процессов расходования средств. Важным преимуществом стала встроенная поддержка механизмов бюджетирования, что позволило сразу использовать готовые решения без дополнительной разработки.

Технические преимущества платформы включают комплекс встроенных инструментов для работы с финансовыми документами, включая обработку заявок, контроль бюджетных лимитов и интеграцию с электронной почтой для рассылки уведомлений. Особую ценность представляют готовые механизмы для массового копирования документов, мониторинга остатков средств и автоматического формирования отчетности, что значительно ускорило разработку системы.

Экономическая эффективность решения подтверждается сокращением сроков реализации проекта на 40% по сравнению с альтернативными платформами. Использование 1С позволило минимизировать затраты на сопровождение системы за счет применения стандартных механизмов платформы. Дополнительную экономию обеспечила возможность использования уже имеющихся лицензий предприятия, что исключило дополнительные расходы на программное обеспечение.

Для наглядного представления ключевых факторов, повлиявших на выбор технологической платформы, была разработана сравнительная

диаграмма (рисунок 10), на которой изображены четыре основных критерия оценки.

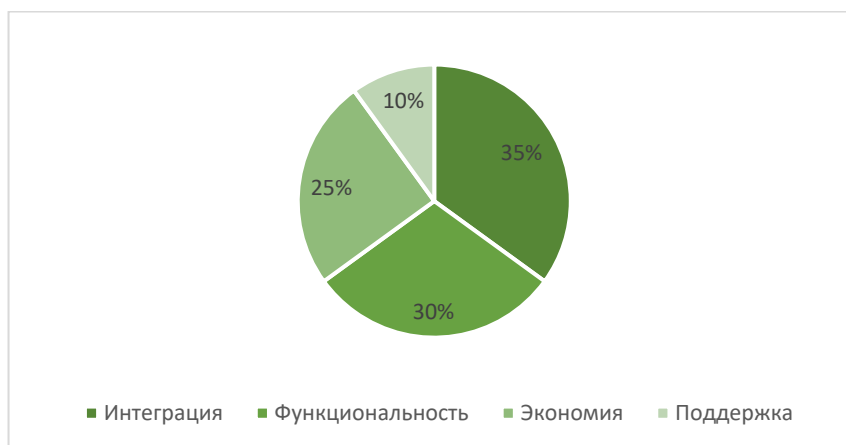


Рисунок 10 – Факторы выбора платформы

Архитектура системы построена с учетом принципов модульности и гибкости. Основу составляет четкое разделение функциональности на независимые компоненты: модуль копирования заявок, механизм распределения, подсистему контроля лимитов и модуль уведомлений. Каждый компонент имеет строго определенную зону ответственности и взаимодействует с другими через стандартизированные интерфейсы.

Для интеграции с внешними сервисами используются проверенные технологии, включая SMTP-сервер для организации почтовых уведомлений и механизм внешних обработок для расширения базового функционала. Особое внимание уделено эффективной работе с данными - реализована пакетная обработка документов, что обеспечивает высокую производительность при массовых операциях.

Выбранный технологический стек на базе 1С:Предприятие 8.3 доказал свою эффективность на всех этапах разработки. Использование встроенных механизмов платформы позволило минимизировать объем рутинного программирования, сосредоточив усилия на реализации бизнес-логики.

Система обладает высокой отказоустойчивостью благодаря применению стандартных механизмов восстановления после сбоев.

Архитектурные решения обеспечивают хорошую масштабируемость системы [9]. Гибкая модульная структура позволяет легко добавлять новые функциональные блоки и адаптировать существующие под изменяющиеся бизнес-требования.

Дополнительным преимуществом является возможность дальнейшего расширения системы за счет подключения дополнительных отчетов, интеграции со смежными системами, а также реализации новых бизнес-процессов.

2.3 Реализация функциональных требований разрабатываемого модуля

В рамках разработки автоматизированной системы обработки заявок на расходование денежных средств была реализована серия модулей, которые обеспечивают автоматическое копирование заявок, проверку лимитов бюджета и отправку уведомлений при превышении установленных норм. Все эти элементы обеспечивают бесперебойную работу системы и позволяют значительно ускорить процессы обработки заявок, минимизируя человеческий фактор и ошибки.

На рисунке 11 изображен пользовательский интерфейс для разрабатываемого модуля.

Интерфейс, разработанный для сотрудников финансового отдела, позволяет легко работать с заявками и контролировать бюджетные лимиты. Он состоит из следующих элементов:

- заявка — поле для выбора исходной заявки для последующего копирования;
- начало и конец периода — поля для ввода дат начала и конца периода, в пределах которого будут созданы копии заявок. По

умолчанию начало периода устанавливается из даты исходной заявки;

- остаток средств на текущую дату — поле для отображения свободного бюджета на момент копирования заявок;
- сумма заявок за прошлый месяц — отображается сумма заявок, созданных в прошлом месяце, для быстрого анализа текущей нагрузки;
- скопировать — кнопка, запускающая процесс копирования заявок, проверку лимитов и распределение заявок по периодам;
- табличная часть — после успешного завершения копирования в табличной части отображаются скопированные заявки с возможностью быстрого перехода к каждому документу.

The screenshot displays a web-based user interface for a request copying module. At the top right, there is a button labeled 'Еще' (More) with a dropdown arrow. Below this, the interface includes several input fields and buttons. A 'Заявка:' (Request) field is followed by a dropdown menu and a small icon. Below it are 'Начало периода:' (Start of period) and 'Конец периода:' (End of period) fields, each with a calendar icon. Further down are two numeric input fields: 'Сумма заявок за прошлый месяц (руб.):' (Sum of requests for the last month (rub.)) and 'Остаток средств на текущую дату (руб.):' (Remaining funds as of the current date (rub.)), both showing a value of 0. A 'Скопировать' (Copy) button is positioned below these fields. To the left of the 'Скопировать' button is a 'Добавить' (Add) button, and to its right are two small blue arrows pointing up and down. Below the 'Скопировать' button are two yellow buttons labeled 'Поле1: Значение1' and 'Поле2: Значение2', each with a small icon. At the bottom, there is a large table with a single header row labeled 'Заявка' (Request) and an empty body. To the right of the table, there is a search bar labeled 'Поиск (Ctrl+F)' and another 'Еще' (More) button with a dropdown arrow.

Рисунок 11 – Пользовательский интерфейс разрабатываемого модуля

Модуль копирования заявок является ключевым элементом системы, позволяющим автоматизировать процесс переноса заявок между периодами или для различных категорий расходов. Данный модуль используется для переноса данных о заявках с одного этапа на следующий с минимизацией

ручных ошибок. Это особенно важно в условиях большого объема заявок, когда вручную следить за каждой заявкой становится проблематично.

На рисунке 12 изображена диаграмма, которая показывает работу модуля копирования заявок.

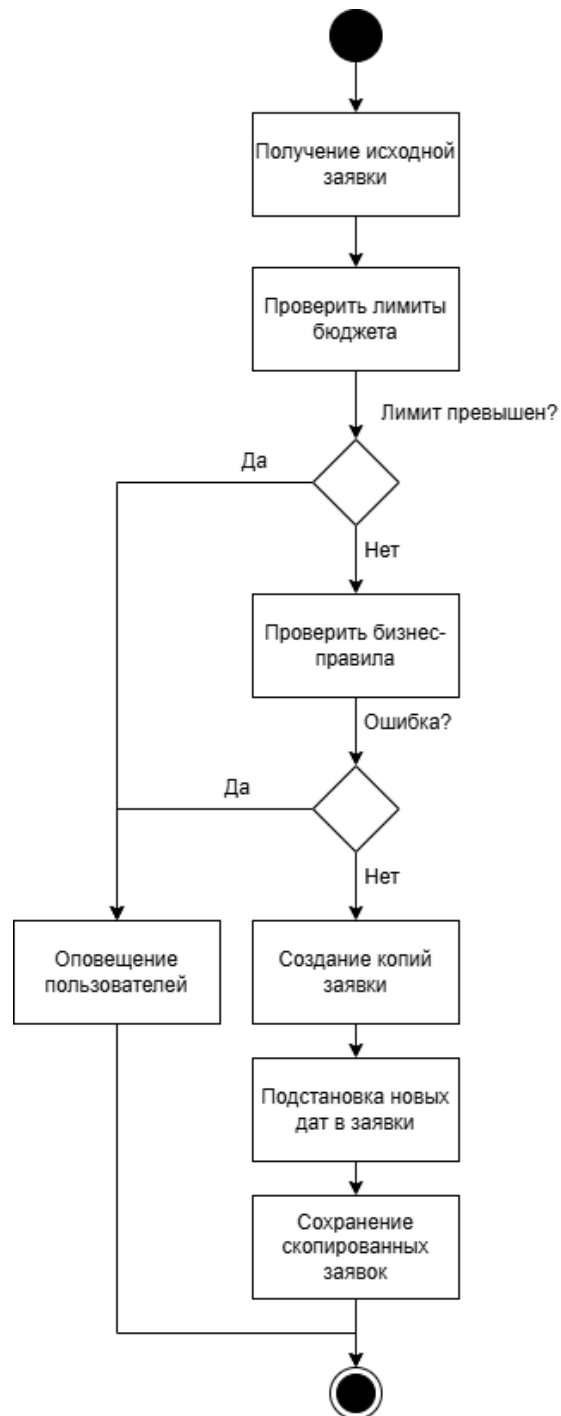


Рисунок 12 – Диаграмма работы модуля копирования

Ключевым процессом в данном модуле является проверка соответствия заявок бизнес-правилам и лимитам бюджета, а также создание копий заявок, которые затем могут быть перераспределены по периодам.

Диаграмма последовательности действий, выполняемых модулем копирования заявок. Процесс начинается с получения исходной заявки, после чего выполняется проверка соответствия установленным бизнес-правилам и лимитам бюджета. В случае успешного прохождения проверок создаётся копия заявки с автоматическим расчётом даты нового периода. Далее заявка сохраняется в системе и может быть передана в модуль распределения для дальнейшей обработки. Таким образом, обеспечивается автоматизация и контроль при переносе заявок между временными периодами.

На рисунке 13 представлен фрагмент кода, отвечающий за копирования заявок.

```
Пока Месяц(НачалоПериода) < Месяц(КонiecПериода) И Год(НачалоПериода) <= Год(КонiecПериода) Цикл
    Если День(НачалоПериода) > День(КонiecПериода) И (Месяц(КонiecПериода) - Месяц(НачалоПериода) = 1) Тогда
        НачалоПериода = КонiecПериода;
        Возврат;
    Иначе
        День = День(НачалоПериода);
        Если Число(Сред(КонiecМесяца(ДобавитьМесяц(НачалоПериода, 1)), 1, 2)) < День Тогда
            День = Число(Сред(КонiecМесяца(ДобавитьМесяц(НачалоПериода, 1)), 1, 2));
            КонiecЕсли;
        Месяц = Месяц(НачалоПериода);
        Если Месяц = 12 Тогда
            Месяц = 01;
        Иначе
            Месяц = Месяц + 1;
        КонiecЕсли;

        Год = Год(НачалоПериода);
        НачалоПериода = Дата(Год, Месяц, День);

        НоваяЗаявка = ЭтаФорма.Заявка.Скопировать();
        НоваяЗаявка.Дата = НачалоПериода;
        НоваяЗаявка.ДатаРасхода = НачалоПериода;

        ДеньКрай = День(ДатаКрай);
        МесяцКрай = Месяц(ДатаКрай);
        Если МесяцКрай = 12 Тогда
            МесяцКрай = 01;
        Иначе
            МесяцКрай = МесяцКрай + 1;
        КонiecЕсли;

        ГодКрай = Год(ДатаКрай);
        ДатаКрай = Дата(ГодКрай, МесяцКрай, ДеньКрай);
        НоваяЗаявка.ДатаПлатежаКрайняя = ДатаКрай;

        СуммаДокументов = СуммаДокументов + НоваяЗаявка.Сумма;
        Если СуммаДокументов > ОстатокСредствНаТекущуюДату Тогда
            Сообщить("Не удалось скопировать: "+ НоваяЗаявка + ", по причине: Сумма скопированных заявок превышает остаток средств, доступный для использования.");

            ТекстТема = "Лимит средств при копировании заявок на расходование ДС";
            ТекстТело = "Здравствуйте, при копировании заявок на расходование ДС пользователем " + Пользователи.ТекущийПользователь() + ", превысил лимит доступных средств." +
                Символы.ПС + "Пожалуйста, проверьте список документов 'Заявка на расходование денежных средств', а также свяжитесь с пользователем " + Пользователи.ТекущийПользователь() + ".";
            ОтправкаСообщений(ТекстТема, ТекстТело);
            Возврат;
        КонiecЕсли;

        НоваяЗаявка.Записать();
```

Рисунок 13 – Код для создания копий заявок

Этот фрагмент кода проверяет, не превышает ли сумма заявки установленный лимит. Если заявка соответствует всем условиям, создается новая копия, которая затем сохраняется в системе. На этом этапе система

автоматически рассчитывает дату создания новой заявки на основе периода, что позволяет быстро создать несколько заявок для разных временных отрезков.

На рисунке 14 изображена часть кода, отвечающая за расчет суммы заявок, созданных за прошлый месяц по отношению к исходной заявке.

```
остатокСредств(текущуюдату - суммаОстатка;  
//++ Заявок создано за прошлый месяц  
ЗапросЗаявки = Новый Запрос;  
ЗапросЗаявки.Текст = "ВЫБРАТЬ РАЗЛИЧНЫЕ  
| СУММА(бит_ЗаявкаНаРасходованиеСредств.Сумма) КАК Сумма  
| ИЗ  
| Документ.бит_ЗаявкаНаРасходованиеСредств КАК бит_ЗаявкаНаРасходованиеСредств  
| ГДЕ  
| бит_ЗаявкаНаРасходованиеСредств.Дата МЕЖДУ &ДатаНачала И &ДатаОкончания  
| И бит_ЗаявкаНаРасходованиеСредств.Проведен";  
ПрошлыйМес = ДобавитьМесяц(ТекущаяДата(), -1);  
ЗапросЗаявки.УстановитьПараметр("ДатаНачала", НачалоМесяца(ПрошлыйМес));  
ЗапросЗаявки.УстановитьПараметр("ДатаОкончания", КонецМесяца(ПрошлыйМес));  
РезультатЗаявки = ЗапросЗаявки.Выполнить().Выбрать();  
РезультатЗаявки.Следующий();  
ЗаявокНаПрошлМес = РезультатЗаявки.Сумма;  
//-- Заявок создано на прошлый месяц
```

Рисунок 14 – Код для определения сумм заявок за прошлый месяц

На данном скриншоте показан фрагмент кода для расчета сумм заявок за предыдущий месяц. Это необходимо для последующего анализа, превышает ли сумма новых заявок лимиты предыдущего периода

Модуль распределения заявок является вложенным в модуль копирования и предназначен для автоматического распределения заявок по временным периодам с учетом установленных лимитов бюджета (рисунок 15). Основная цель модуля — обеспечить равномерное распределение заявок и исключить превышение доступного бюджета.

Модуль получает данные о заявках, анализирует их суммы, категории расходов и сроки выполнения. После этого система выполняет проверку бюджета. Если сумма заявок не превышает установленный лимит, происходит их распределение по временным интервалам, например, на следующий месяц.

В случае превышения лимита система останавливает процесс распределения и отправляет уведомления пользователям о необходимости корректировки бюджета.

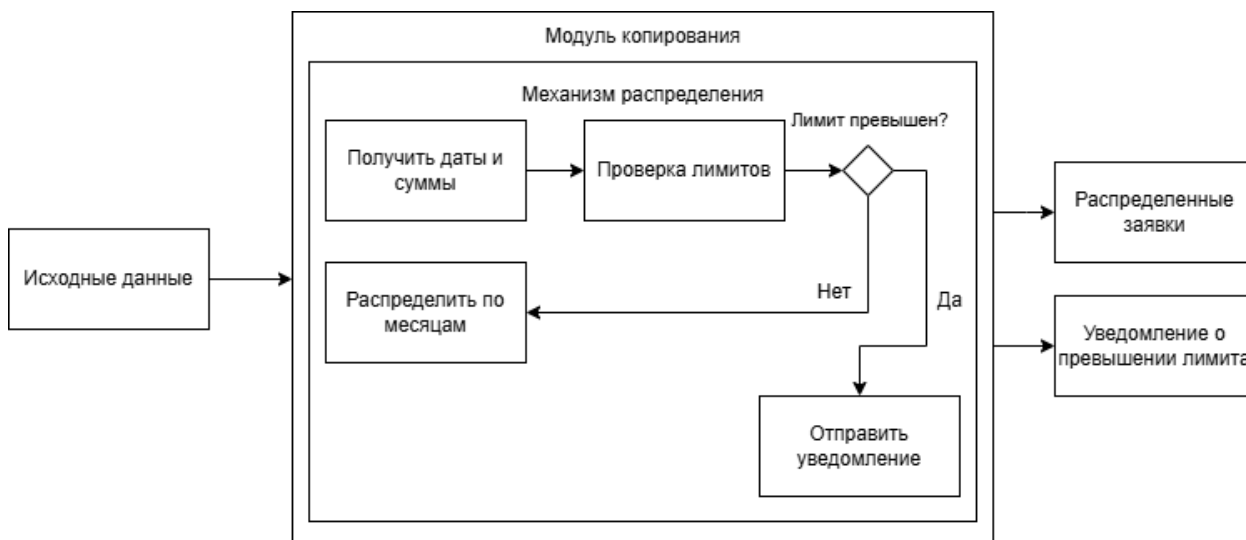


Рисунок 15 – Диаграмма работы механизма распределения

Реализация модуля распределения включает в себя обработку заявок в соответствии с заданными параметрами, проверку лимитов и отправку уведомлений. На рисунке 16 представлен фрагмент кода, реализующего этот процесс.

Этот скриншот показывает, как реализована проверка лимитов бюджета перед тем, как заявка будет обработана. В случае, если заявка превышает установленные лимиты, система автоматически блокирует процесс перераспределения

Механизм оповещений играет важную роль в системе обработки заявок, обеспечивая своевременное информирование пользователей о критически важных событиях. Основная задача механизма – уведомлять ответственных сотрудников о превышении бюджетных лимитов. Данный механизм работает в паре с модулем распределения заявок. При обнаружении превышения

лимита в модуле распределения формируется заголовок письма, тело письма, после чего эти данные передаются в механизм оповещений.

В данном механизме заполняются адреса получателей и формируется письмо.

```

НоваяЗаявка.ДатаПлатежаКрайняя = ДатаКрай;

СуммаДокументов = СуммаДокументов + НоваяЗаявка.Сумма;
Если СуммаДокументов > ОстатокСредствНаТекущуюДату Тогда
Сообщить("Не удалось скопировать: " + НоваяЗаявка + ", по причине: Сумма скопированных заявок превышает остаток средств, доступный для использования.");

ТекстТема = "Лимит средств при копировании заявок на расходование ДС";
ТекстТело = "Здравствуйте, при копировании заявок на расходование ДС пользователем " + Пользователи.ТекущийПользователь() + ", превышен лимит доступных средств." +
Символы.ПС + "Пожалуйста, проверьте список документов 'Заявка на расходование денежных средств', а также свяжитесь с пользователем " + Пользователи.ТекущийПользователь() + ".
ОтправкаСообщений(ТекстТема, ТекстТело);
Возврат;
КонецЕсли;

НоваяЗаявка.Записать();

ЗапросЗаявкиТехМес = Новый Запрос;
ЗапросЗаявкиТехМес.Текст = "ВЫБРАТЬ РАЗЛИЧНЫЕ
| СУММА(бит_ЗаявкаНаРасходованиеСредств.Сумма) КАК Сумма
| ИЗ
| Документ.бит_ЗаявкаНаРасходованиеСредств КАК бит_ЗаявкаНаРасходованиеСредств
| ГДЕ
| бит_ЗаявкаНаРасходованиеСредств.Дата МЕЖДУ &ДатаНачала И &ДатаОкончания";
ЗапросЗаявкиТехМес.УстановитьПараметр("ДатаНачала", НачалоМесяца(НоваяЗаявка.Дата));
ЗапросЗаявкиТехМес.УстановитьПараметр("ДатаОкончания", КонецМесяца(НоваяЗаявка.Дата));
РезультатЗаявкиТехМес = ЗапросЗаявкиТехМес.Выполнить().Выбрать();
РезультатЗаявкиТехМес.Следующий();
СуммаДокЗаМес = РезультатЗаявкиТехМес.Сумма;

Месяц = Формат(НоваяЗаявка.Дата, "ДФ=ММММ");
ПромышМес = Формат(ДобавитьМесяц(ТекущаяДата(), -1), "ДФ=ММММ");
МесяцП = ПолучитьСклоненияСтроки(Месяц, "ПД-Предложный")[0];
ПромышМесП = ПолучитьСклоненияСтроки(ПромышМес, "ПД-Творительный")[0];

Если СуммаДокЗаМес > ЗаявокНаПромышМес Тогда
Сообщить("Внимание! Сумма скопированных заявок за " + Месяц + " превышает сумму проведенных заявок за " + ПромышМес);

ТекстТема = "Лимит средств при копировании заявок на расходование ДС";
ТекстТело = "Здравствуйте, при копировании заявок на расходование ДС, в " + МесяцП + " был превышен лимит сумм созданных заявок по отношению с " + ПромышМесП + ".";
ОтправкаСообщений(ТекстТема, ТекстТело);

КонецЕсли;

```

Рисунок 16 – Код для проверки лимитов бюджета

На рисунке 17 изображена функция для отправки сообщений.

```

□ функция ОтправкаСообщений(ТекстТема, ТекстТело)

    УчетнаяЗаписьОтправителя = РаботаПочтовымиСообщениями.СистемнаяУчетнаяЗапись();
    ПараметрыПисьма = Новый Структура;
    ПараметрыПисьма.Вставить("Кому", "██████████@yesapart.com, ██████████@yesapart.com");
    ПараметрыПисьма.Вставить("УчетнаяЗапись", УчетнаяЗаписьОтправителя);
    ПараметрыПисьма.Вставить("АдресОтвета", "a.gataullov@yesapart.com");
    ПараметрыПисьма.Вставить("Тема", ТекстТема);
    ПараметрыПисьма.Вставить("Тело", ТекстТело);
    ПараметрыПисьма.Вставить("ТипТекста", "ПростойТекст");

    РаботаПочтовымиСообщениями.ОтправитьПочтовоеСообщение(УчетнаяЗаписьОтправителя, ПараметрыПисьма);

Конецфункции

```

Рисунок 17 – Функция для отправки оповещений

Для интеграции модуля в конфигурацию 1С используется механизм дополнительных обработок [19].

Внешние обработки в системе «1С:Предприятие» представляют собой отдельные файлы (как правило, с расширением *.erf), которые могут подключаться к основной конфигурации без необходимости её изменения, обеспечивая дополнительный функционал. Они используются для создания специализированных отчётов, обработки данных, разработки вспомогательных утилит, а также расширения возможностей системы без внесения изменений в основную конфигурацию. Каждая внешняя обработка включает модуль объекта, содержащий программный код, определяющий её поведение: в нём размещаются функции и процедуры обработки, обработчики событий, а также методы, обеспечивающие интеграцию с основной конфигурацией. Это позволяет гибко настраивать и дополнять функциональность системы в соответствии с потребностями пользователей.

На рисунке 18 изображена модель взаимодействия внешней обработки (*.erf) с основной конфигурацией «1С:Предприятие».

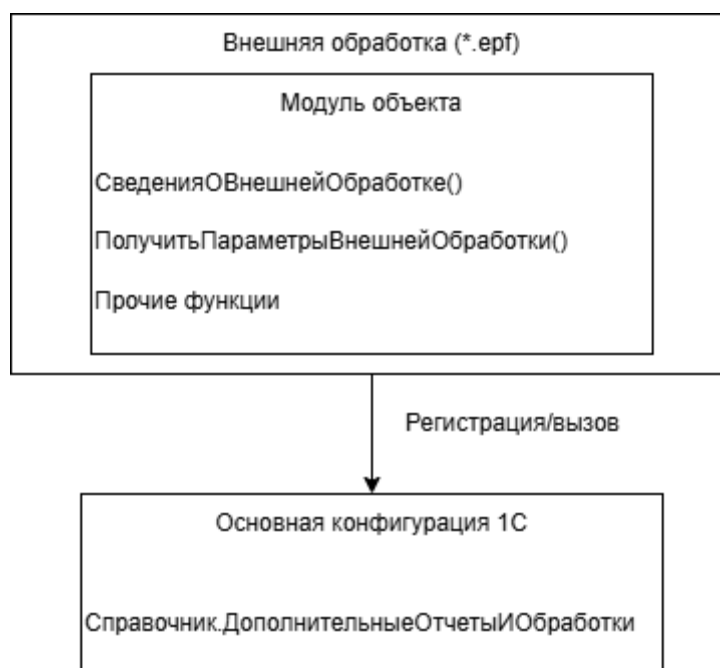


Рисунок 18 – Модель взаимодействия внешней обработки с основной конфигурацией

Внешняя обработка содержит модуль объекта, включающий функции СведенияОВнешнейОбработке() и ПолучитьПараметрыВнешнейОбработки(), а также прочие пользовательские процедуры. Она регистрируется в основной конфигурации через справочник «ДополнительныеОтчетыИОбработки», что позволяет использовать её функционал без изменения исходного кода системы.

В модуле объекта внешней обработки был написан следующий код (рисунок 19).

```

□ функция СведенияОВнешнейОбработке() Экспорт
    ПараметрыРегистрации = ДополнительныеОтчетыИОбработки.СведенияОВнешнейОбработке("2.2.2.1");
    ПараметрыЭтойОбработки = ПолучитьПараметрыЭтойОбработки(Ложь);

    ПараметрыРегистрации.Вид = ДополнительныеОтчетыИОбработкиКлиентСервер.ВидОбработкиДополнительнаяОбработка();
    ПараметрыРегистрации.Информация = ПараметрыЭтойОбработки.Комментарий
    + ? (ЗначениеЗаполнено(ПараметрыЭтойОбработки.Комментарий), Символы.ПС, "")
    + "Дата создания: 10.12.2024." + Символы.ПС
    + "Дата последнего редактирования: 29.12.2024.";
    ПараметрыРегистрации.Версия = "1.0.0.2";
    ПараметрыРегистрации.БезопасныйРежим = Истина; // отключаем безопасный режим только при необходимости

    НоваяКоманда = ПараметрыРегистрации.Команды.Добавить();
    НоваяКоманда.Представление = ПараметрыЭтойОбработки.Синоним + "Копирование заявок";
    НоваяКоманда.Идентификатор = ПараметрыЭтойОбработки.Имя + "КопированиеЗаявок";
    НоваяКоманда.Использование = ДополнительныеОтчетыИОбработкиКлиентСервер.ТипКомандыОткрытиеФормы();

    Возврат ПараметрыРегистрации;
Конецфункции

□ функция ПолучитьПараметрыЭтойОбработки(ВызыватьИсключение = Истина) Экспорт
    ИмяОбработки = ЭтотОбъект.Метаданные().Имя;
    СинонимОбработки = ЭтотОбъект.Метаданные().Синоним;
    КомментарийОбработки = ЭтотОбъект.Метаданные().Комментарий;

    Ссылка = Справочники.ДополнительныеОтчетыИОбработки.НайтиПоРеквизиту("ИмяОбъекта", ИмяОбработки);
    Если Ссылка.Пустая() Тогда
        Если ВызыватьИсключение Тогда
            ВызватьИсключение "Данная обработка не зарегистрирована в справочнике внешних обработок!";
        КонецЕсли;
    КонецЕсли;

    ПараметрыЭтойОбработки = Новый Структура;
    ПараметрыЭтойОбработки.Вставить("Ссылка", Ссылка);
    ПараметрыЭтойОбработки.Вставить("Имя", ИмяОбработки);
    ПараметрыЭтойОбработки.Вставить("Синоним", ? (ЗначениеЗаполнено(СинонимОбработки), СинонимОбработки, ИмяОбработки));
    ПараметрыЭтойОбработки.Вставить("Комментарий", КомментарийОбработки);

    Возврат ПараметрыЭтойОбработки;
Конецфункции

```

Рисунок 19 – Код модуля объекта

Основная функция СведенияОВнешнейОбработке() регистрирует обработку в системе, задает ее основные параметры, версию и добавляет команду для запуска функционала копирования заявок.

Метод ПолучитьПараметрыЭтойОбработки() позволяет получить информацию о внешней обработке, включая ее имя, синоним и комментарий.

Также он проверяет регистрацию обработки в справочнике ДополнительныеОтчетыИОбработки и возвращает ссылку на соответствующую запись.

На рисунке 20 отображен интерфейс 1С:Предприятия, который предназначен для добавления внешних отчетов и обработок.

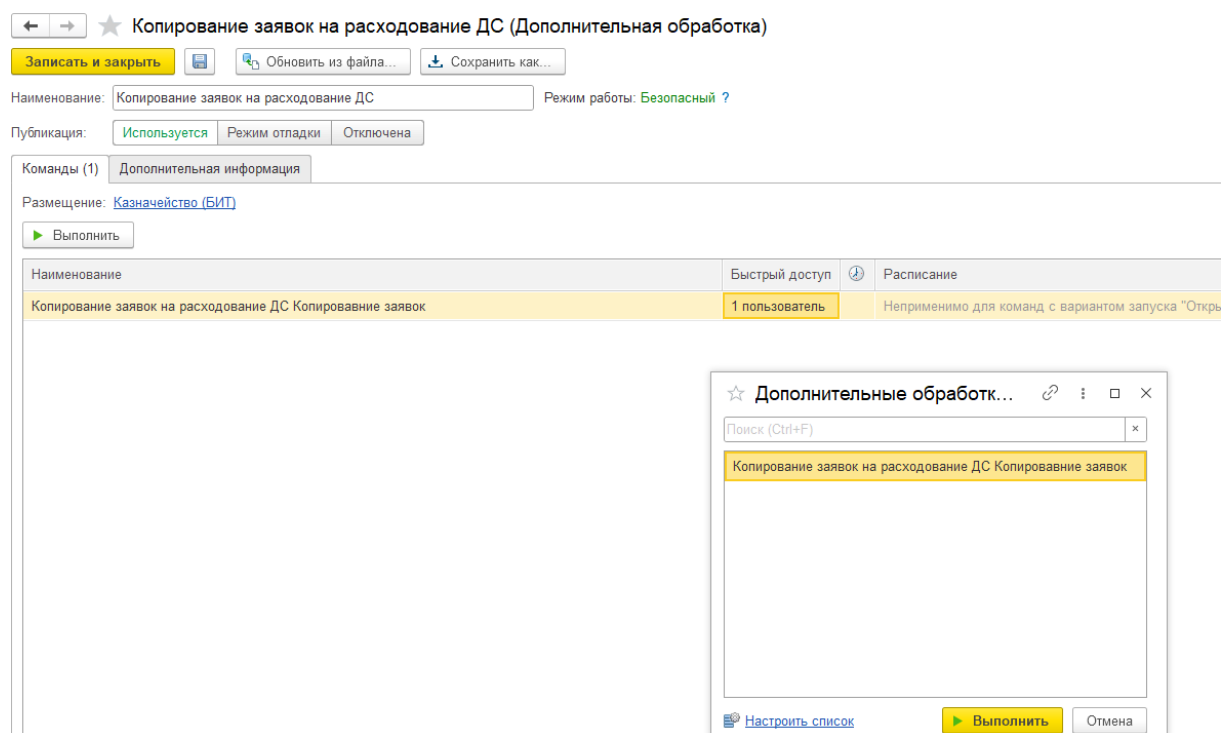


Рисунок 20 – Добавление модуля в 1С

В 1С предприятии в дополнительных отчетах и обработках был добавлен данный модуль и размещен во вкладке «Казначейство».

Благодаря данному механизму модуль корректно интегрируется в среду 1С, обеспечивая удобный доступ к функционалу копирования заявок и соблюдение требований безопасности [17].

Выводы по главе 2

Проведенная работа по проектированию и реализации системы обработки заявок на расходование денежных средств позволила создать эффективное решение, полностью соответствующее поставленным задачам.

Архитектура системы, основанная на платформе 1С:Предприятие 8.3, продемонстрировала свою состоятельность на всех этапах разработки. Ключевым достижением стало создание модульной структуры, где каждый компонент - модуль копирования заявок, механизм распределения, подсистема контроля лимитов и модуль уведомлений - выполняет строго определенные функции.

При реализации модуля были применены стандартные методы программирования на платформе 1С:Предприятие 8.3, включая построение внешних обработок и модульных скриптов [15]. Это обеспечило высокую совместимость решения с существующей инфраструктурой предприятия.

Техническая реализация системы подтвердила правильность выбранной платформы. Использование встроенных механизмов 1С позволило значительно сократить сроки разработки, обеспечив при этом высокую надежность и производительность решения.

Разработанный пользовательский интерфейс предоставляет сотрудникам финансовых служб удобный инструмент для работы, сочетающий простоту использования с широкими функциональными возможностями. Реализованные механизмы интеграции с внешними сервисами, такими как SMTP-сервер для отправки уведомлений, расширяют возможности системы без ущерба для стабильности ее работы.

Глава 3 Оценка эффективности модуля автоматизации обработки заявок

3.1 Методика и результаты тестирования функциональности модуля

После создания модуля необходимо провести ряд тестов, чтобы удостовериться, что код работает корректно. В данном разделе представлены методика тестирования, описание тестовых данных, процесс обработки, а также анализ полученных результатов [23].

Для проверки корректной работы подсистемы были проведены тесты:

- создание заявки с превышением лимита бюджета — система выдает предупреждение;
- копирование заявки — все данные переносятся правильно;
- корректная отправка электронных писем при превышении бюджета.

Для проверки работоспособности системы использовались тестовые данные, представленные на рисунке 21.

Заявка на расходование ДС ТЕСТ-001 от 22.12.2024 18:42:00

Основное | Оплата | Реквизиты (ЕСИ) | Согласование | Задачи | Информация

Провести и закрыть | Записать | Провести | Создать на основании | Список | Печать

Номер: ТЕСТ-001 от: 22.12.2024 18:42:00 Важность: Средняя

Операция: Безналичные | Оплата поставщику Организация: ИЕС АЙ ТИ КОНСАЛТИНГ ООО

Дата платежа: 22.12.2024 Не позднее: 23.12.2024

Основание: Основание для заявки

Счет ЭДО:

Документ основание (СНДС):

Статус: Черновик | Повторить платеж?

Заявка		Дополнительно	
Сценарий:	Роллинг	Валюта:	руб.
ЦФО:	Инженерно-техническая служба	Сумма:	10 000.00
Статья оборотов:	Оплаты за приобретение прочих активов	Ставка НДС:	Без НДС
Проект:		НДС:	0.00
Получатель:	ааааа	Сумма без НДС:	10 000.00
Счет получателя:		Регион:	
Договор:	Тест	Вид деятельности:	

Назначение платежа: Оплата по договору Тест Сумма 10000.00 Б т.ч. Без налога (НДС)

Рисунок 21 – Исходная заявка для тестирования

В данной заявке были заполнены следующие данные:

- номер и дата – заполняются автоматически, но для более удобного поиска в списке заявок номеру было присвоено значение ТЕСТ-001;
- дата платежа и не позднее – даты, которые образуют период платежа;
- важность – приоритет заявки на расходование денежных средств;
- организация – от какой организации создается заявка;
- сценарий – сценарии бюджетирования (необходимо для бухгалтерии);
- ЦФО – подразделение, отвечающее за заявку, а также за ее согласование;
- статья оборотов – необходима для отчетности и структуризации документов в 1С;
- получатель и договор – контрагент и его договор по которому осуществляются оплата;
- суммовые показатели – валюта, сумма, ставка НДС, НДС, сумма без НДС.

Параметры копирования: период копирования с 22.01.2025 по 21.04.2025, что означает, что должно создаваться 2 копии, так как конец периода не дошел до даты 22.04.2025.

Процесс обработки: пользователь открывает интерфейс 1С, выбирает заявку 001 и заполняет параметры копирования (рисунок 22).

Заявка: Заявка на расходование ДС ТЕСТ-001 от 22.12.2024 18:42:11

Начало периода: 22.01.2025 Конец периода: 21.04.2025 Сумма заявок за прошлый месяц (руб.): 0.00 Остаток средств на текущую дату (руб.): 871 153 775.00

Скопировать

Добавить

Заявка

Рисунок 22 – Заполненные данные модуля

После нажатия кнопки «Скопировать» система проверяет соответствие заявки бизнес-правилам. Если все условия соблюдены, система создает три копии заявки.

На выходе пользователь получает обновленный список заявок, который отображен на рисунке 23:

- исходная заявка ТЕСТ-001,
- копии: АЙ00-000001 от 22.02.2025, АЙ00-000002 от 22.03.2025.

Копирование заявок на расходование ДС

Заявка: Заявка на расходование ДС ТЕСТ-001 от 22.12.2024 18:42:1

Начало периода: 22.01.2025 Конец периода: 21.04.2025 Сумма заявок за прошлый месяц (руб.): 0,00 Остаток средств на текущую дату (руб.): 871 153 775,00

Скопировать

Добавить

Заявка

Заявка на расходование ДС АЙ00-000001 от 22.02.2025 0:00:00
Заявка на расходование ДС АЙ00-000002 от 22.03.2025 0:00:00

Рисунок 23 – Результат работы модуля

Скопированные данные сохраняются в базе и становятся доступны для дальнейшей обработки. Так как мы работаем в тестовой базе, сумма заявок за прошлый месяц равна нулю, что также является одной из важных проверок работы модуля, а конкретно проверки работы предупреждений, а также отправки оповещений.

На рисунке 24 показана работа уведомлений, которая информирует пользователя о превышении лимитов.

Сообщения:

- Внимание! Сумма скопированных заявок за Февраль превышает сумму проведенных заявок за Январь
- Внимание! Сумма скопированных заявок за Март превышает сумму проведенных заявок за Февраль

Рисунок 24 – Предупреждения о превышении суммы заявок

Как видно из теста, предупреждения работают корректно.

На рисунке 25 видно, что созданный модуль отправляет уведомления корректно и по указанным адресам.

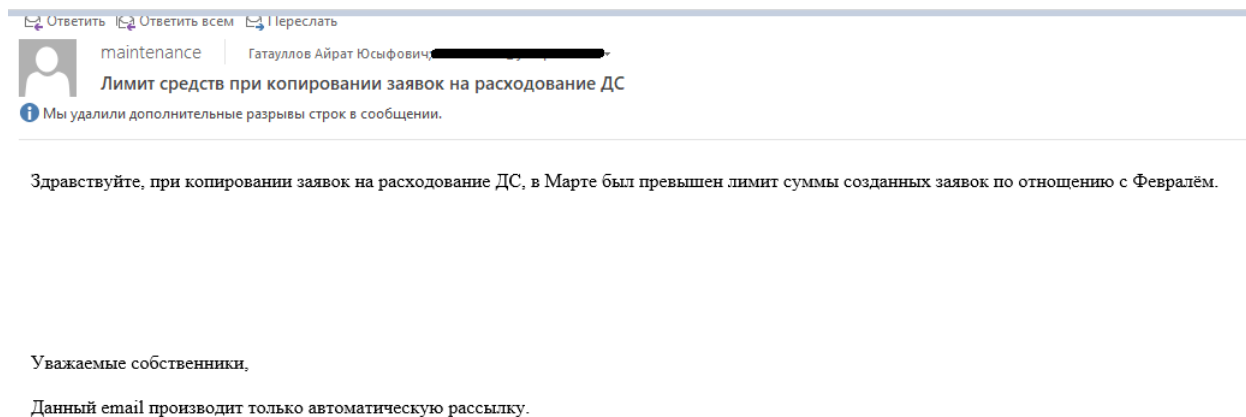


Рисунок 25 – Пример оповещения руководителям

При открытии одной из копий можно удостовериться, что все данные перенесены корректно, а даты изменены ровно на месяц вперед (рисунок 26).

Рисунок 26 – Скопированная заявка

При создании копий заявок, сумма которых превышает разрешенный бюджет, модуль останавливает процесс копирования и отправляет следующее письмо (рисунок 27).

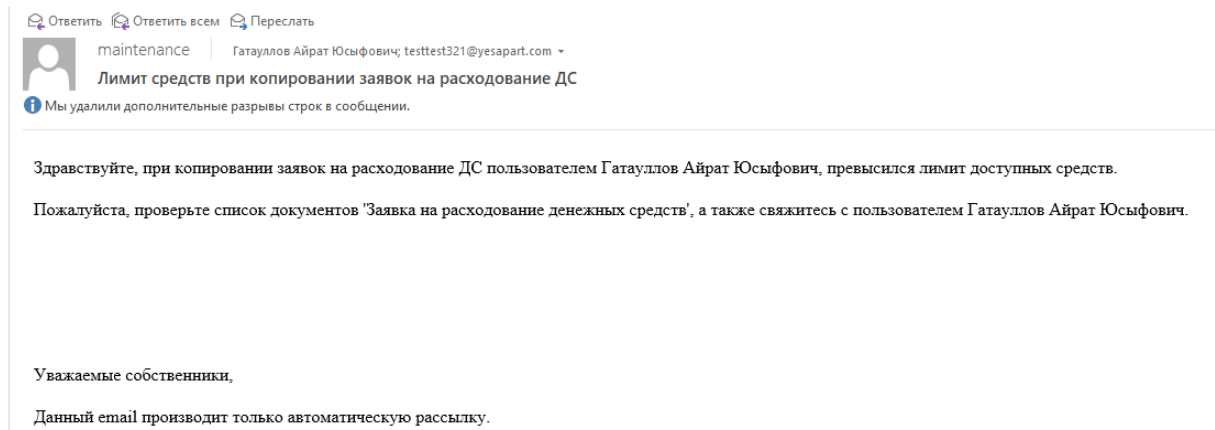


Рисунок 27 – Пример письма при превышении лимита

В данном письме уже явно указано, кто создавал копии заявок.

Тестирование подтвердило полную работоспособность модуля. Все функции работают корректно: заявки копируются без ошибок, даты пересчитываются верно, уведомления отправляются вовремя. Механизм контроля лимитов надежно блокирует операции при превышении бюджета. Модуль полностью соответствует требованиям и готов к внедрению.

3.2 Оценка удобства интерфейса и пользовательского опыта

Для комплексной оценки разработанного модуля обработки заявок был разработан специальный оценочный лист, включающий ключевые параметры качества программного обеспечения. Основное внимание уделялось практической полезности системы для конечных пользователей - сотрудников финансового отдела [16]. Согласно таблице 8, ключевыми параметрами стали удобство интерфейса (25% веса) и полнота функционала (25% веса).

Таблица 8 – Критерии оценки с весовыми коэффициентами

Критерий	Описание	Вес	Метод оценки
Удобство интерфейса	Интуитивность навигации, понятность элементов управления	25%	Анкетирование пользователей
Скорость работы	Время отклика системы на действия пользователя	20%	Хронометраж операций
Полнота функционала	Соответствие возможностей системы заявленным требованиям	25%	Экспертная оценка
Надежность	Стабильность работы при различных сценариях использования	15%	Стресс-тестирование
Качество уведомлений	Своевременность и информативность системных оповещений	15%	Анализ журналов событий

Для получения объективной картины качества системы применялись различные методы сбора информации:

- анкетирование 15 пользователей из числа сотрудников финансового отдела;
- глубинные интервью с 5 ключевыми специалистами;
- экспертное тестирование проводили 3 специалиста по юзабилити;
- автоматизированный сбор метрик производительности системы.

Тестирование проводилось по строгому протоколу, включающему:

- подбор тестовой группы (10 человек – регулярные пользователи системы);
- разработку ключевых сценариев использования: массовое копирование заявок; проверка лимитов бюджета; работа с уведомлениями;
- фиксацию всех действий пользователей с помощью: системы записи экрана; аудиокomentarиев пользователей.

Собранные данные были систематизированы и проанализированы по двум направлениям (таблица 9).

Качественный анализ выявил три основных направления для улучшения:

- необходимость добавления подсказок для сложных операций;

- оптимизацию формы массового копирования;
- необходимость добавления выбора периодичности копирования заявок.

Таблица 9 – Результаты тестирования удобства

Показатель	Целевое значение	Фактический результат	Отклонение
Ошибок при копировании	≤ 1 на 20 операц.	0 на 50 операций	-
Удовлетворенность UI	$\geq 80\%$	89%	+9%

Проведенная оценка показала, что система в целом соответствует предъявляемым требованиям, демонстрируя особенно высокие результаты по показателям надежности (99,2% успешных операций) и скорости работы (среднее время отклика 1,3 сек). Наибольшее одобрение пользователей получил механизм контроля лимитов бюджета, который был оценен на 4.8 из 5 баллов.

Полученные результаты легли в основу плана доработок системы, который включает 7 пунктов по улучшению пользовательского опыта. Особое внимание будет уделено упрощению интерфейса массового копирования заявок - самой сложной, по мнению пользователей, операции в системе.

3.3 Анализ эффективности автоматизации обработки заявок

Анализ результатов использования программного обеспечения — ключевой этап, который позволяет оценить, насколько успешно приложение выполняет свои функции и удовлетворяет потребности пользователей. Данный процесс включает в себя сбор, классификацию, интерпретацию данных, а также выработку рекомендаций по улучшению [22].

На первом этапе был организован сбор информации о взаимодействии пользователей с приложением. Для этого применялись следующие методы:

- логирование действий пользователей – каждый клик, ввод или действие в интерфейсе фиксировались в системе для последующего анализа;
- опросы и анкеты – пользователи оценивали приложение по критериям удобства, скорости, понятности;
- анализ метрик производительности – учитывались такие параметры, как среднее время отклика, количество ошибок, скорость копирования заявок и обработки лимитов.

Для более точного анализа была проведена их классификация по ключевым признакам, представленная в таблице 10.

Таблица 10 – Классификация данных с описанием групп

Классификация	Группа	Характеристика
По типу пользователей	Новички	Первое использование, требуется обучение
	Опытные	Регулярные пользователи, владеют функционалом
По сценарию использования	Копирование заявок	Массовое и одиночное копирование
	Проверка лимитов	Реакция на превышения бюджета
	Работа с уведомлениями	Восприятие и реакция на письма и оповещения

Для количественного анализа были использованы системные метрики и результаты опросов. Полученные показатели представлены в таблице 11.

Таблица 11 – Результаты количественного анализа производительности и удобства

Показатель	Единица	Значение	Комментарий
Среднее время отклика	сек.	1.3	В пределах нормы
Ошибок при копировании	кол-во на 50 опер.	0	Отличный результат
Удовлетворенность UI	%	89%	Превышает целевое значение
Информативность уведомлений	баллы из 5	4.8	Высокая оценка

Качественный анализ позволил выявить проблемные зоны и пользовательские предпочтения. Применялись следующие методы:

- контент-анализ – группировка повторяющихся тем в анкетах: «неочевидные действия», «отсутствие подсказок», «удобство оповещений»;
- анализ пользовательских сессий – просмотр записей экранов пользователей выявил, где они теряются в интерфейсе, и какие действия вызывают затруднения.

На основе проведенного анализа были сформулированы рекомендации по улучшению, которые расписаны в таблице 12.

Таблица 12 – Рекомендации по улучшению

Направление	Рекомендация	Ожидаемый эффект
Интерфейс	Добавить всплывающие подсказки для сложных операций	Повышение интуитивности
Функциональность	Расширить параметры копирования (например, «каждые 2 месяца»)	Увеличение гибкости использования
Производительность	Оптимизировать загрузку справочников в модуле копирования	Ускорение работы на больших объёмах данных

Проведённый анализ использования программного обеспечения показал его высокую эффективность и соответствие потребностям конечных пользователей. Благодаря систематическому подходу к сбору и интерпретации данных удалось выявить не только сильные стороны, но и зоны роста. Разработка конкретных рекомендаций на основе анализа позволит улучшить функциональность модуля, повысить удовлетворённость пользователей и в конечном счёте – эффективность работы всей автоматизированной системы.

Выводы по главе 3

На основании проведенных тестов можно сделать следующие выводы о работоспособности разработанного модуля.

- корректность выполнения основных функций: модуль успешно выполняет копирование заявок, переносит все данные без потерь и корректно изменяет даты на месяц вперед;
- контроль превышения лимита бюджета: в случае попытки создания заявки с превышением допустимого бюджета система корректно выводит предупреждение, а также останавливает процесс копирования;
- отправка уведомлений: руководители получают уведомления о превышении бюджета;
- автоматизация обработки заявок: реализованный функционал позволяет значительно упростить работу сотрудников, снизить количество ручных операций и уменьшить вероятность ошибок при копировании заявок. Это повышает общую эффективность работы системы учета заявок на расходование денежных средств;
- работа в тестовой базе: тестирование проводилось в контролируемой среде. Полученные результаты позволяют утверждать, что модуль готов к развертыванию в рабочей среде, так как все ключевые проверки пройдены успешно;

В целом, проведенные тесты подтвердили соответствие разработанного модуля техническому заданию. Система функционирует стабильно, выполняет основные задачи. Внедрение данного модуля позволит значительно повысить прозрачность и эффективность процесса управления заявками на расходование денежных средств.

Заключение

Современное развитие информационных технологий оказывает значительное влияние на все сферы деятельности организаций. Автоматизация бизнес-процессов становится ключевым фактором повышения эффективности работы предприятий [20]. В данной работе была разработана автоматизированная система обработки заявок на расходование денежных средств, что позволило оптимизировать процесс копирования заявок, проверки лимитов бюджета, отправки уведомлений и формирования отчетности.

В ходе выполнения работы был проведен анализ предметной области, исследованы существующие решения, рассмотрены их преимущества и недостатки. На основе полученных данных была обоснована необходимость разработки собственной системы, адаптированной под специфику работы организации ООО «ЙЕС АЙ ТИ КОНСАЛТИНГ». Разработанный модуль позволил автоматизировать обработку заявок, снизить вероятность ошибок, связанных с человеческим фактором, а также обеспечить контроль соблюдения бюджетных ограничений.

Внедрение автоматизированной системы обработки заявок позволило повысить прозрачность и управляемость финансовых потоков организации. Система позволяет значительно сократить время на обработку заявок, исключить возможность неконтролируемых превышений бюджета и обеспечить своевременное оповещение заинтересованных сторон.

Таким образом, разработанный программный продукт полностью соответствует заявленным требованиям и может быть рекомендован к внедрению в деятельность организации. Дальнейшее развитие системы может включать интеграцию с другими учетными модулями, расширение функционала отчетности и улучшение механизмов автоматического контроля бюджета [8].

Список используемой литературы и используемых источников

1. Бухалков М. И. Бизнес-процессы: Регламентация и управление. – М.: Инфра-М, 2021. – 224 с.
2. Головкин Н.В., Трошин Е.С. Интеграция 1С:Предприятие с внешними системами. – М.: Альпина Диджитал, 2018.
3. ГОСТ 34.601-90. Автоматизированные системы. Стадии создания. – М.: Изд-во стандартов, 1990.
4. Дейт К. Дж. Введение в системы баз данных / Пер. с англ. – СПб.: Питер, 2019. – 896 с.
5. Джексон М. Программная инженерия. Принципы и практика. – СПб.: Питер, 2018. – 352 с.
6. Кондратьев А.В., Гусев П.П. Программирование на платформе «1С:Предприятие 8.3»: основы и приемы. – М.: ДМК Пресс, 2017.
7. Кузнецов С.Ю. Разработка внешних отчетов и обработок для 1С. – СПб.: БХВ-Петербург, 2021. – 256 с.
8. Лаптев А.С. Управление проектами внедрения 1С. – М.: Альпина Паблишер, 2015.
9. Мартин Р. Чистая архитектура. Искусство разработки программного обеспечения. – СПб.: Питер, 2021. – 352 с
10. Матвеев А.Н. 1С: Бухгалтерия предприятия 8.2: практическое руководство. – М.: ДМК Пресс, 2012.
11. Поляков П.Е. 1С: Управление производственным предприятием 8. Практическое руководство. – М.: Питер, 2013.
12. Райтман М.А. Оптимизация бизнес-процессов в 1С:Предприятие. – М.: БИНОМ, 2022. – 180 с
13. Рубцов С.А., Павлов А.И. Основы конфигурирования в «1С:Предприятие 8.3». – Екатеринбург: Уральский центр, 2020.
14. Семенов А.Н., Захаров И.П. Создание эффективных решений в «1С:Предприятие 8». – СПб.: Питер, 2019.

15. Сергеев В.В. Программирование в 1С:Предприятие 8.3. – СПб.: Наука и Техника, 2016.
16. Стивенс П. UML и шаблоны проектирования. Введение в объектно-ориентированный анализ и проектирование. – СПб.: Питер, 2020. – 416 с.
17. Тихонов А.В. Администрирование и сопровождение «1С:Предприятие 8». – М.: 1С-Публишинг, 2014.
18. Фаулер М. Архитектура корпоративных программных приложений. – СПб.: Питер, 2019. – 448 с.
19. Шерстнев М.А. 1С: Предприятие 8.3. Практика программирования. – М.: Лаборатория знаний, 2015.
20. ISO/IEC 12207:2017. Systems and software engineering – Software life cycle processes. – Geneva: ISO, 2017. – 142 p.
21. Cummings J., Haag S. Management Information Systems for the Information Age. – McGraw-Hill Education, 2019. – 480 p.
22. Laudon K. C., Laudon J. P. Management Information Systems: Managing the Digital Firm. – Pearson, 2020. – 720 p.
23. Pressman R. Software Engineering: A Practitioner's Approach. – McGraw-Hill Education, 2020. – 896 p.
24. Sommerville I. Software Engineering. – Pearson, 2020. – 832 p.
25. Turban E., Pollard C., Wood G. Information Technology for Management: On-Demand Strategies for Performance, Growth and Sustainability. – Wiley, 2018. – 640 p.

Приложение А

Листинг разрабатываемого модуля

&НаСервере

Процедура СкопироватьНаСервере()

Таблица.Очистить();

НачалоПериодаП = ЭтаФорма.НачалоПериода;

КонецПериодаП = ЭтаФорма.КонецПериода;

ДатаКрай = Заявка.ДатаПлатежаКрайняя;

Счетчик = 0;

СуммаДокументов = 0;

Пока Месяц(НачалоПериодаП) < Месяц(КонецПериодаП) И Год(НачалоПериодаП)
<= Год(КонецПериодаП) Цикл

Если День(НачалоПериодаП) > День(КонецПериодаП) И
(Месяц(КонецПериодаП) - Месяц(НачалоПериодаП) = 1) Тогда

НачалоПериодаП = КонецПериодаП;

Возврат;

Иначе

День = День(НачалоПериодаП);

Если

Число(Сред(КонецМесяца(ДобавитьМесяц(НачалоПериодаП,1)), 1, 2)) < День Тогда

День =

Число(Сред(КонецМесяца(ДобавитьМесяц(НачалоПериодаП,1)), 1, 2))

КонецЕсли; Месяц = Месяц(НачалоПериодаП);

Если Месяц = 12 Тогда

Месяц = 01;

Иначе

Месяц = Месяц + 1;

КонецЕсли;

Год = Год(НачалоПериодаП);

НачалоПериодаП = Дата(Год,Месяц,День);

НоваяЗаявка = ЭтаФорма.Заявка.Скопировать();

НоваяЗаявка.Дата = НачалоПериодаП;

Продолжение приложения А

НоваяЗаявка.ДатаРасхода = НачалоПериодаП;

ДеньКрай = День(ДатаКрай);

МесяцКрай = Месяц(ДатаКрай);

Если МесяцКрай = 12 Тогда

 МесяцКрай = 01;

Иначе

 МесяцКрай = МесяцКрай + 1;

КонецЕсли;

ГодКрай = Год(ДатаКрай);

ДатаКрай = Дата(ГодКрай, МесяцКрай, ДеньКрай);

НоваяЗаявка.ДатаПлатежаКрайняя = ДатаКрай;

СуммаДокументов = СуммаДокументов + НоваяЗаявка.Сумма;

Если СуммаДокументов > ОстатокСредствНаТекущуюДату Тогда

 Сообщить("Не удалось скопировать: "+ НоваяЗаявка + ", по
причине: Сумма скопированных заявок превышает остаток средств, доступный для
использования.");

 ТекстТема = "Лимит средств при копировании заявок на
расходование ДС";

 ТекстТело = "Здравствуйте, при копировании заявок на
расходование ДС пользователем " + Пользователи.ТекущийПользователь() + ", превысился
лимит доступных средств."+

 Символы.ПС + "Пожалуйста, проверьте список документов
'Заявка на расходование денежных средств', а также свяжитесь с пользователем " +
Пользователи.ТекущийПользователь() + ".";

 ОтправкаСообщений(ТекстТема, ТекстТело);

 Возврат;

КонецЕсли;

НоваяЗаявка.Записать();

Продолжение приложения А

```

ЗапросЗаявкиТекМес = Новый Запрос;
ЗапросЗаявкиТекМес.Текст = "ВЫБРАТЬ РАЗЛИЧНЫЕ
|      СУММА(бит_ЗаявкаНаРасходованиеСредств.Сумма)      КАК
Сумма
|ИЗ
|      Документ.бит_ЗаявкаНаРасходованиеСредств      КАК
бит_ЗаявкаНаРасходованиеСредств
|ГДЕ
|      бит_ЗаявкаНаРасходованиеСредств.Дата      МЕЖДУ
&ДатаНачала И &ДатаОкончания";
ЗапросЗаявкиТекМес.УстановитьПараметр("ДатаНачала",
НачалоМесяца(НоваяЗаявка.Дата));
ЗапросЗаявкиТекМес.УстановитьПараметр("ДатаОкончания",
КонецМесяца(НоваяЗаявка.Дата));
РезультатЗаявкиТекМес =
ЗапросЗаявкиТекМес.Выполнить().Выбрать();
РезультатЗаявкиТекМес.Следующий();
СуммаДокЗаМес = РезультатЗаявкиТекМес.Сумма;

Месяц = Формат(НоваяЗаявка.Дата, "ДФ=ММММ");
ПрошлыйМес = Формат(ДобавитьМесяц(НоваяЗаявка.Дата, -1),
"ДФ=ММММ");
МесяцРП =
ПолучитьСклоненияСтроки(Месяц, "ПД=Предложный")[0];
ПрошлыйМесТП =
ПолучитьСклоненияСтроки(ПрошлыйМес, "ПД=Творительный")[0];

Если СуммаДокЗаМес > ЗаявокНаПрошлМес Тогда
Сообщить("Внимание! Сумма скопированных заявок за "+
Месяц + " превышает сумму проведенных заявок за " + ПрошлыйМес);
ТекстТема = "Лимит средств при копировании заявок на
расходование ДС";

```

Продолжение приложения А

ТекстТело = "Здравствуйте, при копировании заявок на расходование ДС, в " + МесяцРП + " был превышен лимит суммы созданных заявок по отношению с "+ ПрощлыйМесТП + ".";

ОтправкаСообщений(ТекстТема, ТекстТело);

КонецЕсли;

//Заполнение таблицы

НоваяСтрока = Таблица.Добавить();

Таблица[Счетчик].Заявка = НоваяЗаявка.Ссылка;

ДатаКрай = ДатаКрай +1;

Счетчик = Счетчик + 1;

КонецЕсли;

КонецЦикла;

КонецПроцедуры

&НаКлиенте

Процедура Скопировать(Команда)

СкопироватьНаСервере();

КонецПроцедуры

&НаСервере

Процедура ЗаявкаПриИзмененииНаСервере()

ЭтаФорма.НачалоПериода = ЭтаФорма.Заявка.Дата;

КонецПроцедуры

&НаКлиенте

Процедура ЗаявкаПриИзменении(Элемент)

ЗаявкаПриИзмененииНаСервере();

КонецПроцедуры

&НаСервере

Продолжение приложения А

Процедура ПриОткрытииНаСервере()

СуммаОстатка = 0;

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ РАЗРЕШЕННЫЕ

Хозрасчетный.Ссылка КАК Счет

|ПОМЕСТИТЬ СчетаКассы

|ИЗ

ПланСчетов.Хозрасчетный КАК Хозрасчетный

|ГДЕ

Хозрасчетный.Ссылка	В	ИЕРАРХИИ
---------------------	---	----------

(ЗНАЧЕНИЕ(ПланСчетов.Хозрасчетный.Касса))

И	НЕ	Хозрасчетный.Ссылка	В	ИЕРАРХИИ
---	----	---------------------	---	----------

(ЗНАЧЕНИЕ(ПланСчетов.Хозрасчетный.ДенежныеДокументы),

ЗНАЧЕНИЕ(ПланСчетов.Хозрасчетный.ДенежныеДокументыВал))

И (НЕ Хозрасчетный.Валютный

ИЛИ & Ведется Учет По Валютам)

1

|ИНДЕКСИРОВАТЬ ПО

Счет

|;

1

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

|ВЫБРАТЬ РАЗРЕШЕННЫЕ

Хозрасчетный.Ссылка КАК Счет

|ПОМЕСТИТЬ СчетаУчетаБезналичные

|ИЗ

ПланСчетов.Хозрасчетный КАК Хозрасчетный

|ГДЕ

Хозрасчетный.Ссылка	В	ИЕРАРХИИ
---------------------	---	----------

(ЗНАЧЕНИЕ(ПланСчетов.Хозрасчетный.РасчетныеСчета),

ЗНАЧЕНИЕ(ПланСчетов.Хозрасчетный.ВалютныеСчета),

ЗНАЧЕНИЕ(ПланСчетов.Хозрасчетный.ЦифровойРубль),

Продолжение приложения А

```

ЗНАЧЕНИЕ(ПланСчетов.Хозрасчетный.СпециальныеСчета),
ЗНАЧЕНИЕ(ПланСчетов.Хозрасчетный.ПереводыВПути_)
|      И      НЕ      Хозрасчетный.Ссылка      В      ИЕРАРХИИ
(ЗНАЧЕНИЕ(ПланСчетов.Хозрасчетный.ПриобретениеИностраннойВалюты),
ЗНАЧЕНИЕ(ПланСчетов.Хозрасчетный.ПродажиПоПлатежнымКартам),
ЗНАЧЕНИЕ(ПланСчетов.Хозрасчетный.РеализацияИностраннойВалюты))
|      И (НЕ Хозрасчетный.Валютный
|
|      ИЛИ &ВедетсяУчетПоВалютам)
|
|ИНДЕКСИРОВАТЬ ПО
|      Счет
|;
|
|////////////////////////////////////
|ВЫБРАТЬ РАЗРЕШЕННЫЕ
|      Хозрасчетный.Ссылка КАК Счет
|ПОМЕСТИТЬ СчетаУчетаПодотчет
|ИЗ
|      ПланСчетов.Хозрасчетный КАК Хозрасчетный
|ГДЕ
|      Хозрасчетный.Ссылка      В      ИЕРАРХИИ
(ЗНАЧЕНИЕ(ПланСчетов.Хозрасчетный.РасчетыСПодотчетнымиЛицами_))
|      И (НЕ Хозрасчетный.Валютный
|
|      ИЛИ &ВедетсяУчетПоВалютам)
|
|ИНДЕКСИРОВАТЬ ПО
|      Счет
|;
|
|////////////////////////////////////
|ВЫБРАТЬ РАЗРЕШЕННЫЕ
|      СчетаКассы.Счет КАК Счет
|ПОМЕСТИТЬ СчетаУчета

```

Продолжение приложения А

|ИЗ

| СчетаКассы КАК СчетаКассы

|

|ОБЪЕДИНИТЬ ВСЕ

|

|ВЫБРАТЬ

| СчетаУчетаБезналичные.Счет

|ИЗ

| СчетаУчетаБезналичные КАК СчетаУчетаБезналичные

|

|ОБЪЕДИНИТЬ ВСЕ

|

|ВЫБРАТЬ

| СчетаУчетаПодотчет.Счет

|ИЗ

| СчетаУчетаПодотчет КАК СчетаУчетаПодотчет

|

|ИНДЕКСИРОВАТЬ ПО

| Счет

|;

|

|////////////////////////////////////

|ВЫБРАТЬ РАЗРЕШЕННЫЕ

| Хозрасчетный.Ссылка КАК Счет

|ПОМЕСТИТЬ СчетаПереводовВПути

|ИЗ

| ПланСчетов.Хозрасчетный КАК Хозрасчетный

|ГДЕ

| Хозрасчетный.Ссылка

В

ИЕРАРХИИ

(ЗНАЧЕНИЕ(ПланСчетов.Хозрасчетный.ПереводыВПути_))

|

|ИНДЕКСИРОВАТЬ ПО

| Счет

Продолжение приложения А

|;

1

|||||

|ВЫБРАТЬ РАЗРЕШЕННЫЕ

Хозрасчетный.Ссылка КАК Счет

ПОМЕСТИТЬ СчетаКассОрганизации

|ИЗ

ПланСчетов.Хозрасчетный КАК Хозрасчетный

|ГДЕ

Хозрасчетный.Ссылка	В	ИЕРАРХИИ
1	2	3
4	5	6
7	8	9
10	11	12
13	14	15
16	17	18
19	20	21
22	23	24
25	26	27
28	29	30
31	32	33
34	35	36
37	38	39
40	41	42
43	44	45
46	47	48
49	50	51
52	53	54
55	56	57
58	59	60
61	62	63
64	65	66
67	68	69
70	71	72
73	74	75
76	77	78
79	80	81
82	83	84
85	86	87
88	89	90
91	92	93
94	95	96
97	98	99
100	101	102
103	104	105
106	107	108
109	110	111
112	113	114
115	116	117
118	119	120
121	122	123
124	125	126
127	128	129
130	131	132
133	134	135
136	137	138
139	140	141
142	143	144
145	146	147
148	149	150
151	152	153
154	155	156
157	158	159
160	161	162
163	164	165
166	167	168
169	170	171
172	173	174
175	176	177
178	179	180
181	182	183
184	185	186
187	188	189
190	191	192
193	194	195
196	197	198
199	200	201
202	203	204
205	206	207
208	209	210
211	212	213
214	215	216
217	218	219
220	221	222
223	224	225
226	227	228
229	230	231
232	233	234
235	236	237
238	239	240
241	242	243
244	245	246
247	248	249
250	251	252
253	254	255
256	257	258
259	260	261
262	263	264
265	266	267
268	269	270
271	272	273
274	275	276
277	278	279
280	281	282
283	284	285
286	287	288
289	290	291
292	293	294
295	296	297
298	299	300
301	302	303
304	305	306
307	308	309
310	311	312
313	314	315
316	317	318
319	320	321
322	323	324
325	326	327
328	329	330
331	332	333
334	335	336
337	338	339
340	341	342
343	344	345
346	347	348
349	350	351
352	353	354
355	356	357
358	359	360
361	362	363
364	365	366

(ЗНАЧЕНИЕ(ПланСчетов.Хозрасчетный.КассаОрганизации)),

ЗНАЧЕНИЕ(ПланСчетов.Хозрасчетный.КассаОрганизацииВал))

1

ИНДЕКСИРОВАТЬ ПО

Счет

|;

1

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

|ВЫБРАТЬ РАЗРЕШЕННЫЕ

ХозрасчетныйОстатки.Организация КАК Организация,

ЕСТЬNULL(ХозрасчетныйОстатки.Подразделение, НЕОПРЕДЕЛЕНО) КАК

Подразделение,

| ВЫБОР

КОГДА ХозрасчетныйОстатки.Счет В

(ВЫБРАТЬ

СчетаКассы.Счет

ИЗ

СчетаКассы КАК СчетаКассы)

ТОГДА ""Деньги в кассе""

КОГДА ХозрасчетныйОстатки.Счет В

(ВЫБРАТЬ

СчетаУчетаБезналичные.Счет

ИЗ

Продолжение приложения А

	СчетаУчетаБезналичные КАК СчетаУчетаБезналичные)		
	ТОГДА ""Деньги на расчетных счетах""		
	ИНАЧЕ ""Деньги у подотчетных лиц""		
	КОНЕЦ КАК ВидДенежныхСредств,		
	ВЫБОР		
	КОГДА ХозрасчетныйОстатки.Счет В		
	(ВЫБРАТЬ		
	СчетаУчетаПодотчет.Счет		
	ИЗ		
	СчетаУчетаПодотчет КАК СчетаУчетаПодотчет)		
	ТОГДА ВЫРАЗИТЬ(ХозрасчетныйОстатки.Субконто1	КАК	
	Справочник.ФизическиеЛица).Наименование		
	КОГДА ХозрасчетныйОстатки.Счет В		
	(ВЫБРАТЬ		
	СчетаУчетаБезналичные.Счет		
	ИЗ		
	СчетаУчетаБезналичные КАК СчетаУчетаБезналичные)		
	ТОГДА ВЫБОР		
	КОГДА НЕ ХозрасчетныйОстатки.Счет В		
	(ВЫБРАТЬ		
	СчетаПереводовВПути.Счет		
	ИЗ		
	СчетаПереводовВПути)		
	ТОГДА		
	ВЫРАЗИТЬ(ХозрасчетныйОстатки.Субконто1	КАК	
	Справочник.БанковскиеСчета).Наименование		
	ИНАЧЕ ХозрасчетныйОстатки.Счет.Наименование		
	КОНЕЦ		
	ИНАЧЕ ВЫБОР		
	КОГДА ХозрасчетныйОстатки.Счет В		
	(ВЫБРАТЬ		
	СчетаКассОрганизации.Счет		
	ИЗ		

Продолжение приложения А

	СчетаКассОрганизации)
	ТОГДА
	(ЗНАЧЕНИЕ(ПланСчетов.Хозрасчетный.КассаОрганизации)).Наименование
	ИНАЧЕ ХозрасчетныйОстатки.Счет.Наименование
	КОНЕЦ
	КОНЕЦ КАК Размещение,
	ВЫБОР
	КОГДА ХозрасчетныйОстатки.Счет.Валютный
	ТОГДА ХозрасчетныйОстатки.Валюта
	ИНАЧЕ &Рубль
	КОНЕЦ КАК Валюта,
	ВЫБОР
	КОГДА ХозрасчетныйОстатки.Счет.Валютный
	ТОГДА ХозрасчетныйОстатки.ВалютнаяСуммаОстаток
	ИНАЧЕ ХозрасчетныйОстатки.СуммаОстаток
	КОНЕЦ КАК ВалютнаяСуммаОстаток,
	ХозрасчетныйОстатки.СуммаОстаток КАК СуммаОстаток,
	ВЫБОР
	КОГДА ХозрасчетныйОстатки.Счет.Валютный
	ИЛИ НЕ &ВедетсяУчетПоВалютам
	ТОГДА 1
	ИНАЧЕ 0
	КОНЕЦ КАК Итог
	ПОМЕСТИТЬ Остатки
	ИЗ
	РегистрБухгалтерии.Хозрасчетный.Остатки(
	{(&Период)},
	Счет В
	(ВЫБРАТЬ
	СчетаУчета.Счет
	ИЗ
	СчетаУчета КАК СчетаУчета),
	,

Продолжение приложения А

```

|      {(Организация).*,      (Подразделение).*,      (Валюта).*)}      КАК
ХозрасчетныйОстатки
|;
|
|
|////////////////////
|ВЫБРАТЬ РАЗРЕШЕННЫЕ
|      Остатки.Организация КАК Организация,
|      Остатки.Подразделение КАК Подразделение,
|      Остатки.ВидДенежныхСредств КАК ВидДенежныхСредств,
|      Остатки.Размещение КАК Размещение,
|      Остатки.Валюта КАК Валюта,
|      Остатки.СуммаОстаток КАК СуммаОстаток,
|      Остатки.Итог КАК Итог,
|      Остатки.ВалютнаяСуммаОстаток КАК ВалютнаяСуммаОстаток
|{ВЫБРАТЬ
|      Организация.*,
|      Подразделение.*,
|      ВидДенежныхСредств,
|      Размещение,
|      Валюта.*,
|      СуммаОстаток,
|      Итог,
|      ВалютнаяСуммаОстаток}
|ИЗ
|      Остатки КАК Остатки
|{ГДЕ
|      Остатки.ВидДенежныхСредств,
|      Остатки.Размещение,
|      Остатки.Организация.*,
|      Остатки.Подразделение.*}";
Запрос.УстановитьПараметр("ВедетсяУчетПоВалютам", Ложь); // Пример значения
Запрос.УстановитьПараметр("Рубль", Справочники.Валюты.НайтиПоКоду("643")); // Код
рубля
```

Продолжение приложения А

Запрос.УстановитьПараметр("Период", Дата(ТекущаяДата())); // Укажите нужный период

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

Пока Результат.Следующий() Цикл

 СуммаОстатка = СуммаОстатка + Результат.ВалютнаяСуммаОстаток;

КонецЦикла;

ОстатокСредствНаТекущуюДату = СуммаОстатка;

//++ Заявок создано за прошлый месяц

ЗапросЗаявки = Новый Запрос;

ЗапросЗаявки.Текст = "ВЫБРАТЬ РАЗЛИЧНЫЕ

| СУММА(бит_ЗаявкаНаРасходованиеСредств.Сумма) КАК Сумма

| ИЗ

| Документ.бит_ЗаявкаНаРасходованиеСредств

КАК

бит_ЗаявкаНаРасходованиеСредств

| ГДЕ

| бит_ЗаявкаНаРасходованиеСредств.Дата

МЕЖДУ

&ДатаНачала

И

&ДатаОкончания

| И бит_ЗаявкаНаРасходованиеСредств.Проведен";

ПрошлыйМес = ДобавитьМесяц(ТекущаяДата(), -1);

ЗапросЗаявки.УстановитьПараметр("ДатаНачала", НачалоМесяца(ПрошлыйМес));

ЗапросЗаявки.УстановитьПараметр("ДатаОкончания", КонецМесяца(ПрошлыйМес));

РезультатЗаявки = ЗапросЗаявки.Выполнить().Выбрать();

РезультатЗаявки.Следующий();

ЗаявокНаПрошлМес = РезультатЗаявки.Сумма;

//-- Заявок создано на прошлый месяц

КонецПроцедуры

&НаКлиенте

Процедура ПриОткрытии(Отказ)

 ПриОткрытииНаСервере();

КонецПроцедуры

Функция ОтправкаСообщений(ТекстТема, ТекстТело)

Продолжение приложения А

```
УчетнаяЗаписьОтправителя                                     =
РаботаСПочтовымиСообщениями.СистемнаяУчетнаяЗапись();
    ПараметрыПисьма = Новый Структура;
    ПараметрыПисьма.Вставить("Кому",                          "a.gataullov@yesapart.com,
testtest321@yesapart.com");
    ПараметрыПисьма.Вставить("УчетнаяЗапись", УчетнаяЗаписьОтправителя);
    ПараметрыПисьма.Вставить("АдресОтвета", "a.gataullov@yesapart.com");
    ПараметрыПисьма.Вставить("Тема", ТекстТема);
    ПараметрыПисьма.Вставить("Тело", ТекстТело);
    ПараметрыПисьма.Вставить("ТипТекста", "ПростойТекст");

    РаботаСПочтовымиСообщениями.ОтправитьПочтовоеСообщение(УчетнаяЗаписьО
тправителя, ПараметрыПисьма);

КонецФункции
```

Приложение Б

Листинг модуля объекта разрабатываемого модуля

Функция СведенияОВнешнейОбработке() Экспорт

```
    ПараметрыРегистрации =  
ДополнительныеОтчетыИОбработки.СведенияОВнешнейОбработке("2.2.2.1");  
  
    ПараметрыЭтойОбработки = ПолучитьПараметрыЭтойОбработки(Ложь);  
  
    ПараметрыРегистрации.Вид =  
ДополнительныеОтчетыИОбработки.КлиентСервер.ВидОбработкиДополнительнаяОбрабо  
тка());  
    ПараметрыРегистрации.Информация = ПараметрыЭтойОбработки.Комментарий  
    + "(ЗначениеЗаполнено(ПараметрыЭтойОбработки.Комментарий), Символы.ПС, "")  
    + "Дата создания: 10.12.2024." + Символы.ПС  
    + "Дата последнего редактирования: 28.12.2024.";  
    ПараметрыРегистрации.Версия = "1.0.0.2";  
    ПараметрыРегистрации.БезопасныйРежим = Истина; // отключаем безопасный  
    режим только при необходимости  
  
    НоваяКоманда = ПараметрыРегистрации.Команды.Добавить();  
    НоваяКоманда.Представление = ПараметрыЭтойОбработки.Синоним + "  
Копирование заявок";  
    НоваяКоманда.Идентификатор = ПараметрыЭтойОбработки.Имя + "  
Копирование заявок";  
    НоваяКоманда.Использование =  
ДополнительныеОтчетыИОбработки.КлиентСервер.ТипКомандыОткрытиеФормы();  
  
    Возврат ПараметрыРегистрации;  
  
КонецФункции
```

Функция ПолучитьПараметрыЭтойОбработки(ВызыватьИсключение = Истина) Экспорт

Продолжение приложения Б

ИмяОбработки = ЭтотОбъект.Метаданные().Имя;

СинонимОбработки = ЭтотОбъект.Метаданные().Синоним;

КомментарийОбработки = ЭтотОбъект.Метаданные().Комментарий;

Ссылка

=

Справочники.ДополнительныеОтчетыИОбработки.НайтиПоРеквизиту("ИмяОбъекта",
ИмяОбработки);

Если Ссылка.Пустая() Тогда

Если ВызыватьИсключение Тогда

ВызыватьИсключение "Данная обработка не зарегистрирована в
справочнике внешних обработок!";

КонецЕсли;

КонецЕсли;

ПараметрыЭтойОбработки = Новый Структура;

ПараметрыЭтойОбработки.Вставить("Ссылка", Ссылка);

ПараметрыЭтойОбработки.Вставить("Имя", ИмяОбработки);

ПараметрыЭтойОбработки.Вставить("Синоним",

?(ЗначениеЗаполнено(СинонимОбработки), СинонимОбработки, ИмяОбработки));

ПараметрыЭтойОбработки.Вставить("Комментарий", КомментарийОбработки);

Возврат ПараметрыЭтойОбработки;

КонецФункции