**БУ ВО**

**«СУРГУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Ханты-Мансийского автономного округа – Югры»**

**Политехнический институт**

Кафедра: «Автоматизированные системы обработки информации и управления»

**Бажаев арман бейсембаевич**

**АВТОМАТИЗИРОВАННАЯ СИСТЕМА ОПРЕДЕЛЕНИЯ ОПТИМАЛЬНОГО ВРЕМЕНИ ПРОВЕДЕНИЯ РЕГЛАМЕНТНОГО ОБСЛУЖИВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ НА ОСНОВЕ ДАННЫХ ТЕХНИЧЕСКОГО МОНИТОРИНГА**

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

на соискание академической степени магистра по направлению

09.04.01 «Информатика и вычислительная техника»

Магистерская программа «Информационное и программное обеспечение автоматизированных систем»

Научный руководитель

магистерской программы

д.т.н., профессор, Бушмелева Кия Иннокентьевна.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

Научный руководитель

к.т.н., Яценко Елена Александровна.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

Рецензент

и.о. начальника отдела ОАПО SAP

Шаяхметов Руслан Фаритович.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

Магистрант группы № 606-71м

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

Сургут 2019 г.

# **РЕФЕРАТ**

Выпускная квалификационная работа содержит 97 страниц, 31 рисунок, 13 таблиц, 7 формул и 32 использованных источника.

БИЗНЕС-ПРОЦЕСС, КОРПОРАТИВНАЯ ИНФОРМАЦИОННАЯ СИСТЕМА, SAP, ВРЕМЯ РЕАКЦИИ, ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ, ПРОИЗВОДИТЕЛЬНОСТЬ СИСТЕМЫ.

В данной работе представлены описания разработки алгоритмов, методов и средств расчета сроков регламентного обслуживания информационных систем SAP. Проведено обследование предметной области и аналогов разрабатываемой системы. Выполнен поиск и анализ средств автоматизированного сбора данных производительности SAP систем. Разработаны алгоритмы и методы расчета сроков технического обслуживания SAP-систем. Разработана организационно-функциональная модель системы. Спроектирована концептуальная модель базы данных АС. Разработан интерфейс АС.

# **СПИСОК СОКРАЩЕНИЙ**

АС – автоматизированная система;

БД – база данных;

КИС – корпоративная информационная система;

SAP - System Applications and Products in Data Processing;

HANA – High-Performance Analytic Appliance;

ТОИР – техническое обслуживание и ремонт;

GUI – графический пользовательский интерфейс;

ЦПУ – центральное процессорное устройство;

SSH – secure shell.

**СОДЕРЖАНИЕ**

[РЕФЕРАТ 2](#_Toc12691883)

[СПИСОК СОКРАЩЕНИЙ 3](#_Toc12691884)

[ВВЕДЕНИЕ 6](#_Toc12691885)

[1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 9](#_Toc12691886)

[1.1. Техническое обслуживание КИС 9](#_Toc12691887)

[1.2. Архитектура систем SAP 10](#_Toc12691888)

[1.3. Архитектура SAP HANA 13](#_Toc12691889)

[1.4. Время реакции информационной системы 15](#_Toc12691890)

[2. ОБЗОР АНАЛОГОВ 20](#_Toc12691891)

[2.1. SAP PM 20](#_Toc12691892)

[2.2 1С:ТОИР 21](#_Toc12691893)

[2.3. IBM Maximo 23](#_Toc12691894)

[2.4. Выводы по обзору аналогов 24](#_Toc12691895)

[3. ИСТРУМЕНТЫ МОНИТОРИНГА ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМ SAP 26](#_Toc12691896)

[3.1. St03 26](#_Toc12691897)

[3.2. SMON 27](#_Toc12691898)

[3.3. top 28](#_Toc12691899)

[3.4. Параметры производительности SAP систем 29](#_Toc12691900)

[4. ОПРЕДЕЛЕНИЕ ПАРАМЕТРОВ АНАЛИЗА ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМ 31](#_Toc12691901)

[4.1. Параметры анализа загрузки БД систем 31](#_Toc12691902)

[4.2. Параметры анализа загрузки сервера приложений 35](#_Toc12691903)

[4.3. Итоговый выбор параметров анализа производительности систем 37](#_Toc12691904)

[5. ОРГАНИЗАЦИОННО-ФУНКЦИОНАЛЬНАЯ СХЕМА СИСТЕМЫ 39](#_Toc12691905)

[6. ПРОЕКТИРОВАНИЕ КОНЦЕПТУАЛЬНОЙ МОДЕЛИ БАЗЫ ДАННЫХ СИСТЕМЫ 41](#_Toc12691906)

[7. ВИДЫ ОБЕСПЕЧЕНИЯ СИСТЕМЫ 43](#_Toc12691907)

[7.1. Информационное обеспечение 43](#_Toc12691908)

[7.2. Математическое обеспечение 48](#_Toc12691909)

[7.3. Алгоритмическое обеспечение 50](#_Toc12691910)

[7.3.1. Алгоритм анализа статистических данных 50](#_Toc12691911)

[7.3.2. Алгоритм расчета сроков регламентного обслуживания 53](#_Toc12691912)

[7.4. Программное обеспечение 54](#_Toc12691913)

[7.4.1. SAP GUI Scripting 54](#_Toc12691914)

[7.4.2. SAP HANA Client 57](#_Toc12691915)

[7.4.3. Remote-exec 58](#_Toc12691916)

[7.4.4. Инструменты разработки графического пользовательского интерфейса 59](#_Toc12691917)

[7.4.4. Cервер автоматизированной системы 60](#_Toc12691918)

[8. ОПИСАНИЯ МЕТОДОВ ОБРАБОТКИ ЗАПРОСОВ ИМПОРТА И АНАЛИЗА ДАННЫХ СИСТЕМЫ 63](#_Toc12691919)

[9. ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС СИСТЕМЫ 64](#_Toc12691920)

[ЗАКЛЮЧЕНИЕ 69](#_Toc12691921)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 70](#_Toc12691922)

[ПРИЛОЖЕНИЯ](#_Toc12691923) [73](#_Toc12691924)

[ПРИЛОЖЕНИЕ 1](#_Toc12691923). [Листинг программы обработки действий по редактированию таблицы системных пользователей 73](#_Toc12691924)

[ПРИЛОЖЕНИЕ 2](#_Toc12691925). [Листинг программы импорта данных на прикладном уровне через транзакцию st03 76](#_Toc12691926)

[ПРИЛОЖЕНИЕ 3](#_Toc12691927). [Листинг программы импорта данных на уровне БД 79](#_Toc12691928)

[ПРИЛОЖЕНИЕ 4](#_Toc12691929). [Графический интерфейс системы 80](#_Toc12691930)

[ПРИЛОЖЕНИЕ 5](#_Toc12691931). [Листинг программы генерации календаря событий 92](#_Toc12691932)

[ПРИЛОЖЕНИЕ 6](#_Toc12691933). [Диаграмма процесса обработки запроса импорта статистических данных 93](#_Toc12691934)

[ПРИЛОЖЕНИЕ 7](#_Toc12691935). [Диаграмма процесса обработки запроса анализа статистических данных 94](#_Toc12691936)

[ПРИЛОЖЕНИЕ 8](#_Toc12691937). [Декомпозиционная диаграмма первого уровня процесса определения сроков технического обслуживания 95](#_Toc12691938)

# **ВВЕДЕНИЕ**

Бизнес-процесс – совокупность взаимосвязанных и взаимодействующих видов деятельности с участием человека, использующих входные данные для получения намеченного результата [1].

Корпоративная информационная система - информационная система, доступ к которой имеет ограниченный круг лиц, определенный ее владельцем или соглашением участников этой информационной системы (круг лиц могут составлять граждане и юридические лица) [2].

Основной задачей корпоративных информационных систем является быстрое и своевременное исполнение бизнес-процессов. Одним из факторов, способствующих возникновению критических ситуаций для достижения вышеописанной задачи, является техническое обслуживание. Техническое обслуживание – комплекс технологических операций и организационных действий по поддержанию работоспособности или исправности объекта при использовании по назначению, ожидании, хранении и транспортировании [3].

SAP (System Applications and Products in Data Processing) - компания, специализирующаяся на разработке систем управления по автоматизации бизнес-процессов классов бухгалтерского учета, торговли, финансов, управления персоналом и других направлений для крупных и средних предприятий [4]. Текущий метод определения сроков технического обслуживания ориентирован на анализ выполняющихся в системе бизнес-процессов, с последующей оценочной установкой времени проведения профилактических работ. Данный подход имеет недостатки, связанные с отсутствием анализа реальной нагрузки на системы. Современные программные средства класса технического обслуживания и ремонта (ТОИР) ориентированы на мониторинг и анализ производительности аппаратного обеспечения (SAP PM, 1C:ТОИР, IBM Maximo). Таким образом, решением задачи поиска временных сроков технического обслуживания является анализ статистических данных производительности систем при помощи прикладных технических инструментов SAP.

Актуальность данной работы заключается в повышении эффективности выполнения бизнес-процессов корпоративной информационной системы, сокращении времени поиска сроков технического обслуживания.

Целью выпускной квалификационной работы является автоматизация процесса расчета сроков проведения регламентных работ в корпоративных информационных системах SAP.

Объектом исследования является процесс расчета сроков технического обслуживания. Предметом исследования работы являются методы и средства определения времени технического обслуживания систем.

Для выполнения цели были поставлены следующие задачи:

1. Проанализировать функциональные возможности программных средств мониторинга SAP с целью определения множества технических параметров производительности систем.
2. Выявить и проанализировать критические состояния функционирования систем с целью определения множества целевых параметров производительности при расчете сроков регламентного обслуживания.
3. Разработать алгоритм анализа и корректировки статистических данных производительности систем.
4. Разработать алгоритм расчета сроков технического обслуживания.
5. Разработать метод получения и обработки значений параметров производительности систем SAP.
6. Спроектировать автоматизированную систему расчета сроков регламентного обслуживания систем SAP.

Научная новизна работы заключается в разработке метода, который позволяет получить время проведения технического обслуживания с более высокой точностью и наименьшей степенью влияния на бизнес-процессы предприятия.

Теоретическая значимость работы заключается в применении результатов исследования для решения прикладных задач администратора SAP, частью которых является автоматизированный сбор данных производительности систем на прикладном уровне, уровне ОС, уровне БД.

Практической значимостью работы является более точное и оперативное получение результатов расчета в процессе определения сроков регламентного обслуживания.

# **1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ**

## **1.1. Техническое обслуживание КИС**

Функциональная модель верхнего уровня процесса расчета срока проведения технического обслуживания представлена на рис. 1

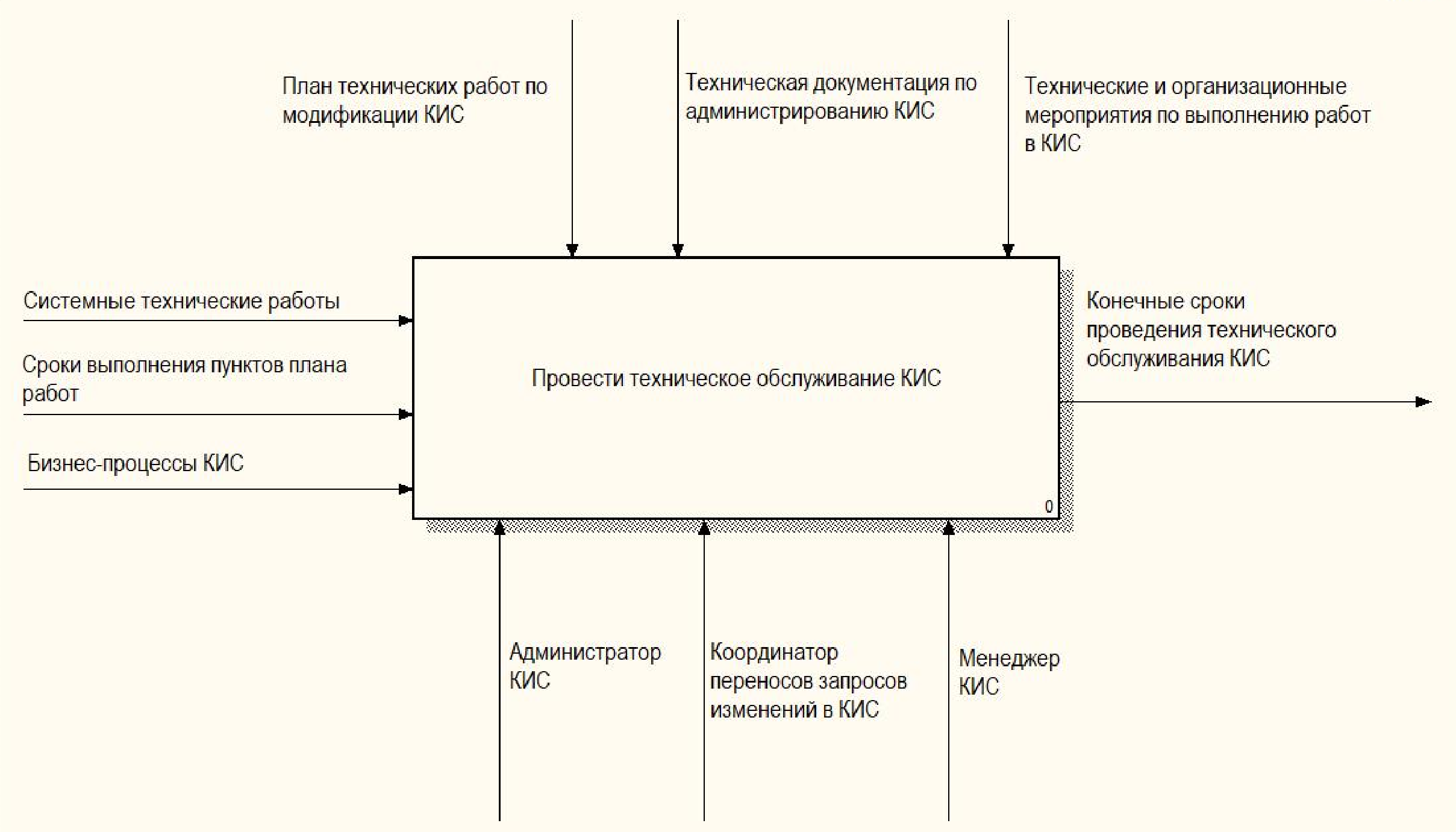


Рис. 1. Контекстная диаграмма процесса проведения технического обслуживания

Входными элементами блока являются системные технические работы, требуемые к выполнению в информационной системе в соответствии с установленным планом работ и сроки их выполнения; бизнес-процессы, определяемые на пересечении расчетов промежуточных сроков выполнения технического обслуживания.

Функциональная модель первого уровня представлена в приложении 8:

Диаграмма состоит из четырех блоков. Первый блок описывает процесс оценки объема последовательности действии для выполнения поставленных задач в соответствии с планом работ. Выходными данными блока являются определенный администратором корпоративной информационной системы алгоритм решения задач исполнения технических работ. Второй блок описывает процесс расчета ориентировочного времени исполнения работ в соответствии с указанным алгоритмом действий. Выходными данными блока являются первичные рассчитанные сроки выполнения запланированных работ. На этапе корректировки сроков технического обслуживания (3 блок декомпозиционной диаграммы) выполняется изменение временных промежутков менеджерами информационных систем на основании знаний принципов их функционирования и подробных деталей выполнения бизнес-процессов. Последний блок функциональной модели описывает этап согласования предложенных сроков с координатором переносов изменений для преобразования порядка проведения модификаций в корпоративных информационных системах. В случае возникновения противоречий координатор переносов совместно с менеджером информационной системы проводит корректировку значений.

## **1.2. Архитектура систем SAP**

Системы SAP спроектированы по классической трехкомпонентной архитектуре ERP-систем. В настоящее время эта технология успешно применяется во многих сложных системах ПО (рис. 2):

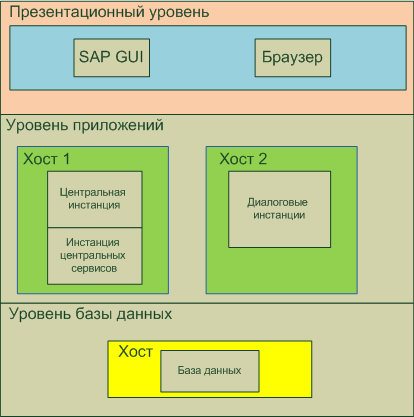


Рис. 2. Архитектура системы SAP

Системы спроектированы по трехуровневой архитектурной модели, предполагающей в них наличие трёх компонентов: клиентское приложение, сервер приложений и сервер базы данных [5].

С точки зрения ПО трехзвенная архитектура клиент/сервер состоит из презентационного уровня, уровня приложений и уровня БД. С точки зрения аппаратных средств эти уровни независимо функционируют на разных машинах или совместно на одном компьютере.

Инстанция – административная единица, которая объединяет системные компоненты SAP для обеспечения выполнения пользовательских процессов [6].

В отличие от презентационного уровня, где каждый компонент внешнего интерфейса работает независимо (возможно, на разных компьютерах), все процессы уровня приложений (которые также могут выполняться на разных машинах) образуют логически связанную единицу.

Уровень приложений в системе SAP предусматривает следующие сервисы: служба диалога, служба обновлений, служба управления блокировками, служба фоновой пакетной обработки, служба сервера сообщений, служба шлюза.

Сервер приложений инсталлирован на платформе SAP NetWeaver. SAP NetWeaver – сервисно-ориентированная программная платформа для развертывания приложений SAP Business Suite; построена на основе открытых спецификаций и отраслевых стандартов и может быть расширена путём использования технологий .NET и Java EE [7].

Архитектура NetWeaver представлена на рис. 3:

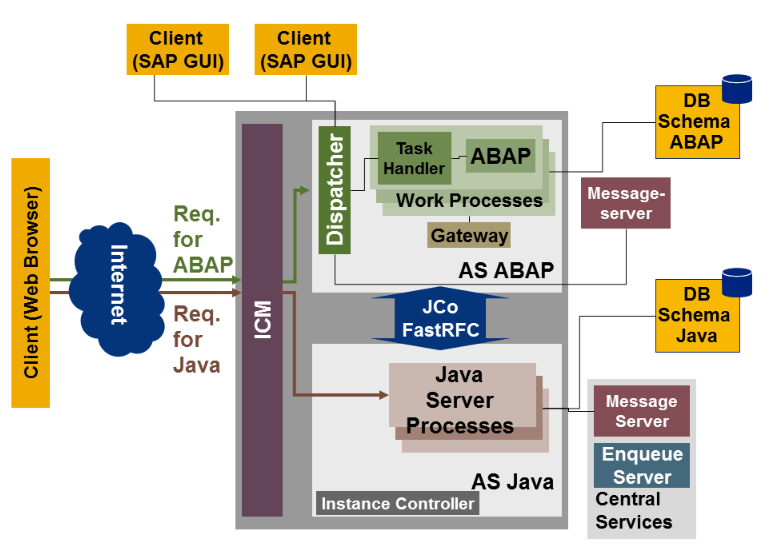


Рис. 3. Архитектура NetWeaver

Платформа состоит из следующих компонентов:

1. ICM (менеджер веб-запросов): обработка запросов сервером приложений по протоколам http, https, smtp;
2. Сервер Java: обработка JAVA-запросов;
3. Сервер ABAP: обработка ABAP-запросов;
4. Диспетчер: определение типа входящего ABAP-запроса;
5. Gateway: обработка запросов по RFC-соединениям;
6. Сервер сообщений: распределение пользовательских запросов по инстанциям с наилучшей производительностью;
7. Сервер блокировок: компонент системы SAP по управлению таблицей блокировок [8].

## **1.3. Архитектура SAP HANA**

SAP HANA – платформа для хранения и обработки данных и одноименная СУБД, в основе которой лежит технология вычислений in-memory с использованием поколоночного хранения данных [9].

Архитектура базы данных представлена на рис. 4:

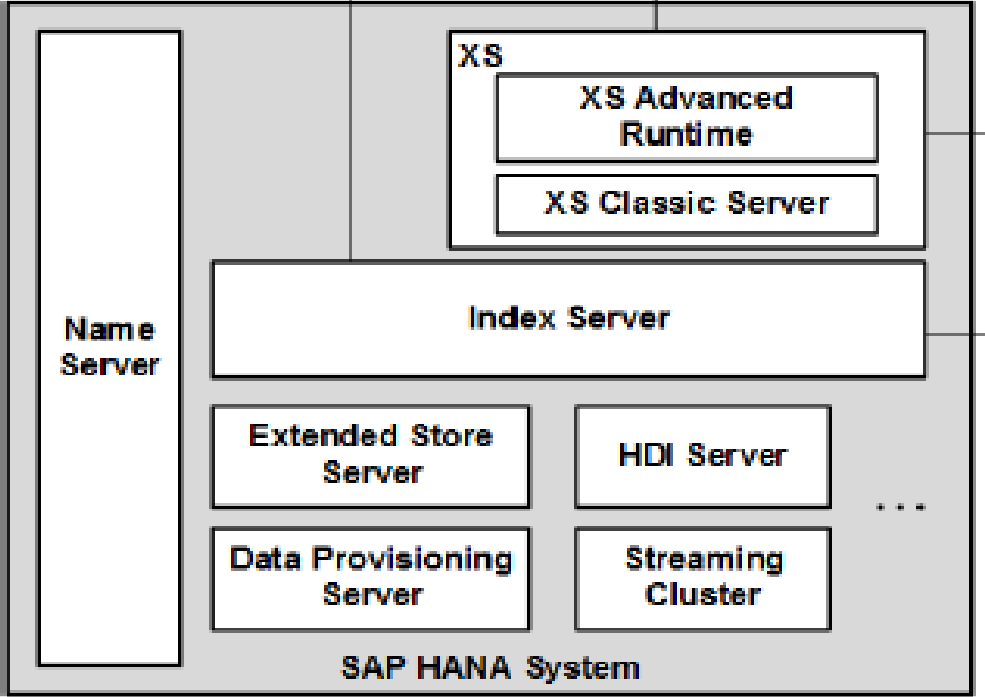


Рис. 4. Архитектура базы данных SAP HANA

SAP HANA состоит из нескольких серверов, основным из которых является index server. Серверные компоненты системы представлены в таблице 1:

**Таблица 1**

Компоненты SAP HANA

|  |  |
| --- | --- |
| **Компонент** | **Назначение** |
| Index server | обеспечивает управление сеансами, авторизацию, управление транзакциями и обработку команд; обеспечивает соответствие между файлами в постоянной памяти и оперативной памятью, содержащей кэшированные образы объектов БД и файлов журналов |
| Name server | содержит информацию о топологии системы SAP HANA. В распределенной системе с экземплярами базы данных SAP HANA на нескольких хостах на сервере имен хранится информация о рабочих компонентах и какие данные находятся на каком сервере. |
| XS advanced runtime, XS Classic server | XS advanced runtime обеспечивает функционирование развернутых на SAP HANA приложений. SAP HANA XS Classic server – сервер развертывания веб-приложений; может являться как автономным, так и встроенным компонентом index-сервера. |
| Preprocessor server | для анализа текстовых данных и извлечения информации, на которой основаны возможности текстового поиска[10]. |
| Extended store server | сервер динамического размещения данных; обеспечивает высокую производительность дискового хранилища данных в размере до петабайта |

|  |  |
| --- | --- |
| **Компонент** | **Назначение** |
| Data Provisioning server | cервер ведения пользовательских данных и разработки адаптеров |

SAP HANA поддерживает распределение своих серверных компонентов по нескольким хостам с целью масштабируемости и доступности.

Распределенная система SAP HANA - это система, которая установлена ​​на нескольких хостах. В противном случае это система с одним хостом. Основная причина распределения системы по нескольким хостам - масштабирование. Система с несколькими узлами может преодолеть аппаратные ограничения одного физического сервера и может распределять нагрузку между несколькими серверами. В системе с несколькими хостами каждому index-серверу обычно назначается собственный хост для максимальной производительности. Можно назначать разные таблицы разным хостам (разделять базу данных), разбивать одну таблицу между хостами (разбивать таблицы) и реплицировать таблицы на несколько хостов.

Система SAP HANA, установленная на нескольких хостах, идентифицируется одним идентификатором системы (SID). Это воспринимается как единое целое с точки зрения администратора, который может установить, обновить, запустить, завершить работу или выполнить резервное копирование системы в целом. Различные серверные компоненты системы совместно используют одни и те же метаданные, и запросы от клиентских приложений могут прозрачно отправляться на разные серверы в системе.

## **1.4. Время реакции информационной системы**

Выделяют три типа SAP-запросов:

* 1. ICM: URL-запросы;
  2. RFC-запросы(Gateway процессы): внешние запросы, данные которых передаются через сконфигурированное RFC-соединение;
  3. SAP GUI: запросы пользователей через десктопную программу SAP Logon [11].

При входе в SAP-систему пользователь занимает небольшой объем расширенной памяти, в котором хранится начальный контекст пользователя. В данном объеме памяти хранятся данные полномочий, значения SET/GET параметров. Во время работы пользователя в системе к его контексту добавляются данные необходимые для работы транзакций, например, номера документов.

Все запросы от пользователей (шаги диалога) попадают в очередь к диспетчеру рабочих процессов. Для выполнения шага диалога рабочему процессу необходимы данные текущего пользователя - его контекст.

Основным параметром мониторинга технического состояния информационной системы является её время отклика (реакции). Согласно документации SAP, время отклика – интервал времени, измеряемый на сервере приложений, который начинает отсчет с момента приема диспетчером системы запроса и моментом отправки окончательных данных на пользовательский терминал[12]. Время соединения пользователя с сервером приложений не включается в данный параметр.

Структурные компоненты времени отклика представлены на рис. 5:

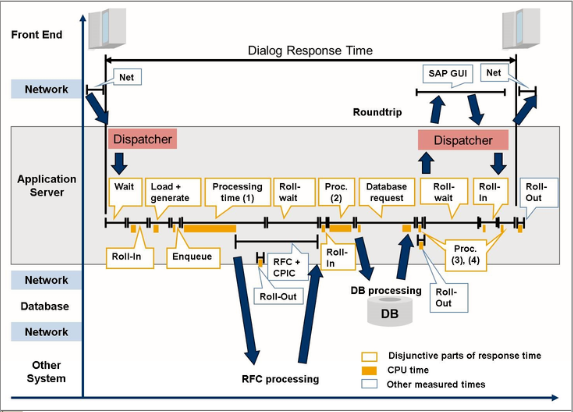


Рис. 5. Время отклика

Время отклика состоит из следующих компонентов:

1. Время ожидания: диспетчер сервера приложений SAP принимает входящий запрос и добавляет в соответствующую очередь процессов по типу (диалоговый, фоновый, процесс-спул, обновление, процесс управления блокировками); когда свободный процесс найден, время ожидания завершается; как правило, значение данного параметра занимает несколько миллисекунд;
2. Roll-in время: время копирования контекста пользователя из расширенной памяти в память рабочего процесса [13];
3. Время загрузки, генерации программ и экранов;
4. Время выполнения;
5. Время отклика БД: интервал времени между моментом отправки запроса к базе данных системы и моментом приема сервером приложений выходных данных из буфера;
6. Roll-out время: в конце диалогового шага, рабочий процесс выводит данные пользовательского контекста обратно из roll-буфера, параллельно выводу диспетчера системы отправляет данные на фронтенд[14];
7. Процессорное время: суммарное время использования ресурсов центрального процессора в течение диалогового шага;
8. Roll-wait время: время ожидания пользовательского контекста в случаях RFC-вызовов во внешние системы и промежуточных действий пользователя [15] (рис. 6):

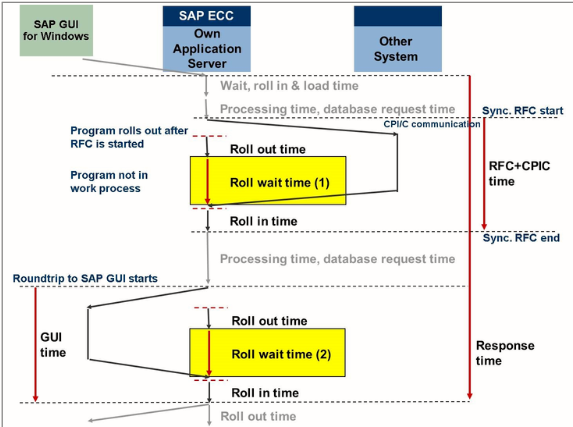


Рис. 6. Структурная взаимосвязь компонентов времени реакции roll out, roll wait, roll in

Высокое время отклика возникает в следующих ситуациях:

1. Высокое время ожидания: нехватка свободных рабочих процессов;
2. Высокое roll-in время: переполнение памяти;
3. Высокое время загрузки и генерации: требуется расширение размера SAP буферов;
4. Высокое время отклика БД: высокая загрузка БД, сетевые проблемы, блокировки;
5. Время обработки больше чем в два раза процессорного времени: высокая загрузка ЦПУ [11].

Для анализа высокого времени выполнения программ, транзакций в SAP-системе разработаны программы, позволяющие проводить расширенный анализ разных временных ситуаций.

# **2. ОБЗОР АНАЛОГОВ**

Современное ПО по техническому обслуживанию специализируется на анализе состояния производительности оборудования. Ввиду сложности архитектуры современных КИС модули ТОРО разрабатываются под определенную программную платформу ИТ-инфраструктур. Пользователями программ данного класса являются работники сферы технического обслуживания, служб материально-технического снабжения и производственных подразделений.

## **2.1. SAP PM**

SAP PM – программный модуль компании SAP по ведению технического состояния и ремонта оборудования, внедряемый в систему SAP ERP.

Пользовательский интерфейс системы представлен на рис. 10:

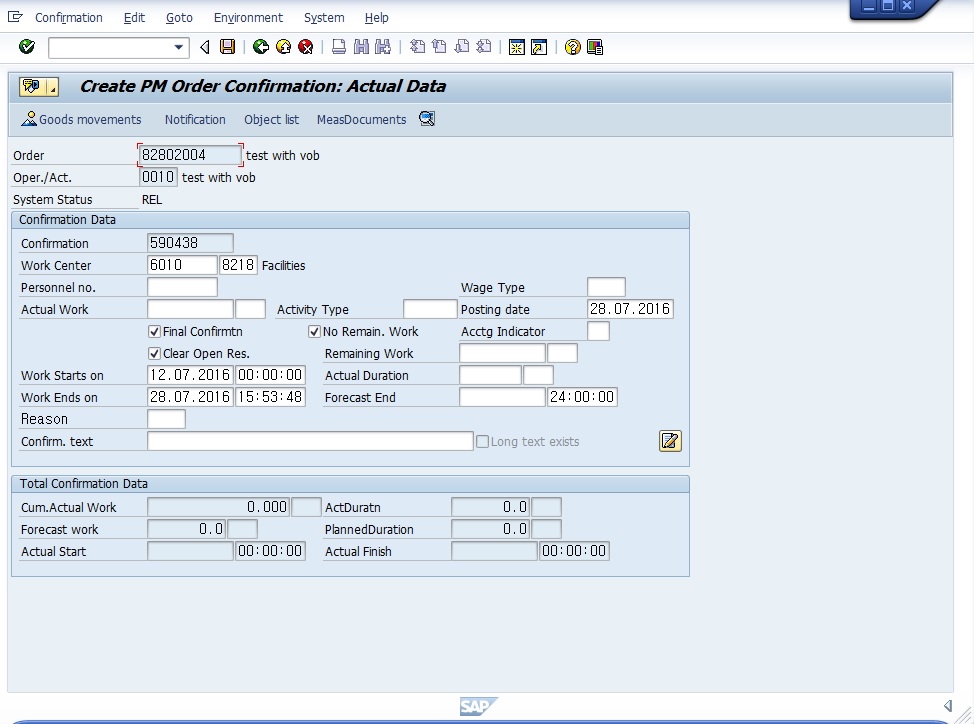


Рис. 10. Пользовательский интерфейс программы по ведению заказа модуля SAP PM

Функционал SAP PM позволяет пользователю выполнять следующие операции:

1. Автоматизация ведения данных по оборудованию (единицы оборудования, классификация, спецификация технических объектов);
2. Паспортизация оборудования;
3. Расчет потребности в материалах, запасных частях;
4. Учет и контроль выполнения ремонтных работ;
5. Ведение управления закупками [18].

Использование модуля в ИТ-инфраструктуре SAP имеет следующие преимущества:

1. Ведение большого числа множества бизнес-процессов технического обслуживания и ремонта аппаратного обеспечения предприятий;
2. Возможность доработки функционала программного обеспечения.

Недостатки:

1. Дороговизна разработки ИТ-решений;
2. Дороговизна сопровождения.

## **2.2 1С:ТОИР**

Интерфейс программы представлен на рис. 11:

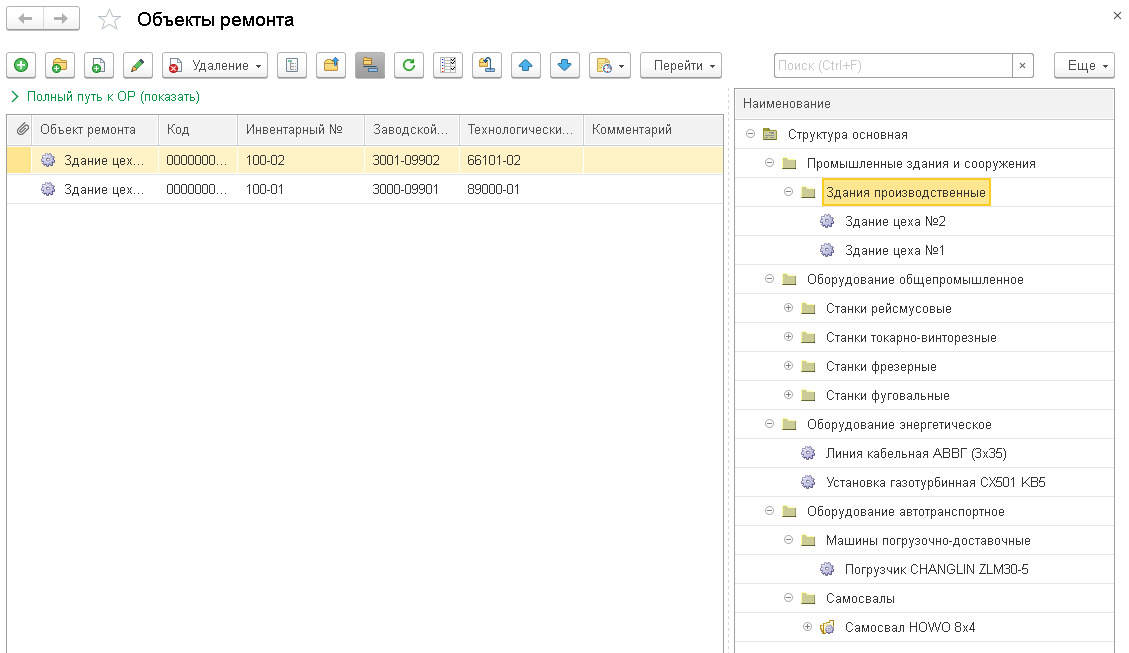


Рис. 11. Пользовательский интерфейс программы 1С:ТОИР

1C: ТОИР разработан для разработки и ведения бизнес-процессов для малых и средних предприятий.

1С: ТОИР поддерживает следующие бизнес-процессы:

1. Учет оборудования и нормативов: ведение информации об оборудовании предприятия, организация ремонтных служб оперативного доступа к необходимым в работе документам и истории ремонтов оборудования;
2. Ведение показателей эксплуатации: сбор, хранение и анализ сведений о техническом состоянии оборудования во время его эксплуатации;
3. Управление материально-техническим обеспечением ремонтов: потребность предприятий в материалах, инструментах и оборудовании.
4. Управление персоналом: определение численности необходимого ремонтного и эксплуатационного персонала;
5. Управление нарядами и работами: формирование и учет выполнения нарядов на ремонтные работы и регламентные мероприятия, при необходимости формирование нарядов-допусков к ним, а также контроль степени выполнения ремонтных работ и учет выполненных ТО, ремонтов и регламентных мероприятий.
6. Мониторинг показателей процессов (KPI) и формирование отчетности различной целевой направленности: отслеживание количественно определенных целей путем анализа показателей эффективности использования или обслуживания оборудования [19].

Преимуществом использования 1С: ТОИР являются:

1. Интеграция разработок под российские стандарты по сравнению с SAP PM;
2. Дешевизна и относительно небольшой период внедрения ИТ-решений.

Недостатком модуля является плохая масштабируемость и низкая отказоустойчивость.

## **2.3. IBM Maximo**

IBM Maximo Asset Management – программный комплекс управления физическими ресурсами на общей платформе для различных отраслей, использующих активы (рис. 12). С точки зрения бизнеса, активы – ресурсы, которые контролируются предприятием, использование которых в перспективе приводит к получению экономической прибыли.



Рис. 12. Сервис-ориентированная архитектура IBM Maximo

В состав IBM Maximo Asset Management входят следующие компоненты:

* 1. Управление активами – все функции и инструменты, необходимые для тщательного отслеживания и эффективного управления данными о корпоративных активах и их размещении на протяжении всего их жизненного цикла.
  2. Управление работами – управление работами по плановому и внеплановому обслуживанию активов – от генерации заявок и нарядов на проведение работ до регистрации фактически осуществленных мероприятий.
  3. Управление сервисами – определение предложений по обслуживанию, установление соглашений об уровне обслуживания (SLAs), более активное отслеживание уровня предоставляемых услуг и внедрение процедур эскалации.
  4. Управление договорами – обеспечение полной поддержки договоров купли-продажи, аренды, лизинга, гарантийных, трудовых договоров, договоров на поставку программного обеспечения, главных/групповых договоров, комплексных договоров, а также специфических видов договоров, определяемых пользователем.
  5. Управление материальными запасами – доступ к полной информации о связанных с активами материальных запасах и их использовании.
  6. Управление закупками – поддержка всех операций снабжения в масштабах предприятия, включая прямые закупки и пополнение товарно-материальных запасов [20].

Альтернативной системой для IBM Maximo в решениях SAP является программный комплекс SAP EAM (Управление активами предприятия). Данный комплекс содержит SAP PM, рассмотренный ранее.

## **2.4. Выводы по обзору аналогов**

Результаты сравнения представлены в таблице 3:

**Таблица 3**

Сравнение программных комплексов технического обслуживания и ремонта

|  |  |  |  |
| --- | --- | --- | --- |
| **Система/критерии** | **SAP PM / SAP EAM** | **1C:ТОИР** | **IBM Maximo** |
| Стоимость | Высокая | Низкая | Высокая |
| Отказоустойчивость | Высокая | Низкая | Высокая |
| Масштабируемость | Высокая | Средняя | Высокая |
| Длительность периода внедрения | До 5 лет | До 3 лет | До 3 лет |

|  |  |  |  |
| --- | --- | --- | --- |
| **Система/критерии** | **SAP PM / SAP EAM** | **1C:ТОИР** | **IBM Maximo** |
| Поддержка оптимизации бизнес-процессов | + | - | - |

По сравнению с IBM Maximo, 1C:ТОИР внедрение SAP EAM/SAP PM является более дорогим решением. SAP EAM поддерживает глубокую интеграцию с Интернетом-вещей по сравнению с IBM Maximo, 1C:ТОИР (поддержка мобильной интеграции) и имеет более расширенный набор инструментальных средств ведения бизнес-процессов предприятия (машинное обучение и различная операционная аналитика для создания сложных моделей данных о производительности оборудования). IBM Maximo, 1C:ТОИР предназначены преимущественно для централизованного ведения материальных активов и работ предприятия.

# **3. ИСТРУМЕНТЫ МОНИТОРИНГА ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМ SAP**

## **3.1. St03**

Монитор рабочей нагрузки (транзакция St03) – инструмент для мониторинга производительности систем SAP.

Интерфейс представлен на рис 7:

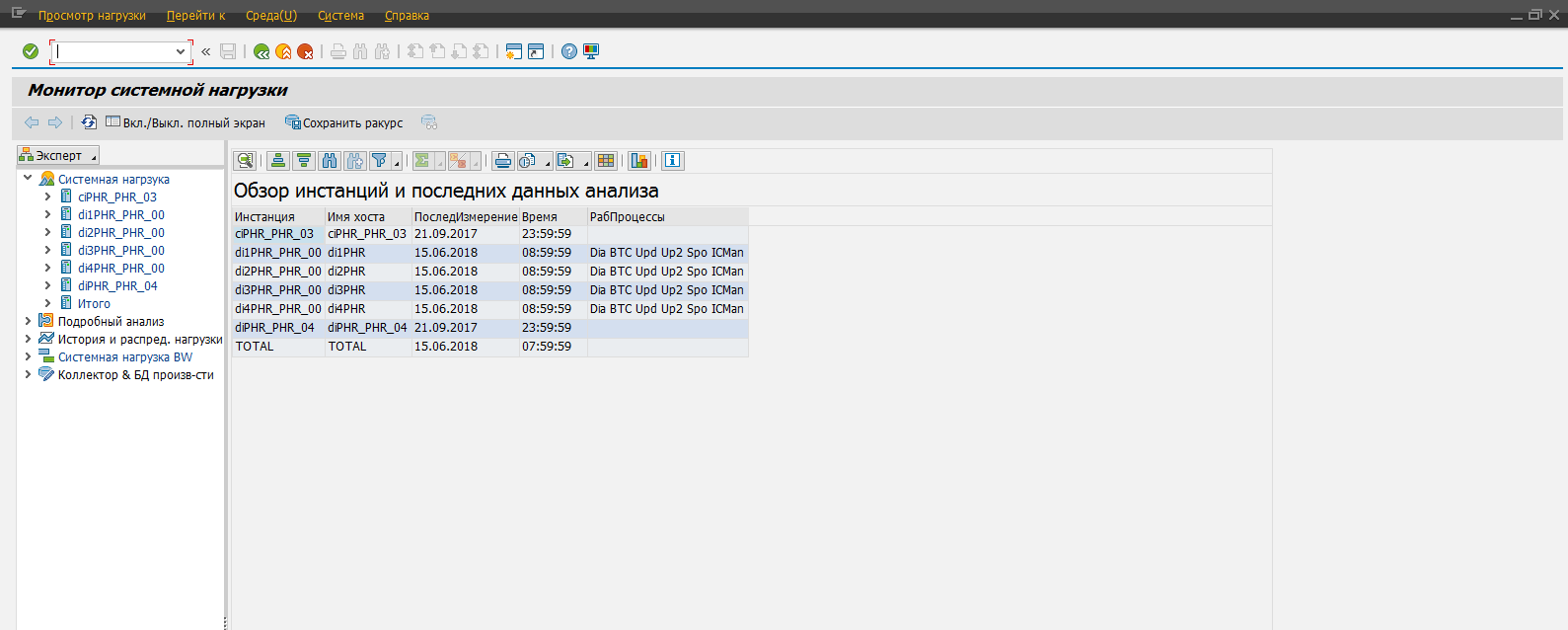


Рис. 7. Пользовательский интерфейс программы st03

Транзакция позволяет проводить следующие типы анализа:

1. Анализ системной нагрузки: позволяет получить данные рабочей нагрузки как по отдельным инстанциям, так и итоговую сводку по нагрузке системы в целом; на каждом из уровней SAP предоставляет отчетность по трем периодам: анализ однодневной рабочей нагрузки, анализ недельной рабочей нагрузки, анализ месячной рабочей нагрузки; инструмент позволяет получить состояние рабочей нагрузки в течение выбранного периода времени;
2. Подробный анализ: позволяет получить данные в течение дня за конкретный промежуток времени;
3. История и распределение нагрузки: проведение сравнения и анализа производительности по нескольким периодам;
4. Коллектор и производительность базы данных: позволяет редактировать периоды сбора и хранения информации о производительности БД; также является средством формирования аналитических данных для инструмента STAD [11].

В программе существует возможность изменения ракурсов отображения распределения системной нагрузки:

1. Временной профиль: вывод статистических данных по определенному типу задачи, сегментированный в соответствии с пользовательской выборкой по времени;
2. Статистика потребления памяти: вывод ранжированного списка программ, транзакций по памяти;
3. Статистика пользователей: журнал запущенных программ, транзакций в системе;
4. Статистика спула;
5. Распределение времени отклика по элементам системы (пользовательский терминал, сервер приложений, БД);
6. Статистика нагрузки на внешние системы;
7. Транзакционный профиль программы позволяет вывести статистическую информацию по активности использования программ, распределению времени отклика между разными типами задач и др.

## **3.2. SMON**

SMON – технический инструмент мониторинга снимков производительности сервера приложений SAP, выполняет сбор различных типов системной информации через заранее определенные интервалы времени (рис. 8):

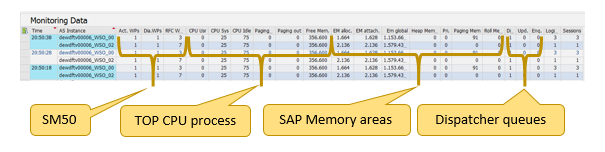


Рис. 8. Отчет аппаратной загрузки на инстанциях

Все параметры делятся на 4 группы:

1. Информация о численности процессов на инстанциях (свободные, диалоговые, rfc-процессы);
2. Загрузка ЦПУ (Top CPU Process);
3. Загрузка памяти (SAP Memory areas);
4. Информация об очередях запросов (Dispatcher queues); допускается возможность более детально проанализировать данные по очередям, выбрав опцию «Global Data».

SMON работает путем запуска технических сборщиков данных, по одному на сервер приложений. Коллекторы находятся в режиме ожидания (WAIT UP TO ...) между моментальными снимками производительности системы и запускаются как RFC-вызовы в асинхронном соединении без результата. Собранные снимки хранятся в системных таблицах [16].

Техническое средство позволяет получать информацию о распределении ресурсов системы (ЦПУ, память) по отчетам, программам, транзакциям и на основании этих данных производить оценку производительности сервера приложений системы.

## **3.3. top**

Мониторинг нагрузки базы данных SAP HANA с уровня ОС выполняется при помощи консольной linux-команды top.

top – монитор, выводящий список активных процессов и значения параметров нагрузки каждого отдельного процесса на сервер с частотой обновления 2 секунды [17]. Пример представлен на рис. 9:

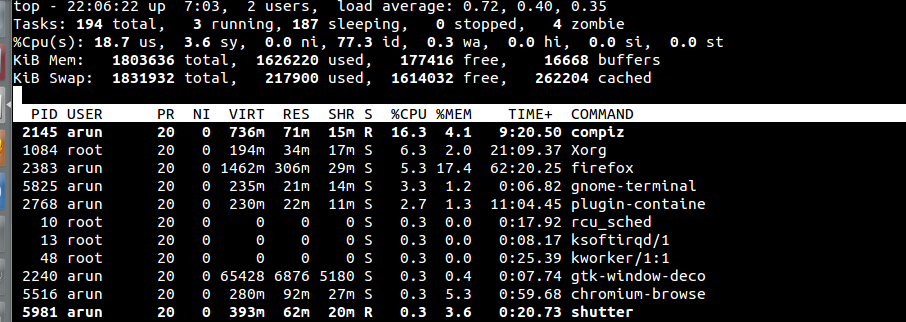


Рис. 9. Монитор нагрузки top

Среди параметров выделим необходимые из них для описания производительности сервера базы данных SAP HANA (табл. 2):

**Таблица 2**

Параметры top

|  |  |
| --- | --- |
| **Параметр** | **Описание** |
| USERNAME | Владелец процесса |
| %CPU | Загрузка ЦПУ в процентах |
| %MEM | Загрузка по памяти в процентах |
| COMMAND | Имя функционального элемента базы данных |

Среди процессов SAP HANA основным процессом мониторинга является hdbindexserver (компонент indexserver), таким образом, анализ производительности БД сводится к мониторингу нагрузки на процесс сервера indexserver-а.

## **3.4. Параметры производительности SAP систем**

В результате изучения технических средств мониторинга производительности систем SAP были определены следующие технические параметры:

1. St03
2. Общее время обработки (с);
3. Сумм. время ЦП (с);
4. Общее время ожидания roll (с);
5. Суммарное время ожидания (с);
6. Сумм. сетевое время фронтэнда (с);
7. Суммарное GUI-время (с);
8. Число шагов (диалога);
9. Число roundtrips;
10. Число последовательных считываний;
11. Общее время последовательных считываний (с);
12. Число прямых считываний;
13. Число логических изменений БД;
14. Общее время для логических изменений БД (с);
15. Общее время процедур БД;
16. Число логических вызовов БД;
17. Физические доступы к БД для чтения;
18. Число модифицированных записей БД;
19. Число считанных записей из буфера;
20. Число вызовов процедур БД.
21. SMON
22. Потребление ЦПУ (польз.) (%);
23. Число активных процессов;
24. Потребление памяти (%).
25. Top
26. MEM (%);
27. USERNAME;
28. COMMAND;

3.2. CPU (%).

Корректировка данного множества параметров выполняется на этапе анализа критических состояний систем.

# **4. ОПРЕДЕЛЕНИЕ ПАРАМЕТРОВ АНАЛИЗА ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМ**

Для выполнения задачи расчета сроков регламентного обслуживания требовалось выполнить анализ технического состояния систем в период их эксплуатации, с целью определения набора параметров для расчета искомых значений.

Для поиска параметров были использованы следующие технические программные средства SAP и Linux:

1. st03;
2. sdf/mon;
3. top.

В течение периода, равный одному году, за который проводился анализ, были зафиксированы следующие системные критические состояния:

* 1. Высокое значение потребления ЦПУ;
  2. Высокое значение времени отклика (больше одной секунды).

Для поиска необходимых параметров был использован математический инструмент корреляционного анализа.

## **4.1. Параметры анализа загрузки БД систем**

Определение параметров анализа производительности было выполнено при помощи сравнения коэффициентов корреляции времени реакции с остальными параметрами моментов времени, при котором система функционировала в штатном режиме, т.е. время реакции системы находилось в пределах допустимых значений, и времени, при котором было зафиксировано критическое состояние. Для расчета коэффициентов корреляции технических параметров при работе системы в штатном режиме были использованы статистические данные за день с шагом 1 минуты.

Рассмотрим коэффициенты корреляции для первого случая (табл. 4):

**Таблица 4**

Коэффициенты корреляции значений технических параметров со временем реакции системы

|  |  |
| --- | --- |
| **Параметр** | **Значение** |
| общее время обработки | 0,95 |
| сумм. Время ЦП | 0,94 |
| общее время ожидания roll (с) | 0,15 |
| суммарное время ожидания (с) | 0,21 |
| суммарное сетевое время фронтенда (с) | 0,07 |
| суммарное GUI-время | 0,37 |
| число шагов диалога | 0,55 |
| число roundtrips | 0,03 |
| число последовательных считываний | 0,01 |
| общее время последовательных считываний (с) | 0,05 |
| число прямых считываний | 0,93 |
| число логических изменений БД | 0,02 |
| число логических вызовов БД | 0,93 |
| физические доступы к БД для чтения | 0,10 |
| число модифицированных записей БД | 0,02 |
| число считанных записей из буфера | 0,93 |
| число вызовов процедур БД | 0,17 |
| общее время для логических изменений БД | 0,04 |
| общее врем. процедур БД | 0,15 |

Если значение времени реакции системы находится в пределах допустимой нормы:

1. Значение коэффициента корреляции со значениями параметров общего времени ожидания roll, GUI-времени низкое, т.е. данные пользовательского контекста не хранятся долгое время в памяти системы;
2. Значение коэффициента корреляции со значениями времени обработки, суммарного времени ЦП, обработки данных в БД больше 0,9, что означает отсутствие задержек при обработке запросов пользователей.

Рассмотрим ситуацию в высокой нагрузкой БД системы (табл.5):

**Таблица 5**

Коэффициенты корреляции значений технических параметров со временем реакции системы

|  |  |
| --- | --- |
| **Параметр** | **Значение** |
| общее время обработки | 0,76 |
| сумм. Время ЦП | 0,82 |
| общее время ожидания roll (с) | 0,73 |
| суммарное время ожидания (с) | 0,04 |
| суммарное сетевое время фронтенда (с) | 0,78 |
| суммарное GUI-время | 0,20 |
| число шагов диалога | 0,01 |
| число roundtrips | 0,70 |
| число последовательных считываний | 0,19 |
| общее время последовательных считываний (с) | 0,97 |
| число прямых считываний | 0,33 |

|  |  |
| --- | --- |
| **Параметр** | **Значение** |
| число логических изменений БД | 0,15 |
| число логических вызовов БД | 0,22 |
| физические доступы к БД для чтения | 0,07 |
| число модифицированных записей БД | 0,15 |
| число считанных записей из буфера | 0,37 |
| число вызовов процедур БД | 0,25 |
| общее время для логических изменений БД | 0,94 |
| общее врем. процедур БД | 0,79 |

Если значение времени реакции системы находится выше пределов допустимой нормы:

1. Значение коэффициента корреляции со значениями параметров общего времени ожидания roll, GUI-времени высокое, т.е. время хранения данных пользовательского контекста в памяти системы высокое;
2. Корреляция с параметром времени обработки ниже 0,8, что означает невысокую взаимосвязь;
3. Значение коэффициента корреляции со значениями времени обработки, суммарного времени ЦП, обработки данных в БД больше 0,9, что означает задержки при обработке запросов пользователей.

Недостатком данного сравнения параметра времени реакции является большое число системных параметров и отсутствие возможности определения точной причины высокой загрузки. Альтернативным решением является мониторинг состояния нагрузки БД на уровне ОС при помощи утилиты top и анализа данных технических таблиц на уровне БД.

Базы данных инсталлированы на платформе SAP HANA. Главными компонентом БД SAP HANA является indexserver, функциями которого являются управление сеансами пользователей, авторизация, управление транзакциями и обработка команд. Техническими инструментом анализа производительности баз данных систем была выбрана программа мониторинга на уровне ОС top (в качестве параметров анализа выбраны MEM(%),CPU(%), USER, COMMAND).

Возможны ситуации, при которых не наблюдается корреляционная связь параметров времени реакции со значениями параметров потребления ЦПУ, памяти. Причиной узких мест производительности может являться высокое число блокировок на уровне БД. M\_TABLE\_LOCKS – ракурс базы данных SAP HANA технической схемы “SYS”, содержащий детальную информацию о блокировках данных [21].

## **4.2. Параметры анализа загрузки сервера приложений**

При анализе потребления ЦПУ сервером приложений среди ситуаций, встречающихся при исследовании значений параметров, возможны такие, что высокая загрузка ЦПУ не является критичной ввиду высокого значения числа активных процессов, т.к. высокие значения вызваны запусков большого количества отчетов и фоновых заданий. Противоположным является случай, когда число шагов не связано с числом вызываемых отчетов (рис. 13, 14):

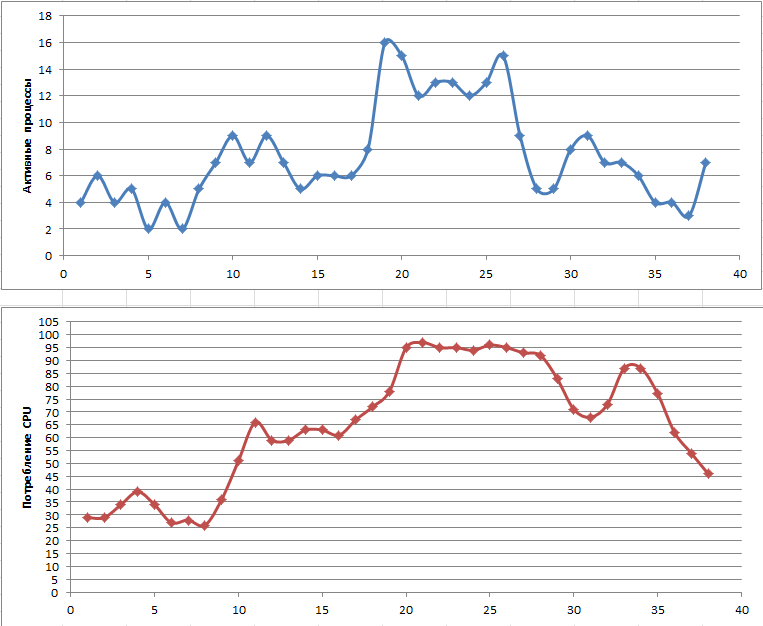


Рис. 13. Динамика изменения показателей числа системных активных процессов (синий) и потребления ЦПУ (красный)

Коэффициент корреляции между ЦПУ и активными процессам равен 0,67, что характеризует высокую связь между параметрами. Выводом является то, что высокое потребление ЦПУ связано с большим числом запущенных процессов в системе. Противоположная ситуация показана на рис. 14, где коэффициент корреляции равен 0,08:

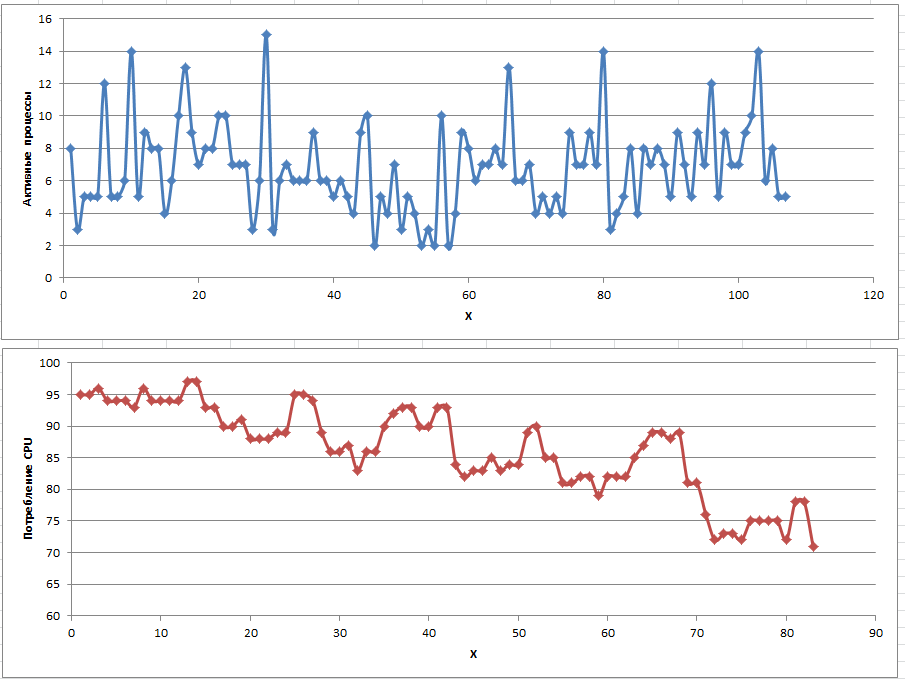


Рис. 14. Динамика изменения показателей системных активных процессов (синий) и потребления ЦПУ (красный)

Вывод – высокое потребление ЦПУ связано с недоработкой ПО, требуется оптимизация.

## **4.3. Итоговый выбор параметров анализа производительности систем**

1. SAP GUI:
2. Потребление ЦПУ(%) (SMON);
3. Число активных процессов (SMON);
4. Время отклика(мс).
5. Уровень ОС:
6. Потребление ЦПУ (%) (программа top);
7. Потребление памяти (%) (программа top);
8. Наименование процесса – indexserver (программа top);
9. Имя пользователя (top).
10. Уровень БД:
11. Число блокировок (M\_TABLE\_LOCKS).

Основным параметром для анализа выбрано время отклика. В ситуациях, когда время отклика находится не в пределах допустимой нормы, выполняется корреляционный анализ по всем параметрам, с целью поиска причины критичного состояния систем.

# **5. ОРГАНИЗАЦИОННО-ФУНКЦИОНАЛЬНАЯ СХЕМА СИСТЕМЫ**

Контекстная диаграмма процесса расчета сроков регламентного обслуживания представлена на рис. 15:

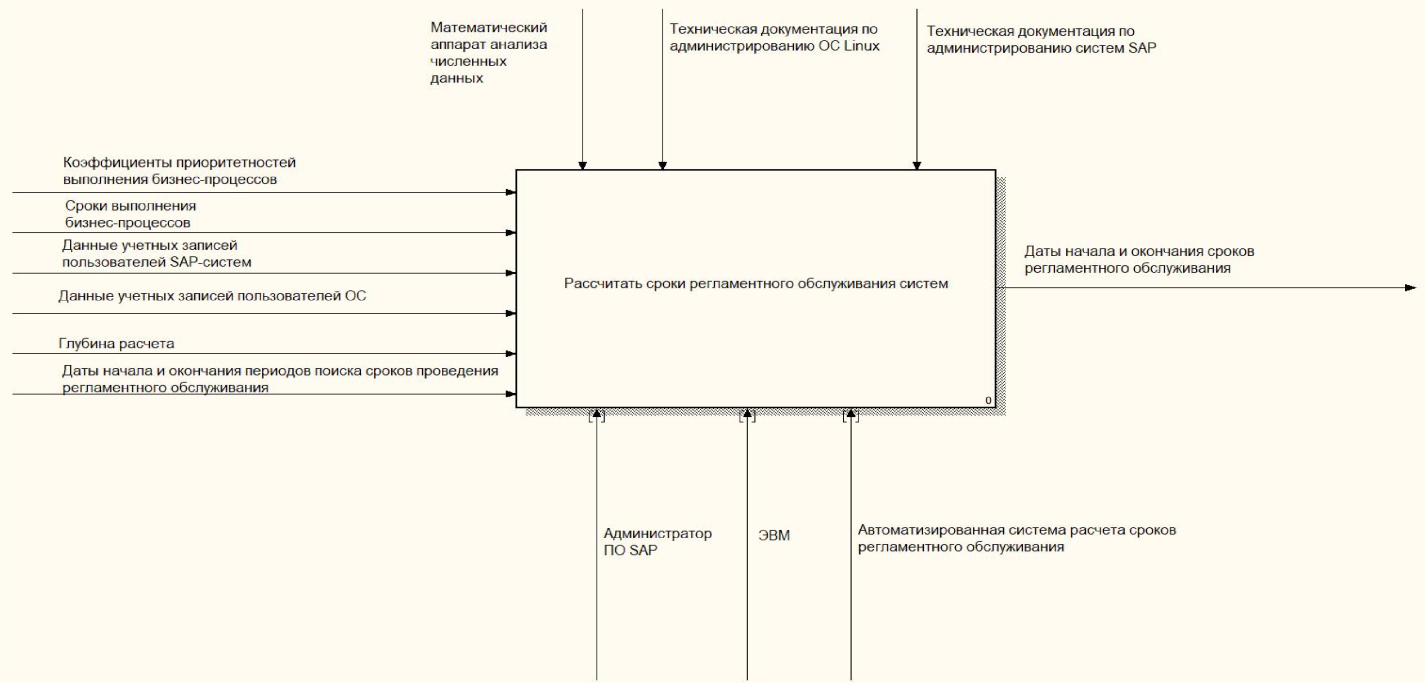


Рис. 15. Контекстная диаграмма

Входными данными являются время реакции, потребление ЦПУ, памяти, активных процессов серверов приложений и БД, число блокировок таблиц БД, данные учетных записей пользователей SAP систем, пользователей ОС.

Декомпозиционная диаграмма представлена в приложении 8.

Входными данными первого блока функциональной модели являются данные учетных записей пользователей систем SAP,ОС Linux, данные подключения к ИС, даты начала и окончания периодов поиска сроков регламентного обслуживания. Выходными данными являются множества значений параметров производительности, необходимых для расчета искомых значений. Второй блок функциональной модели описывает анализ причин отклонения значений времени реакции от допустимых значений (больше 1 секунды). Выходными данными блока являются текстовые результаты анализа высоких значений времени реакции. Третий блок диаграммы описывает действие преобразования значений времени реакции по результатам анализа. Выходными данными являются множества модифицированных значений времени реакции, изменение которых проводится вручную пользователем АС. Последний блок описывает расчет сроков регламентного обслуживания. Дополнительными входными данными блока являются коэффициенты приоритетностей бизнес-процессов, сроки выполнения бизнес-процессов, длительность регламентного обслуживания.

# **6. ПРОЕКТИРОВАНИЕ КОНЦЕПТУАЛЬНОЙ МОДЕЛИ БАЗЫ ДАННЫХ СИСТЕМЫ**

Для расчета сроков регламентного обслуживания необходимо хранить в БД АС информацию об анализируемых системах (статистические данные производительности), при этом окончательный расчет также выполняется на основании данных деталей выполнения бизнес-процессов в системах.

В результате анализа предметной области была спроектирована инфологическая модель на языке ER-диаграмм (Entity Relationship, модель «сущность-связь») в нотации Чена (рис. 16):

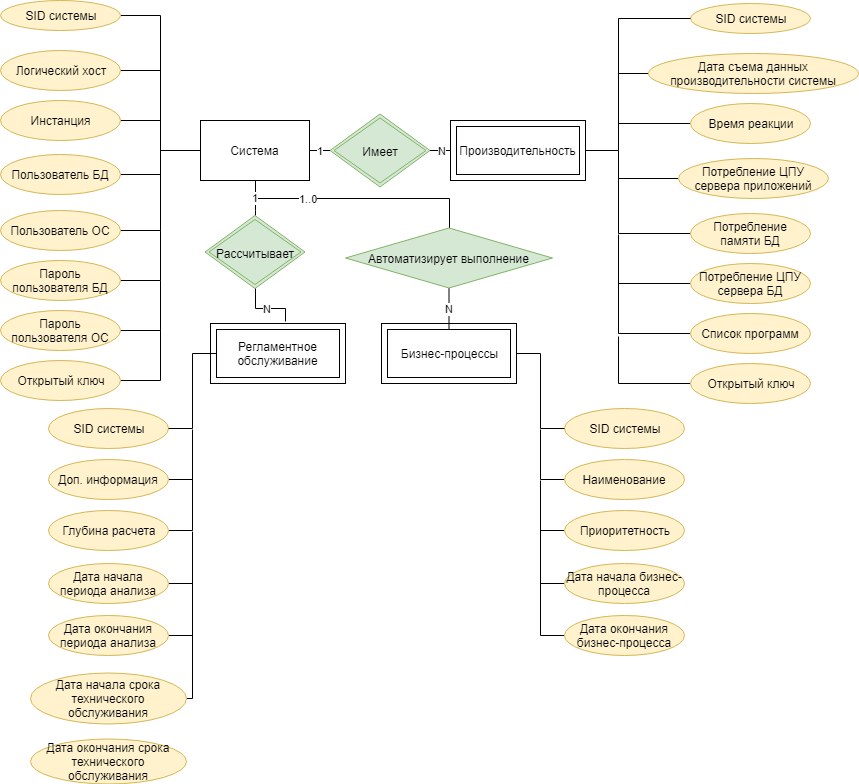


Рис. 16. Инфологическая модель предметной области

Модель содержит 2 идентифицирующие связи, одну неидентифицирующую связь (ведение бизнес-процессов не требует наличия данных системы), одну независимую сущность, три зависимые сущности. Стержневые сущности обозначены прямоугольниками, зависимые сущности- двойные прямоугольники, отношения – ромбами, атрибуты сущностей – овалы [22].

Модель состоит из 4 сущностей:

Система: сущность-родитель, описывает технические параметры подключения к серверу БД, приложений (SID системы, логины пользователей, пароль ключей, хост, инстанция);

Производительность: описывает значения технических параметров функциональных элементов рассматриваемых систем (загрузка ЦПУ, памяти, число активных системных процессов, дата и время снимков информации производительности);

Бизнес-процессы: описывает подробные детали выполнения (приоритетность, дата начала и окончания, наименование);

Регламентное обслуживание: описывает детали расчета сроков на основании определенной пользователем системы глубины расчета, дат начала и окончания периодов анализа.

# **7. ВИДЫ ОБЕСПЕЧЕНИЯ СИСТЕМЫ**

## **7.1. Информационное обеспечение**

Логическая инфологическая модель системы представлена на рис. 17:

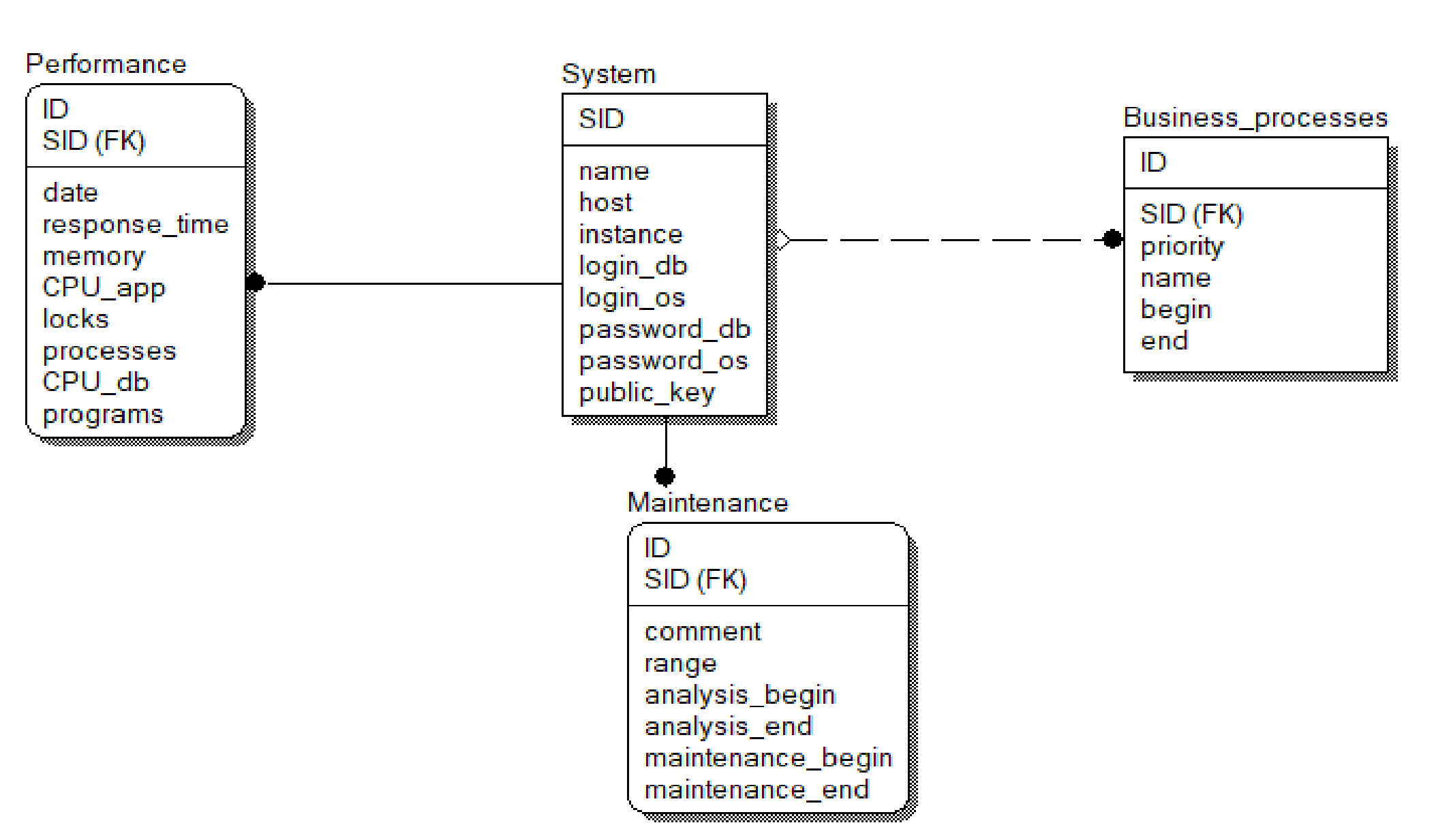


Рис. 17. Логическая модель системы

Логическая модель разработана с помощью CASE-средства AllFusion Erwin Data Modeler в нотации IDEF1X [23].

Физическая модель системы представлена на рис. 18:

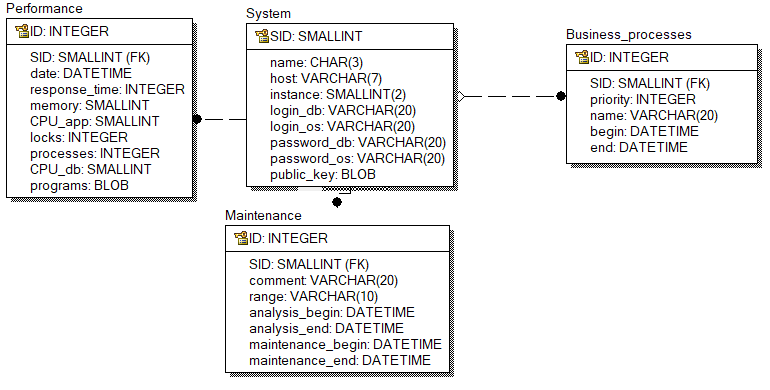


Рис. 18. Физическая модель системы

Описания структурных элементов физической модели системы представлены в таблицах 6, 7,8, 9:

**Таблица 6**

System (Сущность: «Система»)

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование поля** | **Тип поля** | **Описание** | **Пример значения** |
| SID (PK) | INTEGER | Порядковый номер системы | 1 |
| name | CHAR(3) | SID системы | R3R |
| host | VARCHAR(7) | Логический хост | di1R3R |
| instance | SMALLINT(2) | Номер инстанции | 00 |
| login\_db | VARCHAR(20) | Логин пользователя базы данных системы | Admin |
| login\_os | VARCHAR(20) | Логин пользователя ОС сервера приложений | Os\_admin |
| password\_db | VARCHAR(20) | Пароль пользователя БД | 1234567 |
| password\_os | VARCHAR(20) | Пароль пользователя на уровне операционной системы сервера приложений | 1234567 |
| public\_key | BLOB | Файл открытого ключа в двоичном формате | Последовательность байтов |

**Таблица 7**

Performance (Сущность: «Производительность»)

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование поля** | **Тип поля** | **Описание** | **Пример значения** |
| ID (PK) | INTEGER | Порядковый номер снимка информации производительности системы | 1 |
| SID (FK) | SMALLINT | Порядковый номер системы | R3R |
| Date | DATETIME | Дата снимка информации производительности системы | 01.01.2019 |
| Response\_time | INTEGER | Значение времени реакции системы, выраженное в миллисекундах | 431 |
| Memory | SMALLINT | Значение потребления памяти системы, выраженное в процентах | 20 |
| CPU\_app | SMALLINT | Значение потребления памяти сервера приложений в процентах | 20 |
| locks | INTEGER | Значение числа блокировок таблиц | 20 |
| Processes | INTEGER | Значение числа активных процессов системы | 20 |
| CPU\_db | SMALLINT | Значение потребления ЦПУ, выраженное в процентах | 20 |
| **Наименование поля** | **Тип поля** | **Описание** | **Пример значения** |
| programs | BLOB | Файл с листингом программ в двоичном формате | Последовательность байтов |

**Таблица 8**

Business\_processes (Сущность: «Бизнес-процессы»)

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование поля** | **Тип поля** | **Описание** | **Пример значения** |
| ID (PK) | INTEGER | Порядковый номер бизнес-процесса | 1 |
| SID (FK) | SMALLINT | Номер системы | 1 |
| priority | SMALLINT | Приоритетность бизнес-процесса | 1 |
| name | VARCHAR (50) | Наименование бизнес-процесса | Очистка логов |
| Begin | DATETIME | Дата начала бизнес-процесса | 01.01.2019 00:00 |
| end | DATETIME | Дата окончания бизнес-процесса | 01.01.2019 05:00 |

**Таблица 9**

Maintenance (Сущность: «Регламентное обслуживание»)

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование поля** | **Тип поля** | **Описание** | **Пример значения** |
| ID (PK) | INTEGER | Порядковый номер расчета | 1 |
| **Наименование поля** | **Тип поля** | **Описание** | **Пример значения** |
| SID (FK) | SMALLINT | Номер системы | R3R |
| comment | VARCHAR (20) | Информация по расчету с указанием количества дополнительно рассмотренных систем | R/3 ECC |
| range | VARCHAR(10) | Глубина расчета | Час |
| Analysis\_begin | SMALLINT | Дата начала анализа | 20 |
| Analysis\_end | SMALLINT | Дата окончания анализа | 20 |
| Maintenance\_begin | DATETIME | Дата начала регламентного обслуживания |  |
| Maintenance\_end | DATETIME | Дата окончания регламентного обслуживания |  |

В качестве СУБД была использована MySQL.

Так как SID систем SAP имеет 3-символьное строковое значение, то для хранения данных в БД для поля «SID» выбран тип «CHAR» с максимальным числом символов, равным трем.

Для вывода значений инстанций систем требуется 2-символьная строка, таким образом, при обращении к данным полей «instance» таблицы «System» выходная строка имеет 2 символа.

## **7.2. Математическое обеспечение**

Данная задача относится к типу многокритериальной оптимизации. Для оценки решений требовалось определить критерии оптимальности. В рамках данной задачи такими критериями оптимальности являются показатель системной нагрузки и приоритетность системных процессов. Текущая задача оптимизации сводится к минимизации функции F(t) (форм. 1), где t – параметр момента времени, x(t) – функция приоритетности момента времени t по системной нагрузке, y(t) – функция приоритетности момента времени t по выполняемым бизнес-процессам:

(1)

Мощность множеств значений F(t), x(t), y(t) определяется целой частью отношения времени, за который выполняется расчет (), и параметром количества времени () (форм. 2):

(2)

Областью значений функций x(t), y(t) являются диапазон [1; ].

Для выполнения расчетов необходимы статистические и численные методы, позволяющие выполнить анализ промежуточных расчетных значений и принимать дальнейшие решения относительно полученных результатов.

Для анализа статистических данных и исследования взаимосвязей числовых значений показателей производительности систем был выбран инструмент корреляционного анализа.

Корреляционная связь – связь, описывающая изменение признаков, которые отражают изменение одного признака в соответствии с изменчивостью другого.

Применение корреляционного анализа обосновано при наличии достаточного количества наблюдений более чем одной переменной.

Мерой корреляции случайных величин служит расчетный коэффициент r. Коэффициент корреляции принимает значения [-1,1]. Значение «-1» описывает отсутствие корреляции между величинами, «1» - полная корреляция величин [24].

Применение разного вида шкал представлено в таблице 10:

**Таблица 10**

Применение коэффициентов корреляции

|  |  |  |
| --- | --- | --- |
| **Типы шкал** | | **Мера связи** |
| **Переменная X** | **Переменная Y** |
| Интервальная | Интервальная | Коэффициент Пирсона |
| Ранговая, интервальная | Ранговая, интервальная | Коэффициент Спирмена |
| Ранговая | Ранговая | Коэффициент Кендалла |
| Дихотомическая | Дихотомическая | Четырехполевая корреляция |
| Дихотомическая | Ранговая | Рангово-бисериальный коэффициент |

Т.к. данные рассчитаны по интервальной шкале (потребление ЦПУ, памяти, время реакции, число блокировок, число активных процесов), то в качестве основного коэффициента корреляции выбран коэффициент Пирсона.

Коэффициент корреляции Пирсона «r» рассчитывается по формуле (3):

, (3)

где , – выборочные средние.

Входными данными для расчета сроков регламентного обслуживания являются следующие числовые массивы:

1. Массив числовых данных приоритетности бизнес-процессов;
2. Массивы времени реакции систем.

Среднее значение времени реакции за момент времени t рассчитывается по формуле 4:

, (4)

где ,– значения времени реакции i-ой системы в момент времени t, n – число систем.

Среднее значение приоритетности за момент времени t рассчитывается по формуле 5:

, (5)

где ,– значение приоритетности i-ого бизнес-процесса в момент времени t, n – число бизнес-процессов в момент времени t.

Отклонение значений времени реакции между элементами рассчитывается по формуле 6:

(6)

где значение времени реакции i-ой системы в момент времени t, значение времени реакции j-ой системы в момент времени t.

Отклонение значений приоритетности бизнес-процессов рассчитывается по формуле 7:

(7)

где значение приоритетности i-ого бизнес-процесса в момент времени t, значение приоритетности j-ого бизнес-процесса в момент времени t.

## **7.3. Алгоритмическое обеспечение**

### **7.3.1. Алгоритм анализа статистических данных**

Алгоритмы анализа статистических данных представлены на рис. 19, 20:



Рис. 19. Блок-схема алгоритма анализа статистических данных

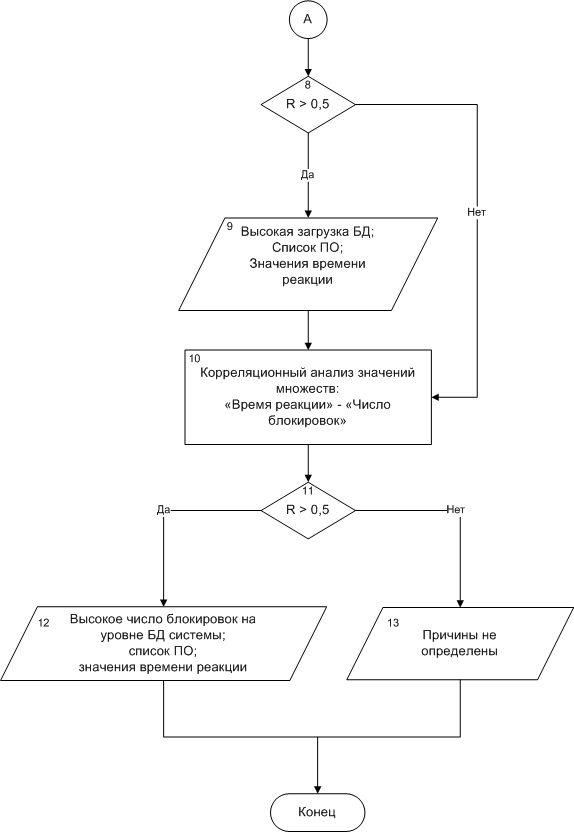


Рис. 20. Блок-схема алгоритма анализа статистических данных

Входные данные статистического анализа определяются на основании введенных дат начала и окончания анализа, а также систем, требуемых к рассмотрению (блок 1, 2).

Анализ данных делится на 2 типа: анализ высокой загрузки ЦПУ, анализ значений времени отклика больше 1 секунды.

Для первого типа анализа первым действием является поиск множеств высоких значений потребления ЦПУ (> 90%) (блок 3). Высокое значение потребления ЦПУ может возникать по двум причинам: увеличение числа процессов в системе, недоработка ПО. Для поиска причины выполняется корреляционный анализ множеств значений потребления ЦПУ и числа активных процессов (блок 4). В результате мониторинга данных ситуации были определены следующие коэффициенты корреляции: {0.7;0.62;0.53;0.7;0.71;0.63;0.61;0.7}. В качестве основного коэффициента корреляции, используемого при анализе статистических данных было выбрано значение 0,65.

Для поиска причины высоких значений времени реакции выполняется поиск промежутков значений времени реакции больше 1 секунды. Далее выполняется последовательность корреляционных анализов с целью поиска причины высокой нагрузки системы. В качестве значения коэффициента корреляции, относительно которого выполняется оценка связи выборок, выбрано 0,6. Данное значение определено в результате расчета среднего значения коэффициента за периоды критической работы систем (Значения коэффициентов корреляции в период критической нагрузки {0.7;0.62;0.53;0.7;0.71;0.63;0.61;0.7}).

Если по окончании выполнения серии корреляционных анализов не выявлено взаимосвязей времени реакции с параметрами производительности, то выводится информация об отсутствии решения поиска причины критической ситуации при эксплуатации системы.

### **7.3.2. Алгоритм расчета сроков регламентного обслуживания**

Алгоритм расчета сроков регламентного обслуживания представлен на рис. 21:

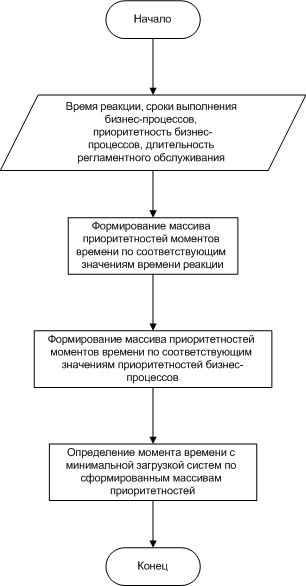


Рис. 21. Расчет сроков регламентного обслуживания

Входными данными расчета являются массив значений времени реакции, массив приоритетностей бизнес-процессов, временные даты выполнения бизнес-процессов, длительность регламентного обслуживания.

Массивы приоритетности формируются из двух параметров, рассчитанным по значениям времени реакции, приоритетностей бизнес-процессов:

1. Среднее значение параметров;
2. Среднее отклонение.

Если в выборке наблюдаются одинаковые значения среднего времени реакции, приоритетности бизнес-процессов, тогда выполняется сравнение средних отклонений параметров: выбирается минимальное.

При поиске конечного оптимального времени, если наблюдаются равные средние значения, выбирается момент времени, при котором среднее отклонение максимально (разность значения элемента из ранжированной выборки времени реакции и значения элемента из ранжированной выборки приоритета).

## **7.4. Программное обеспечение**

В данном подразделе описаны программные средства автоматизированного сбора данных производительности SAP-систем на прикладном уровне, уровне ОС, уровне БД, а также описаны детали разработки сервера автоматизированной системы.

### **7.4.1. SAP GUI Scripting**

Начиная с версии 6.20, ПО SAP поддерживает ActiveX.

ActiveX – технология реализации систем от Microsoft, основанная на стандарте COM. Каждый компонент имеет уникальный идентификатор GUID и взаимодействует с другими элементами через COM-интерфейсы (рис. 22).

SAP GUI Scripting API – набор классов, процедур, функций, обеспечивающих автоматизацию повторяющихся задач конечных пользователей путем создания макросов на языках программирования VBScript и Java [25].

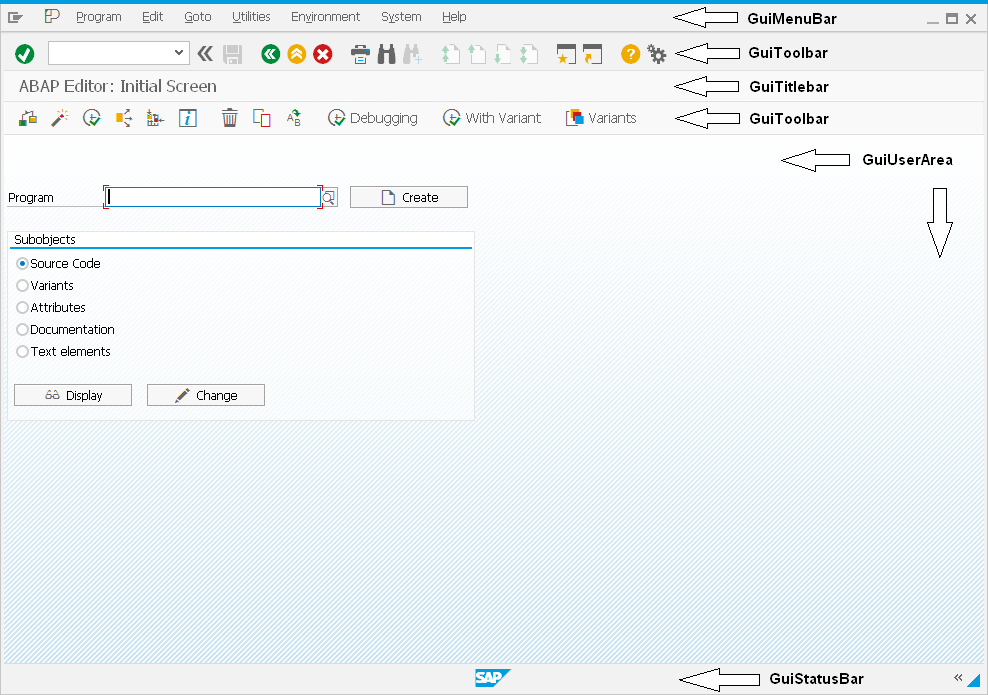


Рис. 22. Группы COM-объектов SAP GUI

Примерами задач являются:

1. Тестирование функциональности внедряемых программ;
2. Разработка индивидуальных интерфейсных приложений;
3. Разработка обучающих программ (демонстрации с ручными действиями пользователя).

Набор элементов интерфейса SAP GUI – это множество объектов, представляющих иерархическое дерево (рис. 23), корневым элементом которого является GUIApplication.

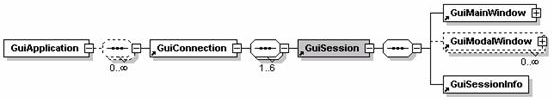


Рис. 23. Иерархическая структура объектов SAP GUI

GUIApplication – основной процесс SAP GUI, который генерируется при запуске сессий SAP GUI.

При создании соединения между SAP GUI и сервером приложений генерируется объект GUIConnection. Соединения создаются из программы SAPLogon, либо через программный вызов  openConnection и openConnectionByConnectionString GuiApplication. Число соединений является неограниченным. При создании подключения автоматически генерируется сессия пользователя по умолчанию.

GUI Session - контекст, в котором пользователь выполняет определенную задачу, например, работу с транзакцией. Максимальное число сессий для одного соединения ограниченно до шести.

Рассмотрим пример макроса (листинг 1):

**If** **Not** **IsObject**(application) **Then**

**Set** SapGuiAuto = **GetObject**("SAPGUI")

**Set** application = SapGuiAuto.GetScriptingEngine

**End** **If**

**If** **Not** **IsObject**(connection) **Then**

**Set** connection = application.Children(0)

**End** **If**

**If** **Not** **IsObject**(session) **Then**

**Set** session = connection.Children(0)

**End** **If**

**If** **IsObject**(WScript) **Then**

WScript.ConnectObject session, "on"

WScript.ConnectObject application, "on"

**End** **If**

Листинг 1. Макрос VBS

При определении текущей сессии пользователя выполняется получение множества соединений SAP GUI:

**Set** SapGuiAuto = **GetObject**("SAPGUI")

Далее выполняется идентификация платформы интерфейса разработки (Windows / Java) и получение объекта текущей сессии пользователя:

**Set** application = SapGuiAuto.GetScriptingEngine

**End** **If**

**If** **Not** **IsObject**(connection) **Then**

**Set** connection = application.Children(0)

**End** **If**

**If** **Not** **IsObject**(session) **Then**

**Set** session = connection.Children(0)

**End** **If**

**If** **IsObject**(WScript) **Then**

WScript.ConnectObject session, "on"

WScript.ConnectObject application, "on"

**End** **If**

Представленный скрипт, сгенерированный встроенным в SAP системе макрорекордером, является наиболее простым примером взаимодействия с процессами SAP GUI. Ограничением макрорекордера является отсутствие возможности автоматической генерации сессии пользователя. Для устранения проблемы требуется ручная корректировка сгенерированного кода программы.

### **7.4.2. SAP HANA Client**

SAP HANA Client – модуль NodeJS для взаимодействия с базой данных SAP HANA[26].

Пример взаимодействия с БД (листинг 2):

**const** hanaClient = require("@sap/hana-client");

**const** connection = hanaClient.createConnection();

**const** connectionParams = {

host : "hxehost",

port : 39013,

uid : "SYSTEM",

pwd : "\*\*\*\*\*\*\*\*",

databaseName : "HXE"

}

connection.connect(connectionParams, (err) => {

**if** (err) {

**return** console.error("Connection error", err);

}

**const** whereClause = process.argv[2] ? `WHERE "group" = '${process.argv[2]}'` : "";

**const** sql = `SELECT "name" FROM food\_collection ${whereClause}`;

connection.**exec**(sql, (err, rows) => {

connection.disconnect();

**if** (err) {

**return** console.error('SQL execute error:', err);

}

console.**log**("Results:", rows);

console.**log**(`Query '${sql}' returned ${rows.**length**} items`);

});

});

Листинг 2. Импорт данных БД

При подключении модуля требуется выполнить изменение настроек конфигурации проекта – установить пакет hana-client в проект, добавить дополнительные свойства в package.json:

npm config set @sap:registry https://npm.sap.com

npm install @sap/hana-client

Для подключения к БД требуются следующие параметры:

1. Хост системы;
2. Пользователь;
3. Пароль;
4. Порт;
5. Идентификатор базы данных.

### **7.4.3. Remote-exec**

Для разработки метода по автоматизированному сбору данных на уровне ОС был использован пакет remote-exec.

Remote-exec – пакет NodeJS для обеспечения взаимодействия с удаленными серверами на базе операционной системы Linux по протоколу SSH[27].

Программный интерфейс позволяет получить доступ и удаленно выполнять управление ресурсами сервера.

Листинг примера взаимодействия (листинг 3):

**var** rexec = require('remote-exec');

// see documentation for the ssh2 npm package for a list of all options

**var** connection\_options = {

port: 22,

username: 'myuser',

privateKey: require('fs').readFileSync('~/.ssh/rsa\_id'),

passphrase: 'mypassphrase'

};

**var** hosts = [

'host1.somewhere.com',

'host2.somewhere.else.com',

'250.110.0.13'

];

**var** cmds = [

'ls -l',

'cat /etc/hosts'

];

rexec(hosts, cmds, connection\_options, **function**(err){

**if** (err) {

console.**log**(err);

} **else** {

console.**log**('Great Success!!');

}

});

Листинг 3. Подключение удаленному Linux-серверу по протоколу SSH через программный интерфейс remote-exec

Для подключения к linux-серверу требуются следующие данные:

1. Имя хоста;

2. Имя пользователя уровня ОС;

3. Закрытый ключ;

4. Пароль закрытого ключа.

Генерация открытого и закрытого ключа выполняется при помощи команды:

localhost$ ssh-keygen -t rsa

Пользователю требуется выполнить добавление открытого ключа в файл authorized\_keys:

remote.server.host$ cat ~/id\_rsa.pub >> ~/.ssh/authorized\_keys

### **7.4.4. Инструменты разработки графического пользовательского интерфейса**

Для разработки пользовательского интерфейса системы были использованы Vue.js, bootstrap, JQueryUI.

Vue.js – JavaScript-фреймворк с открытым исходным кодом для создания пользовательских интерфейсов [29]. Листинг фрагмента программы реализации интерфейса представлен в приложении 1,4,5.

Bootstrap – набор HTML и CSS шаблонов для оформления типографики, веб-форм, кнопок, блоков и прочих элементов веб-интерфейса [30].

JQueryUI – JavaScript-библиотека на основе JQuery, реализующая набор взаимодействий, эффектов и графических инструментов пользовательского интерфейса [31].

Детали реализации интерфейса представлены в таблице 11:

**Таблица 11**

Детали применения

|  |  |  |
| --- | --- | --- |
| **Bootstrap** | **Vue.js** | **JQuery** |
| Навигационное меню, вертикальные вкладки | Выпадающие списки, данные таблиц | Инструменты ввода значений выборки импорта, анализа, расчета статистических данных |
| Таблицы с параметрами пользователей, анализа статистических данных | Создание, ведение, удаление записей таблиц | Календарь введения бизнес-процессов |
| Модальные окна создания, редактирования записей таблиц |  |  |

Так как в bootstrap 4 отсутствует элемент datetimepicker, был использован альтернативный GUI-компонент из JavaScript-библиотеки Gijgo, также поддерживающей стили bootstrap 4.

### **7.4.4. Cервер автоматизированной системы**

В качестве платформы для разработки сервера автоматизированной системы был выбрана Node.js.

Node.js – программная платформа, основанная на движке V8 и предназначенная для разработки клиент-серверных приложений; работает как однопоточное приложение на уровне JavaScript и многопоточное на уровне C++ [28]. Схема работы сервера представлена на рис. 24:

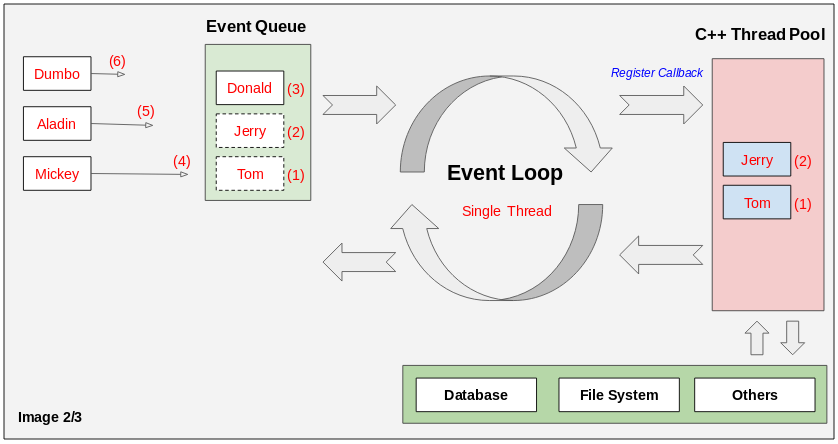


Рис. 24. Схема обработки запросов в Node JS.

Каждый пользовательский запрос добавляется в очередь Event Queue. Node.js использует алгоритм FIFO для обработки входящих запросов. Как бесконечный цикл, он передает запросы в  Thread Pool (Пул потоков), в то же время каждый запрос будет зарегистрирован функцией  Callback. При завершении обработки запроса, соотвествующая функция Callback будет вызвана для выполнения. Как программа написанная на языке C++, поддерживает многопоточность. Node.js также поддерживает многопоточность на уровне ядер.

Сервер системы содержит следующие методы (табл. 12):

**Таблица 12**

Серверные методы

|  |  |
| --- | --- |
| **Метод** | **Описание** |
| hana\_locks | сбор данных технической таблицы SAP HANA M\_TABLE\_LOCKS |
| cpu\_memory | сбор данных потребления ЦПУ, памяти на уровне ОС |

|  |  |
| --- | --- |
| **Метод** | **Описание** |
| import | сбор данных на прикладном уровне SAP-систем |
| analysis | анализ данных производительности систем по выбранному периоду |
| calculating | расчет сроков регламентного обслуживания |

Набор используемых модулей Node.js представлен в табл 13:

**Таблица 13**

Модули Node.js

|  |  |  |
| --- | --- | --- |
| **Модуль** | **Описание** | **Методы** |
| @sap/hana-client | Импорт данных | hana\_locks |
| mysql | Интеграция сервера с БД MySQL | hana\_locks, cpu\_mem, import, analysis, calculating |
| socket.io | Обработка клиент-серверных запросов | hana\_locks, cpu\_mem, import, analysis, calculating |
| body-parser | Синтаксический анализ текстовых запросов | hana\_locks, cpu\_mem, import, analysis, calculating |
| remote-exec | Удаленный доступ к ОС через протокол SSH | cpu\_memory |

Листинг программ импорта данных на прикладном уровне, уровне ОС, БД представлены в приложениях 2.3.

# **8. ОПИСАНИЯ МЕТОДОВ ОБРАБОТКИ ЗАПРОСОВ ИМПОРТА И АНАЛИЗА ДАННЫХ СИСТЕМЫ**

Декомпозиционная диаграмма процесса обработки запроса импорта статистических данных представлена на приложении 6.

Перед выполнением импорта данных выполняется проверка дат начала и окончания периода выборки пользователя. Если данные присутствуют в системе, выполняется поиск соответствующего пользователя: логин и пароль в таблице конфигурации системы. Далее выполняется изменение макроса: указывается SID системы, дата начала, дата окончания выборки, пользователь, пароль. После завершения выполнения работы макроса проводится обработка и добавление в БД автоматизированной системы необходимых данных: время реакции, потребление ЦПУ, число активных сессий, список выполненных отчетов, транзакций в разрезе времени.

Декомпозиционная диаграмма процесса обработки запроса анализа статистических данных представлен в приложении 7. Первым этапом выполнения анализа статистических данных является проверка наличия всех необходимых статистических данных в БД автоматизированной системы. Далее выполняется поиск высоких значений времени реакции (больше 1 секунды). Для определения причин высоких значений корреляции используется инструмент корреляционного анализа. После выполнения сервер отправляет выходные данные, содержащие список программ транзакций, в наибольшей мере повлиявших на значение времени реакции. Пользователь выполняет ручные действия, связанные с модификацией значений времени отклика за соответствующий период времени.

# **9. ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС СИСТЕМЫ**

Интерфейс администратора системы представлен на рис. 25:



Рис. 25. Импорт данных на прикладном уровне системы

Пользователь системы может выполнять следующие действия:

1. Импорт статистических данных;
2. Настройка параметров импорта;
3. Анализ данных на основании введенной выборки;
4. Расчет срока проведения регламентного обслуживания;
5. Создание удаление и редактирование бизнес-процессов совокупности систем.

Импорт данных из систем делится на два типа:

1. Импорт данных на прикладном уровне систем;
2. Импорт данных на уровне ОС;
3. Импорт данных на уровне БД.

Для выполнения импорта данных на прикладном уровне пользователю необходимо выполнить ввод даты начала и даты окончания периода сбора данных, а также директорию, в которую будет выполняться загрузка данных из рассматриваемых систем.

После выполнения импорта результаты будут отображены на таблице, приведенной справа от окна выбора периода импорта.

Для выполнения импорта статистических данных на уровне ОС и БД пользователь выполняет запуск с помощью кнопки «Активировать». Результаты импорта отображаются также в таблице (рис. 26).

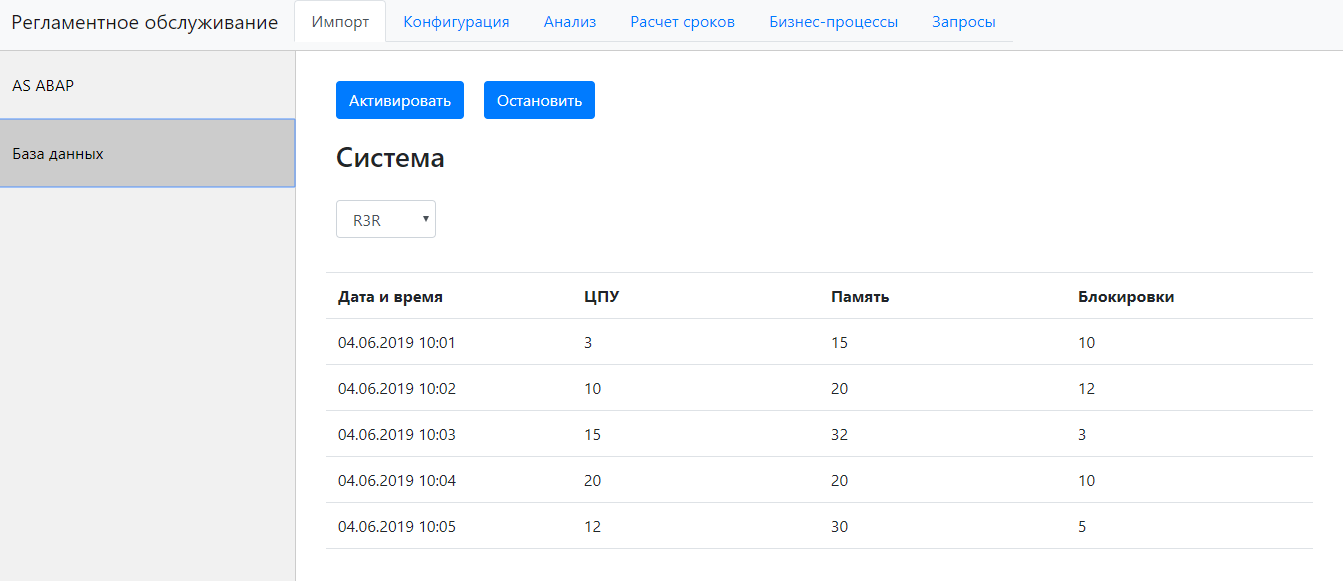


Рис. 26. Импорт данных на уровне ОС и БД

Предварительным шагом импорта является авторизация в системе и добавление vb-скриптов для эмуляций действий пользователя в системе (рис . 27)

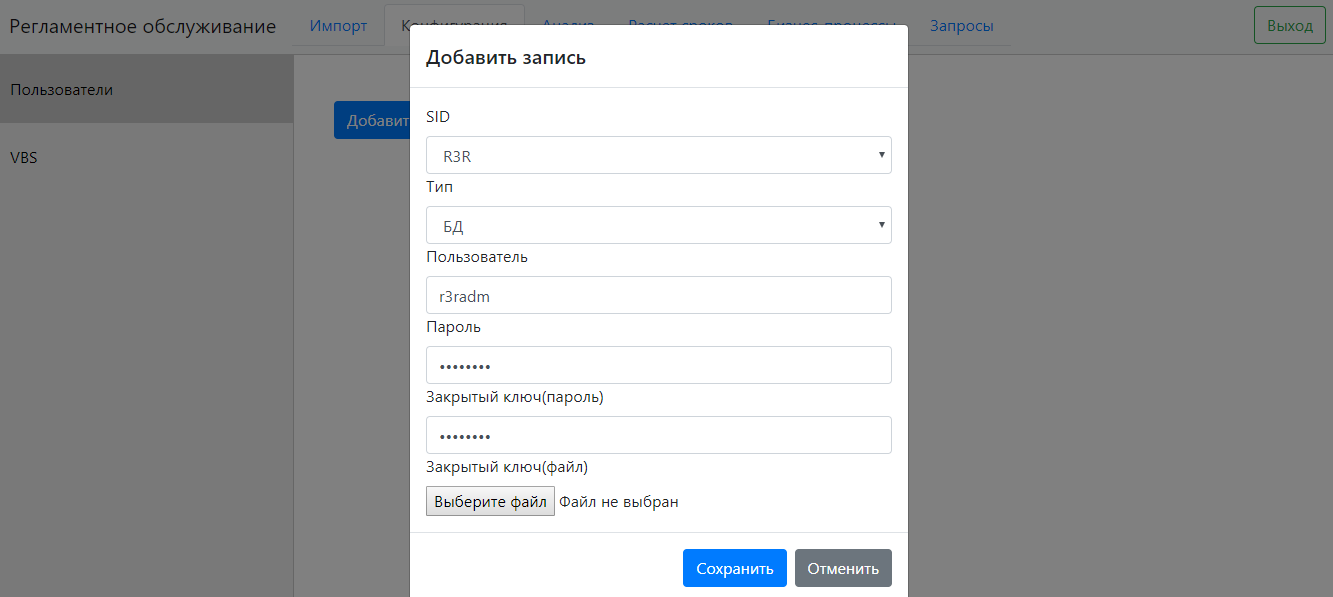


Рис. 27. Модальное окно добавления пользователя

Для реализации выполнения импорта данных требуются следующие параметры пользователей:

1. Прикладной уровень:

1.1. Логин;

1.2. Пароль;

1. Уровень ОС:

2.1. Логин;

2.2. Закрытый ключ (файл);

2.3. Закрытый ключ (пароль);

1. Уровень БД:

3.1. Логин;

3.2. Пароль.

В системе реализована беспарольная авторизация по SSH-протоколу.

Необходимо предварительно сгенерировать пару ключей (закрытый и открытый) на сервере БД анализируемой системы.

Информация о пользователях отображается в таблице (рис. 28):

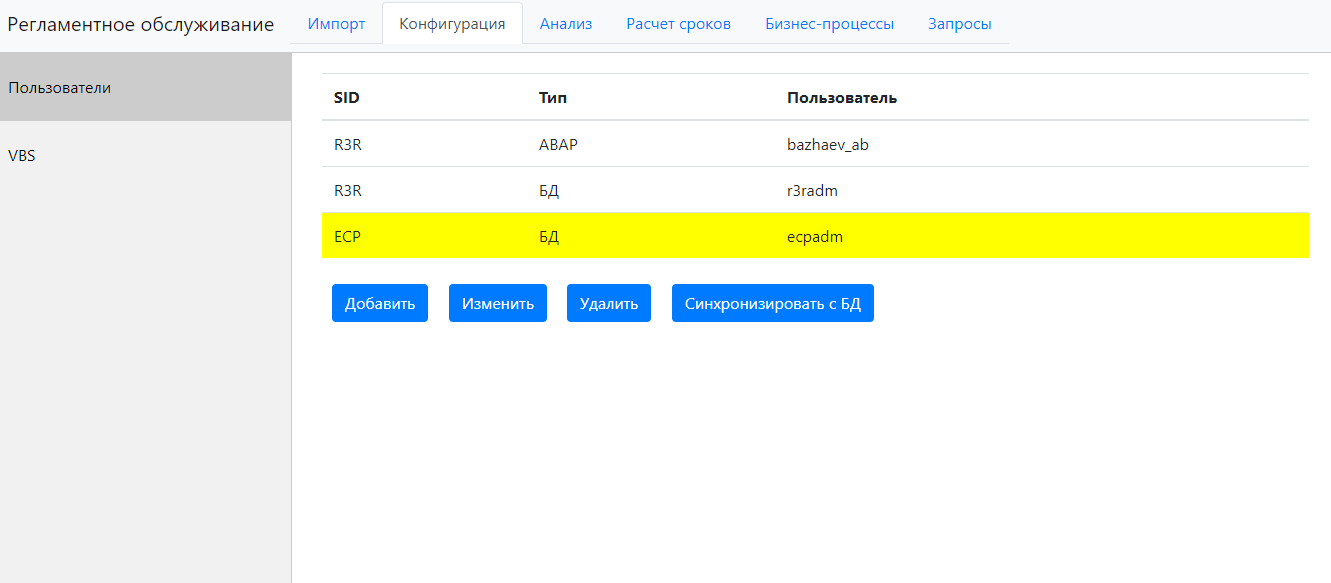


Рис. 28. Таблица с информацией о системных пользователях

Окно анализа статистических данных представлено на рис. 29.

Результатом выполнения является вывод информации о программах с высоким временем отклика, а также отчета, содержащего сведения о выполнении поиска корреляционных зависимостей за указанный период.

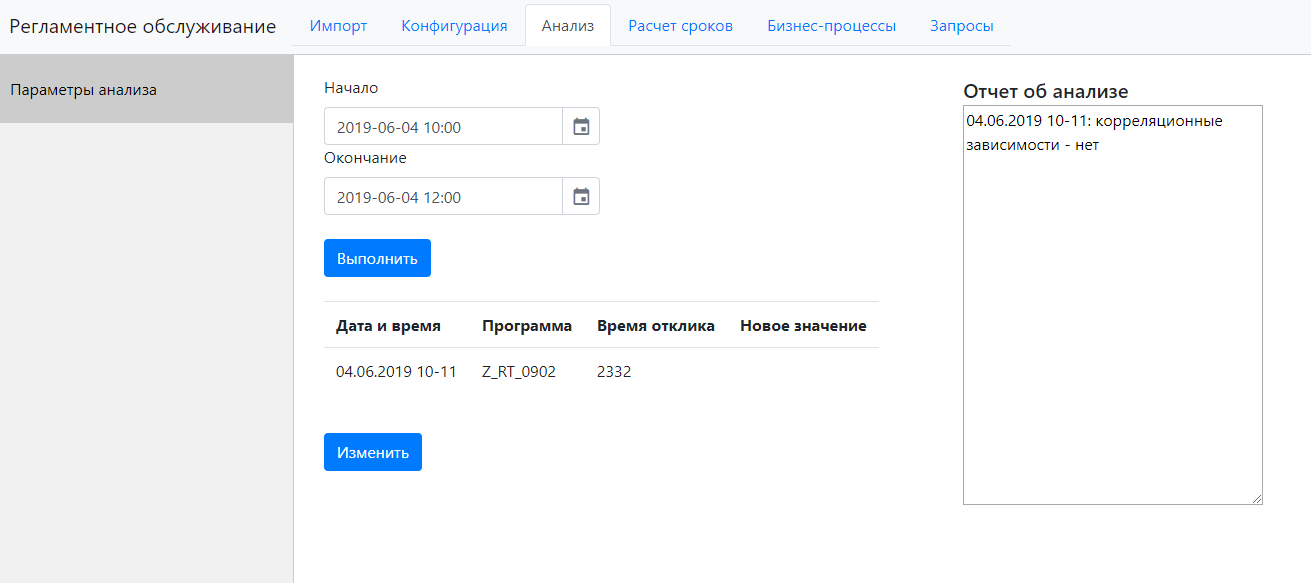


Рис. 29. Окно анализа статистических данных

Для ведения бизнес-процессов в пользовательский интерфейс добавлен JQueryUI-элемент для представления календарных событий FullCalendar [3]. Интерфейс FullCalendar имеет функции создания, ведения, удаления процессов, просмотра событий в разрезе дня, недели, месяца (рис. 30).

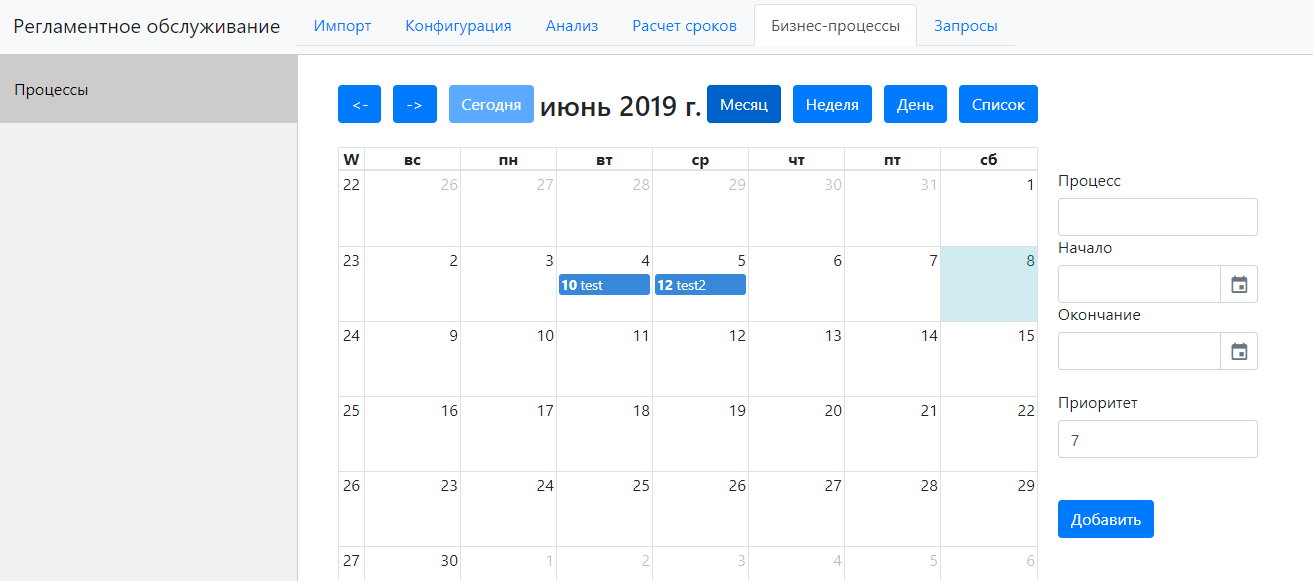


Рис. 30. Окно ведения бизнес-процессов

Вкладка «Расчет сроков» состоит из 2 пунктов (рис. 31). В пункте «Данные» содержится информация о предыдущих данных расчета, представленная в виде таблицы, содержащей следующие столбцы:

1. Тип;

2. Период расчета;

3. Результат выполнения.

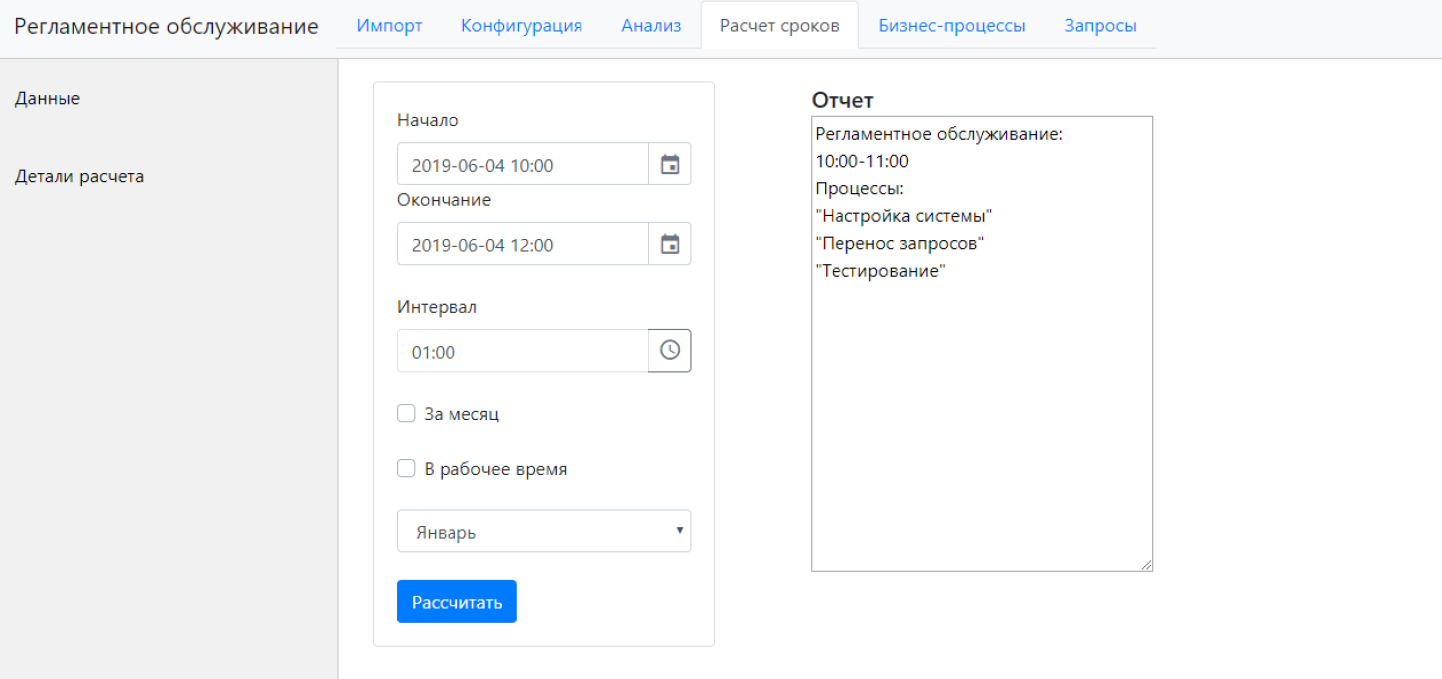


Рис. 31. Расчет сроков

Для выполнения расчета сроков проведения регламентного обслуживания требуется ввести период, за который необходимо выполнить анализ, продолжительность. Если требуется выполнить расчет за месяц, то выполняется активация одноименной опции и выбирается соответствующий месяц.

После выполнения расчета в область отчета будут выведены данные о процессах, имеющих пересечение с интервалом, определенным в результате вычислений и время регламентного обслуживания.

# **ЗАКЛЮЧЕНИЕ**

В ходе работы были разработаны алгоримты, методы и программные средства, позволяющие на основании данных производительности систем и информации о выполнении бизнес-процессов рассчитать сроки регламентного обслуживания информационных систем SAP.

В результате работы были выполнены следующие задачи:

1. Проанализированы средства мониторинга SAP и определены множества технических параметров производительности систем;
2. Выявлены, проанализированы критические состояния функционирования систем и определения множества целевых параметров производительности при расчете сроков регламентного обслуживания.
3. Разработан алгоритм анализа и корректировки статистических данных производительности систем.
4. Разработан алгоритм расчета сроков технического обслуживания.
5. Разработан метод получения и обработки значений параметров производительности систем SAP.
6. Спроектирована автоматизированная система расчета сроков регламентного обслуживания систем SAP.

За время работы были проведены многократные исследования разных ситуаций технического состояния ИС. В результате наблюдений были получены данные, на основании которых выполнялся синтез и корректировка алгоритмов расчета сроков регламентного обслуживания.

В ходе изучения предметной области были получены знания, которые в дальнейшем позволят решать прикладные задачи администрирования SAP систем, частью которых являются автоматизация сбора технических данных, автоматизация ручных действий пользователя.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. ГОСТ Р ИСО 9000-2015. Системы менеджмента качества. Основные положения и словарь. - М.:Росстандарт,2015.-54 с.
2. ГОСТ Р 52653-2006. Информационно-коммуникационные технологии в образовании. Термины и определения.-М.:Росстандарт,2006.- 6 с.
3. ГОСТ 18322-2016. Система технического обслуживания и ремонта техники. Термины и определения. - М.:Росстандарт,2017.- 14 с.
4. About SAP [Электронный ресурс] [2019] Режим доступа: https://www.sap.com/corporate/en/company.html
5. Системное администрирование SAP R/3. Лиане Вилл.-М.:Лори, 2010.- 361с.
6. Instance SAP [Электронный ресурс] [2019] Режим доступа: https://help.sap.com/saphelp\_nwpi71/helpdata/en/c4/3a5e90505211d189550000e829fbbd/content.htm?no\_cache=true
7. SAP NetWeaver [Электронный ресурс] [2019] Режим доступа: https://searchsap.techtarget.com/definition/NetWeaver
8. SAP ADM 100, Administration AS ABAP. – SAP AG, 2011. – 519 с.
9. SAP HANA [Электронный ресурс] [2019] Режим доступа: https://discover.sap.com/hana/ru-ru/what-is-hana.html
10. SAP HANA Administration Guide. - SAP AG, 2017. – 1362 с.
11. SAP ADM 315, AS ABAP - Performance Analysis. – SAP AG, 2019. – 174 с.
12. 8963 - Definition of SAP response time/processing time/CPU time [Электронный ресурс] [2019] Режим доступа: https://launchpad.support.sap.com/#/notes/8963
13. SAP ADM 315, AS ABAP – Performance Analysis. – SAP AG, 2019. – с. 8-9.
14. SAP ADM 315, AS ABAP – Performance Analysis. – SAP AG, 2019. – с. 10.
15. SAP ADM 315, AS ABAP – Performance Analysis. – SAP AG, 2019. – с. 11.
16. SMON [Электронный ресурс] [2019] Режим доступа: https://wiki.scn.sap.com/wiki/display/CPP/All+about+SMON
17. Команда top в Linux [Электронный ресурс] [2019] Режим доступа: http://linux-in.com/top-in-linux/
18. SAP PM Overview [Электронный ресурс] [2019] Режим доступа: https://www.tutorialspoint.com/sap\_pm/sap\_pm\_overview.htm
19. 1С: ТОИР Управление ремонтами и обслуживанием оборудования [Электронный ресурс] [2019] Режим доступа: https://www.gradum.ru/products/1s-toir/
20. IBM Maximo Asset Management [Электронный ресурс] [2019] Режим доступа: https://interprocom.ru/products/ibm/ibm-maximo-asset-management/
21. M\_TABLE\_LOCKS System View [Электронный ресурс] [2019] Режим доступа: https://help.sap.com/viewer/4fe29514fd584807ac9f2a04f6754767/2.0.03/en-US/8d163cd6735e4923ad8f967779606b64.html
22. Обзор нотаций, используемых при построении диаграмм «сущность-связь» [Электронный ресурс] [2019] Режим доступа: http://www.mstu.edu.ru/study/materials/zelenkov/ch\_2\_4.html
23. Основы методологии IDEF1X [Электронный ресурс] [2019] Режим доступа: https://www.cfin.ru/vernikov/idef/idef1x.shtml
24. Гмурман В.Е. Теория вероятностей и математическая статистика. Учебное пособие для вузов. — 10-е издание, стереотипное. — Москва: Высшая школа, 2004. — 479 с.
25. SAP GUI Scrtipting API [Электронный ресурс] [2019] Режим доступа: https://help.sap.com/viewer/b47d018c3b9b45e897faf66a6c0885a8/760.00/en-US
26. SAP HANA XS Advanced - Modules within Node.js SAP HANA applications [Электронный ресурс] [2019] Режим доступа: https://developers.sap.com/cis/tutorials/xsa-node-modules.html
27. Remote-exec [Электронный ресурс] [2019] Режим доступа: https://www.npmjs.com/package/remote-exec
28. About Node.js [Электронный ресурс] [2019] Режим доступа: https://nodejs.org/en/about/
29. Vue.js [Электронный ресурс] [2019] Режим доступа: https://vuejs.org/v2/guide/
30. Bootstrap [Электронный ресурс] [2019] Режим доступа: https://getbootstrap.com/docs/4.3/getting-started/introduction/
31. About JQuery UI [Электронный ресурс] [2019] Режим доступа: https://jqueryui.com/about/
32. Бажаев А.Б., Яценко Е.А. Факторы определения сроков регламентных работ. Вестник современных исследований. 2018. Вып. 11 -3(26). С. 229-231.

# **ПРИЛОЖЕНИЕ 1**

# Листинг программы обработки действий по редактированию таблицы системных пользователей

**var** socket;

**var** str = "";

**var** data = [];

window.onload = **function**() {

socket = io();

**var** table = **new** Vue ({

el: '#table1',

data: {

rows: [

]

},

methods: {

addRowHandlers: **function**() {

**var** t = document.getElementById("table1");

**var** r = t.getElementsByTagName("tr");

**for** (i = 1; i < r.**length**; i++) {

**var** currentRow = r[i];

**var** createClickHandler = **function**(row) {

**return** **function**() {

**var** t = document.getElementById("table1");

**var** rows = t.getElementsByTagName("tr");

**for** (i = 1; i < rows.**length**; i++) {

r[i].style.background = null;

}

**var** cell = row.getElementsByTagName("td")[0];

row.style.background = 'yellow';

**var** id = cell.innerHTML;

};

};

currentRow.onclick = createClickHandler(currentRow);

}

}

},

computed: {

"columns": **function** columns() {

**if** (**this**.rows.**length** == 0) {

**return** [];

}

**return** Object.**keys**(**this**.rows[0])

}

}

});

**var** adding = **new** Vue ({

el: '#add\_button',

methods: {

addItem: **function**() {

**var** t = document.getElementById("table1");

**var** rows = t.getElementsByTagName("tr");

**for** (i = 1; i < rows.**length**; i++) {

rows[i].style.background = null;

}

document.getElementById("sid\_conf").value = "";

document.getElementById("type\_conf").value = ""

document.getElementById("user\_conf").value = "";

$(document).ready(**function**() {

$('#conf\_modal').modal('show')

});

}

}

});

**var** save\_modal = **new** Vue ({

el: '#save\_modal',

methods: {

edit: **function**() {

**var** f = true;

**var** obj = {

"SID": document.getElementById("sid\_conf").value,

"Тип": document.getElementById("type\_conf").value,

"Пользователь": document.getElementById("user\_conf").value

};

**var** t = document.getElementById("table1");

**var** rows = t.getElementsByTagName("tr");

**for** (i = 1; i < rows.**length**; i++) {

**if** (rows[i].style.background == "yellow") {

rows[i].cells[0].innerText = document.getElementById("sid\_conf").value;

rows[i].cells[1].innerText = document.getElementById("type\_conf").value;

rows[i].cells[2].innerText = document.getElementById("user\_conf").value;

f = false;

}

}

**if** (f) {

table.rows.push(obj);

data.push(obj);

}

}

}

});

**var** editing = **new** Vue ({

el: '#edit\_button',

methods: {

edit: **function**() {

**var** t = document.getElementById("table1");

**var** rows = t.getElementsByTagName("tr");

**for** (i = 1; i < rows.**length**; i++) {

**if** (rows[i].style.background == "yellow") {

document.getElementById("sid\_conf").value = rows[i].cells[0].innerText;

document.getElementById("type\_conf").value = rows[i].cells[1].innerText;

document.getElementById("user\_conf").value = rows[i].cells[2].innerText;

$(document).ready(**function**()

$('#conf\_modal').modal('show')

});

}

}

}

}

});

**var** removing = **new** Vue ({

el: '#remove\_button',

methods: {

remove: **function**() {

**var** t = document.getElementById("table1");

**var** rows = t.getElementsByTagName("tr");

**for** (i = 1; i < rows.**length**; i++) {

**if** (rows[i].style.background == "yellow") {

rows[i].parentNode.removeChild(rows[i]);

}

}

}

}

});

**var** act= **new** Vue ({

el: '#process\_save',

methods: {

process\_save: **function**() {

$(document).ready(**function** () {

//alert(document.getElementById("datetimepicker\_begin").value);

calendar.addEvent( {id:1,

title: document.getElementById("process\_text").value,

start: document.getElementById("datetimepicker\_begin").value.**substr**(0, 10) + "T" + document.getElementById("datetimepicker\_begin").value.**substr**(11, 5),

end: document.getElementById("datetimepicker\_end").value.**substr**(0, 10) + "T" + document.getElementById("datetimepicker\_end").value.**substr**(11, 5)

});

document.getElementById("process\_text").value = "";

document.getElementById("datetimepicker\_begin").value = "";

document.getElementById("datetimepicker\_end").value = "";

});

}

}

});

}

# **ПРИЛОЖЕНИЕ 2**

# Листинг программы импорта данных на прикладном уровне через транзакцию st03

Option Explicit

Dim WSHShell, SAPGUIPath, SID, InstanceNo, WinTitle, SapGuiAuto, application, connection, session, BtnCode, count\_initial,data,i, oServ, cProc, oProc, inst, j

set WSHShell = WScript.CreateObject("WScript.Shell")

data = ""

i = 1

inst = 4

If IsObject(WSHShell) Then

SAPGUIPath = "C:\Program Files (x86)\SAP\FrontEnd\SAPgui\"

SID = "Имя\_инстанции"

InstanceNo = "Номер\_инстанции"

WSHShell.Exec SAPGUIPath & "SAPgui.exe " & SID & " " & \_

InstanceNo

WinTitle = "SAP"

While Not WSHShell.AppActivate(WinTitle)

WScript.Sleep 500

Wend

Set WSHShell = Nothing

End If

Wscript.Sleep 2000

If Not IsObject(application) Then

Set SapGuiAuto = GetObject("SAPGUI")

Set application = SapGuiAuto.GetScriptingEngine

End If

If Not IsObject(connection) Then

Set connection = application.Children(Application.Children.Count-1)

end if

If Not IsObject(session) Then

set session = connection.Children(0)

End If

If IsObject(WScript) Then

WScript.ConnectObject session, "on"

WScript.ConnectObject application, "on"

End If

session.findById("wnd[0]").sendVKey 0

session.findById("wnd[0]").Iconify

session.findById("wnd[0]/usr/txtRSYST-MANDT").Text = "300"

session.findById("wnd[0]/usr/txtRSYST-BNAME").text = "Пользователь"

session.findById("wnd[0]/usr/pwdRSYST-BCODE").text = "Пароль"

session.findById("wnd[0]/usr/txtRSYST-LANGU").Text = "RU"

session.findById("wnd[0]/usr/pwdRSYST-BCODE").setFocus

session.findById("wnd[0]/usr/pwdRSYST-BCODE").caretPosition = 10

session.findById("wnd[0]").sendVKey 0

on error resume next

session.findByid("wnd[1]").setFocus

If err.number = 0 Then

session.findById("wnd[1]/tbar[0]/btn[12]").press

End if

on error goto 0

For j = 1 to inst

session.findById("wnd[0]").maximize

session.findById("wnd[0]/tbar[0]/okcd").text = "st03"

session.findById("wnd[0]/tbar[0]/btn[0]").press

session.findById("wnd[0]/shellcont/shell/shellcont[1]/shell").expandNode "B.00" + CStr(j)

session.findById("wnd[0]/shellcont/shell/shellcont[1]/shell").topNode = "B"

session.findById("wnd[0]/shellcont/shell/shellcont[1]/shell").expandNode "B.00" + CStr(j) + ".1"

data = ""

i = 1

do while data <> "Дата"

if i\10 = 0 then

session.findById("wnd[0]/shellcont/shell/shellcont[1]/shell").selectedNode = "B.00" + CStr(j) + ".1.00" + CStr(i)

session.findById("wnd[0]/shellcont/shell/shellcont[1]/shell").doubleClickNode "B.00" + CStr(j) + ".1.00" + CStr(i)

else

session.findById("wnd[0]/shellcont/shell/shellcont[1]/shell").selectedNode = "B.00" + CStr(j) + ".1.0" + CStr(i)

session.findById("wnd[0]/shellcont/shell/shellcont[1]/shell").doubleClickNode "B.00" + CStr(j) + ".1.0" + CStr(i)

end if

session.findById("wnd[0]/usr/ssubSUBSCREEN\_0:SAPWL\_ST03N:1100/ssubWL\_SUBSCREEN\_1:SAPWL\_ST03N:1110/tabsG\_TABSTRIP/tabpTA00").select

data = session.findById("wnd[0]/usr/ssubSUBSCREEN\_0:SAPWL\_ST03N:1100/txtG\_WL\_DYNPRO-PERIOD\_TYPE").Text

session.findById("wnd[0]/shellcont/shell/shellcont[2]/shell").selectedNode = "E"

session.findById("wnd[0]/shellcont/shell/shellcont[2]/shell").doubleClickNode "E"

session.findById("wnd[0]/usr/ssubSUBSCREEN\_0:SAPWL\_ST03N:1100/ssubWL\_SUBSCREEN\_1:SAPWL\_ST03N:1110/tabsG\_TABSTRIP/tabpTA00/ssubWL\_SUBSCREEN\_2:SAPWL\_ST03N:1130/cntlALVCONTAINER/shellcont/shell").pressToolbarContextButton "&MB\_EXPORT"

session.findById("wnd[0]/usr/ssubSUBSCREEN\_0:SAPWL\_ST03N:1100/ssubWL\_SUBSCREEN\_1:SAPWL\_ST03N:1110/tabsG\_TABSTRIP/tabpTA00/ssubWL\_SUBSCREEN\_2:SAPWL\_ST03N:1130/cntlALVCONTAINER/shellcont/shell").selectContextMenuItem "&XXL"

session.findById("wnd[1]/tbar[0]/btn[0]").press

session.findById("wnd[1]/usr/ctxtDY\_PATH").text = "D:\Statictics\Система\di" + CStr(j) + "Система"

session.findById("wnd[1]/usr/ctxtDY\_FILENAME").text = data + ".XLSX"

session.findById("wnd[1]/usr/ctxtDY\_FILENAME").caretPosition = 7

session.findById("wnd[1]/tbar[0]/btn[11]").press

session.findById("wnd[0]/usr/ssubSUBSCREEN\_0:SAPWL\_ST03N:1100/ssubWL\_SUBSCREEN\_1:SAPWL\_ST03N:1110/tabsG\_TABSTRIP/tabpTA02").select

session.findById("wnd[0]/usr/ssubSUBSCREEN\_0:SAPWL\_ST03N:1100/ssubWL\_SUBSCREEN\_1:SAPWL\_ST03N:1110/tabsG\_TABSTRIP/tabpTA02/ssubWL\_SUBSCREEN\_2:SAPWL\_ST03N:1130/cntlALVCONTAINER/shellcont/shell").pressToolbarContextButton "&MB\_EXPORT"

session.findById("wnd[0]/usr/ssubSUBSCREEN\_0:SAPWL\_ST03N:1100/ssubWL\_SUBSCREEN\_1:SAPWL\_ST03N:1110/tabsG\_TABSTRIP/tabpTA02/ssubWL\_SUBSCREEN\_2:SAPWL\_ST03N:1130/cntlALVCONTAINER/shellcont/shell").selectContextMenuItem "&XXL"

session.findById("wnd[1]/tbar[0]/btn[0]").press

session.findById("wnd[1]/usr/ctxtDY\_PATH").text = "D:\Statictics\Система\di" + CStr(j) + "Система"

session.findById("wnd[1]/usr/ctxtDY\_FILENAME").text = data + "\_past" + ".XLSX"

session.findById("wnd[1]/usr/ctxtDY\_FILENAME").caretPosition = 4

session.findById("wnd[1]/tbar[0]/btn[11]").press

For k = 0 to 14

session.findById("wnd[0]/usr/ssubSUBSCREEN\_0:SAPWL\_ST03N:1100/ssubWL\_SUBSCREEN\_1:SAPWL\_ST03N:1110/tabsG\_TABSTRIP/tabpTA00/ssubWL\_SUBSCREEN\_2:SAPWL\_ST03N:1130/cntlALVCONTAINER/shellcont/shell").setCurrentCell k,"RESPMT"

session.findById("wnd[0]/usr/ssubSUBSCREEN\_0:SAPWL\_ST03N:1100/ssubWL\_SUBSCREEN\_1:SAPWL\_ST03N:1110/tabsG\_TABSTRIP/tabpTA00/ssubWL\_SUBSCREEN\_2:SAPWL\_ST03N:1130/cntlALVCONTAINER/shellcont/shell").doubleClickCurrentCell

session.findById("wnd[0]/shellcont[1]/shell").pressToolbarContextButton "&MB\_EXPORT"

session.findById("wnd[0]/shellcont[1]/shell").selectContextMenuItem "&XXL"

session.findById("wnd[1]/tbar[0]/btn[0]").press

session.findById("wnd[1]/usr/ctxtDY\_PATH").text = "D:\test"

session.findById("wnd[1]/usr/ctxtDY\_FILENAME").text = data + CStr(k) + ".XLSX"

session.findById("wnd[1]/usr/ctxtDY\_FILENAME").caretPosition = 8

session.findById("wnd[1]/tbar[0]/btn[11]").press

Set oServ = GetObject("winmgmts:")

Set cProc = oServ.ExecQuery("Select \* from Win32\_Process")

For Each oProc1 In cProc

If oProc1.Name = "EXCEL.EXE" Then

oProc1.Terminate()

End If

Next

session.findById("wnd[0]/shellcont[1]").close

Next

i = i+1

Set oServ = GetObject("winmgmts:")

Set cProc = oServ.ExecQuery("Select \* from Win32\_Process")

For Each oProc In cProc

If oProc.Name = "EXCEL.EXE" Then

oProc.Terminate()

End If

Next

loop

session.findById("wnd[0]/shellcont/shell/shellcont[1]/shell").collapseNode "B.00" + CStr(j)

Next

# **ПРИЛОЖЕНИЕ 3**

# Листинг программы импорта данных на уровне БД

**var** connection\_r3r = hanaClient.createConnection();

**var** hana\_option\_r3r = {

host: 'hanar3r',

port: 35015,

username: пользователь,

password: пароль,

databaseName: 'R30'

};

**var** connection\_option\_r3r = {

port: 22,

username: 'r30adm',

privateKey: require('fs').readFileSync('закрытый\_ключ'),

passphrase: 'закрытый\_ключ\_пароль'

};

**var** cmds = [

'uptime','free -m'

];

**function** cpu\_mem(host,connection) {

adding(“db\_cpu\_mem”,rexec(host, cmds, connection, **function**(err){

**if** (err) {

console.**log**(err);

} **else** {

console.**log**('Great Success!!');

}

}););

}

**function** hana\_locks() {

connection\_r3r.connect(hana\_option\_r3r, (err) => {

**if** (err) {

**return** console.error("Connection error", err);

}

**const** whereClause = process.argv[2] ? `WHERE "group" = '${process.argv[2]}'` : "";

**const** sql = 'select count(\*) from SYS.M\_TABLE\_LOCKS';

connection\_r3r.**exec**(sql, (err, rows) => {

connection1.disconnect();

**if** (err) {

**return** console.error('SQL execute error:', err);

}

console.**log**("R3R Results:", rows);

adding("db\_lock", console.rows);/\*число блокировок\*/

});

});

}

# **ПРИЛОЖЕНИЕ 4**

# Графический интерфейс системы

<!DOCTYPE>

<**html** xmlns="http://www.w3.org/1999/xhtml">

<**head**>

<**meta** charset="utf-8">

<**link** href="Stylesheet1.css" type="text/css" rel="stylesheet">

<**link** rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

<**link** rel="stylesheet" type="text/css" href="js/fullcalendar/packages/core/main.css"/>

<**link** rel="stylesheet" type="text/css" href="js/fullcalendar/packages/bootstrap/main.css"/>

<**link** rel="stylesheet" type="text/css" href="js/fullcalendar/packages/daygrid/main.css"/>

<**link** rel="stylesheet" type="text/css" href="js/fullcalendar/packages/timegrid/main.css"/>

<**link** rel="stylesheet" type="text/css" href="js/fullcalendar/packages/list/main.css"/>

<**link** href="https://unpkg.com/gijgo@1.9.13/css/gijgo.min.css" rel="stylesheet" type="text/css" />

<**title**></**title**>

</**head**>

<**body**>

<**script** src="import.js"></**script**>

<**script** src="selection.js"></**script**>

<**script** src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></**script**>

<**script** src="https://code.jquery.com/jquery-3.3.1.min.js"></**script**>

<**script** src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></**script**>

<**script** src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></**script**>

<**script** src='http://fullcalendar.io/js/fullcalendar-2.1.1/lib/moment.min.js'></**script**>

<**script** src='http://fullcalendar.io/js/fullcalendar-2.1.1/fullcalendar.min.js'></**script**>

<**script** src="js/fullcalendar/packages/core/main.js"></**script**>

<**script** src="js/fullcalendar/packages/bootstrap/main.js"></**script**>

<**script** src="js/fullcalendar/packages/daygrid/main.js"></**script**>

<**script** src="js/fullcalendar/packages/timegrid/main.js"></**script**>

<**script** src="js/fullcalendar/packages/list/main.js"></**script**>

<**script** src="js/fullcalendar/packages/interaction/main.js"></**script**>

<**script** type ="module" src="js/FullCalendar.js"></**script**>

<**script** type ="module" src="js/listener.js"></**script**>

<**script** src="https://unpkg.com/gijgo@1.9.13/js/gijgo.min.js" type="text/javascript"></**script**>

<**script** src="https://unpkg.com/gijgo@1.9.13/js/messages/messages.ru-ru.js" type="text/javascript"></**script**>

<**script** src="/socket.io/socket.io.js"></**script**>

<**nav** class="navbar navbar-expand-lg navbar-fixed-top navbar-light bg-light ">

<**a** class="navbar-brand" href="#">Регламентное обслуживание</**a**>

<**button** class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<**span** class="navbar-toggler-icon"></**span**>

</**button**>

<**div** class="collapse navbar-collapse" id="navbarSupportedContent">

<**ul** class="nav nav-tabs">

<**li** class="nav-item">

<**a** class="nav-link active" href="#import" role="tab" data-toggle="tab">Импорт</**a**>

</**li**>

<**li** class="nav-item">

<**a** class="nav-link" href="#conf" role="tab" data-toggle="tab">Конфигурация</**a**>

</**li**>

<**li** class="nav-item">

<**a** class="nav-link" href="#analysis" role="tab" data-toggle="tab">Анализ</**a**>

</**li**>

<**li** class="nav-item">

<**a** class="nav-link" href="#calculating" role="tab" data-toggle="tab">Расчет сроков</**a**>

</**li**>

<**li** class="nav-item">

<**a** class="nav-link" href="#processes" role="tab" data-toggle="tab">'Бизнес-процессы</**a**>

</**li**>

<**li** class="nav-item">

<**a** class="nav-link" href="#requests" role="tab" data-toggle="tab"Запросы</**a**>

</**li**>

</**ul**>

</**div**>

</**nav**>

<**div** class="tab-content">

<**div** role="tabpanel" class="tab-pane active" id="import">

<**div** class="tab">

<**button** class="tablinks" onclick="openCity(event, 'app')">AS ABAP</**button**>

<**button** class="tablinks" onclick="openCity(event, 'db')">'База данных</**button**>

</**div**>

<**div** id="app" class="tabcontent">

<**div** style="**display**:inline-block;**vertical-align**:top;">

<**div** class="card" style="**width**:320px">

<**div** class="card-body">

<**h3**>Система</**h3**>

<**select** class="form-control" id="systemOption" name="systemOption" style="**width**:276px">

<**option** value="R3R">R3R</**option**>

<**option** value="ECP">ECP</**option**>

<**option** value="PHR">PHR</**option**>

<**option** value="PBW">PBW</**option**>

</**select**>

<**br**>

<**h3**>Период</**h3**>

<**label** for="datetimepicker\_begin\_import">Начало</**label**>

<**input** id="datetimepicker\_begin\_import" width="276" />

<**br**>

<**script**>

**var** datepicker, config;

config = {

locale: 'ru-ru',

uiLibrary: 'bootstrap4',

format: 'yyyy-mm-dd HH:MM:SS'

};

$(document).ready(**function** () {

datepicker = $('#datetimepicker\_begin\_import').datetimepicker(config);

datepicker = $('#datetimepicker\_end\_import').datetimepicker(config);

});

</**script**>

<**label** for="datetimepicker\_end\_import">Окончание</**label**>

<**input** id="datetimepicker\_end\_import" width="276" />

<**br**>

<**div** class="input-group mb-3" style="**width**:276px">

<**div** class="custom-file">

<**input** class="custom-file-input" id="inputGroupFile02" />

<**label** class="custom-file-label" for="inputGroupFile02">'Выберите директорию...</**label**>

</**div**>

</**div**>

<**script**>

$('#inputGroupFile02').on('change', **function** () {

**var** fileName = $(**this**).val();

$(**this**).next('.custom-file-label').html(fileName);

})

</**script**>

<**button** id="import\_button" type="button" class="btn btn-primary" v-on:click="import\_ok()">'Выполнить</**button**>

</**div**>

</**div**>

</**div**>

<**div** style="**display**:inline-block; **padding-left**: 80px;">

<**label** for="type\_load">Тип нагрузки</**label**>

<**select** class="form-control" id="type\_load" style="**width**:276px">

<**option** value="�"иалог">�"иалог</option>

<**option** value="Фоновый">Фоновый</**option**>

<**option** value="RFC">RFC</**option**>

</**select**>

<**br**>

<**table** id="import\_vbs\_table" class="table">

<**tr**>

<**th**>"Дата и время</**th**>

<**th**>'Время отклика (мс)</**th**>

<**th**>ЦПУ (%)</**th**>

<**th**>Число сеансов</**th**>

</**tr**>

</**table**>

</**div**>

</**div**>

<**div** id="db" class="tabcontent">

<**table** cellpadding="10">

<**tr**>

<**td**>

<**button** id="db\_active" type="button" class="btn btn-primary" v-on:click="active()">Активировать</**button**>

</**td**>

<**td**>

<**button** id="db\_stop" type="button" class="btn btn-primary" v-on:click="stop()">Остановить</**button**>

</**td**>

</**tr**>

<**tr**>

<**td**><**h3**>Система</**h3**></**td**>

</**tr**>

<**tr**>

<**td**>

<**select** class="form-control" id="import\_system\_db" style="**width**:100px" onChange="selection()">

<**option** value="R3R">R3R</**option**>

<**option** value="ECP">ECP</**option**>

<**option** value="PHR">PHR</**option**>

<**option** value="HAP">HAP</**option**>

</**select**>

</**td**>

</**tr**>

</**table**>

<**br**>

<**table** class="table" width="100">

<**tr** style="**font-weight**:bold">

<**td**>"Дата и время</**td**>

<**td**>ЦПУ</**td**>

<**td**>Память</**td**>

<**td**>'Блокировки</**td**>

</**tr**>

<**tr**>

<**td**>

<**div** id="import\_db\_R3R">

<**textarea** id="text\_el" readonly="readonly" style="**height**:500px; **font-size**:12px;**border**:none;**text-align**:center"></**textarea**>

</**div**>

<**div** id="import\_db\_ECP" style="**display**:none">

<**textarea** id="text\_el2" readonly="readonly" style="**height**:500px; **font-size**:12px;**border**:none"></**textarea**>

</**div**>

</**td**>

<**td**>

<**textarea** id="text\_el1" readonly="readonly" style="**height**:500px;**font-size**:12px;**border**:none;**text-align**:center"></**textarea**>

</**td**>

<**td**>

<**textarea** id="log\_db" readonly="readonly" style="**height**:500px; **font-size**:12px;**border**:none;**color**:red"></**textarea**>

</**td**>

<**td**>

<**textarea** id="log\_db" readonly="readonly" style="**height**:500px; **font-size**:12px;**border**:none;**color**:red"></**textarea**>

</**td**>

</**tr**>

</**table**>

</**div**>

</**div**>

<**div** role="tabpanel" class="tab-pane fade" id="conf">

<**div** class="tab">

<**button** class="tablinks" onclick="openCity(event, 'users')">Пользователи</**button**>

<**button** class="tablinks" onclick="openCity(event, 'vbs')">VBS</**button**>

</**div**>

<**div** id="users" class="tabcontent">

<**table** id="table1" class="table">

<**thead**>

<**tr**>

<**th** v-for="col in columns">

{{col}}

</**th**>

</**tr**>

</**thead**>

<**tbody**>

<**tr** v-for="row in rows" class="clickable-row" v-on:click="addRowHandlers()">

<**td** v-for="col in columns">

{{row[col]}}

</**td**>

</**tr**>

</**tbody**>

</**table**>

<**table** cellpadding="10">

<**tr**>

<**td**><**button** id="add\_button" type="button" class="btn btn-primary" v-on:click="addItem()">�"обавить</**button**></**td**>

<**td**><**button** id="edit\_button" type="button" class="btn btn-primary" v-on:click="edit()">�зменить</**button**></**td**>

<**td**><**button** id="remove\_button" type="button" class="btn btn-primary" v-on:click="remove()">Удалить</**button**></**td**>

<**td**><**button** type="button" class="btn btn-primary" onclick="conf\_table('save')">Синхронизировать с �'�"</**button**></**td**>

</**tr**>

</**table**>

<**div** id="conf\_modal" class="modal" tabindex="-1" role="dialog">

<**div** class="modal-dialog" role="document">

<**div** class="modal-content">

<**div** class="modal-header">

<**h5** class="modal-title">�"обавить запись</**h5**>

</**div**>

<**div** class="modal-body">

<**label** for="sid\_conf">SID</**label**>

<**select** class="form-control" id="sid\_conf">

<**option** value="R3R">R3R</**option**>

<**option** value="ECP">ECP</**option**>

<**option** value="PHR">PHR</**option**>

<**option** value="PBW">PBW</**option**>

<**option** value="HAP">HAP</**option**>

</**select**>

<**label** for="type\_conf">Тип</**label**>

<**select** class="form-control" id="type\_conf">

<**option** value="ABAP">ABAP</**option**>

<**option** value="�'�"">�'�"</option>

</**select**>

<**label** for="user\_conf">Пользователь</**label**>

<**input** type="text" class="form-control" id="user\_conf">

<**label** for="password\_conf">Пароль</**label**>

<**input** type="password" class="form-control" id="password\_conf">

<**label** for="key\_conf">�-акрытый ключ(пароль)</**label**>

<**input** type="password" class="form-control" id="key\_conf">

<**label** for="key\_path\_conf">�-акрытый ключ(файл)</**label**>

<**br**>

<**input** type="file" id="key\_path\_conf">

</**div**>

<**div** class="modal-footer">

<**button** id="save\_modal" type="button" class="btn btn-primary" data-dismiss="modal" v-on:click="edit()">Сохранить</**button**>

<**button** type="button" class="btn btn-secondary" data-dismiss="modal">Отменить</**button**>

</**div**>

</**div**>

</**div**>

</**div**>

</**div**>

<**div** id="vbs" class="tabcontent">

<**listing** id="listing1" style="**height**:90%;**width**:100%; **overflow-y**:scroll; **overflow-x**:scroll;">

Option Explicit

Dim WSHShell, SAPGUIPath, SID, InstanceNo, WinTitle, SapGuiAuto, application, connection, session, BtnCode, count\_initial,data,i, oServ, cProc, oProc, inst, j

set WSHShell = WScript.CreateObject("WScript.Shell")

data = ""

i = 1

inst = 4

If IsObject(WSHShell) Then

SAPGUIPath = "C:\Program Files (x86)\SAP\FrontEnd\SAPgui\"

SID = "di1pbw"

InstanceNo = "00"

WSHShell.Exec SAPGUIPath & "SAPgui.exe " & SID & " " & \_

InstanceNo

WinTitle = "SAP"

While Not WSHShell.AppActivate(WinTitle)

WScript.Sleep 500

Wend

Set WSHShell = Nothing

End If

Wscript.Sleep 2000

If Not IsObject(application) Then

Set SapGuiAuto = GetObject("SAPGUI")

Set application = SapGuiAuto.GetScriptingEngine

End If

If Not IsObject(connection) Then

Set connection = application.Children(Application.Children.Count-1)

end if

If Not IsObject(session) Then

set session = connection.Children(0)

End If

If IsObject(WScript) Then

WScript.ConnectObject session, "on"

WScript.ConnectObject application, "on"

End If

session.findById("wnd[0]").sendVKey 0

session.findById("wnd[0]").Iconify

session.findById("wnd[0]/usr/txtRSYST-MANDT").Text = "300"

session.findById("wnd[0]/usr/txtRSYST-BNAME").text = "bazhaev-ab"

session.findById("wnd[0]/usr/pwdRSYST-BCODE").text = "Lo4815162342!"

session.findById("wnd[0]/usr/txtRSYST-LANGU").Text = "RU"

session.findById("wnd[0]/usr/pwdRSYST-BCODE").setFocus

session.findById("wnd[0]/usr/pwdRSYST-BCODE").caretPosition = 10

session.findById("wnd[0]").sendVKey 0

on error resume next

session.findByid("wnd[1]").setFocus

If err.number = 0 Then

session.findById("wnd[1]/tbar[0]/btn[12]").press

End if

on error goto 0

For j = 1 to inst

session.findById("wnd[0]").maximize

session.findById("wnd[0]/tbar[0]/okcd").text = "st03"

session.findById("wnd[0]/tbar[0]/btn[0]").press

session.findById("wnd[0]/shellcont/shell/shellcont[1]/shell").expandNode "B.00" + CStr(j)

session.findById("wnd[0]/shellcont/shell/shellcont[1]/shell").topNode = "B"

session.findById("wnd[0]/shellcont/shell/shellcont[1]/shell").expandNode "B.00" + CStr(j) + ".1"

data = ""

i = 1

do while data <> "24.03.2019"

if i\10 = 0 then

session.findById("wnd[0]/shellcont/shell/shellcont[1]/shell").selectedNode = "B.00" + CStr(j) + ".1.00" + CStr(i)

session.findById("wnd[0]/shellcont/shell/shellcont[1]/shell").doubleClickNode "B.00" + CStr(j) + ".1.00" + CStr(i)

else

session.findById("wnd[0]/shellcont/shell/shellcont[1]/shell").selectedNode = "B.00" + CStr(j) + ".1.0" + CStr(i)

session.findById("wnd[0]/shellcont/shell/shellcont[1]/shell").doubleClickNode "B.00" + CStr(j) + ".1.0" + CStr(i)

end if

session.findById("wnd[0]/usr/ssubSUBSCREEN\_0:SAPWL\_ST03N:1100/ssubWL\_SUBSCREEN\_1:SAPWL\_ST03N:1110/tabsG\_TABSTRIP/tabpTA00").select

data = session.findById("wnd[0]/usr/ssubSUBSCREEN\_0:SAPWL\_ST03N:1100/txtG\_WL\_DYNPRO-PERIOD\_TYPE").Text

session.findById("wnd[0]/shellcont/shell/shellcont[2]/shell").selectedNode = "E"

session.findById("wnd[0]/shellcont/shell/shellcont[2]/shell").doubleClickNode "E"

session.findById("wnd[0]/usr/ssubSUBSCREEN\_0:SAPWL\_ST03N:1100/ssubWL\_SUBSCREEN\_1:SAPWL\_ST03N:1110/tabsG\_TABSTRIP/tabpTA00/ssubWL\_SUBSCREEN\_2:SAPWL\_ST03N:1130/cntlALVCONTAINER/shellcont/shell").pressToolbarContextButton "&MB\_EXPORT"

session.findById("wnd[0]/usr/ssubSUBSCREEN\_0:SAPWL\_ST03N:1100/ssubWL\_SUBSCREEN\_1:SAPWL\_ST03N:1110/tabsG\_TABSTRIP/tabpTA00/ssubWL\_SUBSCREEN\_2:SAPWL\_ST03N:1130/cntlALVCONTAINER/shellcont/shell").selectContextMenuItem "&XXL"

session.findById("wnd[1]/tbar[0]/btn[0]").press

session.findById("wnd[1]/usr/ctxtDY\_PATH").text = "D:\Statictics\PHR\di" + CStr(j) + "phr"

session.findById("wnd[1]/usr/ctxtDY\_FILENAME").text = data + ".XLSX"

session.findById("wnd[1]/usr/ctxtDY\_FILENAME").caretPosition = 7

session.findById("wnd[1]/tbar[0]/btn[11]").press

session.findById("wnd[0]/usr/ssubSUBSCREEN\_0:SAPWL\_ST03N:1100/ssubWL\_SUBSCREEN\_1:SAPWL\_ST03N:1110/tabsG\_TABSTRIP/tabpTA02").select

session.findById("wnd[0]/usr/ssubSUBSCREEN\_0:SAPWL\_ST03N:1100/ssubWL\_SUBSCREEN\_1:SAPWL\_ST03N:1110/tabsG\_TABSTRIP/tabpTA02/ssubWL\_SUBSCREEN\_2:SAPWL\_ST03N:1130/cntlALVCONTAINER/shellcont/shell").pressToolbarContextButton "&MB\_EXPORT"

session.findById("wnd[0]/usr/ssubSUBSCREEN\_0:SAPWL\_ST03N:1100/ssubWL\_SUBSCREEN\_1:SAPWL\_ST03N:1110/tabsG\_TABSTRIP/tabpTA02/ssubWL\_SUBSCREEN\_2:SAPWL\_ST03N:1130/cntlALVCONTAINER/shellcont/shell").selectContextMenuItem "&XXL"

session.findById("wnd[1]/tbar[0]/btn[0]").press

session.findById("wnd[1]/usr/ctxtDY\_PATH").text = "D:\Statictics\PHR\di" + CStr(j) + "phr"

session.findById("wnd[1]/usr/ctxtDY\_FILENAME").text = data + "\_past" + ".XLSX"

session.findById("wnd[1]/usr/ctxtDY\_FILENAME").caretPosition = 4

session.findById("wnd[1]/tbar[0]/btn[11]").press

i = i+1

Set oServ = GetObject("winmgmts:")

Set cProc = oServ.ExecQuery("Select \* from Win32\_Process")

For Each oProc In cProc

If oProc.Name = "EXCEL.EXE" Then

oProc.Terminate()

End If

Next

loop

session.findById("wnd[0]/shellcont/shell/shellcont[1]/shell").collapseNode "B.00" + CStr(j)

Next

</**listing**>

</**div**>

</**div**>

<**div** role="tabpanel" class="tab-pane fade" id="analysis">

<**div** class="tab">

<**button** class="tablinks" onclick="openCity(event, 'data\_analysis')">Параметры анализа</**button**>

</**div**>

<**div** id="data\_analysis" class="tabcontent">

<**div** style="**display**:inline-block;**vertical-align**:top;">

<**label** for="begin\_process">Начало</**label**>

<**input** id="datetimepicker\_analysis\_begin" width="276" />

<**script**>

**var** datepicker, config;

config = {

locale: 'ru-ru',

uiLibrary: 'bootstrap4',

format: 'yyyy-mm-dd HH:MM:SS'

};

$(document).ready(**function** () {

datepicker = $('#datetimepicker\_analysis\_begin').datetimepicker(config);

datepicker = $('#datetimepicker\_analysis\_end').datetimepicker(config);

});

</**script**>

<**label** for="end\_process">Окончание</**label**>

<**input** id="datetimepicker\_analysis\_end" width="276" />

<**br**>

<**button** id="analysis\_start" type="button" class="btn btn-primary" v-on:click="analysis\_start">'Выполнить</**button**>

<**br**>

<**br**>

<**table** id="analysis\_table" class="table">

<**tr**>

<**th**>"Дата и время</**th**>

<**th**>Программа</**th**>

<**th**>'Время отклика</**th**>

<**th**>Новое значение</**th**>

</**tr**>

</**table**>

<**br**>

<**button** id="analysis\_change" type="button" class="btn btn-primary" v-on:click="analysis\_change()">Изменить</**button**>

</**div**>

<**div** style="**display**:inline-block;**vertical-align**:top;**padding-left**: 80px;">

<**h5** class="modal-title">Отчет об анализе</**h5**>

<**textarea** id =”table\_analysis” style="**height**:400px;**width**:300px"><**textarea**>

</**div**>

</**div**>

</**div**>

<**div** role="tabpanel" class="tab-pane fade" id="processes">

<**div** class="tab">

<**button** class="tablinks" onclick="openCity(event, 'proc\_tab')">Процессы</**button**>

</**div**>

<**div** id="proc\_tab" class="tabcontent">

<**table** cellpadding="10">

<**tr**>

<**td**>

<**div** id="calendar" style="**width**:700px">

</**div**>

</**td**>

<**td**>

<**label** for="process\_text">Процесс</**label**>

<**input** type="text" class="form-control" id="process\_text">

<**label** for="begin\_process">Начало</**label**>

<**input** id="datetimepicker\_begin" width="200" />

<**script**>

**var** datepicker, config;

config = {

locale: 'ru-ru',

uiLibrary: 'bootstrap4',

format: 'yyyy-mm-dd HH:MM:SS'

};

$(document).ready(**function** () {

datepicker = $('#datetimepicker\_begin').datetimepicker(config);

datepicker = $('#datetimepicker\_end').datetimepicker(config);

});

</**script**>

<**label** for="end\_process">Окончание</**label**>

<**input** id="datetimepicker\_end" width="200" />

<**br**>

<**label** for="priority">Приоритет</**label**>

<**br**>

<**input** id="priority" type="number" class="form-control" width="200" />

<**br**>

<**br**>

<**button** id="process\_save" type="button" class="btn btn-primary" data-dismiss="modal" v-on:click="process\_save()">"Добавить</**button**>

</**td**>

</**tr**>

</**table**>

</**div**>

</**div**>

<**div** role="tabpanel" class="tab-pane fade" id="calculating">

<**div** class="tab">

<**button** class="tablinks" onclick="openCity(event, 'calculating\_data')">"Данные</**button**>

<**button** class="tablinks" onclick="openCity(event, 'calculating\_details')">"Детали расчета</**button**>

</**div**>

<**div** id="calculating\_details" class="tabcontent">

<**script**>

**var** datepicker, config;

config = {

locale: 'ru-ru',

uiLibrary: 'bootstrap4',

format: 'yyyy-mm-dd HH:MM:SS'

};

$(document).ready(**function** () {

datepicker = $('#datetimepicker\_calculating\_begin').datetimepicker(config);

datepicker = $('#datetimepicker\_calculating\_end').datetimepicker(config);

datepicker = $('#interval').timepicker(config);

});

</**script**>

<**div** style="**display**:inline-block;**vertical-align**:top;">

<**div** class="card" style="**width**:300px">

<**div** class="card-body">

<**label** for="begin\_process">Начало</**label**>

<**input** id="datetimepicker\_calculating\_begin" />

<**label** for="end\_process">Окончание</**label**>

<**input** id="datetimepicker\_calculating\_end" />

<**br**>

<**label** for="interval">Интервал</**label**>

<**input** id="interval" />

<**br**>

<**div** class="custom-control custom-checkbox">

<**input** type="checkbox" class="custom-control-input" id="calculating\_month\_check">

<**label** class="custom-control-label" for="defaultUnchecked">За месяц</**label**>

</**div**>

<**br**>

<**div** class="custom-control custom-checkbox">

<**input** type="checkbox" class="custom-control-input" id="calculating\_\_check">

<**label** class="custom-control-label" for="defaultUnchecked1">'В рабочее время</**label**>

</**div**>

<**br**>

<**select** v-model="selected" class="form-control" id="calculating\_month">

<**option** v-for="option in options" v-bind:value="option.value">

{{ option.text }}

</**option**>

</**select**>

<**br**>

<**button** id="calculating\_button" type="button" class="btn btn-primary" v-on:click="caclulate()">Рассчитать</**button**>

</**div**>

</**div**>

</**div**>

<**div** style="**display**:inline-block;**vertical-align**:top; **padding-left**:80px">

<**h5** class="modal-title">Отчет</**h5**>

<**textarea** style="**height**:400px;**width**:300px" id="result"></**textarea**>

</**div**>

</**div**>

<**div** id="calculating\_data" class="tabcontent">

<**table** id="calculating\_table" class="table">

<**tr**>

<**th**>"Дата и время</**th**>

<**th**>'Время отклика</**th**>

<**th**>ЦПУ (сервер приложений)</**th**>

<**th**>Число сеансов</**th**>

<**th**>ЦПУ (сервер БД)</**th**>

<**th**>Память (сервер приложений")</**th**>

<**th**>Число блокировок</**th**>

</**tr**>

</**table**>

</**div**>

</**div**>

</**div**>

</**body**>

</**html**>

# **ПРИЛОЖЕНИЕ 5**

# Листинг программы генерации календаря событий

**export** **var** calendar;

document.addEventListener('DOMContentLoaded',

**function** () {

**var** calendarEl = document.getElementById('calendar');

calendar = **new** FullCalendar.Calendar(calendarEl, {

plugins: ['interaction', 'dayGrid', 'timeGrid', 'list', 'bootstrap'],

locale: 'ru',

height: 500,

themeSystem: 'bootstrap',

header: {

left: 'prev, next today',

center: 'title',

right: 'dayGridMonth, timeGridWeek, timeGridDay, listMonth, myButton'

},

buttonText: {

prev: '<-',

next: '->',

today: 'Сегодня',

dayGridMonth: 'Месяц',

timeGridWeek: 'Неделя',

timeGridDay: '�"ень',

listMonth: 'Список'

},

weekNumbers: true,

selectable: true,

editable: true,

eventClick: **function** (event, element) {

**alert**(JSON.**stringify**(event));

event.title = "CLICKED!";

$('#calendar').fullCalendar('updateEvent', event);

}

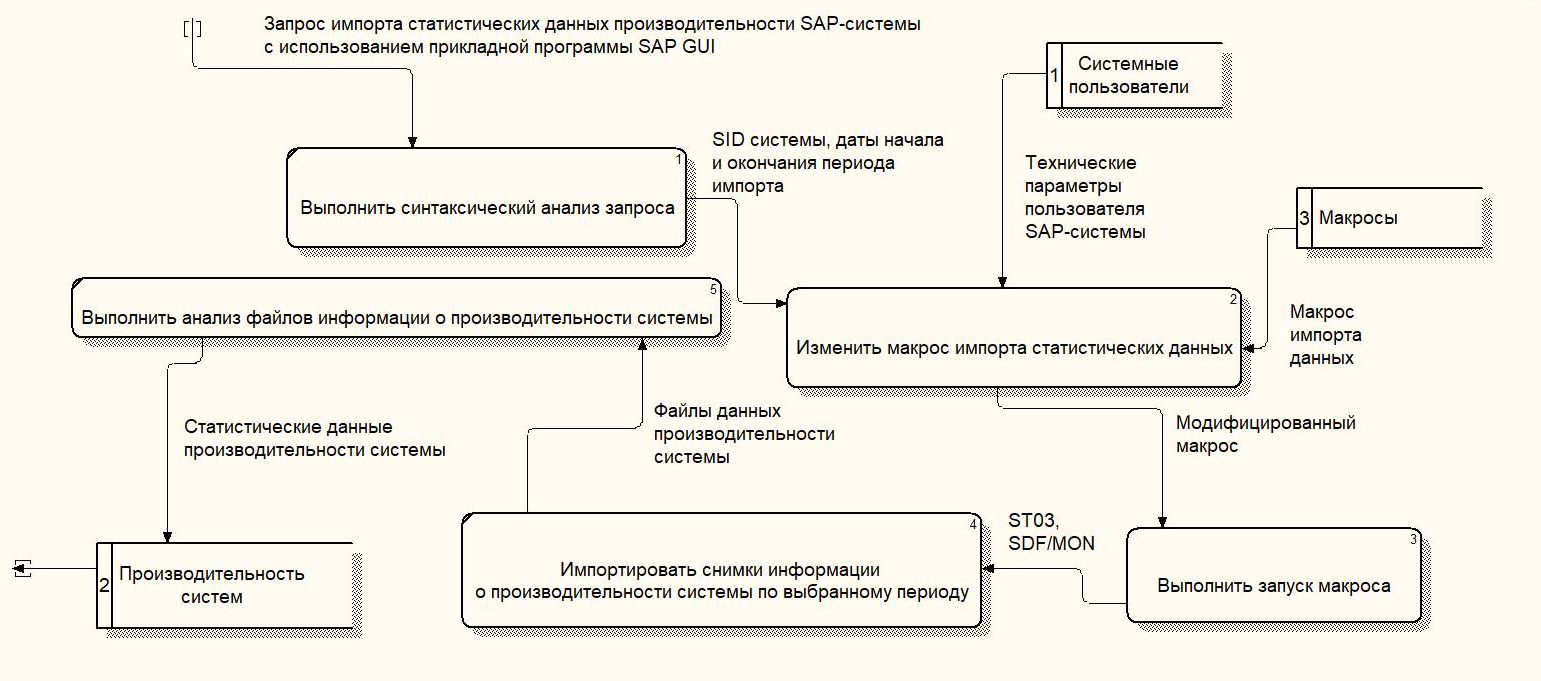
});

calendar.render();

});

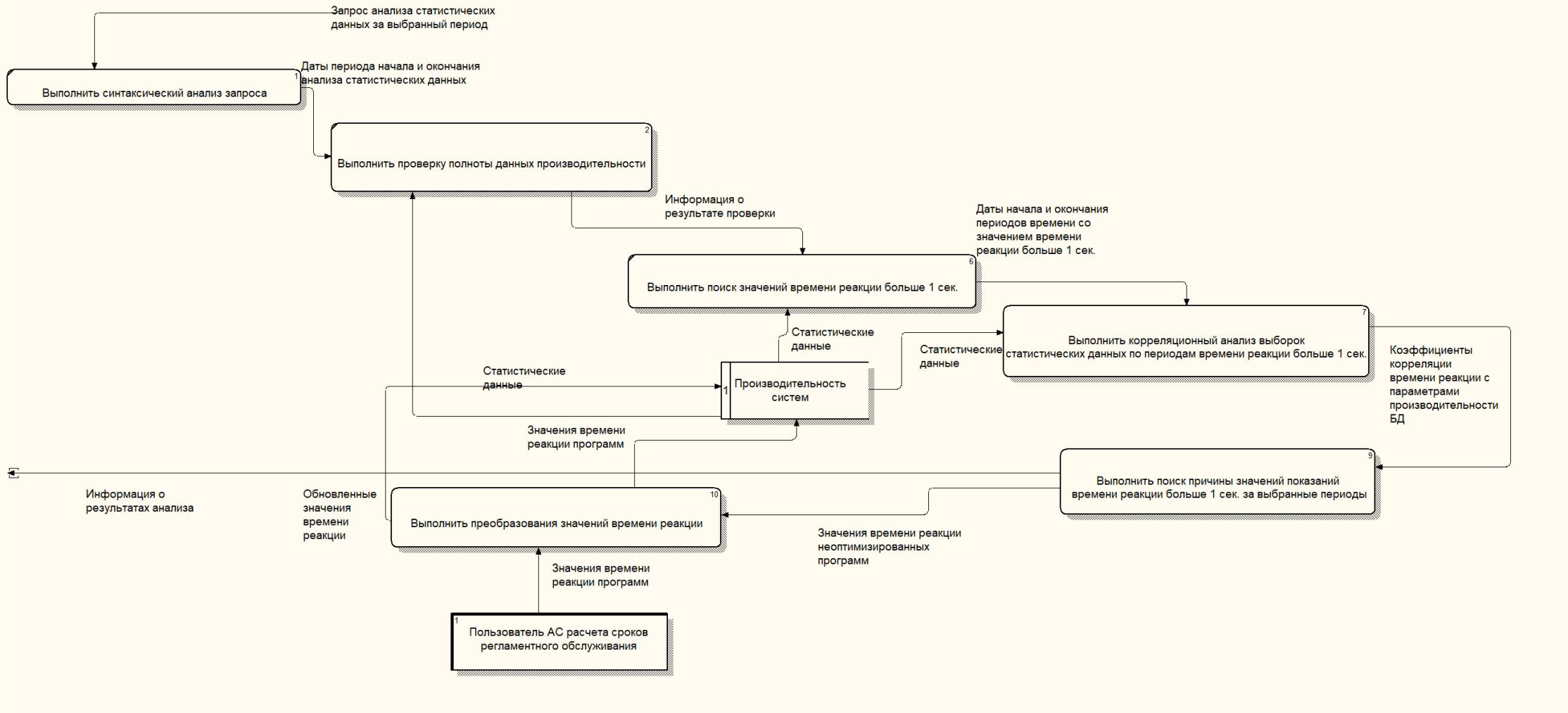
# **ПРИЛОЖЕНИЕ 6**

# Диаграмма процесса обработки запроса импорта статистических данных



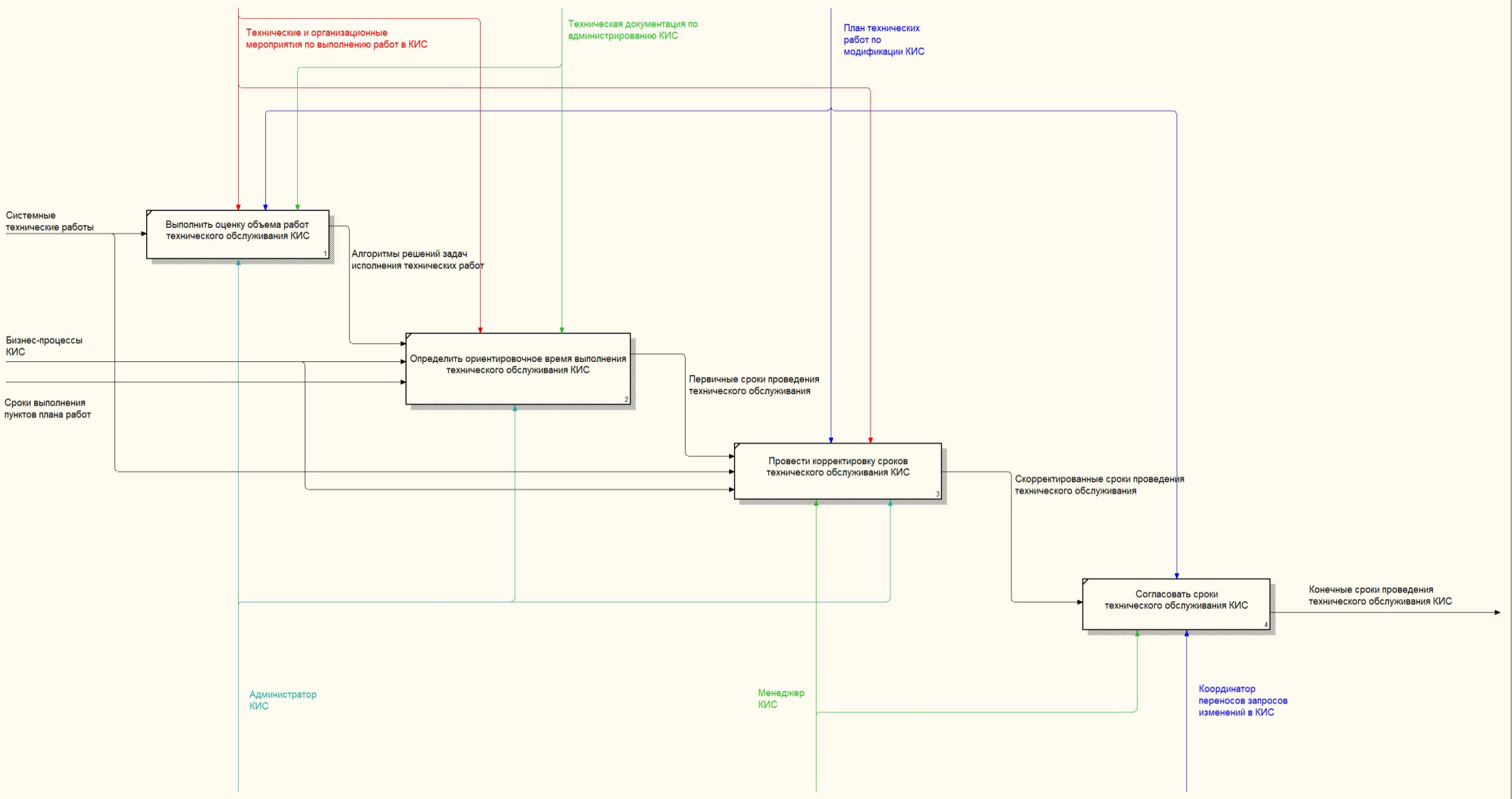
# **ПРИЛОЖЕНИЕ 7**

# Диаграмма процесса обработки запроса анализа статистических данных



# **ПРИЛОЖЕНИЕ 8**

# Декомпозиционная диаграмма первого уровня процесса определения сроков технического обслуживания



**ПРИЛОЖЕНИЕ 9**

Декомпозиционная диаграмма первого уровня процесса расчета сроков технического обслуживания

