

**Система универсального программного решения для
интеллектуальной поддержки принятия решений по
оптимизации и прогнозированию процессов
управления организацией**

Руководство администратора

Москва, 2023

СОДЕРЖАНИЕ

СПИСОК ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ	4
1 ВВЕДЕНИЕ	5
1.1 ОБЛАСТЬ ПРИМЕНЕНИЯ	5
1.2 УРОВЕНЬ ПОДГОТОВКИ АДМИНИСТРАТОРА	6
1.3 ПЕРЕЧЕНЬ ДОКУМЕНТОВ, С КОТОРЫМИ НЕОБХОДИМО ОЗНАКОМИТЬСЯ АДМИНИСТРАТОРУ	6
2 ПОДГОТОВКА К РАБОТЕ	7
2.1 НАСТРОЙКА СЕРВЕРА БАЗЫ ДАННЫХ POSTGRESQL.....	7
2.1.1 Установка базы данных.....	7
2.1.2 Развертывание и настройка базы данных Системы.....	7
2.2 НАСТРОЙКА СЕРВЕРА БАЗЫ ДАННЫХ REDIS.....	9
2.2.1 Подготовка сервера к настройке и установка.....	9
2.2.2 Проверка правильности установки	9
2.3 НАСТРОЙКА WEB-СЕРВЕРА.....	9
2.3.1 Конфигурирование специального ПО.....	9
2.4 РАЗВЕРТЫВАНИЕ И НАСТРОЙКА СИСТЕМЫ.....	10
2.4.1 Проверка правильности установки	10
2.4.2 Добавление службы systemd.....	11
2.5 НАСТРОЙКА СЕРВЕРА СБОРА ОШИБОК	11
2.6 ОПИСАНИЕ ОКРУЖЕНИЯ СИСТЕМЫ	12
3 ОПИСАНИЕ ОПЕРАЦИЙ	14
3.1 ПРОВЕРКА ПРАВИЛЬНОСТИ УСТАНОВКИ СИСТЕМЫ	14
3.2 ЗАПУСК WEB-СЕРВЕРА.....	14
3.3 КОНФИГУРИРОВАНИЕ СИСТЕМЫ (ДИРЕКТОРИИ РАЗМЕЩЕНИЯ ФАЙЛОВ, ПАРАМЕТРЫ СИСТЕМЫ И Т. Д.).....	14
3.4 ПРОСМОТР ЖУРНАЛА СОБЫТИЙ	15
3.5 ВЫДАЧА И ОПЕРАТИВНЫЙ ОТЗЫВ ПАРАМЕТРОВ АДМИНИСТРАТИВНОГО ДОСТУПА К ЭЛЕМЕНТАМ СИСТЕМЫ	17
3.6 РАЗГРАНИЧЕНИЕ ПРАВ ДОСТУПА НА ОСНОВАНИИ РОЛЕВОЙ МОДЕЛИ.....	17
3.7 ОТКАТ К ПРЕДЫДУЩЕЙ ВЕРСИИ	17
3.7.1 Откат артефактов.....	17
3.7.2 Откат изменений в БД.....	18
3.7.3 Откат конфигурации.....	18
3.7.4 Проверка работы web-сервера.....	18
4 АВАРИЙНЫЕ СИТУАЦИИ	19
5 КОНТРОЛЬ ЦЕЛОСТНОСТИ ОБРАБАТЫВАЕМЫХ И ТЕХНОЛОГИЧЕСКИХ ДАННЫХ СИСТЕМЫ И КОМПОНЕНТОВ	20

СПИСОК ТЕРМИНОВ И ОПРЕДЕЛЕНИЙ

Термин	Определение
Артефакт	Файл, содержащий скомпилированный исходный код, готовый к использованию сервером приложений
Возможность Системы	Совокупность действий Системы, направленная на достижение определенной цели
Пользователь	Лицо, которое использует систему для выполнения конкретной функции
Система, «Graph_PM»	Система универсального программного решения для интеллектуальной поддержки принятия решений по оптимизации и прогнозированию процессов управления организацией

СПИСОК ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

Сокращение	Расшифровка
БД	База данных
ИБ	Информационная безопасность
ОС	Операционная система
ПО	Программное обеспечение
СУБД	Система управления базами данных
ID	Идентификатор – уникальный признак объекта, позволяющий отличать его от других объектов.
IP адрес	(от англ. Internet Protocol Address «адрес Интернет-протокола») уникальный сетевой адрес узла в компьютерной сети, построенной на основе стека протоколов TCP/IP.
PostgreSQL 13.9	Ответвление от системы управления базами данных MySQL, разрабатываемое под лицензией GNU GPL
SQL	Декларативный язык программирования, применяемый для создания, модификации и управления данными
Redis	Резидентная система управления базами данных класса NoSQL с открытым исходным кодом
Ruby	Динамический язык программирования с открытым исходным кодом
Ruby On Rails	Веб-фреймворк с открытым кодом
URL	(от англ. Uniform Resource Locator «единый локатор ресурсов») единообразный определитель местонахождения ресурса
Web-сервер	Сервер, принимающий HTTP-запросы и выдающий им HTTP-ответы

1 ВВЕДЕНИЕ

1.1 Область применения

Универсальное программное решение для интеллектуальной поддержки принятия решений по оптимизации и прогнозированию процессов управления организацией (далее – Система, «Graph_PM») предназначено для:

- сбора информации о реально исполняемых процессах из транзакционных систем;
- структурирования информации в виде цепочек событий в журнал событий;
- визуализации цепочек событий в виде модели процесса;
- анализа процессов, в том числе:
 - выявление лучших экземпляров процессов и экземпляров с существенными отклонениями от существующих регламентных процедур;
 - сравнение процессов во времени;
 - прогнозирование хода производственных и бизнес-процессов в организации с использованием алгоритмов нейросетевого моделирования..

Использование возможностей универсального программного решения для интеллектуальной поддержки принятия решений по оптимизации и прогнозированию процессов управления организацией позволит:

- повысить скорость реагирования на риск и принятие решений менеджментом на основе полученной информации;
- получить инструменты сравнения состояния процессов, автоматизированных в ИС в динамике (ДО и ПОСЛЕ выполнения мероприятий, направленных на улучшения процесса);
- обеспечить возможность качественной оценки потенциала роботизации процессов;
- снизить трудоемкость аудита процессов и оценки их эффективности, обеспечив автоматический постоянный мониторинг хода их выполнения.
- провести симуляцию, моделирование, прогнозирование новых производственных и бизнес-процессов с использованием алгоритмов самообучающихся нейросетей;
- обеспечить визуализацию и мониторинг процессов информационного взаимодействия между системами.

1.2 Уровень подготовки администратора

Администратор Системы должен обладать навыками и знаниями, перечисленными ниже (Таблица 1).

Таблица 1. Навыки и знания администратора

№ п/п	Навыки и знания администратора
1.	Знание английского языка (технический перевод), сертификация не требуется
2.	Опыт администрирования UNIX-подобных операционных систем (далее — ОС) (уровень знаний определяется сертификатами «SUSE Certified Administrator (SCA) in Enterprise Linux», «Linux Professional Institute (LPI) / Advanced Level Linux Professional (LPIC-2)» и аналогичные)
3.	Опыт администрирования PostgreSQL / Redis
4.	Базовые знания языка программирования Ruby для настройки и запуска Системы

1.3 Перечень документов, с которыми необходимо ознакомиться администратору

Перед началом работы с Системой администратору необходимо ознакомиться с настоящим документом технической архитектурой ПО и Руководством пользователя.

2 ПОДГОТОВКА К РАБОТЕ

При подготовке к работе с Системой необходимо выполнить действия по установке и настройке.

Система работает в связке нескольких системных продуктов: Ruby, Ruby On Rails, PostgreSQL, Redis, а также дополнительных библиотек Ruby, которые устанавливаются из репозитория библиотек Ruby (<https://rubygems.org>).

Перед началом работы по установке Системы необходимо установить cURL с помощью команды:

```
sudo apt-get curl
```

Далее установите утилиту RVM для создания окружения на основе различных версий Ruby. Для установки используются следующие команды:

```
curl -sSL https://rvm.io/mpapis.asc | gpg --import -  
curl -sSL https://rvm.io/pkuczynski.asc | gpg --import -  
curl -sSL https://get.rvm.io | bash -s stable  
source /home/user/.rvm/scripts/rvm
```

Для установки Ruby введите команды:

```
sudo apt-get update  
rvm install 3.2.2.
```

Для установки библиотек Ruby введите следующую команду:

```
sudo apt-get install imagemagick libmagick++-dev libpq-dev git nodejs
```

2.1 Настройка сервера базы данных PostgreSQL

Указанные ниже действия выполняются на сервере \$ redis-server7.

2.1.1 Установка базы данных

Для установки серверной и клиентской частей PostgreSQL выполнить команду:

```
sudo apt-get install postgresql  
sudo systemctl enable postgresql
```

Установка будет следовать в автоматическом режиме. Во время установки появится диалог для задания пароля администратора (root) сервера PostgreSQL. Необходимо ввести и запомнить пароль для администратора сервера PostgreSQL.

2.1.2 Развертывание и настройка базы данных Системы

Далее необходимо создать базу данных Системы и пользователя базы данных для Системы. Для этого необходимо запустить клиентскую часть PostgreSQL, выполнив

команду:

```
sudo -u postgres psql
```

В процессе запуска клиентской части будет запрошен пароль администратора (root) сервера PostgreSQL, который был задан в процессе установки (Подраздел 2.1 данного документа).

Для создания базы данных приложения необходимо в клиентской части PostgreSQL выполнить следующую команду:

```
CREATE DATABASE наименование_БД;
```

Для создания пользователя базы данных Системы необходимо в клиентской части PostgreSQL выполнить следующую команду:

```
CREATE USER имя_пользователя WITH PASSWORD 'пароль';
```

Для выхода из клиентской части PostgreSQL необходимо выполнить команду:

```
exit
```

Для запуска Системы потребуется предварительно загрузить структуру и данные в базу данных. Данные предоставляются в архиве формата bz2. Для загрузки данных базы данных необходимо выполнить команду:

```
bzip2 -d < dump.sql.bz2 | sudo -u postgres psql наименование_БД
```

В процессе запуска загрузки снимка базы данных будет запрошен пароль администратора (root) сервера PostgreSQL, который был задан в процессе установки (Подраздел 2.1 данного документа).

Для проверки правильности загрузки снимка базы данных необходимо запустить клиентскую часть PostgreSQL, выполнив команду:

```
sudo -u postgres psql;
```

В процессе запуска клиентской части будет запрошен пароль администратора (root) сервера PostgreSQL, который был задан в процессе установки (Подраздел 2.1 данного документа).

Для проверки создания таблиц базы данных Системы необходимо в клиентской части PostgreSQL выполнить следующую команду:

```
\dt
```

В результате выполнения команды клиентская часть PostgreSQL выведет список таблиц базы данных приложения. Если список не пустой, то снимок базы данных был загружен корректно.

Для выхода из клиентской части PostgreSQL необходимо выполнить команду:

```
Exit
```

2.2 Настройка сервера базы данных Redis

Указанные ниже действия выполняются на сервере \$ redis-server7.

2.2.1 Подготовка сервера к настройке и установка

Для установки redis из репозитория ОС необходимо выполнить следующие команды:

```
sudo apt-get install redis
sudo systemctl enable redis-server
```

2.2.2 Проверка правильности установки

Для проверки работоспособности можно произвести подключение к серверу Redis. Используя команду INFO можно проверить настройки сервера.

Пример вывода служебной информации в данном документе не приводится, т.к. в каждом определенном случае он будет отличаться.

2.3 Настройка web-сервера

Указанные ниже действия выполняются на сервере \$ redis-server7.

2.3.1 Конфигурирование специального ПО

После завершения установки Ruby необходимо произвести обновление системного менеджера библиотек RubyGems. Для этого необходимо выполнить команду:

```
gem update --system
```

Далее при помощи системного менеджера библиотек RubyGems необходимо установить менеджер зависимостей пакетов Bundler. Для этого необходимо выполнить команду:

```
gem install bundler
```

После завершения установки Bundler необходимо приступить к копированию дистрибутива Системы.

После завершения процедуры копирования необходимо перейти в директорию Системы и запустить установку требуемых пакетов при помощи менеджера зависимостей пакетов Bundler. Для этого необходимо выполнить команду:

```
bundle install --without development test
```

Указанную команду необходимо выполнять только в директории Системы, т.к.

только там содержится описание пакетов зависимостей Системы.

Во время установки пакетов зависимостей Системы могут потребоваться дополнительные компоненты и пакеты операционной системы. Необходимо следовать указаниям менеджера пакетов Bundler, устанавливая недостающие компоненты и повторно выполнять запуск установки пакетов зависимостей Системы.

Для проверки правильности установки пакетов зависимостей Системы необходимо выполнить команду получения списка установленных пакетов зависимостей:

```
bundle list
```

В результате выполнения команды будет выведен построчный список установленных пакетов с указанием их версий.

2.4 Развертывание и настройка Системы

Далее необходимо указать параметры сервера балансировки нагрузки на сервера баз данных в конфигурационном файле подключения Системы к серверам баз данных, который расположен в директории Системы, в config/database.yml. Для редактирования файла конфигурации можно использовать любой установленный текстовый редактор, например, nano. Для запуска редактирования перейдите в директорию Системы и выполните команду:

```
cd приложение  
bundle install  
rake db:migrate RAILS_ENV=production
```

Необходимо запустить сборку ресурсов Системы, выполнив в директории Системы команду:

```
rake assets:precompile RAILS_ENV=production.
```

2.4.1 Проверка правильности установки

Для запуска консоли приложения введите команду:

```
rails c -e production
```

Если открылось окно консоли приложения, то развертывание системы произошло успешно.

Далее введите команду quit.

2.4.2 Добавление службы systemd

Для автоматизации запуска и перезапуска Системы необходимо запустить службу systemd с помощью следующей команды:

```
sudo nano /etc/systemd/system/puma.service
```

В открывшееся окно текстового редактора необходимо добавить следующий текст:

```
[Unit]
Description=APP HTTP Server
After=network.target

[Service]
Type=simple
WorkingDirectory=путь_к_папке_приложения
User=имя_системного_пользователя
ExecStart=/bin/bash -ilc 'bundle exec rails s -p 3000 -b 0 -e production'
Restart=always
ExecReload=/bin/kill -s SIGUSR1 $MAINPID
KillMode=mixed
RestartLimitBurst=5
StartLimitBurst=5

[Install]
WantedBy=multi-user.target
```

Для запуска приложения введите команды:

```
sudo systemctl daemon-reload
```

```
sudo systemctl start puma
```

```
sudo systemctl enable puma
```

2.5 Настройка сервера сбора ошибок

Разворачивается система errbit по инструкции производителя (доступна по

ссылке: <https://github.com/errbit/errbit> в разделе Installation).

2.6 Описание окружения Системы

В таблице ниже представлено описание аппаратного обеспечения (Таблица 2).

Таблица 2. Описание аппаратного обеспечения

№ п.п.	Параметр	Минимальное рекомендуемое значение	Примечание
1	Архитектура (разрядность) ОС	64-bit	В соответствии с архитектурой (разрядностью) процессоров
2	Количество ядер процессора	16	Количество ядер влияет на вычислительную производительность путем разделения выполняемых процессов между ядрами. Значение приведено из расчета одновременной работы 500 пользователей. Увеличение количества одновременно работающих пользователей потребует увеличения количества ядер
3	Объем оперативной памяти	64 Гб	Объем оперативной памяти влияет на вычислительную производительность, позволяя обрабатывать больше выполняемого машинного кода. Значение приведено из расчета одновременной работы 500 пользователей. Увеличение количества одновременно работающих пользователей потребует увеличения объема оперативной памяти.
4	Свободное дисковое пространство	1 ТБ	Объем занимаемого дискового пространства будет отводиться протоколированию действий пользователей, в расчете 100Гб в год. Также прирост БД при средней нагрузке составляет 300 Гб в год. Загружаемые пользователями файлы также будут постепенно занимать объем, который зависит от потребностей пользователей

В таблице ниже представлено описание программного окружения (Таблица 3).

Таблица 3. Описание программного окружения

№ п.п.	Программный компонент	Версия	Примечание
1	Ruby	3.2.2	На всех web-серверах Системы должна быть одна версия дистрибутива
2	Ruby on Rails	6.1.*	На всех web-серверах Системы должна быть одна версия дистрибутива
3	Gems	-	При запуске команды bundle install будут в автоматическом режиме установлены необходимые компоненты Системы
4	СУБД PostgreSQL	13.9 и выше	На всех серверах базы данных Системы должна быть одна версия ПО.
5	Redis	5.0 и выше	Резидентная СУБД класса NoSQL
6	Errbit	-	Сервер приложений
7	Resque	-	Сервер выполнения отложенных задач

3 ОПИСАНИЕ ОПЕРАЦИЙ

3.1 Проверка правильности установки Системы

Для проверки правильности установки Системы и проверки работоспособности Системы после установки необходимо запустить Систему на клиентской части в браузере.

Должна открыться экранная форма авторизации в Системе по составу реквизитов «Логин» и «Пароль»;

Необходимо выполнить вход, после этого откроется главная страница Системы.

3.2 Запуск web-сервера

Для запуска веб-сервера необходимо выполнить команду

```
sudo systemctl start puma
```

Благодаря запущенной службе systemd (п.2.4.2) сервер запустится автоматически.

3.3 Конфигурирование Системы (директории размещения файлов, параметры Системы и т. д.)

Директория размещения Системы: /home/user/graph_PM.

Приложение имеет протокол действий и записывает его в каталог, приведенный ниже (Рисунок 1).

```
< app_folder >/logs
```

Рисунок 1 Местоположение протокола действия в каталоге

Где app_folder - директория, в которую было скопировано приложение.

– Для проверки состояния Системы необходимо выполнить команду в директории Системы, приведенную ниже (Рисунок 2).

```
pgrep -fa 'puma'
```

Рисунок 2. Команда для проверки состояния Системы

После выполнения команды в консоль будет выведена актуальная информация о состоянии запущенного Системы.

В случае возникновения сбоев в работе Системы необходимо:

- выполнить остановку Системы (отключить СУБД и веб-сервер);
- запустить Сервис (запустить СУБД и веб-сервер);

- обратиться к администраторам системы.

3.4 Просмотр журнала событий

Приложение имеет протокол действий и записывает его в каталог, приведенный ниже (Рисунок 3).

```
< app_folder >/logs
```

Рисунок 3. Местоположение протокола действий

Где app_folder - директория, в которую было скопировано приложение.

Журналирование событий в Системе имеет стандартный формат.

Журналирование происходит с помощью методов:

- [дата_время];
- информация о событии: [Контроллер Action], которая включает параметры, пришедшие в данный контроллер, статус HTTP запроса в ответе и общее время генерации страницы в ответ (разбивка времени генерации ответа на время работы кода / время на работу с БД).

События пишутся каждый раз, как приходит запрос от пользователя, и/или от внутренних систем. Пример события указан ниже (Рисунок 4).

3.5 Выдача и оперативный отзыв параметров административного доступа к элементам Системы

До начала этапа опытной эксплуатации Подрядчик передает Заказчику полный набор логинов, паролей и других параметров доступа к модернизированному Сервису, необходимых для его развертывания и эксплуатации. При первичной авторизации на сервере приложений следует использовать логин «admin» и пароль «admin1», после успешной авторизации необходимо изменить эти данные.

Выдача и оперативный отзыв параметров административного доступа к элементам Системы осуществляется в несколько этапов:

1. подготовка и отправка заявки в эксплуатирующую организацию;
2. получение подтверждения от эксплуатирующей организации об успешном выполнении заявки;
3. учет факта выдачи или отзыва доступа.

3.6 Разграничение прав доступа на основании ролевой модели

В Системе реализована ролевая модель предоставления доступа к функциям, обеспечивающая:

- Возможность создавать и настраивать различные роли, для которых определены права доступа к данным и функциям.
- Поддержку глобальных и контекстных ролей.
- Возможность определения глубины прав доступа роли к данным.
- Возможность определения уровня прав доступа роли к данным (чтение, запись, отсутствие доступа, аннулирование, дополнение, архивирование, возврат из архива).
- Возможность определения области прав доступа к данным.
- Возможность определения прав доступа к данным в зависимости от даты создания данных.
- Возможность определять состав доступных функций для роли.

3.7 Откат к предыдущей версии

3.7.1 Откат артефактов

Удалить содержимое директории с артефактами сервера и скопировать в нее артефакты, сохраненные при резервном копировании.

3.7.2 Откат изменений в БД

Для отката миграционных изменений используется тот же механизм, что и для их проведения. Скрипты, проводящие изменения должны содержать секцию, описывающую как изменения откатить.

Для отката нескольких последовательных обновлений схемы БД необходимо:

- перейти в директорию Системы, описанную в подразделе 3.3 данного документа, и выполнить:

```
bundle exec rake db:rollback STEP=<count>
```

Где:

<count> - количество миграций, которые нужно откатить. Сообщается разработчиком при описании плана отката к предыдущей версии

Для отката определенных обновлений схемы БД необходимо:

- перейти в директорию Системы и выполнить:

```
bundle exec rake db:migrate:down VERSION=<version>
```

Где:

<version> - версия миграции, которую нужно откатить. Сообщается разработчиком при описании плана отката.

3.7.3 Откат конфигурации

Заменить конфигурацию кластера на сохраненный при резервном копировании файл конфигурации.

3.7.4 Проверка работы web-сервера

Для проверки состояния web-сервера необходимо открыть браузер и ввести в адресную строку IP адрес машины, на которой установлен web-сервер Системы. Если в браузере появится форма авторизации, то web-сервер работает исправно.

4 АВАРИЙНЫЕ СИТУАЦИИ

Аварийные ситуации классифицируются по следующему принципу:

- инфраструктура;
- сервер приложений;
- база данных.

Аварийные ситуации инфраструктурного характера детерминируются и разрешаются с использованием существующих средств мониторинга и управления инфраструктурой Системы, а также в соответствии с установленным порядком разрешения инцидентов данной категории. Устранение причин аварийных ситуаций в части инфраструктуры выполняется сотрудниками эксплуатирующей организации Системы.

Аварийная ситуация характеризуется полной или частичной недоступностью функциональных возможностей Системы в связи со сбоями в ее программно-техническом обеспечении.

В случае возникновения аварийной ситуации в ходе работы Системы Администратору необходимо:

- завершить работу всех видов прикладного ПО, с сохранением данных, в соответствии с официальной документацией поставщика системного и прикладного ПО, используемого для функционирования Системы;
- проверить наличие связи со смежными системами. проверить целостность данных;
- выполнить перезапуск Системы;
- если вышеперечисленные меры не помогли, выполнить повторную установку и настройку Системы, согласно настоящему документу.

После восстановления работоспособности Системы анализ ситуации производится совместными усилиями сотрудников технических служб смежных систем и, при необходимости, разработчиков Системы.

5 КОНТРОЛЬ ЦЕЛОСТНОСТИ ОБРАБАТЫВАЕМЫХ И ТЕХНОЛОГИЧЕСКИХ ДАННЫХ СИСТЕМЫ И КОМПОНЕНТОВ

Система обеспечивает корректную обработку неверных действий пользователей, некорректного формата или недопустимых значений входных данных. В указанных случаях Система выдает пользователю соответствующие сообщения, после чего возвращается в рабочее состояние, предшествовавшее неверной (недопустимой) команде или некорректному вводу данных.

Обеспечение целостности и сохранности данных, а также обработка нестандартных ситуаций в ходе работы Системы (таких как ввод неправильных данных, удаление используемой записи и т.д.), выполняются на нескольких архитектурных уровнях:

- пользовательские и программные интерфейсы, предоставляющие доступ к функциям Системы;
- программные модули, реализующие функции Системы;
- транзакционные механизмы СУБД, механизмы обеспечения целостности данных СУБД.

Связь архитектурных уровней с точки зрения обеспечения целостности и сохранности данных представлена ниже (Таблица 4).

Таблица 4. Связь архитектурных уровней с точки зрения обеспечения целостности и сохранности данных

№ п.п.	Действия в отношении данных	Архитектурный уровень	Способы обеспечения целостности и сохранности данных
1	Ввод данных	Пользовательские и программные интерфейсы, предоставляющие доступ к функциям Системы	Проверка корректности форматов, типов, наличия обязательных параметров
2	Обработка данных	Программные модули, реализующие функции Системы	Логические проверки данных
3	Хранение данных	СУБД	Механизмы транзакционности и обеспечения целостности, предоставляемые СУБД

Целостность и сохранность информации Системы обеспечивается мерами, реализованными средствами ППО и с помощью функций системного ПО:

- запись и хранение истории изменений данных;

- контроль входа в Систему;
- наличие процедур резервного копирования и восстановления;
- возможность возврата данных в корректное состояние.

В случае невозможности завершить пользовательскую транзакцию, а также в случае возникновения ошибки в работе программных средств, Система информирует пользователей о произошедшем сбое, выдавая аварийное сообщение и время устранения неисправности.