

	<b>Спецификация требований к программному обеспечению для Проекта СКАТ</b>	Автор: TEAM18 № документа: TEAM18-v0.1 Дата: 2023-10-10 Страниц: 100
---	--	---

## Содержание

СОДЕРЖАНИЕ	1
ИСТОРИЯ ИЗМЕНЕНИЙ	2
1 ВВЕДЕНИЕ	4
1.1 Цели	4
1.2 Границы применения	4
1.3 Термины, аббревиатуры, сокращения	5
1.4 Ссылки	5
1.5 Краткий обзор	5
2 ОБЩЕЕ ОПИСАНИЕ	5
2.1 Описание изделия	5
2.1.1 Интерфейсы системы	5
2.1.2 Интерфейсы пользователя	5
2.1.3 Интерфейсы аппаратных средств ЭВМ	5
2.1.4 Интерфейсы программного обеспечения	6
2.1.5 Интерфейсы коммуникаций	6
2.1.6 Ограничения памяти	6
2.1.7 Действия	6
2.1.8 Требования настройки рабочих мест	6
2.2 Функции изделия	7
2.3 Характеристики пользователей	7
2.4 Ограничения	7
2.5 Предположения и зависимости	7
2.6 Распределение требований	7
3 ДЕТАЛЬНЫЕ ТРЕБОВАНИЯ	8
3.1 Функциональные требования	8
3.1.1 <Functional Requirement One>	8
3.2 Надежность	8
3.2.1 <Reliability Requirement One>	8
3.3 Производительность	8
3.3.1 <Performance Requirement One>	8
3.4 Ремонтопригодность	5
3.4.1 <Maintainability Requirement One>	5
3.5 Ограничения проекта	5
3.5.1 <Design Constraint One>	5
3.6 Требования к пользовательской документации	5
3.7 Используемые приобретаемые компоненты	5
3.8 Интерфейсы	5
3.8.1 Интерфейс пользователя	5
3.8.2 Аппаратные интерфейсы	5
3.8.3 Программные интерфейсы	5
3.8.4 Интерфейсы коммуникаций	5
3.9 Требования лицензирования	5
3.10 Применимые стандарты	5



## История изменений

[illegible]

# 1 Введение

## 1.1 Цели

Этот документ определяет спецификацию требований к программному обеспечению системы контроля оборудования и техники (СКОТ). Системе необходимо хранить перечень ТО с полным набором инструкций необходимых действий и материалов, а также любых технических манипуляций с ОБ (ПИЗДЕЦ, ТО, эксплуатация).

Спецификация предназначена для работников взаимодействующих с ОБ и персонала занимающимся ТО и ПИЗДЕЦом оборудования.

## 1.2 Границы применения

СКОТ - система контроля оборудования и техники.

ПО будет выполнять следующие функции:

- Управление графиками обслуживания оборудования в заводе.
- Отслеживание регламентных работ (ТО) для каждого оборудования.
- Отслеживание незапланированных ремонтов (ПИЗДЕЦ) для каждого оборудования.
- Составление набора материалов и запчастей для каждого ТО.
- Расчет продолжительности обслуживания.

ПО не будет выполнять функции, отличные от управления и контроля оборудования, регламентного обслуживания и незапланированного ремонта.

Применение ПО Система контроля оборудования включает:

- Улучшенный мониторинг и контроль за графиками обслуживания оборудования.
- Автоматизированное составление наборов материалов и запчастей для каждого ТО.
- Точное планирование продолжительности обслуживания, учитывая разные критерии.

Важные преимущества включают повышение эффективности обслуживания, снижение риска отказов оборудования и оптимизацию использования ресурсов.

## 1.3 Термины, аббревиатуры, сокращения

СКОТ	Система Контроля Оборудования и Техники
ТО	Техническое Обслуживание, регламентные работы
ОБ	Оборудование
ПИЗДЕЦ	Незапланированный ремонт
ПО	Программное Обеспечение
ЖУРНАЛ	информация о отдельной единице ОБ
БД	База Данных
ЧС	Чрезвычайная Ситуация

## 1.4 Ссылки

Обозначение	Расшифровка
[IEEE-830]	IEEE Std 830-1998

## 1.5 Краткий обзор

Данный документ структурирован согласно [IEEE-830].

Раздел 2 содержит описание поставляемой системы и схему её использования в Организации. Раздел 3 содержит функциональные и нефункциональные требования, предъявляемые к системе и необходимые для её проектирования.

## 2 Общее описание

### 2.1 Описание изделия

#### 2.1.1 Интерфейсы системы

Веб-интерфейс: Система предоставляет веб-интерфейс для управления и мониторинга оборудования.

API: Предоставляется программный интерфейс приложения для интеграции с другими системами или автоматизации процессов.

#### 2.1.2 Интерфейсы пользователя

Графический интерфейс пользователя (GUI): Пользователи могут взаимодействовать с системой через удобный графический интерфейс, чтобы создавать, редактировать и просматривать данные об оборудовании и ТО.

Консольный интерфейс: Для более технических пользователей или администраторов доступен консольный интерфейс для выполнения команд и скриптов.

#### 2.1.3 Интерфейсы аппаратных средств ЭВМ

Взаимодействие с оборудованием:

- Коммуникация с оборудованием по временному графику: Интерфейс должен поддерживать настройку и управление временными параметрами оборудования, такими как заданные промежутки времени для обслуживания.
- Коммуникация с оборудованием по наработке: Интерфейс должен обеспечивать поддержку мониторинга наработки оборудования и уведомления о необходимости обслуживания на основе наработки.
- Одновременное взаимодействие с несколькими устройствами: Интерфейс должен иметь возможность обработки и управления несколькими устройствами одновременно, чтобы обеспечить эффективное планирование и координацию обслуживания.

Характеристики конфигурации:

- Номера портов: Программное обеспечение должно иметь возможность работы через определенные порты для коммуникации с оборудованием.
- Наборы инструкций: Интерфейс должен поддерживать соответствующие наборы инструкций, необходимые для взаимодействия с разными типами оборудования.

Поддерживаемые устройства:

- Интерфейс должен поддерживать широкий спектр устройств, которые требуют контроля и обслуживания, включая разнообразные типы оборудования, запчасти и материалы.

Способы поддержки и протоколы:

- Интерфейс должен обеспечивать возможность взаимодействия с оборудованием через различные методы связи, такие как последовательный порт, сетевой протокол или другие протоколы, поддерживаемые конкретными устройствами.

#### 2.1.4 Интерфейсы программного обеспечения

Для работы системы требуются:

- базы данных: PostgreSQL, Redis
- веб-сервер: Nginx
- среда выполнения языка JavaScript: Node.js

#### 2.1.5 Интерфейсы коммуникаций

Локальные сетевые протоколы:

- Ethernet: Интерфейс должен поддерживать протокол Ethernet для коммуникации с оборудованием, подключенным к локальной сети. Это обеспечит передачу данных между программным обеспечением и оборудованием через стандартные сетевые возможности.
- TCP/IP: Интерфейс должен поддерживать протокол TCP/IP для установки соединения и передачи данных по сети. TCP/IP является широко распространенным протоколом в сетевых коммуникациях и обеспечивает надежное и согласованное взаимодействие.

Беспроводные протоколы:

- Wi-Fi: Если в вашем заводе используется беспроводная сеть, то интерфейс должен поддерживать протокол Wi-Fi для беспроводной связи с оборудованием. Это позволит программному обеспечению обмениваться данными с оборудованием посредством беспроводной сети.
- Bluetooth: Если у вас есть оборудование, которое поддерживает связь по Bluetooth, то интерфейс должен поддерживать соответствующий протокол для установки связи и передачи данных через Bluetooth.

Протоколы удаленного доступа:

- SSH (Secure Shell): Интерфейс должен поддерживать протокол SSH для безопасного удаленного доступа к оборудованию. Это позволит операторам или техническим специалистам подключаться к оборудованию из удаленных мест и выполнять необходимые действия по контролю и обслуживанию.
- VPN (Virtual Private Network): Если ваш завод использует VPN для безопасного удаленного подключения, то интерфейс должен обеспечивать поддержку соответствующих VPN-протоколов, чтобы обеспечить защищенную коммуникацию с оборудованием посредством виртуальной частной сети.

#### 2.1.6 Ограничения памяти

Для развертки базы данных необходимо минимум 10 гигабайт на сервере.

Для оперативной памяти минимум 4 гигабайт.

#### 2.1.7 Действия

- Улучшенный мониторинг и контроль за графиками обслуживания оборудования.
- Автоматизированное составление наборов материалов и запчастей для каждого ТО.
- Точное планирование продолжительности обслуживания, учитывая разные критерии.

#### 2.1.8 Требования настройки рабочих мест

Рабочее место должно иметь минимальное разрешение экрана, которое обеспечит отображение всех необходимых элементов интерфейса программного обеспечения без искажений или обрезания информации.

Изменение конфигурации интерфейса: Программное обеспечение должно предоставлять возможность настроить интерфейс в соответствии с конкретными потребностями рабочего места

или задачи. Это может включать изменение расположения и размеров элементов интерфейса, добавление или удаление определенных функций или настройку предпочтений пользователя.

Поддержка различных языков: Если на заводе работают люди из разных стран или языковых групп, то программное обеспечение должно поддерживать многоязычный интерфейс. Это позволит настраивать программу на специфический язык для каждого рабочего места, облегчая взаимодействие с программой для всех пользователей.

Гибкость настроек времени и форматов: Рабочее место должно предоставлять возможность настройки временных параметров и форматов даты/времени в соответствии с требованиями местной временной зоны или корпоративных стандартов.

## **2.2 Функции изделия**

- Управление графиками обслуживания оборудования в заводе.
- Отслеживание регламентных работ (ТО) для каждого оборудования.
- Отслеживание незапланированных ремонтов (ПИЗДЕЦ) для каждого оборудования.
- Составление набора материалов и запчастей для каждого ТО.
- Расчет продолжительности обслуживания.

## **2.3 Характеристики пользователей**

- Образовательный уровень: Пользователи могут иметь различные уровни образования, от среднего до высшего. Они могут обладать базовыми навыками работы с компьютером и программным обеспечением.
- Опыт: Пользователи могут иметь разный опыт работы с оборудованием и программным обеспечением. Некоторые могут быть опытными и хорошо знакомыми с процессами обслуживания оборудования, в то время как другие могут быть новичками и нуждаются в дополнительной поддержке и руководстве.
- Техническая квалификация: Пользователи могут иметь разный технический уровень и квалификацию. Они могут быть инженерами, техниками, операторами оборудования и другими специалистами. Некоторые пользователи могут обладать специализированными знаниями в области обслуживания и ремонта определенного оборудования.

## **2.4 Ограничения**

- Доступ к БД должен быть разделен на несколько уровней привилегий от работника до админа.
- В случае ЧС связанных с отключением или ошибкой(ПО) , ПО должно сохранить все данные, для последующего восстановления в ограниченный срок.
- БД должна поддерживать редактирование несколькими пользователями если они не редактируют один ЖУРНАЛ.

## **2.5 Предположения и зависимости**

- На территории предприятия должна быть организована локальная сеть для подключения всего ОБ
- Связь с мировым интернетом осуществляется только через центральный сервер
- Пользователи прошли обучение по работе с ПО и знакомы с техникой безопасности каждого отдельного ОБ

## **2.6 Распределение требований**

- Поддержка Безопасности ПО уровня AA+.ru

### 3 Детальные требования

This section of the **SRS** should contain all the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. When using use-case modelling, these requirements are captured in the Use-Cases and the applicable supplementary specifications.]

#### 3.1 Функциональные требования

[This section describes the functional requirements of the system for those requirements which are expressed in the natural language style. For many applications, this may constitute the bulk of the **SRS** Package and thought should be given to the structure of this section. This section is typically structured by feature, but alternative structures may also be appropriate, for example, structure by user or by subsystem. Functional requirements may include feature sets, capabilities, and security.

Where application development tools, such as requirements tools, modelling tools, etc., are employed to capture the functionality, this section will refer to the availability of that data, indicating the location and name of the tool that is used to capture the data.]

##### 3.1.1 <Functional Requirement One>

[The requirement description.]

#### 3.2 Надежность

[Requirements for reliability of the system should be specified here. Some suggestions follow:

- Availability—specify the percentage of time available ( xx.xx%), hours of use, maintenance access, degraded mode operations, etc.
- Mean Time Between Failures (MTBF) — this is usually specified in hours, but it could also be specified in terms of days, months or years.
- Mean Time To Repair (MTTR)—how long is the system allowed to be out of operation after it has failed?
- Accuracy—specify precision (resolution) and accuracy (by some known standard) that is required in the system's output.
- Maximum Bugs or Defect Rate—usually expressed in terms of bugs per thousand of lines of code (bugs/KLOC) or bugs per function-point( bugs/function-point).
- Bugs or Defect Rate—categorized in terms of minor, significant, and critical bugs: the requirement(s) must define what is meant by a “critical” bug; for example, complete loss of data or a complete inability to use certain parts of the system's functionality.]

##### 3.2.1 <Reliability Requirement One>

[The requirement description.]



### 3.3 Производительность

[The system's performance characteristics should be outlined in this section. Include specific response times. Where applicable, reference related Use Cases by name.]

- response time for a transaction (average, maximum)
- throughput, for example, transactions per second
- capacity, for example, the number of customers or transactions the system can accommodate
- degradation modes (what is the acceptable mode of operation when the system has been degraded in some manner)
- resource utilization, such as memory, disk, communications, etc.

#### 3.3.1 <Performance Requirement One>

[The requirement description goes here.]

### 3.4 Ремонтопригодность

[This section indicates any requirements that will enhance the maintainability of the system being built, including coding standards, naming conventions, class libraries, maintenance access, maintenance utilities.]

#### 3.4.1 <Maintainability Requirement One>

[The requirement description goes here.]

### 3.5 Ограничения проекта

[This section should indicate any design constraints on the system being built. Design constraints represent design decisions that have been mandated and must be adhered to. Examples include software languages, software process requirements, prescribed use of developmental tools, architectural and design constraints, purchased components, class libraries, etc.]

#### 3.5.1 <Design Constraint One>

[The requirement description goes here.]

### 3.6 Требования к пользовательской документации

[Describes the requirements, if any, for on-line user documentation, help systems, help about notices, etc.]

### 3.7 Используемые приобретаемые компоненты

[This section describes any purchased components to be used with the system, any applicable licensing or usage restrictions, and any associated compatibility and interoperability or interface standards.]

## **3.8 Интерфейсы**

[This section defines the interfaces that must be supported by the application. It should contain adequate specificity, protocols, ports and logical addresses, etc. so that the software can be developed and verified against the interface requirements.]

### **3.8.1 Интерфейс пользователя**

[Describe the user interfaces that are to be implemented by the software.]

### **3.8.2 Аппаратные интерфейсы**

[This section defines any hardware interfaces that are to be supported by the software, including logical structure, physical addresses, expected behaviour, etc. ]

### **3.8.3 Программные интерфейсы**

[This section describes software interfaces to other components of the software system. These may be purchased components, components reused from another application or components being developed for subsystems outside of the scope of this **SRS** but with which this software application must interact.]

### **3.8.4 Интерфейсы коммуникаций**

[Describe any communications interfaces to other systems or devices such as local area networks, remote serial devices, etc.]

## **3.9 Требования лицензирования**

[Defines any licensing enforcement requirements or other usage restriction requirements that are to be exhibited by the software.]

## **3.10 Применимые стандарты**

[This section describes by reference any applicable standard and the specific sections of any such standards which apply to the system being described. For example, this could include legal, quality and regulatory standards, industry standards for usability, interoperability, internationalization, operating system compliance, safety, security, etc.]

## **Индекс**