

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Институт информационных технологий и анализа данных  
наименование института

Допускаю к защите  
Руководитель



подпись

И.А. Серышева  
И.О. Фамилия

Основные проектные решения по разработке информационной системы для  
автоматизации конфигурирования сетевых устройств корпоративной  
вычислительной сети  
наименование темы

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
к курсовому проекту по дисциплине**

Проектирование информационных систем

1.006.00.00 ПЗ

обозначение документа

Выполнил студент

ИСТ6-19-2  
инфр группы



подпись

Д.В. Качалин  
И.О. Фамилия

Нормоконтроль



подпись

И.А. Серышева  
И.О. Фамилия

Курсовой проект защищен с оценкой



Иркутск 2022 г.

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ЗАДАНИЕ  
НА КУРСОВОЕ ПРОЕКТИРОВАНИЕ**

По курсу Проектирование информационных систем

Студенту Качалину Д.В.  
(фамилия, инициалы)

Тема проекта Основные проектные решения по разработке информационной системы для автоматизации конфигурирования сетевых устройств корпоративной вычислительной сети

Исходные данные Программа IT-academy, направление сетевой и серверной инфраструктуры – Концепция проекта;  
Производственная практика (научно-исследовательская работа).

Рекомендуемая литература: Гутгарц, Римма Давыдовна. Проектирование автоматизированных систем обработки информации и управления : учебное пособие для академического бакалавриата по инженерно-техническим и экономическим направлениям / Р.Д. Гутгарц. – Санкт-Петербург : Юрайт, 2019. – 303 с.

Графическая часть на \_\_\_\_\_ листах.

Дата выдачи задания « 26 » сентября 2022 г.

Задание получил \_\_\_\_\_ Д.В. Качалин  
подпись И.О. Фамилия

Дата представления проекта руководителю « 23 » декабря 2022 г.

Руководитель курсового проектирования \_\_\_\_\_ И.А. Серышева  
подпись И.О. Фамилия

## Словарь терминов и сокращений

Термин	Значение
автоматизация	освобождение или уменьшение участия человека в каком-либо процессе.
конфигурирование	настройка или корректировка параметров конфигурации устройства.
конфигурация	набор программного и аппаратного обеспечения устройства.
мониторинг	наблюдение за процессами.
сетевое устройство	устройства, обеспечивающие связь и взаимодействие между устройствами в компьютерной сети.
компьютерная сеть, сетевая инфраструктура	система, обеспечивающая передачу данных между вычислительными устройствами.
документирование	процесс закрепления информации на материальном носителе.
ПО	Программное обеспечение. Совокупность программ и документации для использования устройства в решении задач.
CLI	Command-line interface. Способ взаимодействия человека с устройством путем отправки команд в виде последовательности символов.
SSH	Secure Shell. Протокол удаленного управления ОС устройства.
протокол	«язык», используемый компьютерами для обмена данными при работе в сети.
ОС	операционная система. ПО, управляющее устройством.
IP	Internet Protocol. Набор правил, регулирующий формат данных, передаваемых через глобальную или локальную сеть.
IP-адрес	уникальный идентификатор устройства в сети.
IP-пространство	совокупность IP-адресов определенной сети.
подсеть	логическое разделение IP-пространства.
автоматизация	освобождение или уменьшение участия человека в каком-либо процессе.
VLAN	Виртуальная локальная компьютерная сеть. Представляет собой группу хостов с общим набором требований, которые взаимодействуют так, как если бы они были подключены к широковещательному домену независимо от их физического местонахождения.
инициализация устройства	создание, активация, подготовка к работе, определение параметров. Приведение программы или устройства в состояние готовности к использованию.
полигон (полигон для тестирования)	Виртуальная сеть или часть сети, на которой проводится тестирование конфигураций устройств.
playbook	специальные программы Ansible, которые позволяют объявлять конфигурацию устройств, управлять порядком любой ручной установки в определенном порядке на нескольких группах устройств.
ЭВМ	электронное вычислительное устройство. Комплекс технических, аппаратных и программных средств, предназначенных для автоматической обработки информации, вычислений, автоматического управления.
администрирование	управление, руководство, распоряжение, менеджмент.
безопасность информации	такое состояние защищенности информации, при котором обеспечены её конфиденциальность, целостность и доступность.
конфиденциальность	необходимость предотвращения разглашения, утечки какой-либо информации.

<b>Термин</b>	<b>Значение</b>
целостность	данные не были изменены при выполнении любой операции над ними, будь то передача, хранение или представление.
доступность	состояние информации, при котором субъекты, имеющие права доступа, могут реализовывать их беспрепятственно.
информационная безопасность	Практика предотвращения несанкционированного доступа, использования, раскрытия, искажения, изменения, исследования, записи или уничтожения информации.
сервер	выделенный или специализированный компьютер для выполнения сервисного программного обеспечения.
сервис	процесс, который обычно предоставляет услуги по сети и выполняется на сервере
эксплуатация	исполнение программ на ЭВМ для обработки информации и получения результатов, являющихся целью создания ПО, а также в обеспечении достоверности и надежности выдаваемых данных.
БП	Бизнес-процесс – это логическая последовательность действий человека (или нескольких человек) в коллективе. Цель описания бизнес-процесса – анализ и регламентация тех или иных действий в коллективе.

## Содержание

Введение.....	6
1 Анализ предметной области и обоснование необходимости разработки системы.....	8
1.1 Моделирование процессов конфигурирования и документирования сетевых устройств в компании EN+ DIGITAL .....	8
1.2 Назначение и цели создания системы.....	26
1.3 Требования к системе .....	26
1.4 Обзор систем автоматизации конфигурирования и документирования сетевой инфраструктуры .....	32
1.5 Описание выбранных средств реализации системы .....	36
2 Проектирование системы .....	39
2.1 Моделирование процессов работы с сетевыми устройствами с учетом внедрения системы.....	39
2.2 Варианты использования системы .....	57
2.3 Разработка программ автоматизации.....	65
2.4 Архитектура системы .....	70
2.5 Развертывание системы .....	72
3 Реализация.....	74
3.1 Состав и содержание работ по созданию системы.....	74
3.2 Фактически выполненные работы и полученные результаты .....	76
3.3 Примеры наработки системы.....	77
3.4 Анализ выполненной работы и дальнейшие планы .....	82
Заключение .....	83
Список использованных источников .....	84

## Введение

С давних времен задача по передачи информации из одной точки в другую была востребована. С развитием информационных и компьютерных технологий этот факт не изменился. Но изменились и скорость, и объемы передаваемых данных. Ещё в 2000 году средняя скорость передачи данных составляла от 32 до 128 Кбит/с. Чтобы скачать файл размером 1 Гбайт, было необходимо больше 18 часов. Сегодняшняя скорость позволяет в течение 10 минут скачивать подобные объемы информации [1].

Вместе с развитием технологий передачи данных, развивался и инструментарий сетевого инженера для управления компьютерными сетями. Было разработано множество различного ПО для мониторинга сетей и сетевых устройств, инвентаризации и сбора данных: Zabbix, NetFlow Analyzer, WhatsUp Gold, NetBox и другие. Но инструментарий для автоматизации сети практически не развивался. На протяжении многих лет сетевые инженеры используют самописные скрипты и заготовленные заранее конфигурации, чтобы настраивать сетевые устройства. И только в последние несколько лет начали появляться ПО для автоматизации рутинных задач при работе с сетевыми устройствами [2].

Компания EN+ DIGITAL занимается предоставлением IT-услуг компаниям партнерам. Среди которых и обслуживание сетевого оборудования. Сотрудники сетевого отдела ежедневно обслуживают более 2000 устройств. И хоть на данный момент сотрудники справляются со своими задачами, руководство отдела планирует внедрение системы автоматизации конфигурирования и документирования сетевых устройств. Несмотря на довольно высокую стоимость лицензии, основная проблема покупки специализированного ПО заключается не в цене, а в том, что не известно какой результат повлечет внедрение такой системы. К тому же для использования системы требуется квалифицированный сотрудник, что также занимает время. Поэтому было решено создать прототип на основе бесплатного ПО, чтобы понять необходимость внедрения автоматизации в бизнес-процессы при работе с сетевым оборудованием

Цель создания данной системы – уменьшить время, проводимое сотрудником сетевого отдела ООО "ЭН+ ДИДЖИТАЛ" на решение рутинных задач, связанных с обслуживанием сетевых устройств за счет введения автоматизированной системы конфигурирования и документирования сетевых устройств в корпоративной вычислительной информационной сети.

Идея создания подобной системы появилась во время прохождения производственной практики. При анализе бизнес-процессов сетевого отдела компании было выявлено, что специалисты отдела вручную проводят настройку сетевого оборудования (при помощи заранее заготовленных конфигураций). Это не только расходует время работы сотрудника на рутинные задачи, но также имеет влияние со стороны человеческого фактора, которое может привести к неполадкам в работе сети. А это в свою очередь к неполадкам в работе всей компании и её партнеров.

Результат работы, т.е. систему автоматизации, стоит рассматривать как прототип системы для автоматизации рутинных задач, выполняемых в сетевом отделе при работе с сетевым оборудованием. Данная система создается в рамках:

- курсового проекта по дисциплине «Проектирование информационных систем»;

- проектного обучения Ирниту;

- проекта программы IT-academy совместно с EN+ Group и Русал.

После внедрения и эксплуатации будет решено стоит ли развивать функциональность данного прототипа и доводить его до полноценной системы или отказаться от него. Также возможен вариант приобретения лицензий готового ПО, решающих задачи работы с сетевым оборудованием.

## **1 Анализ предметной области и обоснование необходимости разработки системы**

В данном разделе приводится анализ предметной области, т.е. обоснование причин создания прототипа системы автоматизации задач конфигурирования и документирования сетевого оборудования в компании EN+ DIGITAL.

### **1.1 Моделирование процессов конфигурирования и документирования сетевых устройств в компании EN+ DIGITAL**

На сегодняшний день в компании EN+ DIGITAL обслуживается более 2.000 сетевых устройств. Задачами мониторинга, конфигурирования, программным и аппаратным обслуживанием занимается отдел «По адаптации сетевой инфраструктуры для программ ЭВМ, баз данных». В отделе работает около 30 сетевых инженеров.

На данный момент большинство задач отдела являются неавтоматизированными. Для решения часто возникающих рутинных задач по настройке оборудования, в большинстве случаев используется готовая конфигурация (или её шаблон), которую сотрудник корректирует для каждого устройства индивидуально. И только потом подключается к устройству через telnet или ssh вставляет конфигурацию. Такое решение задач крайне неэффективно расходует время сотрудника и к тому же связано с риском возникновения неисправностей в работе устройства или даже в части сети (ввиду человеческого фактора). Также присутствует и другой вариант решения малого числа задач – ранее написанные скрипты. Это частичная автоматизация, которая во многом схожа с проблемами ручной настройки по шаблонам. Скрипт нужно корректировать, чтобы он корректно отработал с различными устройствами. Это может также вызвать неполадки в работе сетевого оборудования при совершении ошибки в исправлении скрипта.

Помимо конфигурации, стоит проблема и с документированием сетевых устройств. На сегодняшний день в компании ведется excel файл инвентаризации, который храниться в общей папке. Это затрудняет работу по анализу информации: не известно какая информация актуальная, а какая – нет. Процесс ввода новой информации или обновление устаревшей также является затруднительным и связан с человеческим фактором.

Помимо инвентаризации вводится документирование сети при помощи схем Visio, Zabbix и WhatsUp Gold. Создание и обновление схем связано с процессом сбора информации с устройств. Сбор информации осуществляется в ручном режиме: специалист заходит в файл инвентаризации, фильтрует по интересующей его части сети, перебором заходит на устройства, собирает с них необходимую информации, анализирует и только потом начинает документировать. Процесс очень рутинный и занимает много времени. Так как в файле инвентаризации не всегда актуальная информация, это приводит к неточностям или ошибкам в построении схем.

Чтобы детальней изучить проблему стоит смоделировать бизнес-процессы, связанные с выполнением задач конфигурирования и документирования сетевых устройств на данный момент в компании.



Моделирование бизнес-процессов компании было выполнено при помощи методологии IDEF0 [3]. Модель AS-IS позволяет наглядно представить существующие бизнес-процессы в компании. Данную модель можно использовать для анализа слабых мест в процессах или мест возможной оптимизации.

Для более компактного представления бизнес-процессов использовалось туннелирование стрелок диаграмм. Туннели позволяют указать, где детализация определенной сущности на диаграмме является избыточной. В таблице 1.1 приводится описание всех туннелей при моделировании бизнес-процессов AS-IS.

Таблица 1.1 – Описание туннелей для модели процессов AS-IS.

Сущность	На каких уровнях диаграммы встречается	Описание применения туннеля
Сотрудник отдела EN+ DIGITAL	Все	Так как сотрудник участвует во всех процессах нижних уровней, то его появление обозначается туннелем. Это означает, что в диаграммах нижнего уровня он не обозначается, но подразумевается его присутствие во всех блоках в виде исполняющего бизнес-процессы
Регламент документирования устройства	A0 A2 A21 A22 A221 A222 A223	Используется в качестве нормативного документа для документирования сетевых устройств. Обозначается туннелем для процесса «Документирование устройства». Не отображается на нижнем уровне процесса, но подразумевается его использование.
Регламент конфигурирования устройства	A1 A12	Используется в качестве нормативного документа для конфигурирования сетевых устройств. Обозначается туннелем для процесса «Написание конфигурации». Не отображается на нижних уровнях процесса, но подразумевается его использование.
Текстовый редактор	A1 A12	Используется как инструмент для написания конфигурации. Обозначается туннелем для процесса «Написание конфигурации». Не отображается на нижнем уровне процесса, но подразумевается его использование.

Продолжение таблицы 1.1

Сущность	На каких уровнях диаграммы встречается	Описание применения туннеля
Zabbix.	A22 A221	Используется как инструмент для документирования сети. Обозначается туннелем для процесса «Работа с Zabbix». Не отображается на нижнем уровне процесса, но подразумевается использование его.
WhatUp Gold	A22 A222	Используется как инструмент для документирования сети. Обозначается туннелем для процесса «Работа с WhatUp Gold». Не отображается на нижнем уровне процессов, но подразумевается использование его.
NetFlow Analyzer	A22 A223	Используется как инструмент для документирования сети. Обозначается туннелем для процесса «Работа с NetFlow Analyzer». Не отображается на нижнем уровне процессов, но подразумевается его использование.
Excel	A2 A21	Используется как инструмент для инвентаризации сети. Обозначается туннелем для процесса «Изменение файла инвентаризации». Не отображается на нижнем уровне процесса, но подразумевается его использование в качестве инструмента.

Далее приводится детальное описание основных бизнес-процессов, связанных с обслуживанием сетевого оборудования в компании.

### 1.1.1 Работа с сетевым устройством (А-0)

На рисунке 1.1 представлена диаграмма IDEF0, описывающая верхний уровень бизнес-процесса «Работа с сетевым устройством» (А-0) на данный момент в компании.

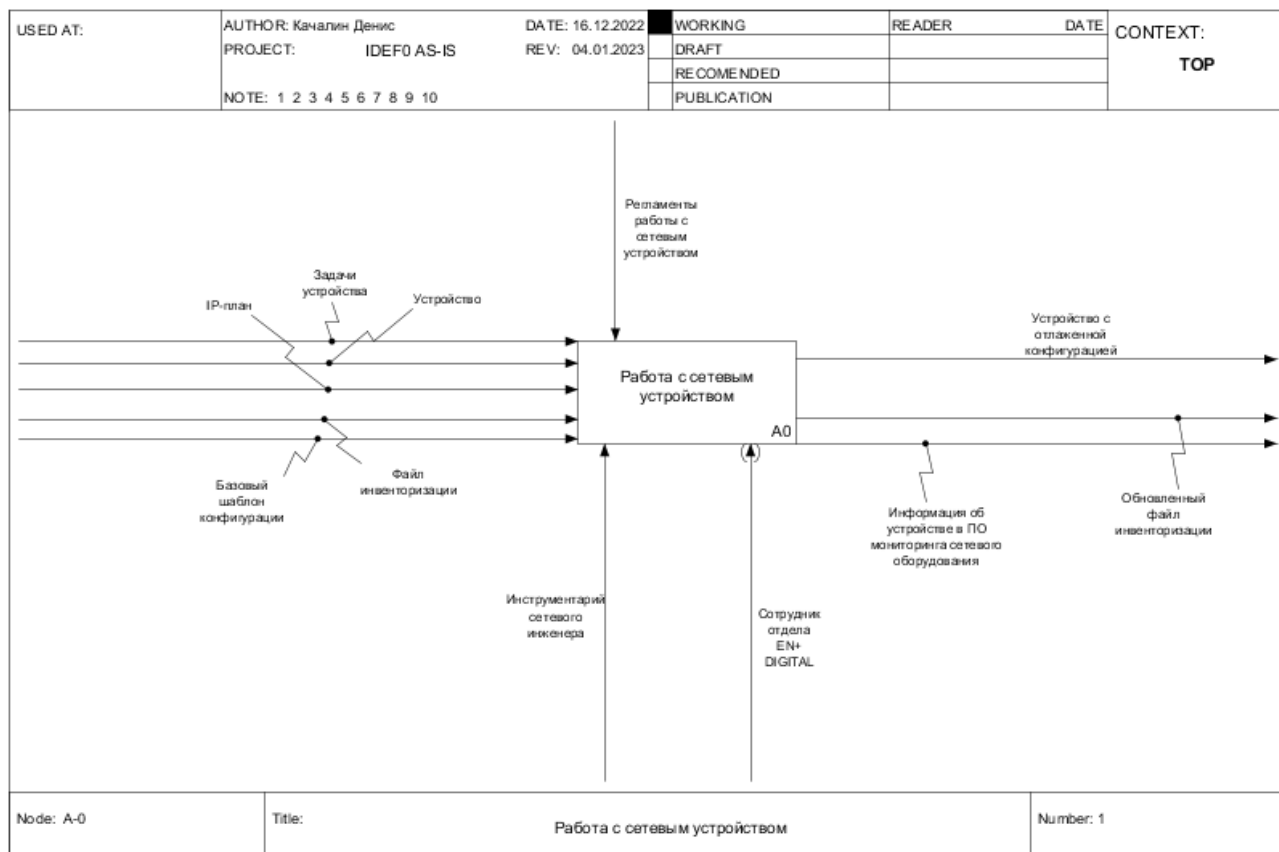


Рисунок 1.1 – Диаграмма IDEF0 AS-IS для бизнес-процесса «Работа с сетевым устройством» верхнего уровня.

На данный момент для работы с сетевым оборудованием сотрудник сетевого отдела EN+ DIGITAL использует инструментарий сетевого инженера, чтобы настраивать и документировать устройства.

Чтобы работать с сетевым устройством ему требуются:

- задачи устройства. Какие функции оно должно выполнять;
- устройство, обслуживание которого проводится;
- IP-план сети, в которой устройство должно находиться;
- файл инвентаризации. Файл в формате excel, который хранит всю инвентарную информацию о сетевых устройствах;
- базовый шаблон конфигурации. Текстовый файл, в котором храниться универсальная для большинства часть конфигурации устройства.

Работа с устройством выполняется согласно регламентам компании и сетевого отдела по работе с сетевым оборудованием.

В результате процесса ожидаются следующие результаты:

- устройство с отлаженной конфигурацией;
- файл инвентаризации, в котором добавилась или обновилась информация об устройстве;
- информация в ПО для мониторинга сетевого оборудования, которое уже работает в отделе.

### 1.1.2 Работа с сетевым устройством (A0)

На рисунке 1.2 представлена диаграмма IDEF0 AS-IS, детализирующая описание бизнес-процесса «Работа с сетевым устройством» (A0) на данный момент в компании.

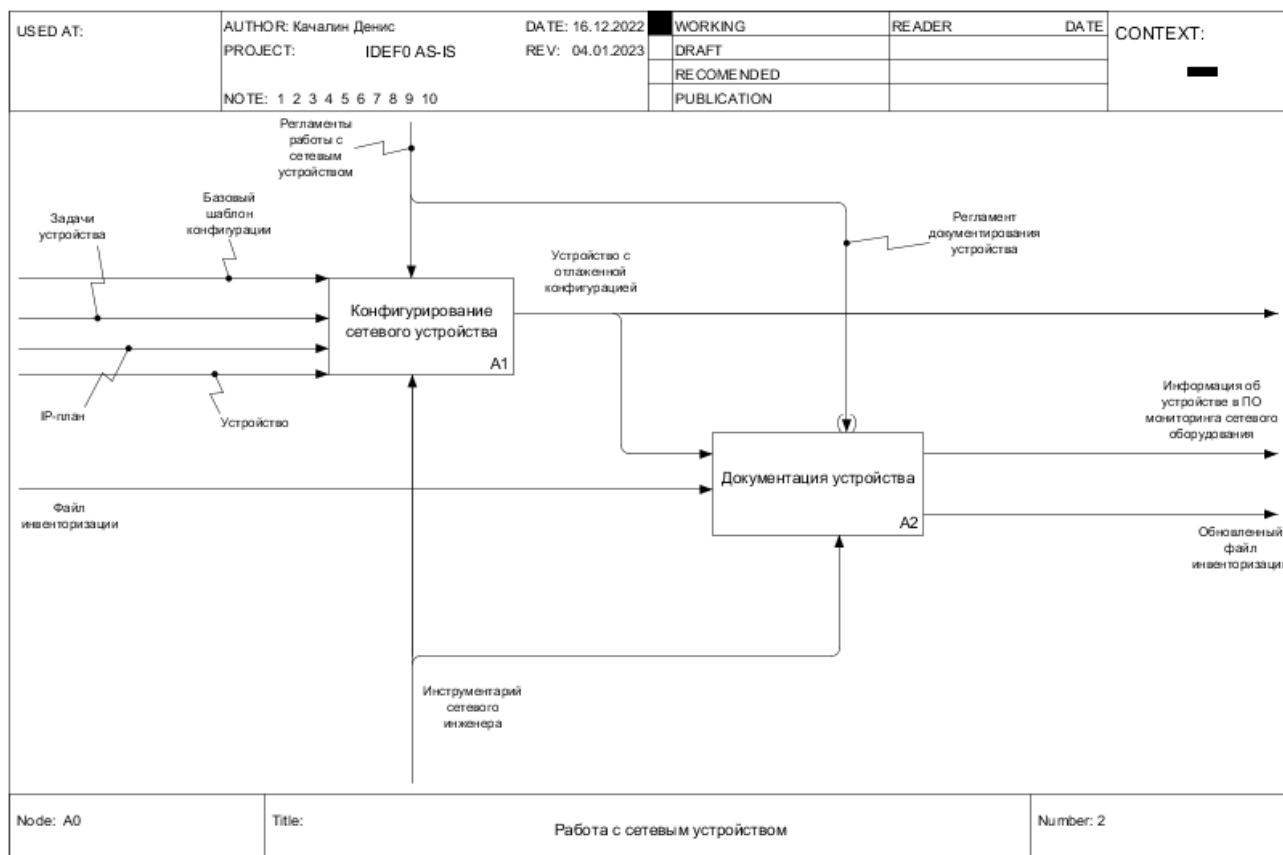


Рисунок 1.2 – Диаграмма IDEF0 AS-IS для бизнес-процесса «Работа с сетевым устройством» нижнего уровня.

Два основных бизнес-процесса, посвященных работе с сетевыми устройствами – это конфигурирование и документирование.

#### Конфигурирование

На данный момент устройство конфигурируется, если оно:

- не сконфигурировано (новое устройство или со склада);
- вызывает неполадки в работе сети;
- имеет несоответствующую функциональность (добавление или уменьшение функциональности).

Конфигурирование – процесс нечастый, но требует много времени на подготовку и исполнение.

## Документирование

Документирование – крайне хаотичный процесс из-за влияния человеческого фактора. Часты случаи, когда устройство заменили новым или сменили на нем конфигурацию, но изменения нигде обозначены не были. Такие случаи выявлялись только при полной инвентаризации части сети, во время прохождения производственной практики. Поэтому в файле инвентаризации могут как отсутствовать записи о новых устройствах, так и записи, где информация уже неактуальна.

Из-за такой хаотичной документации страдает и ПО для мониторинга сетевого оборудования, так как, чтобы занести в него какую-либо часть сети необходимо связываться с человеком, работающем на объекте и запрашивать актуальную информацию. Информацию, которую сотруднику ещё будет необходимо собрать.

Также хаотичная документация может замедлять время решения некоторых проблем, возникших в работе сети

### 1.1.3 Конфигурирование сетевого устройства (A1)

На рисунке 1.3 представлена диаграмма IDEF0 AS-IS, детализирующая описание бизнес-процесса «Конфигурирование сетевого устройства» (A1) на данный момент в компании.

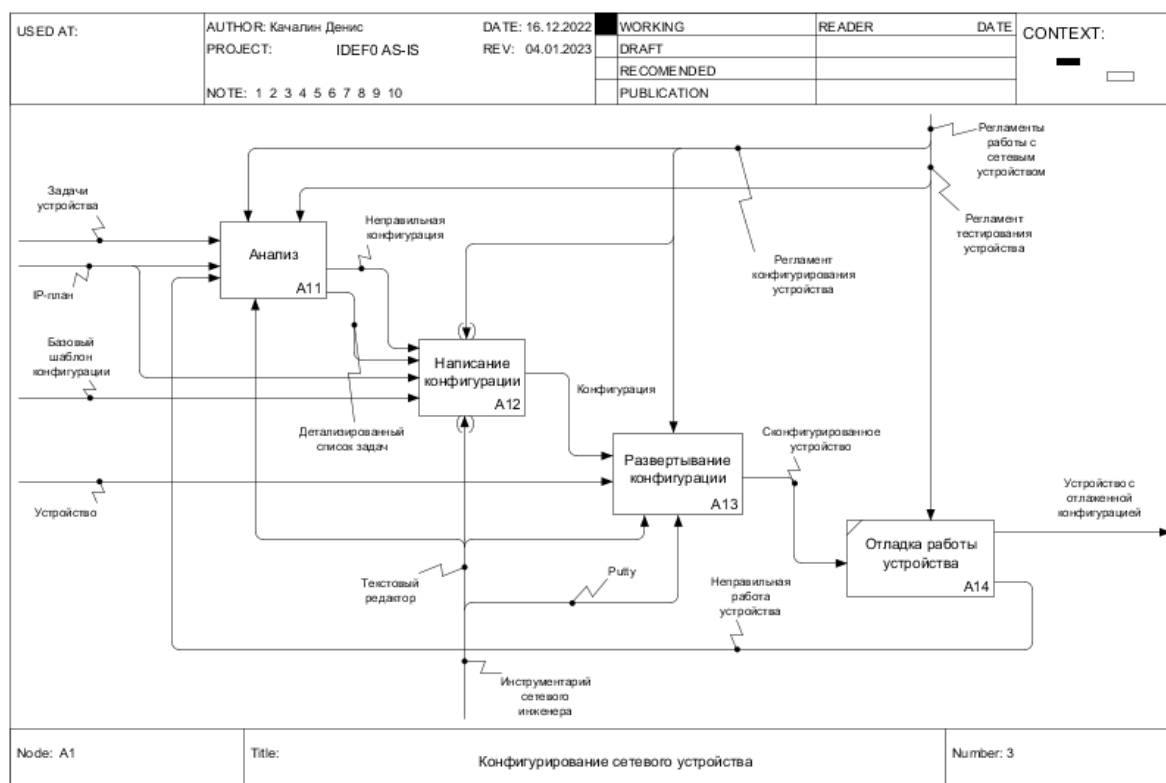


Рисунок 1.3 – Диаграмма IDEF0 AS-IS для бизнес-процесса «Конфигурирование сетевого устройства».

Чтобы корректно сконфигурировать устройство, сотруднику нужно иметь:

- задачи, которые устройство должно выполнять;
- IP-план, где (в какой подсети) устройство должно находиться;
- устройство, которое необходимо сконфигурировать;
- базовый шаблон конфигурации. Текстовый файл, в котором храниться универсальная для большинства устройств часть конфигурации.

При конфигурировании устройства сотрудник, руководствуясь регламентом конфигурирования и тестирования устройств, которые установлены в компании.

При помощи инструментария сетевого инженера выполняются все операции.

По окончании бизнес-процесса должен получить устройство с отлаженной конфигурацией.

При конфигурировании сетевого устройства в компании установлена следующая последовательность бизнес-процессов:

### **1 этап – Анализ (A11)**

Сетевой инженер при помощи текстового редактора (один из инструментов сетевого инженера):

- анализирует задачи устройства, IP-план и неправильную работу устройства (если устройство уже сконфигурировано, но не прошло процесс отладки);
- должен детализировать список задач устройства;
- выявить неправильную конфигурацию (если устройство не прошло отладку).

Анализ совершается в соответствии с регламентом конфигурирования устройства (один из регламентов работы с сетевыми устройствами).

### **2 этап – Написание конфигурации (A12)**

Чтобы написать конфигурацию, сетевому инженеру необходимо иметь: базовый шаблон конфигурации, детализированный список задач, IP-план и неправильную конфигурацию (если устройство не прошло отладку).

Написание конфигурации совершается в соответствии с регламентом конфигурирования устройства (один из регламентов работы с сетевыми устройствами) в текстовом редакторе.

В результате должна получиться конфигурация устройства.

### **3 этап – Развертывание конфигурации (A13)**

Получив конфигурацию и устройство, сотрудник может начать процесс конфигурирования устройства.

Для удаленного подключения к устройству необходимо ПО Putty и текстовый редактор для использования написанной конфигурации.

Процесс совершается согласно регламенту конфигурирования устройств (один из регламентов работы с сетевыми устройствами).

В результате устройство должно быть сконфигурировано.

#### 4 этап – Отладить работу устройства (A14)

Сконфигурированное устройство нужно отладить в соответствии с регламентом тестирования устройств (один из регламентов работы с сетевыми устройствами).

Если устройство прошло отладку оно переходит к следующему процессу.

Если устройство не прошло отладку, то сотруднику требуется разобраться в проблеме, т.е. перейти к анализу ситуации вновь, но уже для анализа неправильной работы устройства.

##### 1.1.4 Анализ (A11)

На рисунке 1.4 представлена диаграмма IDEF0 AS-IS, детализирующая описание бизнес-процесса «Анализ» (A11) на данный момент в компании.

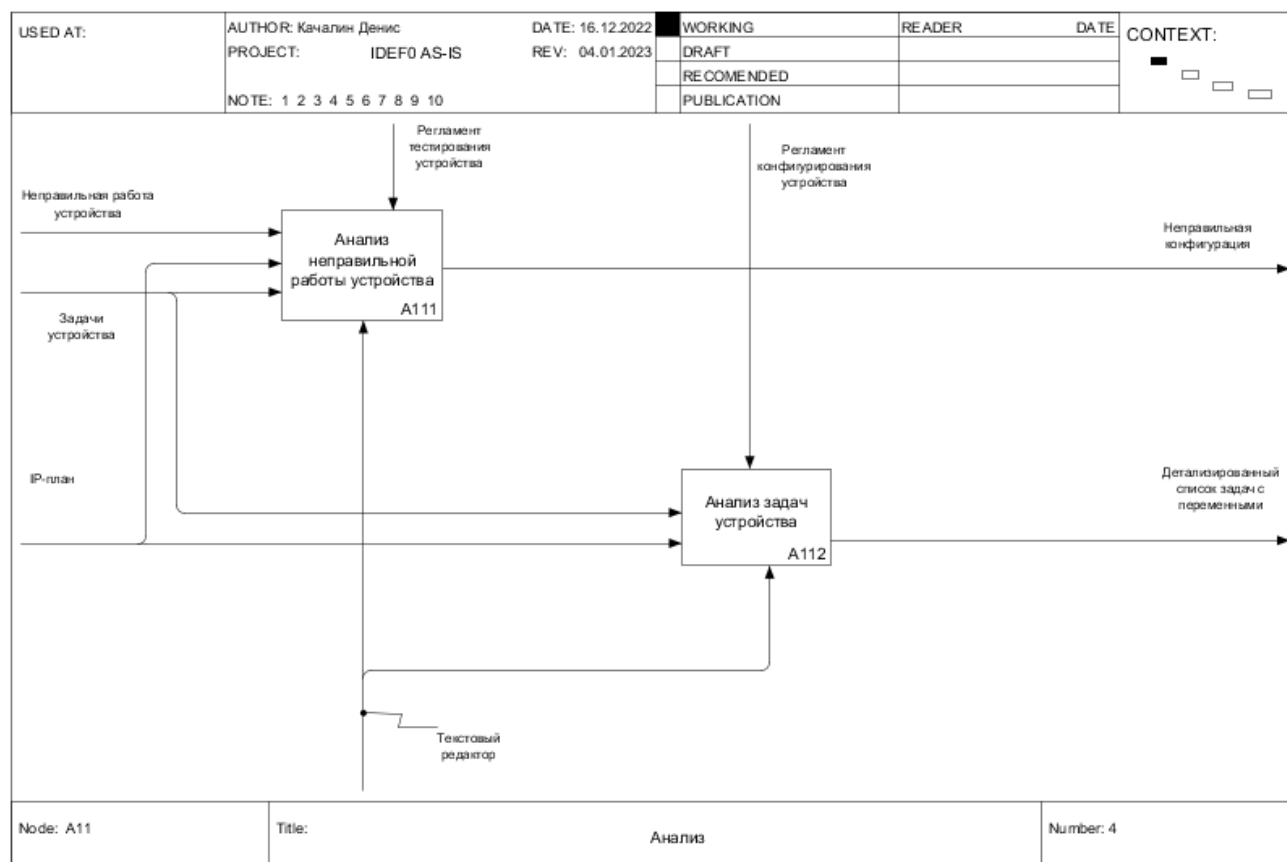


Рисунок 1.4 – Диаграмма IDEF0 AS-IS для бизнес-процесса «Анализ».

Под анализом подразумевается либо анализ, выявляющий почему устройство работает неправильно, либо анализ задач устройства.

### 1.1.5 Анализ неправильной работы устройства (A111)

На рисунке 1.5 представлена диаграмма IDEF0 AS-IS, детализирующая описание бизнес-процесса «Анализ неправильной работы устройства» (A111) на данный момент в компании.

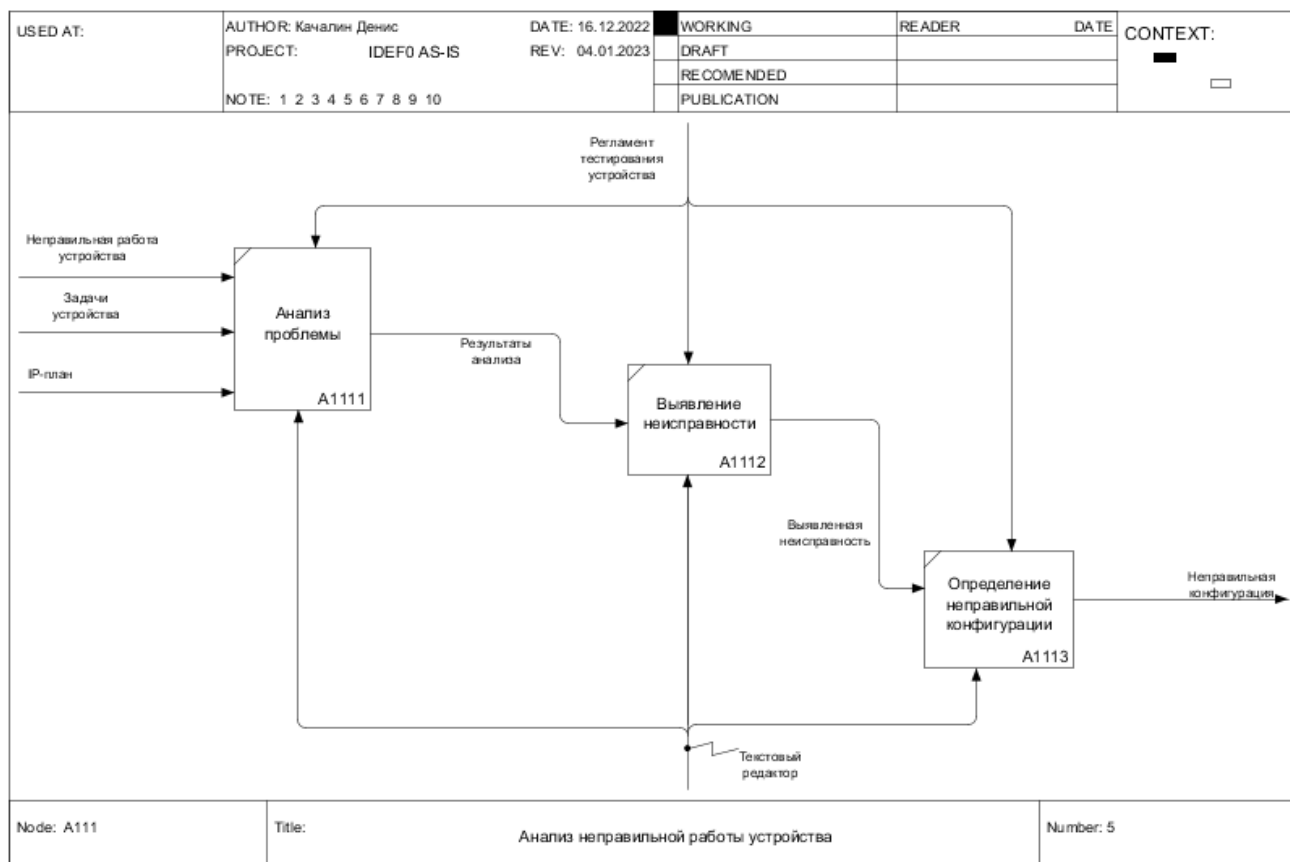


Рисунок 1.5 – Диаграмма IDEF0 AS-IS для бизнес-процесса «Анализ неправильной работы устройства».

Данный процесс возникает, когда в работе устройства была обнаружена ошибки или устройство не полной мере выполняет свои задачи.

Сотрудник должен по регламенту тестирования устройств выявить причину различия между требуемой и текущей функциональностью.

Помимо задач нужно учитывать и IP-план, т.к. ошибка в адресах часто вызывает некорректную работу устройства.

Причину различия, неправильную конфигурацию следует отправить в следующий процесс (исправление конфигурации).



### 1.1.6 Анализ задач устройства (A112)

На рисунке 1.6 представлена диаграмма IDEF0 AS-IS, детализирующая описание бизнес-процесса «Анализ задач устройства» (A112) на данный момент в компании.

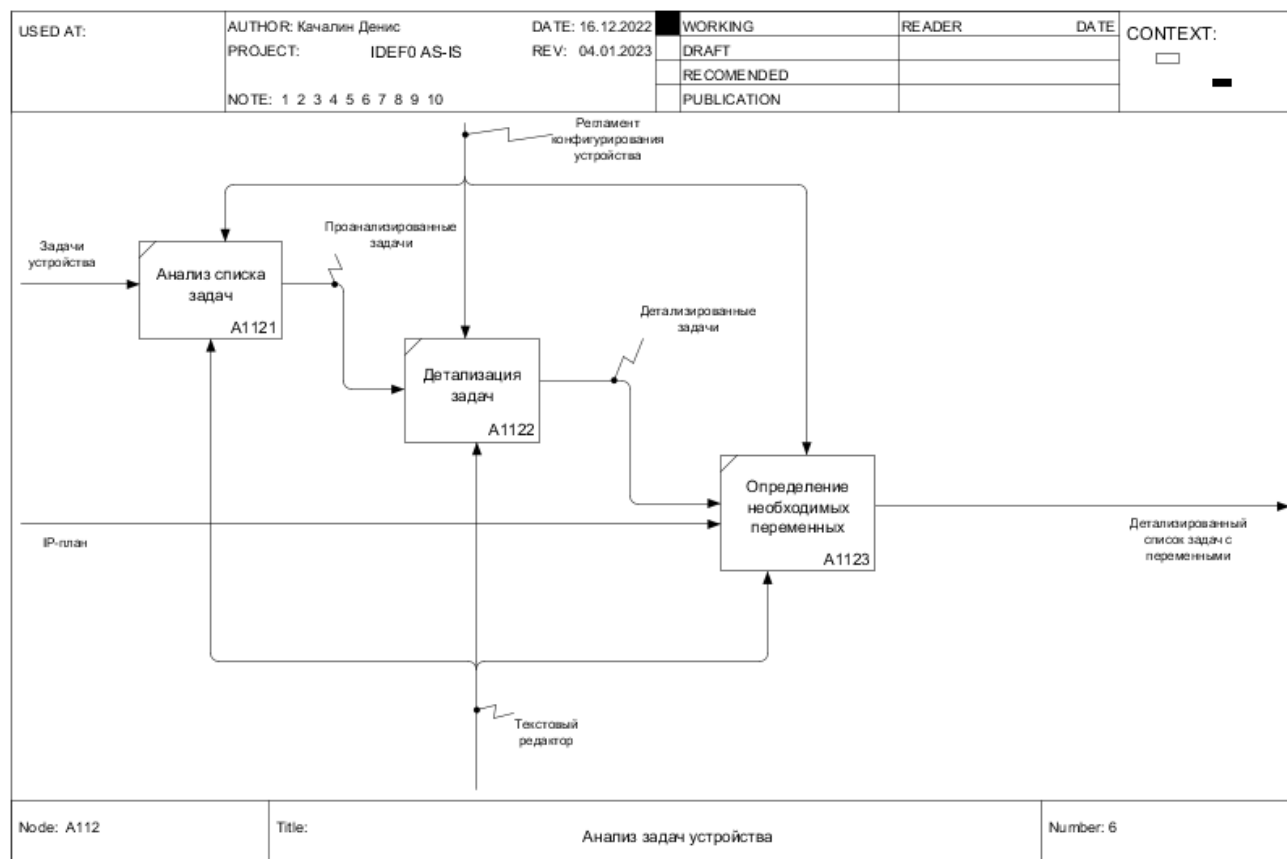


Рисунок 1.6 – Диаграмма IDEF0 AS-IS для бизнес-процесса «Анализ задач устройства».

Бизнес-процесс возникает при первой конфигурации устройства, т.е. с нуля, или когда следует значительно изменить функциональность устройства.

Сотрудник согласно регламенту конфигурирования, должен проанализировать список задач.

После этого, детализировать каждую задачу, чтобы упростить последующую конфигурацию.

Далее, нужно определить переменные исходя из детализированных задач и IP-плана.

По итогу должен получиться детализированный список задач с переменными.

### 1.1.7 Написание конфигурации (A12)

На рисунке 1.7 представлена диаграмма IDEF0 AS-IS, детализирующая описание бизнес-процесса «Написание конфигурации» (A12) на данный момент в компании.

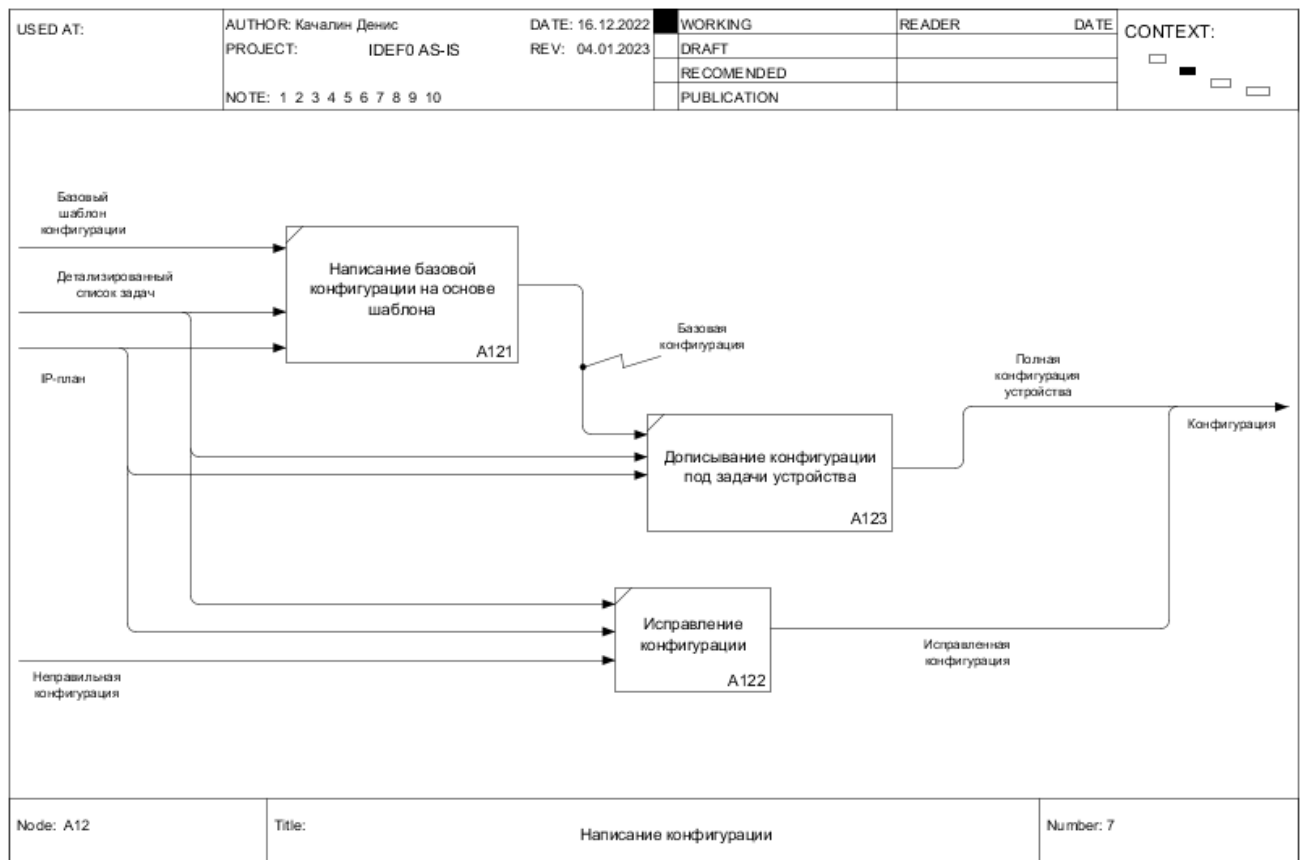


Рисунок 1.7 – Диаграмма IDEF0 AS-IS для бизнес-процесса «Написание конфигурации».

Бизнес-процесс «Написание конфигурации» можно разделить на две основных задачи: написание новой конфигурации и изменение имеющейся конфигурации. В написание новой конфигурации входят процессы «Написание базовой конфигурации на основе шаблона» и «Дописывание конфигурации под задачи устройства». В изменение имеющейся конфигурации – процесс «Исправление конфигурации».

#### **1.1.7.1 Написание базовой конфигурации на основе шаблона (A121)**

В первую очередь, сотруднику требуется написать базовую конфигурацию, используя базовый шаблон конфигурации, детализированный список задач и IP-план.

Процесс выполняется согласно регламенту конфигурирования.

Процесс выполняется в текстовом редакторе.

По итогу должна получиться базовая конфигурация устройства.

#### **1.1.7.2 Исправление конфигурации (A122)**

Когда сотрудник выявил неправильную конфигурацию устройства, её необходимо исправить.

Для этого требуется обратиться к списку задач и IP-плану.

Процесс выполняется в текстовом редакторе и согласно регламенту конфигурирования. В итоге должна получиться исправленная конфигурация.

#### **1.1.7.3 Дописывание конфигурации под задачи устройства (A123)**

Далее требуется дописать базовую конфигурацию до полной.

Для этого сотруднику необходимы детализированный список задач, IP-план и базовая конфигурация, написанная ранее.

Процесс выполняется, согласно регламенту конфигурирования.

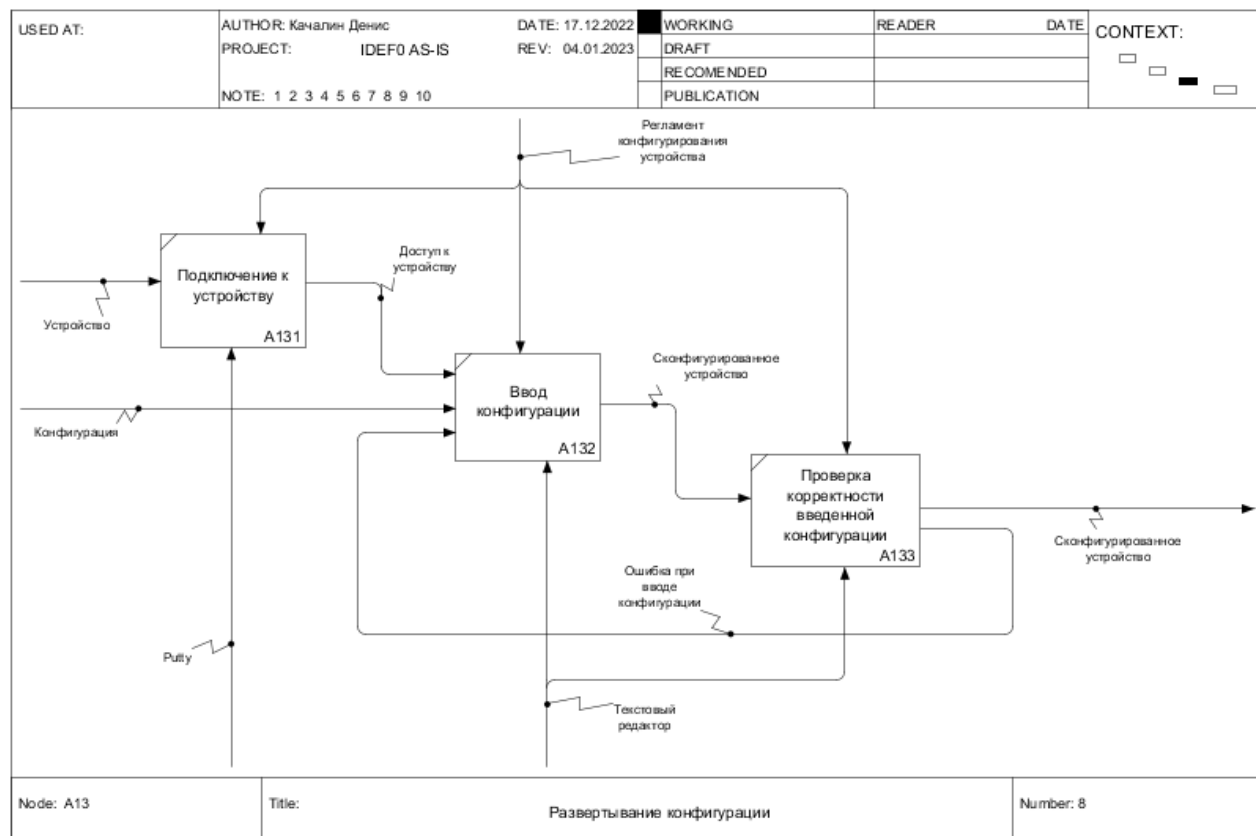
Процесс выполняется в текстовом редакторе.

По итогу должна получиться полная конфигурация устройства.

Полная конфигурация устройства и исправленная конфигурация для дальнейших процессов обобщается в «Конфигурация».

#### **1.1.8 Развертывание конфигурации (A13)**

На рисунке 1.8 представлена диаграмма IDEF0 AS-IS, детализирующая описание бизнес-процесса «Развертывание конфигурации» (A13) на данный момент в компании.



**Рисунок 1.8 – Диаграмма IDEF0 AS-IS для бизнес-процесса «Развертывание конфигурации».**

Под развертыванием конфигурации подразумевается перенос ранее подготовленной конфигурации в текстовом файле на само устройство.

Для этого сотруднику требуется сначала подключиться к устройству. Данная операция выполняется обычно в ПО Putty или в его аналогах. Порядок действий регулирует регламент конфигурирования.

Когда доступ к устройству будет получен начинается ввод конфигурации, написанной ранее. Обычно её сохраняют во временном текстовом файле, поэтому необходимо использование текстового редактора или программ для чтения текстовых файлов.

Когда конфигурация будет полностью перенесена на устройство, сотрудник должен вручную проверить всю конфигурацию и убедиться в её корректности.

Если он выявляет ошибку, то начинает её исправлять, т.е. вводить часть конфигурации заново.

Устройство, в конфигурации которого не было обнаружено ошибок отправляется в следующий процесс «Отладка работы устройства».

### 1.1.9 Документирование сетевого устройства (A2)

На рисунке 1.9 представлена диаграмма IDEF0 AS-IS, детализирующая описание бизнес-процесса «Документирование устройства» (A2) на данный момент в компании.

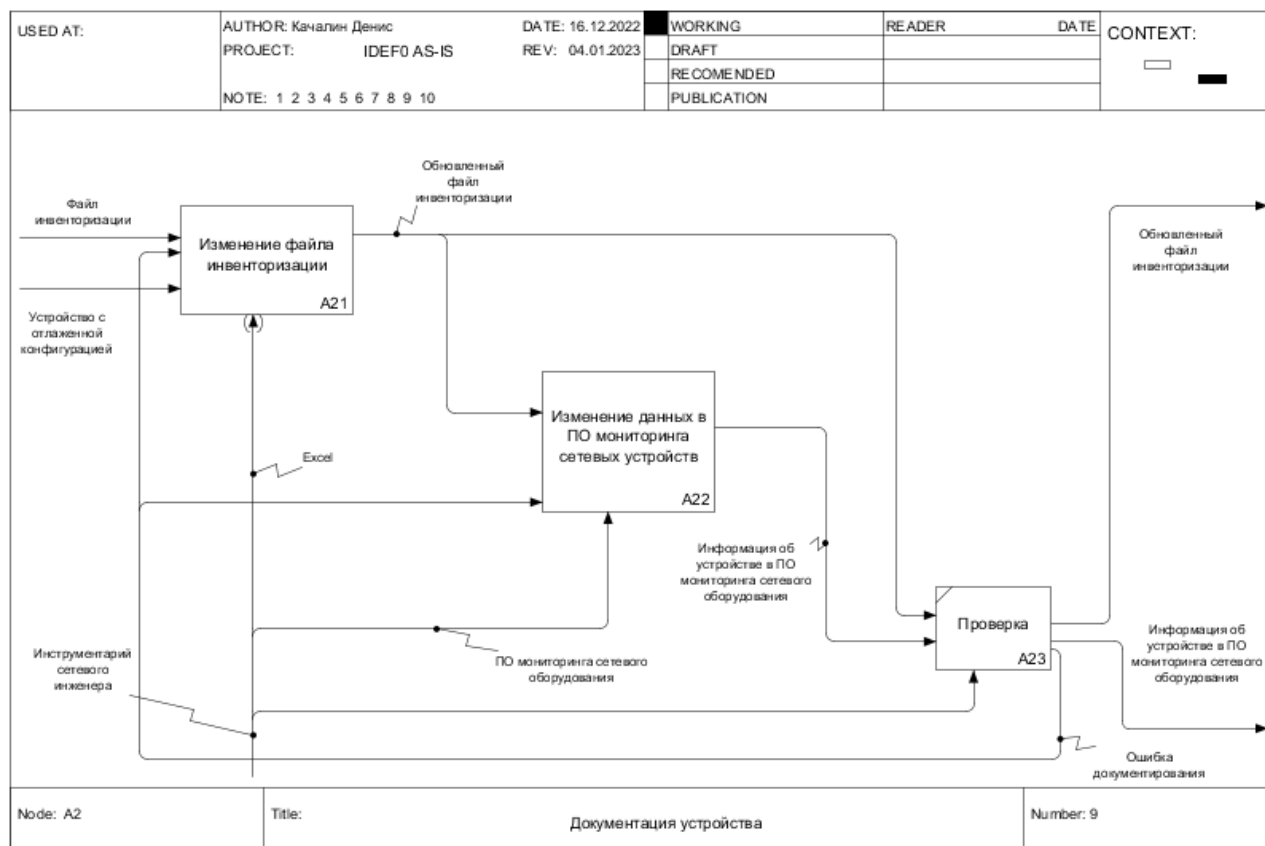


Рисунок 1.9 – Диаграмма IDEF0 AS-IS для бизнес-процесса «Документация устройства».

Чтобы задокументировать устройства сотруднику нужно иметь:

- устройство с отлаженной документацией;
- файл инвентаризации сетевого оборудования.

Документирование устройства совершается в соответствии с регламентом документирования устройств.

Для выполнения операции сотрудник использует инструментарий сетевого инженера.

По окончании бизнес-процесса должны быть получены:

- файл инвентаризации с обновленной информацией;
- информацию об устройстве в различном ПО для мониторинга работы сетевого оборудования.

Документирование устройства состоит из трех этапов:

1. Внесение информации об устройстве в файл инвентаризации;
2. Внесение информации об устройстве в ПО для мониторинга работы сетевых устройств;
3. Проверка внесенной информации на корректность.

### 1.1.10 Изменение файла инвентаризации (A21)

На рисунке 1.10 представлена диаграмма IDEF0 AS-IS, детализирующая описание бизнес-процесса «Изменение файла инвентаризации» (A21) на данный момент в компании.

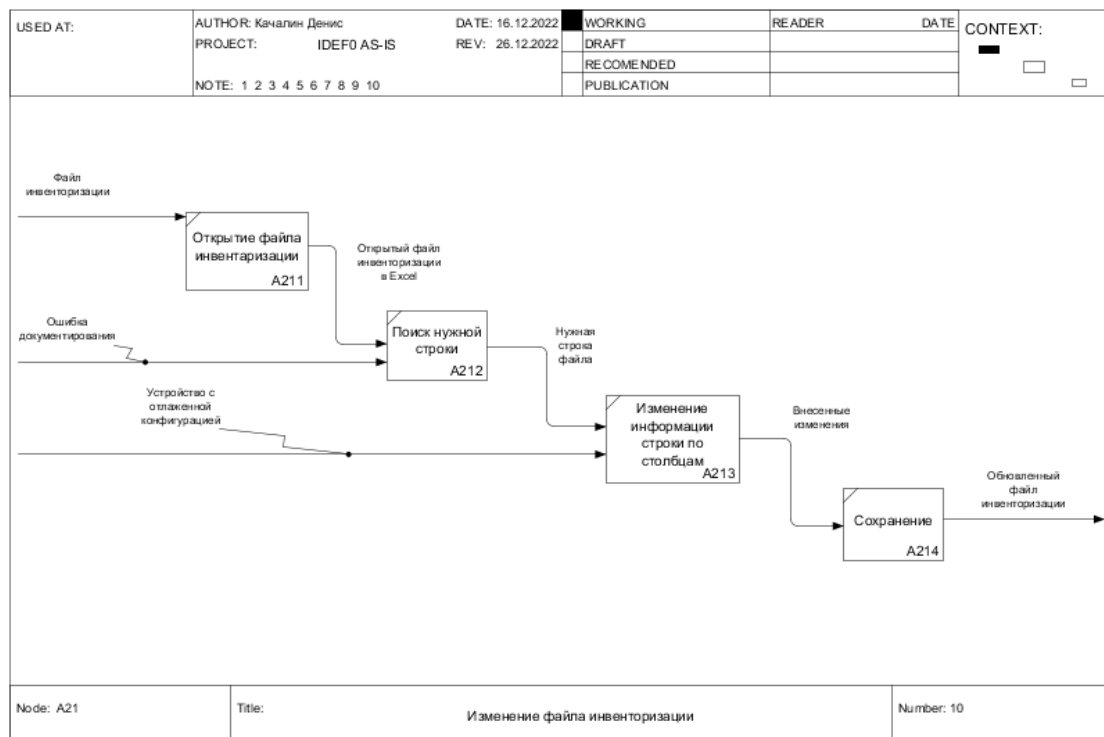


Рисунок 1.10 – Диаграмма IDEF0 AS-IS для бизнес-процесса «Изменение файла инвентаризации».

Согласно регламенту документирования устройств, сотруднику требуется внести в файл инвентаризации информацию.

Информация берется с устройства, которое ранее было сконфигурировано.

В файле храниться информация в табличном виде, поэтому для работы с ней требуется использование ПО Microsoft Excel.

Файл нужно открыть и найти в нем строку, в которую вносится информация. В строке внести изменения. Файл сохранить. В итоге обновлённый файл инвентаризации используется в следующих процессах.

### 1.1.11 Изменение данных в ПО для мониторинга работы сетевых устройств (A22)

На рисунке 1.11 представлена диаграмма IDEF0 AS-IS, детализирующая описание бизнес-процесса «Изменение данных в ПО для мониторинга работы сетевых устройств» (A22) на данный момент в компании.

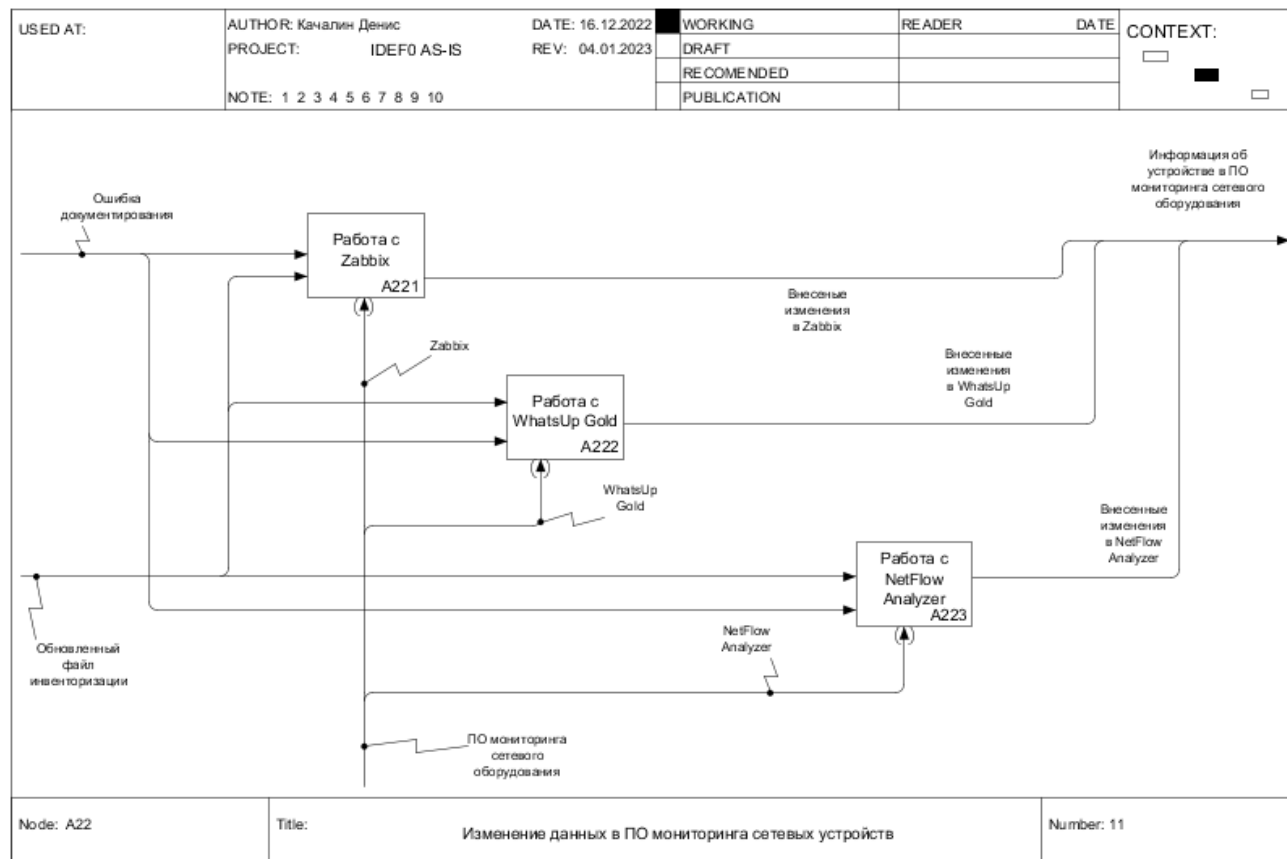


Рисунок 1.11 – Диаграмма IDEF0 AS-IS для бизнес-процесса «Изменение данных в ПО мониторинга сетевых устройств».

Согласно регламенту документирования сотрудникам нужно обеспечить мониторинг важного для сетевой инфраструктуры оборудования. Для этого используется специальное ПО для мониторинга.

На данный момент в сетевом отделе используется три ПО для мониторинга сетевых устройств:

- Zabbix – для мониторинга нагрузки на каналы связи и документирование сети в виде схем [4];
- WhatsUp Gold – для мониторинга состояния устройства и документирования сети [5];
- NetFlow Analyzer – для мониторинга трафика в сети [6].

Эти ПО работают в WEB приложениях и принцип работы с ними схож: чтобы внести устройство в ПО, нужно создать форму устройства, открыть её, заполнить информацией и сохранить.

Если требуется изменить информацию или исправить ошибку в документировании форму заново не требуется, достаточно найти соответствующую оборудованию.

В итоге информация об устройстве должна храниться в ПО мониторинга сетевого оборудования.

С бизнес-процессами при работе с различным ПО можно ознакомиться на рисунках 1.12-1.14.

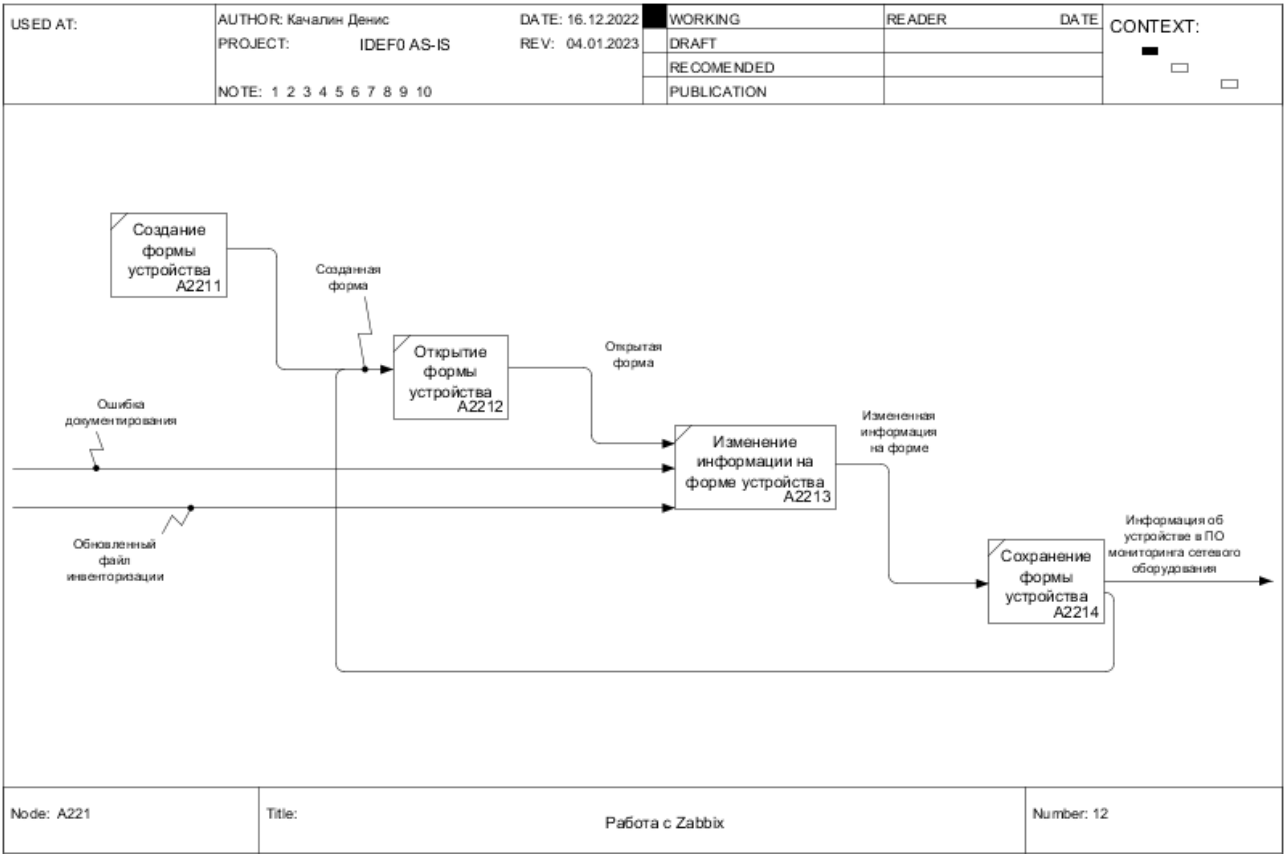


Рисунок 1.12 – Диаграмма IDEF0 AS-IS для бизнес-процесса «Работа с Zabbix».



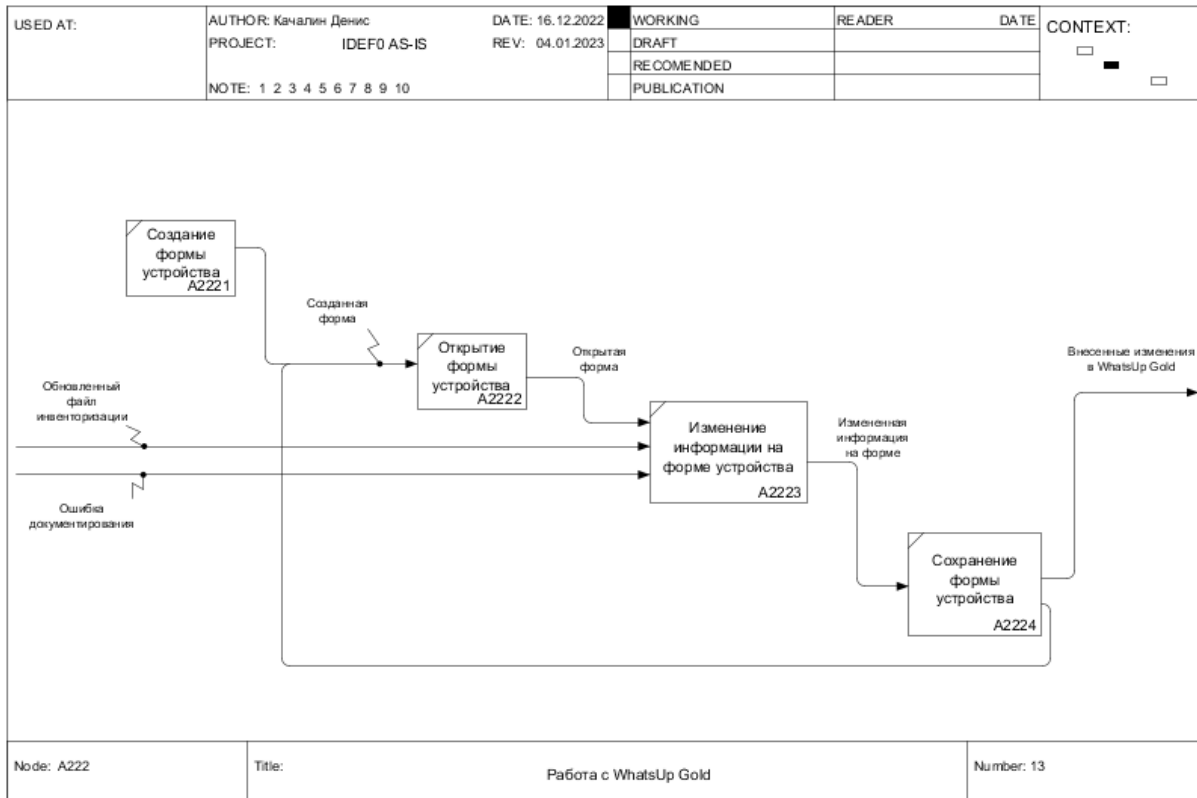


Рисунок 1.13 – Диаграмма IDEF0 AS-IS для бизнес-процесса «Работа с WhatsUp Gold».

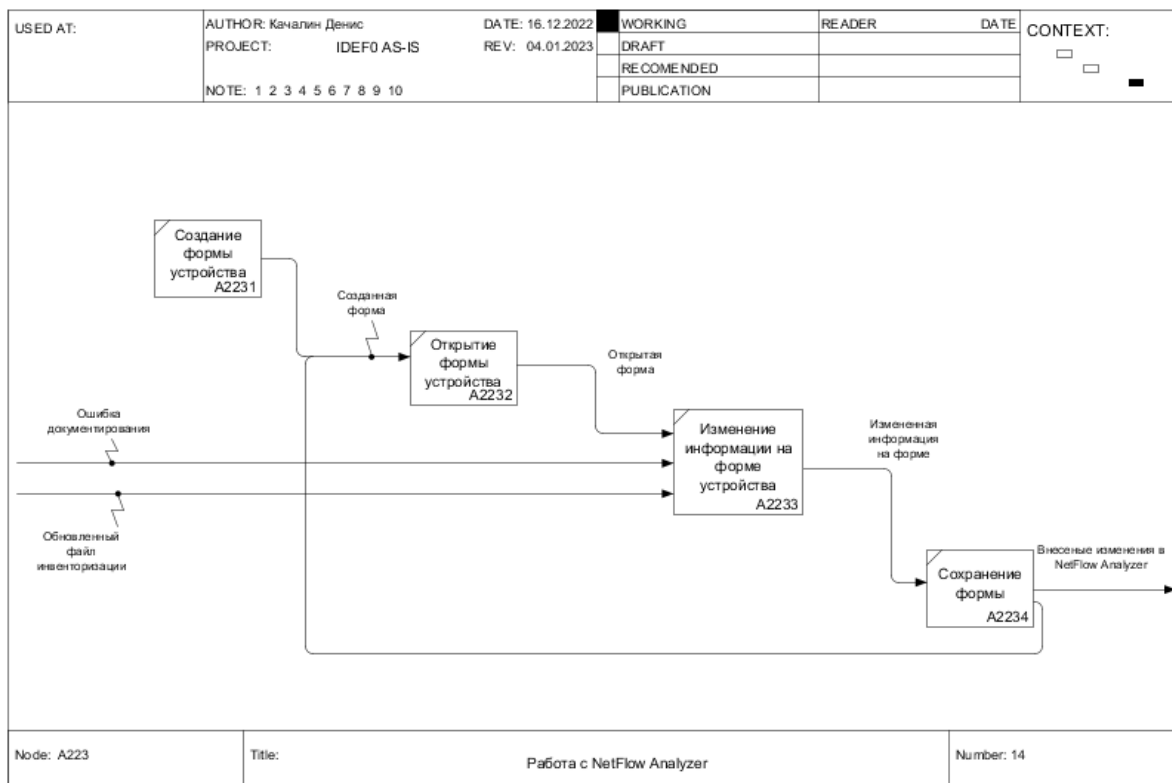


Рисунок 1.14 – Диаграмма IDEF0 AS-IS для бизнес-процесса «Работа с NetFlow Analyzer».

### 1.1.12 Проверка внесенной информации на корректность (A23)

Согласно регламенту документирования, сотруднику требуется проверить правильность внесенной информации в файл инвентаризации и ПО мониторинга.

Если проверка не выявила ошибок в документировании в дальнейшем сотрудники используют обновленный файл инвентаризации и информацию из ПО мониторинга.

Если ошибки были обнаружены, то требуется внести исправления.

## **1.2 Назначение и цели создания системы**

Исходя из анализа существующих бизнес-процессов компании (п. 1.1), было выявлено, что для повышения эффективности работы компании в целом и сетевого отдела в частности, следует внедрить автоматическую систему документирования и конфигурирования сетевого оборудования.

### **1.2.1 Назначение системы**

Система предназначена для автоматизации конфигурирования, мониторинга, поддержания работы сетевых устройств в корпоративной вычислительной информационной сети.

### **1.2.2 Цель системы**

Уменьшить время, проводимое сотрудником сетевого отдела ООО "ЭН+ ДИДЖИТАЛ" на решение рутинных задач, связанных с обслуживанием сетевых устройств за счет введения автоматизированной системы конфигурирования и документирования сетевых устройств корпоративной вычислительной информационной сети.

## **1.3 Требования к системе**

### **1.3.1 Требованиям к функциональности системы**

Исходя из описания существующей ситуации в бизнес-процессах сетевого отдела компании EN+ DIGITAL при работе с сетевым оборудованием можно выделить следующий набор требований к ПО:

- автоматизация конфигурирования сетевых устройств;
- автоматизация документирования сетевых устройств;
- автоматизация документирования сети;
- конфигурирование множества сетевых устройств одновременно;
- работа с группами устройств;
- автоматизация сбора информации с устройств;
- добавление, обновление и удаление информации об устройствах в подсистему инвентаризации (модель, производитель, интерфейсы и прочее);
- добавление, обновление и удаление информации о компании в подсистему инвентаризации (офисы, географическое расположение и прочее);
- добавление, обновление и удаление информации о сети в подсистему инвентаризации (ip-диапазоны, ip-адреса, VLAN'ы и прочее);
- использование информации из подсистемы инвентаризации в задачах автоматизации;
- отладка новых конфигураций на специально выделенном полигоне;
- добавление, обновление и удаление информации из других подсистем документирования сети (Zabbix, NetFlow Analyzer, WhatsUp Gold).

### **1.3.2 Требования к видам обеспечения к системе**

При планировании разработки системы следует предъявлять требования к системе с позиции различного программного обеспечения. Далее приводятся требования с позиции информационного, программного, технического и организационного обеспечения.

#### **1.3.2.1 Информационное обеспечение**

ПО Ansible должен иметь доступ к полигону (если он находится на другой ЭВМ).

ПО Ansible должно иметь доступ к ПО NetBox (если он находится на другой ЭВМ).

Система должна иметь доступ к интернету для скачивания обновлений

Регулирование способа подключения и использования системы отдаются на предпочтение сотрудникам компании.

#### **1.3.2.2 Программное обеспечение**

Для функционирования системы требуются следующий набор ПО (версии должны быть не ниже приведенных):

- Ubuntu (v22.04);
- Python (3.10.8);
- Ansible (6.5.0);
- Netbox (3.3.5);
- PostgreSQL (14.5);
- Redis (4.0);
- GNS3 (2.2.34);
- GNS3 VM for VirtualBox (2.2.34);
- Oracle VM VirtualBox Manager (6.1.38).

#### **1.3.2.3 Техническое обеспечение**

Для нормального функционирования системы требуется выделить ЭВМ с характеристиками, не уступающих представленным в таблице 1.2.

Таблица 1.2 – Требуемое аппаратное обеспечение

Компонент	Экземпляр
Процессор	AMD Ryzen 5 5600H, 3.30 GHz
Оперативная память	DDR4 16GB
Дисковое пространство	SSD 256GB

#### 1.3.2.4 Организационное обеспечение

Данная система будет использоваться в отделах серверной и сетевой инфраструктуры компании EN+ DIGITAL

Сопровождение системы должен осуществлять отдел серверной инфраструктуры в совместной работе с сетевым отделом (консультации, совместные работы и прочее).

Сопровождающий персонал должен быть ознакомлен с принципом работы системы и её особенностями перед непосредственной работой. Обучение должно проводиться разбирающимся в этой системе специалисте в соответствии с документацией о системе.

Регулирование работы пользователей с системой должно составляться начальниками отделов сетевой и серверной инфраструктуры, информационной безопасности.

Взаимодействие системы с внешними и внутренними системами компании регулируется и настраивается сотрудниками отдела серверной и сетевой инфраструктуры, информационной безопасности.

#### 1.3.3 Общие технические требования

##### 1.3.3.1 Требования к численности и квалификации персонала и пользователей системы

Для нормальной работы системы требуется наличие не менее 3-х, обученных к работе с системой, сотрудников отдела сетевой и серверной инфраструктуры. При возникновении ситуаций, связанных с системой, хотя бы 1 сотрудник должен выйти на рабочую смену.

Для администрирования системы требуется не менее 2-3 человек серверной инфраструктуре, которые разбираются с работой стека технологий системы.

Персонал, работающий с системой, должен совершать только те действия, которые решают проблемы, оговоренные в его рабочем наряде.

Под отказом системы понимается:

- внеплановое выключение системы (потеря питания, ошибка системы, неисправность компонентов системы и прочее);

- внеплановая потеря доступа к сети, автоматизацию который осуществляет система.

Персонал, работающий с системой автоматизации, должен быть квалифицирован в работе с:

– ПО Ansible:

- конфигурационные файлы Ansible;
- синтаксис Ansible;
- отладка playbook;
- инвентарные файлы;
- плагины;
- файловыми разметками YAML и INI;

– OS Linux:

- текстовые редакторы (VIM или NANO);
- базовое понимание работы Linux систем;
- терминал Linux;

– GNS3 (или любой другой сетевой полигон);

– гипервизор (WMWARE, VIRTUAL BOX, HyperV и другие);

– ПО NetBox:

- WEB GUI;
- API.

Персонал, администрирующий систему автоматизации, должен быть квалифицирован в работе с:

– OS Linux:

- текстовые редакторы (VIM или NANO);
- глубокое понимание работы Linux систем;
- терминал Linux;
- работа с сервисами (web, firewall и другие);

– GNS3 (или любой другой сетевой полигон);

– гипервизор (WMWARE, VIRTUAL BOX, HyperV и другие).

### **1.3.3.2 Требования к показателям назначения**

В случае нарушения в системе электропитания функционирование системы отдается на предпочтение сотрудникам компании. Рекомендуется организовать подключение ЭВМ системы к сети через источник бесперебойного питания.

Корректная работа системы при одновременном обращении к ресурсам нескольких пользователей отдается на предпочтение сотрудникам компании. Рекомендуется организовать обработку таких ситуаций.

При подключении новых пользователей система должна работать в нормальном режиме.

Система должна обеспечивать несколько сетевых устройств одновременно. Не менее 3 устройств одновременно.

Обслуживание устройств при помощи системы, должно значительно сокращать время, требуемое для решения задачи. К примеру, задачу, которую сотрудник решает за 30 минут, при помощи системы должна выполняться в течении 5-15 минут.

Поиск и решение неисправностей в системе автоматизации не должен занимать более 6 часов для документированной ситуации. И не более 24 часов – для уникальной ситуации.

#### **1.3.3.3 Требования к надежности**

При обнаружении ошибки или неточности в работе системы, персоналу требуется обратиться к администраторам системы с подробным отчётом.

Частота резервного копирования системы отдается на предпочтение сотрудникам компании. Рекомендуется совершать резервное копирование системы раз в сутки.

#### **1.3.3.4 Требования по безопасности**

Техника безопасности внедрения и эксплуатации отдается на предпочтение сотрудникам компании. Рекомендуется устанавливать ЭВМ системы в серверную комнату.

#### **1.3.3.5 Требования к эргономике и технической эстетике**

Условия труда в помещениях персонала, работающего с системой, отдается на предпочтение сотрудникам компании.

#### **1.3.3.6 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы**

При эксплуатации системы персонал может:

- находится в офисе компании;
- у себя дома (удаленный рабочий стол) с выполнением техники безопасности эксплуатации систем компании.

Рекомендуется выполнять быстрое обслуживание системы раз в сутки, тщательное обслуживание – раз в неделю, в нерабочее время.

#### **1.3.3.7 Требования к защите информации от несанкционированного доступа**

Безопасность информации системы отдается на предпочтение сотрудникам компании (отделу информационной безопасности) в соответствии с внутренними регламентами о защите информации.

Рекомендуется организовать не менее 2 учетных записей системы:

- администратор;
- пользователь.

Для каждой записи стоит использовать сложный пароль с периодической его заменой.

Доступ к учетным записям должен осуществляться только сотрудникам, имеющим соответствующие права и обязанности.

#### **1.3.3.8 Требования к защите от влияния внешних воздействий**

ЭВМ с системой должна находиться в серверной комнате. Расположение ЭВМ рекомендуется организовать в стойке для минимизации воздействия внешних факторов.

#### **1.3.3.9 Требования к патентной чистоте и патентоспособности**

ПО, используемое для создания системы, является бесплатным и свободно распространяемым в сети интернет. Рекомендуется использовать версии с официальных сайтов разработчиков для сохранения авторских прав.

### **1.3.4 Дополнительные требования**

#### **1.3.4.1 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие**

К моменту сдачи подсистемы в опытную эксплуатацию должно быть:

- предоставлено техническое обеспечение, соответствующее требованиям;
- установлено системное программное обеспечение, соответствующее требованиям;
- установлено программное обеспечение, разработанное в соответствии с требованиями.

#### **1.3.4.2 Требования к документированию**

Настоящее Техническое задание на подсистему должно разрабатываться в соответствии с ГОСТ 34.602-89 (автоматизированные системы).

Для разрабатываемого программного обеспечения проектная и эксплуатационная документация должна разрабатываться с учетом требований ГОСТ 34.201-89 и РД 50-34.698-90.

Перечень разрабатываемой и представляемой документации включает:

- техническое задание на создание системы;
- документация Технического Проекта Системы в составе:
  - Пояснительная записка;
  - Руководство администратора.

Документация, разработанная Исполнителем, представляется в печатном виде.

Система, разработанная Исполнителем, представляется в виде образа виртуальной машины на носителе или через репозиторий.

## **1.4 Обзор систем автоматизации конфигурирования и документирования сетевой инфраструктуры**

Выбор ПО для решения конкретных задач всегда сопровождается анализом и сравнением аналогов. Далее приводится анализ возможностей конкурентного ПО для автоматизации конфигурирования и документирования сетевых устройств на основании [7-9].

### **1.4.1 NAPALM**

Network Automation and Programmability Abstraction Layer with Multivendor support – это библиотека для языка программирования Python, которая позволяет выполнять взаимодействие с различными ОС сетевого оборудования, используя объединённый API [10].

Другими словами, мы пишем программы на Python, которые взаимодействуют с различными сетевыми устройствами. Библиотека же позволяет упростить многие задачи. Такие программы часто называют скриптами. В программе должна быть четко прописана логика работы, иначе программа может работать некорректно.

Благодаря своей доступности и хорошей документации обрела значительное комьюнити.

### **1.4.2 NETMIKO**

Аналогично NAPALM – это библиотека для Python, которая позволяет упростить взаимодействие с сетевым оборудованием [11]. Главным отличием от NAPALM является, то, что NETMIKO взаимодействует через CLI (Console Line Interface), что увеличивает число поддерживаемого ПО. Но нужно и учитывать и тот факт, что на оборудовании от разных производителей некоторые принципы работы CLI могут отличаться. Так как это может породить некорректную работу программы или устройства.

Распространяется бесплатным образом. А наличие хорошей документации создало довольно большое комьюнити, которое продолжает развивать библиотеку.

### **1.4.3 Ansible**

Это ПО с открытым исходным кодом, которое автоматизирует поставку программного обеспечения, управление конфигурацией и развёртывание приложений [12]. Ansible помогает DevOps-специалистам автоматизировать сложные задачи.



Главными преимуществами Ansible являются:

- не требует установки дополнительного ПО на устройствах, на которых производится автоматизация;
- не важно сколько раз будет выполняться операция, результат будет всегда одинаковым;
- множество дополнительных модулей позволяет расширять функциональность ПО;
- в основе Ansible лежат Python и YAML, которые считаются довольно простыми для освоения.

#### **1.4.4 Ansible Tower**

Целая платформа автоматизации сети, в ядре которой работает Ansible. В отличие от обычного Ansible имеет встроенный WEB GUI и REST API. Наличие отличной документации и поддержки со стороны разработчиков, позволяет решать большинство возникающих проблем с системой [13].

Имеет различные инструменты для мониторинга устройств и предоставления информации в виде графиков и схем. А система инвентаризации уже встроена в платформу. Права пользователей можно делегировать, а их действия документируются в автоматическом режиме. Помимо этого, имеет бесплатную пробную версию, доступную все желающим.

Из недостатков можно отметить это цену, о которой нужно договариваться с разработчиком, а также отсутствие полноценной поддержки сетевого оборудования. Для создания автоматизации множества задач, нужно в программах автоматизации учитывать логику работы сетевых устройств.

#### **1.4.5 NetYCE**

Как и Ansible Tower – является платной платформой автоматизации, которая работает из коробки. Но существенным отличием от предыдущей – это полная ориентированность на работу с сетевыми устройствами. Поэтому работа с сетевым оборудованием проходит быстрее и проще [14]. В системе встроен целый набор инструментов для обслуживания сетевых устройств:

- инвентаризация;
- распределение IP-пространства;
- визуализация информации (схемы, графики);
- автоматизация конфигурирования.

Система также обладает GUI, API, распределением ролей пользователей, документированием их действий.

Но из-за платной политики распределения комьюнити платформы нельзя назвать большим.

#### **1.4.6 Общее сравнение**

В таблице 1.3 приведено общее сравнение аналогов исходя из ранее проведенного обзора.

Таблица 1.3 – Общее сравнение аналогов

Название	Приоритет	Плюсы	Минусы
NetYCE	1	<ul style="list-style-type: none"> <li>– Ориентированность на работу с сетевым оборудованием;</li> <li>– Все инструменты в одном месте;</li> <li>– Поддержка производителя;</li> <li>– GUI;</li> <li>– Документирование действий пользователей;</li> <li>– Распределение ролей пользователей;</li> <li>– API;</li> <li>– Подробная документация;</li> <li>– Визуализация информации.</li> </ul>	<ul style="list-style-type: none"> <li>– Высокая цена;</li> <li>– Небольшое комьюнити.</li> </ul>
Ansible Tower	2	<ul style="list-style-type: none"> <li>– Тесная интеграция подсистем;</li> <li>– GUI для Ansible;</li> <li>– Распределение ролей пользователей;</li> <li>– Документирование действий пользователей;</li> <li>– Представление информации в виде графиков и схем служба поддержки;</li> <li>– REST API;</li> <li>– Подробная документация;</li> <li>– Поддержка производителя;</li> <li>– Бесплатная пробная версия.</li> </ul>	<ul style="list-style-type: none"> <li>– Высокая цена;</li> <li>– Нужно прописывать логику программ автоматизации для работы с сетевым оборудованием;</li> <li>– Требуется обучиться работе с Ansible.</li> </ul>
Ansible	3	<ul style="list-style-type: none"> <li>– Бесплатный;</li> <li>– Большое комьюнити;</li> <li>– Подробная документация;</li> <li>– Большое количество сторонних модулей, расширяющих функциональность.</li> </ul>	<ul style="list-style-type: none"> <li>– Нужно прописывать логику программ автоматизации для работы с сетевым оборудованием;</li> <li>– Необходимо использовать сторонне ПО и специализированные модули для получения нужной функциональности;</li> <li>– Требуется обучиться работе с Ansible.</li> </ul>

Продолжение таблицы 1.3

Название	Приоритет	Плюсы	Минусы
NETMIKO	4	<ul style="list-style-type: none"> <li>– Бесплатный;</li> <li>– Хорошая документация;</li> <li>– Комьюнити;</li> <li>– Поддерживает CLI.</li> </ul>	<ul style="list-style-type: none"> <li>– Требуется знание Python;</li> <li>– Нужно прописывать всю логику автоматизации.</li> </ul>
NAPALM	5	<ul style="list-style-type: none"> <li>– Бесплатный;</li> <li>– Хорошая документация;</li> <li>– Комьюнити.</li> </ul>	<ul style="list-style-type: none"> <li>– Требуется знание Python;</li> <li>– Нужно прописывать всю логику автоматизации.</li> </ul>

Исходя из данных таблицы 1.3 следует, что NetYCE является самым лучшим вариантом для работы с сетевыми устройствами. Наличие всех необходимых инструментов для работы с сетевым оборудованием, делают данную платформу самым привлекательным решением, но стоит учитывать, что данное ПО платное и за пользование придется периодически платить. Поэтому для прототипа системы оно не подходит.

Ansible и Ansible Tower универсальное решения большинства задач автоматизации (не только при работе с сетевым оборудованием), но из-за этой универсальности приходится прописывать логику многих действий. Ansible Tower стоит рассматривать, если в компании будет планироваться внедрение системы автоматизации также в отдел серверной инфраструктуры.

NETMIKO и NAPALM – являются лишь библиотекой для автоматизации самых распространенных задач. Для работы с ними также требуется знание языка Python, поэтому они заметно проигрывают Ansible, который имеет свой довольно упрощенный язык, направленный на задачи автоматизации (по сравнению с полноценными языками программирования).

В таблице 1.4 представлено сравнение аналогов по критериям.

Таблица 1.4 – Сравнение аналогов по критериям

Решение Критерии	NetYCE	Ansible Tower	Ansible + NetBox	NETMIKO	NAPALM
Ориентированность на работу с сетевыми устройствами	✓	✗	✓	✓	✗
Система инвентаризации	✓	✓	✓	✗	✗
Цена за месяц	1.000.000р	2.500.000р	0р	0р	0р
Документация	✓	✓	✓	✓	✓
Визуализация информации	✓	✓	✗	✗	✗
Распределение ролей	✓	✓	Частично	✗	✗
Документирование действий пользователей	✓	✓	Частично	✓	✓
Поддержка от производителя	✓	✓	✗	✗	✗
GUI	✓	✓	Частично	✗	✗
Мультивендорная поддержка	✓	Частично	Частично	✓	✗

Из таблицы 1.4 понятно, что использование Ansible и NetBox для задач автоматизации работы с сетевыми устройствами является самым оптимальным решением, т.к. решение является бесплатным, а функционал, который может понадобиться в будущем может быть расширен интеграцией с другими ПО посредством модулей. Модули пишутся как разработчиками, так и членами сообщества.

Если же мы выбираем между платными аналогами, то следует учитывать такой фактор, как «нужна ли система автоматизации в отделе серверной инфраструктуре». Если ответ отрицательный, то рекомендуется приобрести лицензию NetYCE. Иначе стоит задуматься над покупкой Ansible Tower.

Использование библиотек NETMIKO и NAPALM является самым неоптимальным вариантом, т.к. возможности библиотек сильно ограничены, хотя и не отрицается их использование в каких-то конкретных задачах или нуждах.

## 1.5 Описание выбранных средств реализации системы

В данном разделе приводится описание ПО, которое было выбрано для реализации системы автоматизации.

### 1.5.1 Ansible

Ansible – это программный продукт, с открытым исходным кодом. Ansible позволяет автоматизировать передачу, настройку и развертывание различных приложений и ПО [12].

Из ключевых особенностей можно выделить:

- Ansible не требует установки дополнительного ПО на удаленном оборудовании. Все функции либо выполняются на управляющем устройстве, либо на языке Python, который присутствует в большинстве систем;

- Не важно сколько раз будет выполняться какая-то задача, результат будет один и тот же. Ansible отслеживает, что повторное выполнение задачи не принесет нового результата и просто его пропускает;

- Так как Ansible написан на Python и использует YAML разметку, поэтому считается довольно простым в изучении.

Ansible состоит из множества различных элементов, выполняющие различные функции. Далее приводится их описание:

- модули. Программы, которые решают определенные задачи на удаленном устройстве;

- плагины. Другое их название – коллекции. Для их распространения существует сайт Ansible-Galaxy [15]. Коллекции могут быть написаны разработчиками ПО, сторонними разработчиками или пользователями;

- инвентарь конфигурируемых хостов. Инвентарём может служить файл или БД;

- playbooks. Программы автоматизации задач на удаленных хостах;

- переменные. Файлы с определенными переменные, которые позволяют гибко настроить работу с отдельным хостом или целой группой;

- роли. Способ группировки множества задач для автоматизации работы и упрощения структуры файлов Ansible.

### 1.5.2 NetBox

NetBox – web-приложение с открытым исходным кодом, которое позволяет управлять и документировать компьютерные сети [16].

NetBox позволяет:

- управлять IP-пространством (IP-адреса, VRF, VLAN);

- описать устройства (типы, расположение, производитель и прочее);

- описать подключения устройств (сеть, консоль);

- описать виртуализацию (виртуальные машины и кластеры);

- описать передачу данных (провайдеры, дальняя связь);

- описать способов шифрования.

На рисунке 1.15 представлен стек приложений NetBox:

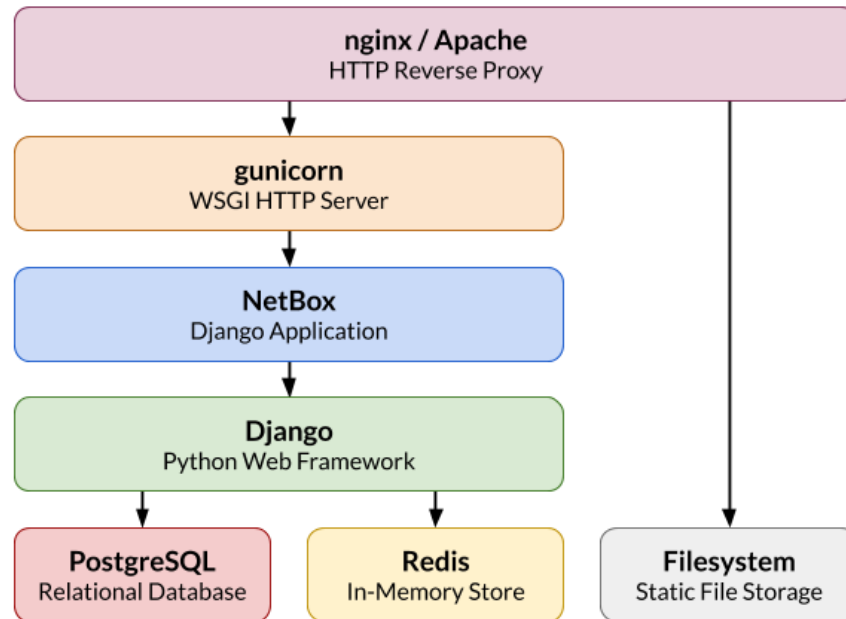


Рисунок 1.15 – Стек приложений NetBox.

Стек работает следующим образом [17]:

1. Nginx или apache сервер принимает и обрабатывает HTTP-запросы браузера, далее, передает их в Gunicorn;
2. Gunicorn разбирает данные. Согласно своей конфигурации, через протокол WSGI отдает их в Netbox;
3. NetBox – это web-приложение, написанное на фреймворке Django. Обрабатывает полученные данные, результат отдает обратно nginx/Apache серверу. Те отправляют их клиенту.

У ПО NetBox также имеется аналог в виде nautobot. Аналог имеет схожую функциональность. Предпочтение отдавалось NetBox ввиду его большей популярности и более высоких оценок от сообщества.

### 1.5.3 GNS3

GNS3 это ПО, которое способно реализовать эмуляцию виртуальной сетевой инфраструктуры [18].

GNS3 используется для:

- обучения сетевым технологиям;
- ознакомления с особенностями устройств;
- тестирование конфигураций устройств;
- реализация демонстрационного лабораторного стенда;
- замещение части реальной сетевой инфраструктуры.

## 2 Проектирование системы

В данном разделе приводится описание процесса проектирования системы автоматизации.

### 2.1 Моделирование процессов работы с сетевыми устройствами с учетом внедрения системы

Идея внедрения системы, автоматизирующей рутинные задачи в сетевом отделе, появилась при анализе существующих бизнес-процессов отдела, связанных с работой с сетевыми устройствами. Подразумевается, что внедрение системы должно изменить существующие бизнес-процессы при работе с сетевым оборудованием в положительную сторону: упростить алгоритм выполнения задач, структурировать задачи, уменьшить время на выполнение задач и так далее.

Моделирование бизнес-процессов компании после внедрения системы, позволяет оценить эффект от внедрения [3]. Модель ТО-ВЕ позволяет спланировать как изменяться существующие бизнес-процессы в компании после внедрения системы автоматизации. Данную модель можно использовать для планирования системы, представления заказчику ожидаемых результатов и оценки конечных результатов.

В таблице 2.1 приводится описание всех туннелей при моделировании бизнес-процессов ТО-ВЕ.

Таблица 2.1 – Описание туннелей для модели процессов ТО-ВЕ.

Сущность	На каких уровнях диаграммы встречается	Описание применения туннеля
Руководство Ansible	A0 A2 A21 A213 A22 A23 A231 A232 A233	Используется в качестве нормативного документа для использования ПО Ansible для процесса «Работа с Ansible» и всех нижестоящих. Не отображается на нижних уровнях процесса, но подразумевается его использование.
Файл конфигурации Ansible	A2	Текстовый файл, который хранит в себе конфигурацию ПО Ansible, т.е. отвечает за особенности работы ПО. Файл предварительно требуется создать и описать основные параметры. Обозначается туннелем, т.к. не имеет особой важности на вышестоящих процессах.

Продолжение таблицы 2.1

Сущность	На каких уровнях диаграммы встречается	Описание применения туннеля
Сотрудник отдела EN+ DIGITAL	A2 A21 A22	Сотрудник принимает участие в процессах «Настройка Ansible» и «Работа с группами и переменными». Обозначается туннелем т.к. не отображается на нижестоящих диаграммах, но подразумевается его использование в качестве исполняющего бизнес-процессы.
Конфигурация Ansible	A2 A22 A23 A231 A232 A233	Используется как документ, регулирующий работу ПО Ansible. Обозначается туннелем для процессов «Работа с группами и переменными» и «Работа с Playbook». На нижестоящих диаграммах не отображается, но подразумевается его использование.
Регламент документирования устройств	A2 A23 A231 A232 A233	Используется в качестве нормативного документа для документирования сетевых устройств. Обозначается туннелем для процесса «Работа с Playbook». Не отображается на нижестоящих процессах, но подразумевается его использование.
Регламент конфигурирования устройства	A2 A23 A231 A232 A233	Используется в качестве нормативного документа для конфигурирования сетевых устройств. Обозначается туннелем для процесса «Работа с Playbook». Не отображается на нижестоящих процессах, но подразумевается его использование.
Готовые Playbook	A231	Ранее написанные программы автоматизации, которые можно повторно использовать в новых задачах. Обозначается туннелем, т.к. не имеет особой важности в вышестоящих процессах.
Другие системы документирования	A233	Системы документирования сети, которые используются в сетевом отделе компании: Zabbix, NetFlow Analyzer, WhatsUp Gold. Обозначается туннелем, т.к. не имеет особой важности в вышестоящих процессах.

### 2.1.1 Работа с сетевым устройством (А-0)

На рисунке 2.1 представлена диаграмма IDEF0, описывающая верхний уровень бизнес-процесса «Работа с сетевым устройством» (А-0) после внедрения системы.



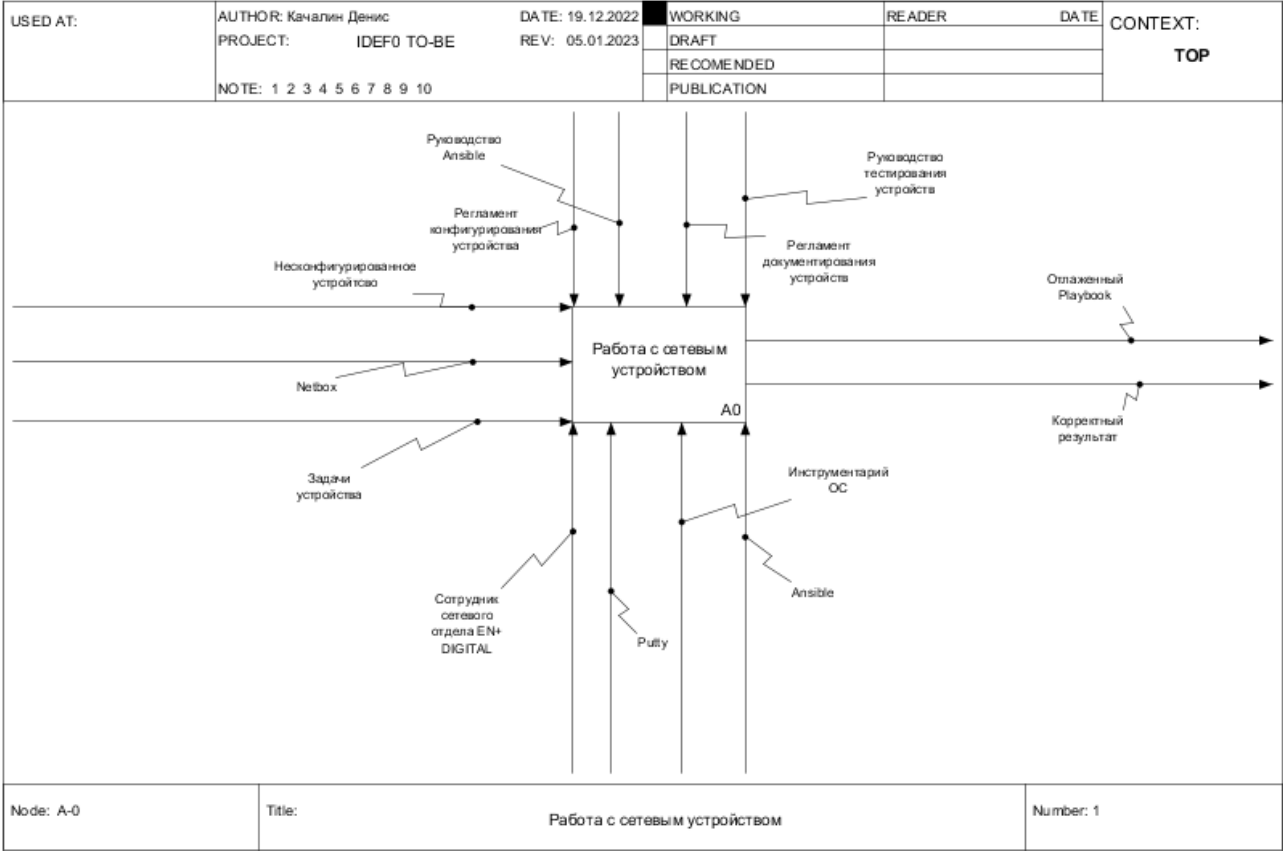


Рисунок 2.1 Диаграмма IDEF0 TO-BE для бизнес-процесса «Работа с сетевым устройством» верхнего уровня.

После внедрения системы бизнес-процесс процесс работы с сетевым устройством систематизируется, поэтому вместо конкретных результатов выполнения процесса, результат становится абстрактной сущностью, которая зависит от поставленной задачи. Таким образом задачи по документированию и конфигурированию после внедрения системы рассматриваются как однородные.

Помимо регламентов конфигурирования, документирования и тестирования сетевых устройств, появляется руководство по использованию ПО Ansible. Оно позволяет сетевому инженеру заниматься процессом автоматизации рутинных задач, используя ПО.

Бизнес-процесс выполняет сотрудник сетевого отдела EN+ DIGITAL, используя ПО Putty и Ansible, а также инструменты, встроенные в ОС.

Так как система, направлена на автоматизацию процессов конфигурирования и документирования, то для выполнения процесса требуется:

- несконфигурированное устройство;
- задачи, которые оно решает;
- инвентарную систему NetBox для документирования устройства и сети.

## 2.1.2 Работа с сетевым устройством (A0)

На рисунке 2.2 представлена диаграмма IDEF0 TO-BE, детализирующая описание бизнес-процесса «Работа с сетевым устройством» (A0) после внедрения системы.

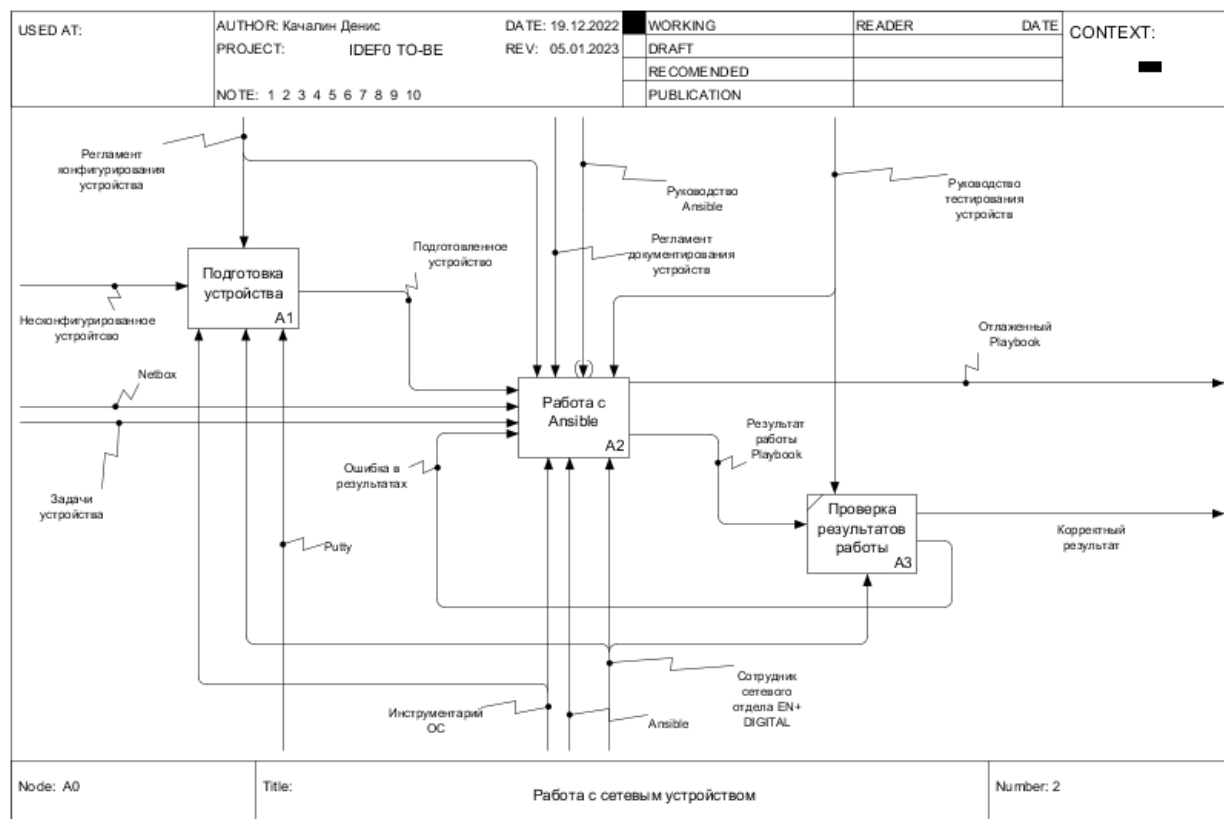


Рисунок 2.2 – Диаграмма IDEF0 TO-BE для бизнес-процесса «Работа с сетевым устройством» нижнего уровня.

Используя систему, работа с сетевым оборудованием сводится к 3 этапам:

### 1 этап – Подготовка устройства

Первичная подготовка устройства, для работы с системой. Для каждого устройства требуется хотя бы один раз провести эту настройку. В дальнейшем, если никаких глобальных изменений в работе устройства не будет, то перенастройка не требуется.

Подготовка устройства заключается в конфигурировании SSH соединения между устройством и машиной с системой автоматизации.

В качестве нормативного документа используется регламент конфигурирования.

Для подключения к устройству используется ПО Putty.

Для работы с SSH достаточно использование инструментария ОС.

Бизнес-процесс выполняет сотрудник сетевого отдела.

По итогу устройство будет подготовлено к работе с системой автоматизации.

## **2 этап – Работа с Ansible**

Процесс работы с ПО заключается в настройке ПО, описании устройств (работа с инвентарем) и написании программ автоматизации (работа с Playbook).

Для данного бизнес-процесса используются все регламенты (конфигурирование, документирование и тестирование), а также руководство Ansible.

Процесс исполняет сотрудник сетевого отдела, используя инструментарий ОС и ПО Ansible.

Чтобы выполнить задачи автоматизации, требуется система NetBox, подготовленное устройство и задачи, которое оно должно выполнять.

По итогу будут получены: новый отлаженный Playbook (программа автоматизации) и результат его исполнения, который нужно в дальнейшем проверить на корректность.

## **3 этап – Проверка результатов работы**

Результаты операций, которые выполнил Playbook, нужно проверить на корректность.

Проверка совершается согласно с регламентом тестирования устройства.

Процесс выполняет сотрудник сетевого отдела.

Если результат удовлетворителен, то на этом работа с устройством заканчивается.

Если результат не соответствует ожидаемому, то требуется выявить в чем была ошибка при исполнении Playbook и решить её.

### **2.1.3 Подготовка устройства (A1)**

На рисунке 2.3 представлена диаграмма IDEF0 TO-BE, детализирующая описание бизнес-процесса «Подготовка устройства» (A1).

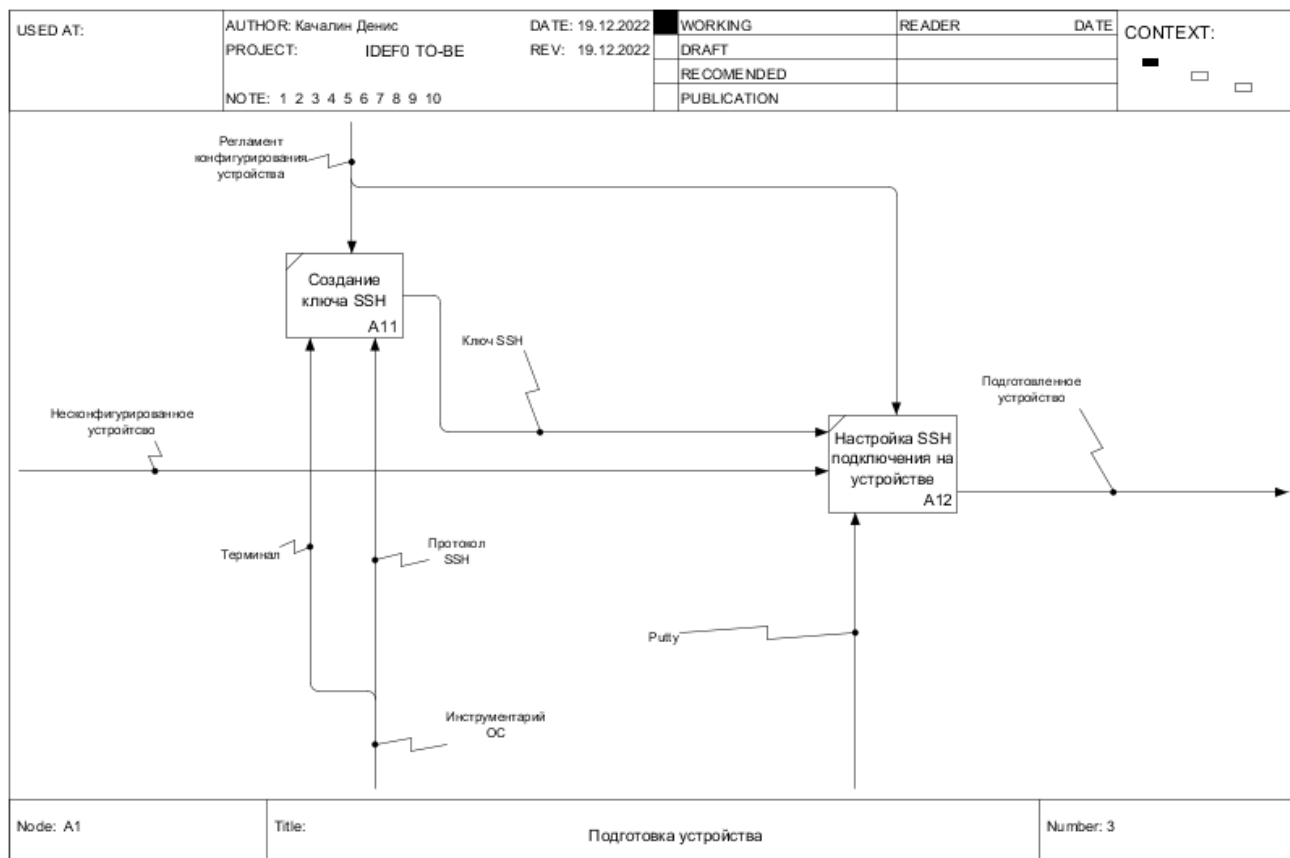


Рисунок 2.3 – Диаграмма IDEF0 TO-BE для бизнес-процесса «Подготовка устройства».

Чтобы работать с устройством, используя систему автоматизации, требуется установка удаленного доступа по протоколу SSH. Для повышения безопасности рекомендуется использование пары ключей SSH.

На машине с системой автоматизации необходимо создать пару ключей SSH согласно регламенту конфигурирования устройства.

Для этого используется терминал и протокол SSH – два инструмента ОС.

Ключи создает сотрудник.

Когда ключи были созданы, необходимо настроить SSH доступ к устройству. Для этого сотрудник, используя ПО Putty, удаленно подключается к устройству. Настраивает на нем протокол SSH для удаленного доступа по ключу.

Настройка SSH на устройстве совершается согласно регламенту конфигурирования устройства.

После этого устройство можно считать подготовленным к работе с системой автоматизации.

## 2.1.4 Работа с Ansible (A2)

На рисунке 2.4 представлена диаграмма IDEF0 TO-BE, детализирующая описание бизнес-процесса «Работа с Ansible» (A2).

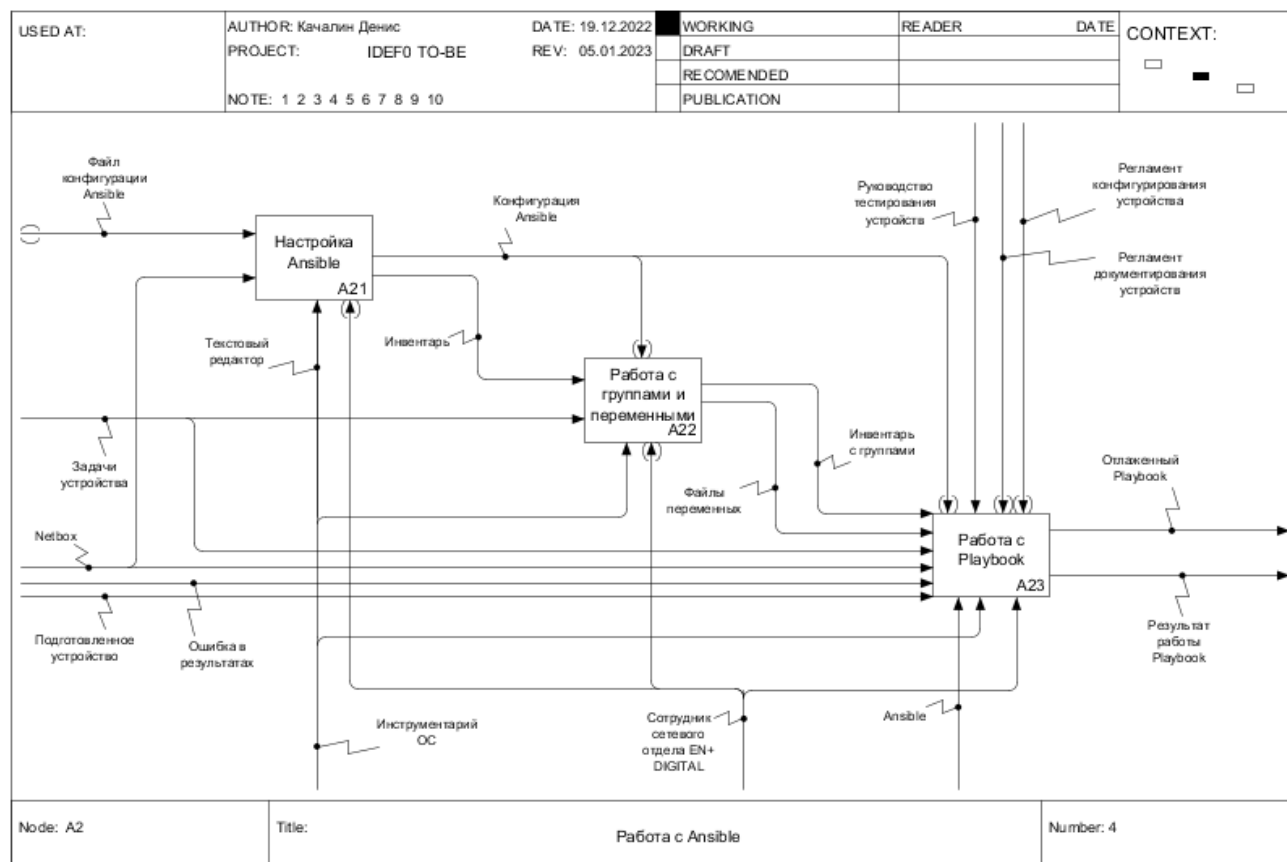


Рисунок 2.4 – Диаграмма IDEF0 TO-BE для бизнес-процесса «Работа с Ansible».

Работа с ПО Ansible выполняется в 3 этапа:

### 1 этап – Настройка Ansible

Чтобы Ansible решал поставленные задачи в самом начале его требуется настроить, т.е. указать ПО принципы его работы. В дальнейшем ПО можно перенастроить в зависимости от потребностей и задач.

Чтобы настроить Ansible, требуется файл конфигурации, который необходимо создать в начале работы с ПО.

Сотрудник сетевого отдела, используя терминал – инструмент ОС, настраивает ПО согласно руководству Ansible.

По итогу должны получиться конфигурация и инвентарь.

## **2 этап – Работа с группами и переменными**

Процесс посвящен описанию устройств, на которых выполняется автоматизация задач. Группы – объединяют устройства, чтобы можно было одновременно работа с несколькими устройствами одновременно. Переменные позволяют описывать как группы устройств, так и отдельные устройства.

Процесс выполняется по руководству Ansible и в соответствии с конфигурацией Ansible.

Для выполнения процесса, сотрудник сетевого отдела использует инструментарий ОС.

По итогу из задач устройства и инвентаря, будут составлены файлы с переменными, а в инвентаре появится описание групп устройств.

## **3 этап – Работа с Playbook**

Работа с программами автоматизации заключается в их написании, отладки и исполнении.

Для процесса требуются все регламенты (документирование, конфигурирование и тестирование устройств) и руководство Ansible.

Чтобы получить новый отлаженный Playbook и результат его исполнения, требуется: инвентарь, файлы переменных, ПО NetBox, подготовленное устройство.

Для выполнения процесса, сотрудник сетевого отдела использует инструментарий ОС и ПО Ansible.

### **2.1.5 Настройка Ansible (A21)**

На рисунке 2.5 представлена диаграмма IDEF0 TO-BE, детализирующая описание бизнес-процесса «Настройка Ansible» (A21).

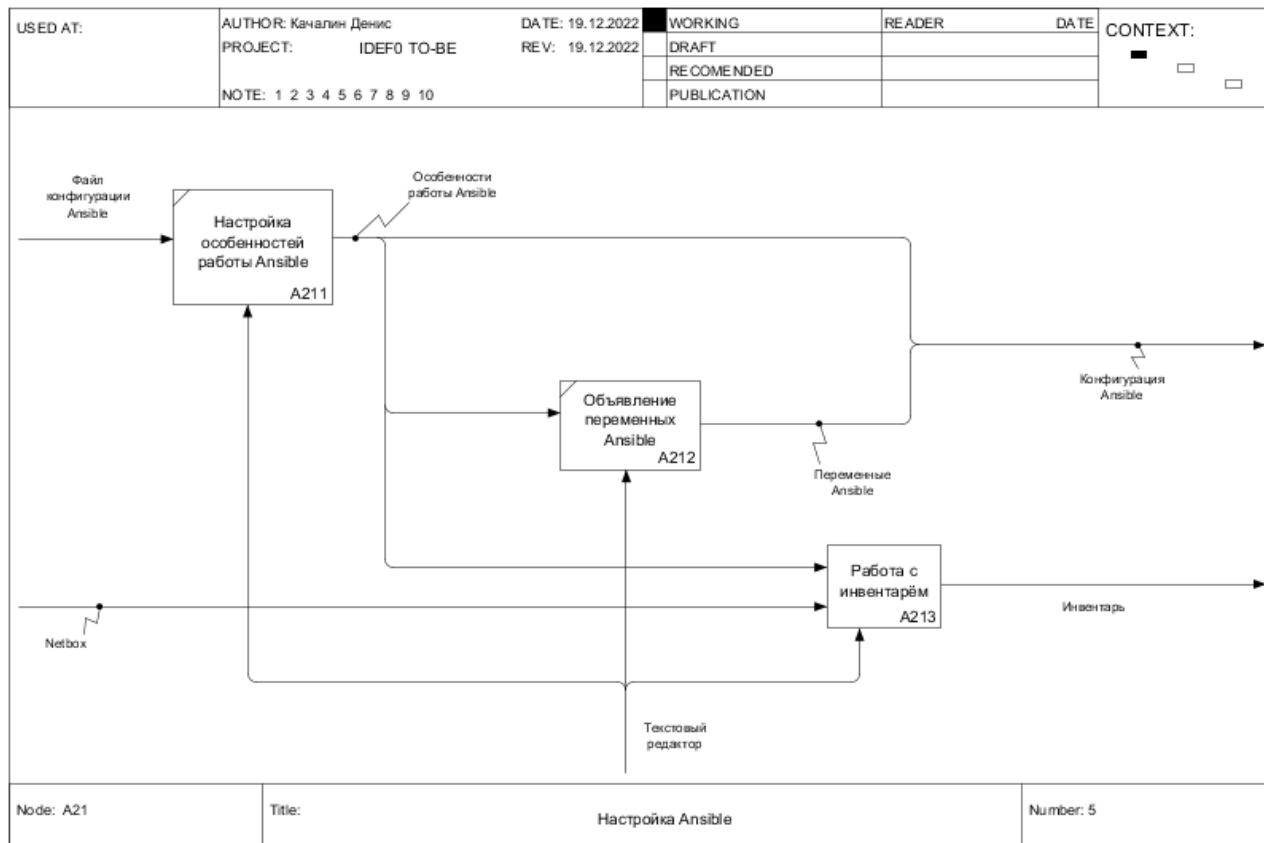


Рисунок 2.5 – Диаграмма IDEF0 TO-BE для бизнес-процесса «Настройка Ansible».

Под настройкой Ansible понимается работа с конфигурацией Ansible, т.е. описание работы ПО, объявление переменных Ansible и работа с инвентарём.

Сотрудник сетевого отдела, используя тестовый редактор изменяет файл конфигурации, тем самым описывая его принцип работы. Процесс выполняется в соответствии с руководством Ansible.

После описания особенностей работы ПО, потребуется объявить соответствующие переменные Ansible. Переменные и особенности работы ПО вместе образуют конфигурации Ansible. Процесс выполняется в файле конфигурации.

При настройке Ansible важно создать и описать инвентарь, в котором хранится информация об устройствах. В качестве инвентаря используется инвентарь Ansible и инвентарная система NetBox.

### 2.1.6 Работа с инвентарем (A213)

На рисунке 2.6 представлена диаграмма IDEF0 TO-BE, детализирующая описание бизнес-процесса «Работа с инвентарем» (A213).

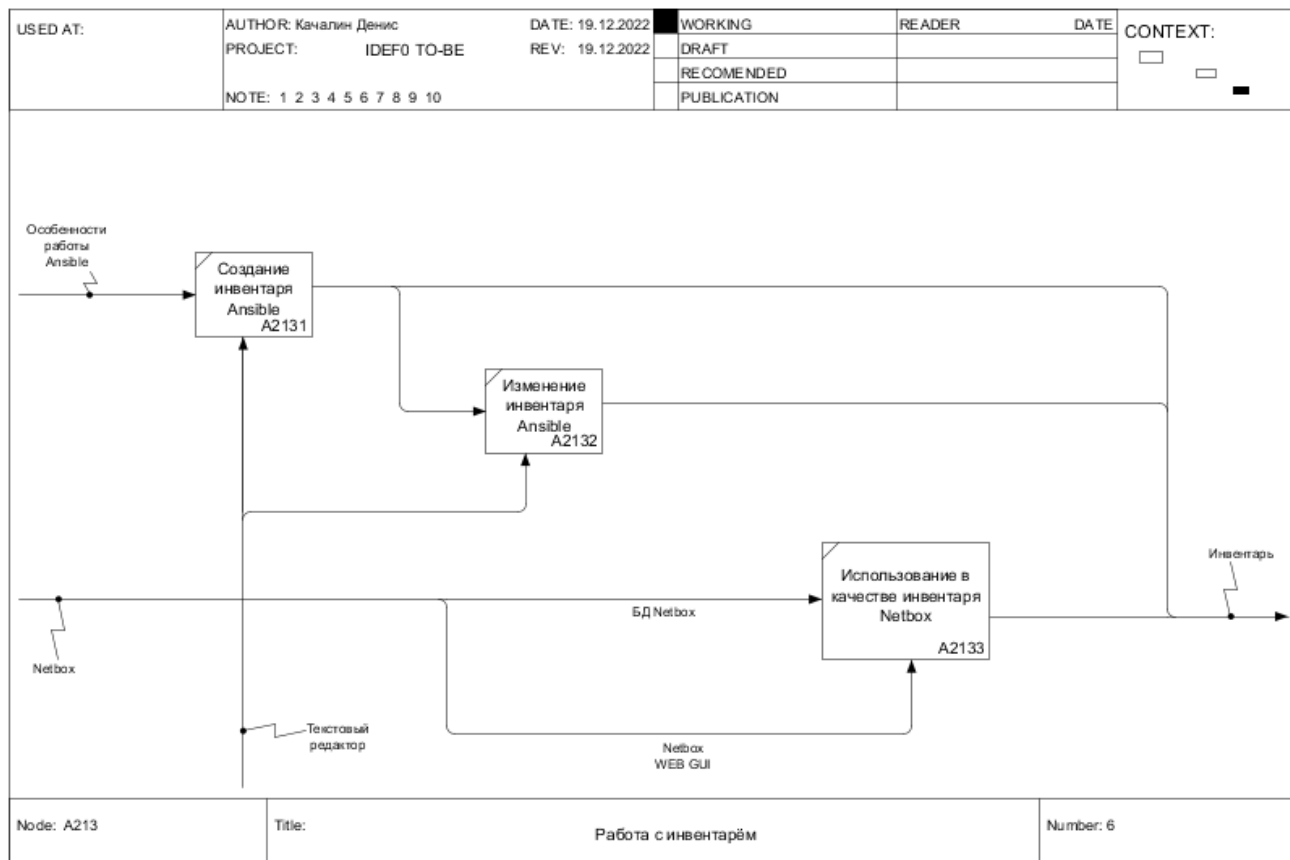


Рисунок 2.6 – Диаграмма IDEF0 TO-BE для бизнес-процесса «Работа с инвентарем».

Для более гибкой работы системы автоматизации, предполагается использование двух инвентарей. Инвентарь Ansible – служит для первичного описания устройства, которое будет использоваться в задачах автоматизации. NetBox – инвентарная система, которая хранит подробную информацию устройства и представляет её в наглядном виде. В отличие от инвентаря Ansible, информация в NetBox будет использоваться всеми сотрудниками сетевого отдела, т.е. не только в задачах автоматизации.

Работа с инвентарем Ansible заключается в создании файла инвентаря и наполнение его информацией об устройствах и дальнейшее его обновление. Для выполнения данных процессов используется текстовый редактор. Процесс выполняется сотрудником сетевого отдела в соответствии с руководством Ansible.

Использование NetBox как инвентаря позволяет подробно и систематизировано описать устройство. Для этого сотрудник сетевого отдела, используя WEB GUI, изменяет информацию, хранящуюся в БД NetBox.



### 2.1.7 Работа с группами и переменными (A22)

На рисунке 2.7 представлена диаграмма IDEF0 TO-BE, детализирующая описание бизнес-процесса «Работа с группами и переменными» (A22).

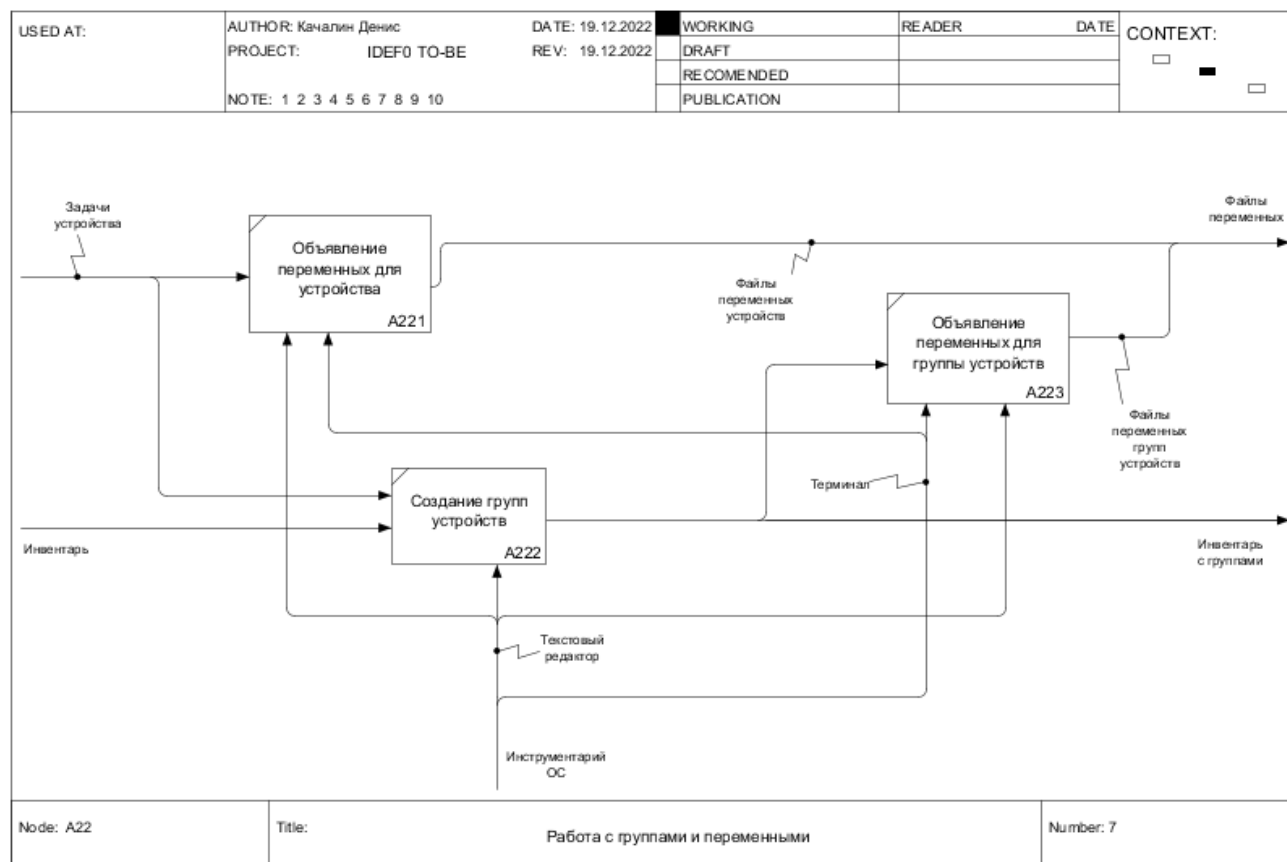


Рисунок 2.7 – Диаграмма IDEF0 TO-BE для бизнес-процесса «Работа с группами и переменными».

Создание групп позволяет выполнять задачи одновременно на нескольких подобных устройствах, а объявление переменных позволяет описать как с устройством работать и как его конфигурировать.

Переменные можно объявлять как для отдельных устройств, так и для целых групп. Для работы с переменными и группами было выделено 3 этапа:

## **1 этап – Объявление переменных для устройства**

Для объявления переменных для конкретного устройства, необходимо создать соответствующий файл в директории, где хранятся файлы с переменными для отдельных хостов.

Переменные объявляются исходя из задач устройства. Для корректного объявления переменных нужно учитывать конфигурацию Ansible и алгоритм, описанный в руководстве Ansible.

Чтобы объявить переменные, сотруднику достаточно воспользоваться терминалом ОС и текстовым редактором, чтобы создать и изменить соответствующий текстовый файл.

## **2 этап – Создание групп устройств**

Для объявления переменных для групп устройств, необходимо сначала создать эти группы.

Создаются группы в инвентаре исходя из их функциональности (задач, которые они решают).

В зависимости от инвентаря (Ansible или NetBox) будет отличаться инструмент, которым необходимо воспользоваться сотруднику сетевого отдела.

Если это инвентарь Ansible, то достаточно воспользоваться текстовым редактором.

Если это NetBox, то необходимо воспользоваться его WEB GUI.

Создание групп выполняется в соответствии с конфигурацией и руководством Ansible

По итогу в инвентаре появляются группы, к которым принадлежат схожие устройства по какому-либо признаку (местоположение или функциональность, к примеру).

## **3 этап – Объявление переменных для групп устройств**

Когда группа была создана, можно аналогично создать файл с переменными для группы устройств. Стоит учитывать, что файлы переменных для групп устройств и для отдельных устройств находятся в разных директориях.

Процесс выполняется согласно конфигурации и руководству Ansible.

Так же, как и для объявления переменных для устройств сотруднику достаточно использовать терминал ОС и текстовый редактор для работы с текстовыми файлами переменных групп устройств.

Для дальнейших процессов файлы переменных для групп устройств и файлы переменных для отдельных устройств, будут обозначаться просто как «Файлы переменных».

## 2.1.8 Работа с Playbook (A23)

На рисунке 2.8 представлена диаграмма IDEF0 TO-BE, детализирующая описание бизнес-процесса «Работа с Playbook» (A23).

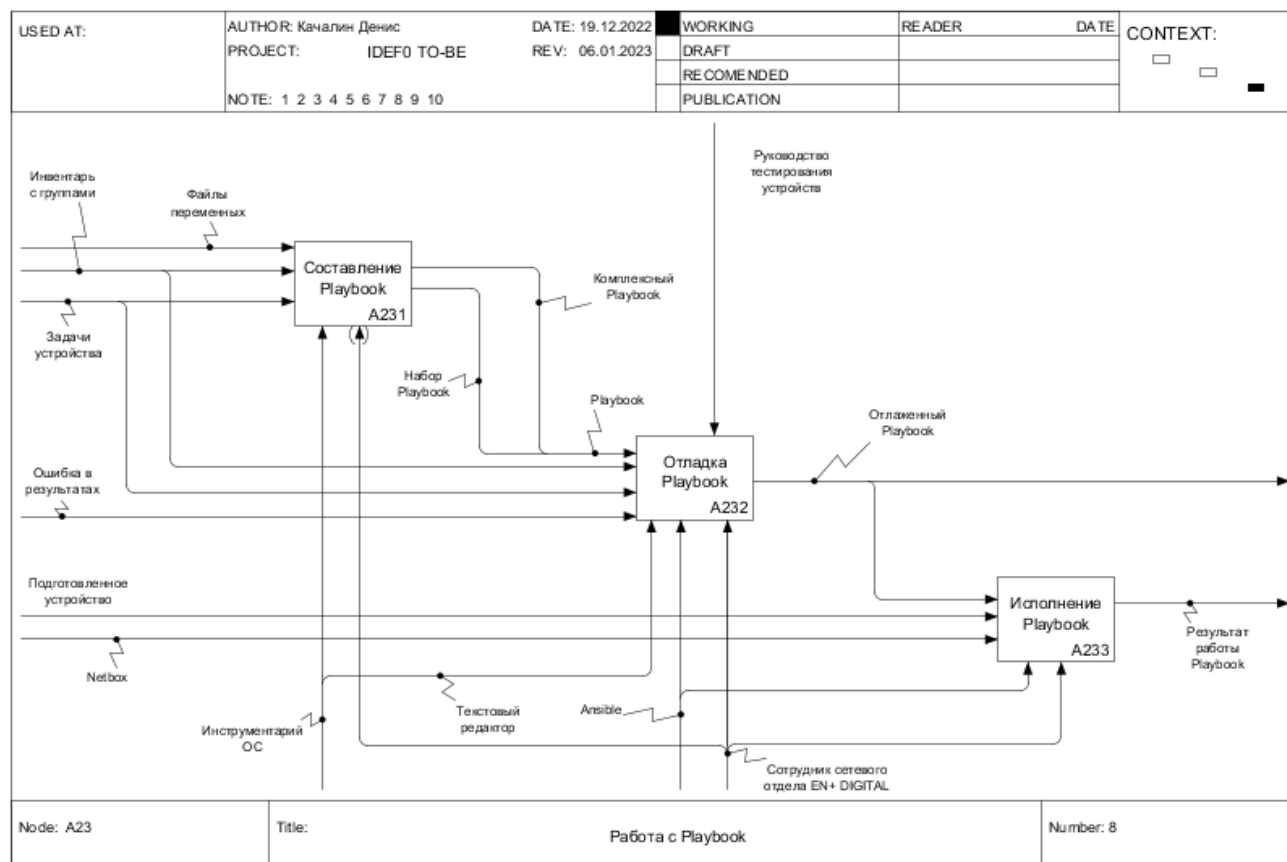


Рисунок 2.8 – Диаграмма IDEF0 TO-BE для бизнес-процесса «Работа с Playbook».

Работа с Playbook является самым важным процессом при работе с сетевым оборудованием после внедрения системы. Именно исполнение Playbook отвечает за все изменения, которые будут происходить в конфигурации устройства и в системах документации сети.

Процесс разделен на 3 этапа:

### 1 этап – Составление Playbook

Составление Playbook – это написание программы автоматизации какой-либо задачи. При написании необходимо учитывать логику работы устройства или системы документации, чтобы результат соответствовал результатам.

Основываясь на задачах устройства, инвентаре и переменных создается набор Playbook и один комплексных Playbook. Такой принцип рекомендован руководством Ansible для последующих отладки и поиска неисправностей.

Для написания Playbook сотруднику сетевого отдела достаточно воспользоваться инструментами ОС.

Процесс выполняется в соответствии с регламентами, которые регулируют автоматизированную задачу. К примеру, если проводится документирование устройства, то используются регламент документирования устройства. А если конфигурируем, то регламент конфигурирования устройства.

Так же при написании Playbook необходимо учитывать и конфигурацию Ansible.

## **2 этап – Отладка Playbook**

Написанный Playbook перед исполнением необходимо отладить на неисправности и ошибки, а также протестировать на виртуальном полигоне результат его исполнения.

Сотрудник сетевого отдела выполняет отладку в соответствии с регламентами конфигурирования, документирования и тестирования сетевого устройства, а также согласно руководству и конфигурации Ansible.

Отладка проводится при помощи средств Ansible и инструментов ОС.

При проведении отладки важно убедиться, что:

- задачи автоматизированы именно на тех устройствах, на которых было необходимо;
- устройства имеют именно ту функциональность, которая была необходима.

Когда отладка была проведена успешно, Playbook считается отлаженным и готовым к исполнению в реальной сетевой инфраструктуре.

## **3 этап – Исполнение Playbook**

Когда Playbook был написан и отлажен им можно пользоваться для решения конкретной задачи (задач) в настоящей сети.

Чтобы исполнить Playbook, сотруднику требуется выполнить соответствующую команду ПО Ansible.

Для исполнения Playbook требуется подготовленное устройство и NetBox.

Сотрудник сетевого отдела выполняет исполнение Playbook в соответствии с регламентами конфигурирования и документирования сетевого устройства, а также согласно руководству и конфигурации Ansible.

Результат исполнения далее следует проверить на корректность.

### **2.1.9 Составление Playbook (A231)**

На рисунке 2.9 представлена диаграмма IDEF0 TO-BE, детализирующая описание бизнес-процесса «Составление Playbook» (A231).

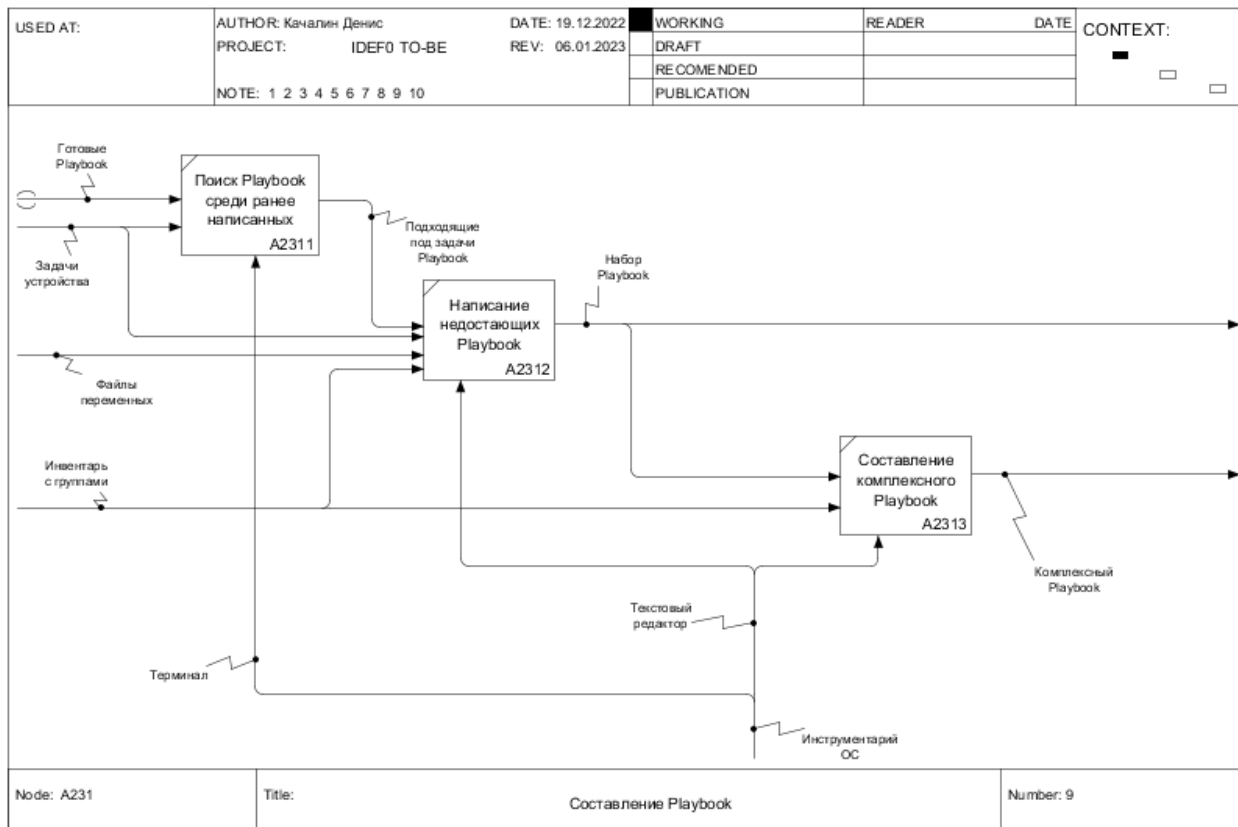


Рисунок 2.9 – Диаграмма IDEF0 TO-BE для бизнес-процесса «Составление Playbook».

Чтобы составить Playbook, который решает конкретную задачу на конкретном устройстве (группе устройств), следует:

- использовать готовые Playbook, которые решают часть задачи;
- написать новые Playbook, которые решают недостающие задачи;
- соединить набор Playbook в один комплексный.

### 1 этап – Поиск Playbook среди ранее написанных

Среди написанных Playbook следует отыскать те, которые можно использовать для решения текущей задачи. Поэтому важно грамотно структурировать файлы Ansible. Файлы Playbook стоит также называть так, чтобы было понятно какую задачу и на каких устройствах они решают. Это описано в руководстве Ansible и должно быть описано в регламентах конфигурирования и документирования сетевых устройств в компании после внедрения системы.

Чтобы осуществить поиск Playbook, сотруднику достаточно использовать терминал ОС.

### 2 этап – Написание недостающих Playbook

Если ранее написанные Playbook не могут в полной мере решить поставленную задачу, требуется написать новые.

На основе задач устройства, переменных и инвентаря сотрудник сетевого отдела пишет новые программы автоматизации. Для этого ему необходимо использовать текстовый редактор.

Написание новых Playbook выполняется в соответствии с регламентами конфигурирования и документирования сетевого устройства, а также согласно руководству и конфигурации Ansible.

В итоге должен получиться набор Playbook, которые полностью решают задачу. Далее требуется их объединить в один комплексный Playbook.

### 3 этап – Составление комплексного Playbook

Ранее написанные Playbook следует объединить в один комплексный, чтобы упростить исполнение и отладку.

Сотрудник сетевого отдела, используя текстовый редактор, из набора Playbook и инвентаря получает комплексный Playbook, решающий поставленную задачу.

Процесс выполняется в соответствии с регламентами конфигурирования и документирования сетевого устройства, а также согласно руководству и конфигурации Ansible.

#### 2.1.10 Отладка Playbook (A232)

На рисунке 2.10 представлена диаграмма IDEF0 TO-BE, детализирующая описание бизнес-процесса «Отладка Playbook» (A232).

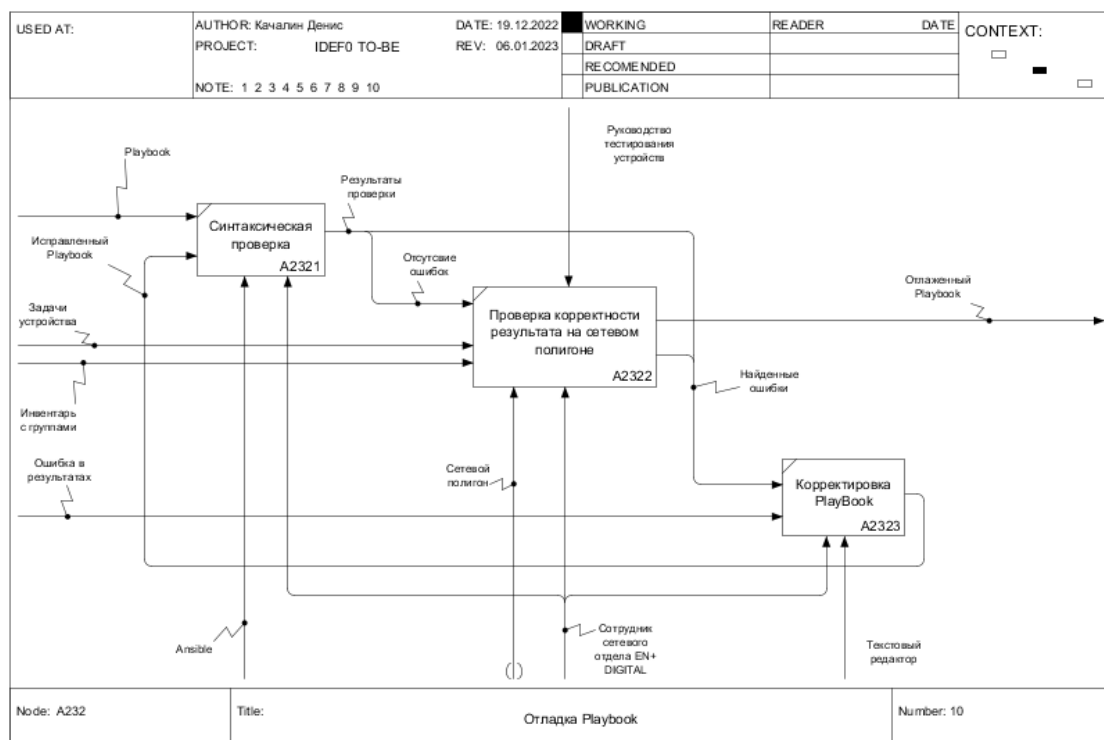


Рисунок 2.10 – Диаграмма IDEF0 TO-BE для бизнес-процесса «Отладка Playbook».

Написанный Playbook необходимо отладить перед исполнением.

Отладка состоит из 3 этапов:

### **1 этап – Синтаксическая проверка**

Требуется проверить кода программы автоматизации на соответствие синтаксису Ansible.

Для этого сотруднику сетевого отдела необходимо вызвать соответствующую команду Ansible и указать файл для проверки.

Процесс выполняется в соответствии с регламентами конфигурирования и документирования сетевого устройства, а также с руководством и конфигурацией Ansible.

Если в результате проверки ошибки не были обнаружены, то проводится проверка на логику программы автоматизации на специальном виртуальном сетевом полигоне.

Если в результате проверки были обнаружены ошибки, то Ansible выдаст соответствующее сообщение. В сообщении будут описаны синтаксические ошибки, которые необходимо исправить.

### **2 этап – Проверка корректности результатов на сетевом полигоне**

Сетевой полигон – это виртуальная часть сети, которая имитирует работу настоящей.

В данном процессе сотруднику требуется на сетевом полигоне выявить, что задачи были выполнены на определенных устройствах.

Процесс выполняется в соответствии с регламентами конфигурирования и документирования сетевого устройства, а также согласно руководству и конфигурации Ansible.

Если ошибок не было выявлено можно приступить к исполнению Playbook в реальной сети. А Playbook считается отлаженным.

Если были выявлены ошибки, то код Playbook требуется исправить.

### **3 этап – Корректировка Playbook**

Если Playbook не соответствует требованиям, то его необходимо исправить. Для этого сотруднику сетевого отдела требуется при помощи текстового редактора выявить место несоответствия в коде программы автоматизации и исправить его.

Исправленный Playbook отправляется на повторную синтаксическую и логическую проверку (на полигоне).

#### **2.1.11 Исполнение Playbook (A233)**

На рисунке 2.11 представлена диаграмма IDEF0 TO-BE, детализирующая описание бизнес-процесса «Исполнение Playbook» (A233).

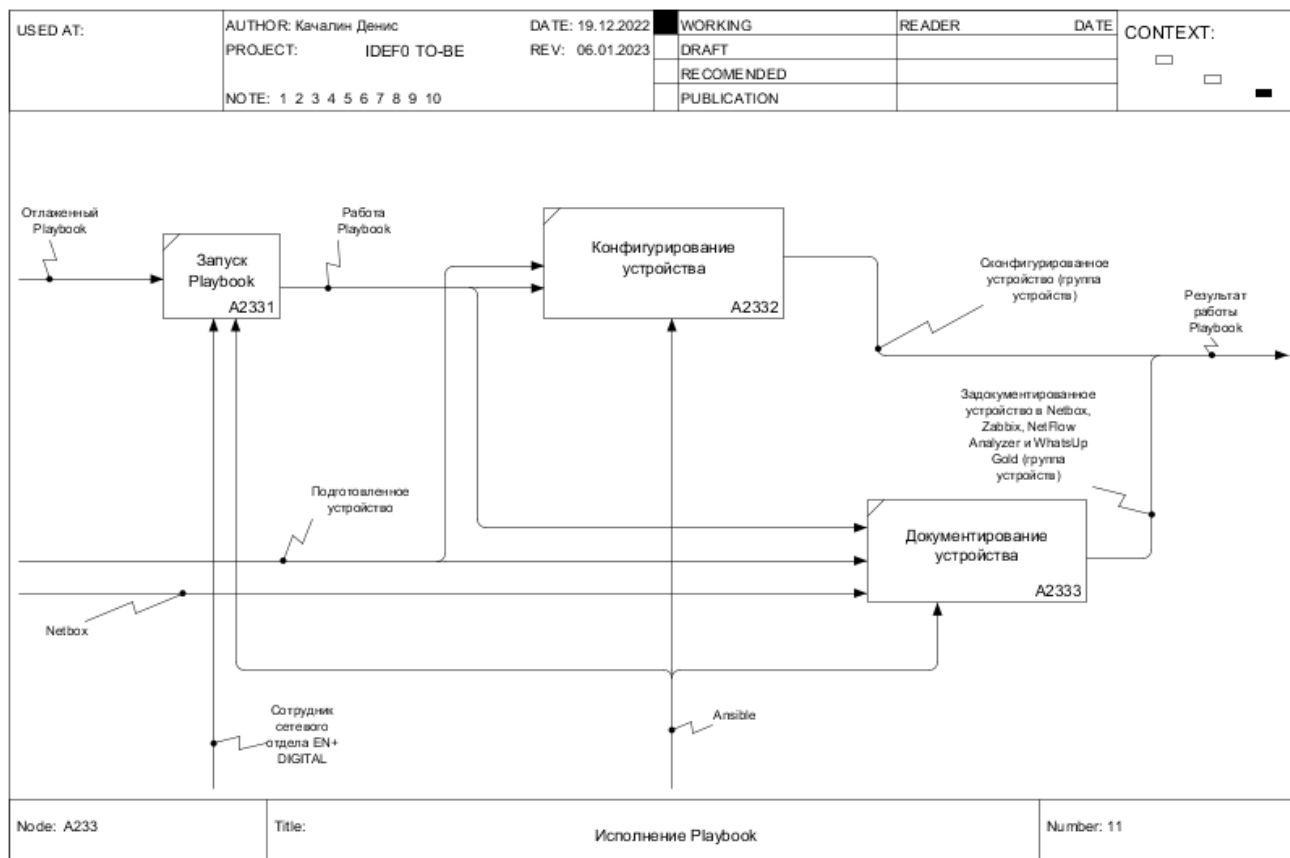


Рисунок 2.11 – Диаграмма IDEF0 TO-BE для бизнес-процесса «Исполнение Playbook».

После того как Playbook был написан и отлажен его можно исполнить, чтобы решить поставленную задачу в реальной сети.

Процесс выполняется в соответствии с регламентами конфигурирования и документирования сетевого устройства, а также согласно руководству и конфигурации Ansible.

Сотрудник сетевого отдела запускает исполнение Playbook при помощи ПО Ansible.

Далее Ansible выполняет код программы автоматизации:

- Конфигурирует подготовленное устройство;
- Документирует подготовленное устройство в системе NetBox и других системах документирования сети (Zabbix, WhatsUp Gold, Netflow Analyzer).

Сконфигурированное устройство и информация о нем в системах документирования сети считается как результат работы Playbook, которые далее требуется проверить на корректность (см. п. 2.1.2 «Работа с сетевым устройством (A0)»).



## 2.2 Варианты использования системы

В данном разделе приводится информация о вариантах использования системы для автоматизации задач при работе с сетевым оборудованием.

На рисунке 2.12 представлена диаграмма вариантов использования системы автоматизации.

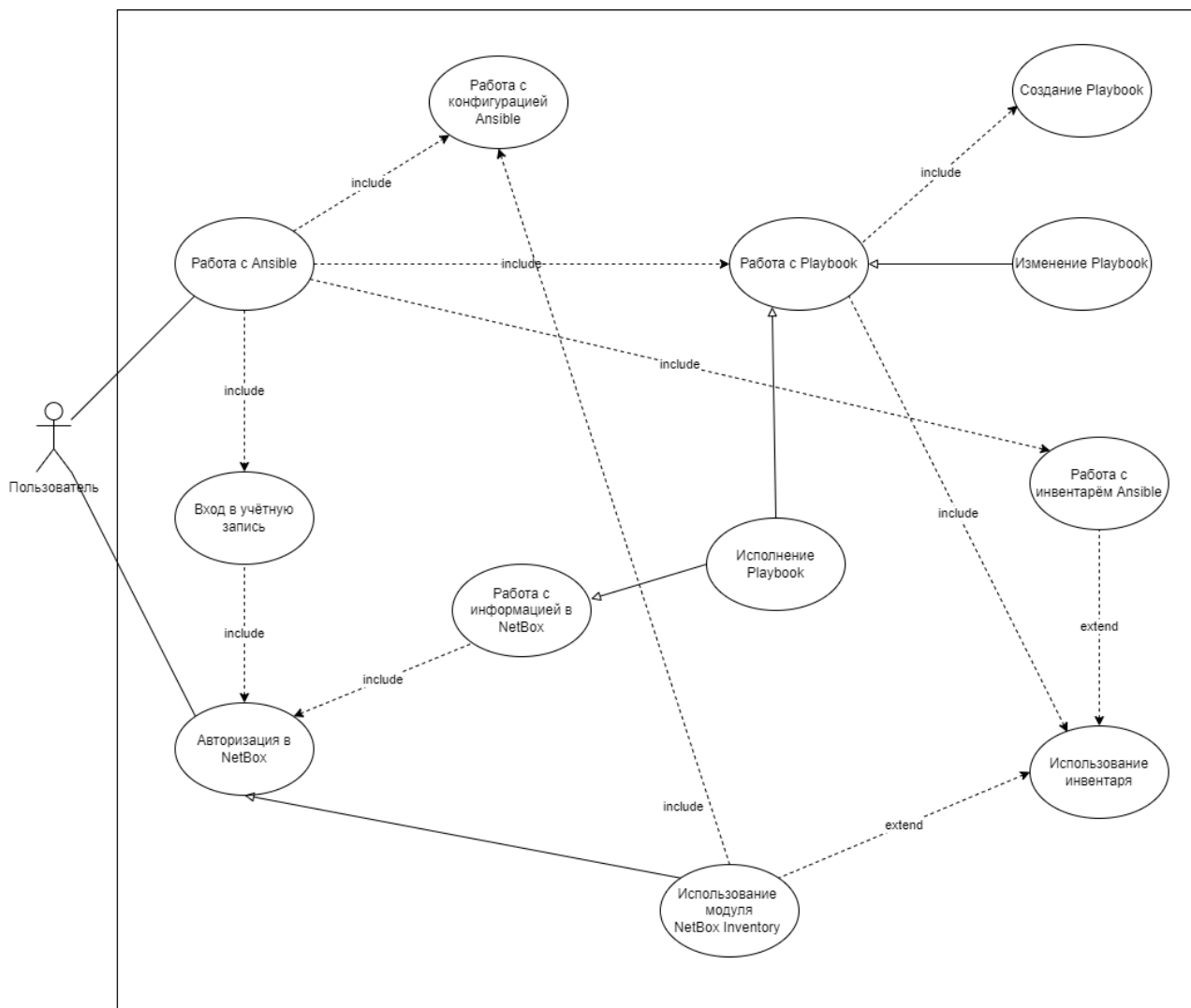


Рисунок 2.12 – Use-case диаграмма.

Далее приводится описание сценариев использования системы для решения конкретных задач [19].

### 2.2.1 Сценарий «Авторизация в системе»

Перед началом работы с системой пользователю требуется подтвердить свой набор прав, посредством авторизации в ОС на ЭВМ с системой и в ПО NetBox. Описание сценария приведено в таблице 2.2. Описание альтернатив сценария приведено в таблице 2.3.

Таблица 2.2 – Описание сценария «Автоматизация в системе»

<b>Действующие лица</b>	Пользователь, ОС, NetBox
<b>Цели</b>	Пользователь: авторизоваться в системе и начать работать с ней; ОС: идентифицировать пользователя и его права при работе с ОС; NetBox: идентифицировать пользователя и его права при работе с NetBox.
<b>Успешный сценарий:</b> <ol style="list-style-type: none"> <li>1. Пользователь подключается по удаленному доступу к ЭВМ с системой;</li> <li>2. ОС открывает сессию пользователя, предлагает ввести логин и пароль;</li> <li>3. Пользователь вводит логин и пароль;</li> <li>4. ОС проверяет логин и пароль;</li> <li>5. ОС выдает пользователю сообщение об успешной авторизации;</li> <li>6. Вызов сценария «Авторизация в NetBox» (п. 2.2.10).</li> </ol>	
<b>Результат</b>	Пользователь авторизован в системе, имеет выданный ему регламентом набор прав и может приступать к работе с системой.

Таблица 2.3 – Описание альтернатив сценария «Авторизация в системе»

<b>Расширения:</b>	
1a	ЭВМ не доступна. ПО для удаленного доступа выдает соответствующее сообщение. Результат: пользователь не может подключиться к ЭВМ с системой.
5a	Пользователь с введенным логином и паролем не найден. ОС выдает соответствующее сообщение. Результат: отказ в авторизации. Переход на шаг 2.

### 2.2.2 Сценарий «Работа с Ansible»

Сценарий работы с Ansible – это комплексный сценарий состоящих из нескольких подсценариев, которые решают разные задачи. Среди них: работа с конфигурацией Ansible, работа с Playbook, работа с инвентарём Ansible.

В таблице 2.4 приводится описание сценария «Работа с Ansible», который является подготовительным для решения дальнейших задач. В таблице 2.5 – описание альтернатив сценария.

Таблица 2.4 – Описание сценария «Работа с Ansible»

<b>Действующие лица</b>	Пользователь, терминал ОС
<b>Цели</b>	Пользователь: начать работать с Ansible; Терминал ОС: предоставить возможности для работы с ОС.
<b>Успешный сценарий:</b> <ol style="list-style-type: none"> <li>1. Вызов сценария «Авторизация в системе» (п. 2.2.1);</li> <li>2. Пользователь открывает терминал;</li> <li>3. Пользователь, используя терминал, создает нужную директорию;</li> <li>4. Пользователь, используя терминал, открывает нужную директорию.</li> </ol>	
<b>Результат</b>	Пользователь успешно начал работать с Ansible.

Таблица 2.5 – Описание альтернатив сценария «Работа с Ansible»

Расширения:	
3a	Директория уже существует. Переход к шагу 4.

### 2.2.3 Сценарий «Работа с конфигурацией Ansible»

Перед работой с Ansible нужно провести базовую настройку. Ansible можно настроить под свои нужды и особенности бизнес-процессов. Описание сценария приведено в таблице 2.6. Описание альтернатив сценария – в таблице 2.7.

Таблица 2.6 – описание сценария «Работа с конфигурацией Ansible»

<b>Действующие лица</b>	Пользователь, терминал ОС, текстовый редактор
<b>Цели</b>	Пользователь: настроить Ansible под свои нужды; Терминал ОС: предоставить возможности для работы с ОС; Текстовый редактор: предоставить возможность работать с текстовыми файлами.
<b>Успешный сценарий:</b> <ol style="list-style-type: none"> <li>1. Вызов сценария «Работа с Ansible» (п 3.2);</li> <li>2. Пользователь, используя терминал, создает файл конфигурации;</li> <li>3. Пользователь, используя терминал, открывает файл конфигурации в текстовом редакторе;</li> <li>4. Пользователь вводит параметры, используя текстовый редактор;</li> <li>5. Пользователь сохраняет файл, используя текстовый редактор.</li> </ol>	
<b>Результат</b>	Пользователь успешно настроил конфигурацию Ansible.

Таблица 2.7 – Описание альтернатив сценария «Работа с конфигурацией Ansible»

Расширения:	
2a	Файл конфигурации уже существует. Переход к шагу 3.

### 2.2.4 Сценарий «Работа с Playbook»

Сам по себе сценарий не имеет последовательности действий и конкретного результата. Работа с Playbook – это обобщение важных для работы с системой сценариев при взаимодействии с Playbook. Данный сценарий включает следующие подсценарии: создание, изменение и исполнение Playbook, а также использование инвентаря. Но для работы с Playbook обязательно наличие хотя бы одного Playbook, поэтому необходимо его создание. Описание сценария представлено в таблице 2.8. Описание альтернатив сценария – в таблице 2.9.

Таблица 2.8 – Описание сценария «Работа с Playbook»

<b>Действующие лица</b>	Пользователь
<b>Цели</b>	Пользователь: начать работать с Playbook.
<b>Успешный сценарий:</b> 1. Начать работать с Playbook.	
<b>Результат</b>	Пользователь начал работу с Playbook.

Таблица 2.9 – Описание альтернатив сценария «Работа с Playbook»

<b>Расширения:</b>	
1a	Ни один Playbook ещё не был создан. Вызов сценария «Создание Playbook» (п. 2.2.5).

### 2.2.5 Сценарий «Создание Playbook»

Создание Playbook – это первый шаг к автоматизации рутинной задачи. Помимо простого создания файла, сценарий включает в себе написания программы автоматизации и синтаксическую проверку Playbook. Описание сценария приведено в таблице 2.10. Описание альтернатив сценария – в таблице 2.11.

Таблица 2.10 – Описание сценария «Создание Playbook»

<b>Действующие лица</b>	Пользователь, терминал ОС, текстовый редактор, Ansible
<b>Цели</b>	Пользователь: создать, написать Playbook и проверить его на синтаксические ошибки; Терминал ОС: предоставить возможности для работы с ОС; Текстовый редактор: предоставить возможность работать с текстовыми файлами; Ansible: проверить корректность синтаксиса кода Playbook.
<b>Успешный сценарий:</b> 1. Вызов сценария «Работа с Playbook» (п. 2.2.4); 2. Пользователь, используя терминал, создает файл Playbook; 3. Пользователь, используя терминал, открывает Playbook в текстовом редакторе; 4. Пользователь пишет код Playbook, используя текстовый редактор; 5. Пользователь сохраняет Playbook, используя текстовый редактор; 6. Пользователь запускает синтаксическую проверку Playbook Ansible; 7. Ansible не выдает ошибок по окончании проверки.	
<b>Результат</b>	Пользователь успешно создал и написал Playbook.

Таблица 2.11 – Описание альтернатив сценария «Создание Playbook».

<b>Расширения:</b>	
2a	Файл уже существует. Вызов сценария «Изменение Playbook» (п. 2.2.6).
7a	Ansible выдает ошибку при синтаксической проверке Playbook. Результат: в Playbook найдена ошибка. Вызов сценария «Изменение Playbook» (п. 2.2.6).

### 2.2.6 Сценарий «Изменение Playbook»

Под изменением Playbook подразумевается либо дополнение существующего кода для детализации автоматизации задачи, либо исправление

ошибок в логике или синтаксисе программы. Описание сценария приведено в таблице 2.12. Описание альтернатив сценария – в таблице 2.13.

Таблица 2.12 – Описание сценария «Изменение Playbook»

<b>Действующие лица</b>	Пользователь, терминал ОС, текстовый редактор, Ansible
<b>Цели</b>	Пользователь: изменить написанный Playbook и проверить его на синтаксические ошибки; Терминал ОС: предоставить возможности для работы с ОС; Текстовый редактор: предоставить возможность работать с текстовыми файлами; Ansible: проверить корректность синтаксиса в Playbook.
<b>Успешный сценарий:</b>	<ol style="list-style-type: none"> <li>1. Вызов сценария «Работа с Playbook» (п 3.4);</li> <li>2. Пользователь, используя терминал, открывает Playbook в текстовом редакторе;</li> <li>3. Пользователь вносит изменения в Playbook, используя текстовый редактор;</li> <li>4. Пользователь сохраняет Playbook, используя текстовый редактор;</li> <li>5. Пользователь запускает синтаксическую проверку Playbook Ansible;</li> <li>6. Ansible не выдает ошибок по окончании проверки.</li> </ol>
<b>Результат</b>	Пользователь успешно изменил Playbook.

Таблица 2.13 – Описание альтернатив сценария «Изменение Playbook»

<b>Расширения:</b>	
2а	Файл не существует. Вызов сценария «Создание Playbook» (п. 2.2.5).
6а	Ansible выдает ошибку при синтаксической проверке Playbook. Результат: в Playbook найдена ошибка. Переход к шагу 2.

### 2.2.7 Сценарий «Работа с инвентарем Ansible»

Инвентарь – неотъемлемый элемент при автоматизации задач. В инвентарном файле описываются устройства, с которыми работает система. Также в файле инвентаризации устройства распределяются на группы, с которыми в последствии можно работать как с единой сущностью. Также важной частью является и объявление переменных устройств и групп устройств. Это позволяет описать важную информацию об устройстве или группе, и в дальнейшем с ней работать в задачах автоматизации. Описание сценария приведено в таблице 2.14. Описание альтернатив сценария в таблице 2.15.

Таблица 2.14 – Описание сценария «Работа с инвентарём Ansible»

<b>Действующие лица</b>	Пользователь, терминал ОС, текстовый редактор
<b>Цели</b>	Пользователь: поработать с инвентарем Ansible; Терминал ОС: предоставить возможности для работы с ОС; Текстовый редактор: предоставить возможность работать с текстовыми файлами.
<b>Успешный сценарий:</b> <ol style="list-style-type: none"> <li>1. Вызов сценария «Работа с Ansible» (п. 2.2.2);</li> <li>2. Пользователь, используя терминал создает файл инвентаря Ansible;</li> <li>3. Пользователь, используя терминал, открывает файл в текстовом редакторе;</li> <li>4. Пользователь вносит изменения в файл, используя текстовый редактор;</li> <li>5. Пользователь сохраняет файл, используя текстовый редактор.</li> </ol>	
<b>Результат</b>	Пользователь успешно поработал с инвентарем Ansible.

Таблица 2.15 – Описание альтернатив сценария «Работа с инвентарём Ansible»

<b>Расширения:</b>	
2а	Файл уже существует. Переход к шагу 3.

### 2.2.8 Сценарий «Исполнение Playbook»

Чтобы выполнить рутинную задачу при работе с сетевым оборудованием, необходимо запустить заранее написанный Playbook. Результат проконтролировать. Если результат не удовлетворяет пользователя или во время исполнения Playbook возникла ошибка, необходимо перепроверить конфигурацию Ansible, описание инвентаря и корректность логики и синтаксиса исполняемого Playbook. Описание сценария приведено в таблице 2.16. Описание альтернатив сценария – в таблице 2.17.

Таблица 2.16 – Описание сценария «Исполнение Playbook»

<b>Действующие лица</b>	Пользователь, терминал ОС, Ansible
<b>Цели</b>	Пользователь: исполнить написанный Playbook для автоматизирования задачи; Терминал ОС: предоставить возможности для работы с ОС; Ansible: исполнение Playbook и информирование пользователя о процессе исполнения.
<b>Успешный сценарий:</b>	<ol style="list-style-type: none"> <li>1. Вызов сценария «Работа с Playbook» (п. 2.2.4);</li> <li>2. Пользователь, используя терминал и команды Ansible запускает исполнение Playbook;</li> <li>3. Ansible исполняет код Playbook, а также информирует пользователя о прогрессе исполнения;</li> <li>4. Ansible заканчивает исполнение Playbook, уведомляет пользователя;</li> <li>5. Пользователь проверяет результаты исполнения Playbook.</li> </ol>
<b>Результат</b>	Автоматизированная задача была выполнена при помощи Playbook. Результат удовлетворяет пользователя.

Таблица 2.17 – Описание альтернатив сценария «Работа с инвентарём Ansible»

<b>Расширения:</b>	
2a	Файл не существует. Вызов сценария «Создания Playbook» (п. 2.2.5).
3a	В процессе исполнения Playbook возникла ошибка. Результат: задача не была выполнена или выполнена частично. Вызов сценария «Работа с конфигурацией Ansible» (п. 2.2.3). Вызов сценария «Изменение Playbook» (п. 2.2.6). Вызов сценария «Работа с инвентарём Ansible» (п. 2.2.7).
5a	Пользователь не удовлетворён результатами исполнения Playbook. Результат: задача была выполнена некорректно или частично корректно. Вызов сценария «Работа с конфигурацией Ansible» (п. 2.2.3). Вызов сценария «Изменение Playbook» (п. 2.2.6). Вызов сценария «Работа с инвентарём Ansible» (п. 2.2.7).

### 2.2.9 Сценарий «Использование инвентаря»

Инвентарь – это описание устройств, с которыми работает система автоматизации. Инвентарь можно описывать множественными способами. Самый основной – это инвентарь Ansible. В нём для конкретного устройства или группы можно указать любую информацию, а потом её использовать в задачах автоматизации. Но для структурирования описания используется отдельная система инвентаризации NetBox, которая хранит информацию об устройствах в БД, а также имеет удобный WEB GUI для работы с пользователем и REST API для работы с приложениями. В зависимости от задачи используется либо инвентарь Ansible, либо инвентарь NetBox. Описание сценария приведено в таблице 2.18.

Таблица 2.18 – Описание сценария «Использование инвентаря»

<b>Действующие лица</b>	Ansible, инвентарь
<b>Цели</b>	Ansible: получения информации об устройствах из инвентаря; Инвентарь: предоставление информации об устройствах.
<b>Успешный сценарий:</b>	1. Ansible запрашивает необходимую ему информации при работе с Playbook; 2. Инвентарь предоставляет всё запрашиваемую информацию Ansible.
<b>Результат</b>	Ansible получил информации об устройствах из инвентаря.

### 2.2.10 Сценарий «Авторизация в NetBox»

Авторизация в ПО NetBox необходимо для просмотра и редактирования информации, которое храниться в нём. Так как система выполняет функции инвентаризации сетевых устройств и документирование сети, то доступна она, не только пользователем системы автоматизации, а всему сетевому отделу. Помимо пользователей в виде людей, которым предоставлен доступ к NetBox. Также подразумевается различное ПО, которое работает с NetBox не через сайт, а через REST API. Описание сценария приведено в таблице 2.19. Описание альтернатив сценария – в таблице 2.20.

Таблица 2.19 – Описание сценария «Авторизация в NetBox»

<b>Действующие лица</b>	Пользователь, NetBox
<b>Цели</b>	Пользователь: авторизоваться в NetBox.
<b>Успешный сценарий:</b>	1. Получение доступа к NetBox; 2. Предоставление данных авторизации; 3. NetBox проверяет данные авторизации; 4. NetBox предоставляет доступ.
<b>Результат</b>	Пользователь авторизовался в NetBox.

Таблица 2.20 – Описание альтернатив сценария «Авторизация в NetBox»

<b>Расширения:</b>	
1a	NetBox недоступен. Результат: пользователь не может получить доступ к NetBox.
5a	Данные не прошли проверку. NetBox отклоняет запрос. Результат: отказ в авторизации. Переход на шаг 1.

### 2.2.11 Сценарий «Работа с информацией в NetBox»

Получив доступ к NetBox, пользователь может просматривать или изменять информацию, которая в нём храниться при наличии соответствующих прав. Так же, как и при авторизации под пользователем может подразумеваться и другое ПО. К примеру, Ansible при исполнении Playbook. Все действия, проводимые с информацией NetBox, документируются в виде логов ПО. Описание сценария приведено в таблице 2.21. Описание альтернатив сценария – в таблице 2.22.



Таблица 2.21 – Описание сценария «Работа с информацией в NetBox»

<b>Действующие лица</b>	Пользователь, NetBox
<b>Цели</b>	Пользователь: поработать с информацией в NetBox.
<b>Успешный сценарий:</b> 1. Вызов сценария «Авторизация в NetBox» (п. 2.2.10); 2. Пользователь работает с информацией; 3. NetBox документирует действия пользователя.	
<b>Результат</b>	Пользователь поработал с информацией в NetBox.

Таблица 2.22 – Описание альтернатив сценария «Работа с информацией в NetBox»

<b>Расширения:</b>	
2а	У пользователя недостаточно прав для изменения информации. Результат: пользователь не внес изменения в NetBox.

### 2.2.12 Сценарий «Использование модуля NetBox Inventory»

Чтобы использовать систему NetBox как инвентарь для задач автоматизации Ansible, существует модуль NetBox Inventory [20], который упрощает взаимодействие Ansible и NetBox REST API. Для авторизации используется специальный токен NetBox. Чтобы указать Ansible, что использовать в качестве инвентаря NetBox, то нужно прописать соответствующие параметры в конфигурации Ansible. Описание сценария в таблице 2.23.

Таблица 2.23 – Описание сценария «Работа с информацией в NetBox»

<b>Действующие лица</b>	Пользователь, NetBox
<b>Цели</b>	Пользователь: начать использовать NetBox в качестве инвентаря Ansible.
<b>Успешный сценарий:</b> 1. Вызов сценария «Авторизация в NetBox» (п. 2.2.10); 2. Пользователь копирует значение токена NetBox; 3. Вызов сценария «Работа с конфигурацией Ansible» (п. 2.2.3).	
<b>Результат</b>	Пользователь начал использовать NetBox в качестве инвентаря Ansible.

## 2.3 Разработка программ автоматизации

В данном разделе приводится описание программ автоматизации, которые должны решать основные рутинные задачи, возникающие в сетевом отделе компании EN+ DIGITAL.

### 2.3.1 Программы автоматизации конфигурирования

Основными задачами по конфигурированию сетевых устройств считается настройка интерфейсов и подсетей, маршрутизация, списки доступа, VPN-туннели и NAT. Поэтому было решено для прототипа системы автоматизации сконцентрироваться на автоматизации этих задач.

#### 2.3.1.1 Автоматизация настройки интерфейсов и подсетей

Чтобы настроить сетевую инфраструктуру, требуется настроить интерфейсы устройства и подсети пользователей. Для разделения трафика

используется тегирование. Этой задачей занимается технология VLAN [21]. Для демонстрационных целей будет достаточно разделить трафик на трафик пользователей и трафик управления сетевыми устройствами. С описанием программы автоматизации можно ознакомиться в таблице 2.24

Таблица 2.24 – Описание программы автоматизации настройки интерфейсов и подсетей.

<b>Название программы</b>	VLAN_users_maintenance.yml
<b>Цель</b>	Настроить интерфейсы устройства и подсети для пользовательского трафика и трафика управления.
<b>Результат</b>	Интерфейсы сконфигурированы и работают. Трафик тегруется при помощи технологии VLAN.
<b>Алгоритм</b>	<ol style="list-style-type: none"> <li>1) собрать информацию об интерфейсах на устройствах;</li> <li>2) сконфигурировать и поднять физические интерфейсы;</li> <li>3) сконфигурировать VLAN-интерфейсы;</li> <li>4) связать VLAN и физические интерфейсы;</li> <li>5) проверить конфигурацию устройств.</li> </ol>
<b>Требования</b>	Устройство подготовлено к работе с системой; Описать IP-пространство в файлах переменных; Описать интерфейсы устройства в файлах переменных.

### 2.3.1.2 Автоматизация настройки маршрутизации

Маршрутизация позволяет устройствам отправлять пакеты по сети. Если устройство имеет маршрут до определенного адресата, тогда можно передать ему информацию. Протоколы динамической автоматизации, позволяют упростить настройку маршрутизации. Для демонстрации возможности автоматизации конфигурирования маршрутизации был выбран один из самых популярных протоколов – OSPF [22]. С описанием программы автоматизации можно ознакомиться в таблице 2.25

Таблица 2.25 – Описание программы автоматизации настройки OSPF.

<b>Название программы</b>	OSPF_office_main_routers.yml
<b>Цель</b>	Настроить протокол OSPF на главных маршрутизаторах офисов.
<b>Результат</b>	Настройки OSPF добавлены в конфигурацию устройств, между офисами установлены маршруты для передачи пакетов.
<b>Алгоритм</b>	<ol style="list-style-type: none"> <li>1) собрать информацию об интерфейсах и подсетях на устройствах;</li> <li>2) запустить протокол OSPF на устройствах;</li> <li>3) указать router-id для протокола OSPF на устройствах;</li> <li>4) добавить интерфейсы и подсети в таблицы маршрутизации OSPF;</li> <li>5) дождаться момента, когда устройства обременяются таблицами маршрутизации;</li> <li>6) проверить доступность сетей из других офисов.</li> </ol>
<b>Требования</b>	<p>Устройство подготовлено к работе с системой;          Описать IP-пространство в файлах переменных;          Описать интерфейсы устройства в файлах переменных;          Устройства должны иметь физическое подключение;          Описать переменные для OSPF в файлах переменных.</p>

### 2.3.1.3 Автоматизация настроек списков доступа

ACL – access control lists, это инструмент позволяющий разрешать или запрещать определенный трафик в определенное направление сети с устройства. Другими словами – это фильтр, который регулирует движения трафика [23]. С описанием программы автоматизации по настройке ACL можно ознакомиться в таблице 2.26

Таблица 2.27 – Описание программы автоматизации настройки ACL.

<b>Название программы</b>	ACL_office.yml
<b>Цель</b>	Настроить списки доступа, чтобы трафик пользователей не мог идти в подсети, где устройств пользователей нет.
<b>Результат</b>	ACL настроены, и пользователи не могут отправить трафик в определенные участки сети.
<b>Алгоритм</b>	<ol style="list-style-type: none"> <li>1) собрать информацию об интерфейсах и подсетях на устройствах;</li> <li>2) создать ACL;</li> <li>3) связать ACL и интерфейс;</li> <li>4) проверить недоступность участка сети со стороны пользователя.</li> </ol>
<b>Требования</b>	<p>Устройство подготовлено к работе с системой;          На устройствах должны быть настроены интерфейсы и подсети;          На устройстве должна быть настроена маршрутизация;          Описать IP-пространство в файлах переменных;          Описать интерфейсы устройства в файлах переменных.</p>

#### 2.3.1.4 Автоматизация настройки VPN-туннелей

VPN-туннели – это средство для защиты трафика через потенциально опасный участок сети [24]. Для демонстрации был выбран популярный протокол VPN-туннелей IKEv2 [25]. С описанием программы автоматизации можно ознакомиться в таблице 2.28.

Таблица 2.28 – Описание программы автоматизации настройки VPN-туннелей.

<b>Название программы</b>	IKEv2_office_main_routers.yml
<b>Цель</b>	Настроить VPN-туннели между главными маршрутизаторами офисов.
<b>Результат</b>	VPN-туннели сконфигурированы и работают. Трафик шифруется.
<b>Алгоритм</b>	1) собрать информацию об интерфейсах и подсетях на устройствах; 2) настроить IKEv2 profile; 3) настроить туннель между устройствами; 4) назначить IKEv2 profile на туннель; 5) добавить туннель в OSPF; 6) проверить работу туннеля и шифрования данных.
<b>Требования</b>	Устройство подготовлено к работе с системой; На устройствах должны быть настроены интерфейсы и подсети; На устройствах должна быть настроена маршрутизация; Описать информацию необходимую для настройки VPN-туннеля в файлах переменных; Описать переменные для OSPF в файлах переменных; Описать IP-пространство в файлах переменных; Описать интерфейсы устройства в файлах переменных.

#### 2.3.1.5 Автоматизация настройки NAT

NAT – это технология подмены IP-адресов. Используется для экономии адресного пространства сети Интернет. При попытке выхода в интернет заменяет серый адрес на белый и вносит соответствующую запись в таблицу NAT [23]. С описанием программы автоматизации можно ознакомиться в таблице 2.29.

Таблица 2.29 – Описание программы автоматизации настройки NAT.

<b>Название программы</b>	NAT_office_main_routers.yml
<b>Цель</b>	Настроить NAT на главных маршрутизаторах офисов.
<b>Результат</b>	NAT сконфигурирован. Адреса подменяются при выходе в сеть.
<b>Алгоритм</b>	<ol style="list-style-type: none"> <li>1) создать пул адресов для подмены;</li> <li>2) создать ACL для определения белого адреса;</li> <li>3) связать ACL и пул (создать NAT);</li> <li>4) навесить NAT на интерфейс;</li> <li>5) проверить работу NAT.</li> </ol>
<b>Требования</b>	<p>Устройство подготовлено к работе с системой;</p> <p>На устройствах должны быть настроены интерфейсы и подсети;</p> <p>На устройствах должна быть настроена маршрутизация;</p> <p>Обозначить IP-пространство в файлах переменных;</p> <p>Описать IP-пространство в файлах переменных;</p> <p>Описать интерфейсы устройства в файлах переменных.</p>

### 2.3.2 Программы автоматизации документирования

Документирование сети важный элемент работы сетевого инженера. К документации он обращается при решении задач или возникновении неполадок на участке сети или устройстве. Поэтому важно чтобы документирование было подробным и имело актуальную информацию [26].

Документирование можно разделить на 2 этапа: инвентаризация и добавление устройства в системы мониторинга сети.

Первый отвечает за документирование информации об устройстве.

Второе – за документирование его работы в сети.

#### 2.3.2.1 Инвентаризация устройства

Для инвентаризации устройства выбрана система NetBox (согласно п. 1.4.6). Инвентаризация происходит при помощи программ автоматизации и специального модуля, написанного разработчиками NetBox. С описанием программы автоматизации можно ознакомиться в таблице 2.30.

Таблица 2.30 – Описание программы автоматизации инвентаризации устройства.

<b>Название программы</b>	device_inventory.yml
<b>Цель</b>	Внести информацию об устройстве в систему NetBox.
<b>Результат</b>	Информация внесена в NetBox и доступна для чтения.
<b>Алгоритм</b>	<ol style="list-style-type: none"> <li>1) собрать всю информацию об устройстве;</li> <li>2) внести запись об устройстве в NetBox;</li> <li>3) проверить корректность информации.</li> </ol>
<b>Требования</b>	<p>Устройство подготовлено к работе с системой;</p> <p>Устройство должно быть настроено для функционирования в сети;</p> <p>Указать данные для авторизации в NetBox в файлах переменных;</p> <p>Указать дополнительную информацию об устройстве в файлах переменных. Информацию, которую невозможно собрать с устройства.</p>

### 2.3.2.2 Добавление устройства в системы мониторинга сети

В отделе на данный момент используется ПО Zabbix, WhatsUp Gold, NetFlow Analyzer для отслеживания работы сети. Поэтому важно, чтобы как можно больше устройств было внесено в данное ПО. Это позволит оперативнее решать различные вопросы и неисправности. С описанием программы автоматизации можно ознакомиться в таблице 2.31.

Таблица 2.32 – Описание программы автоматизации добавление устройства в системы мониторинга сети.

<b>Название программы</b>	device_monitoring.yml
<b>Цель</b>	Внести информацию об устройстве в системы Zabbix, WhatsUp Gold, NetFlow Analyzer.
<b>Результат</b>	Информация внесена в Zabbix, WhatsUp Gold, NetFlow Analyzer и доступна для чтения.
<b>Алгоритм</b>	<ol style="list-style-type: none"> <li>1) собрать всю информацию об интерфейс и адресе управления устройства;</li> <li>2) настроить протокол SNMPv3 на устройстве;</li> <li>3) добавить устройство в Zabbix;</li> <li>4) добавить устройство в WhatsUp Gold;</li> <li>5) добавить устройство в NetFlow Analyzer;</li> <li>6) проверить корректность информации.</li> </ol>
<b>Требования</b>	<p>Устройство подготовлено к работе с системой;</p> <p>Устройство должно быть настроено для функционирования в сети;</p> <p>Указать данные для авторизации в системы в файлах переменных;</p> <p>Указать данные для настройки протокола SNMPv3 в файлах переменных.</p>

## 2.4 Архитектура системы

В Таблице 2.33 приводится состав программных средств системы автоматизации.

Таблица 2.33 – Перечень программных средств системы автоматизации.

№	Наименование (рекомендуемая версия)	Цель использования
1	Ubuntu (v22.04)	ОС, на которой работает система автоматизации [27].
2	Python (3.10.8)	язык программирования, необходимый для работы NetBox [16] и Ansible [12].
3	Ansible (6.5.0)	ПО автоматизации [12].
4	NetBox (3.3.5)	ПО инвентаризации сети [16].
5	PostgreSQL (14.5)	СУБД для NetBox. Хранит данные о сети [16].
6	Redis (4.0)	СУБД для NetBox. Хранит информацию об операциях в ПО [17].
7	GNS3 (2.2.34)	ПО для эмуляции сетевой инфраструктуры [18].
7.1	GNS3 VM for VirtualBox (2.2.34)	образ виртуальной машины GNS3, требуется для работы GNS3. Эмулирует работу устройств [18].
8	Oracle VM VirtualBox Manager (6.1.38)	гипервизор. Для виртуализации ОС и GNS3 VM [28].

На рисунке 2.13 приводится техническая архитектура системы

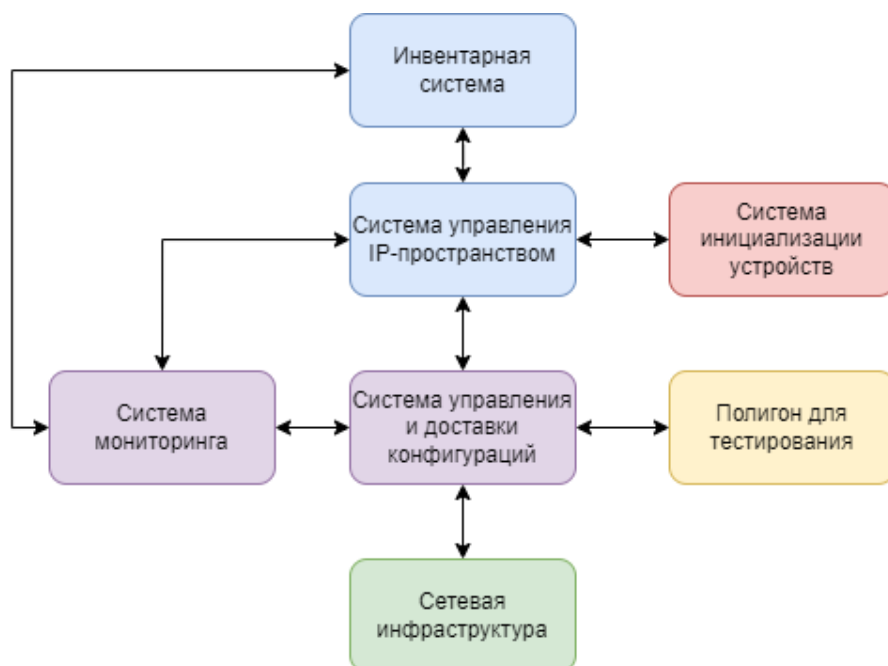


Рисунок 2.13 – Техническая архитектура системы

Инвентарная система позволяет хранить данные об устройствах.

Система управления IP-пространством позволяет контролировать, планировать и выделять IP-адреса, подсети, VLAN'ы под различные нужды компании.

Система инициации устройств осуществляет быстрое внедрение нового устройства в сеть.

Система мониторинга отслеживает состояние устройств.

Система управления и доставки конфигураций позволяет работать с конфигурациями устройств (разрабатывать, изменять, удалять) и осуществляет доставку и установку конфигураций на устройства. Конфигурация устройства зависит от множества факторов (особенности модели, где оно располагается, какие задачи должно выполнять).

Полигон для тестирования – виртуальная копия реальной сетевой инфраструктуры или её части для тестирования новых конфигураций.

Сетевая инфраструктура – реальная сетевая инфраструктура, которая обеспечивает передачу данных группы компаний EN+ GROUP.

## 2.5 Развертывание системы

Развертывание системы важный процесс внедрения системы [29]. Используя методологию UML, было описано как следует систему подготовить к работе. На рисунке 2.14 представлена диаграмма развертывания системы.

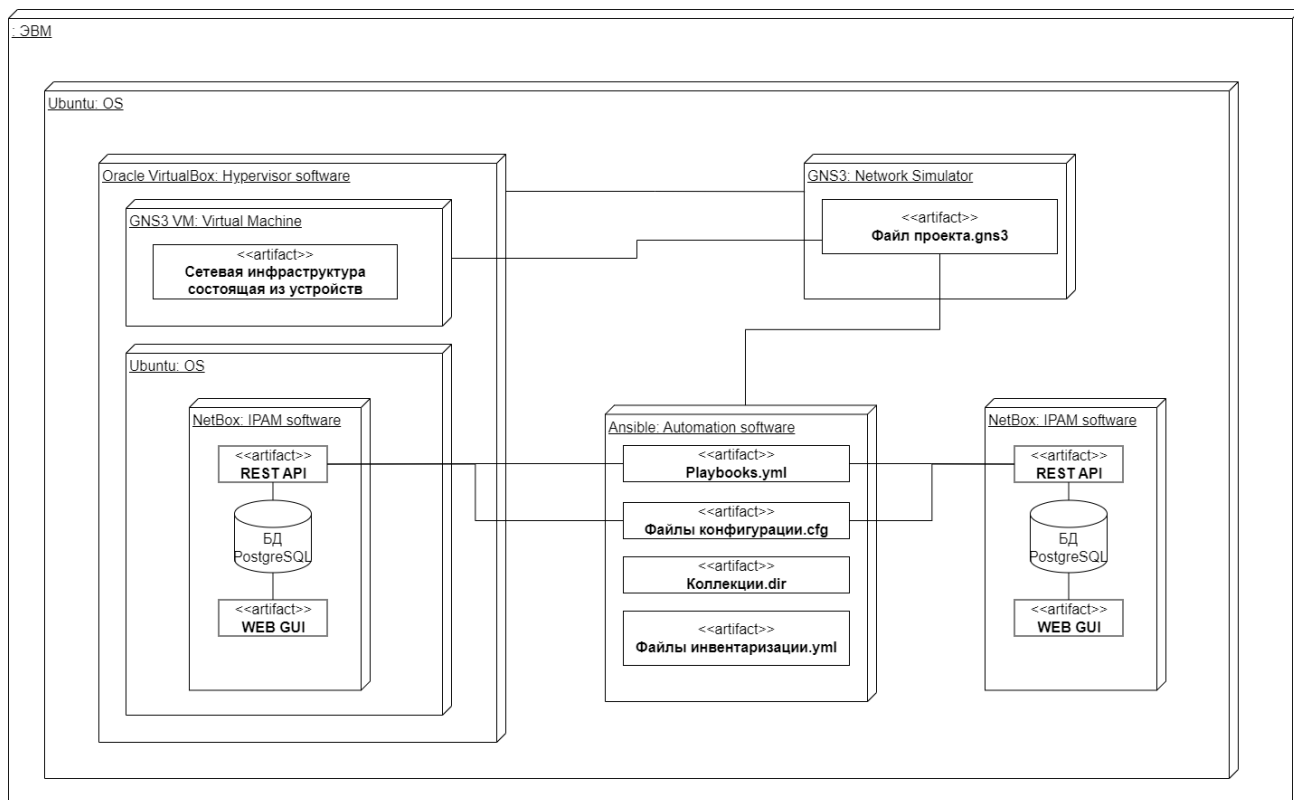


Рисунок 2.14 – Диаграмма развертывания системы.



Система должна быть расположена на компьютере с операционной системой Ubuntu. В операционной системе должны быть установлены:

- гипервизор Oracle VirtualBox;
- симулятор сети GNS3;
- Ansible;
- NetBox.

В Oracle VirtualBox должны быть импортированы образы GNS3 VM и ОС Ubuntu.

Виртуальная машина GNS3 нужна для корректной работы GNS3. Без неё ПО работать не будет. На ней происходит процесс эмуляции сетевой инфраструктуры. В GNS3 должен быть создан файл с проектом, в котором будет храниться вся информация о виртуальных устройствах и связях между ними. Файл будет иметь расширение формата «название\_файла.gns3».

Образ ОС Ubuntu в гипервизоре необходим для развертывания ПО NetBox для тестирования программ автоматизации на виртуальном сетевом полигоне.

На ОС системы и образе в гипервизоре должно быть установлено ПО необходимое для работы Ansible и NetBox (согласно п. 2.4).

В Ansible будут присутствовать следующие артефакты:

- playbook, файлы в формате yaml;
- файл конфигурации, файл в формате «ansible.cfg»;
- коллекции, папки с внешними (написанные сторонними разработчиками) модулями, ролями, плагинами;
- файлы инвентаризации, файлы в формате yaml.

Для NetBox должна быть создана БД в СУБД PostgreSQL, в которой будет храниться вся информация о сети. Через REST API, playbook и файлы конфигурации будут обращаться к БД за информацией.

### 3 Реализация

В данном разделе приводится описание последовательности процессов при реализации системы автоматизации на 14.12.2022.

#### 3.1 Состав и содержание работ по созданию системы

Все работы по реализации выполняются в соответствии с планом работ, представленным в таблице 3.1. Каждый из этапов работ имеет описание содержания работы, результаты и сроки.

Таблица 3.1 – Состав и содержание работ по созданию системы

№	Наименование работ	Основное содержание работ	Результаты выполнения работ	Сроки выполнения работ
1	Постановка задачи	– анализ БП в компании; – описание проблемы в БП; – составление требований; – постановка задачи; – составление отчетности и документации.	– описание БП на текущий момент в компании; – требования к системе; – цель работы; – задача работы.	начало: 01.09.2022  окончание: 15.09.2022
2	Анализ предметной области инструментария автоматизации управления сетевым оборудованием	– изучение сферы автоматизации управления сетевым оборудованием; – поиск готового решения; – сравнение решений; – подведение результатов сравнения; – составление отчетности и документации.	– знание о сфере автоматизации управления сетевым оборудованием; – список готовых решений; – анализ готовых решений; – сравнение готовых решений по различным критериям.	начало: 16.09.2022  окончание: 01.10.2022
3	Выбор решения	– обоснование выбранного решения; – детальное изучение решения; – описание предлагаемого решения; составление отчетности и документации.	– выбранное решение; – обоснование выбора решения; – детализированный анализ выбранного решения.	начало: 02.10.2022  окончание: 15.10.2022

Продолжение таблицы 3.1

№	Наименование работ	Основное содержание работ	Результаты выполнения работ	Сроки выполнения работ
4	Проектирование системы	<ul style="list-style-type: none"> <li>– составление БП при использовании системы;</li> <li>– составление вариантов использования системы;</li> <li>– составление описания программ автоматизации;</li> <li>– составление отчетности и документации.</li> </ul>	<ul style="list-style-type: none"> <li>– описание БП после внедрения системы;</li> <li>– описание вариантов использования системы;</li> <li>– описание программ автоматизации.</li> </ul>	<p>начало: 16.10.2022</p> <p>окончание: 20.11.2022</p>
5	Разработка системы	<ul style="list-style-type: none"> <li>– установка и настройка ПО;</li> <li>– создание сетевого полигона;</li> <li>– подготовка устройств к работе с системой;</li> <li>– создание инвентаря;</li> <li>– создание программ автоматизации;</li> <li>– отладка программ автоматизации;</li> <li>– составление отчетности и документации;</li> <li>– подготовка демо версий для презентации заказчику;</li> <li>– подготовка презентаций;</li> <li>– презентация наработок заказчику.</li> </ul>	<ul style="list-style-type: none"> <li>– рабочий прототип системы, соответствующий требованиям;</li> <li>– рабочий сетевой полигон;</li> <li>– демо версии системы для презентации заказчику;</li> <li>– презентации;</li> </ul>	<p>начало: 21.11.2022</p> <p>окончание: 15.04.2023</p>
6	Сдача системы заказчику	<ul style="list-style-type: none"> <li>– подготовка системы для внедрения в сеть заказчика;</li> <li>– подготовка полной документации системы;</li> <li>– поддержка системы.</li> </ul>	<ul style="list-style-type: none"> <li>– система сдана заказчику в эксплуатацию;</li> <li>– полная документация системы.</li> </ul>	<p>начало: 16.04.2023</p> <p>окончание: 30.06.2023</p>

### 3.2 Фактически выполненные работы и полученные результаты

В таблице 3.2 представлена информации о фактическом выполнении работ и в таблице 3.3 представлена информации о фактически полученных результатах. Данные представлены на 14.12.2022.

Таблица 3.2 – Фактическое выполнение работ на 14.12.2022

Этап	Выполнено	Не выполнено
1	– анализ БП в компании; – описание проблемы в БП; – составление требований; – постановка задачи; – составление отчетности и документации.	Все планируемые работы завершены.
2	– изучение сферы автоматизации управления сетевым оборудованием; – поиск готового решения; – сравнение решений; – подведение результатов сравнения; составление отчетности и документации.	Все планируемые работы завершены.
3	– обоснование выбранного решения; – детальное изучение решения; – описание предлагаемого решения; составление отчетности и документации.	Все планируемые работы завершены.
4	– составление БП при использовании системы; – составление вариантов использования системы; – составление описания программ автоматизации; составление отчетности и документации.	Все планируемые работы завершены.
5	– установка и настройка ПО; – создание сетевого полигона; – подготовка устройств к работе с системой; – создание инвентаря;	– создание программ автоматизации; – отладка программ автоматизации; – составление отчетности и документации; – подготовка демо версий для презентации заказчику; – подготовка презентаций; – презентация наработок заказчику.
6	Работы ещё не начаты	– подготовка системы для внедрения в сеть заказчика; – подготовка полной документации системы; поддержка системы.

Таблица 3.3 – Фактически полученные результаты на 14.12.2022

Этап	Получено	Не получено
1	– описание БП на текущий момент в компании; – требования к системе; – цель работы; – задача работы.	Все планируемые результаты получены.
2	– знание о сфере автоматизации управления сетевым оборудованием; – список готовых решений; – анализ готовых решений; сравнение готовых решений по различным критерия.	Все планируемые результаты получены.
3	– выбранное решение; – обоснование выбора решения; детализированный анализ выбранного решения.	Все планируемые результаты получены.
4	– описание БП после внедрения системы; – описание вариантов использования системы; описание программ автоматизации.	Все планируемые результаты получены.
5	– первая демо версия системы для презентации заказчику; – первая презентация.	– рабочий прототип системы, соответствующий требованиям; – рабочий сетевой полигон; – вторая демо версия системы для презентации заказчику; – вторая презентация.
6	– система сдана заказчику в эксплуатацию; полная документация системы.	– система сдана заказчику в эксплуатацию; – полная документация системы.

### 3.3 Примеры наработки системы

В данном разделе представлены некоторые наработки системы, полученные на 14.12.2023.

#### 3.3.1 Структура файлов Ansible

На рисунке 3.1 представлена структура файлов Ansible. Файлы отображены белым, а папки желтым.

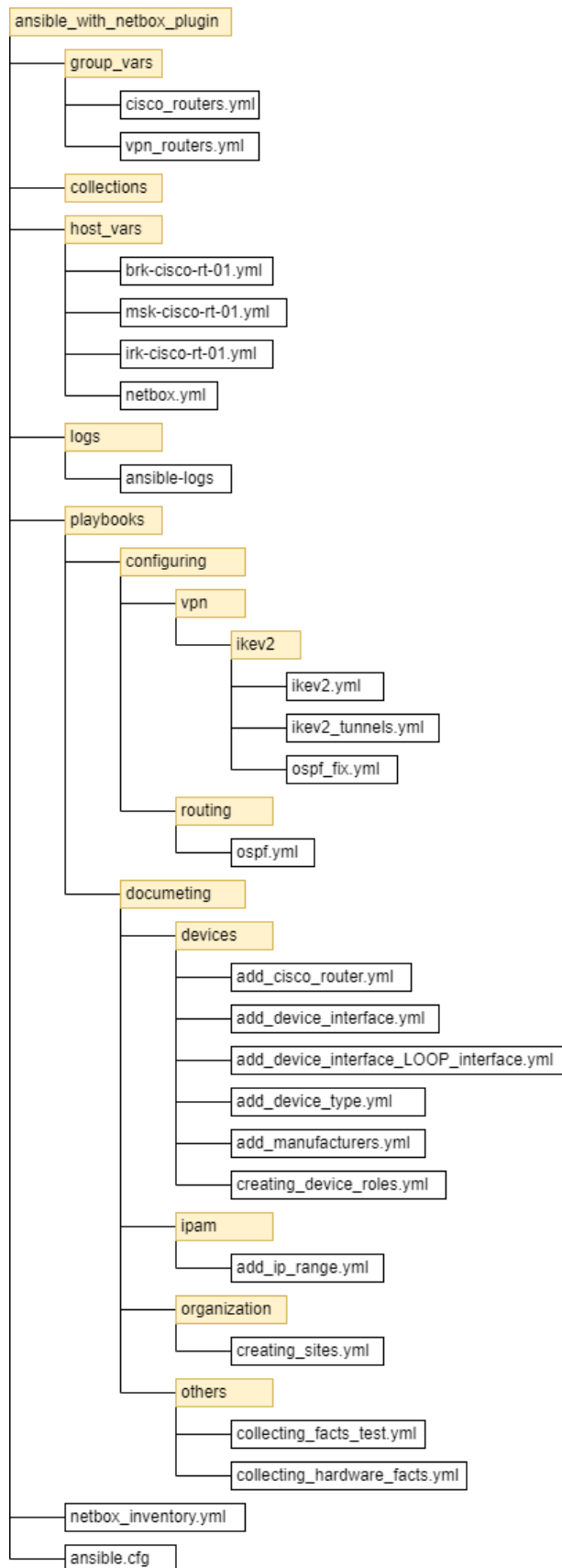


Рисунок 3.1 – Структура файлов Ansible.

group\_vars. Папка хранит переменные для групп устройств. Файл cisco-routers.yml хранит переменные для всех маршрутизаторов cisco. Файл vpn\_routers.yml хранит переменные для всех маршрутизаторов, которые выполняют функцию шифрования данных посредством туннелей.

collections. Папка с коллекциями. На данный момент пуста, из-за отсутствия необходимости в использовании коллекций.

host\_vars. Папка хранит переменные для конкретного устройства. Файлы brk-cisco-rt-01.yml, irk-cisco-rt-01.yml, msk-cisco-rt-01.yml хранят переменные для главных маршрутизаторов различных офисов. Файл netbox.yml хранит информацию о подключении к системе netbox.

logs. Папка, хранящая лог-файлы ПО Ansible. Файл ansible-logs хранит информацию о действиях при работе с ПО Ansible.

playbooks. Папка, хранящая playbook (программы автоматизации). Структура данной папки, следующая:

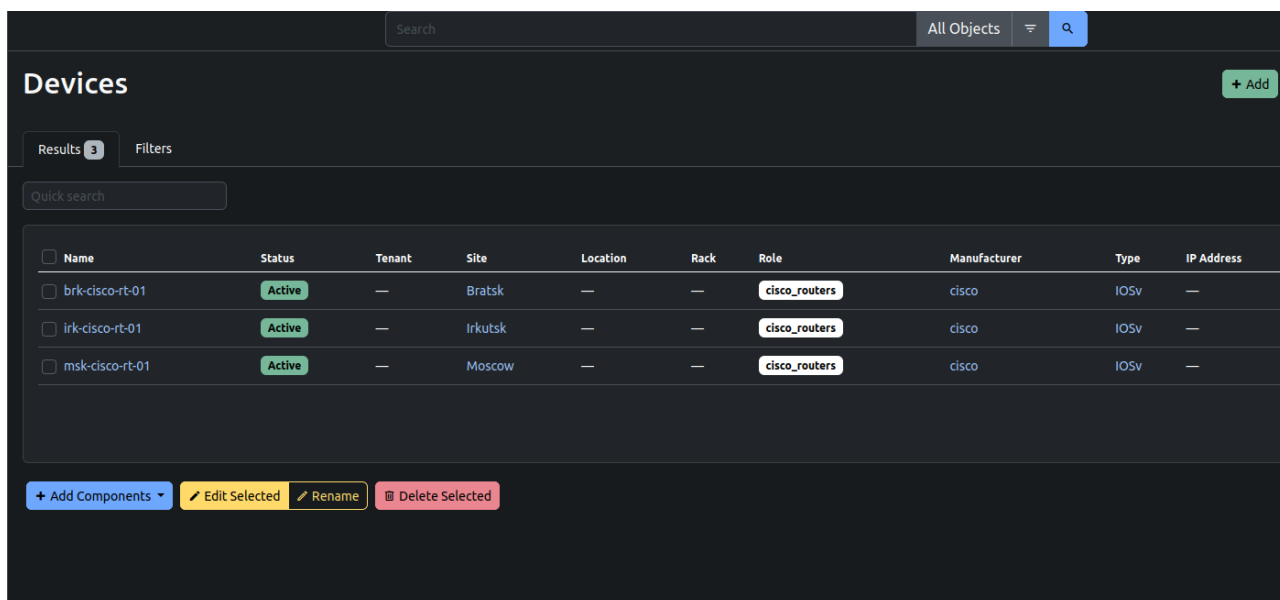
1. Папка configuring. Хранит playbook для задач конфигурации устройств:
  - 1.1. Папка vpn. Хранит playbook для конфигурирования vpn-туннелей на устройствах. На данный момент конфигурируется только протокол IKEv2:
    - 1.1.1. ikev2.yml. Программа автоматизации настройки протокола IKEv2;
    - 1.1.2. ikev2\_tunnels.yml. Программа автоматизации создания туннелей для протокола IKEv2;
    - 1.1.3. ospf\_fix.yml. Программа автоматизации, которая добавляет изменения в протокол OSPF;
  - 1.2. Папка routing. Хранит playbook для конфигурирования маршрутизации на устройствах. На данный момент конфигурируется только протокол OSPF:
    - 1.2.1. Файл ospf.yml. Программа автоматизации для настройки протокола OSPF на устройствах;
2. Папка documenting. Хранит playbook для задач документирования устройств:
  - 2.1. Папка devices. Хранит playbook для документирования информации, касательно устройства:
    - 2.1.1. Файл add\_cisco\_router.yml. Документирует маршрутизатор cisco в систему NetBox;
    - 2.1.2. Файл add\_device\_interface.yml. Документирует интерфейсы устройства в NetBox. Работает только при наличии записи устройства;
    - 2.1.3. Файл add\_device\_interface\_LOOP\_interface.yml. Документирует LOOP интерфейсы устройства в NetBox. Работает только при наличии записи устройства;
    - 2.1.4. Файл add\_device\_type.yml. Документирует информацию о модели устройства в NetBox;
    - 2.1.5. Файл add\_manufacturers.yml. Документирует информацию о производителе устройства;
    - 2.1.6. Файл creating\_device\_roles.yml. Создает роль устройства в NetBox;
  - 2.2. Папка ipam. Хранит playbook для документирования информации, касательно сетевого пространства:
    - 2.2.1. Файл add\_ip\_range.yml. Документирует диапазон адресов в NetBox;
  - 2.3. Папка organization. Хранит playbook для документирования информации касательно организации;

- 2.3.1. Файл `creating_sites.yml`. Документирует офисы в NetBox;
- 2.4. Папка `Others`. Хранит `playbook`, которые ещё не были вынесены в отдельную категорию:
- 2.4.1. Файл `collecting_facts_test.yml`. Собирает информацию с устройства и предоставляет пользователю в сыром виде (в терминале);
- 2.4.2. Файл `collecting_hardware_facts.yml`. Собирает только техническую информацию с устройства и предоставляет пользователю в сыром виде (в терминале).
- `netbox_inventory.yml`. Файл, описывающий использование NetBox как инвентаря Ansible.
- `ansible.cfg`. Конфигурационный файл Ansible.

### 3.3.2 Результаты документирования устройств

В данном разделе приводятся результаты, полученные при автоматизировании документирования устройства в систему NetBox.

На рисунке 3.2 представлены задокументированные маршрутизаторы `cisco`.

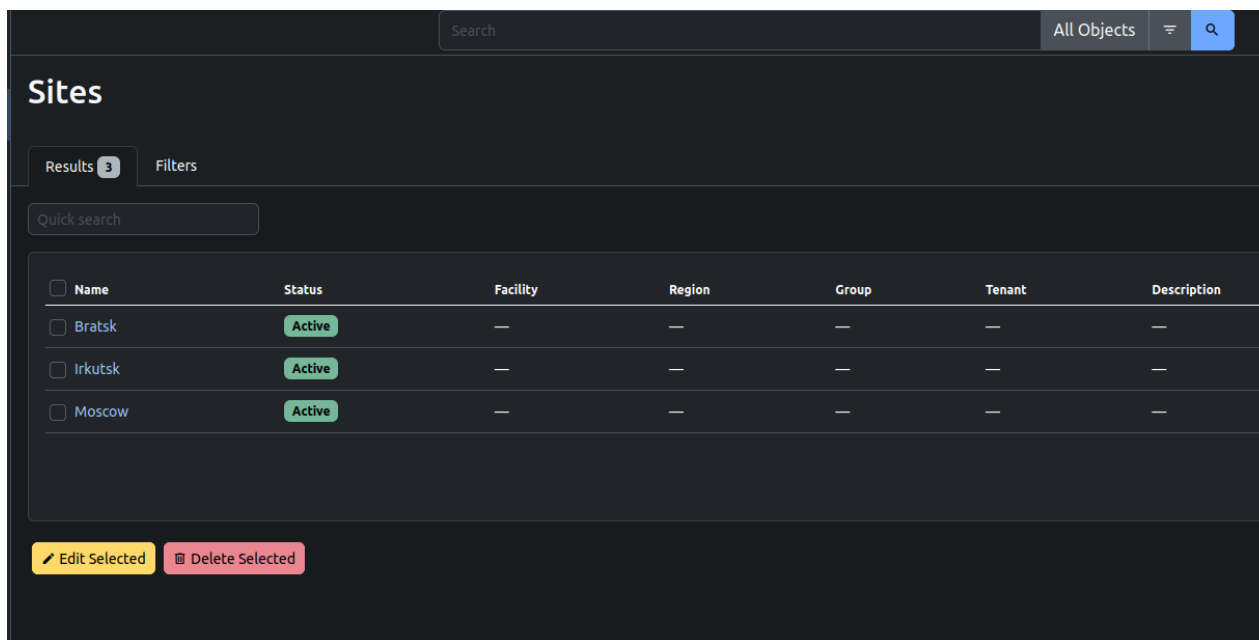


Name	Status	Tenant	Site	Location	Rack	Role	Manufacturer	Type	IP Address
brk-cisco-rt-01	Active	—	Bratsk	—	—	cisco_routers	cisco	IOSv	—
lrk-cisco-rt-01	Active	—	Irkutsk	—	—	cisco_routers	cisco	IOSv	—
msk-cisco-rt-01	Active	—	Moscow	—	—	cisco_routers	cisco	IOSv	—

Рисунок 3.2 – Записи о маршрутизаторах в NetBox.



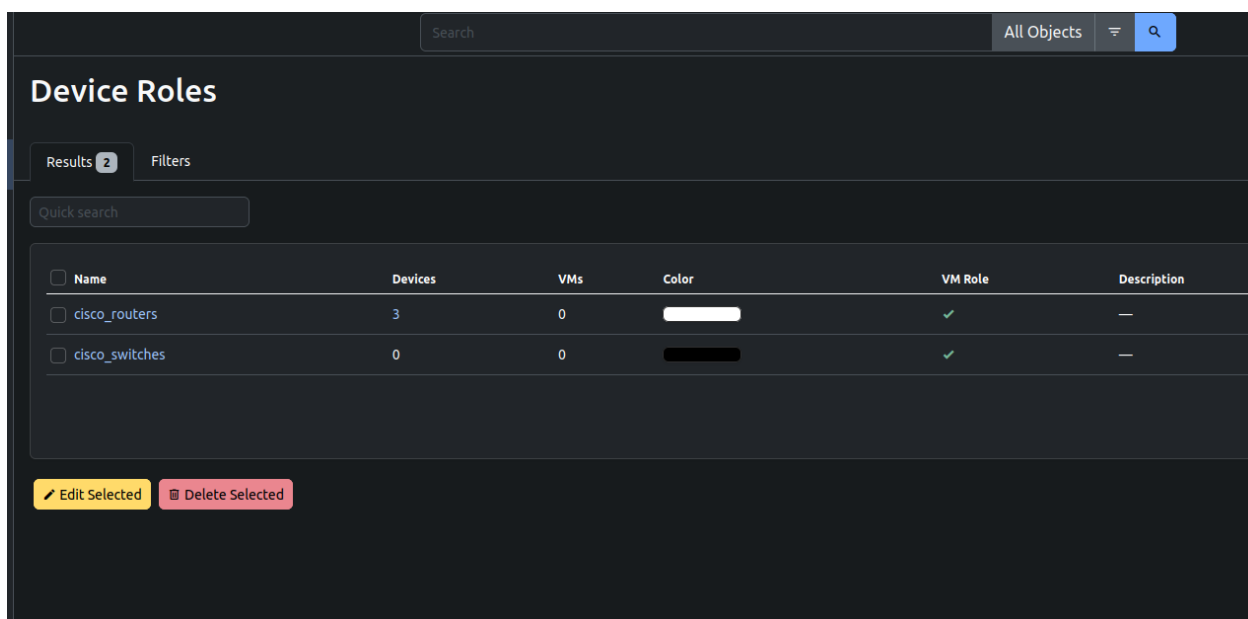
На рисунке 3.3 представлена информация о задокументированных офисах в систему NetBox.

The screenshot shows the 'Sites' page in NetBox. At the top, there is a search bar and a 'Search' button. Below the search bar, there are tabs for 'All Objects', a filter icon, and a search icon. The main heading is 'Sites'. Below this, there are 'Results 3' and 'Filters' tabs. A 'Quick search' input field is present. The table has columns: Name, Status, Facility, Region, Group, Tenant, and Description. There are three rows: 'Bratsk' with status 'Active', 'Irkutsk' with status 'Active', and 'Moscow' with status 'Active'. At the bottom, there are buttons for 'Edit Selected' and 'Delete Selected'.

<input type="checkbox"/>	Name	Status	Facility	Region	Group	Tenant	Description
<input type="checkbox"/>	Bratsk	Active	—	—	—	—	—
<input type="checkbox"/>	Irkutsk	Active	—	—	—	—	—
<input type="checkbox"/>	Moscow	Active	—	—	—	—	—

Рисунок 3.3 – Записи об офисах в NetBox.

На рисунке 3.4 представлена информация о задокументированных ролях устройств в систему NetBox.

The screenshot shows the 'Device Roles' page in NetBox. At the top, there is a search bar and a 'Search' button. Below the search bar, there are tabs for 'All Objects', a filter icon, and a search icon. The main heading is 'Device Roles'. Below this, there are 'Results 2' and 'Filters' tabs. A 'Quick search' input field is present. The table has columns: Name, Devices, VMs, Color, VM Role, and Description. There are two rows: 'cisco\_routers' with 3 devices and 'cisco\_switches' with 0 devices. Both have a 'VM Role' of '✓'. At the bottom, there are buttons for 'Edit Selected' and 'Delete Selected'.

<input type="checkbox"/>	Name	Devices	VMs	Color	VM Role	Description
<input type="checkbox"/>	cisco_routers	3	0		✓	—
<input type="checkbox"/>	cisco_switches	0	0		✓	—

Рисунок 3.4 – Записи о ролях устройств в NetBox.

### **3.4 Анализ выполненной работы и дальнейшие планы**

На данный момент можно считать, что разработка системы идёт успешно, согласно плану работ. Текущий этап работ – пятый, посвящен непосредственной реализации системы и автоматизации. Полученные на текущий момент результаты были представлены заказчику. Представитель сетевого отдела для разработки данной системы оказался доволен результатом, ожидает дальнейшего прогресса разработки и последующих результатов.

Следующее представление работы заказчику состоится в марте 2023 года. К этому моменту планируется расширить возможности автоматизации при работе с сетевым оборудованием. Также в планах:

- расширить разнообразие оборудования, работу с которым можно автоматизировать;
- доработать документирование устройства;
- начать работу над написанием руководства пользования системой;
- плотное сотрудничество с представителем компании, для автоматизации наиболее правдоподобных задач при работе с сетевым оборудованием.

## Заключение

Во время прохождения производственной практики были проанализированы бизнес-процессы сетевого отдела компании EN+ DIGITAL. Рутинность задач конфигурирование и документирование привела к идее об автоматизации этих процессов посредством внедрения автоматизированной системы.

Перед проектированием системы были проведены следующие операции:

- смоделированы бизнес-процессы при работе с сетевым оборудованием;
- составлены цель и назначение системы;
- составлены требования.

Изучение предметной области, посвященной автоматизации управления и работы с сетевым оборудованием, дало не только знания об области, но и список готовых решений.

Готовые решения были изучены. Были выявлены их слабые и сильные стороны, которые позволили выполнить общий сравнительный анализ. Итогом сравнительного анализа стало выбранное решение в виде ПО Ansible и NetBox. Данное ПО не только соответствует всем требованиям, но и имеет подробную документацию, а также является доступным.

После того как решение было утверждено, начался процесс проектирования системы, а именно:

- моделирование бизнес-процессов при работе с сетевым оборудованием после внедрения системы;
- составлены варианты использования системы;
- описаны основные программы автоматизации.

Когда проектирование было завершено начался процесс реализации прототипа системы:

- установка и настройка необходимого ПО;
- подготовка сетевого полигона и устройств;
- работа с автоматизацией задач документирования и конфигурирования.

Реализация системы ещё далека от завершения, но даже сейчас может показывать достойные результаты. Там, где раньше сетевой инженер мог потратить 4 часа на настройку VPN-туннелей, работа с системой сокращает это время до 0.5-1 часа. Большая часть времени при работе с системой уходит на написание программ автоматизации и объявление переменных.

Когда работа над системой будет закончена, начнется процесс составления полной документации, внедрение системы и последующая поддержка.

## Список использованных источников

1. Мария Панасюгина. 2009 VS 2019: КАКИМ БЫЛ ИНТЕРНЕТ 10 ЛЕТ НАЗАД [Электронный ресурс]. URL: <https://morsmagazine.ru/events/fotograf-inzhener-v-sfere-it-i-zhurnalisty-rasskazyvayut-kakim-byl-internet-10-let-nazad/> (Дата обращения 03.10.2022).
2. АДСМ5. История сетевой автоматизации [Электронный ресурс]. URL: <https://habr.com/ru/post/667012/#NOW> (Дата обращения 10.10.2022).
3. О.А. Бияков Графические нотации в стандарте IDEF0 [Электронный ресурс]. URL: <https://studfile.net/preview/416831/> (Дата обращения 17.11.2022)
4. Универсальная система мониторинга Zabbix – введение [Электронный ресурс]. URL: <https://habr.com/ru/post/73338/> (Дата обращения 20.11.2022)
5. Progress WhatsUp Gold [Электронный ресурс]. URL: <https://www.whatsupgold.com/> (Дата обращения 20.11.2022).
6. ManageEngine NetFlow Analyzer [Электронный ресурс]. URL: <https://www.manageengine.ru/netflow/> (Дата обращения 20.11.2022).
7. David Taylor. Best 8 Ansible Alternatives & equivalent in 2023 [Электронный ресурс]. URL: <https://www.guru99.com/ansible-alternative.html> (Дата обращения 17.11.2022).
8. Альтернативы Ansible [Электронный ресурс]. URL: <https://progsoft.net/ru/software/ansible> (Дата обращения 17.11.2022).
9. The 41 Best Network Automation Tools in 2021 [Электронный ресурс]. URL: <https://www.imedita.com/blog/network-automation-tools/> (Дата обращения 18.11.2022).
10. Welcome to NAPALM's documentation! [Электронный ресурс]. URL: <https://napalm.readthedocs.io/en/latest/> (Дата обращения 15.10.2022).
11. Модуль netmiko [Электронный ресурс]. URL: [https://pyneng.readthedocs.io/ru/latest/book/18\\_ssh\\_telnet/netmiko.html](https://pyneng.readthedocs.io/ru/latest/book/18_ssh_telnet/netmiko.html) (Дата обращения 16.10.2022).
12. Ansible Documentation [Электронный ресурс]. URL: <https://docs.ansible.com/ansible/latest/index.html> (Дата обращения 20.10.2022).
13. Tonya Brown. An introduction to Ansible Tower [Электронный ресурс]. URL: <https://www.redhat.com/sysadmin/intro-ansible-tower> (Дата обращения 23.10.2022).
14. NetYce Full Stack Network Automation Platform [Электронный ресурс]. URL: <https://www.tucana.com/products/netyce-full-stack-network-automation-platform/> (Дата обращения 27.10.2022).
15. Ansible-Galaxy [Электронный ресурс]. URL: <https://galaxy.ansible.com/> (Дата обращения 01.11.2022).
16. NetBox documentation [Электронный ресурс]. URL: <https://netbox.arbormesh.com/static/docs/> (Дата обращения 05.11.2022).
17. Django-стек: работа, установка и настройка на одном сервере [Электронный ресурс]. URL: [https://1cloud.ru/blog/django\\_one\\_server](https://1cloud.ru/blog/django_one_server) (Дата обращения 11.11.2022).
18. Getting Started with GNS3 [Электронный ресурс]. URL: <https://docs.gns3.com/docs/> (Дата обращения 13.11.2022).

19.Использование диаграммы вариантов использования UML при проектировании программного обеспечения [Электронный ресурс]. URL: <https://habr.com/ru/post/566218/> (Дата обращения 01.12.2022).

20.netbox.netbox.nb\_inventory inventory – NetBox inventory source [Электронный ресурс]. URL: [https://docs.ansible.com/ansible/latest/collections/netbox/netbox/nb\\_inventory\\_inventory.html](https://docs.ansible.com/ansible/latest/collections/netbox/netbox/nb_inventory_inventory.html) (Дата обращения 05.12.2022).

21.Основы компьютерных сетей. Тема №6. Понятие VLAN, Trunk и протоколы VTP и DTP [Электронный ресурс]. URL: <https://habr.com/ru/post/319080/> (Дата обращения 05.12.2022).

22.Руслан Маммадов. OSPF (часть первая) [Электронный ресурс]. URL: <https://habr.com/ru/post/418391/> (Дата обращения 05.12.2022).

23.Сети для самых маленьких. Часть пятая. ACL и NAT [Электронный ресурс]. URL: <https://habr.com/ru/post/147996/> (Дата обращения 05.12.2022).

24.VPN: ещё раз просто о сложном [Электронный ресурс]. URL: <https://habr.com/ru/post/534250/> (Дата обращения 05.12.2022).

25.IKEv2 и Flex VPN средствами Cisco IOS. Синтаксис и логика работы [Электронный ресурс]. URL: <https://habr.com/ru/post/186126/> (Дата обращения 05.12.2022).

26.Как взять сетевую инфраструктуру под свой контроль. Глава вторая. Чистка и документирование. [Электронный ресурс]. URL: <https://habr.com/ru/post/434750/> (Дата обращения 07.12.2022).

27.Why no Ansible controller for Windows? [Электронный ресурс]. URL: <http://blog.rolpdog.com/2020/03/why-no-ansible-controller-for-windows.html> (Дата обращения 10.12.2022).

28.Oracle VM VirtualBox [Электронный ресурс]. URL: <https://www.oracle.com/virtualization/virtualbox/> (Дата обращения 08.12.2022).

29.UML Deployment Diagrams [Электронный ресурс]. URL: <https://www.uml-diagrams.org/deployment-diagrams.html> (Дата обращения 09.12.2022).