

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ
Школа профессионального и академического образования

ДОПУСТИТЬ К ЗАЩИТЕ ПЕРЕД ГЭК


(подпись) _____
« 29 » мая 2023 г.

_____ Обабков И.Н. _____
(Ф.И.О.)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Разработка ITSM-системы для компании, предоставляющей услуги по
предотвращению магазинных краж

Научный руководитель: Кошелев А.А.
к.ф.-м.н., доцент
Нормоконтролер: Огуренко Е.В.
Студент группы РИМ-210990 Гафина Ю.А.





Екатеринбург
2023

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего
образования
«Уральский федеральный университет имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РтФ
Школа профессионального и академического образования
Направление подготовки 09.04.04-Программная инженерия
Образовательная программа 09.04.04/33.02 - Разработка и управление в программных проектах

УТВЕРЖДАЮ
РОП
«29» мая 2023 г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

студента Гафиной Юлии Андреевны группы РИМ-210990
(фамилия, имя, отчество)

1. **Тема выпускной квалификационной работы** Разработка ITSM-системы для компании, предоставляющей услуги по предотвращению магазинных краж

Утверждена распоряжением по институту от « » 2023 г. №

2. **Руководитель** Кошелев Антон Александрович, к.ф.-м.н., доцент
(Ф.И.О., должность, ученое звание, ученая степень)

3. **Исходные данные к работе** литература по теме работы, требования к программному обеспечению, результаты производственной (НИР) и преддипломной практик

4. **Перечень демонстрационных материалов** презентация, содержащая постановку задачи, схему системы, результаты работы, анализ результатов

5. Календарный план

№ п/п	Наименование этапов выполнения работы	Срок выполнения этапов работы	Отметка о выполнении
1.	<i>Анализ предметной области</i>	до 15.04.2023 г.	<i>Ков</i>
2.	<i>Описание разрабатываемой системы</i>	до 20.05.2023 г.	<i>Ков</i>
3.	<i>Оформление записки и презентации</i>	до 26.05.2023 г.	<i>Ков</i>

Руководитель *Ков*
(подпись)

Кошелев А.А.
Ф.И.О.

Задание принял к исполнению Ю.А. Гафина
дата

Ков
(подпись)

6. **Выпускная квалификационная работа** закончена «27» мая 2023г. считаю возможным допустить Гафину Юлию Андреевну к защите выпускной квалификационной работы закончена в Государственной экзаменационной комиссии.

Руководитель *Ков*
(подпись)

Кошелев А.А.
Ф.И.О.

7. **Допустить** Гафину Юлию Андреевну к защите магистерской диссертации в Государственной экзаменационной комиссии.

РОП *Обабков И.Н.*
(подпись)

Обабков И.Н.
Ф.И.О.

РЕФЕРАТ

Выпускная квалификационная работа магистра: 74 с., 15 рис., 3 табл., 32 источника, 7 прил.

ITSM, ITIL, МЕТРИКИ ЭФФЕКТИВНОСТИ РАБОТЫ, CFD, СИСТЕМА УПРАВЛЕНИЯ ИТ-УСЛУГАМИ, КАНБАН-ДОСКА, ЕДИНОЕ ОКНО

Цель работы – разработка, реализация и внедрение системы управления ИТ-услугами, позволяющей повысить качество и скорость работы технического отдела компании.

Объект исследования – класс ITSM-систем, позволяющих управлять работой ИТ-отдела компании.

Методы исследования:

- анализ и сравнение существующих систем управления ИТ-услугами;
- анализ современных средств разработки WEB-приложений;
- тестирование разрабатываемой системы на реальных пользователях и реальных данных.

Результаты работы: разработана система управления ИТ-услугами (ITSM) для компании по предотвращению магазинных краж. Доказана эффективность внедрения новой системы.

Выпускная квалификационная работа выполнена в текстовом редакторе Microsoft Word и представлена в твёрдой копии.

СОДЕРЖАНИЕ

Реферат.....	3
Содержание.....	4
ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ.....	7
ВВЕДЕНИЕ.....	8
1 Теоретический обзор ITSM-систем.....	11
1.1 Стандарты разработки и организации ITSM-систем	11
1.2 Аспекты и принципы ITIL при разработке ITSM-систем.....	15
1.3 Метрики для анализа эффективности работы сотрудников в ITSM-системах	19
1.3.1 Канбан-доска	19
1.3.2 Lead Time	20
1.3.3 Cycle Time	21
1.3.4 Throughput.....	22
1.3.5 Cumulative Flow Diagram.....	22
1.3.6 Cycle Time Scatter Plot	23
1.4 Монолитный и микросервисный подходы в разработке сложных программных систем.....	24
1.5 Принцип «единого окна»	29
1.6 Выводы.....	30
2 Разработка ITSM-системы для компании, предоставляющей услуги по видеофиксации магазинных краж	31
2.1 Постановка задачи	31
2.2 Анализ подхода технического отдела к работе до внедрения ITSM-системы и определение потребностей по улучшению качества его работы	31

2.3 Анализ существующих решений ITSM-систем.....	33
2.3.1 ServiceNow.....	33
2.3.2 BMC Remedy	33
2.3.3 Jira Service Management.....	34
2.3.4 ITSM365	35
2.3.5 SimpleOne ITSM	36
2.3.6 Naumen Service Desk.....	36
2.3.7 Сравнение систем.....	37
2.4 Преимущества внедрения собственной системы	38
2.5 Выбора средств разработки	39
2.5.1 Angular.....	39
2.5.2 ASP.NET.....	39
2.5.3 PostgreSQL	40
2.5.4 MongoDB.....	40
2.5.5 Elasticsearch.....	41
2.5.6 Vault.....	41
2.5.7 RabbitMQ.....	41
2.5.8 GitLab CI/CD.....	42
2.5.9 Docker	42
2.6 Разработка и внедрение ITSM-системы	43
2.6.1 Микросервис диагностики	44
2.6.2 Микросервис статистики и отчетов	45
2.6.3 Микросервис файлов конфигураций	46
2.6.4 Микросервис кадров с камер	46
2.6.5 Микросервис ТС, дивизионов и магазинов.....	47

2.6.6 Микросервис тикетов	48
2.6.7 Микросервис аутентификации и авторизации.....	49
2.6.8 WebApp и API Gateway	49
2.6.9 CI/CD	50
2.7 Анализ показателей скорости работы технического отдела при старом и новом подходах	51
2.7.1 Сравнение всех задач за равный период	51
2.7.2 Сравнение задач с типом Установка магазина	52
2.7.3 Сравнение задач с типом Переустановка магазина.....	53
2.7.4 Сравнение задач с типом Проверка магазинов.....	54
2.7.5 Выводы.....	55
Заключение	56
Библиографический список	57
Приложение А	60
Приложение Б.....	61
Приложение В	64
Приложение Г	65
Приложение Д	66
Приложение Е.....	71
Приложение Ж.....	73

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

AMQP	Advanced Message Queuing Protocol, открытый протокол прикладного уровня для передачи сообщений между сервисами
API	Application Programming Interface, программный интерфейс, позволяющий связывать приложения между собой
CFD	Cumulative Flow Diagram, накопительная диаграмма потока
CI/CD	Continues Integration / Continues Delivery, непрерывная интеграция и непрерывное развёртывание
Cron	Программа, выполняющаяся с определенной периодичностью
CycleTime	Время, потраченное на задачу, от взятия ее в работу, до передачи на тестирование
ITSM	IT Service Management, система управления ИТ-услугами
LeadTime	Время, потраченное на выполнение задачи, от ее создания до полной готовности
Throughput	Пропускная способность чего-либо (в данной работе, пропускная способность сотрудников при выполнении задач)
Дашборд	Визуальный набор информации о каком-то объекте
Канбан-доска	Виртуальная доска с задачами (тикетами)
Пайплайн	Список инструкций для выполнения
ТС	Торговая сеть
Тикет	Задача на канбан-доске

ВВЕДЕНИЕ

Актуальность темы исследования. В современном мире информационные технологии играют ключевую роль в развитии бизнеса. Однако, управление предоставлением ИТ-услуг может стать сложной задачей, особенно в условиях быстрого развития технологий и увеличения объемов работы. Для решения этой проблемы может быть использована ITSM-система, которая позволяет эффективно управлять ИТ-инфраструктурой и минимизировать время простоя систем.

С помощью ITSM-системы можно управлять всеми процессами, связанными с технической поддержкой, обслуживанием и управлением информационными технологиями. Она позволяет эффективно контролировать процессы, уменьшить количество ошибок и проблем, связанных с ИТ-инфраструктурой, а также оценить эффективность работы отдела ИТ. Кроме этого, такая система представляет собой так называемое «единое окно», через которое происходит вся работа, мониторинг, предоставление услуг, контроль качества и т.д., что позволяет уменьшить количество возможных ошибок и сохранить контекст работы без переключения внимания на другие сервисы.

В текущих реалиях ITSM система становится жизненно необходимой для управления ИТ-инфраструктурой, так как в условиях высоких требований к качеству и скорости работы важно иметь возможность мгновенно определять и устранять проблемы. Она позволяет сократить время разрешения инцидентов и проблем, повысить удовлетворенность клиентов и увеличить производительность работы.

Данный инструмент является необходимым для каждой компании, предоставляющей ИТ-услуги, и может оказать значительную помощь в управлении ИТ-инфраструктурой.

Анализ актуальности обусловил выбор темы исследования: «Разработка ITSM-системы для компании, предоставляющей услуги по предотвращению магазинных краж».

Гипотеза разработки состоит в том, что использование системы управления IT-услугами (ITSM) повысит скорость реакции IT-отдела компании на проблемы клиентов, позволяя выполнять большее количество задач различных типов за меньшее время.

Целью исследования является разработка, реализация и внедрение системы управления IT-услугами, позволяющей повысить качество и скорость работы технического отдела компании.

Для достижения поставленной цели необходимо решить следующие задачи:

- анализ текущих подходов к работе технического отдела и определение потребностей по улучшению качества его работы;
- анализ существующих решений систем управления IT-услугами;
- составление схемы разрабатываемой системы и обоснование преимуществ её внедрения по сравнению с другими системами;
- выбор средств разработки в соответствие с современными технологиями;
- разработка и внедрение системы управления IT-услугами;
- тестирование системы в реальной работе и анализ показателей эффективности работы сотрудников.

Объектом исследования является класс ITSM-систем, позволяющих управлять работой IT-отдела компании.

Предметом исследования является разработка WEB-приложения, представляющего собой ITSM-систему для компании, которая занимается предоставлением услуги по настройке технического и программного обеспечения, позволяющего проводить видеофиксацию магазинных краж.

Методы исследования включают в себя:

- анализ и сравнение существующих систем управления IT-услугами;
- анализ современных средств разработки WEB-приложений;
- тестирование разрабатываемой системы на реальных пользователях и реальных данных.

Теоретической основой исследования стали:

– зарубежные исследования и решения в области разработки систем управления ИТ-услугами (Jira, Trello, YouTrack и т. д.);

– различная документация по технологиям разработки WEB-приложений;

– ITIL (Information Technology Infrastructure Library – библиотека инфраструктуры информационных технологий) – фреймворк, который описывает лучшие практики для проектирования, разработки, доставки и управления ИТ-услугами, являющийся международным стандартом;

– статьи/книги, основанные на исследованиях в области микросервисных архитектур.

Теоретическая значимость работы по разработке системы управления ИТ-услугами заключается в том, что это позволяет усовершенствовать теоретические знания в области управления ИТ-услугами, разработки и внедрения систем управления, обеспечения качества и процессов управления изменениями. Работа позволяет получить более подробное представление о структурах и процессах управления ИТ-услугами, а также дает возможность оценить эффективность различных методов и технологий в этой области.

Практическая значимость заключается в том, что разработка системы управления ИТ-услугами является основой для эффективного управления ИТ-инфраструктурой внутри организации. Такая система позволяет организации эффективно управлять затратами и ресурсами, повышать качество ИТ-услуг, уменьшать риски и улучшать удовлетворенность клиентов, сокращая время решения возникающих проблем.

База исследования: информация о техническом обеспечении клиентов компании ООО «Компания БИТ».

1 Теоретический обзор ITSM-систем

ITSM (IT Service Management) является системой управления услугами в области информационных технологий, которая охватывает широкий спектр задач и принципов для обеспечения надежного и эффективного предоставления ИТ-услуг организацией. ITSM включает в себя управление проектами, конфигурациями, инцидентами, изменениями, проблемами, конфиденциальностью и многое другое.

Практика использования ITSM-систем позволяет обеспечивать более высокий уровень качества предоставления ИТ-услуг за счет:

- сокращения времени на реакцию на инциденты и проблемы;
- лучшей отчетности и мониторингу производительности;
- улучшения коммуникации между ИТ-отделом и бизнесом;
- улучшения обслуживания пользователей;
- увеличения контроля над рабочим процессом.

Для определения эффективности предоставления ИТ-услуг сотрудниками компании используются различные метрики, позволяющие выявить слабые места в процессе оказания услуг. С помощью ITSM-системы можно проконтролировать стадии выполнения возникающих инцидентов, а также оценить скорость и качество реагирования на них.

Ко всему прочему, такая большая и сложная система должна быть разработана определенным образом. Для выбора способа разработки будут рассмотрены такие подходы к организации программного продукта, как монолитная и микросервисная архитектуры.

1.1 Стандарты разработки и организации ITSM-систем

Использование стандартов в разработке ITSM-систем имеет несколько преимуществ. Во-первых, стандарты обеспечивают единообразие подхода к управлению ИТ-услугами, что позволяет организациям лучше понимать и контролировать свои процессы. Во-вторых, стандарты предоставляют проверенные и оптимизированные методы, инструменты, метрики и ключевые

показатели эффективности (KPI), которые помогают организациям измерять, контролировать и улучшать эффективность своих ИТ-отделов. В-третьих, стандарты помогают организациям соответствовать требованиям законодательства и стандартам отрасли, что является важным фактором для обеспечения доверия клиентов и партнеров.

Существует три ключевых стандарта (или набора рекомендаций) в разработке и организации ITSM-систем.

1. ITIL (Information Technology Infrastructure Library, библиотека инфраструктуры информационных технологий) – это набор практик и рабочих процессов, предназначенных для управления ИТ-инфраструктурой в организации, который разработан Британским институтом управления и технологии (British Institute of Management and Technology). В настоящее время библиотека принадлежит компании Axelos [18]. ITIL включает в себя подробные описания процессов, ролей, функций и метрик, которые помогают улучшать качество предоставления ИТ-услуг.

2. COBIT (Control Objectives for Information and Related Technology, цели контроля для информации и связанных с ней технологий) – это фреймворк, предназначенный для управления ИТ-ресурсами в организации и связанной с ней информацией, который разработан Ассоциацией ISACA (Information Systems Audit and Control Association) [19]. COBIT ориентирован на обеспечение соответствия требованиям законодательства и регуляторных органов, на минимизацию рисков для бизнеса, а также согласование с бизнес-целями. Он представляет наиболее современные практики и методологии управления ИТ-рисками, описывает процессы управления изменениями, управления проектами и управления ИТ-безопасностью.

3. ISO 20000 (ГОСТ Р ИСО/МЭК 20000-1-2021) [1] – это международный стандарт, который определяет требования к управлению ИТ-услугами. Он также описывает передовые методологии управления, использование которых должно повысить качество и надежность предоставления ИТ-услуг в организации. Сертификация по ISO 20000

подразумевает стандартизацию процессов, повышение эффективности управления ИТ-ресурсами, увеличение качества обслуживания пользователей и обеспечивает оценку соответствия организаций этим требованиям.

Как правильно подметила Preethiga Narasimman в своей сравнительной статье об ITIL и ISO 20000 [11]: ITIL – это набор наилучших практик в разработке, тогда как ISO – это стандарт, поэтому говорить, что система соответствует ITIL, не имеет никакого смысла, тогда как соответствие стандартам можно явно проверить. Некоторые авторы считают, что ISO 20000 стремится к принципам ITIL при каждой своей новой редакции [3], в самом же стандарте (международной его версии) сказано, что данные подходы не базируются друг на друге, а их совместное использование при разработке ITSM-системы может помочь организациям во внедрении такой системы и получении от этого наибольших преимуществ [20].

Похожим образом сочетаются ITIL и COBIT. Тогда как COBIT сфокусирован больше на самом бизнес-процессе, оценке метрик и управлении рисками всего предприятия, ITIL в свою очередь сфокусирован на процессах и подходах к управлению в контексте предоставления ИТ-услуг. Другими словами, ITIL предоставляет подробное руководство по процессам предоставления ИТ-услуг, а COBIT предлагает общую модель процессов управления, которая может быть использована в сочетании с ITIL для установления контроля и метрик, связанных с процессами ITIL. Так же, как и ISO 20000, COBIT и ITIL могут взаимодополнять друг друга.

В таблице 1 представлено сравнение этих трёх подходов.

Таблица 1 – Сравнение подходов к системе управления ИТ-услугами

Аспект	ITIL	COBIT	ISO 20000
Описание	Набор рекомендаций и методологий для управления ИТ-услугами	Модель для управления и контроля ИТ-процессами	Международный стандарт для управления ИТ-услугами

Продолжение таблицы 1

Аспект	ITIL	COBIT	ISO 20000
Ориентация	Ориентирован на лучшие практики в области управления ИТ-услугами	Ориентирован на цели бизнеса и создание ценности через ИТ	Ориентирован на удовлетворение требований клиентов
Фокус	Управление ИТ-услугами и ИТ-инфраструктурой	Управление ИТ-процессами и целостностью данных	Достижение требований клиентов
Покрытие	Широкий спектр процессов и практик для управления жизненным циклом ИТ-услуг	Обширный набор процессов для управления ИТ-процессами бизнеса в целом	Определяет четких требования к управлению ИТ-услугами и сертификации организаций
Применимость	Широко используется в области управления ИТ-услугами и принято многими организациями	Используется для управления ИТ-процессами и обеспечения соответствия требованиям	Часто используется для сертификации организаций в области управления ИТ-услугами
Взаимодействие	Может быть использован совместно с COBIT для обеспечения более широкого управления ИТ	Может быть использован совместно с ITIL для улучшения управления ИТ-сервисами	Может быть использован совместно с ITIL и COBIT для более комплексного управления ИТ

Так как ITIL является наиболее распространенным набором практик по разработке ITSM-систем, то следует подробнее рассмотреть основные принципы, которые в нем описаны.

1.2 Аспекты и принципы ITIL при разработке ITSM-систем

Компания Axelos выпустила собственную книгу, в полной мере охватывающую все тонкости подхода, разработанного авторами [6].

Как сказано в книге, авторы выделяют четыре аспекта управления IT-услугами:

- организация и люди,
- информация и технологии,
- партнеры и поставщики,
- потоки и процессы создания ценности.

В теории, для достижения максимальной эффективности в работе организация должна учитывать все эти аспекты. На практике же компании могут упускать некоторые из них, либо же не уделять должного внимания.

Каждый аспект определяется несколькими факторами, кроме того, сами аспекты активно взаимодействуют друг с другом. Взаимоотношение аспектов представлено на рисунке 1. Рисунок является переводом аналогичного рисунка за авторством Axelos.



Рисунок 1 – Взаимоотношение аспектов управления услугами

Первый из аспектов: организации и люди. Эффективность организации в области управления услугами зависит не только от формально установленной структуры и системы власти, но также от наличия культуры, которая поддерживает цели организации, а также от уровня компетенции и потенциала сотрудников. Важно, чтобы руководители организации выступали в защиту и пропагандировали ценности, мотивирующие людей работать в желательных направлениях. Также отмечается, что важно обратить внимание на навыки и компетенции команд и отдельных сотрудников, а также на стили управления, коммуникацию и сотрудничество. Подчеркивается необходимость обновления навыков и компетенций вместе с эволюцией практик, а также понимание интерфейсов между специализациями и ролями в организации для обеспечения сотрудничества и координации. Каждый человек в организации должен иметь ясное представление о своем вкладе в создание ценности для организации, ее клиентов и заинтересованных сторон.

Следующий аспект – информация и технологии. Технологии, такие как системы управления рабочим процессом, базы знаний, аналитические инструменты и другие, поддерживают и оптимизируют процессы управления услугами. Они помогают автоматизировать и оптимизировать различные аспекты управления услугами, от стратегического планирования до системного мониторинга и поддержки пользователей. Важно учитывать конкретную информацию, создаваемую и используемую в процессе предоставления услуг, а также выбирать соответствующие технологии, которые обеспечивают ее защиту и эффективное управление. Технологические решения, такие как искусственный интеллект, машинное обучение, облачные решения и мобильные платформы, становятся все более распространенными среди поставщиков услуг. Все это помогает создавать ценность для организации и ее клиентов, а также решать проблемы информационного управления и соответствия нормативным требованиям и стандартам безопасности данных.

Третий аспект – партнеры и поставщики. Организации зависят от услуг, предоставляемых другими организациями, и эта зависимость должна быть учтена при разработке стратегии управления услугами. Взаимоотношения с партнерами и поставщиками включают контракты и другие соглашения, а уровень интеграции и формальности этих отношений может варьироваться. Организации могут выбирать различные подходы к использованию партнеров и поставщиков в соответствии с их стратегией, корпоративной культурой, ресурсами, стоимостными соображениями и другими факторами. Существует также метод управления интеграцией услуг, который помогает координировать отношения с партнерами и поставщиками. Важно разработать подходящую стратегию в использовании партнеров и поставщиков, чтобы обеспечить эффективное управление услугами.

Последний аспект – потоки создания ценности и процессы. Эффективное функционирование организации и достижение ее целей зависят от интеграции и координации различных частей организации в цепочку создания ценности. Путем идентификации и понимания потоков создания ценности, а также определения и оптимизации процессов, организация может повысить свою производительность и обеспечить эффективное предоставление продуктов и услуг. Ключевыми аспектами являются анализ текущего состояния, устранение ненужных действий и постоянное совершенствование потоков создания ценности для достижения поставленных целей.

Поставщики услуг не могут работать в изоляции, так что они подвержены влиянию множества внешних факторов. Эти факторы, такие как политические, экономические, социальные, технологические, правовые и экологические, оказывают влияние на работу поставщика услуг. Организации должны учитывать эти факторы и находить баланс между четырьмя измерениями управления услугами: ценности и результаты, связи и заинтересованные стороны, партнеры и поставщики, а также потоки создания ценности и процессы. Учитывая эти внешние факторы и работая в рамках всех

четырёх измерений, организации могут достичь оптимальной эффективности и создать продукты и услуги, соответствующие потребностям и ожиданиям всех заинтересованных сторон.

Помимо аспектов выделяют 7 ведущих принципов или рекомендаций, которыми следует руководствоваться при любых обстоятельствах в управлении процесса предоставления ИТ-услуг:

- Фокусируйтесь на ценности. Все действия организации должны быть направлены на создание ценности для заинтересованных сторон. Уделяйте внимание опыту клиентов и пользователей;

- Начините с того, что есть. Необходимо использовать уже имеющиеся ресурсы, услуги, процессы, программы и людей, прежде чем строить что-то новое. Изучение текущего состояния и понимание его важны для успешного достижения желаемых результатов;

- Продвигайтесь итеративно с обратной связью. Не пытайтесь сделать все сразу. Делайте работу поэтапно, разделяя ее на более управляемые части. Используйте обратную связь для оценки результатов на каждом этапе и корректировки действий, даже если обстоятельства меняются;

- Сотрудничайте и обеспечивайте видимость. Результаты достигаются лучше, когда люди работают вместе, преодолевая границы и сотрудничая. Для достижения целей необходимы обмен информацией, понимание и доверие. Важно, чтобы работа и последствия были видимыми, а скрытые интересы были исключены;

- Мыслите и работайте целостно. Ни одна услуга или элемент не существует в изоляции. Чтобы достичь желаемых результатов, организация должна работать с услугой как с целым, управляя информацией, технологиями, организацией, людьми, практиками, партнерами и соглашениями в единой динамичной системе;

- Сохраняйте всё простым и практичным. Избегайте излишней сложности. Упрощайте процессы и процедуры, используйте только те шаги, которые действительно необходимы для достижения целей. Сосредоточьтесь

на достижении практических результатов, исключая все, что не приносит ценности;

– Оптимизируйте и автоматизируйте. Используйте ресурсы эффективно. Устраняйте все, что является излишним или неэффективным, и используйте технологии для достижения наилучших результатов. Человеческое вмешательство должно быть ориентировано на добавление ценности.

Важным является осознание взаимодействия и взаимозависимости между принципами ИТIL. Необходимо рассматривать все принципы вместе и применять их взаимосвязанно, учитывая их взаимодействие в конкретной ситуации.

1.3 Метрики для анализа эффективности работы сотрудников в ITSM-системах

В рамках ITSM важно иметь эффективные метрики, которые позволяют оценить производительность и качество работы сотрудников [21]. Кроме того, необходим инструмент, для измерения указанных метрик. В данном случае отлично подойдет концепция канбан-доски, на которой размещаются задачи, выполнение которых требуется для предоставления ИТ-услуг.

1.3.1 Канбан-доска

Как описывают авторы книги Kanban In Action [7, разд. 2], канбан-доска (Kanban board) является инструментом визуализации рабочего процесса и отслеживания хода выполнения задач. Это позволяет сотрудникам и руководителям лучше понимать текущее состояние задач и их прогресс. На канбан-доске задачи представлены в виде карточек, которые перемещаются по столбцам, отражающим различные стадии выполнения. К примеру, столбцы могут включать "Открыто", "В работе", "На проверке" и "Готово". Это позволяет всем участникам проекта ясно видеть, какие задачи выполняются, какие ожидают выполнения и какие уже завершены (Рисунок 2).

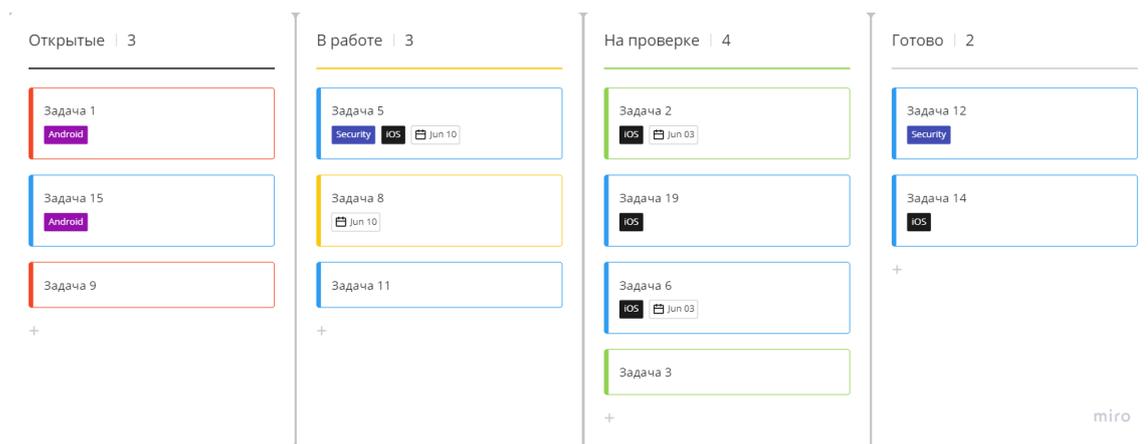


Рисунок 2 – Пример канбан-доски

Кроме этого, доска помогает определить приоритеты задач и управлять рабочим потоком. Каждая задача на доске имеет определенный приоритет, и ее размещение в столбцах отражает текущий статус и приоритет задачи. Это позволяет команде быстро определить, какие задачи требуют наибольшего внимания и какие могут быть выполнены позже.

Также такая система помогает в фиксировании различных метрик, по которым можно оценить эффективность работы команды, скорость и качество выполнения задач, слабые места и задержки в предоставлении услуг клиенту.

1.3.2 Lead Time

Lead Time, или время выполнения, – метрика времени, требуемого для выполнения задачи или обработки запроса от момента поступления до окончания работы. Например, если сотрудник получает запрос на устранение инцидента, Lead Time измеряет время, затраченное с момента попадания задачи на доску и до момента устранения проблемы и восстановления нормальной работы. Эта метрика помогает определить скорость выполнения задач от первого появления инцидента и до конечной доставки выполненной задачи клиенту, выявить возможные задержки или проблемы в процессе работы. Отображение метрики на канбан-доске представлено на рисунке 3.

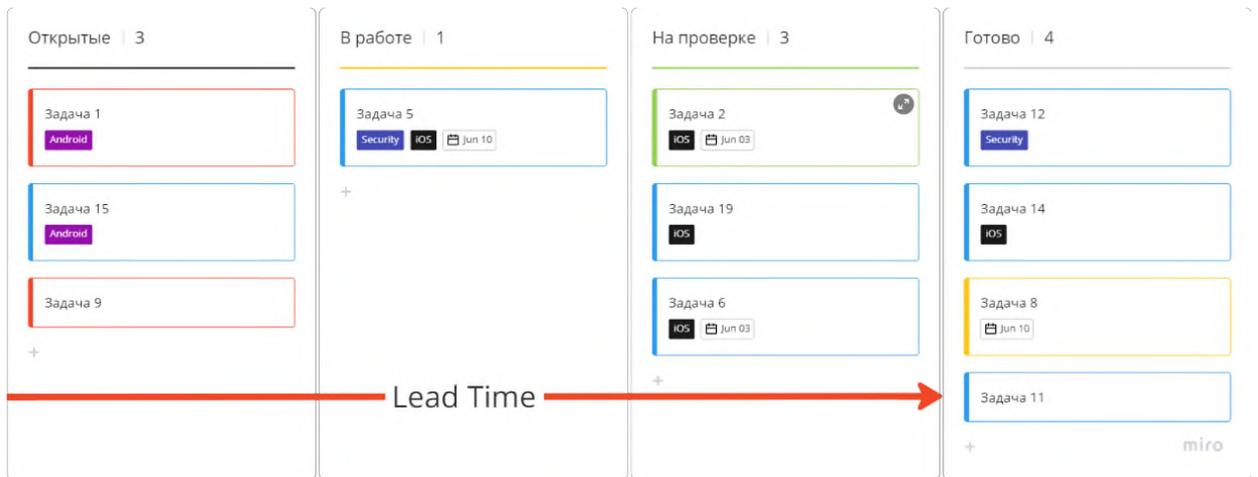


Рисунок 3 – Lead Time

1.3.3 Cycle Time

Cycle Time, или время цикла, – метрика времени, требуемого для завершения одного цикла работы с задачей или запросом. Он включает время обработки задачи, время ожидания и другие задержки, которые возникают в процессе работы. Cycle Time позволяет оценить пропускную способность рабочего процесса и выявить узкие места, которые могут замедлять выполнение задач. Например, можно измерить Cycle Time для разных типов инцидентов и определить среднее время, затрачиваемое на их устранение. Отображение метрики на канбан-доске представлено на рисунке 4.

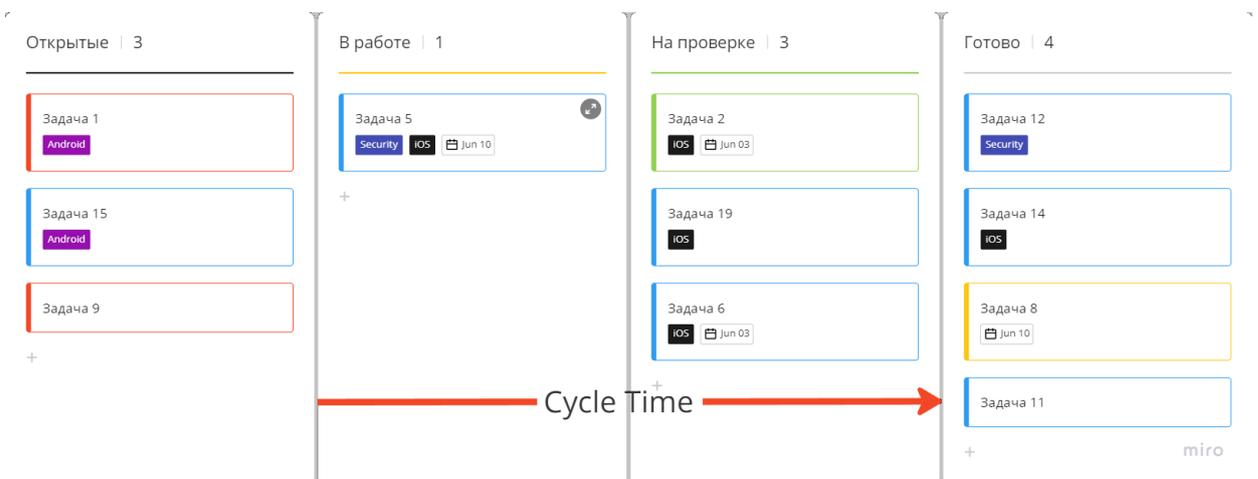


Рисунок 4 – Cycle Time

1.3.4 Throughput

Throughput, или пропускная способность – это метрика, которая отражает количество задач или работ, которые были завершены или обработаны за определенный период времени. Данный показатель можно рассчитать по следующей формуле (1):

$$\text{Throughput} = \frac{N}{\Delta T}, \quad (1)$$

где N – количество завершенных задач;

ΔT – промежуток времени, за который рассчитывается пропускная способность.

Она является ключевым показателем производительности и позволяет определить, насколько хорошо команда справляется с обработкой задач. Чем выше пропускная способность, тем больше задач команда выполняет за определенный период времени, что указывает на более эффективную работу. Throughput можно измерять в количестве запросов на техническую поддержку, которые были обработаны командой за неделю или месяц. Если команда обрабатывает большое количество запросов за короткий промежуток времени, это указывает на высокий Throughput и эффективную работу команды. Сравнение Throughput разных команд или периодов времени может помочь выявить лучшие практики и оптимизировать процессы работы.

Важно отметить, что Throughput следует анализировать совместно с другими метриками, такими как Cycle Time и Lead Time, чтобы получить полную картину эффективности работы сотрудников и команды в ITSM-системах.

1.3.5 Cumulative Flow Diagram

Cumulative Flow Diagram (CFD), или накопительная диаграмма потока, – это график, который отображает изменение размера накопленной задолженности (т.е. количество задач, находящихся в работе) по времени в

каждой стадии рабочего процесса. CFD позволяет визуализировать поток работы, выявить колебания в объеме задач и определить, где возникают задержки или проблемы в рабочем процессе [15]. Например, CFD может показать, что количество задач, находящихся в стадии "В ожидании", растет с течением времени, что может указывать на неэффективность в распределении работы или недостаточное количество ресурсов для выполнения задач. Пример диаграммы представлен на рисунке 5.

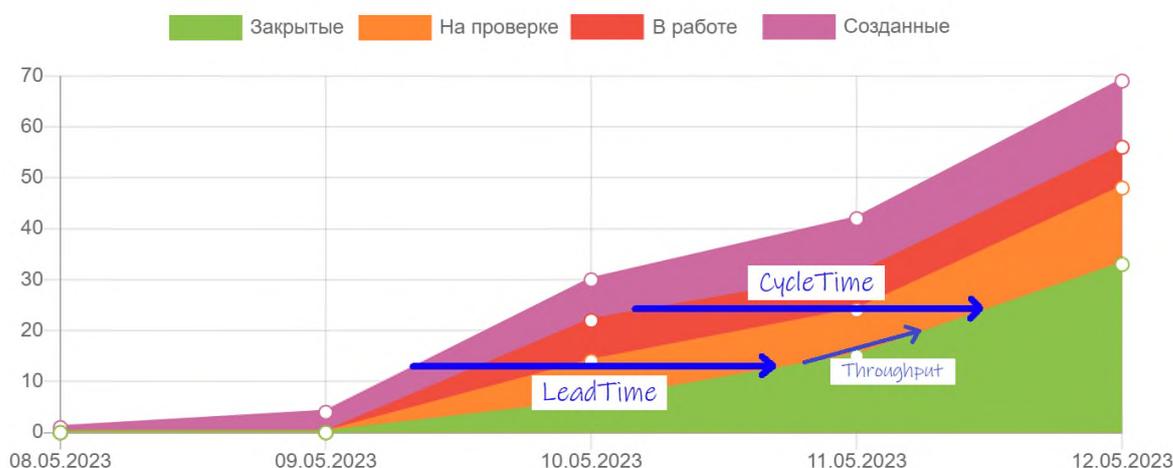


Рисунок 5 – Накопительная диаграмма потока

1.3.6 Cycle Time Scatter Plot

Scatter Plot, или диаграмма рассеяния, позволяет исследовать связь между двумя переменными, например, между временем выполнения задачи (Cycle Time) и другими факторами, такими как приоритет задачи, сложность или нагрузка на сотрудника. Анализ Scatter Plot диаграммы может помочь выявить тенденции или закономерности в данных и понять, какие факторы могут влиять на производительность и эффективность работы сотрудников. Например, Scatter Plot может показать, что задачи с более высоким приоритетом имеют более короткое время выполнения, что может указывать на необходимость более четкой приоритезации задач для оптимизации процесса работы. Кроме этого, на диаграмме сразу можно увидеть необычные

всплески в задачах, например на рисунке 6 видно, что одна из задач заняла гораздо больше времени, чем остальные, а значит на неё следует обратить особое внимание.

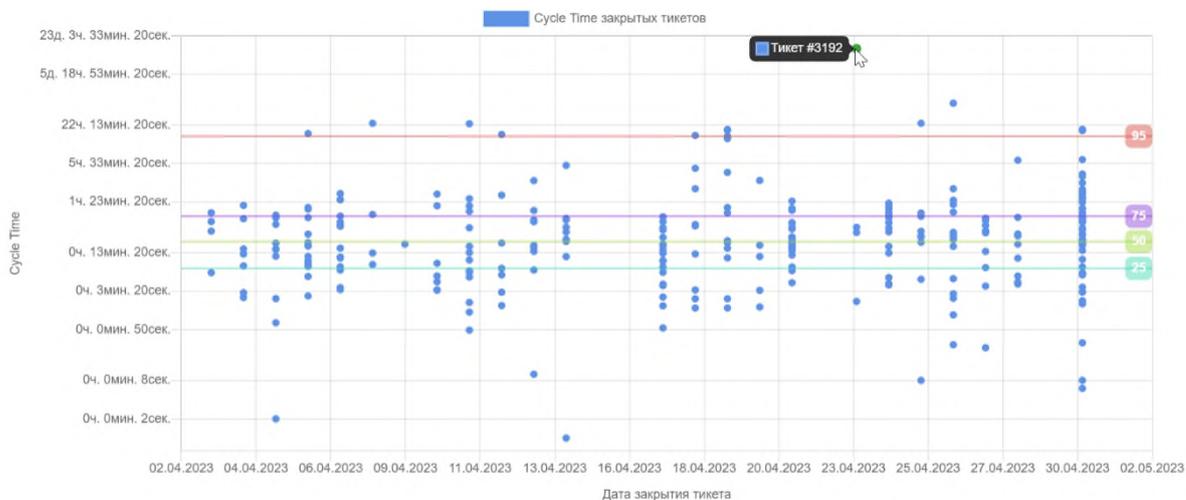


Рисунок 6 – Диаграмма рассеяния

1.4 Монолитный и микросервисный подходы в разработке сложных программных систем

Современное программное обеспечение стало все более сложным и требовательным к масштабируемости, гибкости и независимости компонентов. Для достижения этих целей разработчики исследовали и применяли различные подходы к проектированию и архитектуре систем. Два наиболее распространенных подхода – монолитный и микросервисный – получили широкое признание и нашли свое применение в различных проектах.

Монолитная архитектура представляет собой подход, при котором вся система разрабатывается как единое целое, где все компоненты и функциональность находятся в одной кодовой базе и выполняются в одном процессе. Этот подход является классическим и был широко распространен в прошлом.

Исторически монолитная архитектура была основным подходом к разработке программных систем. Одним из ранних сторонников этого подхода был автор "Паттерны объектно-ориентированного проектирования" Эрих Гамма (Erich Gamma) и его соавторы [4]. В своей книге они представили набор шаблонов проектирования, которые помогли разработчикам строить монолитные системы.

Монолитная архитектура имеет несколько преимуществ. Во-первых, она проще в разработке и развертывании. Все компоненты приложения находятся в одном месте, что упрощает отладку и тестирование. Во-вторых, монолитные системы могут быть более производительными, так как вызовы между компонентами происходят внутри одного процесса, что уменьшает накладные расходы на сетевое взаимодействие. В-третьих, монолиты могут быть более простыми в масштабировании, поскольку требуют меньше инфраструктуры и конфигурации.

Однако, с ростом размера и сложности системы, монолитная архитектура может столкнуться с рядом проблем, таких как сложность масштабирования и поддержки, гибкость внесения изменений и надежность. Кроме того, отказ одного компонента в монолите может привести к полному отказу всей системы. Это связано с тем, что все компоненты работают внутри одного процесса и не имеют изоляции друг от друга. К тому же каждое изменение в системе требует пересборки и развертывания всего приложения. Это может привести к затруднениям в непрерывной интеграции и доставке.

Пример монолитной архитектуры представлен на рисунке 7.

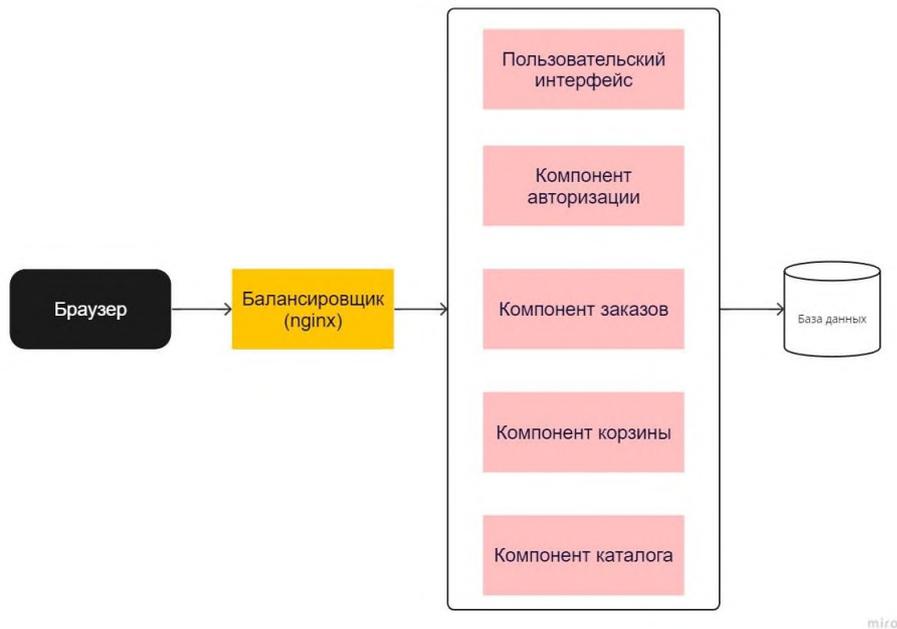


Рисунок 7 – Монолитная архитектура приложения

Мартин Фаулер (Martin Fowler), автор книги "Шаблоны корпоративных приложений", рассматривает проблемы монолитной архитектуры и предлагает альтернативные подходы, включая микросервисную архитектуру [2]. Он подробно описывает концепцию микросервисов и преимущества, которые они могут принести в разработку сложных систем.

Микросервисный подход – это архитектурный стиль, в котором приложение разбивается на набор независимых и слабо связанных сервисов. Каждый сервис представляет собой небольшую и отдельную часть функционала системы и может разрабатываться и развертываться независимо. Сервисы взаимодействуют между собой через легковесные протоколы, такие как HTTP, gRPC или AMQP.

Микросервисная архитектура стремится к высокой гибкости и масштабируемости, позволяя командам разработчиков независимо работать над отдельными сервисами. Каждый сервис имеет свою собственную базу кода, базу данных и набор API для общения с другими сервисами. Это позволяет более гибко масштабировать отдельные компоненты системы в зависимости от нагрузки и требований.

Микросервисный подход также способствует улучшению отказоустойчивости системы. Если один сервис перестает работать, остальные сервисы продолжают функционировать, что позволяет избежать полного простоя системы. Кроме того, использование легковесных протоколов для взаимодействия между сервисами делает систему более гибкой и совместимой с различными технологиями и платформами.

Однако микросервисная архитектура также вносит свои сложности. Взаимодействие между сервисами требует сетевых вызовов, что может повлечь за собой задержки и увеличение нагрузки на сеть. Также, в случае использования разных технологий для разных сервисов, могут возникать сложности при интеграции и поддержке.

Один из примеров современного микросервисного приложения представлен компанией Microsoft [9]. Схема архитектуры приложения представлена на рисунке 8.

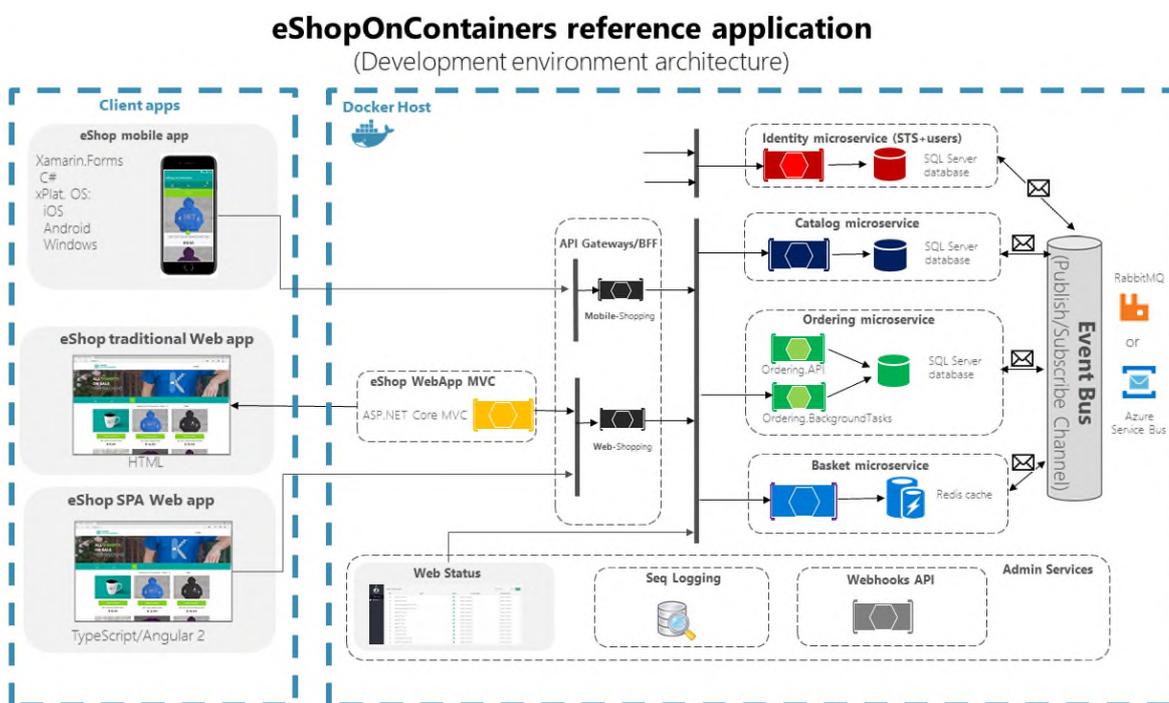


Рисунок 8 – Схема микросервисного приложения

В таблице 2 представлено сравнение монолитной и микросервисной архитектуры. Основные аспекты для сравнения включают разработку, развертывание, масштабируемость, надежность, гибкость и производительность.

Таблица 2 – Сравнение монолитной и микросервисной архитектур

Аспект	Монолитная архитектура	Микросервисная архитектура
Разработка	Проще в разработке, так как имеет одну кодовую базу	Сложнее, но имеет большую гибкость в выборе технологий для каждого сервиса
Развертывание	Легкое развертывание с одним процессом	Требует сложного управления для каждого сервиса, так как каждый сервис должен быть настроен отдельно
Масштабируемость	Ограничена всей системой	Гибкое масштабирование каждого сервиса независимо
Надежность	Могут быть более надежными, так как вызовы между компонентами происходят внутри одного процесса и нет необходимости в сетевом взаимодействии	Обеспечивают большую изоляцию и автономность, что может повысить надежность системы в целом
Гибкость	Изменения требуют пересборки и развертывания всего приложения	Легкое добавление функций и изменения без воздействия на другие сервисы
Производительность	Могут быть более производительными, так как вызовы между компонентами происходят внутри одного процесса	Обеспечивают лучшую горизонтальную масштабируемость и возможность использовать оптимальные технологии для каждого сервиса, что может улучшить производительность в целом

Оба подхода, монолитная и микросервисная архитектуры, имеют свои преимущества и недостатки. Выбор между ними зависит от конкретных требований и характеристик проекта.

Монолитная архитектура предоставляет простоту разработки и управления, особенно для небольших проектов с ограниченными требованиями к масштабированию и гибкости. Она может быть предпочтительна, если система относительно проста и не предполагается значительное увеличение масштаба.

Микросервисная архитектура подходит для сложных систем с высокими требованиями к гибкости, масштабируемости и отказоустойчивости. Она позволяет разрабатывать и масштабировать компоненты независимо, что упрощает разработку и адаптацию к изменениям. Однако, микросервисная архитектура требует большего внимания к управлению и мониторингу системы.

Итак, выбор между монолитной и микросервисной архитектурами зависит от конкретной ситуации и требований проекта. Лучший подход определяется на основе анализа масштабируемости, гибкости, надежности и сложности управления системой.

1.5 Принцип «единого окна»

Строго говоря, данный принцип широко применяется в нормировании торговых, экономических и правовых отношений. Скорее всего, оригинальной формулировкой можно считать формулировку Европейской экономической комиссии ООН. Она определяет «единое окно» как «систему, позволяющую торговым предприятиям представлять информацию единому органу для выполнения всех нормативных требований к импортным или экспортным операциям» [13]. Другими словами, потребитель (клиент) должен иметь возможность работать (предоставлять документы, получать информацию) только с одним сервисом, который скроет за собой остальную сложную работу по взаимодействию между собой других сервисов.

Хотя термин и применяется преимущественно в торговой отрасли, однако он отлично подходит для описания ITSM-системы, так как такая система и являет собой то «единое окно», через которой осуществляется взаимодействие между различными сервисами: аналитики, статистики, мониторинга и других, что даёт возможность сотрудникам компании взаимодействовать с единой системой для осуществления своих задач без переключения контекста работы.

1.6 Выводы

В ходе первой главы были описаны различные подходы к формированию ITSM-системы, рассмотрены различные метрики, позволяющие определить эффективность работы разработанной системы, а также проведено сравнение двух подходов к написанию программных приложений.

Исходя из проведенного анализа, можно сделать выводы, что использование принципов ITIL при проектировании ITSM-системы поможет повысить эффективность предоставления IT-услуг компанией, выявленные метрики помогут проконтролировать и убедиться в эффективности такой системы, а использование микросервисной архитектуры позволит повысить производительность, надежность и масштабируемость системы как программного продукта.

2 Разработка ITSM-системы для компании, предоставляющей услуги по видеофиксации магазинных краж

2.1 Постановка задачи

Требуется спроектировать и разработать ITSM-систему в виде WEB-приложения для компании, занимающейся видеофиксацией магазинных краж.

Приложение должно включать:

- интерфейс и функционал по мониторингу серверов и камер, установленных у клиентов компании;
- возможность диагностики камер, установленных у клиентов путем просмотра с них кадров;
- сбор статистической информации и дальнейшее её агрегирование и предоставление техническому отделу компании;
- управление и настройка ролей пользователей;
- управление настройками торговых сетей и магазинов (серверов, находящихся в них) клиентов;
- систему тикетов для контроля работы технического отдела;
- сбор различных метрик (скорость выполнения задач, их количество, типы) для анализа работы технического отдела;
- функционал по отправке уведомлений на почту об изменениях в тикетах;
- возможность выгрузки статистических отчетов по серверам.

2.2 Анализ подхода технического отдела к работе до внедрения ITSM-системы и определение потребностей по улучшению качества его работы

До внедрения ITSM-системы технический отдел использовал систему трекинга задач YouTrack от компании JetBrains [22]. Данный факт накладывал множество ограничений на работу отдела. Работникам приходилось вручную связывать задачи и ссылки на информацию о магазине в текущем на тот

момент справочнике магазинов, используя несколько сервисов, что существенно замедляло работу из-за необходимости переключения контекста работы.

Для выявления неисправности было необходимо удаленно подключаться на сам сервер клиента через множество специальных приложения и технологий, таких как VPN, SSH или AnyDesk, а также самостоятельно анализировать показатели работы компьютера, а настройки доступа к серверам клиентов хранились в общем Excel-файле, в котором легко можно было запутаться. Из-за такого порядка работ, сотрудниками часто допускались ошибки в настройке программного обеспечения, что плохо сказывалось на репутации компании.

Существенно замедляло работу и отсутствие агрегации статистики мониторинга работы серверов и настройки программного обеспечения на нём, что замедляло выявление неверно настроенных магазинов вплоть до нескольких месяцев.

Таким образом были выявлены определенные потребности компании, которые помогли бы улучшить работу технического отдела и повысить скорость и качества обслуживания серверов клиентов:

- единый сервис мониторинга, сбора и агрегации статистики по работе серверов клиентов;
- возможность просмотра информации по магазину и камерам в нём, настройкам программного и технического обеспечения серверов клиентов;
- система тикетов, которая бы позволила связать задачи и магазины, которые должны быть настроены по данным задачам;
- внедрение системы метрик, которые бы помогли оценить эффективность работы технического отдела и выявить задачи, которые представляют наибольшую трудность в выполнении.

Исходя из выявленных потребностей, компанией было принято решение о внедрении ITSM-системы, которая бы удовлетворила всем заявленным требованиям.

2.3 Анализ существующих решений ITSM-систем

Обзор аналогов ITSM-систем представляет собой важную часть исследования, позволяющую сравнить различные решения и выбрать оптимальное для организации. В данной работе рассмотрены несколько ITSM-систем, как российских, так и зарубежных, с целью оценки их функциональности, гибкости и соответствия требованиям.

2.3.1 ServiceNow

Первым рассмотренным решением является ServiceNow [10]. Эта система предлагает широкий спектр функций, включая управление инцидентами, проблемами, изменениями, конфигурацией и активами. ServiceNow является облачным решением, что обеспечивает высокую масштабируемость и доступность.

Из преимуществ данного решения можно выделить:

- обширный функционал ITSM, включающий управление инцидентами, проблемами, изменениями, конфигурацией и активами;
- гибкая настройка и конфигурация системы под уникальные потребности организации;
- облачное решение с высокой масштабируемостью и доступностью;
- широкая поддержка и активное сообщество пользователей.

У системы также имеются и недостатки:

- высокая стоимость лицензирования и поддержки;
- интеграция с другими системами может требовать дополнительных усилий;
- сложность внедрения и обучения пользователей.

2.3.2 BMC Remedy

Это интегрированная платформа, предлагающая широкий функционал для управления сервисами и автоматизации процессов [23]. BMC Remedy поддерживает ITIL-процессы и стандарты, обладает высоким уровнем

настраиваемости и гибкости. Однако, для некоторых пользователей может быть сложной в настройке и требовать больше усилий для интеграции с другими системами.

Преимущества:

- интегрированная платформа с широким функционалом для управления сервисами и автоматизации процессов;
- поддержка ITIL-процессов и стандартов;
- высокий уровень настраиваемости и гибкости системы.

Недостатки:

- более сложная конфигурация и настройка по сравнению с некоторыми другими системами;
- не интуитивный интерфейс.

2.3.3 Jira Service Management

Система Jira Service Management, интегрирующаяся с другими продуктами Atlassian, позволяет связывать управление сервисами с управлением проектами [5]. Jira Service Management обладает гибкостью через возможности настройки и дополнений. Однако, некоторые пользователи отмечают ограничения в отчетности и аналитике, а также возможные сложности при масштабировании и управлении большими объемами данных.

Преимущества:

- интегрируется с другими продуктами Atlassian и позволяет связывать управление сервисами с управлением проектами;
- гибкость через возможности настройки и дополнений;
- интуитивный интерфейс и знакомый рабочий процесс для пользователей Jira.

Недостатки:

- некоторые пользователи отмечают ограниченные возможности отчетности и аналитики;

- возможны сложности при масштабировании и управлении большими объемами данных;
- большее направление на команды разработчиков, чем на системное администрирование.

2.3.4 ITSM365

ITSM365 – это российская облачная платформа для управления ИТ-услугами, предназначенная для автоматизации и управления процессами информационно-технической поддержки в организациях [16]. Она предлагает широкий набор функциональных возможностей для управления процессами ITSM.

Преимущества:

- облачное решение, которое обеспечивает гибкость и масштабируемость;
- имеет интуитивно понятный интерфейс пользователя и простую настройку;
- поддерживает различные процессы ITSM, включая управление инцидентами, проблемами, изменениями и конфигурациями;
- интегрируется с другими системами, такими как Active Directory и системы мониторинга;
- предлагает аналитические инструменты для отслеживания производительности и оптимизации процессов.

Недостатки:

- могут возникать ограничения в настройке и интеграции по сравнению с более настраиваемыми решениями;
- может быть не подходящим для организаций с особыми требованиями или сложными процессами.

2.3.5 SimpleOne ITSM

Российская ITSM-система, которая помогает организациям эффективно управлять своими IT-процессами [24]. С простым и интуитивно понятным интерфейсом она позволяет регистрировать и отслеживать инциденты, запросы и проблемы пользователей, управлять процессами изменений и конфигураций, а также предоставляет базовые инструменты отчетности и аналитики. SimpleOne ITSM облегчает внедрение и настройку, позволяя организациям быстро начать использование системы и адаптировать ее к своим уникальным требованиям.

Преимущества:

- простой и интуитивно понятный интерфейс, который требует минимального обучения пользователей;
- легко настраивается и внедряется, что упрощает начало использования системы;
- обладает основными функциями ITSM, такими как управление инцидентами, проблемами и изменениями;
- предлагает базовые инструменты отчетности и аналитики.

Недостатки:

- может быть недостаточно расширяемым или настраиваемым для организаций с более сложными потребностями;
- может отсутствовать некоторый функционал, доступный в более полноценных платформах ITSM.

2.3.6 Naumen Service Desk

Naumen Service Desk – это решение для управления IT-сервисами, предлагаемое компанией Naumen и ориентирующееся на практики ITIL [16]. Интуитивно понятный интерфейс, удобные инструменты отчетности и аналитики делают Naumen Service Desk эффективным решением для организаций, стремящихся улучшить управление IT-сервисами и обеспечить высокое качество обслуживания.

Преимущества:

– широкий функционал для управления ITSM, включая управление инцидентами, проблемами, изменениями, конфигурациями и т.д;

– предоставляет гибкие инструменты настройки и конфигурации, позволяющие адаптировать решение к конкретным потребностям организации;

– имеет мощные возможности автоматизации для оптимизации процессов и улучшения эффективности;

– обладает интуитивно понятным интерфейсом и удобными инструментами отчетности и аналитики.

Недостатки:

– может потребоваться более сложная настройка и внедрение по сравнению с некоторыми другими платформами;

– некоторые дополнительные функции могут требовать дополнительных лицензий.

2.3.7 Сравнение систем

В таблице 3 представлено сравнение перечисленных сервисов.

Таблица 3 – Сравнение ITSM-систем

Аспекты / ITSM-системы	Service Now	BMC Remedy	Jira Service Management	ITSM 365	Simple One ITSM	Naumen Service Desk
Система тикетов	Да	Да	Да	Да	Да	Да
Система метрик, необходимых компании (CFD, LeadTime, CycleTime)	Нет	Нет	Нет	Нет	Нет	Нет

Продолжение таблицы 3

Аспекты / ITSM-системы	Service Now	BMC Remedy	Jira Service Management	ITSM 365	Simple One ITSM	Naumen Service Desk
Добавление собственных моделей данных	Нет	Нет	Нет	Нет	Нет	Нет
Аналитика и мониторинг моделей	Нет	Нет	Нет	Нет	Нет	Нет
Агрегация статистики	Нет	Нет	Нет	Нет	Нет	Нет
Возможность поднять систему на собственном хосте	Да	Да	Да	Да	Да	Да
Уровень безопасности хранения данных	Высокий	Высокий	Средний	Высокий	Высокий	Высокий

Как можно видеть, ни одна из систем не удовлетворяет основным требованиям компании (сбор и агрегация статистики, мониторинг добавленных сущностей). Конечно, можно попытаться реализовать сервис по мониторингу и синхронизировать его с внешними ITSM-системами, но это только усложнит работу, а также нарушит принцип «единого окна».

2.4 Преимущества внедрения собственной системы

Проанализировав различные подходы к проектированию ITSM-системы, а также сравнив различные аналоги существующих систем, как отечественных, так и зарубежных, можно сформулировать следующие преимущества от разработки и внедрения собственной системы:

- соблюдение принципа «единого окна», так как взаимодействие всех сервисов будет происходить через один интерфейс;
- возможность мониторинга и агрегации статистики по серверам клиентов напрямую через систему;
- возможность внедрения своих собственных метрик для оценки качества выполнения задач;
- гибкость и масштабируемость системы под любые индивидуальные нужды компании.

2.5 Выбор средств разработки

Для достижения этой цели было решено использовать современные и эффективные инструменты разработки. Ниже приведено обоснование выбранных средств.

2.5.1 Angular

Angular – это открытая и популярная платформа для разработки веб-приложений, созданная командой Angular в Google [25]. Angular позволяет разработчикам создавать мощные и динамические одностраничные приложения (SPA) с использованием языков программирования TypeScript, HTML и CSS. Angular был выбран в качестве основного фреймворка для разработки клиентской части веб-приложения. Его модульная структура, мощные инструменты и простота разработки обеспечивают удобство создания сложных пользовательских интерфейсов. Angular также обладает богатыми возможностями для управления состоянием приложения, маршрутизации и взаимодействия с серверной частью, что является неотъемлемым требованием для современных веб-приложений.

2.5.2 ASP.NET

ASP.NET (Active Server Pages .NET) – это платформа разработки веб-приложений, разработанная компанией Microsoft [12]. Она является частью

более широкого фреймворка .NET, который предоставляет инструменты и среду выполнения для разработки и запуска различных типов приложений, включая веб-приложения. ASP.NET выбран в качестве основного фреймворка для разработки серверной части веб-приложения. Он предоставляет мощные инструменты для создания масштабируемых и безопасных веб-сервисов, а также позволяет эффективно управлять бизнес-логикой и взаимодействием с базами данных. Фреймворк ASP.NET Web API (часть ASP.NET) позволяет создавать веб-службы, которые могут быть использованы другими приложениями для обмена данными по протоколу HTTP.

2.5.3 PostgreSQL

PostgreSQL – это мощная и расширяемая объектно-реляционная система управления базами данных (СУБД) [17]. Она является одной из самых популярных и широко используемых открытых СУБД и предоставляет надежное хранение и обработку структурированных данных. PostgreSQL выбран в качестве реляционной базы данных для хранения и управления данными приложения о магазинах, торговых сетях, пользователях, посетителях и конфигурационных файлах. Его надежность, масштабируемость и широкий функционал позволяют эффективно работать с различными типами данных и выполнять сложные запросы.

2.5.4 MongoDB

MongoDB – это гибкая и масштабируемая документно-ориентированная система управления базами данных (NoSQL), которая обеспечивает хранение и обработку неструктурированных данных [26]. MongoDB обладает высокой производительностью и горизонтальной масштабируемостью, что позволит эффективно обрабатывать большие объемы информации, не снижая производительности приложения. Данная база будет хранить данные об обнаруженных в магазинах ошибках, а также мониторинговую информацию с серверов клиентов.

2.5.5 Elasticsearch

Elasticsearch – это мощный и гибкий поисковый и аналитический движок с открытым исходным кодом, который основан на Apache Lucene [27]. Он предназначен для обработки, хранения и поиска больших объемов структурированных и неструктурированных данных в реальном времени. Его возможности по полнотекстовому поиску, фильтрации и агрегации данных позволят эффективно осуществлять поиск и аналитику в рамках разрабатываемой системы.

2.5.6 Vault

Vault – это инструмент с открытым исходным кодом, разработанный компанией HashiCorp для управления и обеспечения безопасности секретов, таких как пароли, ключи API, сертификаты и другие конфиденциальные данные [28]. Он предоставляет централизованное хранилище для безопасного хранения и управления секретами, а также предоставляет механизмы доступа и аудита для обеспечения контроля над доступом к данным. Его функционал по шифрованию обеспечит высокий уровень безопасности данных для доступа к базам и другим сервисам.

2.5.7 RabbitMQ

RabbitMQ – это мощная и гибкая система посредника сообщений (message broker), которая обеспечивает эффективную коммуникацию между различными компонентами приложения, работающими в распределенной среде [30]. Она реализует протокол Advanced Message Queuing Protocol (AMQP) и предоставляет надежное и масштабируемое решение для обмена сообщениями между различными приложениями и сервисами. Выбрана в качестве системы обмена сообщениями для обеспечения асинхронного взаимодействия между компонентами приложения. Ее надежность и гибкость обеспечат эффективную коммуникацию и обработку задач в реальном

времени, что позволит достичь высокой отзывчивости и отказоустойчивости системы.

2.5.8 GitLab CI/CD

GitLab CI/CD – это инструмент непрерывной интеграции и непрерывной доставки (Continuous Integration/Continuous Delivery), который входит в состав платформы GitLab [31]. Он обеспечивает автоматизацию процессов сборки, тестирования, развертывания и доставки программного обеспечения, позволяя более эффективно и быстро создавать, тестировать и доставлять свои приложения. GitLab CI/CD использует простой и понятный синтаксис YAML для определения конфигурации процессов CI/CD. Благодаря этому можно описать этапы сборки, тестирования и доставки приложения, а также указать требования к окружениям и настройки развертывания.

2.5.9 Docker

Docker – это платформа с открытым исходным кодом, предназначенная для создания, развертывания и управления контейнерами [14]. Контейнеры являются легковесными и изолированными средами, в которых можно запускать приложения и их зависимости, обеспечивая консистентность и портбельность приложений между различными окружениями. Использование Docker позволит контейнеризовать приложение и его компоненты, что обеспечит легковесное развертывание, масштабируемость и изолированность. Это упростит управление приложением и обеспечит согласованность окружения в различных условиях разработки и развертывания.

Использование перечисленных средств позволит создать целостную, масштабируемую и поддерживаемую систему для решения поставленных задач.

2.6 Разработка и внедрение ITSM-системы

Как было сказано ранее, для реализации ITSM-системы выбрана микросервисная архитектура. Всего спроектировано 7 микросервисов, плюс сервис с пользовательским интерфейсом – API Gateway (WebApp):

- ms-diag – диагностическая информация по магазинам и камерам;
- ms-proxucheck – файлы конфигураций серверов;
- ms-frames – кадры с камер магазинов;
- ms-stats – агрегация статистики, формирование и отправка отчетов;
- ms-tickets – система тикетов (задач) для сотрудников;
- ms-storemanager – хранение и настройка ТС, дивизионов, магазинов;
- ms-identity – сервис аутентификации и авторизации пользователей.

Схема сервисов представлена на рисунке 9.

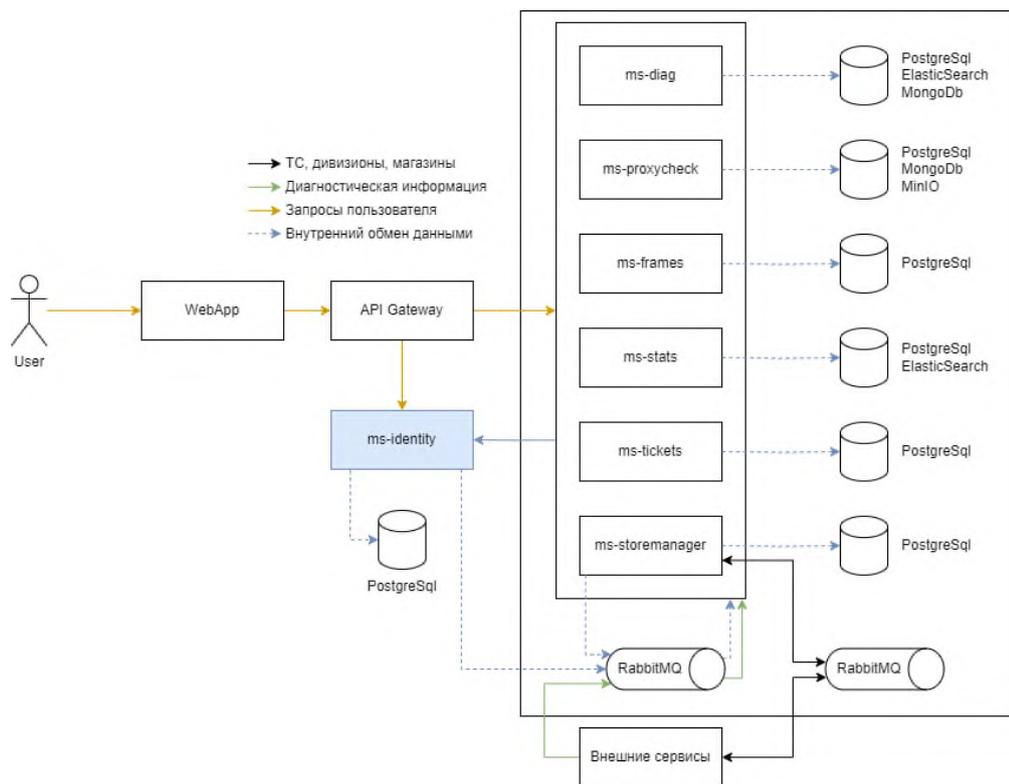


Рисунок 9 – Схема микросервисной архитектуры ITSM

Основные сущности всех сервисов:

– торговая сеть – информация о торговой сети клиента, содержит дивизионы и магазины;

– дивизион – часть торговой сети, обозначающая какой-либо регион расположения, например Север, Юг, Урал и т.д., содержит магазины;

– магазин – часть дивизиона, основная единица системы, представляет собой сервер (компьютер) в магазине клиента, содержит камеры;

– камера – часть магазина, в каждом магазине может быть несколько камер, есть возможность получения кадра с конкретной камеры.

Рассмотрим данные сервисы подробнее.

2.6.1 Микросервис диагностики

Микросервис диагностики `ms-diag` создан для сбора и агрегации мониторинговой информации по серверам клиентов. Он позволяет увидеть актуальное состояние системы, а также выявить ошибки при работе или настройке серверов.

Сервис использует базу PostgreSQL для хранения информации о торговых сетях, дивизионах и магазинах.

Сервис использует базу Elasticsearch для хранения информации о магазинах, камерах, пингах, диагностике серверов за всё время работы. Также отдельно хранит подготовленную информацию о последней диагностике для облегчения построения агрегированной статистики.

Сервис использует базу MonogDb для хранения обнаруженных в ходе работы ошибок.

Данные по диагностике приходят с внешнего сервиса, данные по ТС, дивизионам, магазинам приходят из микросервиса `ms-storemanager`.

В приложении А представлены пользовательские интерфейсы, через которые происходит доступ к данному микросервису.

Таким образом данный микросервис помогает проводить диагностику и следить за состоянием серверов клиентов.

2.6.2 Микросервис статистики и отчетов

Данный микросервис (ms-stats) используется для сбора данных по посещениям, агрегации этих данных в отчеты и последующей отправки отчетов клиентам на электронную почту.

Сервис использует базу данных PostgreSQL для хранения информации о клиентах и их почты для рассылки с описанием, какой тип отчета нужен клиенту.

Сервис использует базу данных Elasticsearch для хранения диагностики серверов, а также статистики посещений посетителей (обезличенная информация, содержащая только визуальные признаки: пол, возраст, эмоцию).

Отчет бывает двух типов:

- обычный,
- расширенный.

В расширенный отчет дополнительно добавляется информация по диагностике сервера (три поля: FacesFailed, Fps, Processing Fps).

В приложении Б показан пример настройки отчета для клиента. Периодичность отчета задается в формате cron. Клиент получает отчет в Excel-файле. Пример расширенного отчета представлен на рисунке 10.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	ExportStoreId	DivisionId	Day	Month	Year	TotalCount	Females	Males	Under18	From18To25	From25To35	From35To45	From45To55	From55	FacesFailed	Fps	Processing Fps
2	19	23	9	3	2022	1090	926	164	271	503	227	51	30	8	0	25,57600004	25,0356742095947
3	25	23	9	3	2022	655	500	155	97	284	191	46	26	11	618	24,6	25,0349605878194
4	37	23	9	3	2022	455	332	123	55	154	147	50	40	9	0	25,60000004	25,0218974403713
5	45	23	9	3	2022	1552	1150	402	159	712	506	122	45	8	0	25,60000004	25,0859150091807
6	55	23	9	3	2022	1544	1344	200	227	827	375	80	25	10	0	25,60000004	25,1097149054209
7	65	23	9	3	2022	1397	1142	255	153	644	382	142	63	13	0	25,60000004	25,0215973854065
8	74	23	9	3	2022	428	320	108	72	178	115	39	16	8	0	25,6043482	25,0534599138343
9	75	23	9	3	2022	2716	2068	648	351	1219	753	274	96	23	0	25,58000004	25,0418579864502
10	78	23	9	3	2022	623	517	106	103	290	149	52	21	8	2	25,60000004	25,0399309539795
11	80	23	9	3	2022	377	279	98	78	133	99	39	23	5	158	25,60000004	25,1295889282227

Рисунок 10 – Пример отчета

Таким образом, микросервис помогает уведомлять клиентов о статистике посещаемости в их магазинах.

2.6.3 Микросервис файлов конфигураций

Микросервис `ms-proxucheck` служит для хранения примеров файлов конфигураций ПО (`config files`), которые должны находиться непосредственно на самих серверах. Кроме этого, сервис хранит списки параметров серверов в формате ключ-значение (`proxucheck files`). Списки параметров приходят в сервис непосредственно с самих серверов и содержат информацию о технической и программной конфигурации, например время в работе, количество CPU, GPU, и другие различные параметры, которые задаются техническим отделом.

Сервис использует базу `PostgreSQL` для хранения данных о магазинах, истории `proxucheck` файлов, списка параметров из этих файлов для возможности агрегации статистики по ним, а также списка конфигурационных файлов.

Сервис использует базу `MongoDb` для хранения последнего пришедшего `proxucheck` файла по каждому магазину. Благодаря такому хранению можно быстро агрегировать статистическую информацию по всем магазинам.

В приложении В представлен интерфейс, через который происходит обращение к данному сервису.

Таким образом, данный сервис позволяет собирать статистику настроек серверов и ПО на них.

2.6.4 Микросервис кадров с камер

Микросервис `ms-frames` служит для хранения кадров с камер магазинов.

Сервис использует базу `PostgreSQL` для хранения списка кадров. Каждая запись в базе о кадре имеет ссылку на хранилище, в котором лежит файл кадра.

Для защиты доступа к кадрам используется библиотека `ImgProхu`, которая позволяет зашифровать ссылку на файл в хранилище.

Диагностический кадр представляет собой кадр с камеры, запечатленный во время запуска диагностики. Такой кадр служит для

проверки настройки камеры, смотрит ли она в верном направлении, правильно ли задана зона видеофиксации.

В приложении Г представлен пример пользовательского интерфейса для работы с данным микросервисом.

Таким образом, данный микросервис позволяет контролировать настройки камер наблюдения в магазинах.

2.6.5 Микросервис ТС, дивизионов и магазинов

Сервис `ms-storemanager` предназначен для настройки ТС, дивизионов и магазинов. Он также содержит информацию по настройкам камер.

Он получает данные об этих сущностях из внешнего сервиса, а также сам отправляет измененные данные в другие указанные здесь микросервисы, а также во внешний сервис. Для этого используется брокер сообщений `RabbitMq`.

Сервис использует базу `PostgreSQL` для хранения всей информации.

Помимо сохранения настроек, которые уже были доступны в прошлом сервисе, с которым работали сотрудники, в данном сервисе появилась возможность сохранения настроек доступа на сервера клиентов, что значительно упростило работу техническому отделу и снизило количество совершаемых ошибок.

В приложении Д показан пример пользовательского интерфейса для работы с данным микросервисом.

Таким образом данный микросервис можно считать основным, так как он предназначен для контроля настроек главных сущностей и передаёт эту информацию в остальные сервисы.

2.6.6 Микросервис тикетов

Микросервис ms-tickets используется для управления задачами (трекинга), а также для сбора различных метрик эффективности работы сотрудников, такие как LeadTime, CycleTime и Throughput, описанные ранее в работе.

Сервис использует базу PostgreSQL для хранения информации о пользователях, тикетах (задачах), комментариях к ним и другое.

В сервис добавлена система уведомлений, которая отправляет информацию об изменениях в тикетах пользователям, прикрепленным к тикету, у которого произошли изменения, на электронную почту.

Также сервис позволяют оценить различные метрики эффективности работы сотрудников, построить диаграммы CFD и ScatterPlot.

Каждый тикет имеет различные параметры:

- заголовок,
- описание,
- приоритет,
- статус,
- поле экспертизы,
- поле документации,
- личные заметки пользователя по тикету,
- исполнитель и инициатор,
- список прикрепленных к тикету магазинов,
- тип тикета.

Типы тикета задаются руководством. Самые распространенные: установка магазина, проверка магазина, настройка звука, проверка звука и другие.

Кроме этого, существует 4 типа приоритета: низкий, обычный, высокий и критический.

Тикет может находиться в 5 статусах: открытые, в работе, на проверке, отложенные и закрытые. Статус «Отложенные» является уникальным для

подобных канбан-досок, так как позволяет отложить выполнение тикета на определенное или неопределенное время. Если тикет откладывается на определенное время, то по его прошествии задача возвращается в статус «Открытые». Если тикет был отложен на неопределенное время, он помечается как «Ожидает ответа клиента», а на карточку тикета наносится специальный значок.

Пример пользовательского интерфейса для взаимодействия с данным микросервисом представлен в приложении Е.

Таким образом, данный микросервис позволяет контролировать выполнение задач сотрудниками, а также оценивать эффективность их работы.

2.6.7 Микросервис аутентификации и авторизации

Микросервис `ms-identity` используется для аутентификации и авторизации пользователей.

Также через него возможно управление ролями и правами доступа.

Каждый из микросервисов имеет доступ к данному микросервису для авторизации пользователя по токену. Запрос на авторизацию происходит каждый раз, когда требуется проверка прав доступа.

Сервис использует базу PostgreSQL для хранения информации о пользователях, ролях и правах доступа.

Пример работы с сервисом через пользовательский интерфейс продемонстрирован в приложении Ж.

2.6.8 WebApp и API Gateway

API Gateway представляет собой прослойку между пользовательским интерфейсом и всеми остальными микросервисами. Для этого в проекте WebApp (C#) используется библиотека YARP (Yet Another Reverse Proxy) [32]. С помощью нее запросы к фронтальной части WebApp, отправленные на бэкенд этого же сервиса, переадресовываются на соответствующие микросервисы, прописанные в запросе.

Например, если запрос будет вызван по адресу `/api/diag/{request}`, то запрос будет перенаправлен в микросервис `ms-diag`, если запрос вызван по адресу `/api/tickets/{request}`, то он будет перенаправлен в `ms-tickets`, и так далее.

2.6.9 CI/CD

Для работоспособности всех микросервисов была использована технология GitLab CI/CD.

Для каждого микросервиса был написан пайплайн (набор инструкция), по которому сервис собирается и запускается в Docker на удаленном хосте.

Работа пайплайна представлена на рисунке 11.

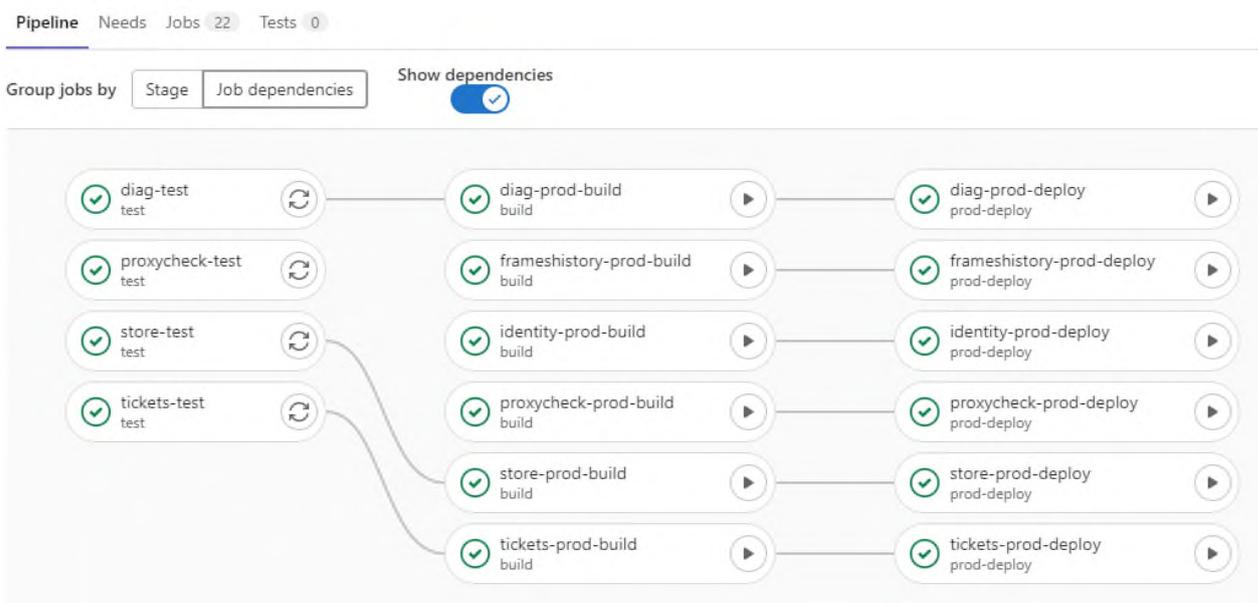


Рисунок 11 – Работа пайплайна по деплою микросервисов

Таким образом, данная технология позволила настроить непрерывную доставку, автоматическое тестирование и развертывание микросервисов, что существенно упростило разработку проекта.

2.7 Анализ показателей скорости работы технического отдела при старом и новом подходах

До внедрения ITSM-системы сотрудники компании пользовались трекером задач YouTrack. Данный сервис не имеет таких же метрик, как в разрабатываемой системе (LeadTime, CycleTime, Throughput), но можно выгрузить список всех задач в Excel-файл, а после самостоятельно вывести все метрики (кроме CycleTime, так как такой информации нет в YouTrack).

На скорость выполнения задач влияет множество факторов, поэтому нельзя делать выводы на основании одного сравнения. Начиная со второго сравнения будут производиться между тремя периодами: во время использования YouTrack, во время использования ITSM до начала 2023 года и во время использования ITSM после начала 2023 года, так как после нового года система была значительно улучшена.

2.7.1 Сравнение всех задач за равный период

Для сравнения взят период 25.07.2023 – 20.05.2023 для новой системы и 25.07.2021 – 20.05.2022 для YouTrack, так как 25.07.2022 система была введена в эксплуатацию и в ней появилась первая задача.

На рисунке 12 представлены два CFD-графика – в YouTrack и в ITSM.

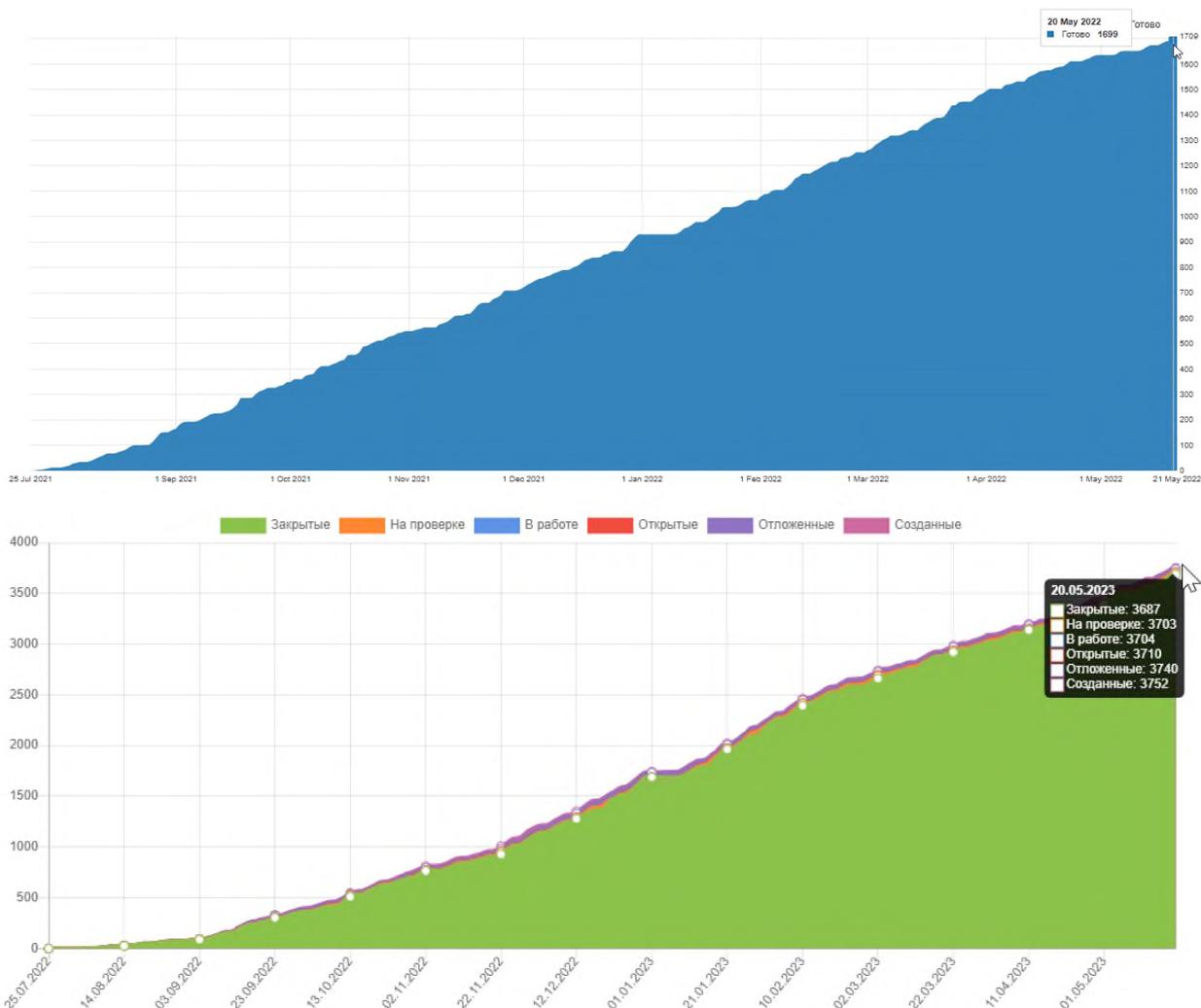


Рисунок 12 – Сравнение всех задач, созданных за указанный период

В таблице 4 представлены значения основных метрик из двух сервисов.

Таблица 4 – Сравнение метрик YouTrack и ITSM

Метрика	YouTrack	ITSM	Разница
LeadTime	4д. 14ч. 56м. 26с.	3д. 14ч. 22м. 43с.	1д. 23м. 43с.
Throughput	5.81	12.33	6.52

Таким образом, за равный период показатели эффективности выросли примерно в 2 раза.

2.7.2 Сравнение задач с типом Установка магазина

На рисунке 13 представлено сравнение количества выполненных задач по установке магазинов в разные периоды длиной в 3 месяца.

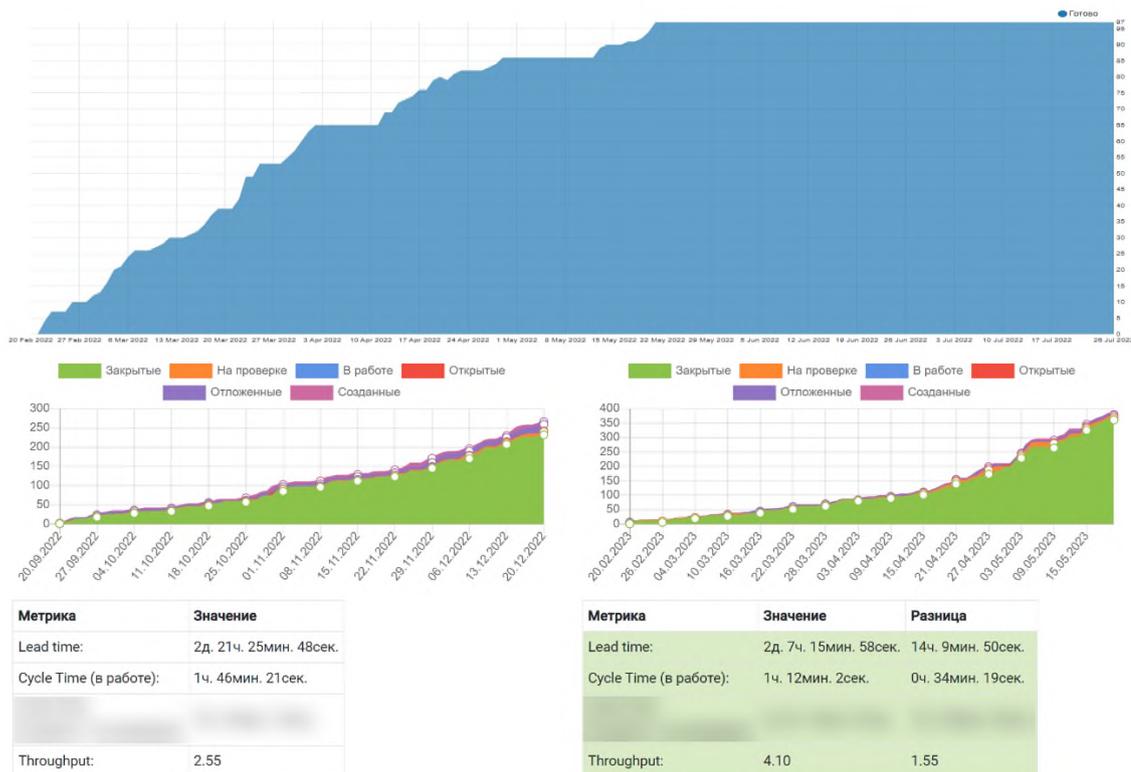


Рисунок 13 – Метрики при установке магазинов

В таблице 5 представлены значения основных метрик из двух сервисов.

Таблица 5 – Сравнение метрик из YouTrack и ITSM

Метрика	YouTrack	ITSM (до НГ)	ITSM (после НГ)	Разница
LeadTime	3д. 12ч. 12м. 38с.	2д. 21ч. 25м. 48с.	2д. 7ч. 15м. 58с.	1д. 4ч. 56м. 40с.
Throughput	0.8	2.55	4.10	3.3

Таким образом, показатель производительности увеличился примерно в 4 раза.

2.7.3 Сравнение задач с типом Переустановка магазина

На рисунке 14 представлено сравнение количества выполненных задач по переустановке магазинов в разные периоды длиной в 3 месяца.



Рисунок 14 – Метрики при переустановке магазинов

В таблице 6 представлены значения основных метрик из двух сервисов.

Таблица 6 – Сравнение метрик из YouTrack и ITSM

Метрика	YouTrack	ITSM (до НГ)	ITSM (после НГ)	Разница
LeadTime	4д. 9ч. 16м. 21с.	1д. 15ч. 56м. 24с.	2д. 14ч. 39м. 46с.	1д. 18ч. 36м. 35с.
Throughput	0.31	0.53	0.67	0.36

Таким образом, эффективность работы увеличилась примерно в 2 раза.

Стоит отметить, что увеличение LeadTime после нового года связано с увеличением времени проверки задач инициатором, что не влияет на эффективность работы исполнителя, которую можно видеть по второй строке таблицы на рисунке 14 (CycleTime в работе).

2.7.4 Сравнение задач с типом Проверка магазинов

На рисунке 15 представлено сравнение количества выполненных задач по проверке магазинов в разные периоды длиной в 3 месяца.

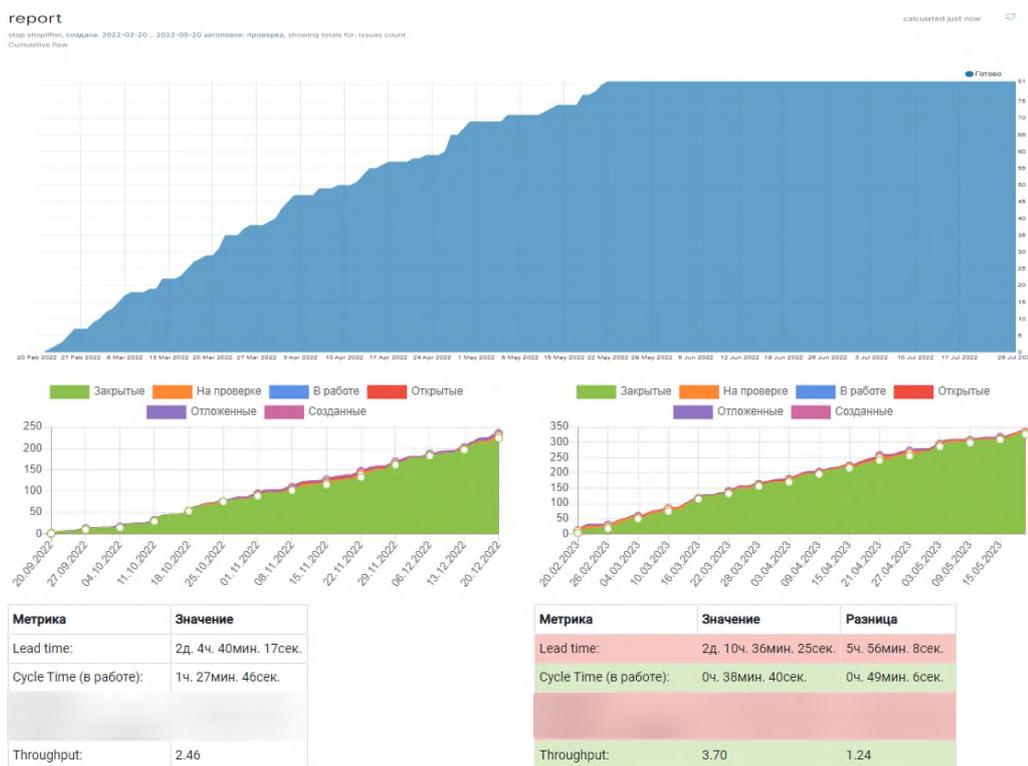


Рисунок 15 – Метрики при проверке магазинов

В таблице 7 представлены значения основных метрик из двух сервисов.

Таблица 7 – Сравнение метрик из YouTrack и ITSM

Метрика	YouTrack	ITSM (до НГ)	ITSM (после НГ)	Разница
LeadTime	3д. 12ч. 48м. 40с.	2д. 4ч. 40м. 17с.	2д. 10ч. 36м. 25с.	1д. 2ч. 12м. 15с.
Throughput	0.65	2.46	3.7	3.05

Таким образом, эффективность работы увеличилась примерно в 3 раза.

2.7.5 Выводы

Сравнив показатели работы сотрудников за разные периоды в разных системах и над разными типами задач, можно сделать вывод, что средний показатель эффективности работы вырос в 3 раза:

- скорость установки магазинов выросла в 4 раза;
- скорость переустановки магазинов выросла в 2 раза;
- скорость проверки магазинов выросла в 3 раза.

ЗАКЛЮЧЕНИЕ

В ходе работы были рассмотрены основные принципы построения ITSM-систем, изучены различные стандарты разработки, а также описаны необходимые метрики для оценки эффективности работы сотрудников.

Было выявлено, что лучшей практикой для проектирования ITSM-системы является соблюдения принципов ITIL, однако они могут выполняться в симбиозе с другими стандартами, такими как COBIT и ISO 20000.

Кроме этого, были рассмотрены различные метрики эффективности работы сотрудников:

– LeadTime – время доставки задачи до клиента от времени её создания, до полного завершения, включая тестирование и подтверждение заказчиком;

– CycleTime – время, затраченное на выполнение задачи, от времени взятия задачи в работу, до перехода задачи в состояние тестирования;

– Throughput – производительность команды, количество задач, выполненных за определенный промежуток времени.

Однако, делать выводы об эффективности работы команды можно только по совокупности всех трёх параметров, так как существует множество сторонних факторов, влияющих на величину показателя, как, например, скорость проверки задач руководством или клиентами.

По итогу разработки и внедрения разработанной ITSM-системы на основании проведенного анализа эффективности работы технического отдела, как в старой, так и в новой системе, можно подтвердить выдвинутую в начале работы гипотезу о том, что такая система поможет повысить скорость выполнения различных типов задач, а также количество выполненных задач в целом.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. ГОСТ Р ИСО/МЭК 20000-1-2021. Информационные технологии. Менеджмент сервисов. Часть 1. Требования к системе менеджмента сервисов [Текст]: дата введения 2022-04-30. – М.: Национальный стандарт Российской Федерации, 2021. – 32 с.
2. Фаулер М. Шаблоны корпоративных приложений / М. Фаулер . – Москва: И.Д. Вильямс, 2016. – 544 с.
3. Новостной поток. ISO 20000-1:2018. Всё ближе к ITIL [Электронный ресурс] // Digital Enterprise. – URL: <https://cleverics.ru/digital/2018/11/iso20000-2018-looks-like-itil/> (дата обращения: 22.05.2023).
4. Гамма Э. [и др.]. Паттерны объектно-ориентированного проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. – СПб: И.Д. Питер, 2021. – 448 с.
5. Atlassian Jira Service Management | A new take on ITSM software [Электронный ресурс] // Atlassian. – URL: <https://www.atlassian.com/software/jira/service-management> (дата обращения: 24.05.2023).
6. AXELOS, ITIL Foundation ITIL 4 Edition / AXELOS. – London: TSO, 2019. – 260 с.
7. Hammarberg M., Sundén J. Kanban in Action / M. Hammarberg, J. Sundén. – NY: Manning, 2014. – 360 с.
8. Naumen Service Desk [Электронный ресурс] // Naumen. – URL: https://www.naumen.ru/products/service_desk/ (дата обращения: 26.05.2023).
9. Знакомство со справочным приложением eShopOnContainers [Электронный ресурс] // Microsoft. – URL: <https://learn.microsoft.com/ru-ru/dotnet/architecture/cloud-native/introduce-eshoponcontainers-reference-app> (дата обращения: 23.05.2023).
10. IT Service Management (ITSM) - ServiceNow [Электронный ресурс] // ServiceNow. – URL: <https://www.servicenow.com/products/itsm.html> (дата обращения: 24.05.2023).

11. ITIL vs ISO 20000: Know the Similarities and Differences [Электронный ресурс] // KnowledgeHut. – URL: <https://www.knowledgehut.com/blog/it-service-management/itil-vs-iso-20000> (дата обращения: 22.05.2023).
12. Обзор ASP.NET [Электронный ресурс] // Microsoft. – URL: <https://learn.microsoft.com/ru-ru/aspnet/overview> (дата обращения: 26.05.2023).
13. Концепция «единого окна», Европейская экономическая комиссия ООН / Записка секретариата. – Женева: Записка, 2002. – 7с.
14. Docker: Accelerated, Containerized Application Development [Электронный ресурс] // Docker. – URL: <https://www.docker.com/> (дата обращения: 26.05.2023).
15. The Cumulative Flow Diagram (CFD) [Электронный ресурс] // Kanbanize Knowledge Base. – URL: <https://knowledgebase.kanbanize.com/hc/en-us/articles/360015034420-The-Cumulative-Flow-Diagram-CFD> (дата обращения: 22.05.2023).
16. ITSM 365 [Электронный ресурс] // ITSM365. – URL: <https://itsm365.com/> (дата обращения: 24.05.2023).
17. PostgreSQL [Электронный ресурс] // PostgreSQL. – URL: <https://www.postgresql.org/> (дата обращения: 26.05.2023).
18. ITIL [Электронный ресурс] // Axelos. – URL: <https://www.axelos.com/certifications/itil-service-management> (дата обращения: 22.05.2023).
19. COBIT [Электронный ресурс] // Isaca. – URL: <https://www.isaca.org/resources/cobit#/store/browse/detail/a2S4w000005E8ecEA> С (дата обращения: 22.05.2023).
20. ISO/IEC TS 20000-11:2021. Информационная технология. Менеджмент сервисов. Часть 11. Руководство по взаимосвязи ISO/IEC 20000-1 и структур менеджмента сервисов: ITIL [Текст]: дата введения: 2021-08-27. – М: JTC 1/SC 40, 2021. – 60 с.
21. Cycle Time vs Lead Time: The Complete Guide [Электронный ресурс] // Kanban Software for Agile Project Management. – URL:

<https://kanbanize.com/kanban-resources/kanban-software/kanban-lead-cycle-time>
(дата обращения: 22.05.2023).

22. YouTrack: управление любыми проектами для вашей компании [Электронный ресурс] // JetBrains. – URL: <https://www.jetbrains.com/ru-ru/youtrack/> (дата обращения: 24.05.2023).

23. BMC Helix ITSM [Электронный ресурс] // BMC Software. – URL: <https://www.bmc.com/it-solutions/bmc-helix-itsm.html> (дата обращения: 24.05.2023).

24. ITSM-система (IT Service Management): управление ИТ услугами и сервисами [Электронный ресурс] // SimpleOne. – URL: <https://simpleone.ru/itsm/> (дата обращения: 24.05.2023).

25. Angular [Электронный ресурс] // Angular. – URL: <https://angular.io/>
(дата обращения: 25.05.2023).

26. MongoDB: The Developer Data Platform [Электронный ресурс] // MongoDB. – URL: <https://www.mongodb.com> (дата обращения: 25.05.2023).

27. Elasticsearch Platform – Find real-time answers at scale [Электронный ресурс] // Elastic. – URL: <https://www.elastic.co/> (дата обращения: 25.05.2023).

28. Vault by HashiCorp [Электронный ресурс] // Vault. – URL: <https://www.vaultproject.io/> (дата обращения: 25.05.2023).

29. imgproxy: fast and secure on-the-fly image processing. [Электронный ресурс] // ImgProxy. – URL: <https://imgproxy.net> (дата обращения: 25.05.2023).

30. Messaging that just works — RabbitMQ [Электронный ресурс] // RabbitMQ. – URL: <https://www.rabbitmq.com/> (дата обращения: 25.05.2023).

31. GitLab CI/CD [Электронный ресурс] // GitLab. – URL: <https://docs.gitlab.com/ee/ci/> (дата обращения: 25.05.2023).

32. YARP Documentation [Электронный ресурс] // GitHub. URL: <https://microsoft.github.io/reverse-proxy/> (дата обращения: 25.05.2023).

Взаимодействие с ms-diag

На рисунке А.1 изображен доступ к таблице со всеми магазинами с возможностью фильтрации по торговым сетям и другим параметрам.

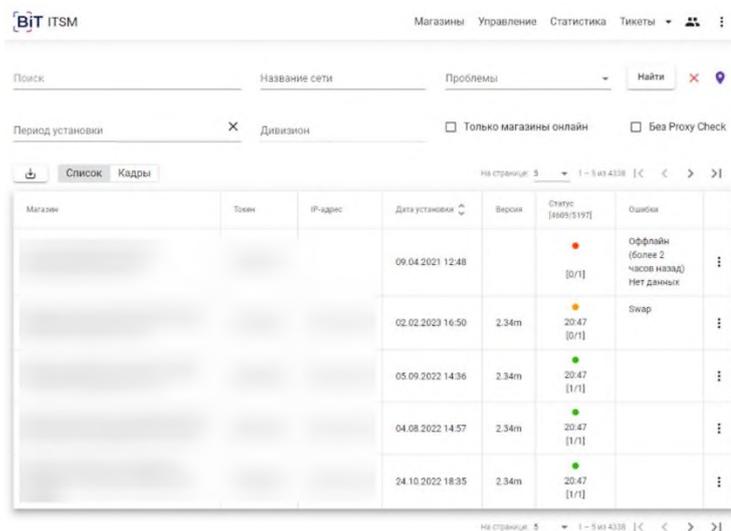


Рисунок А.1 – Таблица магазинов из сервиса ms-diag

С указанной страницы также можно посмотреть список ошибок магазина при нажатии соответствующей ячейки в колонке Ошибки (рисунок А.2).

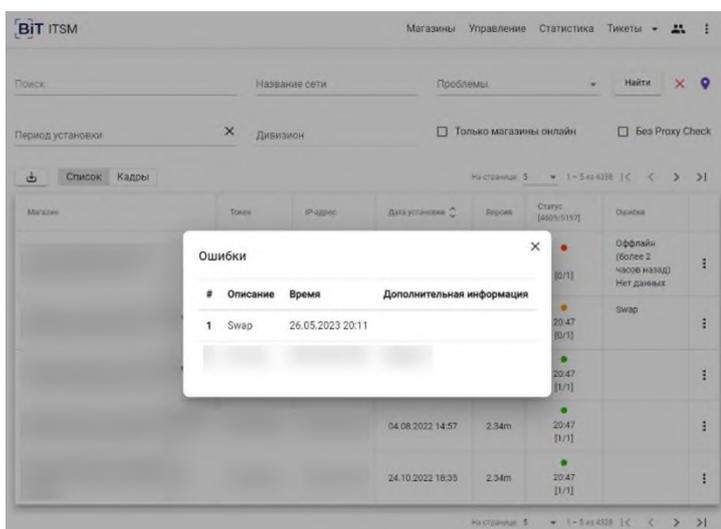


Рисунок А.2 – Ошибки магазина

На рисунке А.3 представлен дашборд отдельного магазина. На нём можно видеть статус магазина, параметры сервера, а также график связи с магазином.

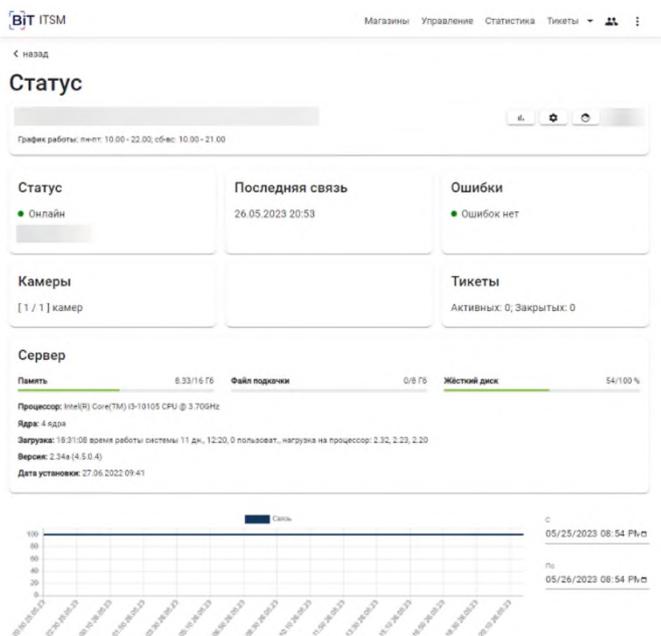


Рисунок А.3 – Дашборд магазина

На рисунке А.4 представлено построение статистики по параметру Swap.

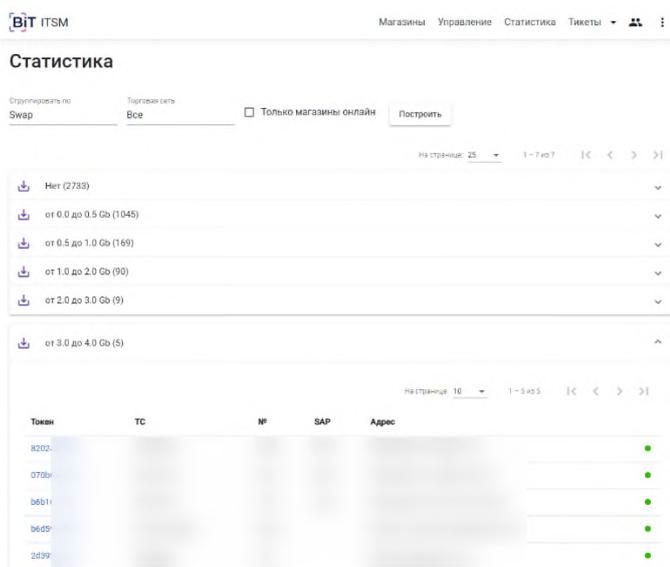


Рисунок А.4 – Статистика по Swap

Взаимодействие с ms-stats

На рисунке Б.1 показан выбор торговых сетей для пользователя с ролью администратор ТС.

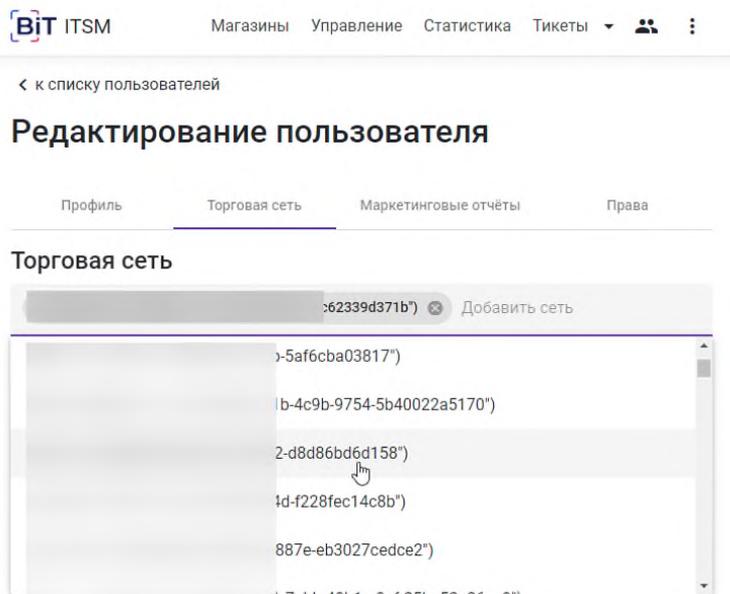


Рисунок Б.1 – Добавление торговых сетей пользователю

После выбора торговой сети для пользователя можно настроить отправку маркетинговых отчётов. На рисунке Б.2 представлен пример добавления подписки на отчеты, которые будут отправляться на почту клиенту в 10 утра каждый день.

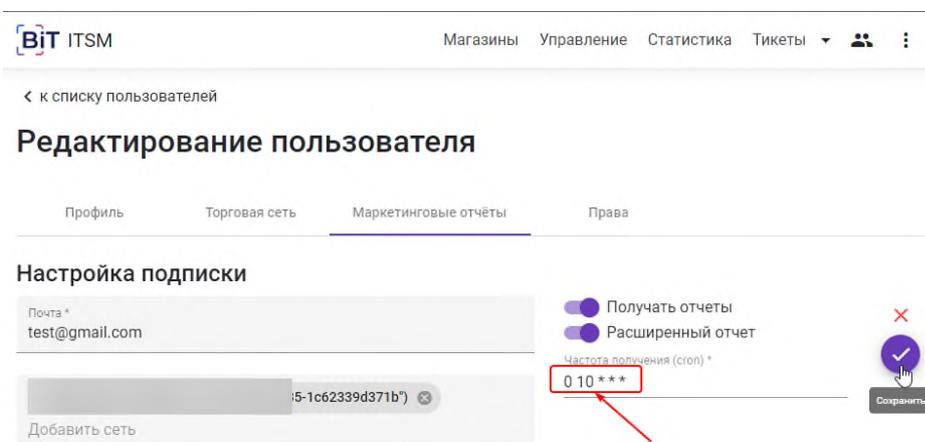


Рисунок Б.2 –Настройка отправки отчета

Продолжение ПРИЛОЖЕНИЯ Б

Затем в специальном интерфейсе можно посмотреть список всех настроенных отчетов (рисунок Б.3).

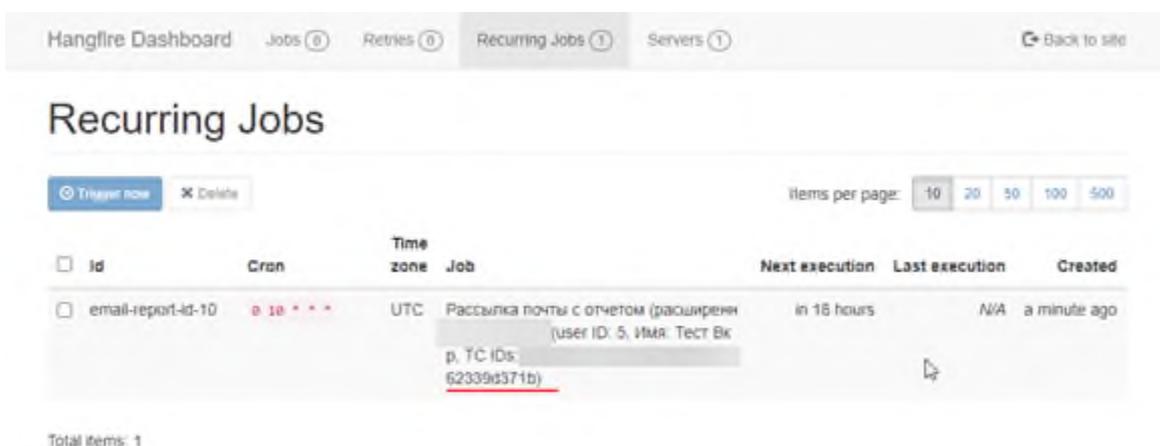


Рисунок Б.3 – Список всех отчетов

После отправки отчета в почте можно найти письмо, показанное на рисунке Б.4.

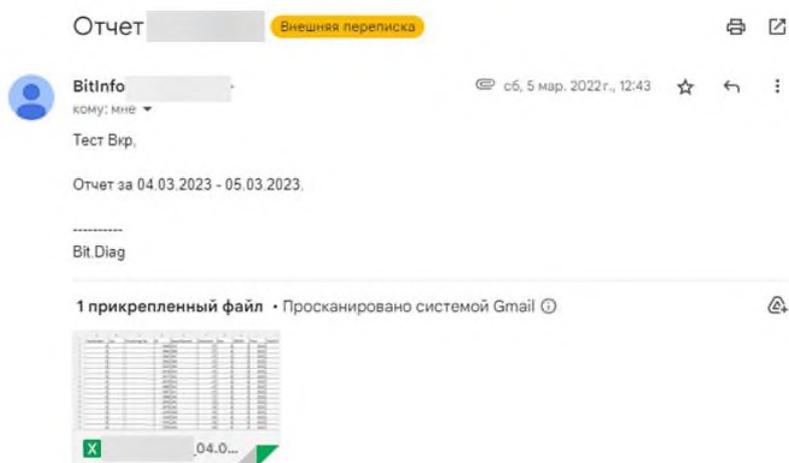


Рисунок Б.4 – Письмо с отчетом

Взаимодействие с ms-proxyccheck

На странице каждого магазина представлен раздел Proxy Check. Это список различных параметров-показателей, которые были зафиксированы на сервере клиента. Список может изменяться техническим отделом самостоятельно. На рисунке В.1 представлены некоторые параметры из этого списка.

Proxy Check	
Дата последней проверки	28.05.2023 20:00
ExportEpld	
ExportStoreId	
IsStoreClosed	false
LastPing	2023-05-20T15:33:06.638Z
RetailChainUniqueId	22fd61afbca7
hardware_cpu	Intel(R) Core(TM) i3-10100 CPU @ 3.60GHz
hardware_disk	KINGSTON SA400S37240G
hardware_memory	R948G3206U2S
hardware_summ	Intel(R) Core(TM) i3-10100 CPU @ 3.60GHz, KINGSTON SA400S37240G, R948G3206U2S
hostname	
http_502_counter	3
license_cache_error_counter	0
link_down_error_counter	0
link_speed	1000

Рисунок В.1 – Список параметров ProxyCheck

Взаимодействие с ms-frames

На главной странице со списком магазинов можно посмотреть последние 5 кадров с каждого магазина. На рисунке Г.1 представлены магазины с диагностическими кадрами.

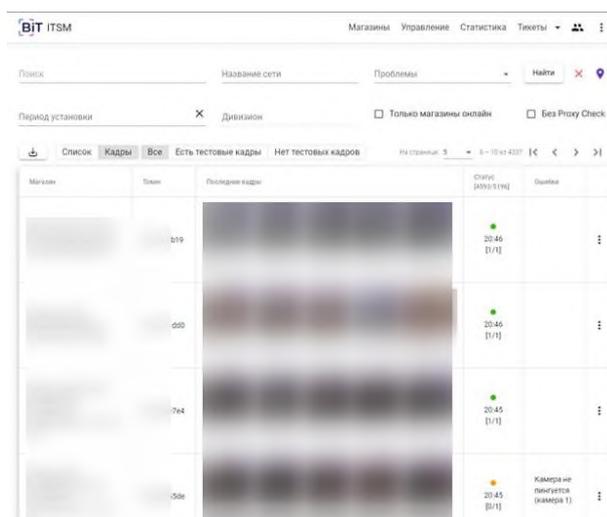


Рисунок Г.1 – Магазины с тестовыми кадрами

На странице конкретного магазина можно посмотреть список его камер и кадры с камер. Присутствует возможность фильтрации. Функционал представлен на рисунке Г.2.

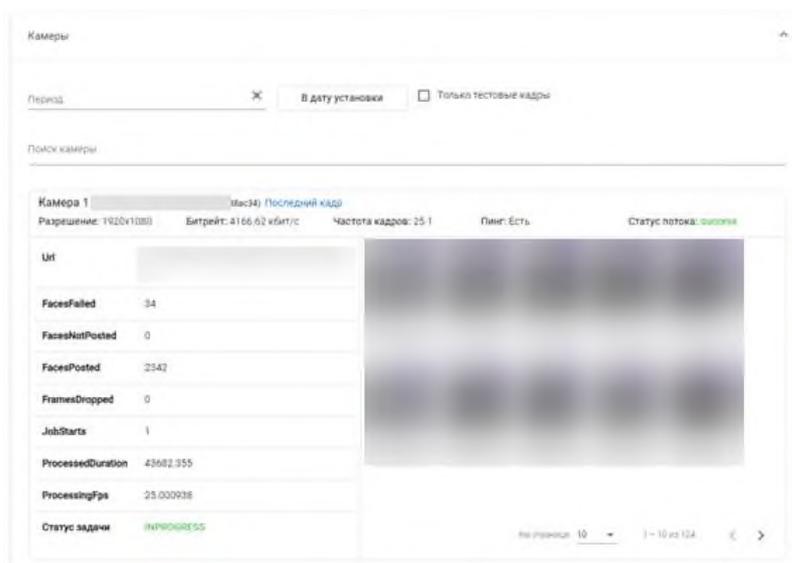


Рисунок Г.2 – Камера магазина

Взаимодействие с ms-storemanager

На рисунке Д.1 представлена таблица с торговыми сетями с возможностью фильтрации.

Логотип	Название	Количество дивизионов	Количество магазинов	Действия
		5	363	Редактировать
		0	1	Редактировать
		96	1590	Редактировать

Рисунок Д.1 – Список торговых сетей

Каждой торговой сети можно задать определенные параметры (рисунок Д.2).

Рисунок Д.2 – Параметры торговой сети

У каждой торговой сети может быть несколько дивизионов. Список дивизионов с возможностью фильтрации представлен на рисунке Д.3.

Продолжение ПРИЛОЖЕНИЯ Д

Название	Торговая сеть	Действия
Екатеринбург		Редактировать
Тест		Редактировать
Москва		Редактировать
Воронеж		Редактировать
Екатеринбург		Редактировать
Казань		Редактировать
Казахстан		Редактировать
Краснодарский Край		Редактировать
Москва		Редактировать
Нижний Новгород		Редактировать

Рисунок Д.3 – Список дивизионов торговых сетей

Каждый дивизион также имеет свой список настроек (рисунок Д.4).

BIT ITSM

Магазины Управление Статистика Тикеты

← НАЗАД

Редактирование дивизиона

Торговая сеть

Имя
Екатеринбург

Описание
Секретный по умолчанию

Это поле
Monitoring

Рисунок Д.4 – Настройки дивизиона

Торговые сети имеют свой список магазинов. Кроме этого, каждый магазин привязан к конкретному дивизиону. Список магазинов представлен на рисунке Д.5. На странице также доступна фильтрация магазинов.

Продолжение ПРИЛОЖЕНИЯ Д

Торговая сеть	Номер магазина	Название магазина	Город	Адрес	Товар	Действия
	1		Екатеринбург		b3c	Редактировать
	2		Екатеринбург		сd3	Редактировать
			Екатеринбург		я8b	Редактировать
			Екатеринбург		638	Редактировать
			Екатеринбург		4c4	Редактировать
	кх9906		Москва		96	Редактировать
	405		Москва		072	Редактировать
			Москва		6dF	Редактировать
			деревня Ладанка		:12	Редактировать
	00001		Химки		3F8	Редактировать

Рисунок Д.5 – Список магазинов

На странице магазина имеется несколько вкладок с настройками:

- общее,
- камеры,
- доступ,
- дополнительно.

На рисунке Д.6 представлены общие настройки магазина.

Общие Камеры Доступ Дополнительно

Товар: 1c:12 Дивизион*: Москва

Номер магазина: SAP: Название:

Страна*: РФ Адрес: РФ, Московская область, деревня Ладанка, влб. х2

Регион*: Московская область

Определить на карте

Рисунок Д.6 – Общие настройки магазина

Продолжение ПРИЛОЖЕНИЯ Д

На рисунке Д.7 представлены настройки камер магазина.

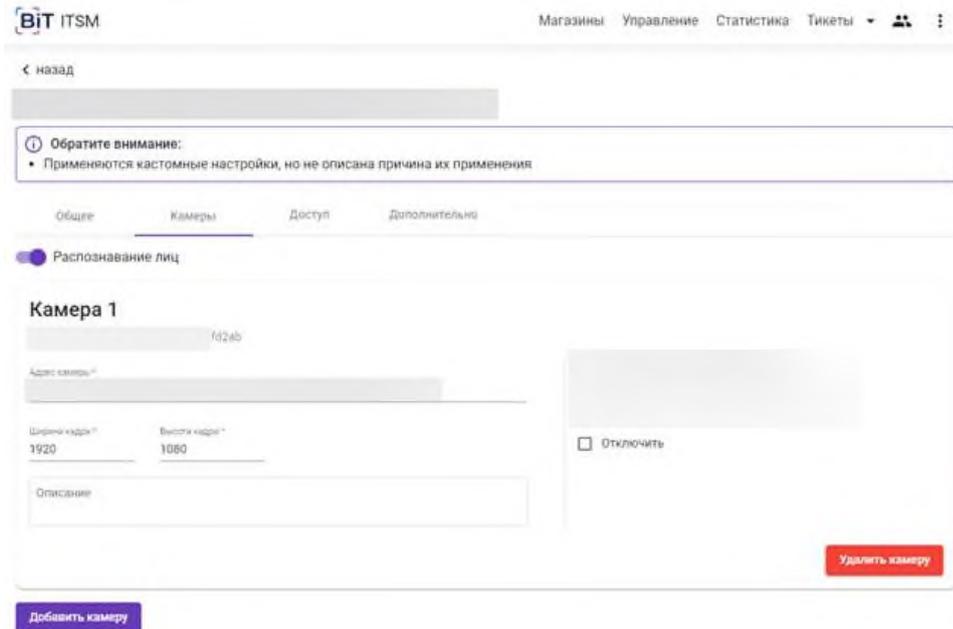


Рисунок Д.7 – Настройки камер магазина

На рисунке Д.8 представлены настройки доступа к магазину.

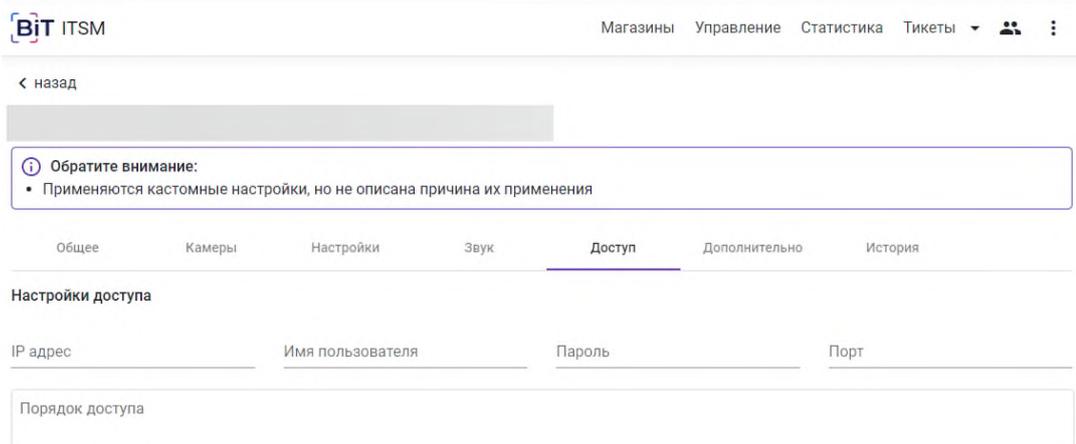


Рисунок Д.8 – Настройки доступа к магазину

На рисунке Д.9 представлены дополнительные настройки магазина.

Продолжение ПРИЛОЖЕНИЯ Д

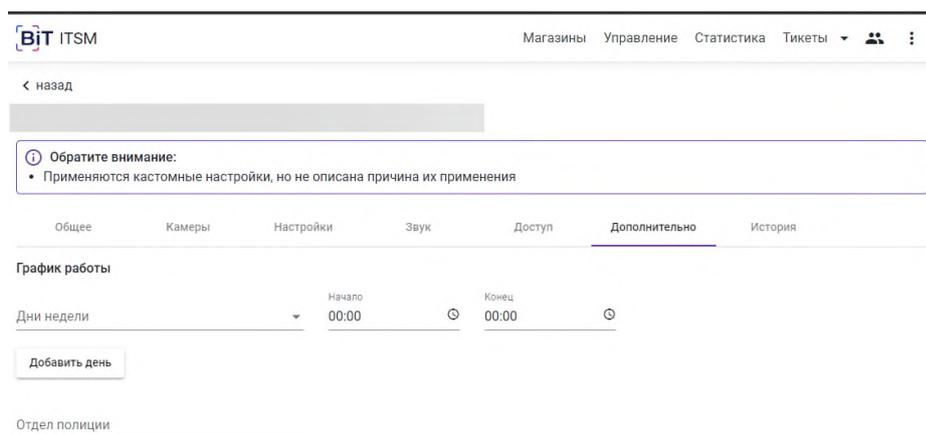


Рисунок Д.9 – Дополнительные настройки магазина

Взаимодействие с ms-tickets

На рисунке Е.1 представлена канбан-доска для работы с тикетами.

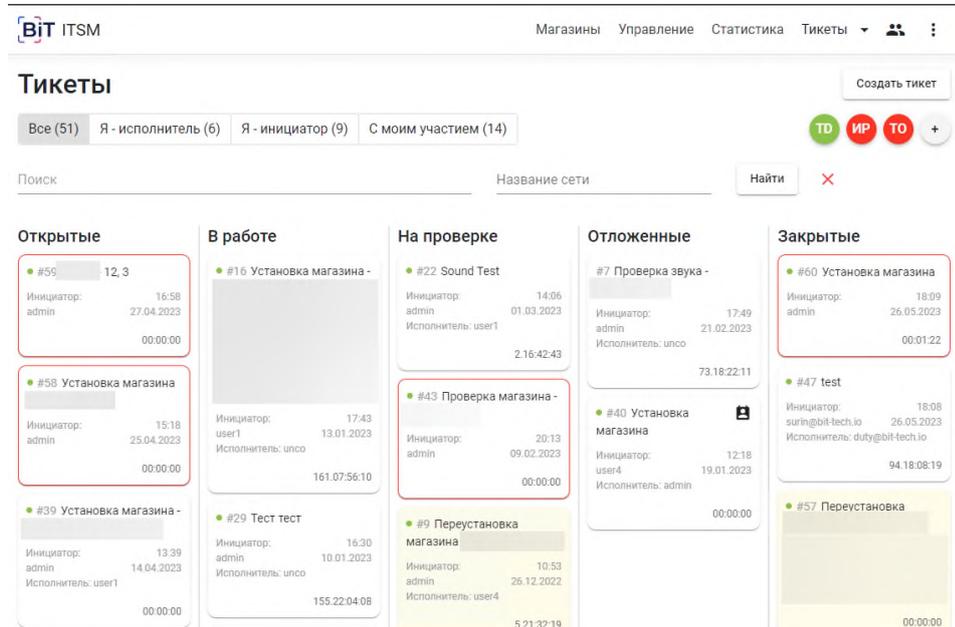


Рисунок Е.1 – Канбан-доска с тикетами

На рисунке Е.2 показан интерфейс конкретного тикета.

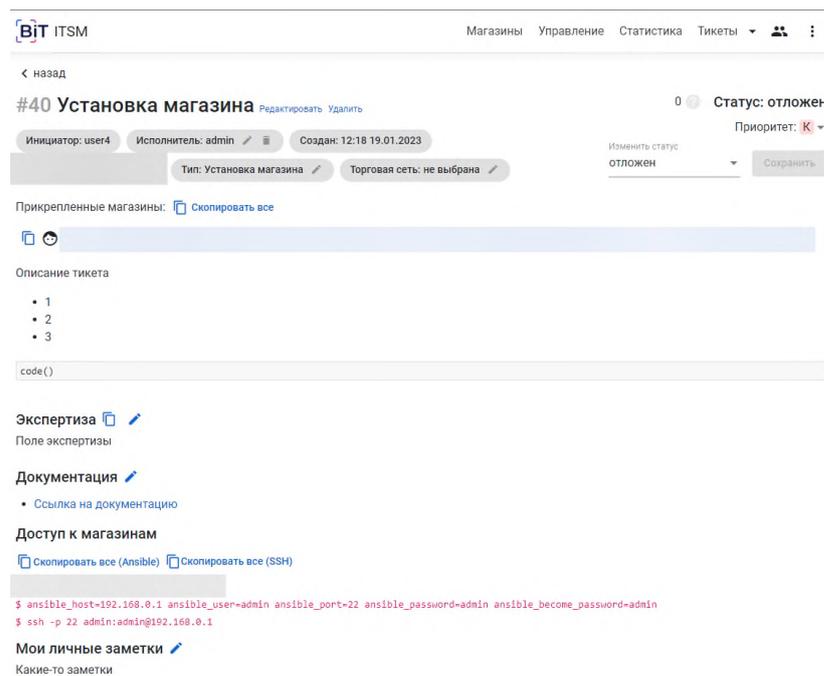


Рисунок Е.2 – Описание тикета

Продолжение ПРИЛОЖЕНИЯ Е

Кроме этого, тикеты можно просматривать в виде списка, а не канбан-доски. Пример списка тикетов представлен на рисунке Е.3. На этой же странице указаны основные метрики эффективности работы сотрудников (справа от тикета).

The screenshot displays the 'Тикеты' (Tickets) section of the ITSM application. At the top, there are navigation links: 'Магазины', 'Управление', 'Статистика', 'Тикеты', and a user profile icon. Below the navigation is a 'Создать тикет' (Create ticket) button. The main area features a filter bar with 'Все (51)', 'Я - исполнитель (6)', 'Я - инициатор (9)', and 'С моим участием (15)'. There are also three circular status indicators: 'TD' (green), 'IP' (red), and 'TO' (red), along with a plus sign. A search bar is present with the placeholder text 'Поиск' and 'Название сети'. Below the search bar are several filter buttons: 'Нулевой Cycle Time', 'Большой Cycle Time', 'Большой Lead Time', 'Тикеты для ревью', and 'Недавно обновлённые тикеты'. Sorting options are available: 'Сортировать по' (Sort by) with 'Номер тикета' (Ticket number) selected, and 'Направление' (Direction) with 'По возрастанию' (Ascending) selected. A 'Найти' (Find) button with a close icon is also present. The ticket list contains four entries:

Иконка	Идентификатор	Статус	Инициатор	Исполнитель	Время	Метрики
К	#1 Название Test	закрыт	unso	user1	8д. 15ч. 30мин. 42сек.	1 0ч. 28мин. 16сек. 0ч. 28мин. 16сек. 92д. 1ч. 12мин. 17сек.
В	#2 Test	закрыт	admin	[redacted]	6д. 0ч. 8мин. 55сек.	1 0ч. 0мин. 15сек. 29д. 11ч. 32мин. 24сек. 86д. 14ч. 56мин. 25сек.
О	#3 324234	закрыт	user1	user1231	1д. 2ч. 15мин. 46сек.	2 3ч. 15мин. 10сек. 3ч. 15мин. 10сек. 54д. 23ч. 5мин. 56сек.
В	#4 Тест ТС	в работе	admin	user5	206д. 10ч. 53мин. 55сек.	1 Нет данных

Рисунок Е.3 – Список тикетов

Взаимодействие с ms-identity

На рисунке Ж.1 представлен список пользователей. Пользователя можно удалять или блокировать. Также доступен функционал фильтрации.

Логин	ФИО	Почта	Торговые сети	Роль	Статус
admin			Администратор системы	SystemAdmin	Активный
				Ops	Активный
			Администратор системы	SystemAdmin	Активный
				Ops	Активный
			Администратор системы	SystemAdmin	Активный
test333@foo				RetailAdmin	Активный
test_identity			Администратор системы	SystemAdmin	Активный

Рисунок Ж.1 – Список пользователей

На странице пользователя можно отредактировать информацию о нём (рисунок Ж.2).

ViT ITSM Магазины

< к списку пользователей

Редактирование пользователя

Профиль | Маркетинговые отчёты | Права

Пользователь (SystemAdmin)

Логин: admin | Почта: [redacted]

Фамилия * | Имя * | Отчество

Сменить пароль

Новый пароль | Подтвердить пароль

Сохранить

Рисунок Ж.2 – Информация о пользователе

Продолжение ПРИЛОЖЕНИЯ Ж

На той же странице можно редактировать роль и права пользователя (рисунок Ж.3).

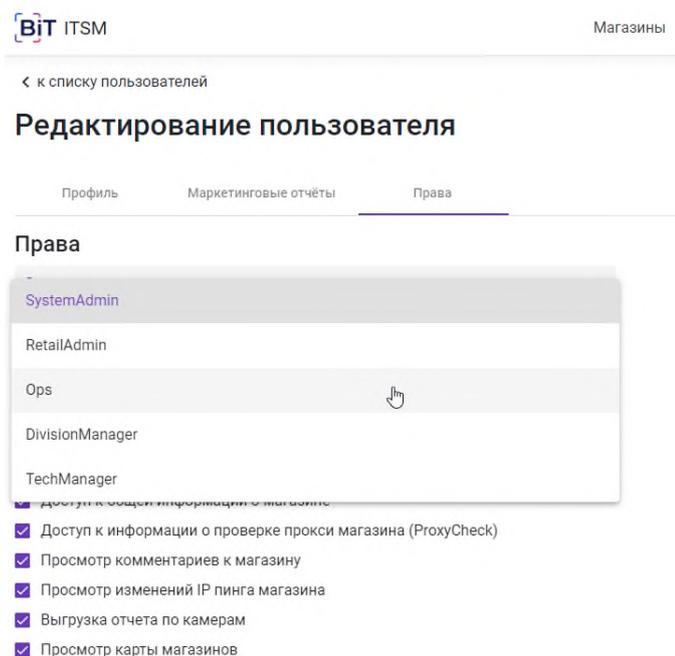


Рисунок Ж.3 – Редактирование прав пользователя

Кроме управления пользователями также присутствует управление ролями и правами ролей. Страница с ролями представлена на рисунке Ж.4.

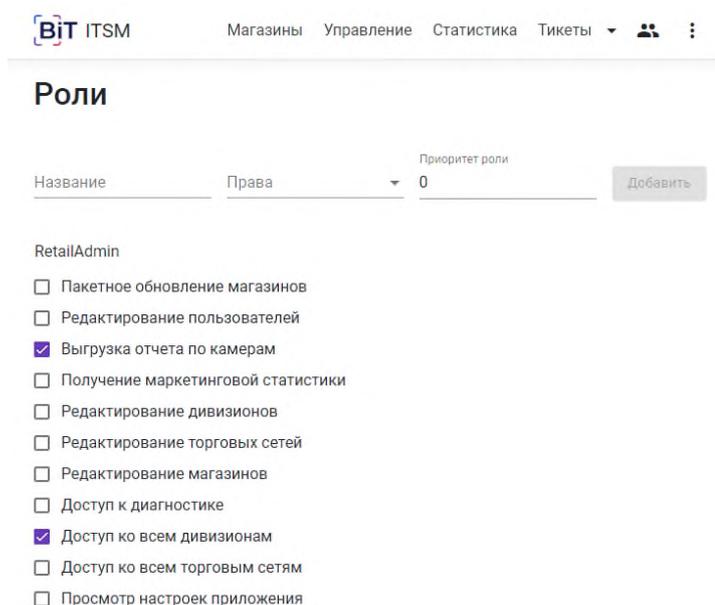


Рисунок Ж.4 – Редактирование ролей