

**Министерство образования и науки Российской Федерации**  
федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Инженерная школа информационных технологий и робототехники  
Направление подготовки 09.04.02 Информационные системы и технологии  
Отделение информационных технологий

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

Тема работы
<b>Проектирование и разработка модуля анализа и оценки эффективности технического обслуживания системы ESPRIT</b>

УДК 004.415.2:004.422.833

**Студент**

Группа	ФИО	Подпись	Дата
8ИМ81	Черепанов Иван Сергеевич		

**Руководитель**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Савельев Алексей Олегович	К.Т.Н.		

**КОНСУЛЬТАНТЫ:**

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН	Конотопский Владимир Юрьевич	К.Э.Н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ООД	Горбенко Михаил Владимирович	К.Т.Н.		

**ДОПУСТИТЬ К ЗАЩИТЕ:**

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Савельев Алексей Олегович	К.Т.Н.		

## ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП

Код результатов	Результат обучения (выпускник должен быть готов)
<b>Общепрофессиональные компетенции</b>	
P1	Воспринимать и самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте.
P2	Владеть и применять методы и средства получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе в глобальных компьютерных сетях.
P3	Демонстрировать культуру мышления, способность выстраивать логику рассуждений и высказываний, основанных на интерпретации данных, интегрированных из разных областей науки и техники, выносить суждения на основании неполных данных, анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями.
P4	Анализировать и оценивать уровни своих компетенций в сочетании со способностью и готовностью к саморегулированию дальнейшего образования и профессиональной мобильности. Владеть, по крайней мере, одним из иностранных языков на уровне социального и профессионального общения, применять специальную лексику и профессиональную терминологию языка.
<b>Профессиональные компетенции</b>	
P5	Разрабатывать стратегии и цели проектирования, критерии эффективности и ограничения применимости, новые методы, средства и технологии проектирования геоинформационных систем (ГИС) или промышленного программного обеспечения.
P6	Планировать и проводить теоретические и экспериментальные исследования в области создания интеллектуальных ГИС и ГИС технологии или промышленного программного обеспечения с использованием методов системной инженерии.
P7	Осуществлять авторское сопровождение процессов проектирования, внедрения и сопровождения ГИС и ГИС технологий или промышленного программного обеспечения с использованием методов и средств системной инженерии, осуществлять подготовку и обучение персонала.
P8	Формировать новые конкурентоспособные идеи в области теории и практики ГИС и ГИС технологий или системной инженерии программного обеспечения. Разрабатывать методы решения нестандартных задач и новые методы решения традиционных задач. Организовывать взаимодействие коллективов, принимать управленческие решения, находить компромисс между различными требованиями как при долгосрочном, так и при краткосрочном планировании.
<b>Общекультурные компетенции</b>	
P9	Использовать на практике умения и навыки в организации исследовательских, проектных работ и профессиональной эксплуатации современного оборудования и приборов, в управлении коллективом.
P10	Свободно пользоваться русским и иностранным языками как средством делового общения.
P11	Совершенствовать и развивать свой интеллектуальный и общекультурный уровень. Проявлять инициативу, в том числе в ситуациях риска, брать на себя всю полноту ответственности.
P12	Демонстрировать способность к самостоятельному обучению новым методам исследования, к изменению научного и научно-производственного профиля своей профессиональной деятельности, способность самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности, способность к педагогической деятельности.

**Министерство образования и науки Российской Федерации**  
 федеральное государственное автономное образовательное учреждение  
 высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
 ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Школа информационных технологий и робототехники  
 Направление подготовки 09.04.02 Информационные системы и технологии  
 Отделение информационных технологий

УТВЕРЖДАЮ:  
 Руководитель ООП  
 \_\_\_\_\_ Савельев А.О.  
 (Подпись) (Дата) (Ф.И.О.)

**ЗАДАНИЕ**

**на выполнение выпускной квалификационной работы в форме:**

Магистерской диссертации
--------------------------

Студент:

<b>Группа</b>	<b>ФИО</b>
8ИМ81	Черепанов Иван Сергеевич

Тема работы:

<b>Проектирование и разработка модуля анализа и оценки эффективности технического обслуживания системы ESPRIT</b>
Утверждена приказом директора (дата, номер)

Срок сдачи студентом выполненной работы:	05.06.2020
--	------------

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ:**

<b>Исходные данные к работе</b>	Требуется разработать модуль анализа и оценки эффективности для обеспечения информационной поддержки процессов управления техническим обслуживанием системы ESPRIT. Функциональной особенностью модуля является возможность автоматизированной оценки эффективности технического обслуживания путем расчета значений соответствующих бизнес-метрик и их визуализации.
<b>Перечень подлежащих исследованию, проектированию и разработке вопросов</b>	<ol style="list-style-type: none"> <li>1) Изучение предметной области;</li> <li>2) Обзор существующих решений;</li> <li>3) Выявление и анализ требований к программному модулю;</li> <li>4) Постановка и распределение задач, обеспечение командной разработки;</li> <li>5) Получение необходимых данных для работы модуля;</li> <li>6) Выбор технологий разработки и используемых библиотек;</li> <li>7) Проектирование и реализация клиентской части программного модуля;</li> </ol>

	8) Разработка алгоритма получения и постобработки данных для отображения требуемых метрик в виде диаграмм; 9) Проектирование и реализация серверной части программного модуля; 10) Реализация функционала по интеграции модуля с приложением Esprit Web; 11) Тестирование программного модуля; 12) Определение стоимости и сроков разработки проекта; 13) Рассмотрение условий труда исполнителей проекта.
<b>Перечень графического материала</b>	1) EPC диаграмма процесса изготовления детали с использованием CAD/CAM системы. 2) UML-диаграмма состояний запроса. 3) UML-диаграмма архитектуры сервисов Esprit Web до локализации. 4) UML-диаграмма переработанной архитектуры сервисов. 5) UML-диаграмма классов модели данных. 6) Диаграмма компонентов серверной части проекта. 7) UML-диаграмма последовательности процесса авторизации. 8) Диаграмма компонентов клиентской части. 9) UML-диаграмма классов серверной части. 10) Блок-схема алгоритма получения и обработки данных требуемых для отображения диаграмм. 11) Диаграмма Ганта
<b>Консультанты по разделам выпускной квалификационной работы</b>	
<b>Раздел</b>	<b>Консультант</b>
Социальная ответственность	Горбенко Михаил Владимирович
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Конотопский Владимир Юрьевич
Раздел на английском языке	Маркова Наталия Александровна
<b>Названия разделов, которые должны быть написаны на русском и иностранном языках:</b>	
Доработка Esprit Web для получения необходимых данных	

<b>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</b>	01.03.2020
---	------------

**Задание выдал руководитель:**

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
доцент ОИТ	Савельев Алексей Олегович	К.Т.Н.		

**Задание принял к исполнению студент:**

<b>Группа</b>	<b>ФИО</b>	<b>Подпись</b>	<b>Дата</b>
8ИМ81	Черепанов Иван Сергеевич		

**Министерство образования и науки Российской Федерации**  
федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Школа информационных технологий и робототехники  
Направление подготовки 09.04.02 Информационные системы и технологии  
Уровень образования Магистратура  
Отделение информационных технологий  
Период выполнения осенний / весенний семестр 2019/2020 учебного года

Форма представления работы:

Магистерская диссертация

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН  
выполнения выпускной квалификационной работы**

Срок сдачи студентом выполненной работы:	05.06.2020
--	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
26.11.2019	Раздел 1. Анализ предметной области	15
04.12.2019	Раздел 2. Постановка задач	10
31.01.2020	Раздел 3. Доработка Esprit Web для получения необходимых данных	20
04.05.2020	Раздел 4. Разработка веб-приложения	30
06.05.2020	Раздел 5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	15
31.05.2020	Раздел 6. Социальная ответственность	10

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент ОИТ	Савельев Алексей Олегович	К.Т.Н.		

**СОГЛАСОВАНО:**

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
доцент ОИТ	Савельев Алексей Олегович	К.Т.Н.		

## ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8ИМ81	Черепанов Иван Сергеевич

Школа	ИШИТР	Отделение (НОЦ)	ОИТ
Уровень образования	Магистратура	Направление/специальность	Информационные системы и технологии

Тема ВКР:

<b>Проектирование и разработка модуля анализа и оценки эффективности технического обслуживания системы ESPRIT</b>	
<b>Исходные данные к разделу «Социальная ответственность»:</b>	
1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Объектом исследования является процесс оказания технической поддержки системы Esprit Web. Целью работы является разработка модуля анализа и оценки эффективности для обеспечения информационной поддержки процессов управления техническим обслуживанием системы ESPRIT.
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
<b>1. Правовые и организационные вопросы обеспечения безопасности:</b> – Специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны); – Правовые нормы трудового законодательства; – Организационные мероприятия при компоновке рабочей зоны.	– СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы – ГОСТ 12.2.032-78 ССБТ Рабочее место при выполнении работ сидя. Общие эргономические требования – ГОСТ Р 50923-96. Дисплеи. Рабочее место оператора. Общие эргономические требования и требования к производственной среде. Методы измерения – ГОСТ 12.1.005-88 с изм. №1 от 2000 г., СанПиН 2.2.4.548-96. Общие санитарно-гигиенические требования к воздуху рабочей зоны – Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 24.04.2020)
<b>2. Производственная безопасность:</b> 2.1. Анализ выявленных вредных и опасных факторов 2.2. Обоснование мероприятий по снижению их воздействия	– Отклонение показателей микроклимата; – Недостаточная освещённость рабочей зоны; – Повышенный уровень шума; – Повышенный уровень электромагнитных излучений; – Нервно-психические перегрузки – Электробезопасность
<b>3. Экологическая безопасность:</b>	Воздействие на окружающую среду вредными и опасными отходами.
<b>4. Безопасность в чрезвычайных ситуациях:</b>	Наиболее типичная ЧС, которая может возникнуть в процессе разработки – пожар в здании, в т.ч. из-за короткого замыкания электропроводки.

	Требуется принять ряд мер, предупреждающих ЧС. При возникновении ЧС, требуется следовать протоколу эвакуации из здания, а также вызвать службы для ликвидации последствий ЧС.
--	---

Дата выдачи задания для раздела по линейному графику	01.03.2020
--	------------

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ООД	Горбенко Михаил Владимирович	к.т.н.		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8ИМ81	Черепанов Иван Сергеевич		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И  
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

<b>Группа</b>	<b>ФИО</b>
8ИМ81	Черепанов Иван Сергеевич

<b>Школа</b>	<b>ИШИТР</b>	<b>Отделение школы (НОЦ)</b>	<b>ОИТ</b>
<b>Уровень образования</b>	Магистратура	<b>Направление/специальность</b>	Информационные системы и технологии

**Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:**

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Использовать действующие ценники и договорные цены на потребленные материальные и информационные ресурсы, а также величину тарифа на эл. энергию
2. Нормы и нормативы расходования ресурсов	–
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	Действующие ставки единого социального налога и НДС

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

1. Оценка коммерческого и инновационного потенциала НТИ	Дать характеристику существующих и потенциальных потребителей (покупателей) результатов ВКР, ожидаемых масштабов их использования
2. Разработка устава научно-технического проекта	Разработать проект такого устава в случае, если для реализации результатов ВКР необходимо создание отдельной организации или отдельного структурного подразделения (возможно временного) внутри существующей организации
3. Планирование процесса управления НТИ: структура и график проведения, бюджет, риски и организация закупок	Построение плана-графика выполнения ВКР, составление соответствующей сметы затрат, расчет цены результата ВКР.
4. Определение ресурсной, финансовой, экономической эффективности	Оценка экономической эффективности использования результатов ВКР, характеристика других видов эффекта

**Перечень графического материала** (с точным указанием обязательных чертежей):

1. Оценка конкурентоспособности технических решений
2. Матрица SWOT
3. График проведения и бюджет НТИ
4. Оценка ресурсной, финансовой и экономической эффективности НТИ

<b>Дата выдачи задания для раздела по линейному графику</b>	
---	--

**Задание выдал консультант:**

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
Доцент ОСГН	Конотопский Владимир Юрьевич	к.э.н.		25.02.2020 г.

**Задание принял к исполнению студент:**

<b>Группа</b>	<b>ФИО</b>	<b>Подпись</b>	<b>Дата</b>
8ИМ81	Черепанов Иван Сергеевич		



## РЕФЕРАТ

Выпускная квалификационная работа содержит 160 страниц, 53 рисунка, 33 таблицы, 30 источников, 6 приложений.

Ключевые слова: веб-приложение, ASP.Net Core, C#, Chart.js, визуализация, техническая поддержка, система автоматизированного проектирования (САПР), Esprit.

Объектом исследования является процесс оказания технической поддержки системы Esprit Web.

Целью работы является разработка модуля анализа и оценки эффективности для обеспечения информационной поддержки процессов управления техническим обслуживанием системы ESPRIT.

В результате работы были получены необходимые данные для работы модуля после доработки приложения Esprit Web. Был спроектирован и реализован программный модуль, представляющий собой Asp.Net Core веб-приложение. Разработанный модуль был внедрен и используется по назначению.

Область применения: анализ и поддержка бизнес-процесса осуществления технической поддержки.

Экономическая эффективность и значимость работы для предприятия, где внедрена разработка, обусловлена снижением трудозатрат на получение и оформление нужной информации, улучшения качества и эффективности обработки запросов в техническую поддержку.

В будущем планируется использовать разработанный модуль для отображения метрик по другим бизнес-процессам предприятия, а также расширить графическую составляющую новыми шаблонами диаграмм и графиков.

## Содержание

Условные обозначения .....	13
Введение .....	15
1. Анализ предметной области .....	16
1.1. Описание предметной области.....	16
1.2. Описание процесса осуществления технической поддержки .....	20
2. Постановка задач .....	26
2.1. Распределение задач .....	27
2.2. Используемые средства командной разработки .....	28
2.3. Работы по обеспечению требуемых данных для диаграмм.....	32
3. Доработка Esprit Web для получения необходимых данных .....	37
3.1. Получение пользовательских оценок работы службы поддержки .....	37
3.2. Сохранение временных показателей для ведения статистики .....	45
4. Разработка веб–приложения.....	50
4.1. Анализ модели данных .....	50
4.2. Выбор стека используемых технологий .....	52
4.3. Анализ и выбор компонентов визуализации диаграмм и фильтров .....	53
4.4. Проектирование структуры приложения .....	55
4.4.1. Проектирование архитектуры клиентской части .....	55
4.4.2. Проектирование пользовательского интерфейса .....	58
4.4.3. Проектирование архитектуры серверной части.....	67
4.5. Интеграция с приложением Esprit Web .....	70
4.5.1. Реализация механизма междоменной авторизации .....	70
4.5.2. Обеспечение интеграции баз данных .....	71
4.6. Программная реализация серверной части .....	72
4.7. Программная реализация клиентской части .....	78
4.8. Тестирование приложения .....	87
5. Финансовый менеджмент и ресурсоэффективность.....	91
5.1. Оценка коммерческого потенциала и перспективности реализации модуля с позиции ресурсоэффективности и ресурсосбережения .....	91
5.1.1. Потребители результатов реализации программного модуля.....	91
5.1.2. Анализ конкурентных программных решений .....	92
5.1.3. SWOT-анализ .....	94
5.2. Определение возможных альтернатив.....	94

5.3. Организация и планирование работ.....	96
5.4. Продолжительность этапов работ.....	97
5.5. Расчет сметы затрат на выполнение проекта .....	100
5.5.1. Расчет затрат на материалы.....	101
5.5.2. Расчет заработной платы.....	101
5.5.3. Расчет затрат на социальный налог .....	103
5.5.4. Расчет затрат на электроэнергию.....	103
5.5.5. Расчет амортизационных расходов.....	105
5.5.6. Расчет прочих расходов.....	106
5.5.7. Расчет общей себестоимости разработки.....	106
5.6. Экономическая эффективность проекта.....	107
5.7. Выводы по разделу .....	108
6. Социальная ответственность.....	109
6.1. Правовые и организационные вопросы обеспечения безопасности.....	109
6.1.1. Правовые нормы трудового законодательства .....	109
6.1.2. Организационные вопросы обеспечения безопасности .....	110
6.2. Производственная безопасность .....	112
6.2.1. Показатели микроклимата.....	113
6.2.2. Освещенность рабочей зоны .....	114
6.2.3. Уровень шума в помещениях.....	117
6.2.4. Уровень электромагнитных излучений .....	118
6.2.5. Нервно-психические перегрузки.....	119
6.2.6. Поражение электрическим током .....	120
6.2.7. Мероприятия по снижению опасного и вредного воздействия .....	121
6.3. Экологическая безопасность.....	122
6.4. Безопасность в чрезвычайных ситуациях.....	123
6.5. Выводы по разделу .....	125
Заключение.....	126
Приложение А .....	127
Приложение Б.....	139
Приложение В .....	151
Приложение Г.....	152
Приложение Д .....	154
Приложение Е .....	157

Список литературы ..... 158

## Условные обозначения

ЧПУ	Числовое программное управление
САПР(CAD)	Система автоматизированного проектирования
CAM	Автоматизированная система, либо модуль автоматизированной системы, предназначенный для подготовки управляющих программ для станков с ЧПУ
UML	Unified Modeling Language – унифицированный язык моделирования
БД	База данных
СУБД	Система управления базами данных
QA	Quality assurance – обеспечение качества
КПИ	Ключевые показатели эффективности
ORM	Object-Relational Mapping – объектно-реляционное отображение
.NET	.NET Framework – программная платформа
SMTP	Simple Mail Transfer Protocol – простой протокол передачи почты
ООП	Объектно-ориентированное программирование
ASP.NET	Active Server Pages для .NET – платформа разработки веб-приложений
MVC	Архитектура веб-приложений Model-View-Controller
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
MS	Microsoft
JS	JavaScript
SPA	Одностраничное приложение (англ. Single Page Application)
AJAX	Asynchronous JavaScript and XML – технология обращения к серверу без перезагрузки страницы
SEO	Поисковая оптимизация (англ. Search Engine Optimization)

UI	User Interface – пользовательский интерфейс
IE	Internet Explorer – браузер
DI	Dependency Injection – внедрение зависимости

## Введение

Для достижения успеха на рынке в условиях конкуренции компании вынуждены работать над повышением качества продукции, снижением ее себестоимости, сокращением сроков ее производства.

На современных предприятиях, производящих и обрабатывающих различные изделия, для решения вышеописанных задач используются станки с ЧПУ в связке с системами автоматизированного проектирования и производства (CAD, CAM системы). Данные системы являются сложными в освоении инструментами, которые требуют глубоких технических знаний от инженеров.

Поэтому для CAD/CAM систем на рынке критически важно наличие качественной технической поддержки, которая позволит пользователям оперативно получать помощь, инструкции и советы по эксплуатации, а также сообщать о найденных ошибках в системе для их своевременного устранения.

Компания «DP Technology», разработчик CAD/CAM системы Esprit, работает над повышением качества работы отдела технической поддержки и собирает различную статистику по КПИ для оптимизации бизнес-процессов.

Ранее процесс сбора этой информации не был достаточно автоматизирован – требовалось участие специалиста для получения и оформления нужной информации. В связи с этим, менеджментом была поставлена задача о разработке веб-приложения для автоматического расчета и отображения нужных метрик.

Целью данной работы является проектирование и разработка модуля анализа и оценки эффективности технического обслуживания системы Esprit.

## **1. Анализ предметной области**

### **1.1. Описание предметной области**

Системы автоматического проектирования представляют собой сложный комплекс программных, технологических, технических и информационных инструментов, основной задачей которых является автоматизация процессов проектирования. Реализация данной задачи позволяет существенно повысить эффективность работы инженеров и снизить временные и финансовые издержки на разработку и внедрение новой продукции.

Предназначение САМ системы заключается в автоматизации процессов подготовки и управления производством. Интеграция системы со станками ЧПУ позволяет автоматизировать процесс обработки изделий и создавать программы для работы ЧПУ. Внедрение такой системы в производство позволяет изготавливать сложнопрофильные продукты при одновременном сокращении цикла их производства. В САМ системах используются модели и сборки, созданные в САД системах. Модели и сборки требуются для формирования траекторий перемещения инструментов при создании управляющих программ, по которым станки с ЧПУ создают физические детали.

Esprit – полнофункциональная САД/САМ система, разработанная компанией «DP Technology» для программирования, оптимизации и симуляции обработки на станках с ЧПУ, поддерживающая производственный процесс целиком, начиная от САД файла и заканчивая готовой деталью. Сертифицированные постпроцессоры обеспечивают получение оптимального G-кода, а программный интерфейс приложения открывает широкие возможности для автоматизации труда технолога-программиста (рис. 1) [21].





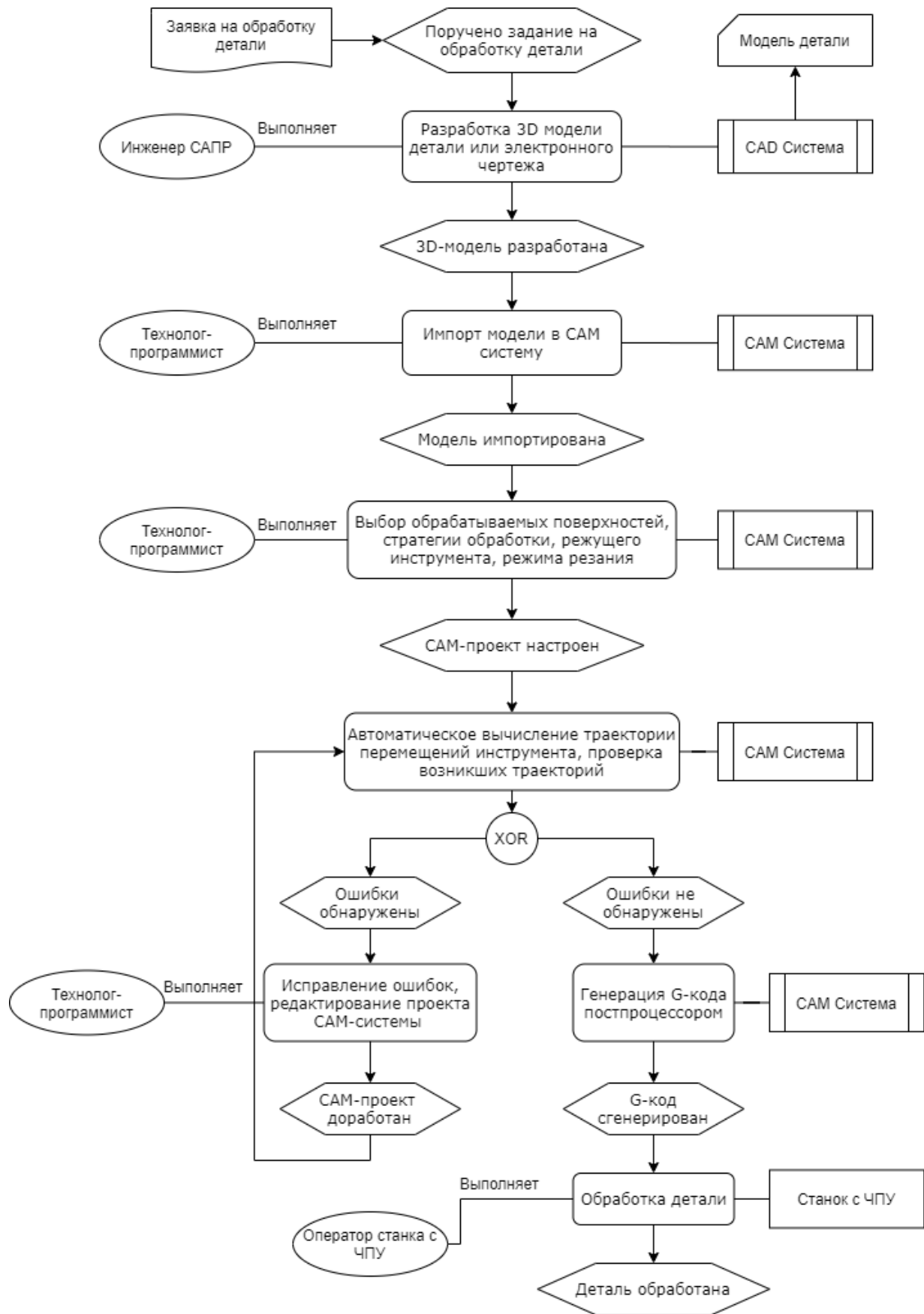


Рисунок 2 – EPC-диаграмма процесса изготовления детали с использованием CAD/CAM системы

На представленной схеме можно выделить следующие основные этапы работы:

1 Разработка 3D модели детали или ее электронного чертежа внутри CAD системы.

2 Импорт 3D модели детали или ее электронного чертежа в САМ систему. После импорта технолог-программист определяет поверхности и геометрические элементы, необходимые для обработки, делает выбор стратегии обработки, режущего инструмента и задает режим резания. Система вычисляет траектории перемещений инструмента.

3 Визуальная проверка возникших траекторий в САМ системе. Программист имеет возможность исправить ошибки, которые могут быть обнаружены на этом этапе, вернувшись к предыдущему.

4 Генерация управляющей программы (G-кода) САМ системой. Такой код создается с помощью постпроцессора, который позволяет учесть в управляющей программе характеристики определенного станка или системы ЧПУ.

5 Обработка заготовки станком с ЧПУ под руководством оператора станка.

Работая по рассмотренной выше схеме, CAD/CAM система Esprit помогает автоматизировать производство на предприятиях. Компания «DP Technology» получает основной доход от продажи лицензий на использование этой системы. Для ее распространения отдел «DP Labs» занимается разработкой и поддержкой веб-приложения Esprit Web, расположенного по адресу: <https://ew.dptechnology.com/>. Основной его функционал заключается в обеспечении эффективного взаимодействия заказчиков, дистрибьюторов, пользователей, команд разработчиков и службы поддержки. Также в Esprit Web распространяются CAD файлы, программы с G-кодом, постпроцессоры и прочие необходимые файлы с соблюдением авторских прав и учетом типов лицензий.

Esprit Web предоставляет многофункциональный раздел технической поддержки, который позволяет пользователям получать советы и инструкции по эксплуатации CAD/CAM системы, сообщать об ошибках в системе, а также запрашивать техническую поддержку либо разработку нового функционала.

## **1.2. Описание процесса осуществления технической поддержки**

Осуществление технической поддержки осуществляется на сайте <https://ew.dptechnology.com/>. Действующих лиц, участвующих в процессе технической поддержки можно разделить на три категории:

- Пользователи;
- Сотрудники DP Technology;
- Дистрибьюторы.

Пользователями являются обладатели лицензии на CAD/CAM систему Esprit. Они могут обращаться в службу технической поддержки за помощью.

Сотрудниками в данном контексте считаются специалисты отдела технической поддержки, оказывающие пользователям помощь, а также разработчики, дорабатывающие CAD/CAM систему Esprit.

Дистрибьюторами являются представители DP Technology в разных регионах, которые нацелены на распространение системы Esprit и продажу лицензий. Крупные дистрибьюторы также оказывают техническую поддержку пользователям.

Процесс оказания технической поддержки начинается с создания пользователем запроса на техническую поддержку (далее просто запрос). Создавать запросы можно в разделе технической поддержки Esprit Web либо в настольном клиенте CAD/CAM системы Esprit. При создании нового запроса пользователи заполняют множество полей для детального описания возникшей проблемы (рис. 3).

The screenshot shows the 'New Support Request' page in the Esprit Web system. The page has a dark sidebar on the left with navigation options: 'Search', '+ New Support Request', and 'Search Companies'. The main content area is titled 'New Support Request' and contains several form fields: 'Product' (dropdown menu with 'ESPRIT' selected), 'Request type' (dropdown menu with 'Reported Incident' selected), 'Category' (dropdown menu with 'CAD Geometry' selected), 'ITAR Request' (checkbox checked), 'Priority (Severity / Area)' (dropdown menu with 'Functional' selected), 'Build Found' (dropdown menu with '19.20.201.2400 (ESPRIT 2020 R1)' selected), 'End User Contact' (dropdown menu with 'Logan Wolverine' selected), 'Subject' (text input field with 'Diploma' entered), and 'Description' (rich text editor with a toolbar containing icons for link, bold, italic, underline, list, and other text formatting options).

Рисунок 3 – Страница создания запроса в Esprit Web

Количество доступных для заполнения полей зависит от уровня доступа пользователя. Общедоступными являются следующие поля:

- Название программного продукта;
- Тип запроса;
- Категория запроса;
- Приоритет запроса;
- Версия продукта;
- Тема запроса;
- Описание запроса.

Запрос представляет собой детальное описание проблемы, которая возникла у пользователя. В процессе обработки запроса пользователь и ответственный специалист могут редактировать его поля, а также добавлять комментарии, примечания и прикреплять необходимые файлы в любой момент времени. Таким образом, ответственный инженер может уточнить у пользователя необходимую информацию, а затем прикрепить свой ответ. На

протяжении жизненного цикла запроса на поддержку изменяется его статус. В таблице 1 представлены все возможные статусы с их описанием.

Таблица 1 – Описание статусов запросов

Статус запроса	Описание статуса запроса
Новый	Присваивается автоматически созданным запросам, в случае если в системе не задан ответственный специалист для запросов данного типа. Также статус присваивается обработанным запросам, которые были открыты вновь (если пользователь посчитал что его проблема не была решена). Статус «Новый» означает, что менеджеру необходимо назначить сотрудника для обработки запроса.
Назначен	Присваивается автоматически созданным запросам, в случае если в системе явно задан ответственный специалист для запросов данного типа, либо выставляется автоматически при назначении работника на запрос со статусом «Новый».
Обрабатывается	Статус означает, что специалист начал работать над решением запроса.
Ожидает	Статус означает, что ответственный специалист прикрепил к запросу ответ и ожидает обратной связи от пользователя. Выставляется автоматически при прикреплении ответа.
Разрешен	Статус означает, что предоставленный ответ удовлетворил пользователя. Выставляется пользователем либо специалистом при получении положительной обратной связи.
В разработке	Статус означает, что запрос не может быть решен без редактирования исходного кода CAD/CAM системы Esprit. Ответственный специалист информирует об этом и

	направляет запрос в отдел разработки используя интерфейс Esprit Web.
Закрит	Статус означает что в течении длительного времени не удалось разрешить запрос, либо пользователь не предоставил обратную связь после добавления ответа. Работа над данным запросом прекращена.

Далее рассмотрим более подробно жизненный цикл запросов. По умолчанию созданные запросы на техническую поддержку получают статус "Новый". Для каждого созданного запроса менеджеры назначают ответственного специалиста либо из службы поддержки «DP Technology», либо из службы поддержки крупнейших дистрибьюторов. При этом запрос получает статус "Назначен". Также запрос может быть создан с уже назначенным специалистом и сразу находиться в статусе "Назначен".

Если после назначения специалиста запрос не был успешно разрешен и не имел никаких изменений на странице, то по истечению двух недель запрос отмечается как "неактивный". По истечению следующих двух недель статус запроса автоматически устанавливается как "Закрит" при отсутствии активности специалиста и пользователя.

Статусы "Разрешен", "Закрит" и "В разработке" являются конечными для специалиста технической поддержки и обозначают, что его работа над запросом окончена. Однако, если пользователь считает, что его проблема не была решена, он может вновь открыть обработанный ранее запрос. Конечные статусы изображены зеленым цветом на UML-диаграмме состояний, наглядно описывающей смену статусов запроса (рис. 4).

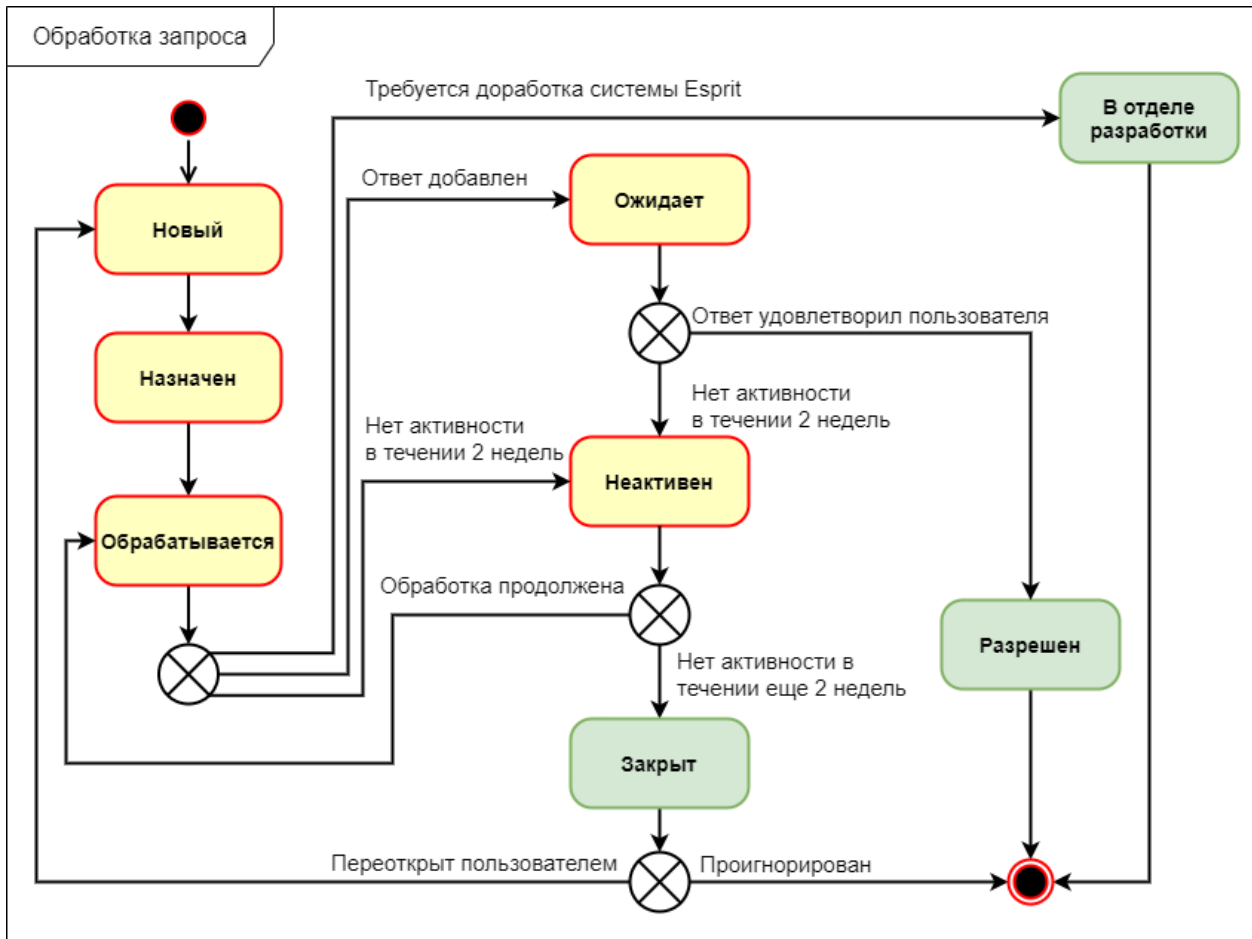


Рисунок 4 – UML-диаграмма состояний запроса

Такие действия как создание запроса, смена ответственного специалиста, добавление ответа, прикрепление файлов, прикрепление комментариев, смена статуса на "Закрыт", "Разрешен", "В разработке", "Неактивен", и "Новый" при открытии запроса вновь сопровождаются отправкой Email-уведомлений для пользователя, специалиста и подписавшихся пользователей. В рамках улучшения процесса оказания технической поддержки была произведена локализация данных рассылок.

Для навигации по запросам в поддержку на сайте был предусмотрен поиск и система фильтров доступная как для пользователей, так и для специалистов службы поддержки, последние в свою очередь обладают более широким набором прав и инструментов (рис. 5).



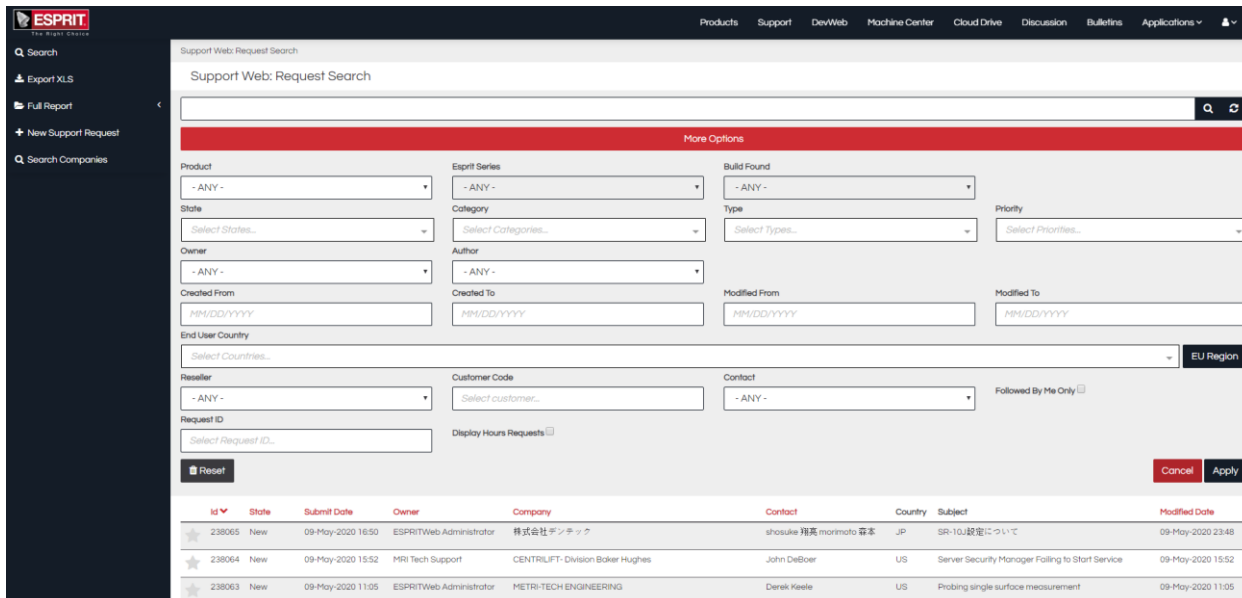


Рисунок 5 – Поиск по запросам на техническую поддержку в Esprit Web

При работе с несколькими запросами специалисты используют панель задач, отображающую все назначенные запросы, сгруппированные по статусу для удобства (рис. 6).

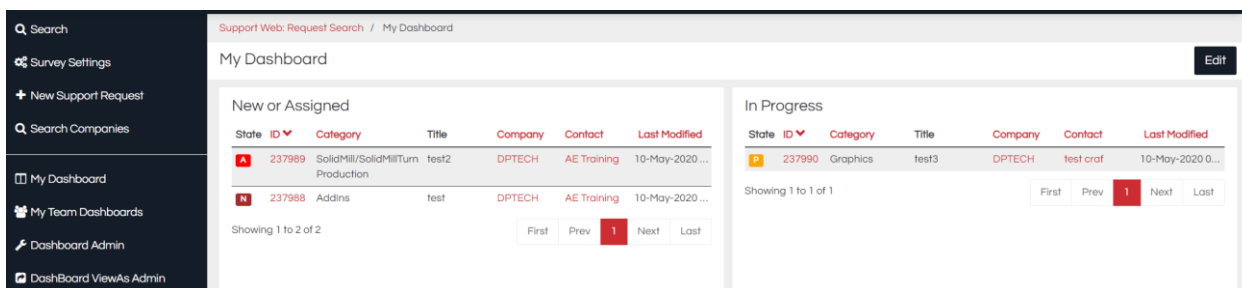


Рисунок 6 – Панель задач специалиста службы технической поддержки в Esprit Web

## 2. Постановка задач

В ходе работы над проектом для управления жизненным циклом задач использовался инструмент YouTrack. Он совмещает в себе онлайн баг-трекер и инструмент для управления задачами, проектами и Agile-процессами. YouTrack поддерживает эффективные поисковые запросы и командный синтаксис, а также позволяет полностью настроить рабочие процессы и атрибуты задач [19].

В системе YouTrack менеджментом было создано несколько задач с описанием бизнес-требований к новому разрабатываемому проекту по отображению статистики работы службы поддержки (рис. 7).

Created by Cedric Cornillet 26 Feb 2019 16:26 Updated by Ivan Cherepanov 29 Jul 2019 20:54 visible to: All Users

★ EWIT-1150 SupportWeb - Reporting

We could have also a column chart for this one with owner of Support Request in X axis and Average Initial Wait Time as Y axis  
By default display only the first 10 possible owners of the list  
In terms of filters, we would need to have the ability to break down or filter data:  
per people owning the Support Request [same list as possible Incidents owner of Incidents for DP, with Departments]  
Multi Select type of filters  
per End User Country  
Multi Select type of filters  
with a EU Preset button like in the SW search  
within a specified Submit Date range (From / To)

SupportWeb: Time to Resolve  
Goal: Measure how quickly an AE closes, resolves, or sends to R&D an incident after it has been created.  
Unit of time: hours  
Should be limited to tickets:  
Owned by a DP Employee  
Get the time between when the Ticket is created and when the status gets changed to Resolved, Closed or in R&D

Same kind of chart than above with similar filters

- For the "SupportWeb Usage" section, I would like to see:  
Version  
Ratio of tickets related to products available in SupportWeb + a break down E20XX and TNG (should not be all in ESPRIT)  
It could be a pie chart like this one:

Version

Version	Percentage
E20XX	65%
ETNG	35%

Project: EspritWeb IT  
Story Points: 0  
EW Category: Feature  
State: In Progress  
Assignee: Aleksey Volyntsev, Ivan Cherepanov  
Priority: Normal  
Platform: MVC  
EW Modules: SupportWeb  
EW Milestone: July-Aug\_2019  
Due Date: No Due Date  
Type: Item  
ProjectID: No ProjectID  
Project Status: No project status

Voters  
Watchers  
Boards

Рисунок 7 – Фрагмент описания поставленных задач в системе YouTrack

При анализе поставленных в системе YouTrack задач, а также в ходе общения с менеджментом, было выявлено несколько групп требований к разрабатываемому проекту которые представлены в таблицах приложения Б.

## 2.1. Распределение задач

Поставленные задачи были декомпозированы и распределены, так как данный объемный проект выполнялся командой из нескольких человек. Совместно с коллегой в рамках данного проекта был реализован весь требуемый функционал. На завершающих стадиях проекта в нем принял участие Quality Assurance инженер для тестирования. В таблице 2 представлены основные назначенные задачи и их исполнители.

Таблица 2 – Поставленные задачи и исполнители

Поставленная задача	Ответственный специалист DP Labs
Формализация требований к приложению.	Черепанов И.С.
Исследование существующих решений, выбор библиотек для визуализации диаграмм.	Черепанов И.С., Волынцев А.И.
Проектирование архитектуры разрабатываемого веб-приложения (Клиентская и серверная часть).	Волынцев А.И., Черепанов И.С.
Доработка Esprit Web под нужды разрабатываемого приложения: <ul style="list-style-type: none"><li>• Сохранение временных показателей для ведения статистики;</li><li>• Получение пользовательских оценок работы службы поддержки.</li></ul>	Черепанов И.С.
Разработка компонентов клиентской части.	Черепанов И.С., Волынцев А.И.
Разработка компонентов серверной части.	Черепанов И.С., Волынцев А.И.
Оптимизация времени получения данных для диаграмм и графиков.	Черепанов И.С.
Покрытие функционала Unit-тестами.	Черепанов И.С.

Ручное и автоматизированное тестирование.	Мурашов П.В.
Проектирование архитектуры сервисов Email-рассылок и внедрение механизма локализации.	Черепанов И.С.

## **2.2. Используемые средства командной разработки**

В рамках данного проекта использовались уже ранее освоенные средства и инструменты, так как в отделе «DP Labs» процесс командной разработки уже был отлажен.

Для совместной работы над проектом использовался Bitbucket - сервис для хостинга проектов и их совместной разработки, основанный на системе контроля версий Mercurial и Git. По назначению и основным предлагаемым функциям он аналогичен GitHub, от которого отличается с одной стороны меньшей пользовательской базой, а с другой, имеет определённые преимущества в плане размещения непубличных проектов [22].

Пользовательский интерфейс веб-сервиса представлен ниже (рис. 8).

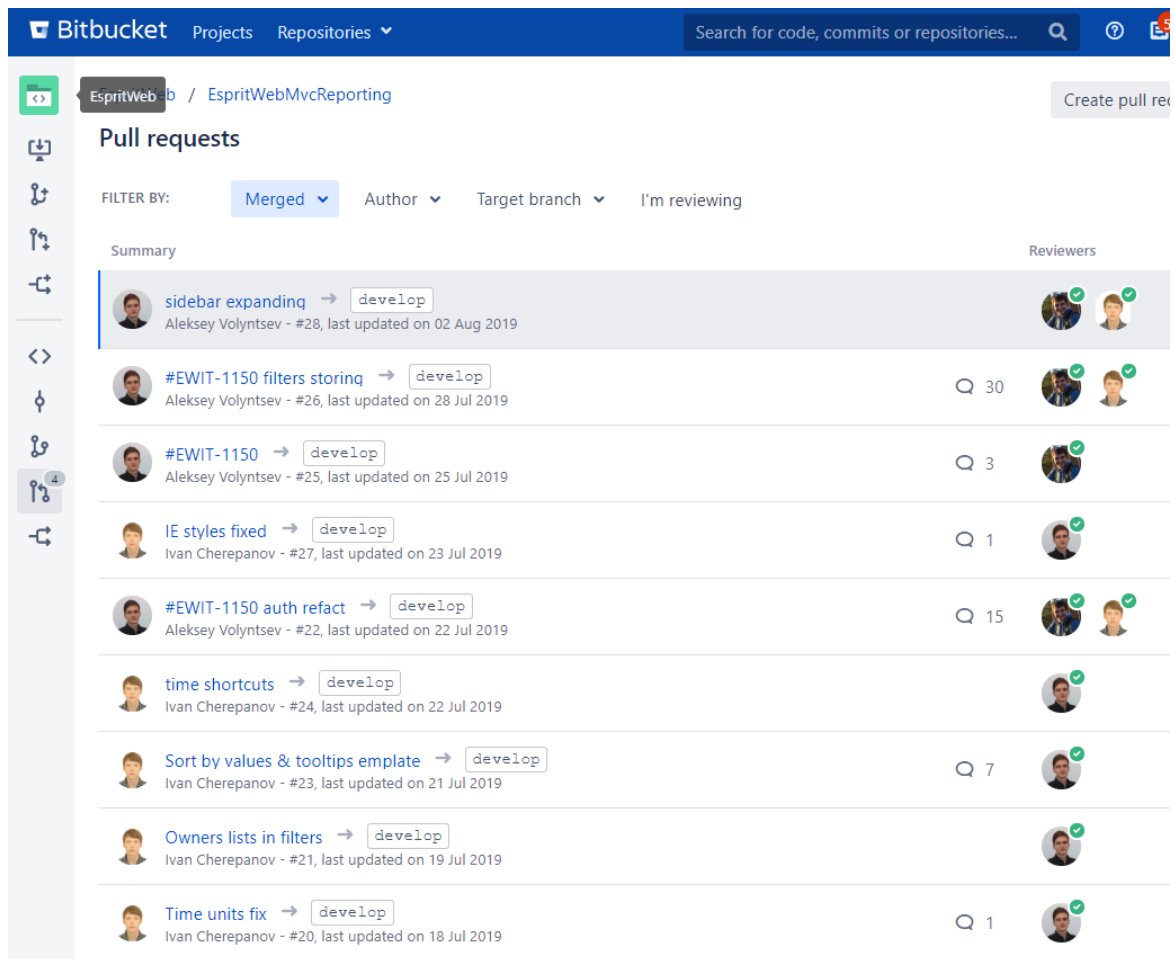


Рисунок 8 – Пользовательский интерфейс сервиса Bitbucket

В ходе разработки изменения в проект вносились лишь после проверки добавляемого кода коллегой. Это было реализовано с помощью использования механизма Pull Requests, при котором для каждой решенной задачи создается запрос на проверку и слияние кода с основной кодовой базой – pull request. Таким образом достигается понимание всех частей проекта всеми участниками разработки и повышается качество кода. При наличии ошибок, несоответствий стандартам оформления кода, использований неоптимальных методов всегда можно уведомить об этом автора кода с помощью комментариев и предложенных исправлений в системе Bitbucket, а созданный запрос на слияние кода отклонить и направить на доработку.

Совместная разработка проводилась с использованием системы контроля версий Git в отдельных ветках кода, ответвленных от основной ветви разработки. Основная ветка всегда содержала рабочую и протестированную версию проекта.

Затем при добавлении нового функционала использовался механизм слияния веток – основная ветвь разработки наполнялась новым кодом. При наличии изменений в одной и той же части кода в обоих объединяемых ветках производится ручное устранение конфликтов в среде Visual Studio 17.

В качестве визуального клиента для работы с системой управления версиями было использовано приложение Source Tree (рис. 9). Оно предоставляет основные функции Git и обладает наглядным графическим интерфейсом. При необходимости всегда можно запустить терминал Git и выполнить команды через консоль.

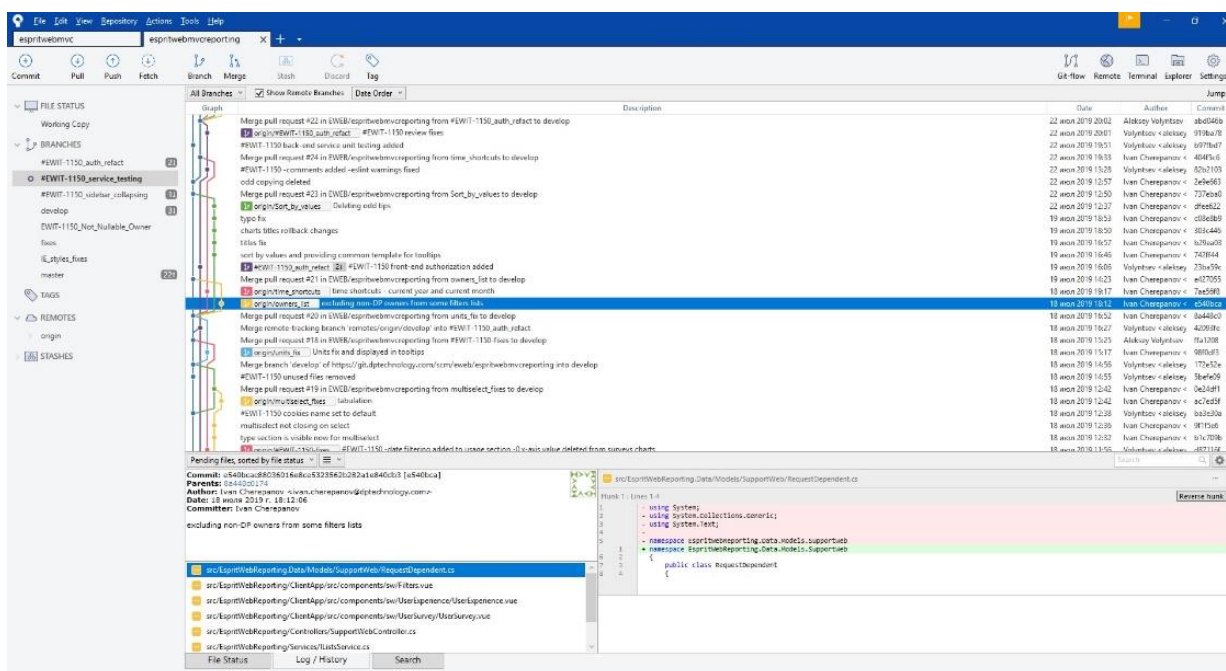


Рисунок 9 – Пользовательский интерфейс приложения Source Tree

Распределение разработчиков по задачам, мониторинг статуса реализуемых задач, переписка и обсуждение методов решения осуществляются в системе YouTrack (рис. 10).

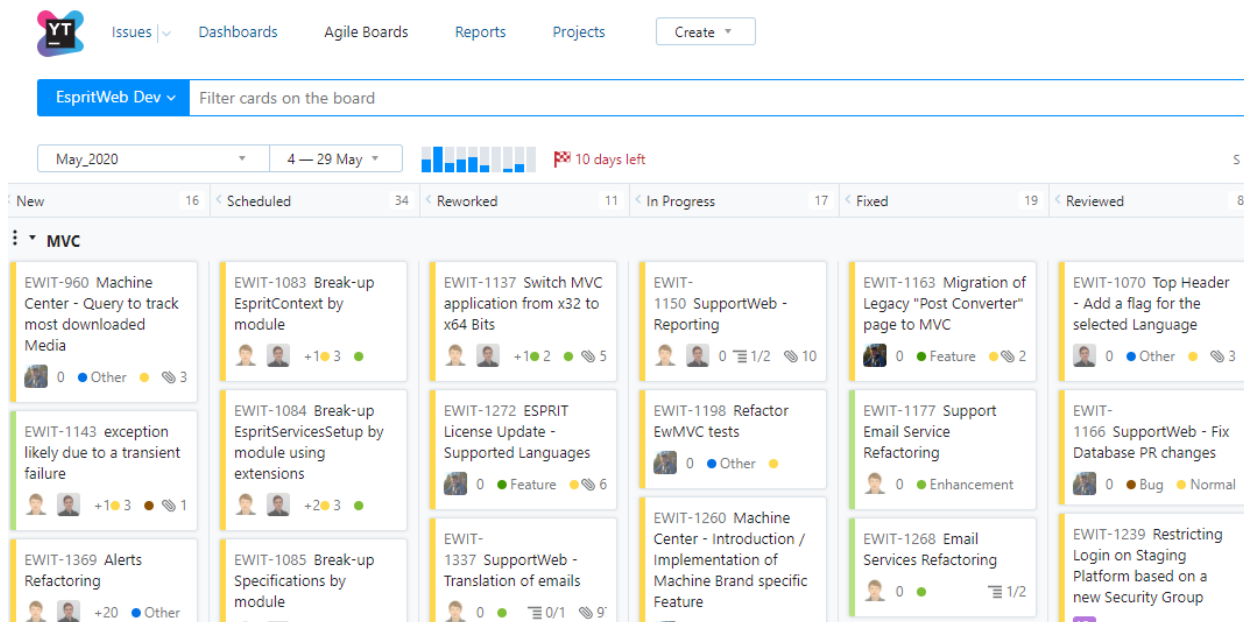


Рисунок 10 – Панель задач в системе YouTrack

Отметки о том, что реализованные задачи по проекту направлены на проверку, либо уже проверены QA-инженером отображаются в YouTrack с помощью атрибутов задач. При найденных ошибках QA-инженер меняет статус задачи на «Reworked», также все найденные ошибки документируются внутри обсуждения задачи всегда с описанным порядком действий для воспроизведения ошибки. После успешного прохождения проверки QA-инженером из томской команды задача также проверяется американскими коллегами из DP Technology. Если ошибок и замечаний не было, то задача является решенной.

Коммуникации с разработчиками, работающими удаленно и с менеджментом DP Technology осуществляются с помощью Skype и корпоративной платформы Microsoft Teams (рис. 11) с помощью рабочих чатов и организации видеоконференций.

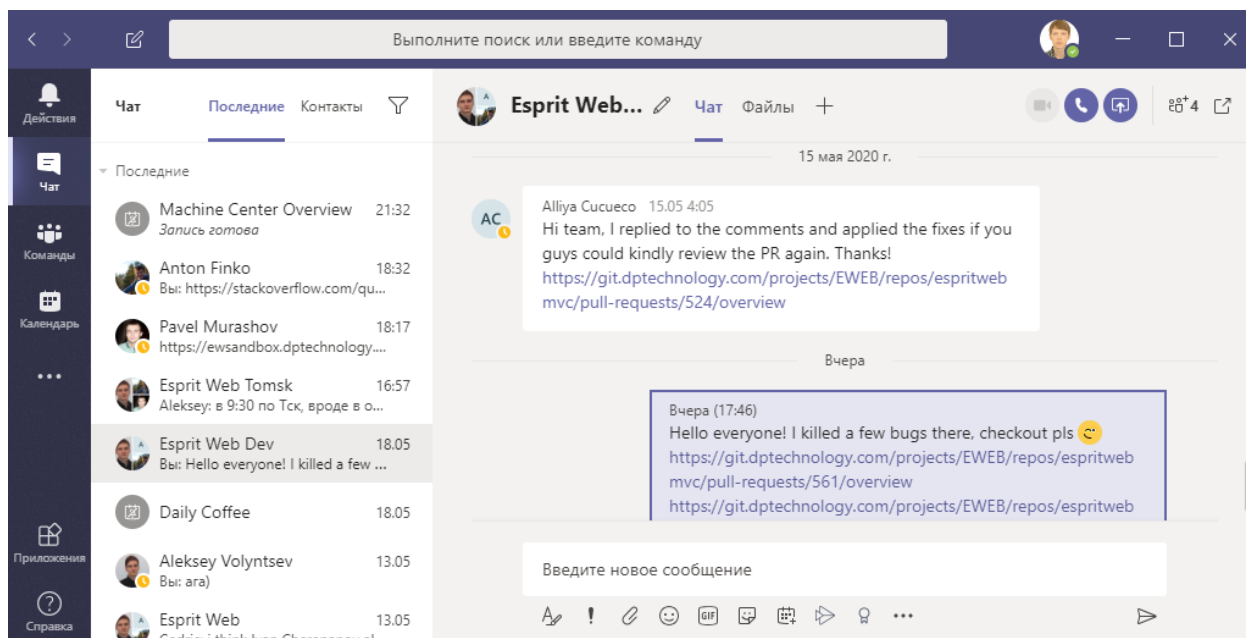


Рисунок 11 – Интерфейс корпоративной платформы Microsoft Teams

### 2.3. Работы по обеспечению требуемых данных для диаграмм

Основным функционалом разрабатываемого приложения является отображение круговых и столбчатых диаграмм. Однако, не все данные для них можно получить, используя основную базу данных приложения Esprit Web. Требуется также определить какие доработки необходимо внести, чтобы получить недостающие данные.

Каждая из диаграмм показывает важные метрики, по которым можно судить о качестве работы технической поддержки DP Technology. Диаграммы отображают обработанные и сгруппированные данные для анализа работы службы технической поддержки. Некоторые из них показывают значения ключевых показателей эффективности (КПИ) в абсолютных значениях, другие показывают КПИ в разрезе времени, третьи предоставляют полезную менеджменту и аналитикам информацию. В таблице 3 представлены все отображаемые диаграммы, описана их смысловая нагрузка, а также акцентировано внимание на том, имеются ли в системе необходимые данные для составления требуемых диаграмм на момент начала работ по проекту.



Таблица 3 – Диаграммы реализуемого проекта

Название диаграммы	Описание диаграммы	Источник данных для диаграммы
Quality Of Technical Response	Диаграмма отражает пользовательские оценки качества обслуживания.	Отсутствует, требуется введение опроса в Esprit Web
Timely Resolution	Диаграмма отражает пользовательские оценки скорости обслуживания.	Отсутствует, требуется введение опроса в Esprit Web
Recommendation Rating	Диаграмма отражает насколько вероятно то, что пользователь рекомендует систему Esprit коллегам или знакомым.	Отсутствует, требуется введение опроса в Esprit Web
Number of Support Requests	Диаграмма отражает количество созданных запросов (сгруппированное по датам создания запросов).	Основная БД проекта Esprit Web
Initial Wait Time	Диаграмма отражает среднее время ожидания ответа от службы технической поддержки. (сгруппированное по датам создания запросов)	Отсутствует, требуется доработка Esprit Web
Time to Resolve	Диаграмма отражает среднее время среднее время обработки запросов службой технической поддержки (сгруппированное по датам создания запросов)	Отсутствует, требуется доработка Esprit Web

Time to Assign	Диаграмма отражает среднее время назначения ответственного работника для обработки запроса (сгруппированное по датам создания запросов)	Отсутствует, требуется доработка Esprit Web
Owner Time to Respond	Диаграмма отражает среднее время затраченное на ответ специалиста после его назначения (сгруппированное по датам создания запросов)	Отсутствует, требуется доработка Esprit Web
Support Request ReOpened	Диаграмма отражает долю вновь открытых запросов относительно всех закрытых запросов.	Основная БД проекта Esprit Web
Support Request ReAssigned	Диаграмма отражает долю переназначенных запросов относительно всех назначенных запросов	Основная БД проекта Esprit Web
Version	Диаграмма отражает количество созданных запросов, сгруппированных в сектора по поддерживаемому программному продукту. (В основном это описанная ранее CAD/CAM система Esprit).	Основная БД проекта Esprit Web
Type	Диаграмма отражает количество созданных запросов, сгруппированных в сектора по типу запроса.	Основная БД проекта Esprit Web

Esprit Category – Numbers	Диаграмма отражает количество запросов сгруппированных по категориям функционала САМ системы.	Основная БД проекта Esprit Web
Esprit Category – Hours	Диаграмма отражает суммарное время обработки запросов сгруппированных по категориям функционала САМ системы.	Основная БД проекта Esprit Web
Support Reseller Numbers by –	Диаграмма отражает суммарное количество запросов обработанное дистрибьюторами, сгруппированное по компаниям дистрибьюторов.	Основная БД проекта Esprit Web
Support Reseller – Hours by	Диаграмма отражает суммарное время обработки запросов дистрибьюторами, сгруппированное по компаниям дистрибьюторов.	Основная БД проекта Esprit Web
Support Region Numbers by –	Диаграмма отражает суммарное количество запросов, сгруппированных в сектора по странам обратившихся пользователей.	Основная БД проекта Esprit Web
Support Region – Hours by	Диаграмма отражает суммарное время обработки запросов, сгруппированных в сектора по странам обратившихся пользователей.	Основная БД проекта Esprit Web

Для построения диаграмм «Quality Of Technical Response», «Timely Resolution», «Recommendation Rating» потребовалось получить пользовательские оценки. В связи с этим был спроектирован, утвержден и реализован опрос пользователей. Для сохранения предоставленных оценок были созданы таблицы в БД. Оценки вносятся при отправке пользователями форм со страницы опроса на сайте Esprit Web. Для оценивания доступны лишь обработанные запросы, которые были созданы после разработки этого механизма.

Страница опроса доступна пользователям по ссылке, внедренной в Email-уведомления. Для более широкого охвата пользователей было решено провести локализацию рассылок центра поддержки Esprit Web – эта задача является объемной и будет более подробно описана далее. Описанные работы по получению пользовательских оценок качества работы представлены в таблице 4 с учетом исполнителей.

Таблица 4 – Работы по получению пользовательских оценок

Задача	Ответственный специалист DP Labs
Доработка БД	Финько А.И.
Разработка клиентской части страницы с формой для ввода оценок	Финько А.И.
Разработка серверной части для валидации и сохранения оценок	Волынцев А.И., Финько А.И.
Внедрение ссылки на страницу оценки в Email-уведомлениях	Черепанов И.С.
Проектирование механизма локализации Email-уведомлений	Черепанов И.С.
Разработка механизма локализации Email-уведомлений	Черепанов И.С.
Тестирование	Мурашов П.В.

Для построения диаграмм «Initial Wait Time», «Time to Resolve», «Time to Assign» и «Owner Time to Respond» потребовалось получить данные о времени наступления таких событий, как:

- Создание запроса;
- Назначение ответственного специалиста;
- Прикрепление работником ответа (комментария либо файла);
- Закрытие запроса.

Данные значения можно получать из разнообразных таблиц-справочников, а затем группировать и фильтровать для получения временных метрик, но без введения индексации по множеству полей этот процесс занимает до нескольких минут для сложных запросов. Введение множества индексов чрезмерно расширяет занимаемый БД объем памяти. Потребовалось обеспечить способ быстрого получения данных для отображения диаграмм. Эти работы представлены в таблице 5 и более подробно описаны в разделе 3.2.

Таблица 5 – Работы по получению временных показателей

Поставленная задача	Ответственный специалист DP Labs
Доработка БД	Финько А.И, Черепанов И.С.
Проектирование изменений и доработка серверной части Esprit Web	Черепанов И.С.
Тестирование	Мурашов П.В.

### **3. Доработка Esprit Web для получения необходимых данных**

#### **3.1. Получение пользовательских оценок работы службы поддержки**

Выделенные в таблице 4 задачи были успешно реализованы. Для предоставления пользователям возможности участвовать в опросах была создана

специальная страница с формой для отправки данных на сервер с последующим сохранением в БД (рис. 12).

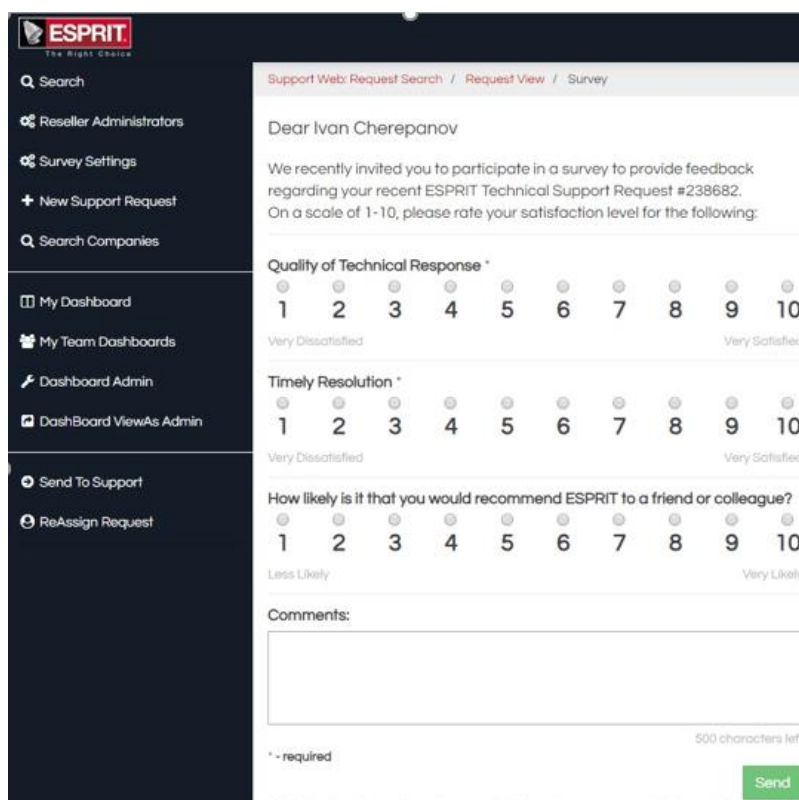


Рисунок 12 – Веб-страница с опросом в Esprit Web

Для предоставления пользователям доступа к опросам в некоторые шаблоны Email-уведомлений были встроены специальные ссылки, ведущие на страницу опроса (рис. 13-14).

```
<!-- Survey link -->
@if (Model.RequestDetails.IsSurveyAvailable)
{
    var link = $"{Model.PublicWebhost}/Support/RequestSurvey/{Model.RequestDetails.RequestId}";
    <tr>
        <td>
            @Raw(Model.Localize(nameof(SwRequestUpdatedConstants.AdviceToTakeSurvey), link))
        </td>
    </tr>
}
}
```

Рисунок 13 – Код с встроенной ссылкой на опрос в разметке шаблона письма



The only CAM system you'll ever need

ner | Performance | Value

Если мы помогли вам по данному запросу, мы хотели бы получить от вас обратную связь в виде краткого опроса из 3 пунктов, пожалуйста нажмите сюда.

### Краткая сводка

ID Запроса 238682  
Дата создания 19-май-2020 18:47  
Дата изменения 26-май-2020 16:06

[Подробности запроса](#)

## Рисунок 14 – Фрагмент локализованного письма с ссылкой

Затем для получения оценок от иностранных пользователей и улучшения качества их взаимодействия с системой Esprit Web была выполнена объемная задача по локализации модуля технической поддержки и Email-рассылок. Задача по локализации Email-рассылок была разбита на следующие подзадачи:

1. Выбор технологий локализации Email-рассылок.
2. Переработка модулей веб-сайта с учетом функционала локализации.
3. Определение списка поддерживаемых языков.
4. Выявление локализуемых строк.
5. Предоставление локализуемых строк для перевода с контекстом и пояснениями.
6. Определение бизнес-логики выбора языка для различных видов сообщений.
7. Реализация функционала по локализации Email-рассылок.
8. Правки существующих тестов.
9. Проведение тестирования производительности.
10. Проведение Code Review с доработками по необходимости.
11. Проведение ручного тестирования.
12. Генерация локализованных писем всех видов для вычитки.
13. Внесение правок в переводы.
14. Покрытие тестами.

Рассмотрим более детально решение вышеперечисленных подзадач. Для локализации .NET приложений предусмотрена стандартная методика с выделением локализуемых строк в resx-файлы. Из данных файлов с помощью интерфейса IStringLocalizer извлекаются требуемые строки на основании объекта CultureInfo, который задает язык. Эта методика описана в официальной документации [23].

При решении задач локализации было решено отказаться от применения машинного перевода в связи с его недостаточным качеством и наличием множества строк, значение которых зависит от контекста, например – «Owner» в значении «ответственный», «Note» в значении «Комментарий, примечание».

Выбранные для локализации уникальные строки были представлены в таблице и переданы в местные отделения «DP Technology» для осуществления ручного перевода. Для обеспечения качественного перевода был подготовлен ряд материалов – комментарии с пояснением контекста, а также изображения подготовленных писем без использования личных данных. (рис. 15)

File	Vartname	Comment	English	Chi	Fre	S	Ital	Ge	Ruz	Japanese
Resources\Email\HeaderDetailsBlocked			Issue							
Resources\Email\Summary\itar\DetailsBlock			This Request has been updated, but your account is							
Resources\Email\AdviceToTakeSurvey		Please be careful with html tags. <a href="{0}"><a/> untouched. "here" need to be translated	If we have successfully solved your request, we'd love to get your feedback with a short 3 - question customer survey, please click <a href="{0}">here</a>.							
Resources\Email\Header\FileDate			Date							
Resources\Email\Header\FileDownloadLink			#Download link							
Resources\Email\Header\FileName			Filename							
Resources\Email\Header\FileSize			File Size							
Resources\Email\Inactive\AutoCloseInfo			Should we not hear back from you, this support request will be closed in two weeks. If not, please respond back to this message or click on the Support Request link above to update your request.							
Resources\Email\Inactive\IfRequestNotSolved			Have we successfully solved your support request?							
Resources\Email\Inactive\IfRequestSolved			If yes, please consider closing this Support Request.							
Resources\Email\Inactive\Solved\CloseAction			If yes, no action is required.							
Resources\Email\Label\Author			Author							
Resources\Email\Label\Category			Category							
Resources\Email\Label\Company			Company							
Resources\Email\Label>Contact			Contact							
Resources\Email\Label>ContactMotherid			Module / Dongle							
Resources\Email\Label\Email			Email Address							
Resources\Email\Label\FileDownloadLink			Download							
Resources\Email\Label\Location			Located in							
Resources\Email\Label\ModifyDate			Modify Date							
Resources\Email\Label\Owner			Owner							

Modify Date: 27.11.2019 15:40:22

This Request has been updated, but your account is not authorized to view ITAR request. Please contact your Reseller or ESPRIT Support at support@dpotechnology.com

---

**Request to take a survey**

We have closed your support request. If we have successfully solved your request, we'd love to get your feedback with a short 3 - question customer survey, please click [here](#). If you still need help, please respond back to this message or click on the Support Request link below to re-open your request.

**Files table**

FileName	File Size	Date	#Download link
1)SurveyCompleted.xlsx	3 KB	09-Dec-19	Download
2)Shared.xlsx	2.8 KB	09-Dec-19	Download
3)CreatedUpdated.xlsx	6 KB	09-Dec-19	Download
4)AssignedReminder.xlsx	3.3 KB	09-Dec-19	Download
5)Layout.xlsx	3.1 KB	09-Dec-19	Download

**Inactivity message**

Support Request ID 224108  
Subject Localization  
Have we successfully solved your support request?

Рисунок 15 – Составленная таблица строк для локализации с дополнительными материалами

Затем из переведенных строк были сформированы resx-файлы для каждого языка. Это было произведено с использованием расширения для Visual Studio – ResXManager (рис. 16).



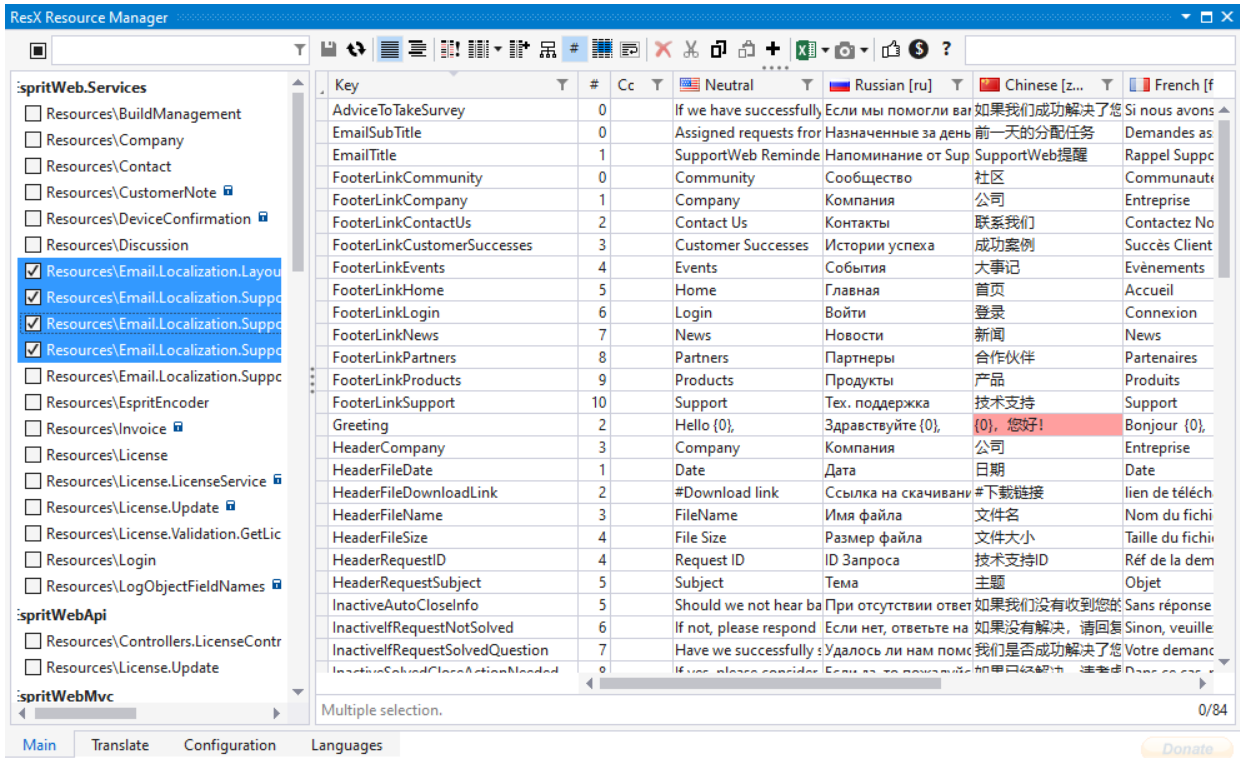


Рисунок 16 – Сформированные resx-файлы в ResXManager

Основные изменения были внесены в проект EspritWeb.Services, так как он отвечает за бизнес-логику приложения. Рассмотрим архитектуру сервисов по отправке сообщений внутри проекта до начала внедрения механизма локализации в виде UML-диаграммы пакетов (рис. 17).

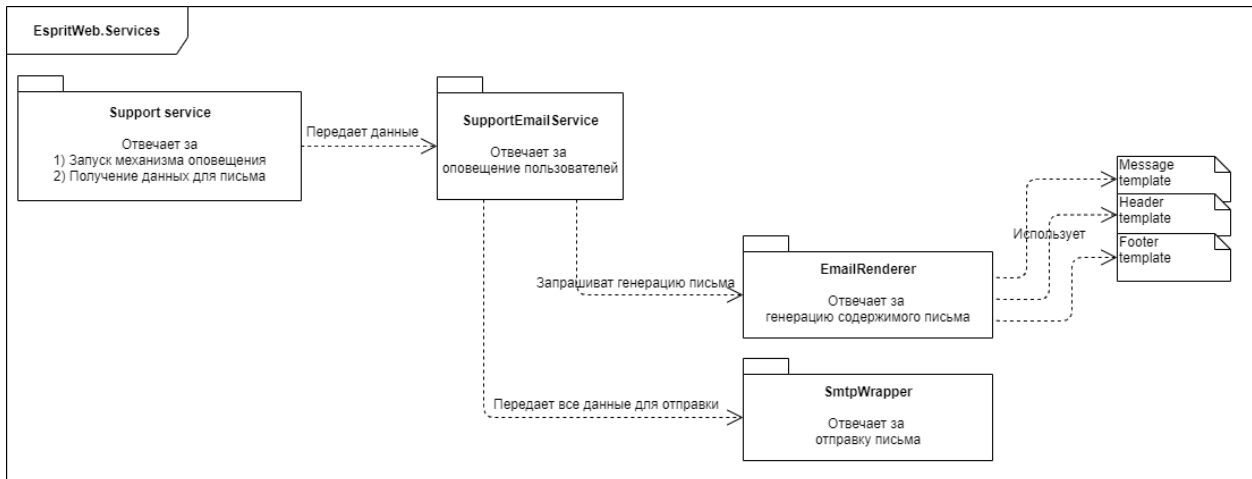


Рисунок 17 – UML-диаграмма архитектуры сервисов до локализации

На рисунке выше представлено 4 модуля:

- SupportService обрабатывает запросы на техническую поддержку и, если необходимо, запускает механизм Email-уведомлений. При этом он собирает все требуемые данные для каждого вида уведомлений. Они передаются в SupportEmailService;
- SupportEmailService на основании полученных данных запрашивает у сервиса EmailRenderer генерацию письма конкретного типа по его html шаблону, формирует список пользователей для уведомления согласно бизнес-логике (заполняет поля Cc, To, Bcc, Reply To) и передает сформированные данные сервису SmtпWrapper для отправки готового письма;
- EmailRenderer производит генерацию письма подставляя текстовые и графические данные в нужный файл HTML-шаблона письма;
- SmtпWrapper производит непосредственную отправку сформированного письма по протоколу SMTP.

Так как модуль EmailRenderer формирует тело письма – именно в нем нужно получать локализованные строки. Было решено, что этот модуль будет при необходимости запрашивать локализованные строки у нового сервиса, который будет отвечать только за локализацию (для обеспечения модульности).

При анализе данной диаграммы и сервиса SupportService (в котором более 3000 строк кода) становится очевидно, что он берет на себя лишнюю ответственность и имеет чрезмерно большой функционал – это является антипаттерном ООП, который называется Monster Object.

Для решения данной проблемы было решено вынести функционал по получению всех необходимых данных для почтовых рассылок в отдельный сервис. Далее архитектура была дополнена сервисом локализации, который будет предоставлять переведенные на нужный язык строки. Переработанная архитектура сервисов представлена в виде UML-диаграммы пакетов (рис. 18).

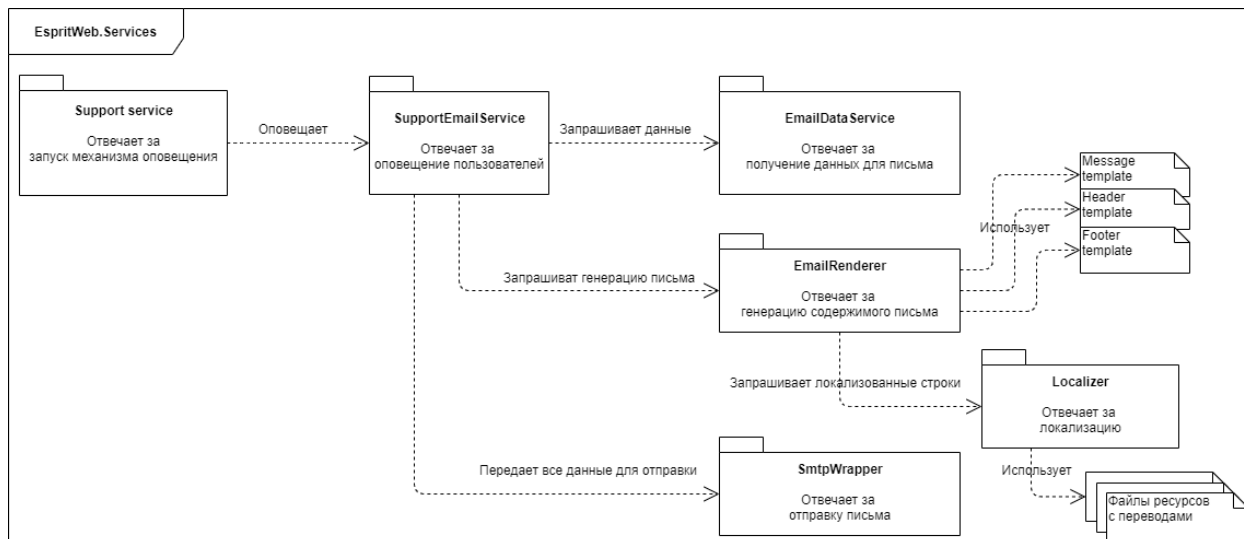


Рисунок 18 – UML-диаграмма переработанной архитектуры сервисов

В переработанной архитектуре сервисов приложения Esprit Web можно выделить следующие изменения:

- SupportService теперь не собирает данные из базы, необходимые для генерации письма. Если требуется отправка уведомлений, сервис передает тип произошедшего события и идентификатор запроса в техподдержку;
- SupportEmailService на основании полученных аргументов запрашивает полные данные у EmailDataService, а не у SupportService;
- EmailDataService выполняет запросы к БД и получает все необходимые для письма данные;
- EmailRenderer обращается к сервису Localizer для вставки переведенных строк в шаблон письма;
- Localizer по запросу возвращает локализованные строки, которые извлекает из созданных ранее resx-файлов с переводами.

При решении задачи по локализации была произведена переработка структуры сервисов веб-приложения, увеличена связность приложения и уменьшено зацепление модулей. Был разработан компонент предоставляющий локализованные строки согласно созданным resx-файлам с переводами строк. Шаблоны Email-сообщений адаптированы для применения локализации. В

результате работы получен масштабируемый механизм по локализации почтовых рассылок.

Язык Email-уведомлений задается исходя из заданного предпочитаемого языка пользователя, обратившегося в техническую поддержку и создавшего запрос. Для большей гибкости также реализована возможность выбирать язык Email-уведомлений из списка в свойствах запроса (рис. 19).

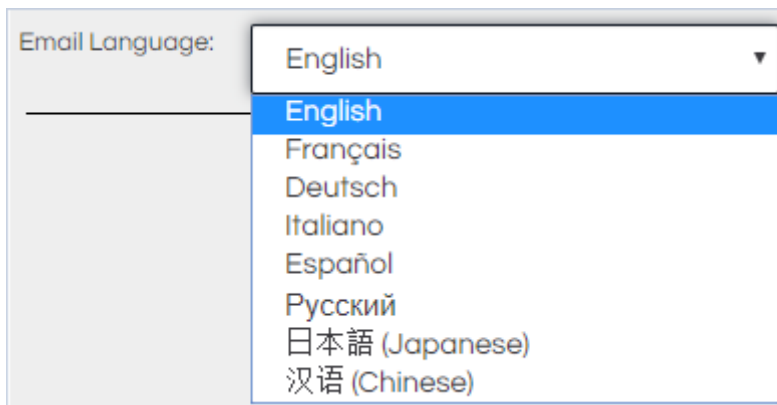


Рисунок 19 – Выпадающий список выбора языка в панели деталей запроса

Реализованный функционал прошел тестирование производительности и Code Review, затем финальные версии переведенных сообщений были вычитаны. После нескольких поправок проект по локализации был внедрен. Модуль генерации и отправки Email-сообщений покрыт более 200 Unit-тестами, которые проверяют содержимое писем, локализацию, корректность списка уведомляемых пользователей.

В результате были получены необходимые данные для отображения диаграмм с пользовательскими оценками (рис. 20).

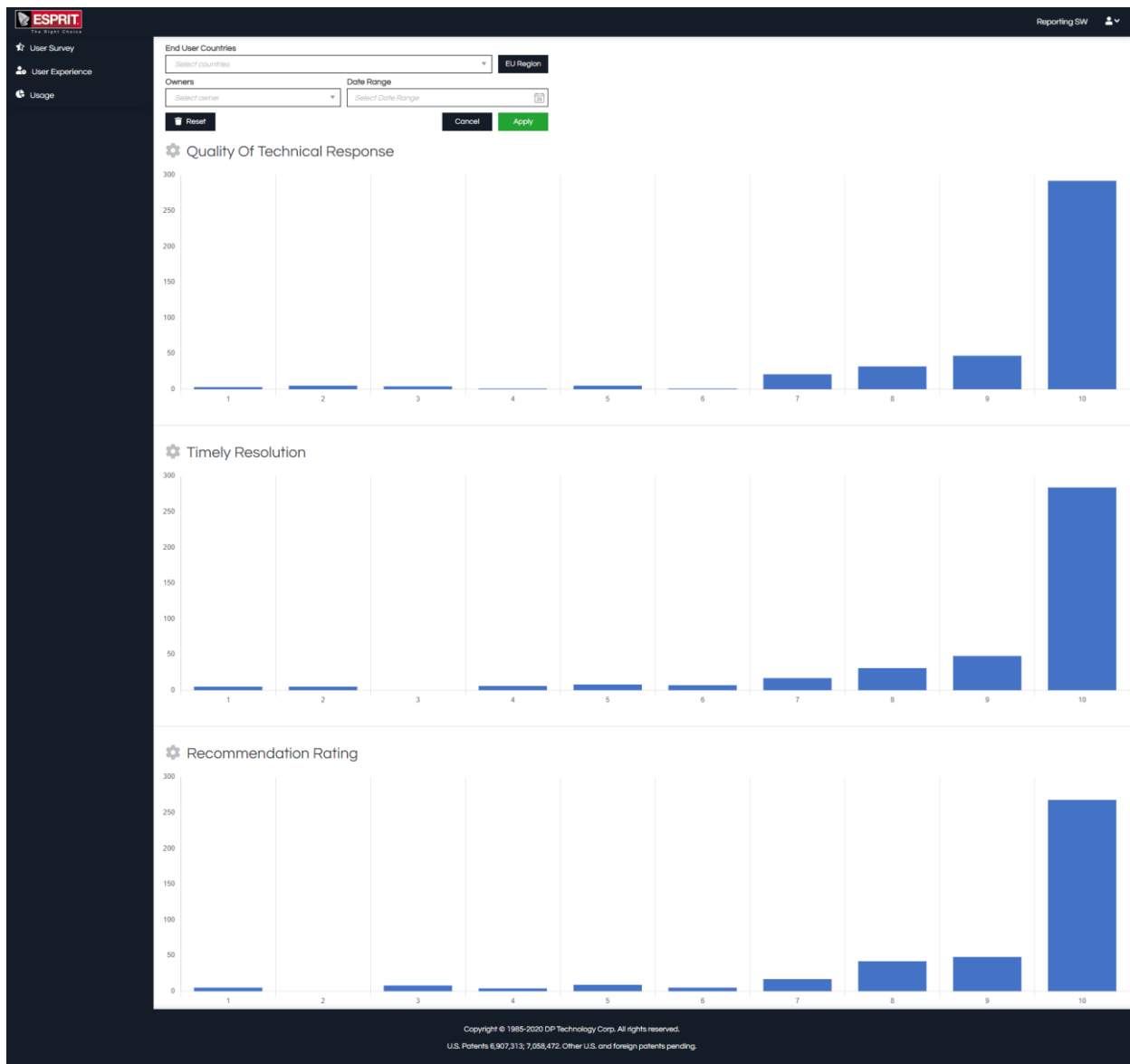


Рисунок 20 – Диаграммы с пользовательскими оценками качества работы службы технической поддержки

### 3.2. Сохранение временных показателей для ведения статистики

Для подсчета временных показателей была введена следующая таблица – SupportWebReportingEvents (рис. 21).

Column Name	Data Type	Allow Nulls
ID	int	<input type="checkbox"/>
IncidentID	int	<input type="checkbox"/>
EventType	int	<input type="checkbox"/>
EventDate	datetime	<input type="checkbox"/>
EventUserID	int	<input type="checkbox"/>

Рисунок 21 – Структура таблицы SupportWebReportingEvents

Запись с типом действия и временем наступления действия сохраняется в БД при создании запроса, назначение ответственного специалиста, прикрепление ответа, закрытие запроса. Таблица состоит из небольшого количества полей, поэтому не занимает много памяти для хранения, а получение данных из нее происходит мгновенно. В системе сохраняются данные о событиях, представленных в таблице 6.

Таблица 6 – Описание сохраняемых событий

Событие	Описание
Запрос создан	Сохраняется при создании запроса
Специалист назначен	Сохраняется при назначении ответственного работника, либо при создании запроса с автоматически назначенным работником.
Сообщение от специалиста добавлено	Сохраняется при прикреплении первого ответа от работника DP Technology (в виде файла, либо комментария).
Обработка завершена	Сохраняется при смене статуса запроса на "Разрешен", "Закрыт" и "В разработке". Это означает что запрос был обработан.
Запрос переоткрыт	Сохраняется при открытии запроса вновь.

Специалист переназначен	Сохраняется при назначении нового ответственного работника на существующий запрос.
-------------------------	--

Разница во времени возникновения этих событий и количество зафиксированных событий являются хорошими метриками для анализа бизнес-процессов. Данные метрики отображаются на диаграммах, что отражено в таблице 7.

Таблица 7 – Взаимосвязь диаграмм и регистрируемых событий

Диаграмма	Связь с сохраненными событиями
Initial Wait Time	Отображает среднее время между событиями «Запрос создан» и «Сообщение от специалиста добавлено»
Time to Resolve	Отображает среднее время между событиями «Запрос создан» и «Обработка завершена»
Time to Assign	Отображает среднее время между событиями «Создан» и «Специалист назначен»
Owner Time to Respond	Отображает среднее время между событиями «Специалист назначен» и «Сообщение от специалиста добавлено»
Support Request ReOpened	Отображает количество запросов, для которых зафиксировано событие «Запрос переоткрыт»
Support Request ReAssigned	Отображает количество запросов, для которых зафиксировано событие «Специалист переназначен»

Для сохранения этих событий был доработан сервис SupportService, более подробно рассмотренный в разделе 3.1. В функциях, ответственных за сохранение изменений в запросе, сохранение файлов и комментариев к запросу, был добавлен функционал по сохранению событий.

В результате были получены необходимые данные для отображения столбчатых диаграмм времени назначения, ответа и обработки запроса, а также круговых диаграмм, отображающих доли переназначенных и переоткрытых запросов (рис. 22).





Рисунок 22 – Построенные по сохраненным событиям диаграммы

## **4. Разработка веб–приложения**

### **4.1. Анализ модели данных**

Согласно требованиям к интеграции, реализуемое приложение должно работать с БД веб-приложения Esprit Web. Для доступа к БД ранее был реализован проект EspritWeb.Data, представляющий собой библиотеку классов с использованием EF Core и подхода Database First.

EspritWeb.Data содержит в себе все классы модели данных и вспомогательные классы для работы с помощью ORM Entity Framework Core – классы контекста данных, наборов данных, миграций БД и т.д. Проект EspritWeb.Data был подключен к созданному приложению в качестве слоя доступа к БД.

Рабочая БД содержит около двух сотен таблиц, в данной работе были задействованы более 20 из них. Чтобы не вдаваться в детали реализации БД и бизнес-логики рассмотрим упрощенную схему модели данных (рис. 23).

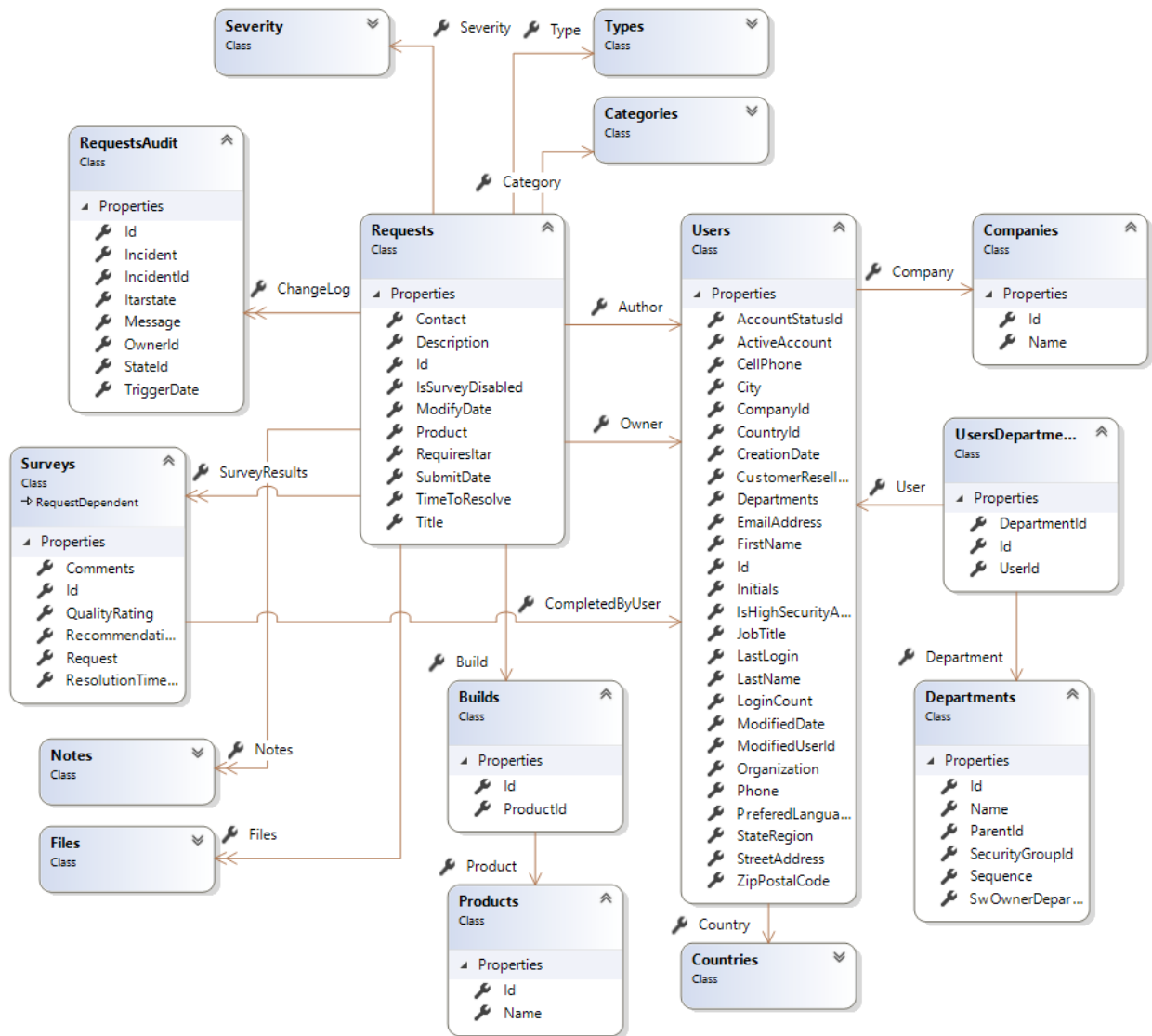


Рисунок 23 – Диаграмма классов модели данных

Стоит уделить особое внимание таблице «Requests», которая хранит данные о запросах. Именно к этой таблице направлено большинство запросов на получение информации в рамках реализуемого проекта.

В БД сохраняются созданные запросы, изменения их статусов, а также прикрепленные файлы и комментарии. Дополнительно ведется логирование изменений значимых полей запроса.

Таблица запросов «Requests» имеет связь со следующими таблицами:

- Справочник типов ошибок;
- Справочник категорий ошибок;
- Справочник приоритета/серьезности ошибок;

- Справочник поддерживаемых продуктов;
- Справочник версий системы;
- Справочник релизов системы;
- Таблица пользователей;
- Таблицей результатов опросов по обработанным запросам и т.д.

#### **4.2. Выбор стека используемых технологий**

С целью обеспечения возможности повторного использования кода из основного приложения Esprit Web и простоты интеграции с ним, был выбран стек технологий, практически не отличающийся от стека основного проекта Esprit Web, а именно:

- ASP.NET Core 2.2;
- MVC;
- Entity Framework 7;
- C# 7.

По той же причине был выбран схожий стек технологий клиентской части:

- Vue.js;
- HTML 5;
- CSS 3;
- Bootstrap.

В проекте используется стандартная реляционная БД, созданная в СУБД MS SQL Server. Основной средой разработки приложения является Microsoft Visual Studio 17.

Средства для обеспечения командной разработки описаны в разделе 2.2.

### 4.3. Анализ и выбор компонентов визуализации диаграмм и фильтров

Анализ и выбор технологий, используемых в проектируемом веб-приложении, заключался в поиске, изучении и выборе наиболее подходящих библиотек и компонентов, ответственных за отображение диаграмм и прочих визуальных элементов. Это связано с тем, что набор технологий не должен иметь много различий со стекком приложения Esprit Web.

Ниже представленные выбранные библиотеки для отображения UI:

- Отображение диаграмм: vue-chart.js;
- Компонент выбора временного диапазона: vue2-datepicker;
- Компонент множественного выбора: vue-multiselect.

Было рассмотрено несколько javascript-библиотек для работы с диаграммами, результат их сравнения представлен в таблице 8.

Таблица 8 – Сравнение JS-библиотек визуализации диаграмм

Название библиотеки	Отзывчивость к изменению размера окна	Поддержка столбчатых и круговых диаграмм	Интеграция с Vue	Поддержка IE	Бесплатный продукт
Chart.js	Да	Да	Да	Да	Да
Google Charts	Частично	Да	Нет	Частично	Да
Chartlist	Да	Да	Нет	Нет	Да
High Charts	Да	Да	Да	Да	Нет

Библиотека Chart.js является наиболее подходящей для реализации данного проекта, так как подходит по всем выделенным критериям. Она снабжена официальной документацией хорошего качества, а также библиотека является достаточно функциональной для реализации нетривиальных задач. В ней используется HTML5 Canvas для отображения диаграмм. Поддержка для устаревших браузеров (таких как IE7, IE78) осуществляется с помощью

типичного полифила, состоящего из двух частей: проверки есть ли встроенная возможность отображения Canvas и затем эмуляция возможности, если её нет [23].

Графики в библиотеке Chart.js отзывчивы по умолчанию. Это означает, что их размер подстраивается под отведенное пространство. Таким образом можно утверждать, что библиотека позволяет корректно отображать графики на мобильных устройствах и планшетах. В завершении стоит отметить, что библиотека имеет предустановленные типы как столбчатых, так и круговых диаграмм.

Далее в качестве компонента для задания диапазона дат был использован vue2-datepicker, так как он, в отличие от конкурентов (vuejs-datepicker, vuematerial-datepicker), поддерживает выбор диапазона дат в едином поле ввода. При настройке других Vue компонентов (vuematerial-datepicker, vuejs-datepicker) пришлось бы использовать 2 поля ввода и вручную задавать связь между ними. Также vue2-datepicker имеет корректно реализованные ограничения ввода при выборе верхней и нижней границ диапазона и настраиваемые предустановленные диапазоны дат.

Затем был выбран компонент выпадающего списка с множественным выбором для реализации фильтра запросов по странам обратившихся пользователей и по ответственным работникам (рис. 24).



Рисунок 24 – Интерфейс компонента vue multiselect

Компонент списка vue-multiselect является лидером по количеству скачиваний, заявленному показателю тестового покрытия, и активности

сообщества в GitHub. Результатов этих показателей у конкурентов оказалось недостаточно для их серьезного рассмотрения.

Описанные выше библиотеки `chart.js`, `vue-multiselect`, `vue2-datepicker`, были изучены, сконфигурированы и применены при реализации веб-приложения. После настройки и доработки `chart.js` были построены круговые и столбчатые диаграммы в соответствии с требованиями к интерфейсу. Библиотеки `vue-multiselect`, `vue2-datepicker` успешно интегрированы и настроены для обеспечения ввода стран и работников в компоненте фильтров.

#### **4.4. Проектирование структуры приложения**

Требования к интеграции накладывают ограничения, не позволяющие реализовать данный проект как внутренний модуль приложения `Esprit Web`. В связи с этим требуется продумать и качественно спроектировать архитектуру приложения. По замыслу менеджмента продукт будет расширяться и впоследствии включать в себя статистику как по работе службы поддержки, так и по другим направлениям деятельности. В связи с этим требуется разработать сопровождаемое и расширяемое приложение.

##### **4.4.1. Проектирование архитектуры клиентской части**

Разработанное приложение является вспомогательным средством аналитики для основного проекта `Esprit Web` и его основной целью является отображение диаграмм. Для того чтобы не расширять стек технологий и не усложнять сопровождаемость проекта, был выбран схожий стек технологий. Основным фреймворком для отображения UI в разработанном приложении является `Vue.js` – JavaScript-фреймворк с открытым исходным кодом для создания пользовательских интерфейсов.

В ходе разработки поменялся подход к отображению клиентской части, и основная страница приложения теперь является одностраничным приложением – `Single Page Application (SPA)`. SPA приложения позволяют имитировать работу десктопных (настольных) приложений. Архитектура устроена так, что при

«переходе» на новую страницу, обновляется только часть содержимого отображаемой страницы. Таким образом избегается повторная загрузка одинаковых элементов, которая происходит при использовании стандартного MVC подхода, что наглядно продемонстрировано на рисунках 25-26.

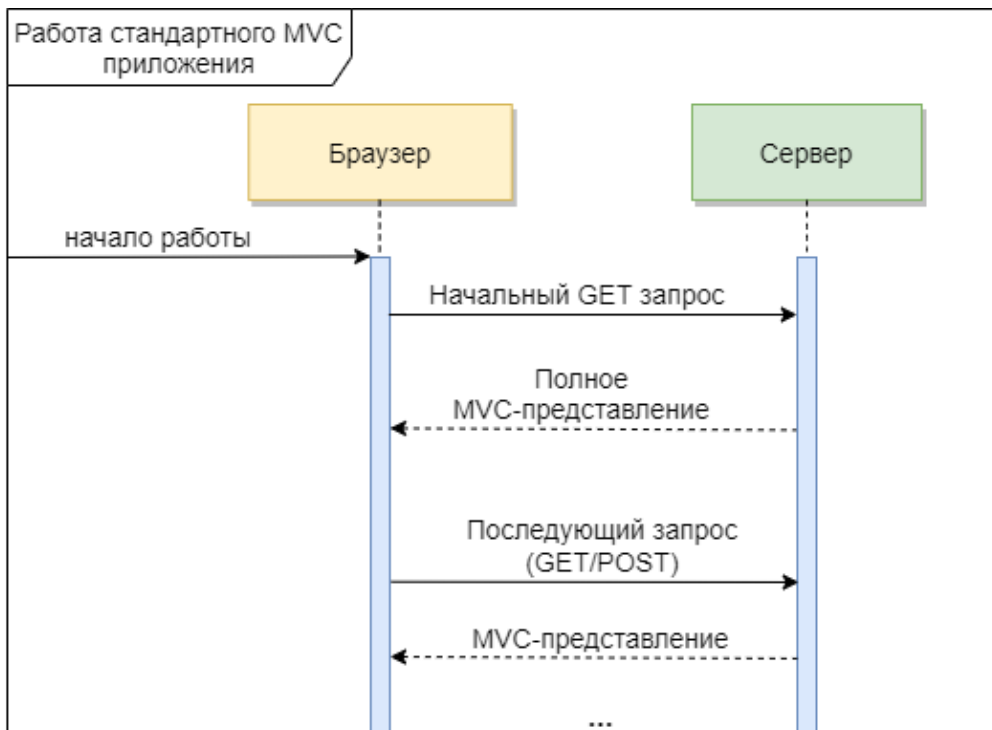


Рисунок 25 – MVC приложение возвращает полное представление



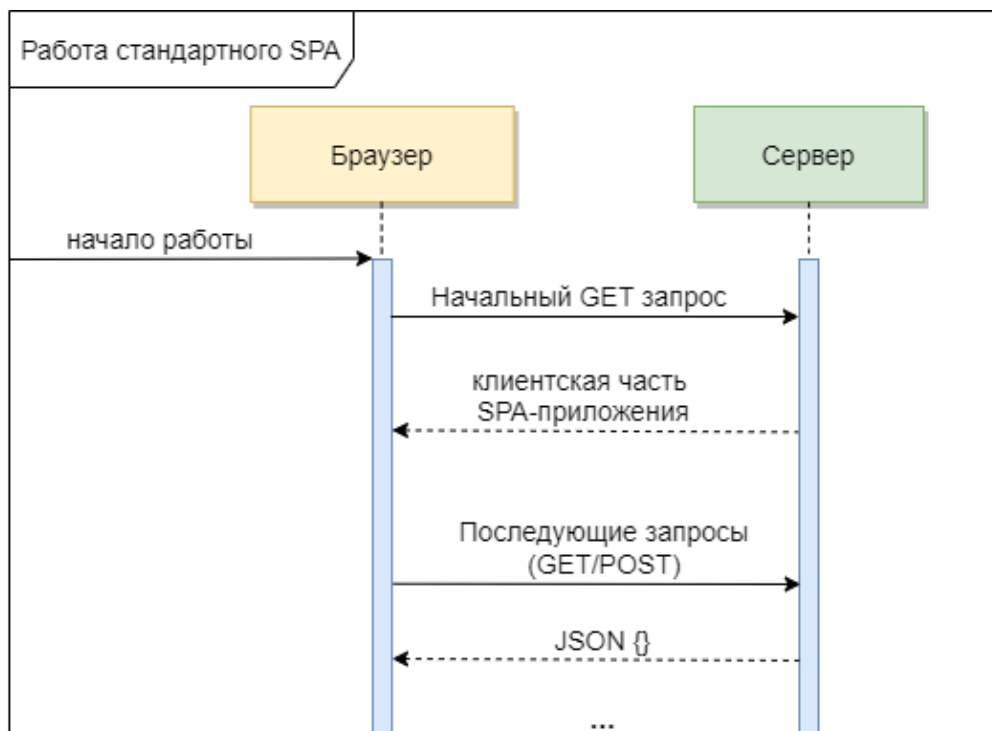


Рисунок 26 – Одностраничное приложение возвращает требуемые данные в формате JSON

Применение архитектуры SPA имеет следующий ряд преимуществ:

- Высокая скорость работы – во время действий на странице подгружаются лишь необходимые данные, что экономит время загрузки и трафик;
- Гибкость и отзывчивость пользовательского интерфейса – за счет того, что веб-страница всего одна, проще построить насыщенный интерфейс, хранить сведения о сеансе, управлять состояниями представлений и анимацией;
- Кэширование данных – легко выделить часто используемые данные для компонентов и сохранить их на стороне клиента.

Основные недостатки использования архитектуры SPA:

- Усложнена поисковая оптимизация (англ. search engine optimization, SEO) из-за того, что контент загружается при помощи технологии AJAX, которая подразумевает динамическое изменение содержания станицы, а SEO требует решений в виде серверного рендеринга;

- Нагрузка на браузер – клиентские фреймворки являются ресурсоемкими, и могут долго загружаться;
- Необходима включенная поддержка javascript – без нее нельзя полноценно пользоваться полным функционалом приложения.

При анализе недостатков становится очевидно, что они не существенны в рамках проекта. Поисковая оптимизация не требуется так как разрабатываемое приложение является внутренним ресурсом с строго ограниченным доступом. Выбранный клиентский фреймворк Vue.js является одним из лидеров по скорости работы и загрузки. Так как JS и Vue.js уже используется для работы Esprit Web, то архитектура SPA не накладывает новых ограничений. Суммируя вышеописанное – использование SPA несет в себе массу плюсов, а недостатки этой архитектуры клиентской части нивелированы особенностями разрабатываемого проекта.

Единственным минусом использования архитектуры SPA являются трудозатраты на проектирование переиспользуемых компонентов. Однако такой подход является масштабируемым и легко сопровождаемым, что выгодно для долгосрочных проектов.

#### **4.4.2. Проектирование пользовательского интерфейса**

Макет разработанного приложения не должен отличаться от основного приложения – слева находятся текущие доступные разделы, вверху панель навигации, в левом верхнем углу находится логотип компании, а в самом низу располагается футер с информацией об авторских правах (рис. 27). Таким образом был соблюден ряд требований к интерфейсу (I.14.1 - I.14.3)

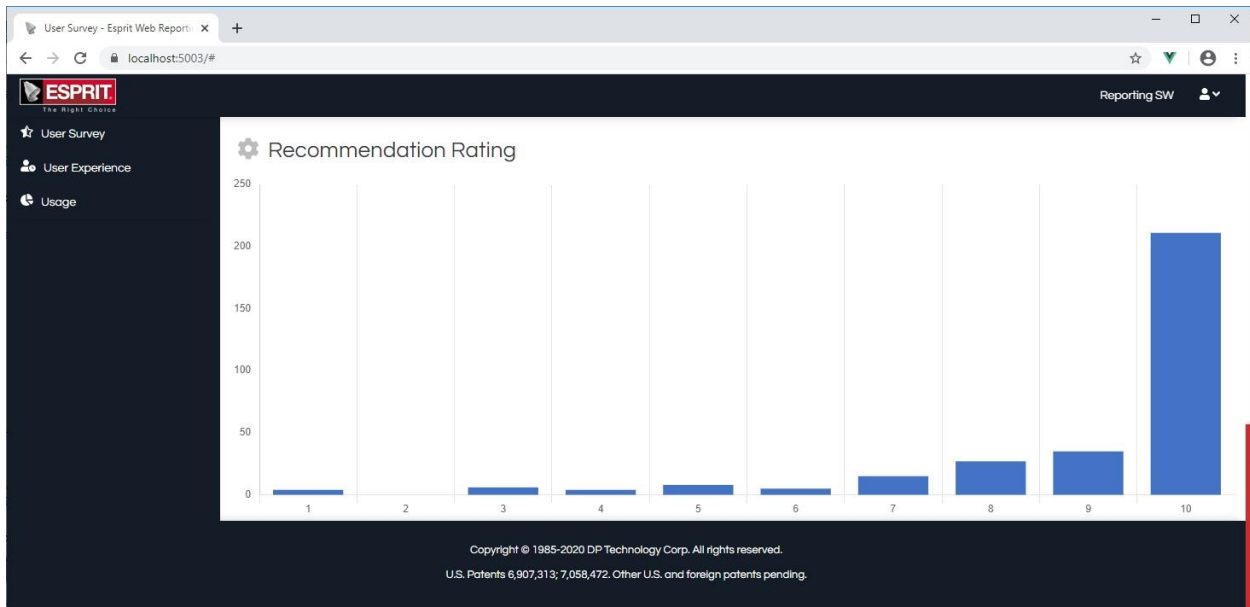


Рисунок 27 – Интерфейс разработанного веб-приложения

Разработанное веб-приложение, согласно требованиям, должно отображать множество диаграмм, данные для которых проходят фильтрацию. Использование по одному компоненту с множеством фильтров на каждую диаграмму не является хорошим решением – заполнение всех фильтров отнимало бы слишком много времени. Было решено сгруппировать схожие диаграммы и выделить по одному компоненту фильтров на всю группу. Такие группы распределены по нескольким разделам SPA для того, чтобы предотвратить разовую загрузку больших объемов данных и оптимально распределить нагрузку на сервер. Распределение представлено в таблице 9.

Таблица 9 – Распределение диаграмм и фильтров по разделам SPA

Страница	Фильтр данных	Диаграммы
User Survey	Стандартный (фильтр по странам, ответственным работникам, дате создания запроса)	Quality Of Technical Response
		Timely Resolution
		Recommendation Rating
User Experience	Стандартный (фильтр по странам, ответственным	Number of Support Requests
		Initial Wait Time

	работникам, дате создания запроса) с предустановленным фильтром временного диапазона за последние 30 дней	Time to Resolve
		Time to Assign
		Owner Time to Respond
		Support Request ReOpened
		Support Request ReAssigned
Usage	Стандартный (фильтр по странам, ответственным работникам, дате создания запроса)	Version
		Type
		Esprit Category – Numbers
		Esprit Category – Hours
	Фильтр по странам, дате создания запроса. Фильтр по ответственным работникам для этих диаграмм не предусмотрен.	Support by Reseller – Hours
		Support by Reseller – Numbers
		Support by Region – Hours
		Support by Region – Numbers

Открыть вышеописанные разделы можно используя кнопки меню навигации (рис. 28):



Рисунок 28 – Меню навигации

Ограничивать отображаемые данные для диаграмм можно с помощью фильтров. Для большинства реализованных диаграмм используется следующий набор фильтров:

- Фильтр запросов по странам обратившихся пользователей;

- Фильтр запросов по назначенным ответственным работникам технической поддержки DP Technology;
- Фильтр запросов по дате создания.

В компоненте присутствует кнопка сброса фильтров «Reset», которая обнуляет выбранные значения и перестраивает диаграммы. Кнопка «Cancel» также очищает поля ввода, но не приводит к перерисовке диаграмм. С помощью кнопки «EU Region» производится выбор всех стран европейского региона. Кнопка «Apply» применяет заданные фильтры и сохраняет их для текущей группы диаграмм, перерисовывает их. Выбранные с помощью «Apply» фильтры сохраняются даже при перезагрузке страницы. Данные кнопки реализованы согласно требованиям F.2.5 - F.2.7. Описанный компонент представлен ниже (рис. 29).

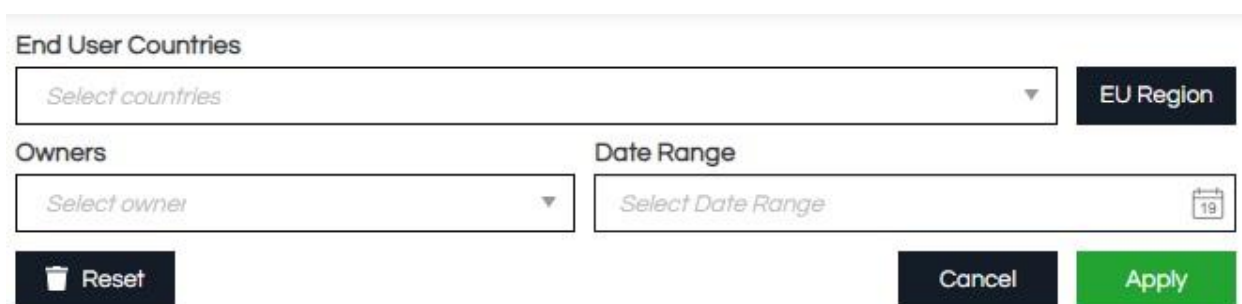


Рисунок 29 – Фильтры запросов на техническую поддержку

Фильтры по странам и работникам являются выпадающими списками с множественным выбором и поддержкой поиска. Они отображают выбранные значения как над полем ввода, так и в выпадающем списке для удобства. Поддержка поиска заключается в том, что в списке присутствуют лишь релевантные значения (рис. 30-31).

### End User Countries

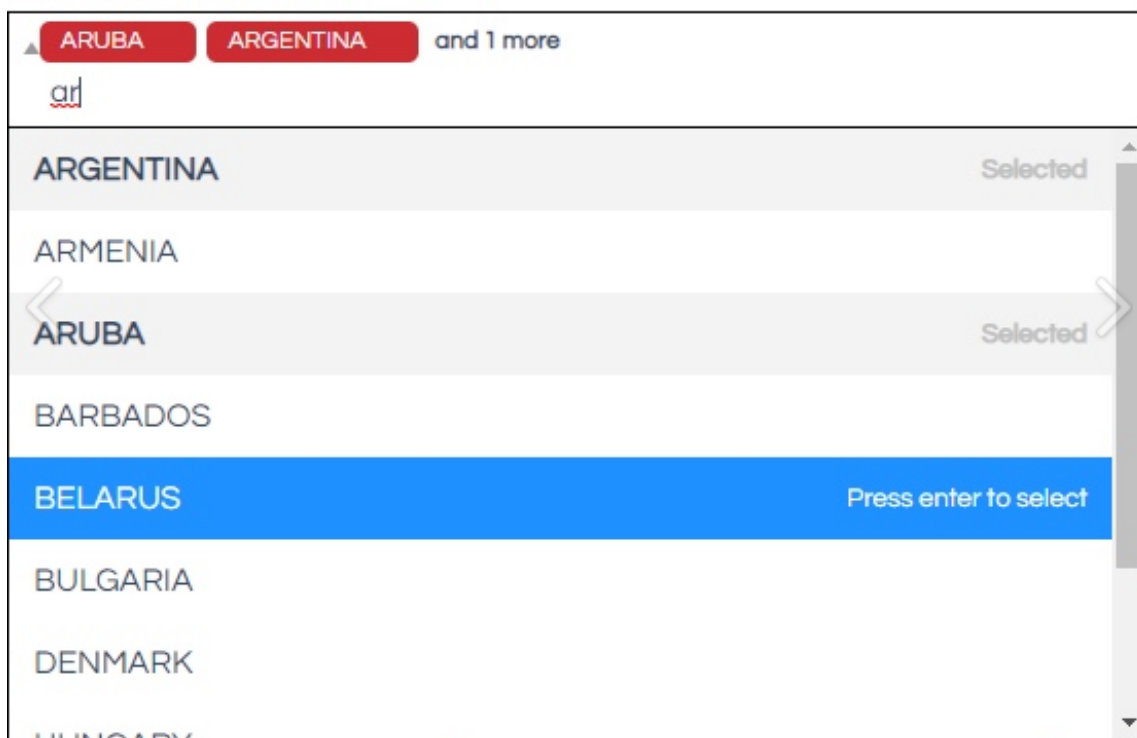


Рисунок 30 – Фильтр по странам

### Owners

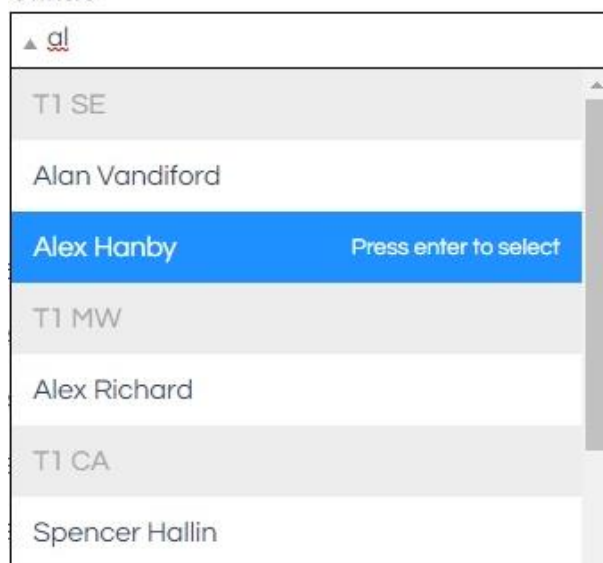


Рисунок 31 – Фильтр по работникам

Реализованный и настроенный компонент vue2-datepicker соответствует указанным требованиям к интерфейсу F.2.3.1- F.2.3.3 (рис.32).

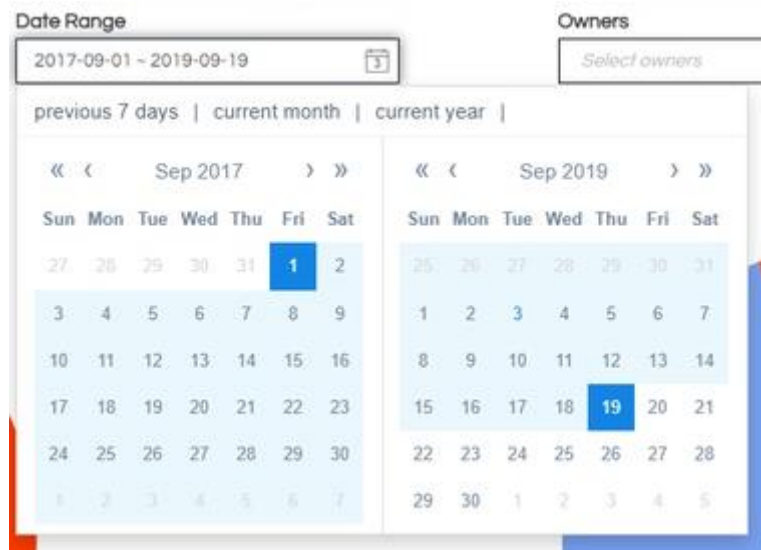


Рисунок 32 – Фильтр временного диапазона

Столбчатые диаграммы, реализованные в приложении, имеют схожий шаблон. Сверху располагается название диаграммы, справа от него находится иконка опций. Данные диаграммы сгруппированы по категориям, которые располагаются на оси X снизу, а высота столбца зависит от значения внутри данной категории и отображается по оси Y (рис. 33).

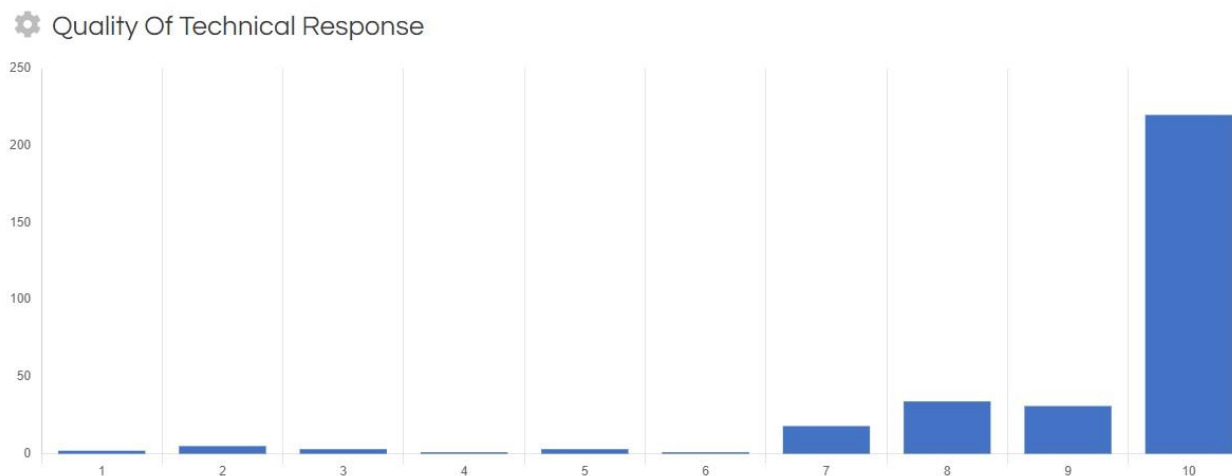


Рисунок 33 – Типовая столбчатая диаграмма

Название и иконка опций для круговых диаграмм располагается также, как и для столбчатых диаграмм – в левом верхнем углу. Категории, по которым сгруппированы значения, представлены секторами различных цветов. Площадь,

которую занимает сектор, отражает собой долю сектора относительно общей суммы значений всех секторов (рис. 34).

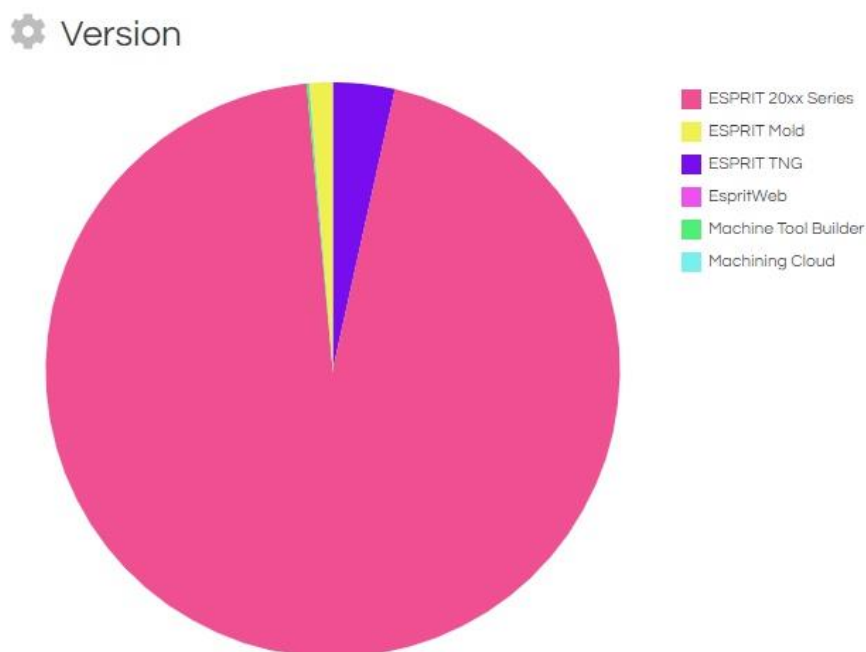


Рисунок 34 – Типовая круговая диаграмма

При наведении на сектора (столбцы) диаграмм отображаются всплывающие подсказки с названием сектора (столбца), то есть категория по оси X и его значение (рис. 35).



Рисунок 35 – Всплывающая подсказка

Для каждой диаграммы предусмотрены надстройки, между которыми можно переключаться. При клике на иконку настроек открывается панель, на



которой можно задать тип отображаемых значений – абсолютные либо относительные; а также тип сортировки – по значениям либо по названиям категорий (рис. 36).



Рисунок 36 – Настройки отображения диаграмм

Интерфейс разработанного веб-приложения является масштабируемым. На широкоформатных экранах с развернутым окном браузера приложение отображается как проиллюстрировано на рисунке 37. При ширине экрана менее 840 пикселей главное меню приложения перестраивается в разворачиваемую панель. Несмотря на то что при маленьком разрешении экрана нет возможности отображать всю легенду диаграммы, при наведении на столбец\сектора всегда можно получить всплывающую подсказку с названием столбца\сектора и его значением (рис. 38).



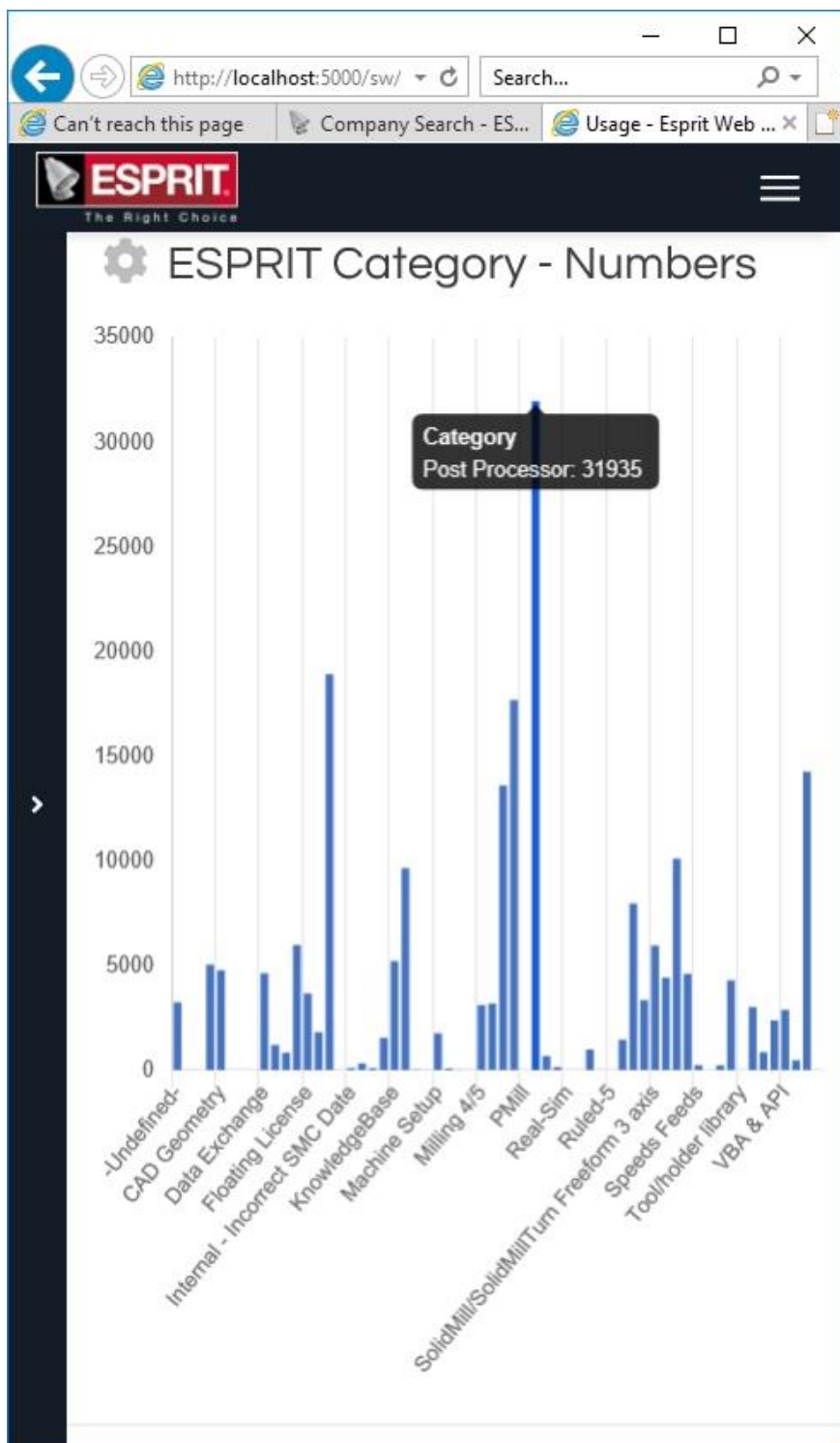


Рисунок 38 – Интерфейс приложения при размере окна в 840x480 пикселей

#### 4.4.3. Проектирование архитектуры серверной части

В данном проекте первичной является клиентская часть, так как заранее известно какие метрики нужно отображать и поставлены требования к

отображению интерфейса. Задачей серверной части является предоставление данных, необходимых клиентской части.

Проекту полностью подходит стек технологий, использованный ранее для Esprit Web, поэтому было решено остановиться на нем. Разработанный в рамках работы проект является ASP.NET Core MVC приложением. Model-View-Controller (MVC, «Модель-Представление-Контроллер») – подход к разделению данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента. В связи с этим модификация каждого компонента может осуществляться независимо. Схема зависимостей компонентов MVC-приложения представлена на UML-диаграмме пакетов (рис. 39).

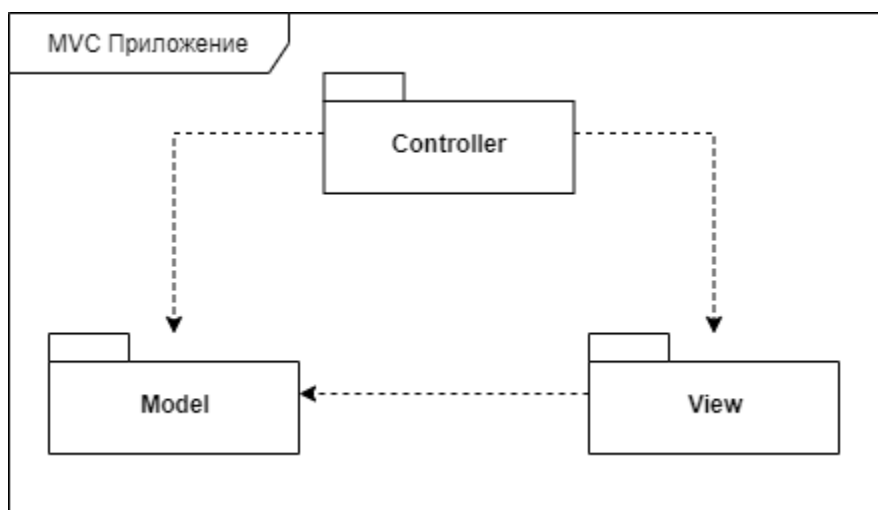


Рисунок 39 – Компоненты MVC-приложения

При проработке архитектуры клиентской части было обосновано использование SPA. Таким образом планируется возвращать единственное представление (View) полностью содержащее в себе SPA. Тогда основная часть методов контроллера будет передавать данные для диаграмм и фильтров в формате JSON для обеспечения работы SPA. Соответственно можно выделить два контроллера – один для отправки представления с SPA, второй для предоставления данных диаграммам и фильтрам.

Так как все фильтры оперируют схожими списками, функционал по предоставлению списков следует вынести в отдельный компонент. Для

предоставления данных для диаграмм недостаточно выделить единственный компонент – он получается слишком объемным и несопровождаемым. Поэтому было использовано три компонента в соответствии с тремя описанными страницами интерфейса – User Survey, User Experience и Usage (см. таблицу 9).

Также следует отдельно описать в строго типизированном виде передаваемые диаграммам и фильтрам объекты. Раздельно реализуются и автоматические Unit-тесты. Результат данного проектирования представлен на диаграмме компонентов (рис. 40).

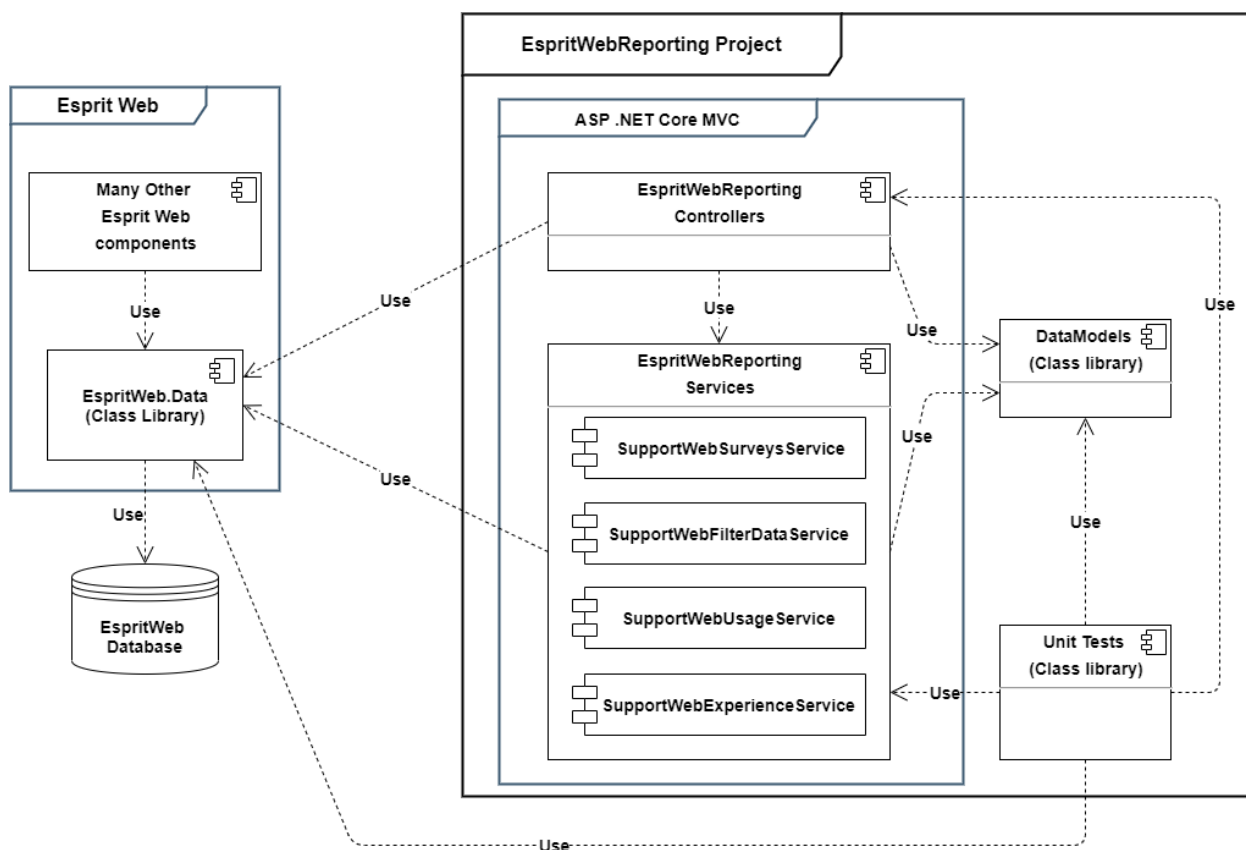


Рисунок 40 – Диаграмма компонентов серверной части проекта

Рассмотрим более подробно связи между компонентами на диаграмме. Контроллеры получают GET запросы на получение данных для обеспечения работы фильтров и отрисовки диаграмм клиентской части. Они используют сервисы, которые обращаются к слою доступа к данным и отвечают за фильтрацию, группировку, округление и сортировку, приведение этих данных к структуре, поддерживаемой диаграммами и фильтрами. Такие структуры будут

описаны в отдельном компоненте проекта. Вышеописанная работа сервисов проверяется в серии Unit-тестов в отдельном компоненте.

## 4.5. Интеграция с приложением Esprit Web

В данном разделе приводится реализация требований к интеграции реализованного приложения и Esprit Web.

### 4.5.1. Реализация механизма междоменной авторизации

Согласно описанным в разделе 2.3 требованию к интеграции Р.1 разработанное приложение должно предоставлять доступ только для авторизованных на сайте Esprit Web пользователей, при этом не запрашивая повторную авторизацию.

Для этого было реализовано совместное использование файлов cookie для проверки подлинности между приложениями. На диаграмме последовательности представлен процесс авторизации клиента веб-сайта (рис. 41).

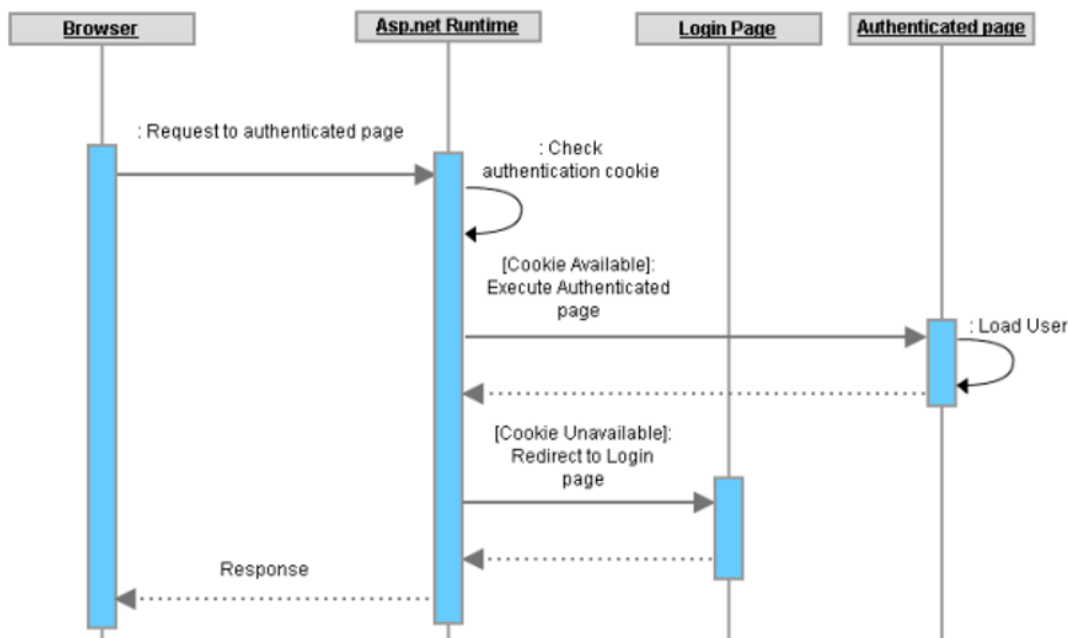


Рисунок 41 – UML-диаграмма последовательности процесса авторизации

При использовании ASP.NET Core для реализации проверки подлинности между приложениями ASP.NET были соблюдены следующие необходимые условия:

- Ключи защиты данных и имя приложения должны быть общими для связанных веб-приложений;
- Для конфигурации cookie-файлов и защиты данных используется метод `ConfigureApplicationCookie` [26].

Для применения вышеописанного типа авторизации доработано как реализованное веб-приложение, так и основное - Esprit Web. Совместное использование файлов cookie для проверки подлинности между приложениями ASP.NET Core было успешно реализовано.

#### **4.5.2. Обеспечение интеграции баз данных**

Менеджментом было решено, что после внедрения нового функционала отзывчивость основного сервера для веб-приложения Esprit Web не должна измениться. Таким образом основное приложение и модуль для отображения статистики должны физически находиться на разных серверах. Поэтому модуль отображения статистики реализован как отдельное веб-приложение.

Как отмечалось в требованиях к интеграции, разрабатываемое приложение должно работать с основной БД, что необходимо для обеспечения актуальности отображаемой статистики.

В ходе работы был использован уже реализованный слой доступа к данным, при этом клиентская и серверная часть реализованного приложения не затрагивает приложение Esprit Web, как и требовалось. Это отражено на диаграмме компонентов (рис. 42), где реализованное приложение, выделенное зеленой рамкой, находится на отдельном сервере.

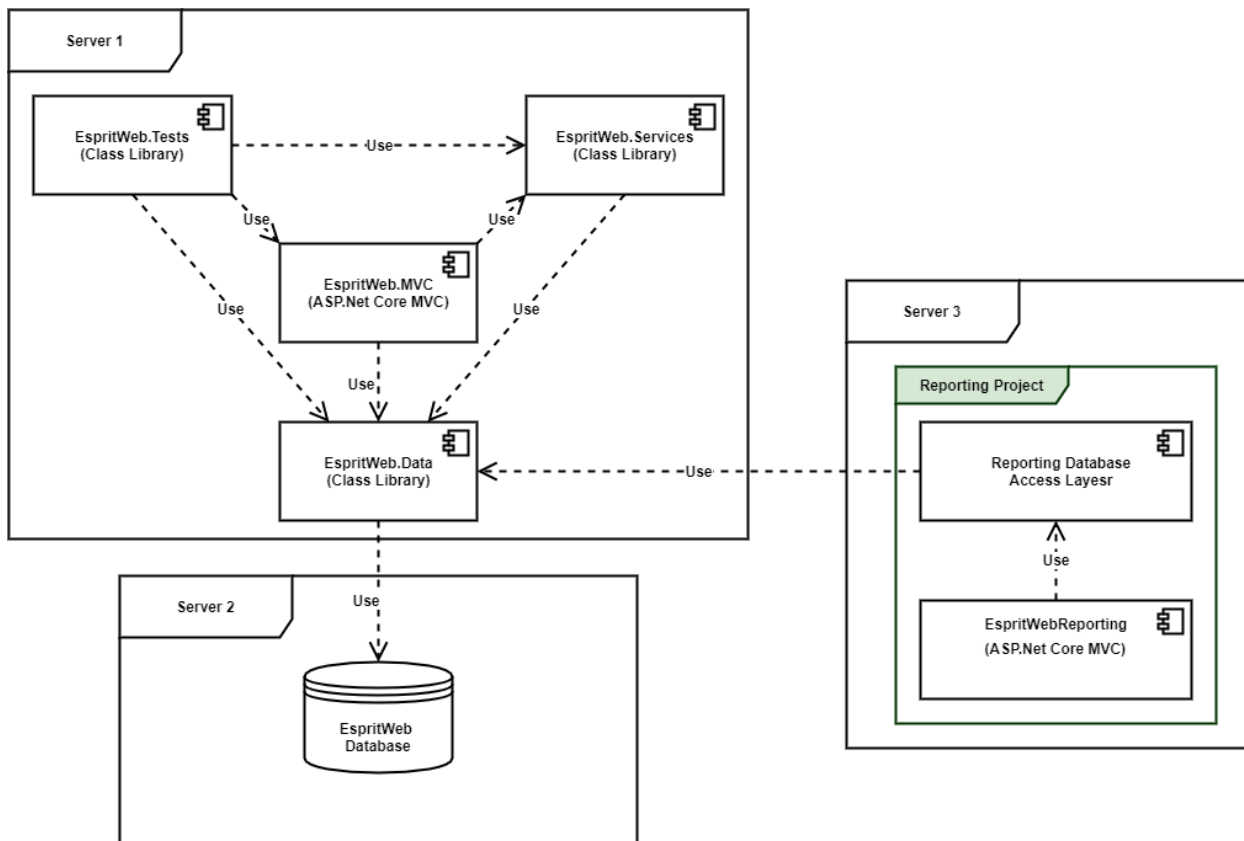


Рисунок 42 – UML-диаграмма компонентов с отображением задействованных серверов

#### 4.6. Программная реализация серверной части

Реализация серверной и клиентской частей начались одновременно. Однако, сперва рассмотрим реализацию серверной части по той причине, что она предоставляет данные для работы клиента веб-приложения.

Для получения этих необходимых данных используются SQL запросы, сгенерированные с помощью ORM EF Core. Данные, полученные этими запросами, преобразуются к объектно-ориентированной модели. Проект EspritWeb.Data содержит классы, обеспечивающие доступ к базе данных, которые используются в методах проекта EspritWebReporting, где и реализована основная бизнес-логика.

ORM контекст БД уже был реализован в приложении Esprit Web, поэтому рассмотрим лишь бизнес логику по получению и обработке данных,



реализованную в рамках данной работы. Основные классы и интерфейсы серверной части представлены ниже на UML-диаграмме классов (рис. 43):

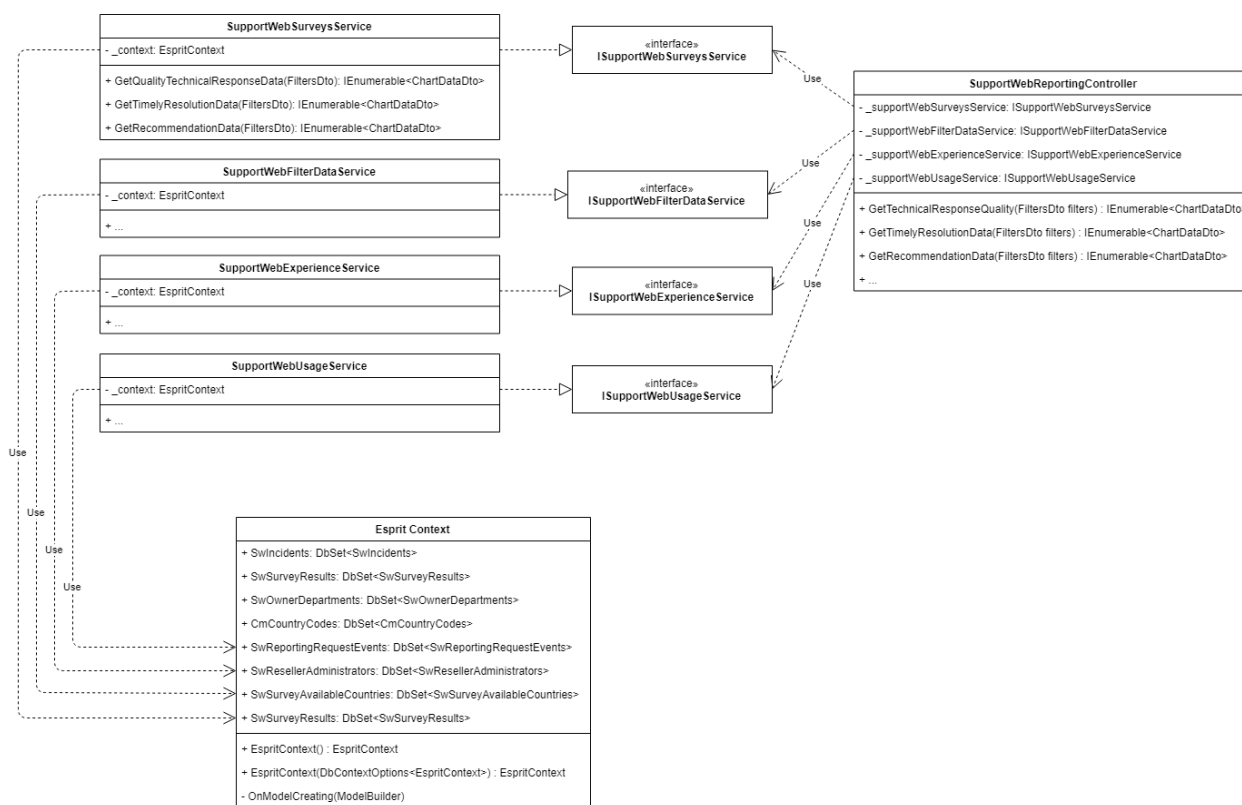


Рисунок 43 – UML-диаграмма основных классов серверной части

Далее представлено краткое описание классов и интерфейсов, отображенных на диаграмме выше.

EspritContext – класс контекста ORM (разработан ранее и задействован в проекте), обеспечивающий выполнение следующих операций:

- Создание и выполнение запросов;
- Представление результатов запроса в форме строго типизированных объектов;
- Отслеживание изменений в модели данных;
- Сохранение изменений в БД.

SupportWebReportingController – класс контроллера, является центральным компонентом в архитектуре MVC. Он принимает запросы с клиента, обрабатывает их и посылает обратно результаты обработки в виде представления (HTML-документа), либо сериализованные в формат JSON

данные. Класс использует реализации нижеописанных интерфейсов для получения необходимых данных;

`ISupportWebSurveysService` – интерфейс, предоставляющий методы для получения данных о результатах опросов пользователей относительно качества технической поддержки. Класс `SupportWebSurveysService` реализует данный интерфейс;

`ISupportWebFilterDataService` – интерфейс, предоставляющий методы для получения списков доступных вариантов в фильтрах по странам, работникам и т.д. Класс `SupportWebFilterDataService` реализует данный интерфейс;

`ISupportWebExperienceService` – интерфейс, предоставляющий методы для получения данных о среднем времени ожидания ответа от техподдержки и среднем времени решения запроса в техническую поддержку. Класс `SupportWebExperienceService` реализует данный интерфейс;

`ISupportWebUsageService` – интерфейс, предоставляющий методы для получения данных о количестве запросов в поддержку, сгруппированным по странам, категориям ошибок, версиям продукта. Класс `SupportWebUsageService` реализует данный интерфейс;

Далее рассмотрим подробнее механизмы получения данных для диаграмм. После заполнения фильтров на форме (рис. 44) пользователь нажимает кнопку «Apply» и запрос отправляется на сервер.

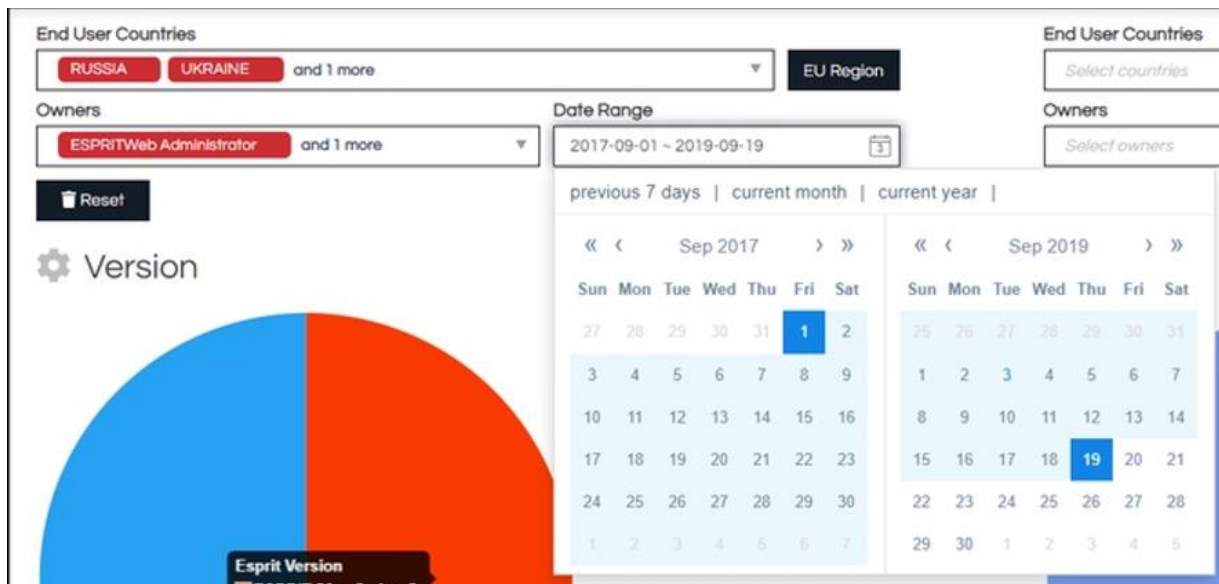


Рисунок 44 – Заполнение фильтров

Запрос авторизованного пользователя принимается в контроллере. Экземпляр контроллера создается для каждого пользовательского запроса и с помощью механизма Dependency Injection он получает необходимые реализации интерфейсов, которым и передает обработку.

На блок-схеме ниже описан алгоритм получения и обработки данных для диаграмм в его самом сложном варианте исполнения (рис. 45).

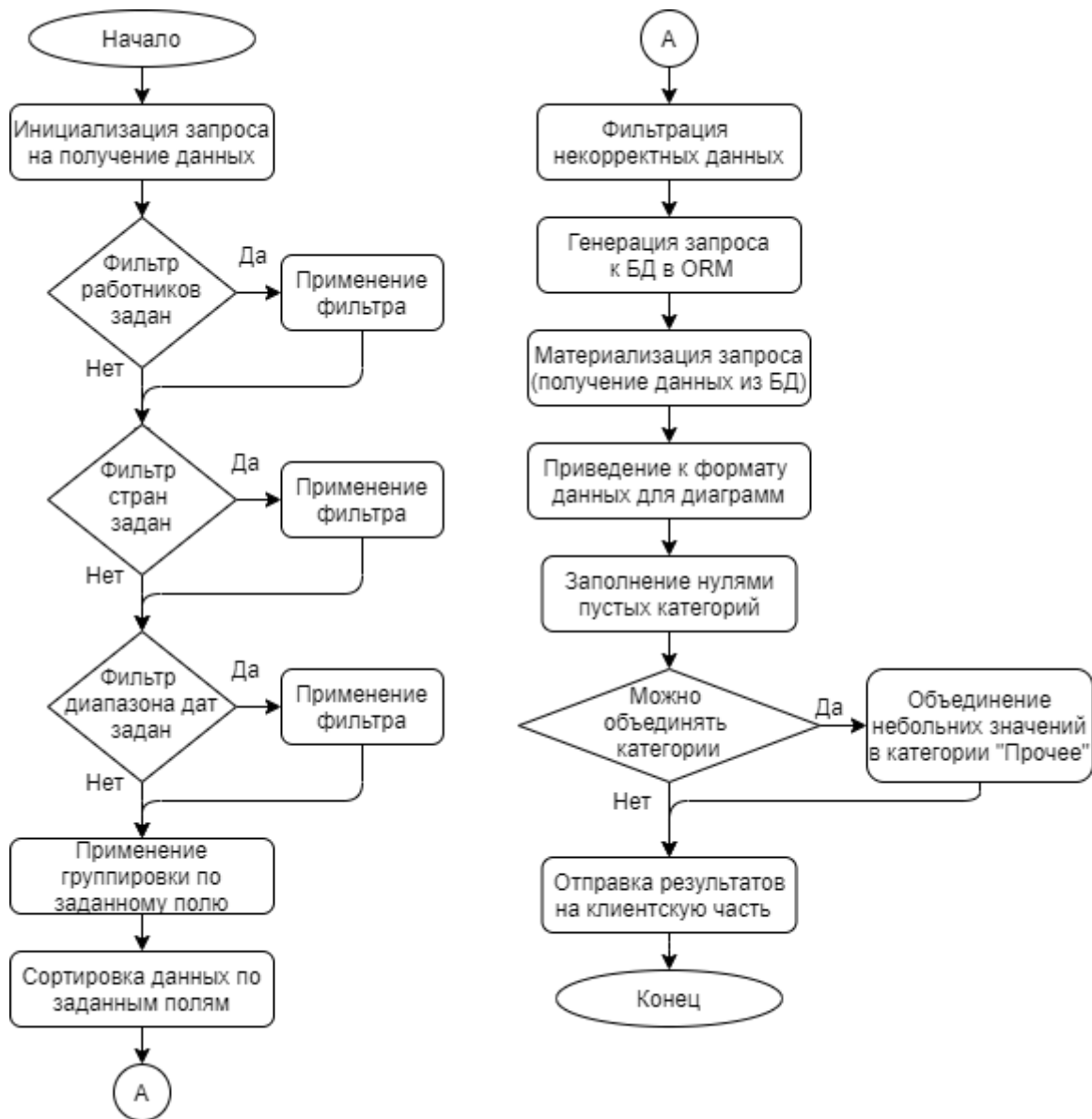


Рисунок 45 – Блок-схема алгоритма получения и обработки данных требуемых для отображения диаграмм

То, что происходит в начале алгоритма сводится к созданию динамического запроса в БД. В LINQ деревья выражений используются для представления структурированных запросов к источникам данных, которые реализуют интерфейс `IQueryable<T>`. Например, поставщик LINQ реализует интерфейс `IQueryable<T>` для выполнения запросов к реляционным хранилищам данных. Компилятор `C#` компилирует запросы к таким источникам данных в код, который строит дерево выражения во время выполнения. Поставщик запросов

может переходить по структуре данных дерева выражения и преобразовать ее в язык запросов, соответствующий источнику данных [19].

После построения фильтрующего выражения происходит группировка данных в соответствии с категориями, которые нужно отображать на диаграмме. Затем производится сортировка данных. Чаще всего в работе использовалась сортировка по названиям категорий либо по датам создания запросов.

При работе со старыми данными возникали непредвиденные ситуации в связи с тем, что до 2010 года валидация некоторых вводимых значений не была реализована корректно. Например, в БД попадали чрезмерно большие (свыше 1000 часов), а также нулевые значения времени обработки запроса, которые на тот момент выставлялись самим работником. Такие данные были исключены на этапе фильтрации.

Когда необходимые LINQ деревья выражений настроены для фильтрации данных, можно производить генерацию SQL-запроса и получать данные из базы. ORM преобразует результат SQL-запроса в типизированные объекты из модели данных. Затем полученные данные нужно преобразовать к такой структуре, которую может принять на вход компонент диаграммы в клиентской части приложения.

Иногда предоставленные данные нужно дополнить. Например, при фильтрации и группировке пользовательских оценок работы технической поддержки (по шкале от 1 до 10) в выборке не оказалось четверок и единиц. Эти столбцы просто не будут отображены на диаграмме, что не корректно. В связи с этим такие значения будут найдены и проинициализированы нулями при постобработке.

Разработанный алгоритм также предоставляет возможность объединять значения, которые меньше некоторой границы в единую категорию «Other». Это позволяет сфокусироваться на значимых данных, уменьшить число категорий в диаграмме и ее легенду. Данная опция реализована для нескольких диаграмм, в которых такое «слияние» категорий является уместным.

#### 4.7. Программная реализация клиентской части

Клиентская часть приложения была реализована согласно спроектированной архитектуре и состоит из ASP.NET cshtml страницы с подключенными Vue.js компонентами и представляет собой SPA.

Для навигации в клиентской части использовалась официальная библиотека маршрутизации для Vue.js – Vue Router, которая интегрируется с Vue.js и позволяет легко создавать SPA-приложения. Она включает следующие возможности:

- Вложенные маршруты/представления;
- Модульная конфигурация маршрутизатора;
- Доступ к параметрам маршрута;
- Анимация переходов, контроль навигации и т.д.

Для обеспечения модульности и повторного использования кода были выделены следующие Vue-компоненты, описанные в таблице 10.

Таблица 10 – Компоненты клиентской части

Выделенный Vue-компонент	Описание компонента
ClientApp.vue	Компонент является корневым контейнером для всего клиентского приложения.
Router.vue	Компонент отвечающий за навигацию между разделами разработанного SPA.
Layout.vue	Компонент отвечающий за разметку и расположение контента на странице.
Header.vue	Компонент отвечающий за отображение «шапки» приложения, в которой размещаются ссылки на основные разделы сайта, лого и т.д.
UserMenu.vue	Компонент отвечающий за отображение меню текущего аккаунта с возможностью прекратить сеанс

	работы в качестве зарегистрированного пользователя, перейти на страницу пользовательских настроек и т.д.
Body.vue	Компонент является контейнером для отображения контента, именно он перезагружается при переходе между разделами SPA.
Footer.vue	Компонент является контейнером для отображения футера с информацией об авторских правах и головном офисе DP Technology.
SupportWeb Component.vue	Компонент отвечающий за отображение меню навигации между разделами SPA с разнообразными диаграммами, сгруппированными по смысловой нагрузке.
UserSurvey.vue	Компонент отвечающий за отображение ряда диаграмм, сгруппированных по смысловой нагрузке
UserExperience.vue	Компонент отвечающий за отображение ряда диаграмм, сгруппированных по смысловой нагрузке
Usage.vue	Компонент отвечающий за отображение ряда диаграмм, сгруппированных по смысловой нагрузке
Chart.vue	Компонент отвечающий за отображение любых видов диаграмм. Создан для сохранения общих настроек и логики работы с диаграммами и их данными
Pie.vue	Компонент отвечающий за логику работы и за отображение круговых диаграмм.
PieLegend.vue	Компонент отвечающий за отображение легенды круговых диаграмм.
Bar.vue	Компонент отвечающий за логику работы и за отображение столбчатых диаграмм.

Filters.vue	Компонент отвечающий за логику работы фильтров и их отображение.
DatePicker	Компонент отвечающий за отображение календаря для выбора диапазона дат.
Multiselect	Компонент отвечающий за отображение выпадающего списка с множественным выбором и поддержкой поиска.
Error.vue	Компонент отвечающий за отображение сообщения об ошибке либо отсутствии прав доступа.

Здесь освещены лишь основные компоненты, отвечающие за отображение элементов интерфейса и работу с данными. На диаграмме ниже представлены связи между ними (рис. 46).



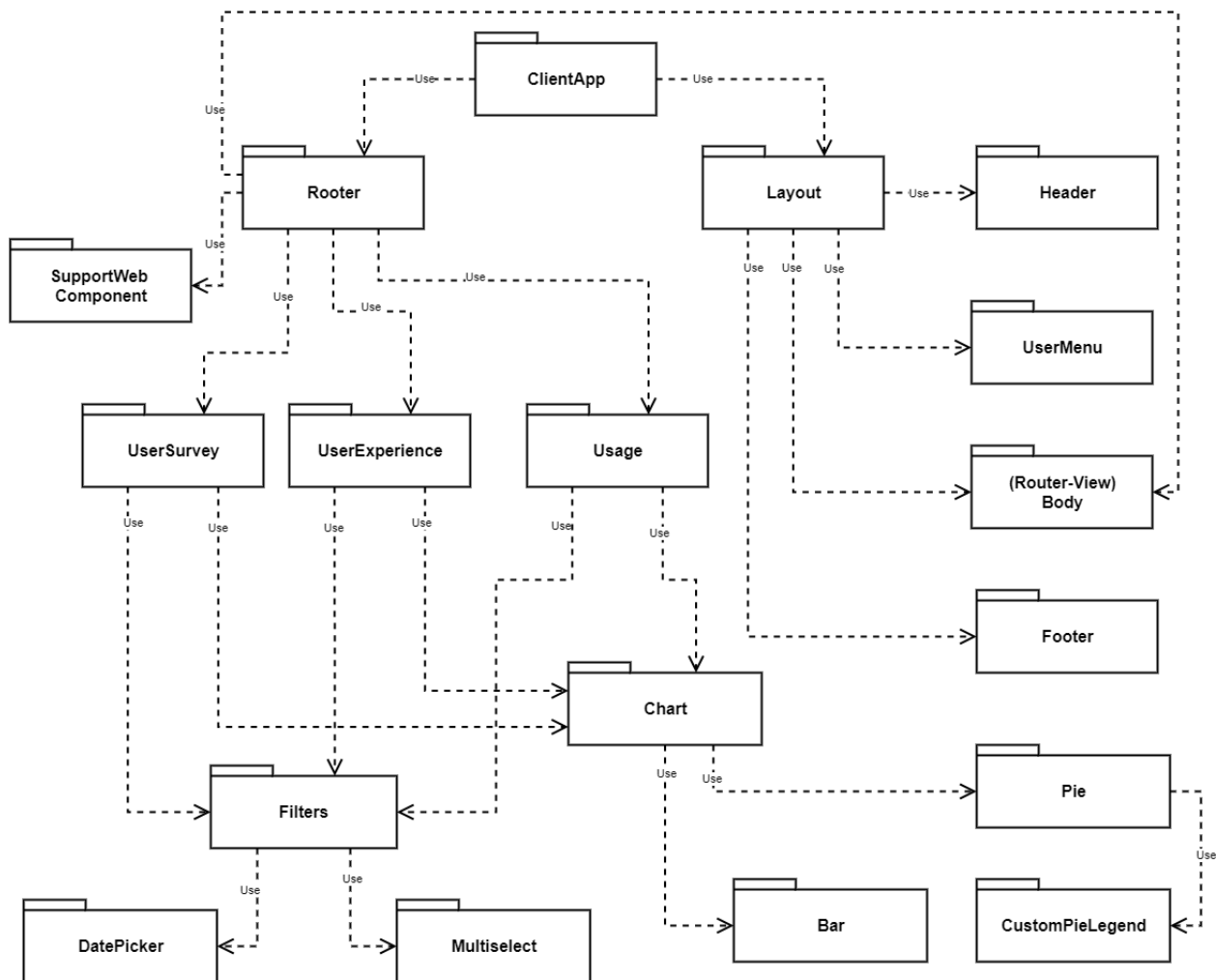


Рисунок 46 – UML-диаграмма пакетов клиентской части приложения

Согласно спроектированной структуре, внешний вид страницы формируется из частей компонента Layout. После загрузки SPA такие части как Header, UserMenu, Footer остаются неизменны и не нагружают трафик. При навигации в SPA приложении в компонент «Router-View» подставляются наборы диаграмм в зависимости от выбранного раздела. Диаграммы хранят общую логику в компоненте Chart, а специфическую логику для круговых и столбчатых диаграмм в Pie и Bar компонентах соответственно. Фильтры используют календарь DatePicker и список с множественным выбором Multiselect для отображения и обработки вводимых данных.

После загрузки SPA при навигации будет перезагружаться только компонент «Router-View». Подключенная и настроенная библиотека Vue Router перезагружает его в зависимости от выбранной страницы. В «Router-View»

могут быть представлены наборы диаграмм UserSurvey, либо UserExperience, либо Usage.

Данные для диаграмм в этих разделах подгружаются независимо и асинхронно на момент инициализации диаграммы во vue-компоненте. Они передаются небольшими порциями, и пользователь видит отзывчивость сайта даже при низкой пропускной способности соединения (рис. 47).

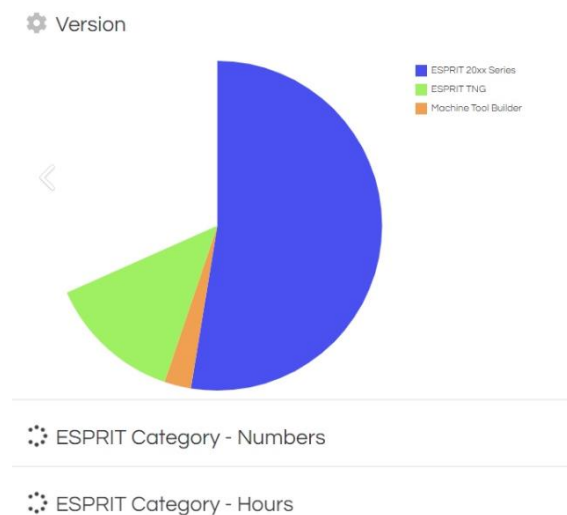


Рисунок 47 – Асинхронная загрузка данных и отрисовка

Для реализации такой загрузки данных был создан API-сервис, предоставляющий данные с использованием Axios – широко известной JavaScript-библиотеки, представляющей собой HTTP-клиент, основанный на JavaScript Promise и предназначенный для браузеров. Объект Promise служит связующим звеном между исполнителем (создающим требуемые данные кодом) и функциями-потребителями, которые получают либо корректный результат, либо ошибку.

Рассмотрим более подробно задействованные компоненты для обеспечения работы и отображения диаграмм и фильтров. Помимо API-сервиса для предоставления данных были разработаны также несколько других вспомогательных компонентов, основные из них отображены на UML-диаграмме пакетов (рис. 48).

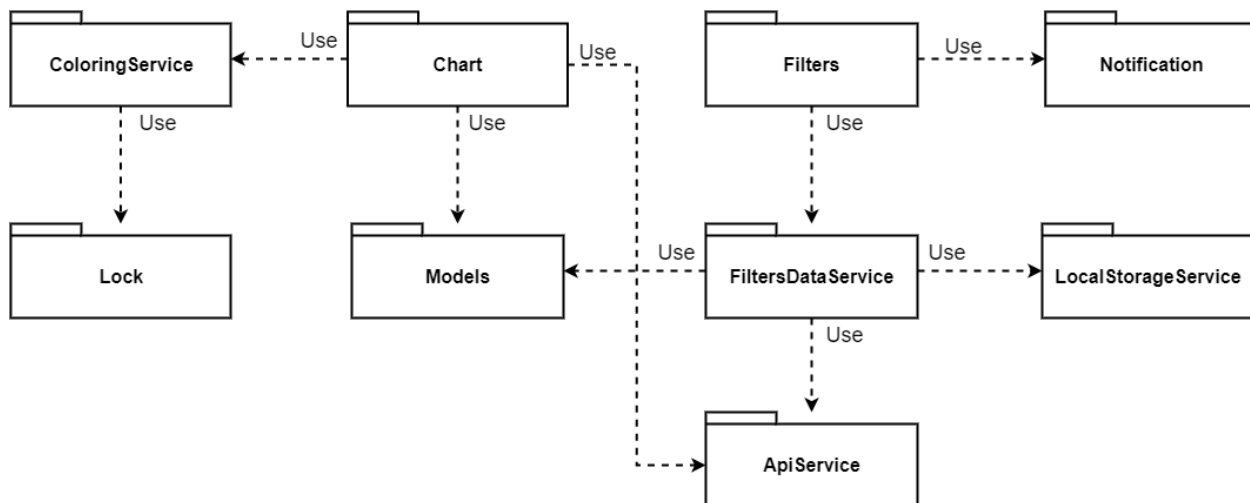


Рисунок 48 –Вспомогательные компоненты для диаграмм и фильтров

Для окрашивания секторов диаграмм в разные цвета был разработан ColoringService. При использовании одинаковых категорий на диаграммах, эти категории будут окрашены в одинаковые цвета. Например, в разделе Usage есть несколько круговых диаграмм, разбитых на сектора по странам. Одинаковые страны на этих диаграммах окрашены в одинаковые цвета.

Для каждой из цветных диаграмм создается словарь из пар категория-цвет. Чтобы одинаковые категории не могли получить разные цвета при одновременной обработке данных, ColoringService использует мьютекс, реализованный в модуле Lock. Благодаря этому одновременно может окрашиваться лишь одна диаграмма. Генератор цветов был разработан таким образом, чтобы соседние сектора получали непохожие друг на друга оттенки – генерируя цвета внутри цикла, он использует сильно различающиеся RGB компоненты для соседних итераций.

На рисунке 48 более подробно описаны компоненты, поддерживающие работу фильтров. С помощью LocalStorageService выбранные фильтры сохраняются в сериализованном виде в LocalStorage на клиентской машине. Фильтры сохраняются между активными сеансами и вкладками, так как объекты LocalStorage не имеют срока давности и не очищаются автоматически. Фильтры разных разделов проекта применяются для разных диаграмм и сохраняются каждый отдельно.



При использовании SPA была достигнута точка, когда разработанный проект состоял из нескольких десятков .js, .css и многих других файлов. В связи с чем возникла проблема управления порядком загрузки файлов/модулей, для обеспечения корректной работы приложения. Файлы требуется подключать согласно графу зависимостей – дереву, описывающему порядок загрузки. Для решения этой проблемы был использован Webpack.

Webpack – это инструмент, основная цель которого состоит в том, чтобы собрать все .js файлы в любое нужное количество пакетов (bundles), а также убедиться, что в собранном пакете файлы находятся в правильном порядке. Помимо этого, Webpack был также сконфигурирован для решения следующих задач:

- Перевод файлов из ES6 в ES5 для поддержки старых браузеров;
- Минификация файлов (уменьшает размер файлов удаляя форматирование и подменяя имена переменных);
- Удаление всех комментариев из всех файлов (для дальнейшего уменьшения размера файла);
- Добавление пользовательских шрифтов и иконок в выходной файл (для доступа без задержки, связанной с дополнительными HTTP запросами);
- Размещение кода проекта в один js файл, а всех используемых библиотек в другие js файлы (для кэширования в браузере);
- Сбор всех .css файлов в один файл (bundle).

С целью обеспечения единого стиля написания кода и проверки на ошибки был использован ESLint – инструмент, который позволяет проводить анализ качества кода, написанного на любом выбранном стандарте JavaScript. Он приводит код к единому стилю, помогает избежать ошибок, умеет автоматически исправлять некоторые найденные проблемы.

Все спроектированные диаграммы поддерживают масштабирование и корректно отображаются в браузерах Chrome, Mozilla Firefox, Opera, Safari, Edge

и Internet Explorer. При этом не было обнаружено никаких проблем с их отображением на мобильных устройствах.

Выбранные пользователем фильтры корректно сохраняются для каждой группы диаграмм. Также фильтры разных диаграмм часто оперируют одинаковыми списками, поэтому они получают данные с помощью реализованного сервиса, который реализует кэширование. Для того чтобы не ждать полной загрузки всех данных была реализована асинхронная загрузка диаграмм и данных для фильтров.

Типичный набор данных для отображения диаграммы состоит из

- Массива имен категорий/секторов;
- Объекта набора значений, состоящего из:
  - Имени набора значений;
  - Массива значений;
  - Цвета либо массива цветов заливки столбцов/секторов.

В ходе реализации проекта поступили новые пожелания по отображению сгруппированных диаграмм. При этом вместо одного объекта набора значений используется их массив. В результате были созданы следующие диаграммы, приведенные на рисунках 50-51.



Рисунок 50 – Сгруппированная диаграмма пользовательских оценок

Average Processing Time for Closed Requests

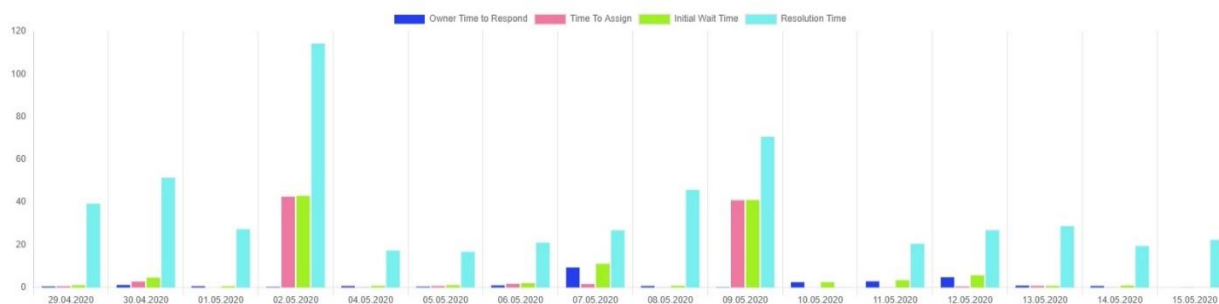


Рисунок 51 – Сгруппированная диаграмма времени обработки запроса

Внедрение Chart.js потребовало доработок в функционале отображения диаграмм и всплывающих подсказок. В итоге требования по обеспечению нужной производительности и по отображению интерфейса были соблюдены.

#### 4.8. Тестирование приложения

В проекте разработана серия Unit-тестов на языке C# для сервисов, предоставляющих данные. При разработке тестовых методов использовалась библиотека xUnit – open-source библиотека для .NET приложений с качественной документацией [30].

Разработанные Unit-тесты проверяют на корректность следующие функциональные особенности веб-приложения:

- Использование пользовательских фильтров;
- Применение группировки данных;
- Округление числовых данных;
- Фильтрацию нерепрезентативных и ошибочных данных;
- Слияние незначительных категорий данных в категорию «Прочее».

Для большинства тестов был использован подход AAA (arrange, act, assert). Он позволяет разделить тестовый метод на составляющие с разной областью ответственности и повысить читаемость:

- Arrange – часть метода ответственная за создание начальных условий перед осуществлением действия;

- Act – часть метода в которой происходят сами действия, результат которых должен быть протестирован;
- Assert – часть метода, отвечающая за проверку выходных данных на корректность и соответствие ожиданиям разработчика.

Разработанные тесты не используют в своих проверках рабочую БД. Во время прохождения тестов для каждого из них используются вновь сгенерированная In-Memory база данных. Такой подход требует использования интерфейсов и DI внутри проекта, а также разработки механизма автозаполнения данных (Seeding), однако обеспечивает следующие преимущества:

- Тесты атомарны и независимы друг от друга;
- Тесты не нагружают и не влияют на рабочую БД;
- Входные данные для каждого теста легко настраиваемы и независимы.

Во время работы приложение использует основную БД, а во время прохождения автоматического Unit-тестирования используется In-Memory БД. Это отображено на UML-диаграмме компонентов (рис. 52), где использование основной БД представлено связями синим цветом, а In-Memory БД – красным.



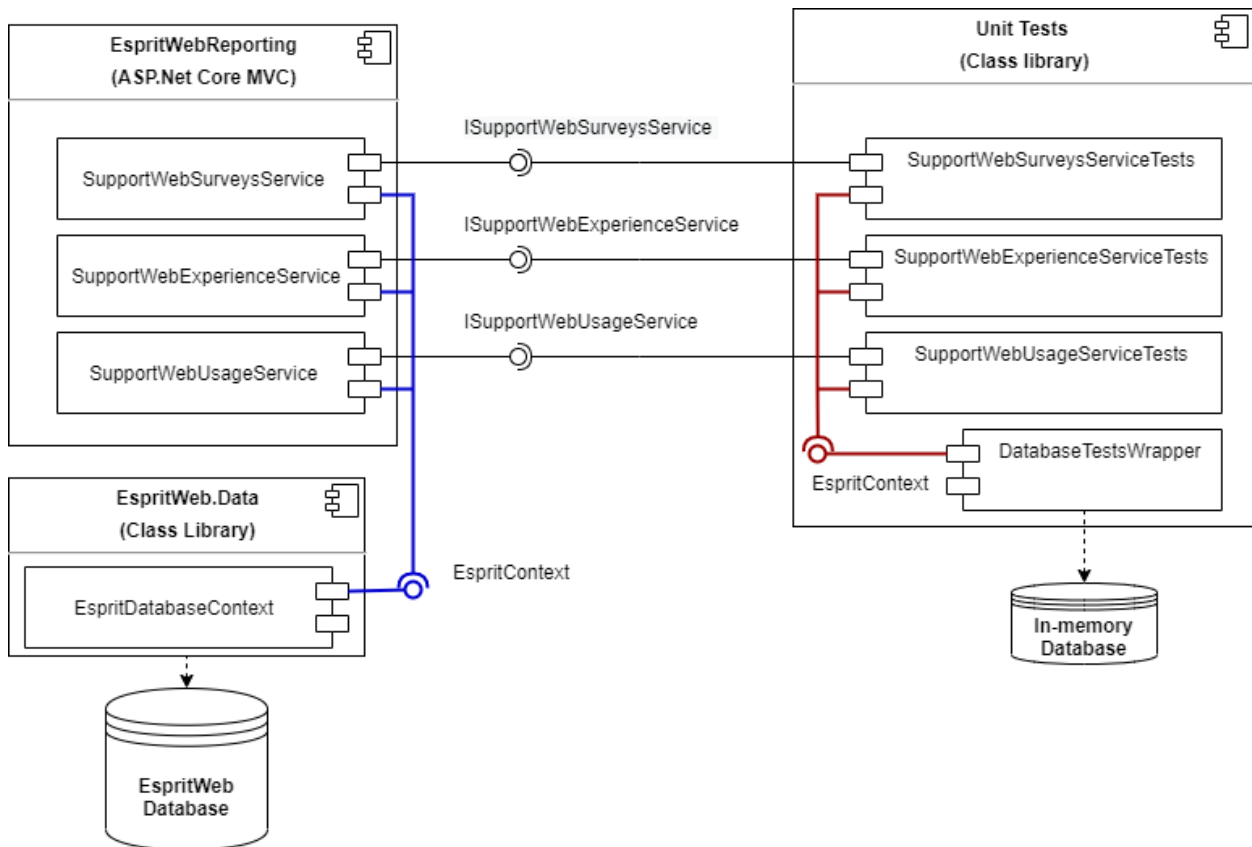


Рисунок 52 – Использование In-Memory базы данных в Unit-тестах

При добавлении изменений в кодовую базу происходит сборка всего проекта на сервере TeamCity, то есть для проекта настроена непрерывная интеграция. Последним шагом перед публикацией результатов сборки является автоматическое прохождение всех Unit-тестов. Таким образом не пройденные тесты будут сигнализировать что новыми вносимыми изменениями была изменена закреплённая в тестах логика, что помогает избежать ошибок и сужает область регрессионного ручного тестирования.

На рисунке ниже можно наблюдать что актуальная сборка проекта проходит все реализованные тестовые методы (рис. 53).

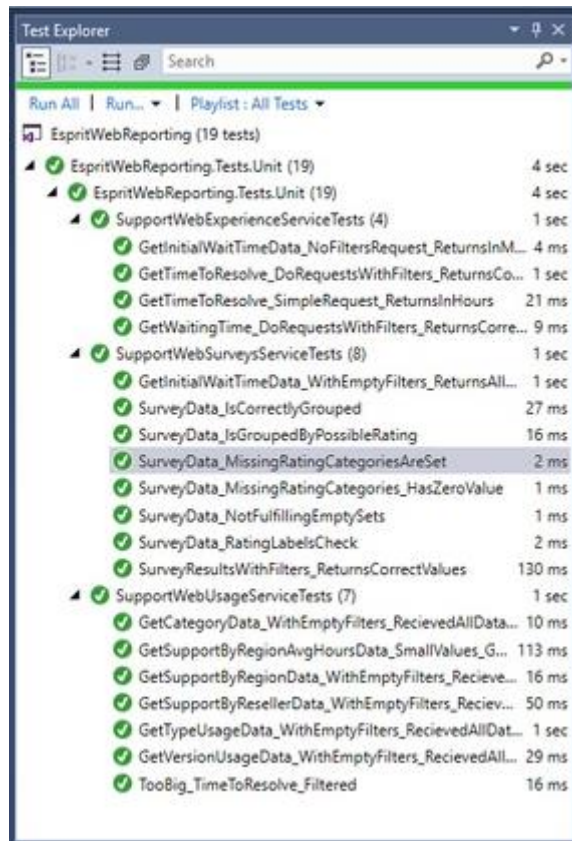


Рисунок 53 – Прохождение Unit-тестов для текущей сборки

## **5. Финансовый менеджмент и ресурсоэффективность**

### **5.1. Оценка коммерческого потенциала и перспективности**

#### **реализации модуля с позиции ресурсоэффективности и ресурсосбережения**

При выполнении данной магистерской диссертации был разработан модуль анализа и оценки эффективности технического обслуживания системы Esprit.

Целью данного раздела является комплексное описание и анализ финансово-экономических сторон проведенной работы. Для этого были проведены:

- Анализ конкурентных технических решений;
- SWOT-анализ;
- Анализ альтернативных путей развития проекта;
- Расчет стоимости реализации проекта.

#### **5.1.1. Потребители результатов реализации программного модуля**

Целевым потребителем программного модуля является компания ООО «DP Technology», работающая над повышением качества работы отдела технической поддержки.

Руководством предприятия была поставлена задача по автоматизации сбора и представления данных по КПИ процесса оказания технической поддержки. Сбор и представление этой информации в читаемом виде ранее требовал участия специалиста для написания запросов к БД и оформления результатов в виде графиков и диаграмм.

При решении поставленной задачи рассматривались варианты использования стороннего программного обеспечения наряду с разработкой собственного модуля по автоматизации данного процесса.

### 5.1.2. Анализ конкурентных программных решений

В данный момент на рынке существует множество аналогов, позволяющих визуализировать данные из реляционных БД, представляя их в виде графиков и диаграмм. При анализе предметной области и конкурентных технических решений были выделены и рассмотрены самые качественные разработки:

- Сервис «Google Data Studio» (К1);
- Приложение «Plotly» (К2);
- Приложение «Power BI» (К3).

В приложении В представлена таблица сравнения данных программных продуктов. Оценочная карта для сравнения конкурентных программных решений представлена в таблице 11.

Таблица 11 – Оценочная карта для сравнения конкурентных программных решений

Критерии оценки	Вес критерия (В)	Баллы				Конкурентоспособность			
		Бф	Бк1	Бк2	Бк3	Кф	Кк1	Кк2	Кк3
<b>Технические критерии оценки ресурсоэффективности</b>									
Степень автоматизации процесса сбора информации	0,2	5	3	3	5	1	0,6	0,6	1
Степень автоматизации процесса представления информации	0,2	5	4	5	3	1	0,8	1	0,6
Функциональная мощность	0,2	3	3	4	5	0,6	0,6	0,8	1
Простота использования	0,1	5	4	2	1	0,5	0,4	0,2	0,1
Гибкость настройки визуальных компонентов	0,1	5	4	4	5	0,5	0,4	0,4	0,5
Производительность и скорость работы	0,1	4	4	3	5	0,4	0,4	0,3	0,5

<b>Итого</b>	<b>1</b>					<b>4,0</b>	<b>3,2</b>	<b>3,3</b>	<b>3,7</b>
<b>Экономические критерии оценки ресурсоэффективности</b>									
Стоимость внедрения решения	0,2	1	4	4	4	0,2	0,8	0,8	0,8
Поддержка продукта	0,6	5	2	2	2	3	1,2	1,2	1,2
Сроки внедрения	0,2	2	5	4	3	0,4	1	0,8	0,6
<b>Итого</b>	<b>1</b>					<b>3,6</b>	<b>3,0</b>	<b>2,8</b>	<b>2,6</b>

Анализ конкурентных технических решений определяется по формуле:

$$K = \sum V_i \cdot B_i, \quad (5.1)$$

Где  $K$  – конкурентоспособность реализованного проекта или конкурента;

$V_i$  – вес показателя (в долях единицы);

$B_i$  – балл  $i$ -го показателя.

Основным недостатком конкурентных программных продуктов можно назвать невозможность достижения требуемого уровня автоматизации процесса сбора и представления требуемых данных – программное обеспечение, рассчитанное на широкое применение не в состоянии учесть все особенности конкретного бизнес-процесса. В большинстве решений нельзя настроить систему фильтров, группировок и сортировок для получения лишь необходимых данных. Вторым недостатком конкурентов является стоимость поддержки – для покрытия всех выявленных требований к модулю подходят лишь платные версии продуктов. Также для нескольких конкурентных решений проблемой является недостаточно гибко настраиваемая составляющая по визуализации полученных данных.

Разрабатываемый модуль предоставляет узкий спектр функциональных возможностей, однако он учитывает все тонкости бизнес-процессов предприятия и сводит к минимуму стоимость поддержки программного решения в долгосрочной перспективе. Интерфейс системы разработан по пожеланиям ее пользователей, что обеспечивает максимальную простоту и удобство

использования. Таким образом, высокая степень автоматизации процессов и низкая стоимость поддержки обеспечивает преимущество над разработками конкурентов.

Оценка основных технических и экономических характеристик конкурентных программных решений показывает, что разрабатываемый модуль анализа и оценки эффективности технического обслуживания системы Esprit является конкурентоспособным по сравнению с представленными аналогами.

### **5.1.3. SWOT-анализ**

Для комплексного исследования проекта на основе анализа конкурентных решений была составлена матрица SWOT-анализа, содержащая сильные и слабые стороны проекта, а также возможности и угрозы для разработки проекта. Данная матрица представлена в приложении Г.

В результате проведения SWOT-анализа были выявлены сильные и слабые стороны проекта, установлены несколько направлений развития проекта, которые подкреплены его сильными сторонами. После анализа слабых сторон и угроз проекта были продуманы действия по уменьшению рисков.

## **5.2. Определение возможных альтернатив**

Для определения возможных альтернативных путей развития проекта использовался морфологический подход.

Морфологическими характеристиками проекта по разработке программного модуля являются:

- Используемая интегрированная среда разработки;
- Используемый язык программирования;
- Используемая формат передачи данных;
- Тип разрабатываемого проекта
- Платформа разрабатываемого проекта;
- Используемая библиотека визуализации графиков и диаграмм
- Способ получения данных для диаграмм

Морфологическая матрица проекта представлена ниже в таблице 12.

Таблица 12 – Морфологическая матрица проекта

	Исполнение 1	Исполнение 2	Исполнение 3
А. Интегрированная среда разработки	JetBrains Rider	Microsoft Visual Studio 2017	PHP Storm
Б. Язык программирования	C#	PHP	Java
В. Формат передачи данных	JSON	YAML	XML
Г. Тип приложения	Веб-приложение	Мобильное приложение	Облачное решение
Д. Платформа приложения	Android	ASP. Net	Laravel
Е. Библиотека для визуализации графиков	Chart.js	Google Charts	High Charts
Ж. Способ получения данных для диаграмм	ORM Entity Framework Core	ORM NHibernate	Использование хранимых процедур

Из данной морфологической матрицы проекта было выделено три варианта исполнения для решения технической задачи:

- И1. А2Б1В1Г1Д1Е1Ж1;
- И2. А1Б3В2Г2Д1Е3Ж2;
- И3. А3Б2В3Г1Д3Е2Ж3.

Эти варианты исполнения рассмотрены далее в расчетах временных показателей реализации проекта в приложении Д, в результате которых вариант исполнения И1 признан самым эффективным и был реализован на практике. Данный вариант является также наиболее гибким и позволяет легко производить доработку и сопровождение разработанного модуля. Расчет показателей был осуществлен по формулам, представленным далее в разделе 5.4.

### 5.3. Организация и планирование работ

Для успешной реализации данного проекта были определены такие показатели как занятость каждого из участников, перечень всех проводимых работ, и сроки выполнения каждого из этапов. Для представления полученных показателей также введем следующие условные обозначения исполнителей проекта:

- НР – Савельев А.О., научный руководитель работы;
- И1 – Черепанов И.С., разработчик;
- И2 – Волынцев А.И., разработчик;
- И3 – Финько А.И., старший программист;
- И4 – Мурашов П.В., тестировщик.

Выполнение проекта было разбито на несколько этапов в ходе планирования, каждому этапу соответствуют назначенные исполнители, представленные в таблице 13.

Таблица 13 – Перечень работ и исполнителей

Этапы работы	Исполнители	Загрузка исполнителей
Постановка задачи	НР	НР – 100%
Анализ предметной области	НР, И1	И1 – 100% НР – 100%
Формализация требований к приложению.	И1	И1 – 100%
Разработка календарного плана	НР, И1	И1 – 100% НР – 100%
Исследование существующих решений	И1, И2, НР	И1 – 100% И2 – 100% НР – 100%
Выбор методов и средств разработки	И3	И2 – 100%
Доработка Esprit Web для получения данных	И1, И2, И3, И4	И1 – 100% И2 – 25% И3 – 33% И4 – 10%
Проектирование архитектуры клиентской части модуля	И1, И2	И1 – 50% И2 – 100%
Проектирование архитектуры серверной части модуля	И1, И2	И1 – 50% И2 – 100%



Разработка компонентов серверной части.	И1, И2	И1 – 100% И2 – 50%
Разработка компонентов клиентской части.	И1, И2	И1 – 100% И2 – 50%
Оптимизация времени получения данных	И1	И1 – 100%
Покрытие функционала Unit-тестами.	И1	И1 – 100%
Тестирование	И4	И4 – 100%
Исправление ошибок	И1	И1 – 100%
Оценка полученных результатов	НР	НР – 100%
Подготовка отчетной документации	И1	И1 – 100%

#### 5.4. Продолжительность этапов работ

Далее необходимо рассчитать продолжительность выполнения каждого из этапов работ. Для этого будет использоваться опытно-статистический метод, который подразделяется на аналоговый и экспертный.

Так как процесс разработки программного обеспечения уникален для каждого отдельного проекта, для оценки времени реализации проекта будет применяться экспертный метод. Он предполагает генерацию необходимых количественных оценок специалистами конкретной предметной области. Экспертная оценка определяется в человеко-днях и носит вероятностный характер. Она рассчитывается с использованием длительности работ в рабочих и календарных днях каждого этапа работ.

Для расчета ожидаемого времени проведения работ  $t_{ож}$ , будут использоваться следующая формула:

$$t_{ож} = \frac{3t_{min} + 2t_{max}}{5}, \quad (5.2)$$

$t_{ож} = \frac{3t_{min} + 2t_{max}}{5}$  где  $t_{min}$  – минимальная продолжительность работы, дни.;

$t_{max}$  – максимальная продолжительность работы, дни.;

При построении рабочего графика необходимо рассчитать длительность этапов в рабочих днях, а после перевести их в календарные.

Продолжительность выполнения каждого этапа в рабочих днях –  $T_{рд}$  рассчитывается по формуле:

$$T_{РД} = \frac{t_{ож}}{K_{ВН}} \cdot K_{Д}, \quad (5.3)$$

Где  $t_{ож}$  – продолжительность работы в днях;

$K_{ВН}$  – коэффициент выполнения работ, учитывающий влияние внешних факторов на соблюдение предварительно определенных длительностей, примем  $K_{ВН} = 1$ ;

$K_{Д}$  – коэффициент, учитывающий дополнительное время на компенсацию непредвиденных задержек и согласование работ, примем  $K_{Д} = 1,2$ .

Для получения продолжительности выполнения этапа в календарных днях –  $T_{КД}$ , используется формула:

$$T_{КД} = T_{РД} \cdot T_{К}, \quad (5.4)$$

Где  $T_{К}$  – коэффициент календарности, который позволяет перейти от длительности работ в рабочих днях к их аналогам в календарных днях, и рассчитывается по формуле:

$$T_{К} = \frac{T_{КАЛ}}{T_{КАЛ} - T_{ВД} - T_{ПД}}, \quad (5.5)$$

Где  $T_{КАЛ}$  – календарные дни;

$T_{ВД}$  – выходные дни;

$T_{ПД}$  – праздничные дни;

Для шестидневной рабочей недели  $T_{К}$  равен:

$$T_{К} = \frac{365}{365 - 52 - 14} = 1,22$$

Для пятидневной рабочей недели  $T_{К}$  равен:

$$T_{К} = \frac{365}{365 - 102 - 14} = 1,46$$

В таблице 14 отражены основные этапы работ и рассчитана их трудоемкость.

Таблица 14 – Трудозатраты на выполнение диссертации

Этап	Исполнители	Продолжительность работ, дни			Трудоемкость работ по исполнителям (человеко-дни)	
		$t_{\min}$	$t_{\max}$	$t_{\text{ож}}$	Т <sub>рд</sub>	Т <sub>к</sub>
Постановка задачи	НР	1	1	1,0	1,2	1,5
Анализ предметной области	НР	2	3	2,4	2,9	3,5
	И1	2	3	2,4	2,9	4,2
Формализация требований к приложению.	И1	2	4	2,8	3,4	4,9
Разработка календарного плана	НР	1	1	1,0	1,2	1,5
	И1	1	1	1,0	1,2	1,8
Исследование существующих решений	И1	1	2	1,4	1,7	2,5
	И2	1	2	1,4	1,7	2,5
	НР	1	2	1,4	1,7	2,0
Выбор методов и средств разработки	И3	1	2	1,4	1,7	2,5
Доработка Esprit Web для получения данных	И1	12	24	16,8	20,2	29,4
	И2	3	5	3,8	4,6	6,7
	И3	4	5	4,4	5,3	7,7
	И4	1	2	1,4	1,7	2,5
Проектирование архитектуры клиентской части модуля	И1	1	1	1,0	1,2	1,8
	И2	2	3	2,4	2,9	4,2
Проектирование архитектуры серверной части модуля	И1	1	1	1,0	1,2	1,8
	И2	1	2	1,4	1,7	2,5
Разработка компонентов серверной части.	И1	8	9	8,4	10,1	14,7
	И2	4	5	4,4	5,3	7,7
Разработка компонентов клиентской части.	И1	12	20	15,2	18,2	26,6
	И2	6	8	6,8	8,2	11,9
Оптимизация времени получения данных	И1	3	4	3,4	4,1	6,0
Покрытие функционала Unit-тестами.	И1	3	4	3,4	4,1	6,0
Тестирование	И4	3	4	3,4	4,1	6,0
Исправление ошибок	И1	2	6	3,6	4,3	6,3

Оценка полученных результатов	НР	1	1	1,0	1,2	1,5
Подготовка отчетной документации	И1	4	6	4,8	5,8	8,4
<b>Итого:</b>		<b>84</b>	<b>131</b>	<b>102,8</b>	<b>123,36</b>	<b>178,1</b>

Далее было рассчитано суммарное количество затраченных календарных дней для каждого участника проекта, данные представлены в таблице 15.

Таблица 15 – Суммарные трудозатраты участников проекта

<b>Участник проекта</b>	<b>Должность/Роль</b>	<b>T<sub>к</sub>, календарные дни</b>	<b>T<sub>рд</sub>, рабочие дни</b>
Савельев А.О	Научный руководитель работы	11,9	8,2
Черепанов И.С.	Студент, разработчик	114,2	78,2
Волынцев А.И.	Разработчик	35,4	24,2
Финько А.И.	Старший программист	10,2	7,0
Мурашов П.В	Тестировщик	8,4	5,8

На основе полученных данных в таблице была построена диаграмма Ганта с графиком всех проведенных работ, а также диаграмма занятости ресурсов, они отражены в приложении Е.

### **5.5. Расчет сметы затрат на выполнение проекта**

Затраты на выполнение проекта складываются из расчета всех расходов, необходимых для реализации комплекса работ, составляющих содержание данной разработки. Их сумма необходима для формирования бюджета и расчета

сметной стоимости всей работы. С учетом специфики магистерской диссертации выделим следующие статьи затрат:

- Материалы;
- Заработная плата;
- Социальные отчисления;
- Расходы на электроэнергию;
- Амортизация.

#### **5.5.1. Расчет затрат на материалы**

Данная статья затрат включает в себя стоимость материалов, оборудования, полуфабрикатов и других материальных ценностей, расходуемых непосредственно в процессе выполнения работ. Сюда же включаются расходы на транспортировку, хранение, и прочие процессы, связанные с движением материалов, а также транзакционные издержки.

При работе над магистерской диссертацией специфическое оборудование и комплектующие не приобретались, поэтому примем затраты на материалы равными  $C_{\text{мат}} = 1300$  рублей – расходы на покупку канцелярских принадлежностей.

#### **5.5.2. Расчет заработной платы**

В данном разделе рассчитывается заработная плата участников проекта, на основе трудоемкости каждого этапа и величины месячного оклада для каждого исполнителя. Так как исполнители Черепанов И.С., Волынцев А.И., Финько А.И., Мурашов П.В. работают в ООО «ДиПи Лабс», то расчет будет производиться на основе данных о месячном окладе в этой организации.

Для руководителя курсовой работы, как для сотрудника ТПУ, месячный оклад будет взят, исходя из данных указанных на портале ТПУ, в соответствии с занимаемой должностью и ученым званием.

Затраты времени каждого исполнителя в рабочих днях взяты из таблицы 15 и округлены до целого числа.

Среднедневная тарифная заработная плата ( $ЗП_{\text{дн-т}}$ ) рассчитывается по формуле:

$$ЗП_{\text{дн-т}} = MO / 25,083, \quad (5.6)$$

Где  $MO$  – оклад исполнителя;

25,083 – среднее количество рабочих дней в месяце, при условии, что рабочих дней в году  $\approx 301$  и шестидневной рабочей неделе.

При расчёте  $ЗП_{\text{дн-т}}$  для исполнителей Черепанова И.С., Волынцева А.И., Финько А.И., Мурашова П.В. оклад исполнителя будет поделен на коэффициент 21,083, учитывая 253 рабочих дня в году и пятидневную рабочую неделю.

Также, при расчете полной заработной платы необходимо учесть в ее составе премии, районные надбавки и дополнительную зарплату. Значение полной (сметной) заработной платы исполнителя, связанной с участием в проекте, определяется по формуле

$$K_{\text{и}} = K_{\text{ГП}} \cdot K_{\text{доп.ЗП}} \cdot K_{\text{р}}, \quad (5.7)$$

Где коэффициент премирования  $K_{\text{ГП}} = 1,1$ ;

Коэффициент дополнительной заработной платы для шестидневной рабочей недели  $K_{\text{доп.ЗП}} = 1,188$ ;

Коэффициент дополнительной заработной платы для пятидневной рабочей недели  $K_{\text{доп.ЗП}} = 1,113$ ;

Районный коэффициент (для Томской области)  $K_{\text{р}} = 1,3$ .

Интегральный коэффициент рассчитанный по формуле 5.7 для шестидневной и пятидневной рабочих недель составляет:

$K_{\text{и}} = 1,1 \cdot 1,188 \cdot 1,3 = 1,699$  – для шестидневной рабочей недели.

$K_{\text{и}} = 1,1 \cdot 1,113 \cdot 1,3 = 1,62$  – для пятидневной рабочей недели.

На основании выше указанных формул и данных сумма основной заработной платы работников представлена в таблице 16.

Таблица 16 – Расчет основной заработной платы

Исполнитель	Оклад, руб./мес.	Среднедневная ставка (ЗП <sub>дн-т</sub> ), руб./раб.день	Затраты времени, раб. дни	Коэффициент (К <sub>и</sub> )	Фонд з/п, руб.
НР	47104	1877,93	8,2	1,70	26035,26
И1	43209	2049,47	78,2	1,62	259768,01
И2	56453	2677,65	24,2	1,62	105148,30
И3	88371	4191,58	7,0	1,62	47260,86
И4	37945	1799,79	5,8	1,62	16794,21
<b>Итого:</b>					<b>455006,63</b>

### 5.5.3. Расчет затрат на социальный налог

Единый социальный налог (ЕСН), включает в себя отчисления во внебюджетные фонды, а именно отчисления в пенсионный фонд, на социальное и медицинское страхование. Они являются обязательными и составляют 30% от заработной платы работников, задействованных в проекте и вычисляются по формуле:

$$C_{\text{соц.}} = C_{\text{зп}} \cdot 0,3, \quad (5.8)$$

Где  $C_{\text{зп}}$  – суммарные затраты на заработную плату.

Для данного проекта  $C_{\text{соц.}} = 455006,63 \cdot 0,3 = 136501,98$

### 5.5.4. Расчет затрат на электроэнергию

В этот подраздел включаются расходы на электроэнергию, которая была потрачена в ходе выполнения работы. Они рассчитываются по формуле:

$$C_{\text{эл.об.}} = P_{\text{об}} \cdot t_{\text{об}} \cdot Ц_{\text{Э}}, \quad (5.9)$$

Где  $P_{\text{об}}$  – мощность, потребляемая оборудованием, кВт;

$t_{\text{об}}$  – время работы оборудования, час;

$Ц_{\text{Э}}$  – тариф на 1кВт·час.

Время работы оборудования  $t_{\text{об}}$  определяется по формуле:

$$t_{\text{об}} = T_{\text{рд}} \cdot K_t, \quad (5.10)$$

Где  $T_{\text{рд}}$  – трудозатраты исполнителя при восьмичасовом рабочем дне;

$K_t \leq 1$  – коэффициент использования оборудования по времени, равный отношению времени его работы в процессе выполнения проекта к  $T_{рд}$ .

Мощность, потребляемая оборудованием  $P_{об}$ , рассчитывается как:

$$P_{об} = P_{ном.} \cdot K_C, \quad (5.11)$$

Где  $P_{ном.}$  – номинальная мощность оборудования;

$K_C \leq 1$  – коэффициент загрузки, зависящий от средней степени использования номинальной мощности.

Для расчета затрат на электроэнергию примем:  $ЦЭ = 3,50$  руб. для исполнителей,  $ЦЭ = 6,59$  руб. для ТПУ;  $K_t = 0,8$ ;  $K_C = 1$ . Время, затраченное инженером  $T_{рд}$ , было взято из таблицы 15. Для сервера вместо затраченных рабочих дней использовались календарные дни, так как он задействован все время работы над проектом, включая выходные и праздничные 24 часа в сутки. Однако сервер работает не всегда на полную мощность, в связи с этим  $K_C = 0,8$ . Ниже представлена формула для расчета затрат на обеспечения сервера электроэнергией:

$$Э_{серв} = T_k \cdot 24 \cdot K_C \cdot P_{об} \cdot ЦЭ \quad (5.12)$$

Номинальная мощность  $P_{об}$  указана в технических характеристиках производителем оборудования. При работе использовались пять ЭВМ и один сервер, полный расчет расходов на электроэнергию представлен в таблице 17.

Таблица 17 – Затраты на электроэнергию

Наименование оборудования	Время работы оборудования $t_{об}$ , час	Потребляемая мощность $P_{об}$ , кВт	Тариф на 1кВт·час ЦЭ, руб	Затраты $Э_{об}$ , руб.
ЭВМ ИР	$8,2 * 0,8 * 8$	0,5	6,59	172,02
ЭВМ И1	$78,2 * 0,8 * 8$		3,5	876,29
ЭВМ И2	$24,2 * 0,8 * 8$			271,49
ЭВМ И3	$7,0 * 0,8 * 8$			77,95
ЭВМ И4	$5,8 * 0,8 * 8$			64,51
Сервер	$178,1 * 24 * 0,8$	0,65		7779,41
<b>Итого:</b>				<b>9241,67</b>



### 5.5.5. Расчет амортизационных расходов

Амортизационные затраты для используемого в проекте оборудования рассчитываются по формуле:

$$C_{AM} = \frac{N_A * Ц_{ОБ} * t_{рф} * n}{F_D}, \quad (5.13)$$

Где  $N_A$  – годовая норма амортизации единицы оборудования;

$Ц_{ОБ}$  – балансовая стоимость единицы оборудования, руб;

$F_D$  – действительный годовой фонд времени работы соответствующего оборудования, час;

$t_{рф}$  – фактическое время работы оборудования в ходе выполнения проекта, час;

$n$  – число задействованных однотипных единиц оборудования.

Годовая норма амортизации  $N_A$  рассчитывается как величина обратная сроку его амортизации. Для сервера и компьютера срок амортизации может составлять от двух до трех лет, примем за норму амортизации 2,5 года, тогда  $N_A = 1/2,5 = 0,4$ . Балансовая стоимость ПК, задействованных при работе над проектом, составляет 53470 рублей за одну единицу, стоимость сервера – 117000 рублей, стоимость компьютера научного руководителя примем за 50000 рублей.

Действительный годовой фонд времени работы оборудования –  $F_D$  рассчитывается из учета рабочих дней и величины недели:

$$F_D = 253 * 8 = 2024 \text{ рабочих часов при пятидневной рабочей неделе}$$

$$F_D = 301 * 8 = 2408 \text{ рабочих часов при шестидневной рабочей неделе}$$

$$F_D = 365 * 24 = 8760 \text{ часов для сервера}$$

Фактическое время работы оборудования в ходе выполнения проекта для всех исполнителей рассчитано с использованием данных таблицы 17:

$$t_{рф} = (78,2 + 24,2 + 7 + 5,8) * 0,8 * 8 = 737,28 \text{ – для ЭВМ исполнителей И1-И4;}$$

$$t_{рф} = 8,2 * 0,8 * 8 = 52,48 \text{ – для ЭВМ научного руководителя;}$$

$$t_{рф} = 178,1 * 24 * 0,8 = 3419,52 \text{ – для сервера.}$$

Таким образом, сумма амортизации, начисленной на ЭВМ исполнителей равна:

$$C_{AM}(И) = \frac{0,4 * 53470 \text{ руб} * 737,28 \text{ часов} * 4 \text{ ед}}{2024 \text{ часов}} = 31163,9 \text{ руб.}$$

Амортизация для компьютера научного руководителя и сервера:

$$C_{AM}(НР) = \frac{0,4 * 50000 \text{ руб} * 52,48 \text{ часов} * 1 \text{ ед}}{2408 \text{ часов}} = 435,8 \text{ руб.}$$

$$C_{AM}(С) = \frac{0,4 * 117000 \text{ руб} * 3419,52 \text{ часов} * 1 \text{ ед}}{8760 \text{ часов}} = 18268,67 \text{ руб.}$$

Общие затраты на амортизацию составляют:

$$C_{AM} = 31163,9 + 435,8 + 18268,67 = 49868,37 \text{ рублей.}$$

### 5.5.6. Расчет прочих расходов

В прочие расходы включаются те затраты, которые ранее не были или не могли быть учтены при расчетах. Они рассчитываются как:

$$C_{проч.} = (C_{мат} + C_{зп} + C_{соц} + C_{эл} + C_{ам}) * 0,1$$

Примем прочие расходы равными 10% от всей суммы расходов проекта:

$$C_{проч.} = (1300 + 455006,63 + 136501,98 + 9241,67 + 49868,37) * 0,1 = 65191,87$$

### 5.5.7. Расчет общей себестоимости разработки

Общая стоимость разработки программного модуля вычисляется путем сложения затрат на каждую из статей расходов. Общая себестоимость проекта представлена в таблице 18.

Таблица 18 – Смета затрат на разработку

Статья затрат	Сумма, руб.
Материалы	1300
Основная заработная плата	455006,63
Отчисления в социальные фонды	136501,98
Расходы на электроэнергию	9241,67
Амортизационные отчисления	49868,37
Прочие расходы	65191,87
<b>Итого:</b>	<b>717110,52</b>

## 5.6. Экономическая эффективность проекта

Данная разработка не ориентирована на внешний рынок, она является продуктом ООО «ДиПи Лабс» и предназначена для внутреннего использования. Компания «ДиПи Лабс» является подрядчиком организации «DP Technology» и разрабатывает программные решения для нее. В связи с подписанным соглашением о неразглашении стоимостная оценка результата данной работы не является доступной.

В разделе с конкурентным анализом приведено обоснование проведения собственной разработки. В SWOT-анализе были рассмотрены сильные стороны проекта и их влияние на возможности предприятия.

Далее перечислены все основные аспекты получаемого прямого экономического эффекта от внедрения разработки:

- Снижение трудозатрат на получение и оформление информации о КПИ работы службы технической поддержки;
- Автоматизация некоторых бизнес-процессов по предоставлению технической поддержки;
- Повышение прозрачности бизнес-процессов оказания технической поддержки.

Косвенный экономический эффект заключается в повышении престижа организации и лояльности пользователей за счет улучшения качества и эффективности обработки запросов в техническую поддержку.

## 5.7. Выводы по разделу

В ходе выполнения магистерской диссертации был разработан модуль анализа и оценки эффективности технического обслуживания системы Esprit, который позволяет получать необходимые данные автоматически, а также фильтровать и представлять их в наглядном виде.

При проведении анализа конкурентных программных продуктов удалось установить, что требуется применение собственной разработки. Существующие на рынке решения не могут быть использованы для обеспечения достаточной степени автоматизации, что обосновывает необходимость разработки модуля.

Проект мог быть реализован несколькими способами с использованием разных стеков технологий и средств разработки. Для выбора самого оптимального варианта был произведен анализ временных показателей для нескольких вариантов исполнения проекта.

В разделе было проведено планирование этапов реализации проекта, рассчитана трудоемкость всех работ и занятость исполнителей при выполнении каждой из них. На основе этих данных была построена диаграмма Ганта для наглядного отображения графика планируемых работ.

## **6. Социальная ответственность**

В рамках данной магистерской диссертации требуется разработать модуль анализа и оценки эффективности для обеспечения информационной поддержки процессов управления техническим обслуживанием системы ESPRIT.

Проектирование, разработка и тестирование модуля проводились в офисе ООО «ДиПи Лабс» с использованием электронной вычислительной машины (ЭВМ). Известно, что при работе с использованием средств компьютерной техники существует ряд неблагоприятных факторов, влияющих на состояние человека, таких как: снижение производительности труда, проблемы со зрением, нервное и умственное истощение.

Таким образом, главной целью данного раздела является рассмотрение ряда следующих пунктов:

- Вопросы охраны труда и здоровья работников;
- Вопросы безопасности окружающей среды;
- Выявление потенциально вредных и опасных факторов при работе с ЭВМ;
- Предотвращение чрезвычайных ситуаций (ЧС) и действия при их возникновении и устранении.

### **6.1. Правовые и организационные вопросы обеспечения безопасности**

#### **6.1.1. Правовые нормы трудового законодательства**

В Российской Федерации трудовые отношения между работником, работодателем и государством регулируются Трудовым кодексом РФ от 30.12.2001 № 197-ФЗ [12]. Данный нормативно-правовой акт регламентирует права и обязанности между сторонами трудовых взаимоотношений, нормы рабочего времени, порядок оплаты труда и компенсаций и т.д. Выделим несколько главных положений Трудового кодекса:

- Продолжительность рабочего времени не может превышать 40 часов в неделю;
- В течение рабочего дня работнику должен быть предоставлен перерыв на обед или отдых, согласно правилам внутреннего распорядка организации;
- Работодатель обязан обеспечить безопасные условия труда и обязательное социальное страхование;
- Запрещена дискриминация по любым признакам и принудительный труд;
- Данные о работнике, предоставленные работодателю, обрабатываются только с согласия работника.

Разработка проводилась в соответствии со всеми требованиями Трудового кодекса.

### **6.1.2. Организационные вопросы обеспечения безопасности**

Так как специфика процесса разработки неразрывно связана с работой за ЭВМ, параметры и организация рабочей зоны имеют существенное значение. Основными нормативно-методическими документами, регулирующими требования к конструкции рабочего места, являются:

- ГОСТ 12.2.032-78 ССБТ «Рабочее место при выполнении работ сидя. Общие эргономические требования» [2];
- ГОСТ Р 50923-96 «Дисплей. Рабочее место оператора. Общие эргономические требования и требования к производственной среде. Методы измерения» [3].

При работе конкретно с ЭВМ необходимо учитывать санитарно-эпидемиологические правила и нормативы СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы» [1]. Кроме требований и факторов, влияющих

конкретно на работу с компьютером, они также включают правила организации рабочего места.

Таким образом, в соответствии с выше перечисленной нормативной документацией, укажем основные требования для эффективной и безопасной работы за ЭВМ:

- Конструкция рабочего места и расположение его элементов должны соответствовать эргономическим, физиологическим и антропометрическим требованиям;
- Расстояние от глаз работника до монитора должно составлять 400 – 700 мм, а конструкция ЭВМ должна обеспечивать возможность поворота и фиксации корпуса монитора в обеих плоскостях;
- Освещенность на поверхности стола и экрана должна быть не менее 300 лк, не допускается наличие бликов на поверхности монитора;
- Расстояние между рабочими местами не может быть менее 2 метров, расстояние между боковыми поверхностями мониторов – не менее 1,5 метров, расстояние рабочих зон от стен – не менее 1 м;
- Рабочий стул (кресло) должны позволять менять положение тела работника, в целях предупреждения усталости и утомления мышц шеи и спины;
- Ширина и высота рабочего стула должны составлять не менее 320 и 400 мм, высота рабочей поверхности – не менее 630 мм, подробнее с допустимыми значениями для различных параметров можно ознакомиться в СанПиН 2.2.2/2.4.1340-03 [1];
- Помещения с электрической и компьютерной техникой (ЭВМ) должны быть обеспечены защитным заземлением и соответствовать всем техническим требованиям;
- По рекомендациям СанПиНа пользователям, постоянно работающим за ЭВМ, необходимо проводить профилактические перерывы длительностью 10-15 минут каждый час.

Планировка и размещение рабочих мест в ООО «ДиПи Лабс», где непосредственно проводилась работа по проектированию и разработке проекта, полностью соответствует выше указанным требованиям.

## 6.2. Производственная безопасность

В этом подразделе рассматриваются вредные и опасные факторы, которые могут возникнуть во время эксплуатации различного оборудования.

Офисные помещения ООО «ДиПи Лабс» оборудованы ЭВМ, лампами, кондиционерами и не относятся к помещениям с повышенной или особо повышенной опасностью.

Согласно ГОСТ 12.0.003-74 и ГОСТ 12.0.003-2015 [4] все производственные факторы условно можно разделить на четыре вида, по группам их элементов: химические, биологические, физические и психофизические. Первые два вида рассматривать нецелесообразно, так как они не характерны для процесса разработки, а прочие будут подробно разобраны в следующих пунктах.

Также необходимо разделить процесс разработки модуля на этапы работ: проектирование, разработка, тестирование (таблица 19).

Таблица 19 – Вредные и опасные производственные факторы при выполнении работ за ЭВМ

Факторы (ГОСТ 12.0.003-2015)	Этапы работ			Нормативные документы
	Проектирование	Разработка	Тестирование	
1. Отклонение показателей микроклимата	+	+	+	1. СанПиН 2.2.4.548-96. Гигиенические требования к микроклимату производственных помещений [5].



2. Недостаточная освещенность рабочей зоны	+	+	+	2. СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95* [6].
3. Превышение уровня шума	+	+	+	3. ГОСТ 12.1.003-2014 ССБТ. Шум. Общие требования безопасности [7].
4. Повышенный уровень электромагнитных излучений	+	+	+	4. ГОСТ 12.1.006-84 ССБТ. Электромагнитные поля радиочастот. Общие требования безопасности [9].
5. Нервно-психические перегрузки	+	+	+	5. СанПин 2.2.4.3359-16. Санитарно-Эпидемиологические требования к физическим факторам на рабочих местах [13].

Так как модуль анализа и оценки эффективности для обеспечения информационной поддержки процессов управления техническим обслуживанием системы ESPRIT не является материальным и осязаемым объектом, а значит неотделим от ЭВМ, следует рассмотреть указанные факторы в отношении непосредственно рабочего места и помещения где проводится разработка.

### **6.2.1. Показатели микроклимата**

Микроклимат помещения можно определить, как состояние его внутренней среды, оказывающей воздействие на человека. К его характеристикам относятся: температура воздуха; относительная влажность воздуха; скорость движения воздуха.

Все показатели должны обеспечивать оптимальный тепловой баланс человека со средой и поддерживать комфортное состояние организма. При повышении или занижении хотя бы одно из показателей от нормы, человек начинает чувствовать дискомфорт и тратить силы на собственную терморегуляцию. Это может повлечь за собой напряжение, упадок производительности труда, а также увеличить вероятность заболевания различными инфекционными и простудными заболеваниями.

Санитарные правила, установленные СанПиН 2.2.4.548-96 [5] содержат гигиенические требования к микроклимату помещений и каждому из названных показателей в зависимости от таких факторов как: период года, интенсивность энергозатрат работающих, сроки выполнения работы. СанПиН содержит показания к измерению и контролю микроклиматических условий.

Работу программиста можно отнести к энергетически наименее затратной (Ia), так как физические нагрузки на рабочем месте почти отсутствуют, и работа производится в положении сидя. Необходимые нормы показателей микроклимата в помещении представлены в таблице 20.

Таблица 20 – Оптимальные величины показателей микроклимата на рабочих местах производственных помещений по СанПиН 2.2.2.548-96 [5]

Период года	Категория работ	Температура воздуха	Температура поверхностей	Относительная влажность воздуха	Скорость движения воздуха
Холодный	Ia	22-24	40-60	0,1	< 0,1
Теплый	Ia	23-25	40-60	0,1	< 0,1

При работе над магистерской диссертацией в помещениях ООО «ДиПи Лабс» дискомфорта не ощущалось, температура воздуха составляла 24 градуса Цельсия, что соответствует оптимальным показателям температуры. В рабочих зонах предусмотрены кондиционеры и центральное отопление, что позволяет подбирать необходимый температурный режим.

### **6.2.2. Освещенность рабочей зоны**

Хорошо организованное освещение в рабочих помещениях является одним из важных факторов в процессе разработки. Согласно классификации зрительных работ – работа программиста относится к 3 разряду, то есть к категории работ высокой точности.

Недостаточное или неправильно подобранное освещение, особенно при работе с ЭВМ, может повлечь за собой такие заболевания как близорукость или синдром туннельного зрения. Длительная работа в условиях плохой освещенности вызывает усталость и повышенное утомление организма. Нормы

освещенности регламентируются СНиП 23-05-95 [6] и представлены в таблице 21.

Таблица 21 – Требования к освещению помещений промышленных предприятий для операторов ЭВМ

Характеристика зрительной работы	Наименьший или эквивалентный размер объекта различения, мм	Разряд зрительной работы	Подразряд зрительной работы	Контраст объекта с фоном	Характеристика фона	Искусственное освещение		
						Освещенность, лк		
						При системе комбинированного освещения		При системе общего освещения
						Всего	В том числе от общего	
Высокой точности	От 0,3 до 0,5	III	г	Средний и большой	Светлый и средний	400	200	200

Также, согласно санитарно-эпидемиологическим правилам СанПиН 2.2.2/2.4.1340-0, для рассеивания естественного освещения необходимо использовать жалюзи, а при его недостатке – дополнительные источники света (люминесцентные лампы и лампы накаливания). Уровень освещения на поверхности рабочего стола при работе с ЭВМ должен быть в диапазоне от 300 до 500 лк [1].

Далее приведен расчет показателей искусственного освещения для рабочего кабинета, который является прямоугольным помещением следующего размера:

- Длина  $A = 5$  м;
- Ширина  $B = 4$  м;
- Высота  $H = 3$  м.

В помещении используются светильники типа ОДР (светильник общего освещения диффузный с экранирующей решеткой) с люминесцентными лампы типа ЛБ (белый свет) мощностью 30 Вт и со световым потоком  $\Phi = 2020$  лм. Общее число ламп в офисе равно  $n = 14$ . Коэффициент пульсации ламп данного типа не превышает 5%, что соответствует нормам.

Освещенность помещения рассчитывается по формуле [15]:

$$E_{\phi} = \frac{n \cdot \eta \cdot \Phi}{S \cdot k \cdot z}, \quad (6.1)$$

где  $n$  – число светильников;

$\eta$  – коэффициент использования светового потока;

$\Phi$  – световой поток светильника, лм;

$S$  – площадь помещения, м<sup>2</sup>;

$k$  – коэффициент неравномерности освещения;

$z$  – это коэффициент неравномерности освещения.

Коэффициент запаса  $k$  учитывает запыленность светильников и их износ.

Для помещений с вычислительной техникой  $k = 1,4$ . Поправочный коэффициент для люминесцентных ламп равен  $z = 1,1$ . Площадь помещения равна:

$$S = A \cdot B = 5 \cdot 4 = 20 \text{ м}^2.$$

Коэффициент использования светового потока определяется при помощи таблицы на основе индекса помещения и коэффициенты отражения от стен, потолка и рабочей поверхности. Поэтому сначала найдем данные показатели.

Индекс помещения определяется по формуле [15]:

$$i = \frac{S}{h \cdot (A+B)}, \quad (6.2)$$

где  $S$  – площадь помещения, м<sup>2</sup>;

$A$  – длина комнаты, м;

$B$  – ширина комнаты, м;

$h$  – высота подвеса светильников, м.

При этом расчетная высота подвеса светильников над рабочей поверхностью ( $h$ ) в офисе компании определяется по формуле [15]:

$$h = H - h_p - h_c,$$

где  $H$  – высота потолка в помещении, м;

$h_p$  – расстояние от пола до рабочей поверхности стола, м;

$h_c$  – расстояние от потолка до светильника, м.

Тогда расчетная высота подвеса светильников равна:

$$h = 3 - 0,8 - 0,03 = 2,17 \text{ м}.$$

Используем полученные значения для расчета индекса помещения по формуле 6.2.

$$i = \frac{20}{2,17 \cdot (5 + 4)} = 1,024 .$$

Для офиса в котором проводилась работа коэффициенты отражения от стен  $\rho_s = 70\%$ , потолка  $\rho_p = 50\%$  и от рабочей поверхности  $\rho_r = 10\%$ .

По таблице коэффициентов использования светового потока для соответствующих значений  $i$ ,  $\rho_s$ ,  $\rho_p$ , определяем интерполяцией коэффициент использования светового потока ( $\eta$  из формулы 6.1). Получается для светильника ОДР при  $i = 1$  коэффициент  $\eta = 49\%$ , а при  $i = 1,1$   $\eta = 51\%$ . Тогда для  $i = 1,024$  коэффициент  $\eta$  равен:

$$\eta = 49 + ((51 - 49) / (1.1 - 1)) \cdot (1,024 - 1) = 49.48\%.$$

Учитывая все параметры, рассмотренные выше, найдем освещенность по формуле 6.1:

$$E_{\phi} = \frac{14 \cdot 0,4948 \cdot 2200}{20 \cdot 1,4 \cdot 1,1} = 454,4 \text{ лк.}$$

В рассматриваемом помещении освещенность должна составлять не менее 300 лк согласно СанПиН 2.2.2/2.4.1340-03[1] и СНиП 23-05-95[6]. В офисе освещенность равна 454,4 лк и находится в пределах нормы, следовательно, дополнительные источники света не нужны.

### **6.2.3. Уровень шума в помещениях**

Превышение уровня шума в помещениях, где производится работа является вредным фактором. Его источником может служить различная офисная техника: компьютеры, кондиционеры, вентиляторы, копировальная техника и различное оборудование.

Данный фактор, при превышении своих нормативных значений негативно воздействует не только на органы слуха человека, но и на его нервную систему, что оказывает влияние на самочувствие работника и его работоспособность.

СанПиН 2.2.2/2.4.1340-03 [1] и ГОСТ 12.1.003-2014 ССБТ [7] регулирует допустимые значения уровней звукового давления в октавных полосах частот и уровня звука, создаваемого ЭВМ. Данные представлены в таблице 22.

Таблица 22 – Допустимые значения уровней звукового давления в октавных полосах частот и уровня звука, создаваемого ЭВМ

Уровни звукового давления в октавных полосах со среднегеометрическими частотами									Уровни звука в дБА
31,5 Гц	63 Гц	125 Гц	250 Гц	500 Гц	1000 Гц	2000 Гц	4000 Гц	8000 Гц	
86 дБ	71 дБ	61 дБ	54 дБ	49 дБ	45 дБ	42 дБ	40 дБ	38 дБ	50

Главными мероприятиями по защите от шума можно назвать: правильную звукоизоляцию; глушители шума; правильное распределение рабочих мест; создание шумозащищенных зон для отдыха и уменьшение шума в его источнике.

При выполнении работ на ЭВМ уровень шума на рабочем месте не должен превышать 50 дБ. В процессе работы в помещении одновременно работало до 4 ЭВМ, максимальный уровень шума в рабочей зоне составлял 30 дБ, что соответствует нормам, приведенным в таблице выше.

#### **6.2.4. Уровень электромагнитных излучений**

Еще одним неблагоприятным фактором при работе с ЭВМ является то, что он, как и любой другой электрический прибор, излучает вокруг себя электромагнитные волны. При повышении уровня подобного излучения, организм человека испытывает нагрузку на уровне нервной и сердечно-сосудистой системы, что ведет за собой увеличение усталости, понижение (повышение) артериального давления, ухудшение общего психофизического состояния человека.

При работе с ЭВМ электромагнитное излучение исходит прежде всего от монитора. Согласно ГОСТ 12.1.006-84 ССБТ [9] и требованиям СанПиН 2.2.2/2.4.1340-03 [1] работа сотрудника возможна при уровне напряженности

ЭМП не превышающем 5 кВ/м и восьмичасовом рабочем дне. Подробнее допустимые нормы и значения указаны в таблице 23.

Таблица 23 - Временные допустимые уровни (ВДУ) ЭМП, создаваемых ЭВМ

Наименование параметров		ВДУ ЭМП	ВДУ ЭМП
Напряженность электрического поля	в диапазоне частот 5 Гц – 2 кГц	25 В/м	27 В/м
	в диапазоне частот 2 кГц – 400 кГц	2,5 В/м	2,5 В/м
Электростатический потенциал экрана видеомонитора		500 В	490 В

Для уменьшения уровня воздействия ЭМП и защите человека от вредного воздействия могут использоваться экранные фильтры и костюмы индивидуальной защиты. Профилактической мерой являются систематические медицинские осмотры и дозиметрический контроль, а также уменьшение контакта с источником излучения (не менее 50 см от сотрудника).

В процессе разработки подобные меры не применялись, так как показатели электромагнитного излучения находились в норме.

#### **6.2.5. Нервно-психические перегрузки**

При постоянной работе за ЭВМ, сотрудник испытывает еще один фактор вредного воздействия – нервно-психическое перенапряжение. При работе с большими объёмами данных существует опасность умственного перенапряжения и в следствие, невозможность дальнейшей продуктивной работы.

При умственной работе потребность мозга в кислороде увеличивается в 15-20 раз и возрастает опасность повышенного артериального давления. Постоянное пребывание в подобных условиях может привести к сердечно-сосудистым заболеваниям.

Согласно ГОИ Р-45-084-01. «Типовая инструкция по охране труда при работе на персональном компьютере» (утв. Приказом Минсвязи РФ от 02.07.2001 N 162) устанавливается 3 категории тяжести и напряженности работы с компьютером, которые определяются: для группы А - по суммарному числу

считываемых знаков за рабочую смену (не более 60000 знаков за смену); для группы Б - по суммарному числу считываемых или вводимых знаков за рабочую смену (не более 40000 знаков за смену); для группы В - по суммарному времени непосредственной работы с компьютером за рабочую смену (не более 6 часов за смену). Это отражено в таблице 24.

Таблица 24 - Уровень нагрузки за рабочую смену при видах работ с компьютером

Категория работ	Уровень нагрузки за рабочую смену при видах работ с компьютером		
	группа А, количество знаков	группа Б, количество знаков	группа В, час.
Ш	До 60000	До 40000	До 6,0

Из методов для предупреждения развития нервного и психического напряжения при работе за ЭВМ можно выделить: наличие частых перерывов, с возможностью небольших физических нагрузок; соответствие рабочего места нормативам, указанным в п. 6.2.1 – 6.2.4; калорийные перекусы, для восстановления энергии.

Каждый сотрудник в ООО «ДиПи Лабс» обеспечен удобным и технологически безопасным рабочим местом, в помещении организации на ряду с рабочими кабинетами располагаются кухня, зона отдыха, площадка с тренажерами.

#### **6.2.6. Поражение электрическим током**

Так как ЭВМ относится к разряду технически сложных электрических приборов, при работе с ним может возникнуть опасность удара электрическим током. Степень губительного воздействия на организм зависит от величины напряжения, силы и вида тока, пути протекания через тело и продолжительности разряда – но в любом случае человек может получить различные виды ожогов и электротравм.



Существуют нормы допустимых значений напряжения и тока, определенные ГОСТ 12.1.038-82 ССБТ [10], они представлены в таблице 25.

Таблица 25 – Допустимые значения напряжения прикосновения и тока

Род тока	Напряжения прикосновения, В	Ток, мА
	Не более	
Переменный, 50 Гц	2,0	0,3
Постоянный	8,0	1,0

Также, при работе с ЭВМ во избежание удара электрическим током необходимо:

- Отсутствие оголенных или неисправных частей проводов и кабелей;
- Наличие заземления всех металлических частей ЭВМ;
- Соблюдение техники безопасности при работе с компьютером (исключается возможность прикосновения к токоведущим приборам, наличие воды в открытой таре);
- Проведение профилактических проверок и осмотра оборудования.

Во время написания магистерской диссертации ситуаций, связанных с поражением электрическим током не возникало, все необходимые показатели находились в допустимых пределах.

#### **6.2.7. Мероприятия по снижению опасного и вредного воздействия**

На основании рассмотренных ранее подразделов выделим те профилактические мероприятия, которые необходимы для снижения опасного и вредного воздействия во время работы с ЭВМ:

Мероприятия для поддержания комфортного микроклимата:

- Оснащенность офисных помещений системами регулировки температуры воздуха;
- Наличие центрального отопления в зимний период;
- Правильно выстроенная система вентиляции и проветривания.

Мероприятия для достижения необходимого уровня освещенности:

- Правильно подобранный тип освещения в зависимости от времени года и суток;
- Достаточное количество и возможность регуляции источников искусственного освещения;
- Наличие штор типа «жалюзи», для эффективного рассеивания естественного света.

Мероприятия по минимизации шумовых воздействий:

- Наличие систем шумоизоляции и шумоподавления;
- Грамотное расположение рабочих зон и источников шума относительно друг друга.

Мероприятия по защите от воздействия электромагнитного излучения:

- Контроль за уровнем электромагнитного излучения;
- Увеличение расстояния от источника ЭВМ.

Мероприятия по профилактике нервно-психического напряжения:

- Проведение частых перерывов, с возможностью физических нагрузок;
- Наличие зон отдыха и принятия пищи в организации.

Мероприятия по защите от удара электрическим током:

- Соблюдение техники безопасности при работе с электрическими приборами;
- Подключение и регулярные проверки оборудования, в соответствии со всеми техническими требованиями.

### **6.3. Экологическая безопасность**

Объект магистерской работы является нематериальным, и, следовательно, может рассматриваться в связке только с ЭВМ. Их производство включает в себя токсичное сырье, которое подлежит специальной утилизации и переработке – без них материалы способны постепенно разрушаясь наносить непоправимый вред экологии и здоровью человека. Многие предметы офисной

техники, после завершения срока своей эксплуатации, становятся опасными отходами, которые могут оказать вред атмосфере, гидросфере и литосфере. Например, ЖК-экраны являются большим источником парниковых газов, а люминесцентные лампы содержат в себе от 10 до 70 мг ртути.

Согласно Кодексу Российской Федерации об административных правонарушениях [14], отработанную технику (в том числе ЭВМ) запрещается выбрасывать наряду с обыкновенным мусором, а необходимо обратиться в специальные службы для ее утилизации или переработки. ГОСТ 12.3.031-83 «Работы со ртутью. Требования безопасности» требует, чтобы все отходы и приборы, содержащие ртуть, подлежали сбору и возврату только сертифицированным лицом (электромонтером).

Также для обеспечения экологической безопасности и безвредной утилизации отходов в офисе используется практика по селективному сбору мусора.

В качестве профилактики и предотвращения опасного воздействия электрической техники на окружающую среду, санитарные нормы СанПиН 2.2.2/2.4.1340-03 [1] рекомендуют использовать ее в экономичном режиме работы, а также обращать внимание на соответствие используемых материалов в ЭВМ нормам и стандартам экологической безопасности.

#### **6.4. Безопасность в чрезвычайных ситуациях**

Можно выделить несколько типов чрезвычайных ситуаций, по характеру их возникновения: техногенные, природные, социальные, экономические, экологические. Во время работы в офисных и производственных помещениях могут возникнуть ЧС следующих типов: (не)преднамеренные; техногенные (пожары, взрывы); социальные (террористические акты); стихийные (природные) и комбинированные.

В качестве анализа мер безопасности при чрезвычайной ситуации была выбрана наиболее распространенная ЧС, а именно пожар. Во время работы организации источником опасности могут стать неправильно подключенные

электрические провода и кабели, неисправно работающая оргтехника, наличие легко воспламеняющихся материалов (дерево, бумага).

Выделим несколько основных факторов, по причине которых, возможно возникновение очага возгорания:

- Возникновение короткого замыкания;
- Разряд статического электричества;
- Неисправное электрооборудование;
- Несоблюдение мер пожарной безопасности.

Согласно ГОСТ Р 22.3.03-94 [11], обеспечение безопасности людей при возникновении чрезвычайной ситуации является обязательной задачей для всех государственных систем и подразделений. Регулирование пожаробезопасности производится СНиП 21-01-97. Для профилактики возникновения ЧС должен проводиться следующий комплекс мер по предотвращению возгорания:

- Не допускается блокирование или загромождение пожарных выходов;
- Необходимо проводить регулярные проверки первичных средств для тушения пожаров и систем оповещения;
- Во всех служебных помещениях должны быть установлены «Планы эвакуации людей при пожаре и других ЧС»;
- Должны проводиться инструктажи по пожарной безопасности и тренировки действий в случае возникновения пожаров;
- Необходимо правильное содержание и эксплуатация электрических приборов;
- Должны соблюдаться установленные в организации правила противопожарной безопасности;
- Помещения с ЭВМ должны быть оборудованы первичными средствами пожаротушения (разрешается использование углекислотных и порошковых огнетушителей);

- Недопустимо использовать для тушения пожара пенные огнетушители или воду, так как они проводят электрический ток;
- Помещения с ЭВМ должны проектироваться согласно I или II степени огнестойкости.

В случае если источник возгорания не может быть ликвидирован самостоятельно необходимо вызвать службы для пожаротушения и организовать эвакуацию персонала в минимально возможные сроки.

### **6.5. Выводы по разделу**

Так как выполнение магистерской диссертации производилось за ЭВМ, в данном разделе были рассмотрены основные вопросы обеспечения технологической и экологической безопасности при работе с ЭВМ; проанализированы различные вредные и опасные производственные факторы; и предложены мероприятия по их снижению.

После анализа условий работы в офисе ООО «ДиПи Лабс», можно сделать вывод о том, что разработка программного модуля в рамках диссертации проводилась в полном соответствии с существующими нормами и показаниями к безопасности труда. Во избежание негативного влияния на здоровье во время работы с ЭВМ необходимо делать перерывы и проводить специализированные комплексы физических упражнений.

Для обеспечения экологической безопасности в офисе ООО «ДиПи Лабс» используется практика по селективному сбору мусора. Отходы в виде бумаги и картона, алюминия, пластика и гальванических элементов сортируются и утилизируются без вреда для окружающей среды.

Показатели микроклимата, освещенности электро- и пожаробезопасности в офисных помещениях соответствуют требованиям, предъявленным в соответствующих нормативных документах.

## **Заключение**

В ходе работы проведен анализ предметной области, актуальных технологий и программных решений для наглядного отображения графиков и диаграмм. Были изучены и применены на практике современные технологии разработки веб-приложений.

Было доработано приложение Esprit Web для предоставления необходимых данных в разработанный модуль анализа и оценки эффективности технического обслуживания системы Esprit. Для получения оценок качества работы от иностранных пользователей была произведена локализация модуля технической поддержки и рассылаемых Email-уведомлений.

В результате разработки модуля анализа и оценки эффективности технического обслуживания системы Esprit компания «DP Technology» получила инструмент для просмотра статистики, который позволяет миновать технически сложные операции по предоставлению и обработке исходных данных. В модуле были представлены наглядные диаграммы и удобная система для фильтрации данных.

Выявленные требования к разработанному проекту были учтены и выполнены. Диаграммы загружаются быстро и асинхронно, корректно отображаются на мобильных устройствах. Разработанный модуль внедрен и активно используется.

Разработанный модуль в будущем планируется доработать для отображения новых метрик по различным бизнес-процессам.

## Приложение А

(справочное)

### Redesigning of Esprit Web application in order to obtain the necessary data

Студент:

Группа	ФИО	Подпись	Дата
8ИМ81	Черепанов Иван Сергеевич		

Руководитель ВКР:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент ОИТ	Савельев Алексей Олегович	К.Т.Н.		

Консультант-лингвист отделения иностранных языков ШБИП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Маркова Наталия Александровна			

### 3. Redesigning of Esprit Web application to obtain the necessary data

#### 3.1. Getting customer reviews of support services

The development team successfully implemented all the tasks of redesigning the Esprit Web application, which were highlighted in Table 4. We created a specific page that contains a form for sending data to the server to provide Esprit customers with the opportunity to participate in user surveys (Figure 12). After filling and sending the form, survey results are saved into the database.

The screenshot shows a web application interface for a survey. On the left is a dark sidebar with the ESPRIT logo and navigation menu items: Search, Reseller Administrators, Survey Settings, New Support Request, Search Companies, My Dashboard, My Team Dashboards, Dashboard Admin, Dashboard ViewAs Admin, Send To Support, and ReAssign Request. The main content area has a breadcrumb trail: Support Web: Request Search / Request View / Survey. It starts with a greeting to Ivan Cherepanov and a message about a survey invitation for request #238682. There are three rating questions, each with a 1-10 scale and 'Very Dissatisfied' to 'Very Satisfied' labels. The questions are: 'Quality of Technical Response', 'Timely Resolution', and 'How likely is it that you would recommend ESPRIT to a friend or colleague?'. Below these is a 'Comments:' text area with a 500-character limit and a 'Send' button. A '\* - required' note is at the bottom left.

Figure 54. The created web page with the survey in Esprit Web application

Special links were built into some email notification templates to provide access to the survey page (Figures 13-14).



```

<!-- Survey link -->
@if (Model.RequestDetails.IsSurveyAvailable)
{
    var link = $"{Model.PublicWebhost}/Support/RequestSurvey/{Model.RequestDetails.RequestId}";
    <tr>
        <td>
            @Raw(Model.Localize(nameof(SwRequestUpdatedConstants.AdviceToTakeSurvey), link))
        </td>
    </tr>
}

```

Figure 55. The template markup code with a built-in link to the survey page

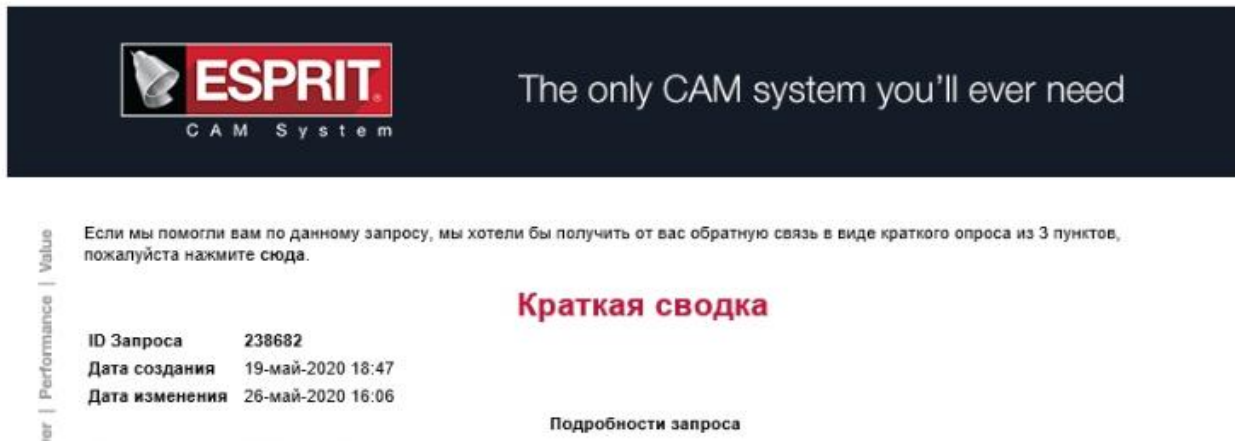


Figure 56. Fragment of a localized letter with the link to the survey

After that, management appointed an expansive project that consists of the localization of the technical support module and email templates. The goals were to get survey ratings from foreign users and to improve the quality of their interaction with Esprit Web. We have broken down the task of localizing email notifications into the following sub-tasks:

1. Choosing localization technologies.
2. Redesign website module architecture to take into account localization functionality.
3. Establishment a list of supported languages.
4. Identifying localized string literals.
5. Providing localized lines with context and explanation in local departments for translation.
6. Determination of the business logic of choosing a language for different types of notification templates.

7. Implementation of the functionality on the localization of notifications.
8. Editing existing Unit-tests.
9. Performance testing.
10. Conducting Code Review, adding fixes if it is needed.
11. Holding manual testing.
12. Generating all templates of localized emails for proofreading.
13. Making edits to translations.
14. Developing Unit-tests.

The main details of solving pointed out sub-tasks are detailed further. To localize .NET applications, there was used the standard technique of keeping localized strings as resx-files. The IStringLocalizer interface implementation allows extracting the required string from these files based on the "CultureInfo" object that sets needed language. This technique is described in the official documentation [8].

Machine translation was not considered for solving localization tasks due to its insufficient quality of translations. Besides, in Email templates, there were many strings, the meaning of which depends on the context. For example, "owner" in the definition of "a responsible employee", "note" in the meaning of "commentary".

The unique strings selected for localization were collected in tables and transferred to the local offices of DP Technology for manual translation. Before that, the number of materials has been prepared to increase the quality of translation. There were comments with explanations of context, as well as images of original email templates without any personal data (Figure 15).

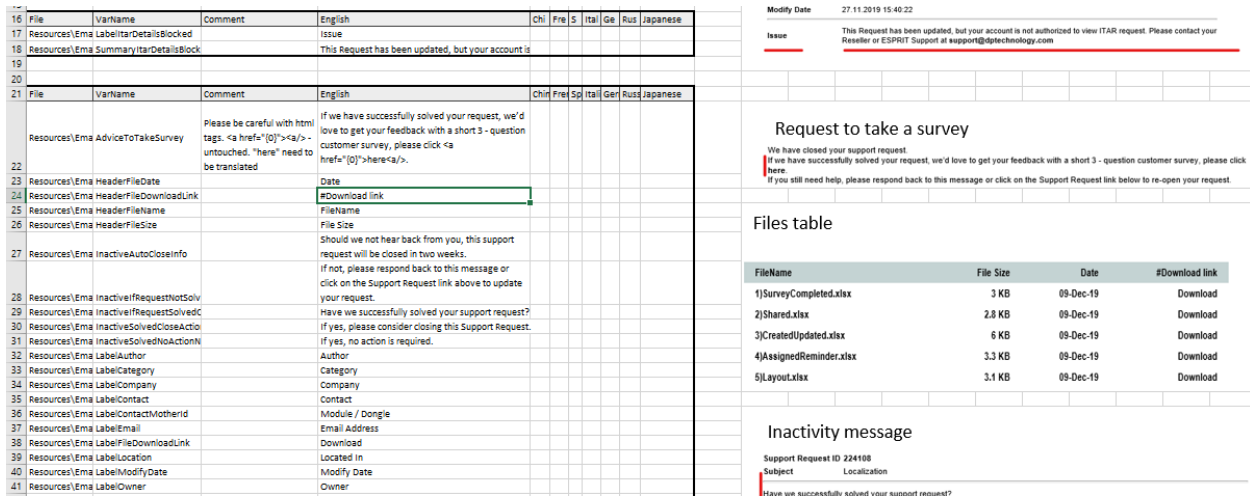


Figure 57. The table with localizable strings and additional materials

The next step was forming the resx-files with the translated lines for each language. It was performed using an extension for Visual Studio - ResxManager (Figure 16).

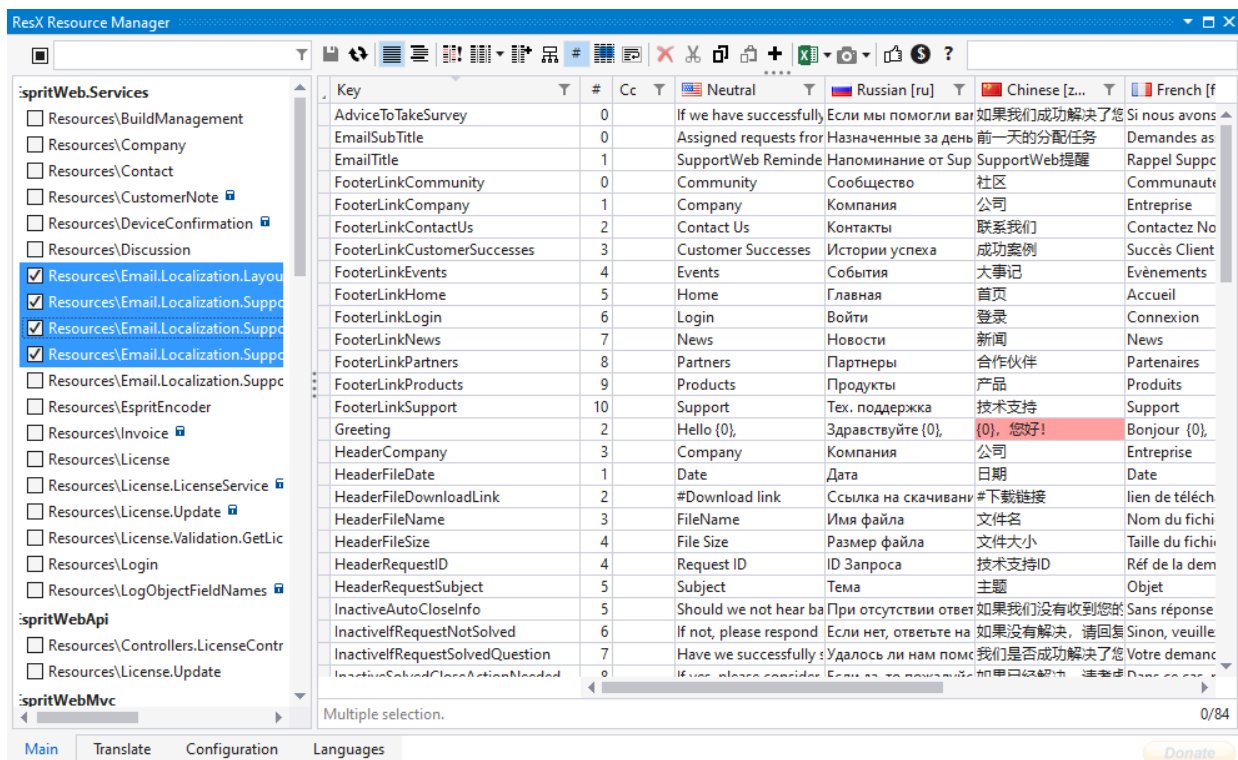


Figure 58. Forming resx files in ResxManager

During the development of the described functionality, significant changes were made in the EspritWeb.Services project because it is responsible for the business logic of the application. The architecture of notification services was studied before the

implementation of the localization mechanism. We performed its structure in the UML package diagram (Figure 17).

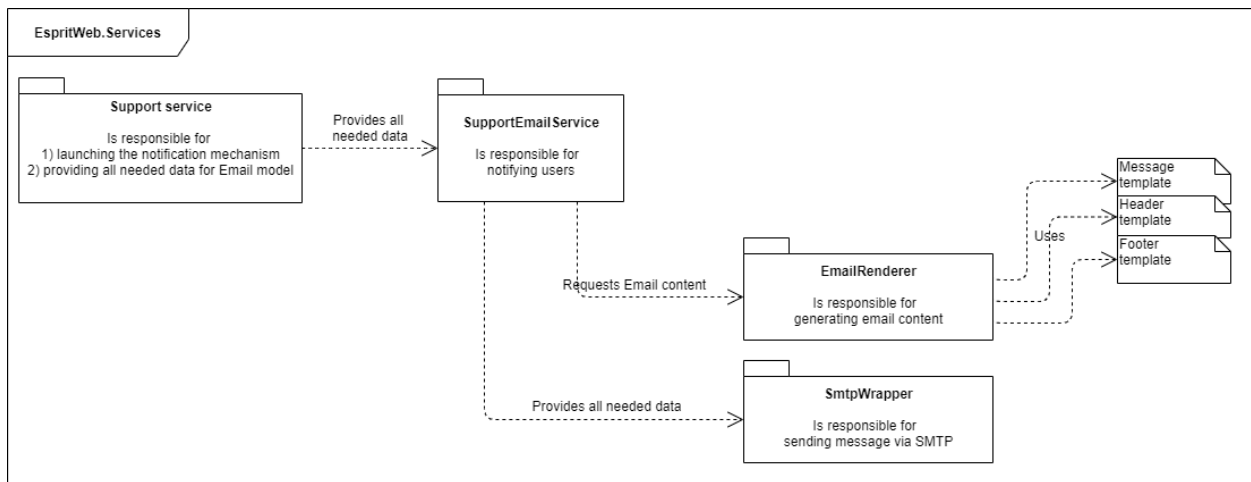


Figure 59. UML service architecture diagram (before localization)

The figure above shows four modules:

- SupportService handles support requests that are created by users and, if it is necessary, it launches email notifications sending services. Furthermore, it gets and process all the required data for each type of notification. This data is passed on to SupportEmailService;
- SupportEmailService calls EmailRenderer to generate a specific type of email based on the received data. Then, it forms a list of users for notification according to business logic (it fills Cc, To, Bcc, Reply-To the fields of the email message) and passes the data generated by the renderer to SmtWrapper that sends the fulfilled message;
- EmailRenderer generates emails content by substituting text and graphical data into the certain HTML-template file of the email;
- SmtWrapper directly sends a generated email message using the SMTP protocol.

Since the EmailRenderer forms the body of the message, it is needed to perform receiving of localized strings inside this module. It was settled, that if localization will be requested for a specific email, this module demand localized strings from the

recently implemented service, which is responsible only for localization. Using this structure procures the modularity of application.

After analyzing this chart and SupportService service (which contains more than 3,000 lines of code), it becomes clear that service takes on odd responsibility and has a lot of irrelevant functionality inside. This is an object-oriented programming (OOP) anti-pattern called "Monster Object."

The functionality of obtaining all the necessary data for email was moved into a separate service to solve this problem. Then, the architecture was supplemented with a localization service that will provide translated into the desired language strings. The redesigned service architecture is represented as a UML package diagram (Figure 18).

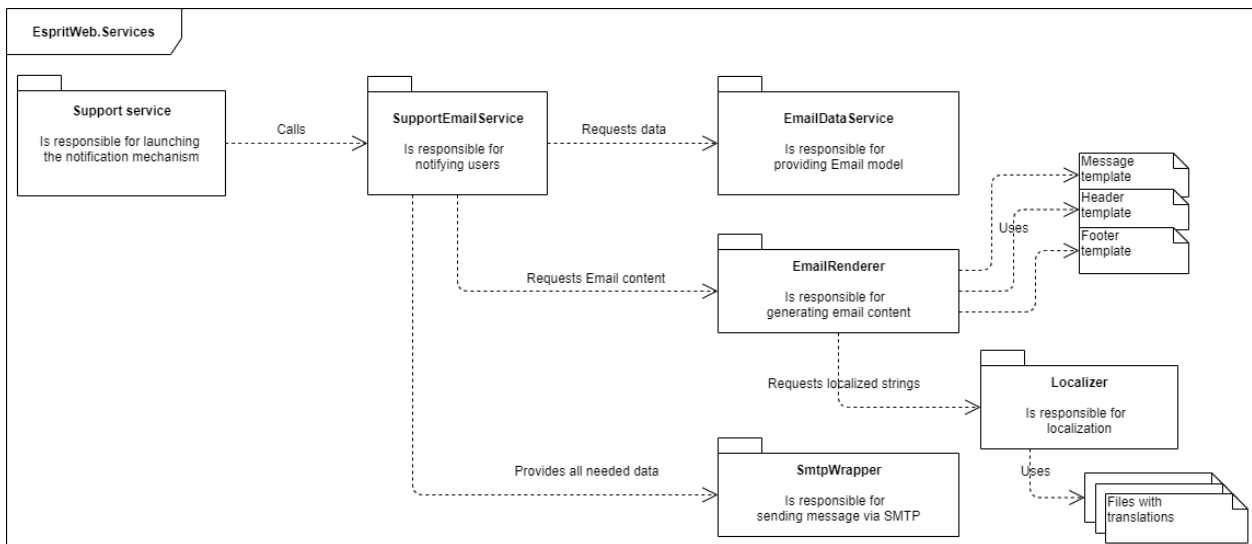


Figure 60. UML diagram of redesigned service architecture

In the redesigned Esprit Web services architecture, there were performed following significant changes:

- SupportService no longer gets and processes data that used for email generation from the database. If notifications are required according to used algorithms, the service provides only information about the type of event that occurred and the technical support request identifier;
- SupportEmailService asks EmailDataService for the evaluation and construction of a complete email data-model based on the received arguments. It doesn't receive this data from SupportService after optimizations;

- EmailDataService performs complicated database queries to receive all the data that is necessary for rendering a full message;
- EmailRenderer calls Localizer service to get translations that will be inserted into the message template;
- Localizer returns localized lines on request that it extracts from previously created resx translations.

While solving the localization problem, the structure of the web application services was reworked, the application connectivity was increased, and the cohesion of modules was decreased. Component that provides localized strings according to the resx files was developed. Email templates were adapted to use localization. As a result of the work, a scalable mechanism for the localization of mailings has been obtained.

The email notification language is set based on the preferred language of the user who created the support request. For greater flexibility, it is also possible to select the language of email notifications from the list in the request details panel (Figure 19).

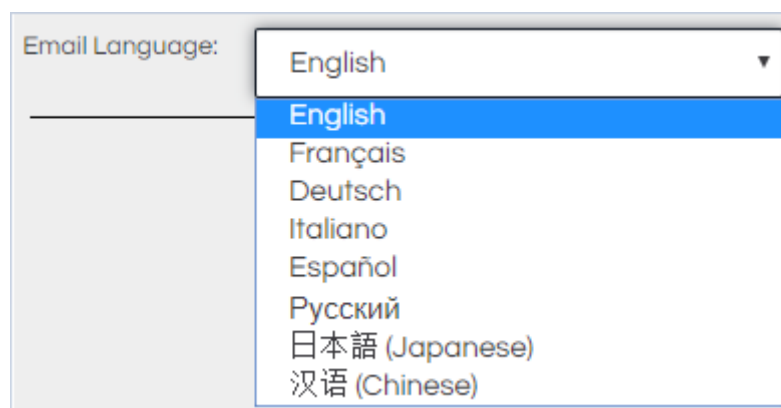


Figure 61. Languages drop-down list in the request details panel

The implemented functionality has passed performance testing and Code Review, then the final version of every translated email templates were proofread. After several amendments, the localization project was deployed. The email generation and sending are covered with more than 200 Unit-tests that checks the contents of emails, localization logic, and correctness of notified users list.

As a result, the necessary data containing user ratings was obtained to display the following diagrams (Figure 20).

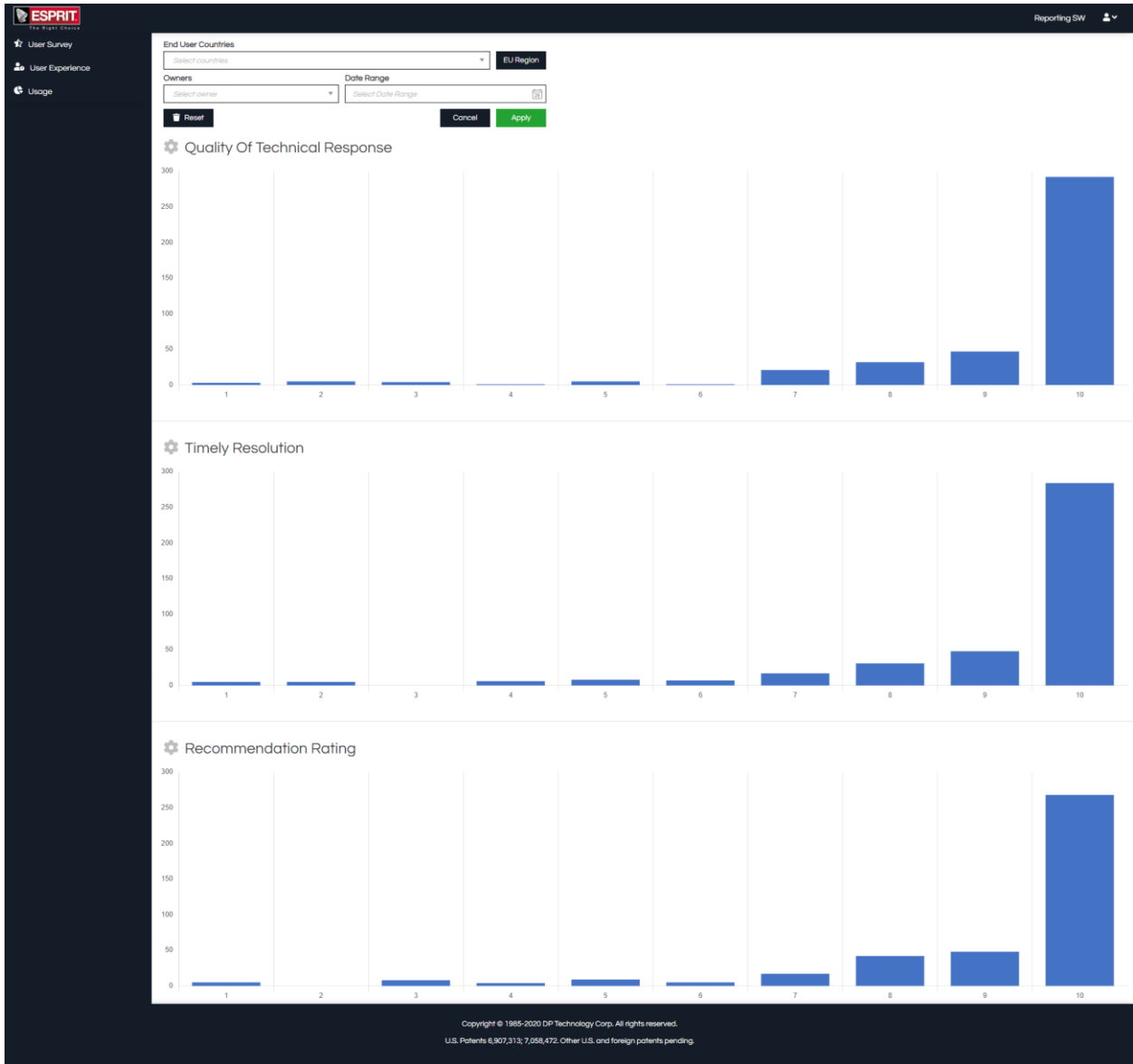


Figure 62. Charts with user quality ratings of technical support

### 3.2. Maintaining time indicators of request processing

The following database table, SupportWebReportingEvents (Figure 21), was introduced to save needed time indicators.

Column Name	Data Type	Allow Nulls
ID	int	<input type="checkbox"/>
IncidentID	int	<input type="checkbox"/>
EventType	int	<input type="checkbox"/>
EventDate	datetime	<input type="checkbox"/>
EventUserID	int	<input type="checkbox"/>

Figure 63. The SupportWebReportingEvents table structure

When a request is created, a responsible specialist is assigned, a response from a specialist is attached or request is resolved, a record with the type of action and the current time is saved in the database. The table does not take much memory to be stored, and reading data from it is performed in a flash. The system saves time data about the events presented in Table 6.

Table 26. Description of saved events

Event type	Description
Request is created	The system saves the event in the database when a customer creates a request.
Specialist is assigned	The system saves the event in the database when a responsible specialist is assigned or when a request was created with the already set owner.
Response from specialist is attached	The system saves the event in the database when the responsible specialist from DP Technology attaches the first response (in the form of a file, or a comment).
Request is resolved	The system saves the event in the database when requests state changes to "Resolved", "Closed" or "In Research and Development department". It means that the request was already processed.
Request is re-opened	The system saves the event in the database when a resolved request is re-opened.



Specialist is re-assigned	The system saves the event in database when a new responsible employee is assigned to an existing request.
---------------------------	--

The time differences between these events as well as the number of saved events are good metrics for analyzing business processes. Designed charts and diagrams display these metrics; this is shown in Table 7.

Table 27. Connection between designed diagrams and recorded events

Charts	Connection with saved events
Initial Wait Time	The chart displays the average time between the «Request is created» and «Response from specialist is attached» events.
Time to Resolve	The chart displays the average time between the «Request is created» and «Request is resolved» events.
Time to Assign	The chart displays the average time between «Request is created» and «Specialist is assigned» events.
Owner Time to Respond	The chart displays the average time between the «Specialist is assigned» and «Response from specialist is attached» events.
Support Request ReOpened	The chart displays the number of requests for which the «Request is re-opened» event is recorded.
Support Request ReAssigned	The chart displays the number of requests for which the «Specialist Reassigned» event is recorded.

Methods of SupportService (that detailed in section 3.1) have been modified to save these events on adding files, commentaries and requests.

Necessary data to display the bar charts that show the assignment time, the response time, and the request processing time was obtained. In addition, pie charts

that show the proportions of reassigned and reopened requests were designed (Figure 22).



Figure 64. The charts that displayed using saved events data

## Приложение Б

В таблице 28 представлены выделенные группы требований.

Таблица 28 – Группы требований

Код	Группа требований
F	Общие функциональные требования
AR	Требования к архитектуре
P	Требования к интеграции
D	Требования к организации и составу данных
I	Требования к интерфейсу

В таблицах 29-33 представлена спецификация вышеописанных групп требований:

Таблица 29 - Общие функциональные требования

Код	Требование	Прим.
F.1	Система должна позволять отображать диаграммы.	См. D.1, D.2, D.3
F.1.1	Система должна позволять отображать <i>столбчатые</i> диаграммы.	
F.1.2	Система должна позволять отображать <i>круговые</i> диаграммы.	
F.1.3	Система должна отображать <i>столбчатую</i> диаграмму «Quality Of Technical Response», отражающую пользовательские оценки качества обслуживания запроса.	
F.1.3.1	На диаграмме «Quality Of Technical Response» по оси X располагаются возможные оценки уровня качества технического обслуживания.	Оценки качества технического обслуживания были выставлены пользователями по шкале от 1 до 10 и сохранены в БД.

F.1.3.2	На диаграмме «Quality Of Technical Response» по оси Y располагается количество оценок с соответствующим значением уровня качества.	
F.1.4	Система должна отображать <i>столбчатую</i> диаграмму «Timely Resolution», отражающую пользовательские оценки времени обслуживания.	
F.1.4.1	На диаграмме «Timely Resolution» по оси X располагаются возможные оценки времени обслуживания (шкала от 1 до 10).	Оценки времени обслуживания были выставлены пользователями по шкале от 1 до 10 и сохранены в БД
F.1.4.2	На диаграмме «Timely Resolution» по оси Y отмечается количество оценок с соответствующим значением оценки времени обслуживания.	
F.1.5	Система должна отображать <i>столбчатую</i> диаграмму «Recommendation Rating», отражающую насколько вероятно то, что пользователь рекомендует систему Esprit коллегам или знакомым.	
F.1.5.1	На диаграмме «Recommendation Rating» по оси X располагаются возможные оценки пользователем вероятности рекомендации продукта (по шкале от 1 до 10).	Оценки вероятности рекомендации были выставлены пользователями по шкале от 1 до 10 и сохранены в БД
F.1.5.2	На диаграмме «Recommendation Rating» по оси Y отмечается количество оценок с	

	соответствующим значением вероятности рекомендации.	
F.1.6	Система должна отображать <i>столбчатую</i> диаграмму «Number of Support Requests», отражающую количество созданных запросов в техническую поддержку в конкретную дату.	
F.1.6.1	На диаграмме «Number of Support Requests» по оси X располагаются даты из выбранного в фильтрах временного диапазона.	См. F.2.3
F.1.6.2	На диаграмме «Number of Support Requests» по оси Y отмечается количество созданных запросов в соответствующую дату.	
F.1.7	Система должна отображать <i>столбчатую</i> диаграмму «Initial Wait Time», показывающую среднее время ожидания первого ответа от службы технической поддержки.	
F.1.7.1	На диаграмме «Initial Wait Time» по оси X располагаются даты из выбранного в фильтрах временного диапазона.	См. F.2.3
F.1.7.2	На диаграмме «Initial Wait Time» по оси Y отмечается среднее время ожидания первого ответа для всех запросов, созданных в соответствующую дату.	
F.1.8	Система должна отображать <i>столбчатую</i> диаграмму «Time to Resolve», показывающую среднее время обработки запросов службой технической поддержки.	

F.1.8.1	На диаграмме «Time to Resolve» по оси X располагаются даты из выбранного в фильтрах временного диапазона.	
F.1.8.2	На диаграмме «Time to Resolve» по оси Y отмечается среднее время обработки запросов службой поддержки для всех запросов, созданных в соответствующую дату.	
F.1.9	Система должна отображать <i>столбчатую</i> диаграмму «Time to Assign», показывающую среднее время назначения ответственного работника для обработки запроса.	
F.1.9.1	На диаграмме «Time to Assign» по оси X располагаются даты из выбранного в фильтрах временного диапазона.	
F.1.9.2	На диаграмме «Time to Assign» по оси Y отмечается среднее время назначения ответственного работника для всех запросов, созданных в соответствующую дату.	
F.1.10	Система должна позволять отображать <i>круговую</i> диаграмму «Support Request Re-Opened», отображающую долю вновь открытых запросов относительно всех закрытых запросов.	Запрос является вновь открытым, если после его обработки пользователь решил, что его проблема не была устранена.
F.1.11	Система должна отображать <i>круговую</i> диаграмму «Support Request Re-Assigned», отражающую долю переназначенных	Запрос является переназначенным, если у него меняется ответственный за

	запросов относительно всех назначенных запросов.	обработку специалист
F.1.12	Система должна позволять отображать <i>круговую</i> диаграмму «Version», отражающую количество созданных запросов, сгруппированных в сектора по программному продукту, в котором возникла проблема.	Основной продукт DP Technology – Esprit CAM, но в системе отслеживаются и другие.
F.1.13	Система должна позволять отображать <i>круговую</i> диаграмму «Type», отражающую количество созданных запросов, сгруппированных в сектора по типу запроса.	Типы запросов: Отчет об ошибке, запрос на программную поддержку, запрос на доработку, помощь по демо-версии и т.д.
F.1.14	Система должна отображать <i>столбчатую</i> диаграмму «Esprit Category - Numbers», показывающую количество запросов сгруппированных по категориям функционала САМ системы.	
F.1.14.1	На диаграмме «Esprit Category - Numbers» по оси X располагаются категории функционала САМ системы.	
F.1.14.2	На диаграмме «Esprit Category - Numbers» по оси Y отображается количество запросов относящихся к заданной категории функционала САМ системы.	
F.1.15	Система должна отображать <i>столбчатую</i> диаграмму «Esprit Category - Hours», показывающую суммарное время обработки	

	запросов сгруппированных по категориям функционала САМ системы.	
F.1.15.1	На диаграмме «Esprit Category - Hours» по оси X располагаются категории функционала САМ системы.	
F.1.15.2	На диаграмме «Esprit Category - Hours» по оси Y отображается суммарное время обработки запросов относящихся к заданной категории функционала САМ системы.	
F.1.16	Система должна отображать <i>столбчатую</i> диаграмму «Support by Reseller - Hours», показывающую суммарное время обработки запросов дистрибьюторами, сгруппированное по организациям дистрибьюторов.	
F.1.16.1	На диаграмме «Support by Reseller - Hours» по оси X располагаются названия организаций дистрибьюторов.	
F.1.16.2	На диаграмме «Support by Reseller - Hours» по оси Y отображается суммарное время обработки запросов заданной организацией.	
F.1.17	Система должна отображать <i>столбчатую</i> диаграмму «Support by Reseller - Numbers», показывающую суммарное количество запросов обработанное дистрибьюторами, сгруппированное по организациям дистрибьюторов.	



F.1.17.1	На диаграмме «Support by Reseller - Numbers» по оси X располагаются названия организаций дистрибьюторов.	
F.1.17.2	На диаграмме «Support by Reseller - Numbers» по оси Y отображается суммарное количество запросов обработанное заданной организацией.	
F.1.18	Система должна отображать <i>круговую</i> диаграмму «Support by Region - Hours», отражающую количество созданных запросов, сгруппированных в сектора по странам обратившихся пользователей.	
F.1.19	Система должна позволять отображать <i>круговую</i> диаграмму «Support by Region - Numbers», отражающую суммарное время обработки запросов, сгруппированных в сектора по странам обратившихся пользователей.	
F.1.20	Система должна отображать <i>столбчатую</i> диаграмму «Owner Time to Respond», показывающую среднее время ответа специалиста после его назначения.	
F.1.20.1	На диаграмме «Owner Time to Respond» по оси X располагаются даты из выбранного в фильтрах временного диапазона.	
F.1.20.1	На диаграмме «Owner Time to Respond» по оси Y отмечается среднее время ответа специалиста после его назначения для всех	

	запросов, созданных в соответствующую дату.	
F.2	Система должна позволять фильтровать запросы, участвующие в формировании данных для диаграмм.	
F.2.1	Система должна предоставлять возможность фильтрации запросов по одной или нескольким странам создателя запроса.	
F.2.2	Система должна предоставлять возможность фильтрации запросов по одному или нескольким ответственным за обработку работникам	Список работников должен состоять из работников технической поддержки DP Technology
F.2.3	Система должна предоставлять возможность фильтрации запросов по вхождению даты создания в заданный временной диапазон.	
F.2.3.1	Фильтр временного диапазона состоит из двух полей ввода которыми задается начало и конец временного диапазона.	
F.2.3.2	Ввод начала или конца временного диапазона должен ограничивать ввод противоположной границы диапазона.	
F.2.4	Выбранные фильтры для диаграмм должны сохраняться для пользователей сайта.	Сохранение фильтров должно происходить на клиентской части приложения
F.2.5	Система должна предоставлять возможность сбрасывать настройки фильтров.	

F.2.6	Система должна перестраивать диаграммы в соответствии с изменениями в фильтрах.	Перестройка диаграмм должна происходить без перезагрузки страницы
F.2.7	Система должна позволять устанавливать страны европейского региона в фильтре стран в одно действие.	
F.2.8	Компонент фильтров должен позволять настраивать фильтрацию запросов сразу для нескольких диаграмм.	

Таблица 30- Требования к архитектуре

Код	Требование	Примечание
AR.1	Система должна быть построена клиент-серверной архитектуре.	
AR.2	Архитектурная модель системы должна состоять из трёх компонентов: клиента, сервера приложений и сервера баз данных.	

Таблица 31 - Требования к интеграции

Код	Требование	Примечание
P.1	Пользователи, авторизованные в приложении Esprit Web должны быть авторизованы автоматически в разрабатываемом приложении и наоборот.	
P.2	Все используемые для отображения статистики данные должны быть взяты из основной БД приложения Esprit Web.	

Таблица 32 - Требования к организации и представлению данных

Код	Требование	Примечание
D.1	Данные диаграмм по умолчанию должны быть представлены в абсолютных величинах.	F.1

D.2	Данные внутри диаграмм должны быть представлены в относительных величинах по требованию пользователя.	F.1
D.3	В легенде диаграмм должны быть представлены названия категорий/критериев, по которым сгруппированы данные.	F.1
D.4	Числовые данные должны быть округлены до двух знаков после запятой по математическому округлению.	
D.5	Данные внутри диаграмм должны быть отсортированы по значениям, либо по названиям элементов.	

Таблица 33 - Требования к интерфейсу

Код	Требование	Примечание
I.1	Система должна поддерживать масштабирование интерфейса.	
I.2	Интерфейс системы должен корректно отображаться на экранах с разрешением 800x600 пикселей и выше.	
I.3	Интерфейс системы должен корректно отображаться на мобильных устройствах.	
I.4	Интерфейс системы должен корректно отображаться в браузерах: <ul style="list-style-type: none"> <li>• Google Chrome версии 52 и выше;</li> <li>• Mozilla Firefox версии 20 и выше;</li> <li>• Opera версии 12.1 и выше;</li> <li>• Internet Explorer версии 11 и выше;</li> <li>• Edge версии 12 и выше;</li> <li>• Safari версии 9 и выше.</li> </ul>	

I.5	<p>Стилистика реализуемого приложения должна соответствовать стилистике основного веб-сайта Esprit. Требуется соответствие</p> <ul style="list-style-type: none"> <li>• Применяемых css стилей;</li> <li>• Используемых шрифтов;</li> <li>• Цветовой схемы.</li> </ul>	
I.6	Сектора на круговых диаграммах должны быть залиты разными цветами.	F.1.2
I.7	На круговой диаграмме должна быть возможность объединять сектора, отображающие незначительные значения (менее 0.5% от общей суммы) в сектор «Other».	F.1.2
I.8	Компонент выбора временного диапазона должен поддерживать ввод как с помощью выпадающих календарей, так и с помощью ввода дат с клавиатуры.	F.2
I.9	<p>Компонент выбора временного диапазона должен содержать кнопки для выбора предустановленных временных диапазонов:</p> <ul style="list-style-type: none"> <li>• Прошедший месяц;</li> <li>• Прошедшая неделя;</li> <li>• Текущая неделя.</li> </ul>	F.2
I.10	Фильтр по ответственным работникам должен поддерживать текстовый поиск по своим элементам.	F.2
I.11	Фильтр по странам должен поддерживать текстовый поиск по своим элементам.	F.2
I.12	Фильтр по ответственным работникам должен поддерживать множественный выбор своих элементов и отображать выбранные значения.	F.2

I.13	Фильтр по странам должен поддерживать множественный выбор своих элементов и отображать выбранные значения.	F.2
I.14	Макет разрабатываемого приложения должен соответствовать макету основного сайта.	
I.14.1	Панель навигации разрабатываемого приложения должна располагаться сверху.	
I.14.2	Футер разрабатываемого приложения должен располагаться снизу и полностью соответствовать футеру основного веб-приложения Esprit.	
I.14.3	Меню доступных действий располагается слева.	

## Приложение В

Таблица сравнения программных продуктов, используемых для создания дистрибутивов

Программный продукт	Поддержка создания шаблонов	Платформа работы с решением	Развертывание	Поддержка работы с MS SQL Server	Поддержка сложных запросов и группировок	Тип лицензии	Сложность освоения	Возможность проверки цифровой подписи	Гибкая настройка отображения данных
Google Data Studio	Да	Веб-приложение	Облако	Нет	Да	Условно-бесплатная	Низкая	Нет	Частично
Plotly	Да	Веб-приложение	Облако	Нет	Нет	Условно-бесплатная	Высокая	Да	Да
Power BI	Да	Десктопное приложение	Локально	Да	Да	Условно-бесплатная	Высокая	Да	Нет

## Приложение Г

### Матрица SWOT-анализа

	<b>Сильные стороны проекта</b>	<b>Слабые стороны проекта</b>
	<ul style="list-style-type: none"> <li>• Высокая степень автоматизации бизнес процессов</li> <li>• Удобство использования</li> <li>• Высокая производительность, быстрое получение нужных метрик</li> <li>• Высокая квалификация и диверсификация команды</li> <li>• Простота сопровождения</li> </ul>	<ul style="list-style-type: none"> <li>• Узкая направленность разработки</li> <li>• Сложная структура проекта</li> <li>• Большие трудозатраты</li> <li>• Появление более приоритетных задач команды разработчиков</li> <li>• Относительно высокие расходы на разработку</li> </ul>
<b>Возможности</b>	<b>Направления развития</b>	<b>Компенсирующие мероприятия</b>
<ul style="list-style-type: none"> <li>• Использование проекта для отображения статистики по новым бизнес-процессам</li> <li>• Оперативное предоставление необходимой информации в наглядном виде.</li> <li>• Улучшение качества работы отдела технической поддержки.</li> <li>• Рациональное распределение трудовых ресурсов</li> <li>• Увеличение конкурентно-способности</li> </ul>	<ol style="list-style-type: none"> <li>1. Отображение новых метрик по другим бизнес-процессам предприятия</li> <li>2. Представление менеджменту новых метрик</li> <li>3. Улучшение эффективности работы службы технической поддержки</li> <li>4. Качественная обработка вопросов пользователей</li> </ol>	<ol style="list-style-type: none"> <li>1. Распределение труда в команде позволяет разрабатывать проект при наличии других критически важных задач</li> <li>2. Использование сильных сторон членов команды разработки позволит снизить трудозатраты</li> <li>3. Качественное проектирование архитектуры проекта упростит его структуру и снизит затраты на сопровождение</li> </ol>
<b>Угрозы:</b>	<b>Угрозы развития</b>	<b>Компенсирующие мероприятия</b>
<ul style="list-style-type: none"> <li>• Некорректная работа используемых в проекте библиотек</li> <li>• Проблемы с интернет соединением</li> <li>• Перебои в работе сервера</li> <li>• Недостаток квалифицированного персонала</li> </ul>	<ol style="list-style-type: none"> <li>1. Квалификация и опыт работы команды способствуют выбору надежных библиотек</li> <li>2. Настроено резервное копирование файлов на сервере</li> <li>3. Сопровождаемость позволяет легко подстраиваться под изменяющиеся бизнес требования</li> <li>4. Для обеспечения бесперебойного соединения сервер использует несколько провайдеров.</li> </ol>	<ol style="list-style-type: none"> <li>1. В проекте используются популярные и активно развивающиеся библиотеки, что уменьшает риск появления неустраняемых ошибок.</li> <li>2. Для уменьшения вероятности задержек реализации из-за недостатка кадров проведено планирование работ с учетом отпусков</li> </ol>



<ul style="list-style-type: none"><li>• Появление новых требований к проекту</li></ul>		3. Основные риски разработки были учтены при планировании сроков и бюджета
--	--	--

## Приложение Д

Временные показатели реализаций проекта для вариантов исполнения И1, И2, И3

Название работы	Исполнители	Трудоемкость работ, человеко-дни									Длительность работ					
		tmin			tmax			toж			Тр, рабочие дни			Тк, календарные дни		
		И1	И2	И3	И1	И2	И3	И1	И2	И3	И1	И2	И3	И1	И2	И3
Постановка задачи	1	1	1	1	2	2	2	1,4	1,4	1,4	1,7	1,7	1,7	2,5	2,5	2,5
Анализ предметной области	2	4	4	4	6	6	6	4,8	4,8	4,8	5,8	5,8	5,8	8,4	8,4	8,4
Формализация требований к приложению	1	2	2	2	4	4	4	2,8	2,8	2,8	3,4	3,4	3,4	4,9	4,9	4,9
Разработка календарного плана	2	2	2	2	1	1	1	1,6	1,6	1,6	1,9	1,9	1,9	2,8	2,8	2,8
Исследование существующих решений	1	3	3	3	6	6	6	4,2	4,2	4,2	5,0	5,0	5,0	7,4	7,4	7,4
Выбор методов и средств разработки	1	1	3	3	2	5	5	1,4	3,8	3,8	1,7	4,6	4,6	2,5	6,7	6,7
Доработка Esprit Web для получения данных	3	20	20	20	36	36	36	26,4	26,4	26,4	31,7	31,7	31,7	46,3	46,3	46,3

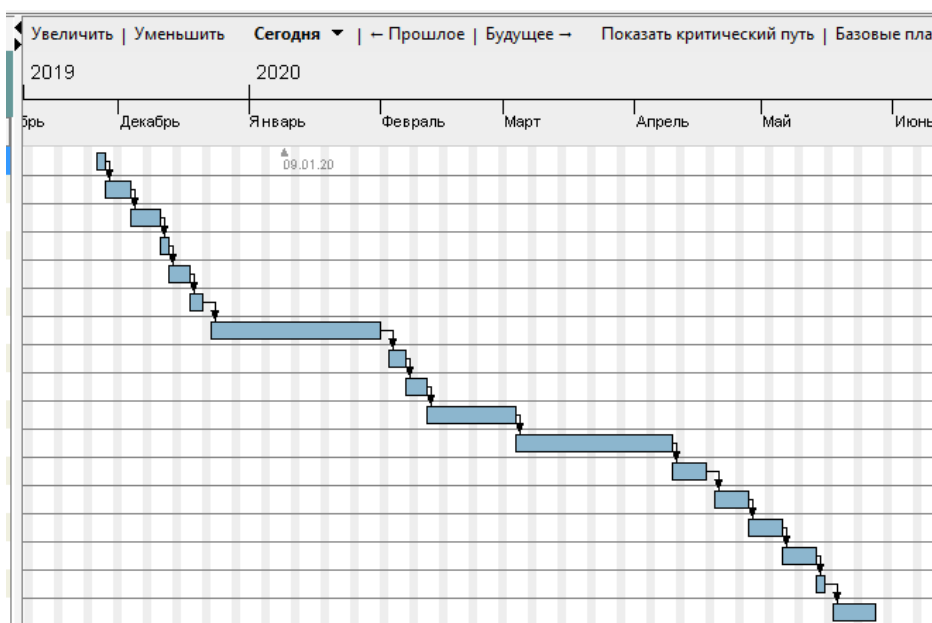
Проектирование архитектуры клиентской части модуля	2	3	6	6	4	7	7	3,4	6,4	6,4	4,1	7,7	7,7	6,0	11,2	11,2
Проектирование архитектуры серверной части модуля	1	2	3	3	3	4	4	2,4	3,4	3,4	2,9	4,1	4,1	4,2	6,0	6,0
Разработка компонентов серверной части	1	12	15	15	14	17	17	12,8	15,8	15,8	15,4	19,0	19,0	22,4	27,7	27,7
Разработка компонентов клиентской части	2	18	14	14	28	28	28	22,0	19,6	19,6	26,4	23,5	23,5	38,5	34,3	34,3
Оптимизация времени получения данных	1	3	3	3	4	4	4	3,4	3,4	3,4	4,1	4,1	4,1	6,0	6,0	6,0
Покрытие функционала Unit-тестами	1	3	3	3	4	4	4	3,4	3,4	3,4	4,1	4,1	4,1	6,0	6,0	6,0
Тестирование	2	3	4	4	4	5	5	3,4	4,4	4,4	4,1	5,3	5,3	6,0	7,7	7,7
Исправление ошибок	1	2	2	2	6	6	6	3,6	3,6	3,6	4,3	4,3	4,3	6,3	6,3	6,3
Оценка полученных результатов	1	1	1	1	2	2	2	1,4	1,4	1,4	1,7	1,7	1,7	2,5	2,5	2,5
Подготовка отчетной документации	1	4	4	4	6	6	6	4,8	4,8	4,8	5,8	5,8	5,8	8,4	8,4	8,4

<b>Итого</b>	103,2	111,2	111,2	123,8	133,4	133,4	180,8	194,8	194,8
--------------	-------	-------	-------	-------	-------	-------	-------	-------	-------

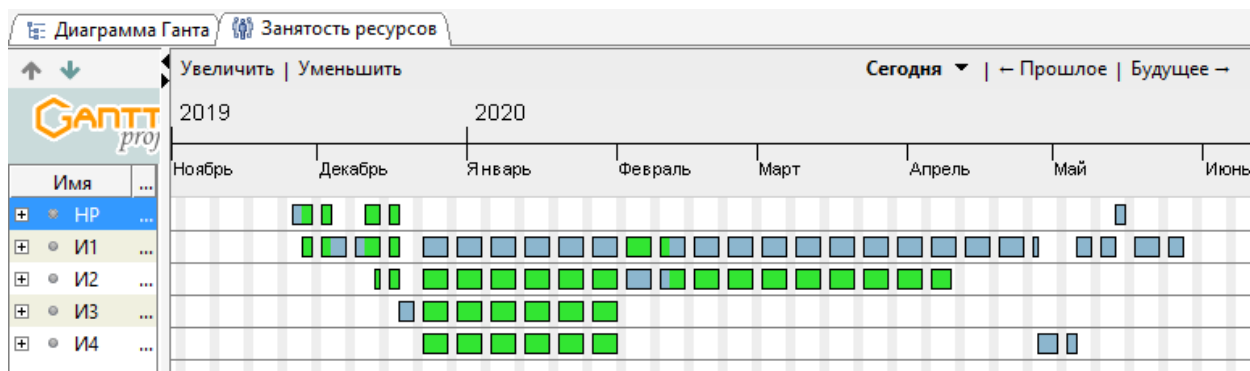
## Приложение Е

### Календарный план-график проведения работ

Название	Дата начала	Дата окончания
Постановка задачи	26.11.19	27.11.19
Анализ предметной области	28.11.19	03.12.19
Формализация требований к приложению	04.12.19	10.12.19
Разработка календарного плана	11.12.19	12.12.19
Исследование существующих решений	13.12.19	17.12.19
Выбор методов и средств разработки	18.12.19	20.12.19
Доработка Esprit Web для получения данных	23.12.19	31.01.20
Проектирование архитектуры клиентской части м...	03.02.20	06.02.20
Проектирование архитектуры серверной части м...	07.02.20	11.02.20
Разработка компонентов серверной части	12.02.20	03.03.20
Разработка компонентов клиентской части	04.03.20	09.04.20
Оптимизация времени получения данных	10.04.20	17.04.20
Покрытие функционала Unit-тестами	20.04.20	27.04.20
Тестирование	28.04.20	05.05.20
Исправление ошибок	06.05.20	13.05.20
Оценка полученных результатов	14.05.20	15.05.20
Подготовка отчетной документации	18.05.20	27.05.20



### Диаграмма занятости ресурсов



## Список литературы

1. СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы.
2. ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования.
3. ГОСТ Р 50923-96. Дисплеи. Рабочее место оператора. Общие эргономические требования и требования к производственной среде. Методы измерения.
4. ГОСТ 12.0.003-2015. Опасные и вредные производственные факторы. Классификация.
5. СанПиН 2.2.4.548-96. Гигиенические требования к микроклимату производственных помещений.
6. СП 52.13330.2016. Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95\*.
7. ГОСТ 12.1.003-2014 ССБТ. Шум. Общие требования безопасности.
8. СН 2.2.4/2.1.8.562-96. Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории застройки.
9. ГОСТ 12.1.006-84 ССБТ. Электромагнитные поля радиочастот. Допустимые уровни на рабочих местах и требования к проведению контроля (с Изменением N 1).
10. ГОСТ 12.1.038-82 ССБТ. Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов (с Изменением N 1).
11. ГОСТ Р 22.3.03-94. Безопасность в ЧС. Защита населения. Основные положения.
12. Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 24.04.2020).
13. СанПин 2.2.4.3359-16. Санитарно-эпидемиологические требования к физическим факторам на рабочих местах.

14. Кодекс Российской Федерации об административных правонарушениях от 30.12.2001 N 195-ФЗ (ред. от 24.04.2020) (с изм. и доп., вступ. в силу с 01.06.2020).

15. Безопасность жизнедеятельности: практикум / Ю.В. Бородин, М.В. Василевский, А.Г. Дашковский, О.Б. Назаренко, Ю.Ф. Свиридов, Н.А. Чулков, Ю.М. Федорчук. – Томск: Изд-во Томского политехнического университета, 2009. – 101 с.

16. Рихтер, Джеффри CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Джеффри Рихтер. - М.: Питер, 2013. - 928 с.

17. Фримен, Адам ASP.NET MVC Framework 5 с примерами на C# 5.0 для профессионалов / Адам Фримен, Стивен Сандерсон. - М.: Вильямс, 2015. - 672 с.

18. Венц, Кристиан Программирование в ASP.NET / Кристиан Венц. - М.: Символ-плюс, 2013. - 512 с.

19. YouTrack – онлайн баг-трекер и инструмент для управления задачами [Электронный ресурс] / Официальный сайт JetBrains. URL: <https://jetbrains.ru/products/youtrack/>, свободный. Яз. Рус.

20. Система автоматизированного проектирования [Электронный ресурс] / Wikipedia. URL: [https://ru.wikipedia.org/wiki/Система\\_автоматизированного\\_проектирования](https://ru.wikipedia.org/wiki/Система_автоматизированного_проектирования), свободный. Яз. Англ.

21. Автоматизация производства Esprit [Электронный ресурс] / Официальный сайт Esprit. URL: <https://www.Espritcam.com/ru-ru/>, свободный. Англ. яз.

22. Bitbucket [Электронный ресурс] / Wikipedia. URL: <https://ru.wikipedia.org/wiki/Bitbucket>, свободный. Яз. Рус.

23. Современный DOM: полифиллы [Электронный ресурс] / Современный учебник JavaScript. URL: <https://learn.javascript.ru/dom-polyfill>, свободный. Яз. Англ.

24. Chart.js [Электронный ресурс] / Официальный сайт библиотеки Chart.js. URL: <https://www.chartjs.org/>, свободный. Англ. яз.

25. Model-View-Controller [Электронный ресурс] / Wikipedia. URL: <https://ru.wikipedia.org/wiki/Model-View-Controller>, свободный. Яз. Рус.

26. Совместное использование файлов cookie проверки подлинности между приложениями ASP.NET [Электронный ресурс] / MSDN – сеть разработчиков Microsoft. URL: <https://docs.microsoft.com/ru-ru/aspnet/core/security/cookie-sharing?view=aspnetcore-2.2>, свободный. Яз. Рус.

27. Приложение SQL Server Profiler [Электронный ресурс] / MSDN – сеть разработчиков Microsoft. URL: <https://docs.microsoft.com/ru-ru/sql/tools/sql-server-profiler/sql-server-profiler?view=sql-server-2017>, свободный. Яз. Рус.

28. Vue.js [Электронный ресурс] / Официальный сайт библиотеки Vue.js. URL: <https://vuejs.org/>, свободный. Яз. Англ.

29. JavaScript: основы работы с LocalStorage [Электронный ресурс] / Цикл статей о программировании [ustimov.org](http://ustimov.org). URL: <https://ustimov.org/posts/16/>, свободный. Яз. Рус.

30. About xUnit.net [Электронный ресурс] / Официальный сайт библиотеки xUnit. URL: <https://xunit.net/>, свободный. Яз. Англ.