

РАЗРАБОТКА И РЕАЛИЗАЦИЯ МЕТОДИКИ УПРАВЛЕНИЯ РЕСУРСАМИ В КОМПАНИИ СЕРВИСНОГО ОБСЛУЖИВАНИЯ

**Львович И.Я., Львович Я.Е., Преображенский А.П.,
Преображенский Ю.П., Чопоров О.Н.**

*(Воронежский институт высоких технологий)
(Воронежский государственный технический университет)*
komkovvibt@yandex.ru, app@vibt.ru

В статье рассматриваются особенности управления компанией, связанной с сервисным обслуживанием. Приведена схема логики обработки входящих инцидентов между линиями поддержки. Дана логическая схема модуля предсказания для определения команды поддержки.

Ключевые слова: компания, управление ресурсами, сервисное обслуживание.

1. Введение

Сервис менеджмента является достаточно распространенным в крупных компаниях. Многие предприятия не стремятся открывать у себя IT отделы в силу экономических выгод или других особенностей, которые нам неизвестны.

Для этого предприятия заключают договор с обслуживающей компанией, которая предоставляет для заказчика оборудование в аренду, инфраструктуру и кадры, которые должны обслуживать все это и конечных пользователей заказчика в том числе.

В данной работе мы рассмотрим работу компании, которая занимается поддержкой бизнес-пользователей из многих других компаний. Несмотря на то, что договоренности для разных проектов будут значительно отличаться [1, 2], принципиальное

построение ИТ поддержки соблюдается согласно процессам сервис менеджмента в рамках методологии ИТIL. Затем мы рассмотрим устройство и взаимодействие обслуживающего ИТ департамента для примера некоторого проекта.

2. Общая схема взаимодействия компонентов модели с учетом условий сервисного обслуживания

Согласно процессам сервис менеджмента, в данном проекте можно выделить следующих участников (рис. 1):

- Конечные пользователи со стороны заказчика
- Первая линии поддержки, предоставленная третьей стороной
- Вторая и третья линии поддержки со стороны обслуживающего ИТ департамента.
- Команды поддержки третьей стороны и вендоров

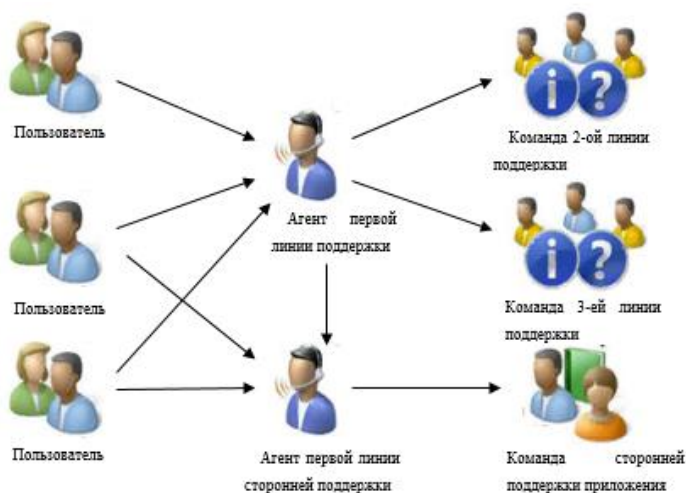


Рис. 1. Общая схема взаимодействия конечных пользователей заказчика с проектной и сторонними техническими поддержками

Конечные пользователи работают в штате Заказчика и выполняют ежедневные задачи бизнеса согласно своей роли. Чтобы была обеспечена непрерывность работы, необходим механизм решения возникающих проблем, которые имеют сущность Инцидент согласно терминам ИТIL. Эти задачи покрывает процесс Инцидент менеджмента. Когда у пользователя возникает проблема, он должен быть уверен, что при оформлении запроса, он в кратчайшие сроки, или временные рамки, установленные в договоре по сервисному обслуживанию, получит необходимое решение проблемы, так как для бизнеса и производства каждая минута простоя наносит ощутимый убыток. Для осуществления правильного и быстрого реагирования на возникающую проблему, должны выполняться следующие условия:

- Пользователь должен иметь несколько видов связи с точкой контакта технической поддержки
- Необходимо высоко функциональное программное обеспечение, которое позволяет вести учет входящих инцидентов в системе, для качественной поддержки и решения возникающих проблем на стороне заказчика
- Дополнительные роли, отвечающие за координацию инцидентов с целью уменьшения времени выполнения, изучения и решения инцидентов. Ниже представлена схема с учетом вышеперечисленных условий.
- Таким образом, в компании соблюдаются минимальные требования по поддержке конечных пользователей. Компания, которая предоставляет сервисное обслуживание конечным пользователям заказчика, отвечает за вторую и третью линии поддержки.

Основные различия такого разделения заключается в том, что инцидент, который не был решен на первой линии поддержки, должен быть отправлен на вторую линию для изучения проблемы. Если же инцидент не может быть решен силами второй линии технической поддержки, то инцидент должен быть переправлен на третью линию поддержки для дальнейшего поиска решения [3, 4]. Такой подход позволяет отсеивать инциденты на соответствующих линиях поддержки, тем самым обеспечивая

возможность изучения проблемы по соответствующим уровням компетенций инженеров на линиях поддержки.

Стоит отметить, что количество линий поддержки может быть сколь угодно много, но обычно используется всего три линии поддержки, и если третья линия не может справиться с инцидентом, то могут быть инициализированы другие процессы, например, процесс Управления Проблемами, или инцидент может быть передан в техническую поддержку приложения или вендора (рис. 2).

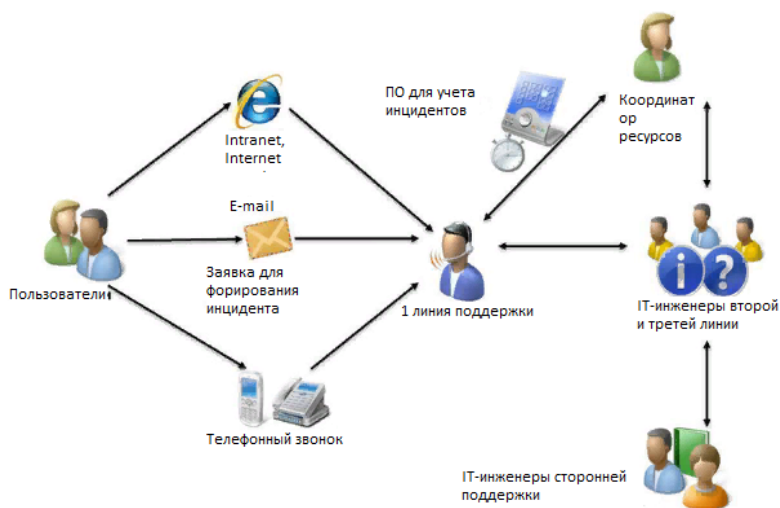


Рис. 2. Общая схема взаимодействия конечных пользователей заказчика с проектной и сторонними техническими поддержками учетом условий сервисного обслуживания

Для обработки и содержания серверов используются дата-центры с большим количеством физических серверных стоек.

Одним из видов деятельности Координатора инцидентов являются задачи распределения инцидентов между соответствующими командами поддержки.

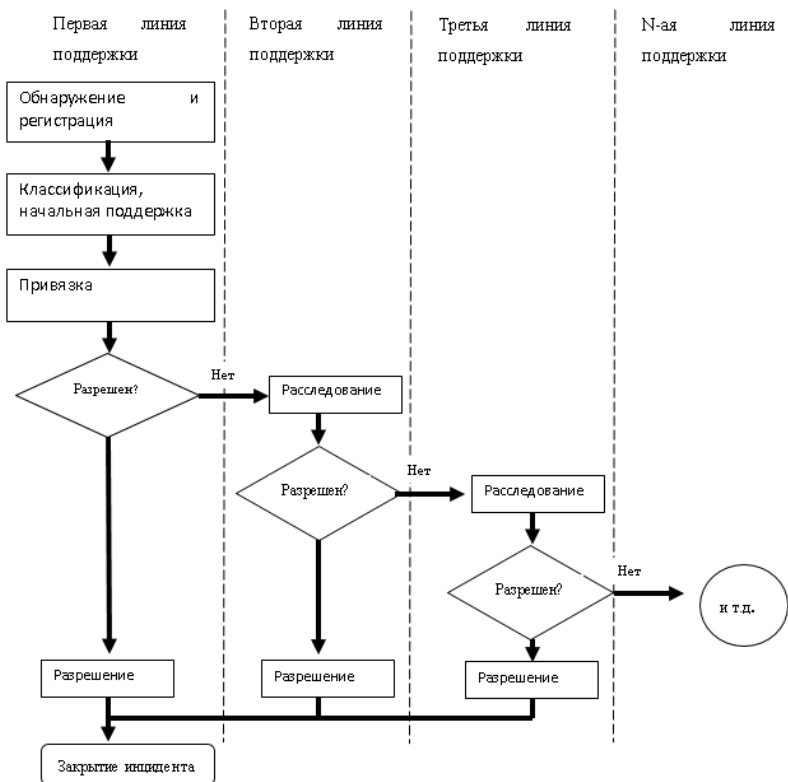


Рис. 3. Схема логики обработки входящих инцидентов между линиями поддержки

Известно, что к инженерам попадает два вида сущности инцидента:

- Пользовательские инциденты
- Мониторинговые инциденты

Система ITSM на исследуемом проекте имеет следующую особенность – когда агенты первой линии поддержки создают инцидент, они не назначают их на конкретную команду поддержки, так как могут не знать в чем основная причина инцидента и это частая ситуация, когда сотрудники первой линии технической

поддержки не обладают достаточными компетенциями для точного определения причины проблемы описанной в инциденте.

Для этого предлагается создать специальную общую очередь для инцидентов (рис. 4), относящихся к командам поддержки ИТ департамента, который содержит в себе участников второй линии поддержки.

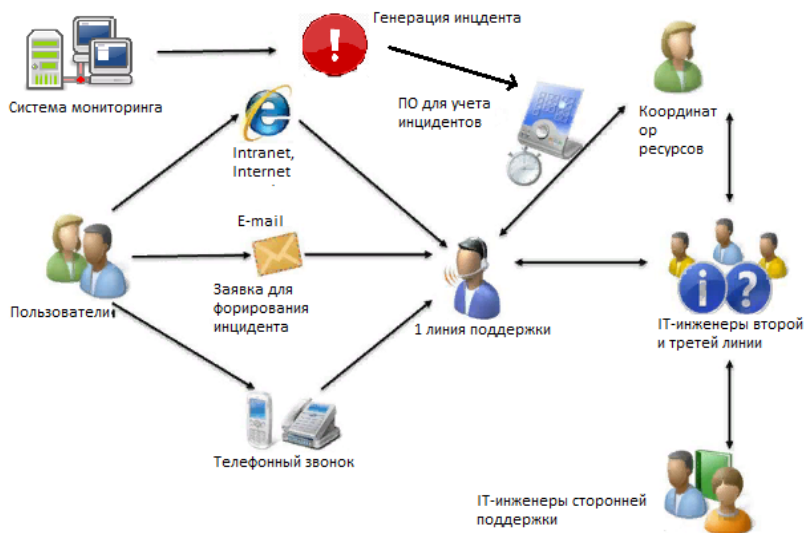


Рис. 4. Схема логики обработки входящих инцидентов между линиями поддержки

Несмотря на то, что общая очередь инцидентов включает в себя всех участников со второй линии поддержки, для каждой команды существует своя очередь инцидентов и в нее входят участники второй и третьей линии по соответствующей поддержке приложения или части инфраструктуры.

Общая очередь используется для пользовательских инцидентов, которые заводятся агентами технической поддержки на первой линии и для мониторинговых инцидентов, которые были изначально настроены на отправку в общую очередь инцидентов в мониторинговой системе. Таким образом инженеры второй

линии должны постоянно следить за общей очередью на наличие появляющихся инцидентов.

Здесь есть большой риск нарушения сервисного времени выполнения инцидентов по причине несогласованности действий инженеров. Причинами могут служить как человеческий фактор, так и не добросовестность инженеров, что приводит к нарушению договора по выполнению и решению инцидента и последующими санкциями со стороны заказчика.

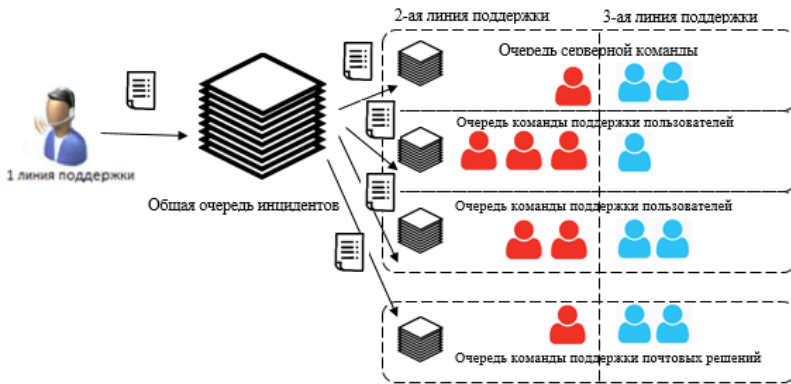


Рис. 5. Логическая схема разделения стеков команд поддержки в ITSM системе

Чтобы избежать подобных проблем, в процесс инцидент менеджмента вводится дополнительная роль Координатора инцидентов. Основной, интересующей в данном исследовании, задачей является высокоуровневая координация инцидентов, их распределение между командами и инженерами, с целью обеспечения максимального быстрого назначения инцидентов на инженеров [5, 6].

На вход обученной модели подается краткое и полное описание инцидента, которое после преобразования выдает результат в виде рейтинга вероятных команд поддержки по убыванию.

Пример предполагаемых команд поддержки:

1. Команда поддержки почтовых серверов – 94%
2. Команда поддержки конечных пользователей серверов – 5%
3. Команда поддержки IIS Web Services – 1%

Рис. 6. Пример выходного результата работы модели в рамках программного решения

Результаты работы программы выводятся в виде удобочитаемого текста, который на данном этапе выполняет информационную роль в помощь Координатору решения в принятии решения о назначении инцидента в стек команды. Выполнение программы с момента входа исходных данных до отображения результата занимает несколько секунд и покрывает первые 4 этапа процедуры Координатора инцидентов с приблизительной точностью около 90%, в то время как Координатор инцидентов тратит около 10 минут времени с высокой долей ошибки определения команды, ответственной за решение инцидента. Координатору остается только выбрать подходящего инженера.

3. Подготовка входных данных и их преобразование

После того, как получены данные в необработанном виде, наступает этап подготовки и описания модели. Можно выделить несколько этапов преобразования данных:

1. Преобразование описания инцидентов. Краткое описание и полное описание объединяется в одну строку для последующей трансформации. Впоследствии используется в качестве входных данных.
2. Проверка маркеров. Маркеры — это результат предсказания модели. На примере данных Таблицы 1 маркерами являются значения столбца Команда поддержки для текущей подзадачи.
3. Текстовый препроцессинг.
 - Векторизация текста. При создании модели использовался векторизатор, потому что он отлично справляется

с трансформацией текста в векторы на сравнительно небольшом количестве данных. Основной идеей векторизации является развертывание всего текста в матрицу, где столбец является элементом документов, а строка – самим документом. К примеру, у нас есть 4 документа:

- Почта не открывается на компьютере, 2. Почта открывается очень долго, 3. Компьютер завис и не отвечает, 4. Компьютер запускается очень долго.
- Каждый элемент документа – слово, которое называется юниграммой. На примере этих документов мы получаем колонки матрицы:

Таблица 1. Заголовки матрицы векторизатора, состоящая из элементов (юниграмм)

Почта	Не	Открывается	На	Компьюте	Завис	Отвеча	И	Запускается
				р		ет		

После этого происходит определение элементов, которое встречается в каждом документе. После того, как шаблон сгенерирован, заполнение матрицы происходит по следующему механизму: если элемент встречается в текущем документе, то ему присваивается 1, иначе присваивается 0.

В конечном итоге образуется многомерная матрица, где для каждого документа составляется вектор слов, которые встречаются в документе.

Стоит отметить, что именованная колонок матрицы следует именовать как характеристики документов.

Был рассмотрен простейший пример векторизации N-грамм. В исследовании используется преобразование текста в число (разреженная матрица) с TF-IDF n грамм, где минимальная частота для функции равна 3.

Таблица 2. Итоговая матрица векторизации

	Почта	не	Открывается	на	Компьютер	Завис	Отвечает	и	Запускается
Почта не открывается на компьютере	1	1	1	1	1	0	0	0	0
Почта открывается очень долго	1	0	1	0	0	0	0	0	0
Компьютер завис и не отвечает	0	1	0	0	1	1	1	1	0
Компьютер запускается очень долго	0	0	0	0	1	0	0	0	1

Использование tf-idf позволяет преобразовать используемые документы в следующую матрицу:

Таблица 3. Итоговая матрица векторизации на базе TF-IDF

	Почта	Не	Открывается	на	Компьютер	Завис	отвечает	и	Запускается
Почта не открывается на компьютере	0,32	0,32	0,61	0,61	0,48	0	0	0	0
Почта открывается очень долго	0,32	0	0,32	0	0	0	0	0	0
Компьютер завис и не отвечает	0	0,32	0	0	0,61	0,48	0,48	0,61	0,48
Компьютер запускается очень долго	0	0	0	0	0,32	0	0	0	0,48

4. Подготовка модели для обучения и тестирования

После того, как все этапы подготовки, преобразования и трансформации текста выполнены, наступает этап обучения модели. Как упомянуто ранее, в качестве модели выбран алгоритм логистической регрессии, который обучается на 70% данных и тестируется на оставшихся тридцати процентах. Так как преобразования текста, создание модели и ее обучение производится в программной среде языка Python, используется библиотека для научных вычислений `sklearn`, которая также содержит необходимый векторизатор `TF_IDF` для трансформации текста в вектора.

В основе обучения, как упоминалось ранее, лежит использование полиномиальной логистической регрессии, результатом которой является прогнозирование соответствующей команды поддержки по определенной проблеме. В ходе эксперимента выясняется, что нельзя брать как результат только одну команду поддержки, так как инцидент может иметь составную проблему, где требуется привлечение сразу несколько команд для решения [7]. С другой стороны, описание инцидента может быть составлено некорректно, и основная причина проблема пользователя может заключаться в области ответственности другой команды. Поэтому было принято решение выводить рейтинг самых вероятных команд для решения проблемы, описанной в инциденте [8].

Согласно определенным требованиям, можно получить ответ как должна обучаться модель [9].

Анализ и проведение экспериментов по повышению качества работы модели

После того, как была получена обученная модель, которая предсказывает команду поддержки, средствами языка Python ее можно сохранить в виде специального файла, который можно потом подключать как самостоятельный модуль (Рис.7.).

Экспериментальным путем, было выяснено, что для каждой команды поддержки будет создана своя обученная модель, которая будет подключаться в зависимости от предсказанной команды поддержки первой моделью. Каждая модель команды поддержки содержит в себе все те же преобразования, которые

использовались для первой модели. Единственное отличие заключается в том, что для каждой модели исходные данные будут преобразовываться и фильтроваться по соответствующей команде поддержки.



Рис. 7. Логическая схема модуля предсказания для определения команды поддержки

Табл. 4. Пример данных по решенным инцидентам для конкретной команды поддержки с инженерами

Краткое описание	Полное описание	Статус	Команда поддержки	Инженер
Почта. Не работает	На моем ноутбуке не работает почта <...>	Решен	Поддержка конечного пользователя	Инженер 1
Интернет отсутствует	Не получается открыть страницу в веб-браузере <...>	Решен	Поддержка конечного пользователя	Инженер 3
Приложение не открывается на ноутбуке	Не удастся открыть приложение	Решен	Поддержка конечного пользователя	Инженер 5

Таким образом, фильтруя данные только по конкретной команде поддержки, можно предотвратить зашумленность данных, которая может повлиять на получаемое предсказание.

Для каждой модели в данных используется маркер столбца инженеров, так как именно их планируется предсказывать по

результатам выполнения работы обученной модели. Как и в первой модели происходят последовательные этапы преобразования текста, в результате которого мы получаем матрицы неточностей по количеству инцидентов и с нормализованными значениями по предсказаниям инженеров.

Engi- neer1	0.59	0.00	0.09	0.09	0.17	0.06
No in the team	0.03	0.00	0.16	0.24	0.57	0.00
Engi- neer2	0.08	0.00	0.54	0.08	0.19	0.11
Engi- neer3	0.05	0.00	0.09	0.59	0.16	0.12
Engi- neer4	0.19	0.00	0.14	0.10	0.54	0.03
Engi- neer5	0.27	0.00	0.18	0.45	0.00	0.09
	Engineer1	No in the team	Engineer2	Engineer3	Engi- neer4	Engi- neer5

Рис. 12. Нормализованная матрица неточностей между инженерами команды поддержки в процентном соотношении.

Рассмотрим матрицу неточностей для одной команды поддержки, участниками которой являются 5 инженеров. Проанализировав рисунок 25, можно сделать вывод, что процент неточностей достаточно высок. Помимо инженеров здесь присутствуют значения Not in the Team, которые указывают на то, что некоторые инциденты могли закрываться без явного указания инженера или этот инженер, когда-то был участником команды поддержки. Так как данные берутся за несколько лет и кадровые перестановки являются неотъемлемой частью динамического бизнеса, нельзя очищать данные, которые могут содержать в себе важные связи, влияющие на обучение и точность предсказания.

Одной из важных характеристик матрицы неточностей с нормализованными значениями является тот факт, что сумма значений по горизонтали для инженера (фактические значения) равна 1 по каждой метке, но суммы по вертикали (предсказанные значения) обычно далеки от единицы. Это означает, что некоторые инженеры будут предсказываться чаще, чем другие, и это может создать общий дисбаланс для модели. Основываясь на этом, модель обучается, корректируя начальные веса, чтобы достичь суммы по вертикали, близкой к 1.0 (+/- 0.05). Таким образом, с точки зрения алгоритма, модель берет базовые веса и начинает их перераспределять, из цикла в цикл. Одним из условий является, что цикл имеет в среднем около 15 итераций, чтобы предотвратить бесконечное выполнение или излишнего переобучения модели.

Важными отличиями моделей предсказания подходящих инженеров от первой модели по определению команды являются:

- Для каждой модели будет строго предопределен словарь актуальных инженеров. Это позволит избежать ошибок в обучении и предсказании инженеров, которые уже не участвуют в деятельности команды поддержки.
- Задается словарь корректировочных весов для инженеров, которые участвовали в решении инцидентов, но теперь не участвуют в деятельности проектной команды (Not in the Team). Эти значения нельзя игнорировать, и они будут равномерно распределяться по весам

остальных, актуальных инженеров в процессе обучения и с помощью математических преобразований

```
modelDefineResolver=LogisticRegression(class_weight=dictweight,  
solver='saga',max_iter=10000, C=1, multi_class='multinomial').fit(X_train, res_train)
```

Рис. 13. Основная функция обучения модели по инженерам команды поддержки

Выше представлена общая функция обучения модели мультиномиальной логистической регрессии, которая имеет следующие параметры:

- `Class_weight = dictWeight` – подключения словаря весов для распределения весов неактуальных инженеров среди действующих инженеров команды поддержки
- `Solver='SAGA'` - нестандартный метод оптимизации алгоритма мультиномиальной логистической регрессии, повышающий скорость и точность обучения алгоритма
- `Max_iter=10000` – количество итераций прогонов при обучении алгоритма
- `C=1` – параметр регуляризации, который был подобран экспериментальным путем
- `Multi_class='multinomial'` – параметр отвечающий за выполнения не бинарной классификации, таким образом количество классов будет зависеть от количества инженеров команды поддержки.
- `Fit(X_train, res_train)` – обучение модели на тренировочных данных `X_train` и последующей тренировкой на тестовых данных `res_train`, которые до этого не участвовали в процессе обучения

После того, как каждая модель будет обучена для каждой команды поддержки по инженерам, полученные модели будут также сформированы в модульные файлы, которые будут подключаться на основе результатов модели определения команды поддержки. Структура обобщенной модели будет выглядеть следующим образом:



Рис. 14. Логическая схема улучшенной модели программного решения

Таким образом, улучшенная модель позволит определять инженеров соответствующей команды поддержки. При определении команды поддержки первой моделью (наиболее вероятную в списке предлагаемых), срабатывает механизм подключения соответствующей модели для предсказанной команды, и после ее выполнения программа выдаст базовое предсказание наиболее подходящих инженеров для решения инцидента.

В ходе эксперимента, автором исследования были обнаружены следующие недостатки:

- Несмотря на высокую точность первой модели по определению команды поддержки, самая вероятная команда может быть не правильно предсказана, что влечет за собой подключение неправильной модели по предсказанию инженеров.
- Базовое предсказание инженеров опирается только на статистические данные, на которых обучалась модель, не учитывая текущее состояние инженера и его доступность для решения инцидента.
- Автором исследования было предпринято решение по поиску оптимизации предсказанного значения, который будет максимально соответствовать действительности. Для этого были рассмотрены следующие направления по улучшению:
- Получение данных о состоянии инженера, посредством получения статуса в коммуникаторе Skype. Статусы: доступен, занят, не в сети имеют свои весовые характеристики, которые будут влиять на значение предсказания инженера. Эмпирическим путем была получена зависимость значения веса от времени отсутствия инженера на месте. Чем больше прошло времени согласно статусу ушел, тем меньше будет

результатирующий вес предсказания для выбора подходящего инженера.

- Получение данных о загрузке инженера. Данный коэффициент влияет на вес предсказания согласно проектной загрузке инженера в проектной работе. Для этого должно быть подсчитано количество инцидентов, назначенных на инженера, количество инцидентов в очереди команды.
- Полученное процентное соотношение также будет влиять на результирующий вес предсказания, чтобы избежать излишней нагрузки на одного инженера.

Автор исследования полагает, что принятые меры по оптимизации предсказанных результатов позволят получить наиболее точное предсказание в сравнении с базовым предсказанием. Ниже приведена таблица с учетом коэффициентов, которые влияют на вес итогового предсказания:

Табл. 5. Пример расчетов числовых характеристик для корректировки базового предсказания инженера команды поддержки

	Кол-во инцидентов	Коэффициент отклонения	Нормализованное значение коэффициента	Базовое предсказание	Предсказание*Загрузка	Нормализованный итоговый результат
Engineer1	8	-1.4	4.38	0.08	0.3504	5.908937605
Engineer2	2	4.6	10.38	0.03	0.3114	5.251264755
Engineer3	10	-3.4	2.38	0	0	0
Engineer4	10	-3.4	2.38	0.09	0.2142	3.612141653
Engineer5	10	-3.4	2.38	0.25	0.595	10.03372681
Engineer6	5	1.6	7.38	0.05	0.369	6.222596965
Engineer7	3	3.6	9.38	0.15	1.407	23.72681282

Как можно видеть из Таблицы 5, базовое предсказание указывало на Инженера 5 как на самого подходящего инженера для решения инцидента. С учетом применения корреляционных весов итоговый результат указывает на Инженера 7 как наиболее подходящего инженера для решения задачи в данный момент времени.

Таким образом было рассмотрено устройство модели принятия решения АИК с точки зрения построения принципов взаимодействия компонентов и ролей в инцидент менеджменте.

5. Вывод

В ходе выполнения работы было разработано программное решение по автоматизации деятельности одной из ролей в области управления ИТ-инфраструктурой. Были выдвинуты и изучены предложения по совершенствованию процессов принятия решений на основе методов машинного обучения, в частности метода логистической регрессии, проведены эксперименты и расчеты, показывающие качественное улучшение точности принятия решений.

Международные исследования в области ИТ обслуживания показывают, что с возрастанием объемов бизнеса и конечных пользователей, пропорционально возрастает и объем данных, которые должны обрабатываться большим количеством инженеров и координироваться компетентными людьми для предоставления качественного сервиса, поэтому в рамках диссертационного работы были изучены методы и алгоритмы машинного обучения, которые входят в основу программного решения по автоматизации деятельности Координатора инцидентов в рамках управления ИТ обслуживанием.

В данной работе спроектировано и создано программное решение на языке Python, которое содержит в себе интеллектуальную обученную модель принятия решения.

Были изучены и описаны процессы управления инцидент менеджмента на существующем проекте сервисного обслуживания. Рассмотрены этапы создания модели принятия решения,

обучаемые на входных данных, отличительные особенности обучения модели и ее параметризация, которая подбиралась опытным путем в рамках проведения экспериментов. Полученные экспериментальные данные показывают, что точность прогнозирования в принятии решения достигает 90%, при наименьших затратах времени.

При разработке программного решения по управлению ИТ-обслуживанием на основе методов машинного обучения были решены следующие задачи:

Углубленное изучение принципов ИТЛ как основополагающую методологию в сфере ИТ-обслуживания для поиска приоритетных проблем с целью их автоматизации

Проведен анализ роли координатора инцидентов в описательной модели сервисного ИТ обслуживания

Были сформулированы и проработаны предложения по совершенствованию управления ИТ-обслуживанием в контексте деятельности инцидент-координатора

Реализована модель принятия решения на основе методов логистической регрессии средствами среды разработки языка Python.

Литература

1. ЧЕРНИКОВ С.Ю., КОРОЛЬКОВ Р.В. *Использование системного анализа при управлении организациями* // Моделирование, оптимизация и информационные технологии. 2014. № 2 (5). С. 16.
2. САМОЙЛОВА У.А. *О некоторых характеристиках управления предприятием* // Вестник Воронежского института высоких технологий. 2014. № 12. С. 176-179.
3. ЗЯБЛОВ Е.Л., ПРЕОБРАЖЕНСКИЙ Ю.П. *Построение объектно-семантической модели системы управления* // Вестник Воронежского института высоких технологий. 2008. № 3. С. 029-030.

4. ИСАКОВА М.В., ГОРБЕНКО О.Н. *Об особенностях систем управления персоналом* // Вестник Воронежского института высоких технологий. 2014. № 12. С. 168-171.
5. КРАВЦОВ Д.О., ПРЕОБРАЖЕНСКИЙ Ю.П. *Методика оптимального управления социально-экономической системой на основе механизмов адаптации* // Вестник Воронежского института высоких технологий. 2008. № 3. С. 133-134.
6. ПАВЛОВА М.Ю. *Об использовании научной составляющей при формировании профессиональных качеств инженера* // Вестник Воронежского института высоких технологий. 2012. № 9. С. 144-145.
7. ПРЕОБРАЖЕНСКИЙ Ю.П. *Разработка методов формализации задач на основе семантической модели предметной области* // Вестник Воронежского института высоких технологий. 2008. № 3. С. 075-077.
8. ЕРМОЛОВА В.В., ПРЕОБРАЖЕНСКИЙ Ю.П. *Методика построения семантической объектной модели* // Вестник Воронежского института высоких технологий. 2012. № 9. С. 87-90.
9. ПРЕОБРАЖЕНСКИЙ Ю.П. *Оценка эффективности применения системы интеллектуальной поддержки принятия решений* // Вестник Воронежского института высоких технологий. 2009. № 5. С. 116-119.