

Федеральное государственное автономное образовательное
учреждение высшего образования
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ Н.Э.БАУМАНА»
(МГТУ им. Н.Э. Баумана)

Автоматизация обработки отчётов об ошибках в программном обеспечении с помощью больших языковых моделей

Студент:

Гуныко Н.М.

Научный руководитель:

Витюков Ф.А.

Москва, 2025



Постановка задачи

Цель работы – разработать программное решение для автоматизации обработки отчётов об ошибках (баг-репортов) в программном обеспечении (ПО) с использованием современных крупных языковых моделей и анализа их эффективности для оптимизации процессов технической поддержки.

Задачи:

- Исследовать и описать структуру баг-репортов, выделить основные компоненты, необходимые для эффективного анализа обращений.
- Выполнить сравнительный анализ современных крупных языковых моделей (GPT-4, Claude, DeepSeek и Grok) с целью выбора наиболее подходящей.
- Разработать архитектуру и программную реализацию автоматизирующей программы-прослойки (АПП), интегрированной с системой Intradesc.
- Провести тестирование и отладку программного решения на реальных обращениях.
- Проанализировать результаты тестирования, оценить стоимость и практическую применимость разработанного решения.

Структура баг-репорта

Баг-репорт представляет собой структурированную форму данных, используемую для фиксации и документирования информации об обнаруженных дефектах или ошибках в ПО.

Полная структура баг-репорта необходима для эффективного устранения ошибок:

- Позволяет точно воспроизвести проблему;
- Обеспечивает полноту данных для анализа;
- Минимизирует коммуникационные потери;
- Ускоряет исправление недочетов и тестирование ПО.



БАГ-РЕПОРТ

1. Заголовок

2. Шаги воспроизведения

3. Ожидаемое поведение

4. Фактическое поведение

5. Дополнительные материалы

6. Контекст выполнения

7. Приоритет и статус

Структура баг-репорта

Проблемы обработки баг-репортов

Существующие системы обработки обращений сталкиваются со следующими наиболее типичными **затруднениями**:

- Рост количества пользовательских баг-репортов в крупных системах.
- Низкое качество некоторых баг-репортов:
 - неполное описание возникшей проблемы;
 - отсутствие ключевой информации.
- Ручная проверка полноты обращений, а также уточнение информации занимает значительное время и ресурсы службы поддержки.

Актуальность

В современных условиях развития технологий открываются **новые возможности** для повышения эффективности обработки баг-репортов:

- Развитие больших языковых моделей (LLM) открывает возможности смыслового анализа текстов.
- Применение LLM позволяет автоматически выявлять недостающие ключевые элементы обращения и формулировать уточняющие комментарии.
- Актуально в условиях необходимости повышения эффективности поддержки и сокращения времени обработки заявок.

Большие языковые модели

Большая языковая модель (англ. Large Language Model, LLM) – это тип искусственного интеллекта, основанный на нейронных сетях, обученных на огромных объемах текстовых данных с целью обработки, понимания и генерации текстов на естественном языке.

Ключевые характеристики LLM:

- Большое число параметров (от сотен миллионов до триллионов), обеспечивающее высокий уровень обобщения.
- Обучение на разнообразных источниках естественного языка, включая документы, диалоги, код, статьи и др.
- Способность выполнять широкий круг задач: генерация текста, перевод, логический вывод, анализ структуры, семантический поиск, ответы на вопросы и др.

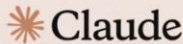
В данной работе использованы модели: GPT-4, Claude, DeepSeek и Grok.

Большие языковые модели

Используемые модели – это современные LLM, обученные на огромных объемах текстовых данных.



GPT - 4



deepseek



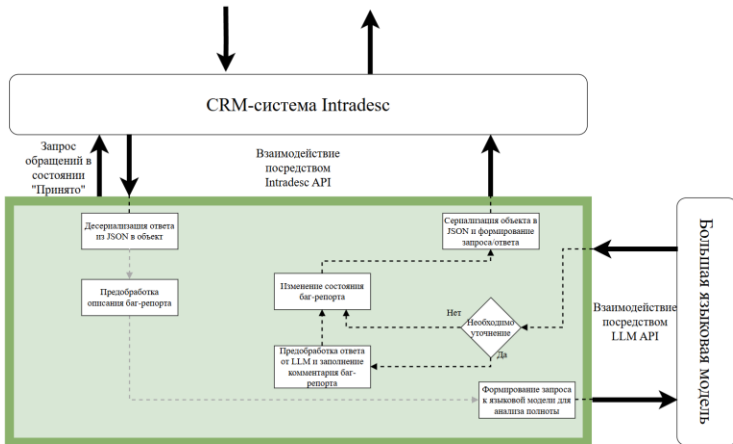
- Обладают высоким уровнем контекстного понимания и логического анализа.
- Подходят для задач семантической интерпретации и выявления неполноты в пользовательских баг-репортах.

Выбраны для исследования как одни из **наиболее продвинутых** и **востребованных** на рынке LLM.

Сравнение по характеристикам и метрикам

Модель	Архитектура	Параметры (оценка)	Контекст (токенов)	MMLU (знания, %)	HumanEval (код, %)	GSM8K (матем., %)	Творческие задачи	Эмоциональный интеллект	Доступность	Поддержка локально
GPT-4	Dense Transformer	~1-1.5 трлн	До 32.768	~92%	~85%	~90%	Очень высоко	Нейтрально-вежливый	Подписка + API	Нет
Claude	Dense Transformer	~100 млрд	До 100.000	~90%+	~83%	n/a	Очень высоко	Эмпатичный	Подписка + API	Нет
DeepSeek	Mixture-of-experts	671 млрд	~32.000	~90.8%	~85%	~90.2%	Умеренно (фактичность)	Сухой и прямой	Подписка + API	Да (DeepSeek-R1)
Grok	Mixture-of-experts	> 2 трлн	До 128.000	~92.7%	~86.5%	~89.3%	Высоко	Вежливый	Подписка	Частично (Grok-1)

Архитектура программного решения



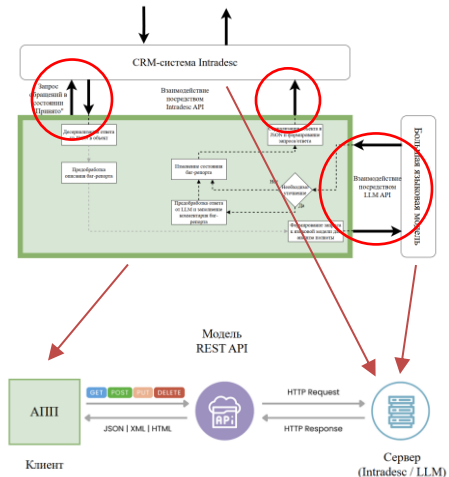
Реализация программных модулей

Взаимодействие АПП с внешними сервисами происходит через **REST API**. В качестве транспортного формата используется JSON, запросы отправляются по протоколу HTTP.

Основные взаимодействия:

- **Система Intradesc:**
 - Получение баг-репортов: GET /tickets?\$filter=...
 - Добавление комментариев: PUT /tickets/comment
 - Обновление статуса: PUT /tickets/update
- **Языковые модели (GPT-4, Claude, DeepSeek):**
 - Формирование запроса: POST /v1/chat/completions
 - Ответ в формате JSON содержит признаки полноты и комментарии.

Все HTTP-запросы обрабатываются централизованно через вспомогательный класс **HttpHelper**, поддерживающий отправку, сериализацию и логгирование обмена.



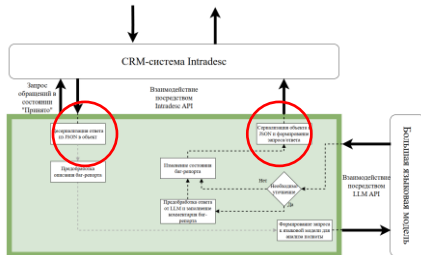
Реализация программных модулей

Десериализация – процесс преобразования JSON-данных в объекты C#:

- входящий JSON-ответ превращается в объекты, из которых можно извлекать поля вроде описания, статуса или даты обновления.

Сериализация – обратный процесс: преобразование объектов C# в JSON:

- применяется при отправке изменений – объекты автоматически превращаются в JSON, соответствующий API Intradesc.



```
1 {
2   "dataCount": 159,
3   "dataCountRelation": 8,
4   "value": {
5     {
6       "id": 280008,
7       "username": 28723,
8       "customerid": 5882,
9       "status": 2079,
10      "priority": 2944,
11      "priorityClass": null,
12      "priorityClass": null,
13      "name": "При формировании отчета по форме 301 в ряд. 6880 -у неизвестно определится срок для счетов 47427 из ИБС (мебанковское кредитование) раздела 2",
14      "description": "При формировании отчета по форме 301 в ряд. 6880 -у неизвестно определится срок для счетов 47427 из ИБС (мебанковское кредитование). В базе открыты счета БС: 32805, 33005 ( от 2 до 7 дней), а в счета 47427 для этих счетов попадает в срок своей даты ист.
```

Пример ответа Intradesc с баг-репортом в формате JSON

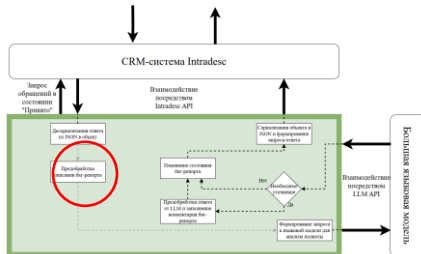
Реализация программных модулей

Предобработка обращения: очистка конфиденциальных данных

Перед отправкой обращения в языковую модель автоматически удаляются персональные данные (ИНН, счета, e-mail и др.).

Используются регулярные выражения для замены чувствительных фрагментов на маркеры (например: ИНН 7707083893 → ИНН [удалено]).

Это гарантирует защиту данных и соответствие требованиям безопасности.



Тип данных	Пример	Регулярное выражение
Телефон	+7 (495) 123-45-67 8 912 345 67 89 +375-29-1234567	(?:\+?\d{1,3})?[\s-]?\ (?:\d{3,4})?[\s-]?\ [\d{2,4}][\s-]?\ [\d{2,4}][\s-]?[\d{2,4}]
Email	user.name+test@example.co.uk	[a-zA-Z0-9._\+\-]+\@[a-zA-Z0-9-]+\.[a-zA-Z]{2,}
Паспорт	1234 567890 1234567890	\b\d{4}\s\d{6}\b
Банковская карта	1234 5678 9876 5432 1234567890123456	(?:\d{4})?(?:\d{4})?(?:\d{4})?(?:\d{4})?

```
public static class TextSanitizer
{
    private static readonly Dictionary<string, string> Patterns = new()
    {
        { @"(?:\d{10,12})(?:\d{1,3})", "[ИНН удален]" },
        { @"(?:\d{10,12})(?:\d{1,3})", "[ОГРН удален]" },
        { @"(?:\d{10,12})(?:\d{1,3})", "[СЧЕТ удален]" },
        { @"(?:\d{1,3})?[\s-]?\  
(?:\d{3,4})?[\s-]?\  
[\d{2,4}][\s-]?\  
[\d{2,4}][\s-]?[\d{2,4}]", "[ТЕЛЕФОН удален]" },
        { @"[a-zA-Z0-9._\+\-]+\@[a-zA-Z0-9-]+\.[a-zA-Z]{2,}", "[EMAIL удален]" },
        { @"(?:\d{4} )?(?:\d{4} )?(?:\d{4} )?(?:\d{4} )?", "[ПАСПОРТ удален]" },
        { @"(?:\d{4} )?(?:\d{4} )?(?:\d{4} )?(?:\d{4} )?", "[КАРТА удалена]" }
    };

    public static string Sanitize(string input)
    {
        foreach (var pattern in Patterns)
        {
            input = Regex.Replace(input, pattern.Key, pattern.Value);
        }
        return input;
    }
}
```

Реализация программных модулей

После очистки обращения от конфиденциальных данных формируется **текстовый промпт**, содержащий четкие инструкции для языковой модели.

```

Ты - ассистент по качеству баг-репортов в службе технической поддержки.

Твоя задача - проверить баг-репорт ниже и определить, все ли обязательные
элементы в нём присутствуют. Если каких-то пунктов не хватает, необходимо
сгенерировать вежливый, но деловой комментарий для автора обращения с
просьбой добавить недостающие сведения.

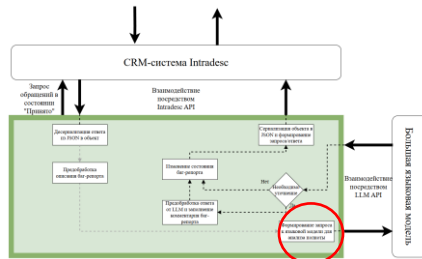
Проверь обращение на наличие следующих пунктов:
1. Что делал пользователь?
2. Что он ожидал?
3. Что произошло на самом деле?
4. Почему это может происходить? (необязательный пункт, is_complete не
зависит от наличия этого пункта)
5. Достаточно ли данных для воспроизведения ошибки?
Если пункт явно не сформулирован, но можно логически понять его суть из
контекста - считай, что он присутствует.

Ответ верни в следующем JSON-формате:
{
  "is_complete": true/false,
  "missing_parts": ["<названия отсутствующих пунктов>"],
  "comment_to_author": "<сформулированный текст комментария оператору>",
  "next_action": "ok" или "request_additional_info"
}

Правила для comment_to_author:
- Комментарий должен быть адресован автору баг-репорта.
- Просьбы о доработке должны быть чёткими и конкретными.
- Не пересказывай баг-репорт, не объясняй суть ошибки.
- Просто напиши, что именно нужно уточнить или добавить, чтобы можно было
воспроизвести проблему.

Текст обращения:
""{ticketText}""
    
```

Промпт к большой языковой модели

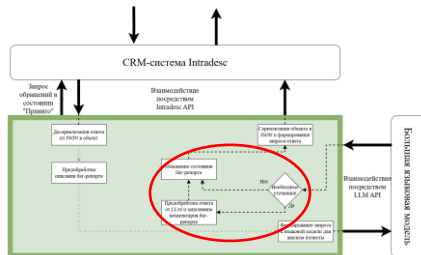


Реализация программных модулей

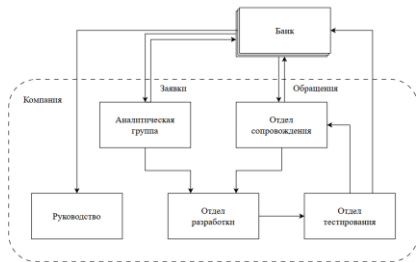
После получения ответа от языковой модели система автоматически анализирует результат:

- Если все необходимые элементы в баг-репорте присутствуют
 - задача переводится в рабочий статус
 - данные структурируются и сохраняются
- Если отчёт неполный
 - формируется комментарий с просьбой уточнить недостающую информацию
 - задача переводится в статус «Передано на уточнение»

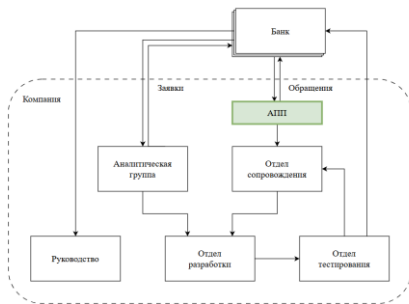
Таким образом, реализована автоматическая обработка с возможностью привлечения оператора при необходимости.



Место программного решения в компании



До внедрения



После внедрения

Баг-репорт №1 для тестирования

№ 469 ×

Номер обращения 114519 Fwd[2]: 113888 Расчет сумм исковых заявлений (Справка)

Поддержка > Обращение


Описание

Расчет сумм исковых заявлений (Справка)
необходимо исправить две ошибки

1. Отображение суммы по просроченным процентам в справку.
2. Исправление сумм в колонке 8 и 7 в разделе "Задолженность по основному долгу"

описание моделирования на базе сопровождения во вложениях.

скрыть

 ошибка.docx

Договор в нашу базу положил
Из р 46885
Отчет строки на 20 09 2022
Договор выдавался 25 05 2022 первый и второй Иплатежи ушли и апросчку, потом чуть чуть
погасили, потом опять просрочка, суть гашение и 13 08 полный вынос а просрочку и после
этого чуть гашений

1. К вопросу о процентах
субсчет по процентам по ОЗ — они только выносились на просрочку

Субсчет - Проценты

Дата	Дебет	Кредит	Остаток
		0.00	
31 05 2022	30.14		30.14
30 06 2022	75.74		105.88
01 08 2022	71.99		177.87
13 08 2022	27.74		205.61
Итого:	205.61	0.00	205.61

из картинки видно что сумма там не только процентно по ОЗ но и с добавками
процентно по ПЗ Во второй строке указано 82.19 = 75.74 (проценты по ОЗ) + 6.45
(проценты по ПЗ)
во вложении файл: с движением средств по договору
Т.е. блок показывает что надо брать проценты по ПЗ, но не берет их полсе того как
заканчивается график — 13 08 2022 А они там есть и точно также копятся, гасятся и
выносятся на просрочку

Результаты работы моделей с баг-репортом №1



GPT - 4

```
{  
  "is_complete": true,  
  "missing_parts": [],  
  "comment_to_author": "",  
  "next_action": "ok"  
}
```



```
{  
  "is_complete": false,  
  "missing_parts": ["Что делал пользователь?", "Что он ожидал?", "Что  
произошло на самом деле?", "Достаточно ли данных для воспроизведения  
ошибки?"],  
  "comment_to_author": "Уточните, пожалуйста, какие действия вы выполняли  
перед возникновением ошибки. Опишите, какой результат ожидали получить и  
что произошло вместо этого. Также проверьте, достаточно ли данных для  
воспроизведения ошибки (например, указаны ли все необходимые шаги или  
параметры).",  
  "next_action": "request_additional_info"  
}
```



```
{  
  "is_complete": true,  
  "missing_parts": [],  
  "comment_to_author": "",  
  "next_action": "ok"  
}
```

Баг-репорт №2 для тестирования

← № 582 ×

> В ДБО открываю вклад, ид типового вклада 103.

Поддержка > Задача

Описание

В ДБО открываю вклад, ид типового вклада 103.

Получаю ошибку: Открытие депозитного договора. Договор 'До востребования, для Мишиных договоров'. Ставка 12 договора не соответствует расчетной! В типовом сказано что должен открыться договор до востребования под счет уплаты % при закрытии.

Параметры которые передает ДБО в функцию Oper_Open_OutSystem такие:

OutSystem : True ;
DateTrn : 30.09.2022 9:34:17 ;
DateOpen : 30.09.2022 9:34:17 ;
IdCurrency : 810 ;
Балансовый номер : ;
NumDivision : 6 ;
Amount : 1000 ;
IdGroupContract : 103 ;
IdClient : 22365 ;
Period : 1 ;
Type : в годах ;
OIContract : 66 ;
OIAccounts : 44 ;
Объявленная ставка процентов : 12 ;
DESCRIPTION : Зачисление суммы для открытия вклада (безналично, со счета) ;
Дополнительные свойства : IUBsParam ;
ContractProperty : | Источник создания | RB |

Результаты работы моделей с баг-репортом №2



GPT - 4

```
{  
  "is_complete": false,  
  "missing_parts": ["Что он ожидал?", "Почему это может происходить?"],  
  "comment_to_author": "Пожалуйста, уточните, какого результата вы ожидали при открытии вклада. Также, если есть предположения о причинах возникновения ошибки, просьба их указать – это поможет в более быстром анализе ситуации.",  
  "next_action": "request_additional_info"  
}
```



```
{  
  "is_complete": true,  
  "missing_parts": [],  
  "comment_to_author": "",  
  "next_action": "ok"  
}
```



```
{  
  "is_complete": true,  
  "missing_parts": [],  
  "comment_to_author": "",  
  "next_action": "ok"  
}
```

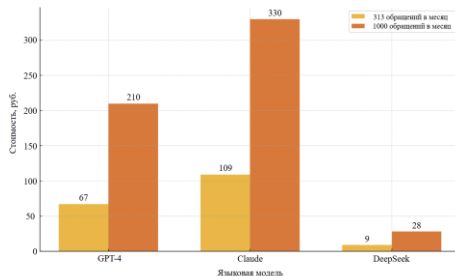
Оценка результатов анализа ответов моделей

Критерий оценки	GPT-4	DeepSeek	Grok
Корректность определения полноты	Высокая	Средняя	Средняя
Точность выявления недостающих пунктов	Высокая	Средняя	Средняя
Ясность и конкретность комментария	Отличная	Средняя	Хорошая
Практическая применимость	Высокая	Средняя	Средняя
Общая эффективность	Отличная	Хорошая	Хорошая

Оценка стоимости использования моделей

Модель	Вход (руб. / 1 млн. токенов)	Выход (руб. / 1 млн. токенов)	Примерная стоимость в месяц (руб.)
GPT-4	~200	~800	~67
Claude	~300	~1500	~109
DeepSeek	~28	~112	~9
Grok	—	—	~1500-2700 (подписка)

Примечание – Оценка дана в рублях по среднему курсу и тарифам на 2025 год.



Выводы

- Разработано решение для автоматизации обработки баг-репортов с применением LLM, интегрированное с системой Intradesk.
- Проведён сравнительный анализ моделей GPT-4, Claude, DeepSeek и Grok – выбрана наиболее эффективная по соотношению точности и стоимости.
- Подтверждена практическая применимость решения: снижение ручной нагрузки и ускорение обработки заявок в технической поддержке.