System Design

Бизнес-проблема

- **Проблема:** Компания, занимающаяся продажей бытовой техники, сталкивается с большим количеством отзывов на различных платформах (сайт, маркетплейсы, социальные сети). Ручная обработка отзывов занимает много времени, что замедляет реакцию на проблемы клиентов и снижает удовлетворённость пользователей.

Формализация проблемы

- **Цель:** Разработать систему автоматической классификации и анализа отзывов, которая будет:
  - Определять тональность отзывов (положительный, отрицательный, нейтральный).
  - Выявлять ключевые темы в отзывах (например, "проблемы с доставкой", "качество товара").
  - Предоставлять аналитику для бизнеса, включая визуализацию трендов (например, рост или снижение количества негативных отзывов, популярные темы за определённый период).
- **Объект:** Отзывы клиентов на платформах.
- **Приоритет:** Высокий, так как оперативное реагирование на отзывы улучшает лояльность клиентов и репутацию бренда.
- **Ограничения:**
  - **Бюджет:** Ограниченный бюджет на разработку и внедрение системы.
  - **Сроки:** Проект должен быть завершён в течение 3 месяцев (28.01.2025 – 28.04.2025).
  - **Технические ограничения:**
    - Необходимость интеграции с множеством платформ (сайт, маркетплейсы, соцсети).
    - Ограниченная вычислительная мощность для обработки отзывов в реальном времени.
  - **Законодательные ограничения:** Необходимость соблюдения GDPR (если отзывы содержат персональные данные).
- **Риски:**
  - **Недостаток данных:**
    - Отзывы на новые продукты могут быть немногочисленными, что затруднит обучение модели. Решение: использование предобученных моделей (например, RuBERT для русского языка) или аугментация данных.
    - Данные могут быть несбалансированными (например, больше положительных отзывов, чем отрицательных).
  - **Низкое качество данных:**
    - Отзывы могут содержать опечатки, сленг, эмодзи или сокращения, что усложнит обработку текста. Решение: добавление этапа предобработки (очистка текста, лемматизация).
  - **Низкая точность модели:**
    - Модель может ошибаться в классификации из-за сложности текста (например, сарказм, ирония).

        - Модель может недостаточно точно выявлять темы в отзывах. Решение: использование современных методов, таких как BERTopic или Top2Vec.
    - **Технические сложности:**
        - Задержки в обработке отзывов в реальном времени из-за ограниченной вычислительной мощности. Решение: использование облачных сервисов (AWS, Google Cloud).
        - Проблемы с интеграцией модели с CRM-системой и другими платформами.
    - **Изменение поведения клиентов:**
        - Со временем стиль и содержание отзывов могут измениться, что потребует регулярного обновления модели. Решение: внедрение механизма мониторинга и дообучения модели.

Тип формальной задачи

Это задача многоклассовой классификации текста с дополнительным анализом тем (topic modeling).
Классификация помогает определить тональность отзыва, а анализ тем — выявить, о чём именно пишут клиенты. Вместе они позволяют получить полную картину обратной связи от клиентов.

- **Многоклассовая классификация текста**
    - **Описание:** Задача заключается в том, чтобы классифицировать каждый отзыв на одну из нескольких категорий:
        - Положительный отзыв.
        - Отрицательный отзыв.
        - Нейтральный отзыв.
    - **Особенности:**
        - Входные данные: текстовые отзывы.
        - Выходные данные: категория отзыва (класс).
        - Метрики оценки: accuracy, precision, recall, F1-score, ROC-AUC (для несбалансированных данных).
    - **Пример алгоритмов:**
        - Логистическая регрессия (Logistic Regression).
        - Метод опорных векторов (SVM).
        - BERT (трансформеры для обработки текста).
    - Современные архитектуры, такие как GPT, RoBERTa или DistilBERT.
- **Анализ тем (Topic Modeling)**
    - **Описание:** Задача заключается в том, чтобы выявить ключевые темы, которые часто встречаются в отзывах:
        - Проблемы с доставкой.
        - Качество товара.
        - Работа службы поддержки.
    - **Особенности:**
        - Входные данные: текстовые отзывы.
        - Выходные данные: набор тем и ключевые слова для каждой темы.

        - Метрики оценки: perplexity, coherence score (более важный для интерпретируемости тем).
    - **Пример алгоритмов:**
        - LDA (Latent Dirichlet Allocation).
        - NMF (Non-Negative Matrix Factorization).
    - Современные методы, такие как BERTopic или Top2Vec, которые используют эмбеддинги для более точного выявления тем.

Приемлемость применения машинного обучения

- ML может быть эффективно применен для решения этой задачи, так как:
    - **Достаточно данных:** Есть исторические данные отзывов клиентов на различных платформах (сайт, маркетплейсы, социальные сети).
    - **Наличие закономерностей:** Отзывы содержат определённые паттерны, которые можно выявить (например, частота упоминания проблем с доставкой, качеством товара и т.д.).
    - **Возможность автоматизации:** ML позволяет автоматически классифицировать отзывы, выявлять ключевые темы и предоставлять аналитику для бизнеса, что значительно ускоряет обработку обратной связи.
    - **Потенциал для улучшения клиентского опыта:** Анализ отзывов позволяет быстро реагировать на проблемы клиентов, улучшая их удовлетворённость и лояльность к бренду.

Система решения

Этап I: Постановка задачи и создание концепции

- **Цель:** Сформулировать основные требования и создать концептуальную модель системы.
- **Основные шаги:**
    - Выявить ключевые потребности бизнеса, включая сокращение времени обработки отзывов и увеличение точности анализа.
    - Провести предварительную оценку ожидаемого эффекта, например, снижение времени реакции на 50%.
    - Спроектировать концептуальную схему системы, указав основные компоненты, такие как базы данных, модули интеграции и аналитики.
- **Результат:** Концепция системы с указанием её ключевых компонентов и функционала.

Этап II: Сбор и обработка данных

- **Цель:** Подготовить данные для анализа и обучения моделей.
- **Основные шаги:**
    - Сбор отзывов с различных источников (социальные сети, маркетплейсы, корпоративный сайт).
    - Очистка данных от шума (дубликаты, пропуски, опечатки).
    - Нормализация текста (лемматизация, токенизация) для упрощения обработки.
    - Разделение данных на тренировочные и тестовые выборки для обучения и валидации.

- **Результат:** Чистый, структурированный и готовый к использованию набор данных.

Этап III: Создание и тестирование модели

- **Цель:** Разработать алгоритмы анализа отзывов.
- **Основные шаги:**
  - Подобрать подходящие алгоритмы для определения тональности и тематического анализа (например, трансформеры или LDA).
  - Провести обучение моделей на подготовленных данных.
  - Оценить эффективность модели с использованием ключевых метрик (F1-score, accuracy).
- **Результат:** Обученные модели с высокими показателями точности и интерпретируемости.

Этап IV: Интеграция с бизнес-процессами

- **Цель:** Обеспечить использование разработанной системы в операционной деятельности компании.
- **Основные шаги:**
  - Разработка API для автоматической обработки данных и взаимодействия с CRM.
  - Настройка инструментов визуализации, чтобы наглядно представлять результаты анализа.
  - Создание системы оповещений для быстрого реагирования на негативные отзывы.
- **Результат:** Рабочая система, интегрированная с операционной деятельностью компании.

Этап V: Мониторинг и развитие системы

- **Цель:** Обеспечить стабильную работу системы и её адаптацию к изменениям.
- **Основные шаги:**
  - Постоянно отслеживать качество анализа, используя данные в реальном времени.
  - Настроить автоматическое дообучение моделей при накоплении новых данных.
  - Обеспечить регулярный сбор обратной связи от пользователей системы.
- **Результат:** Постоянно развивающаяся система, адаптированная к текущим и будущим потребностям бизнеса.
- **Цель:** Обеспечить стабильную работу системы и её постоянное улучшение.
- **Задачи:**
  - Мониторинг метрик качества модели.
  - Регулярное обновление модели для учёта изменений в поведении клиентов.
  - Сбор обратной связи от команды и корректировка модели.
- **Результат:** Стабильно работающая система, адаптируемая к изменениям.

Система с точки зрения ООП

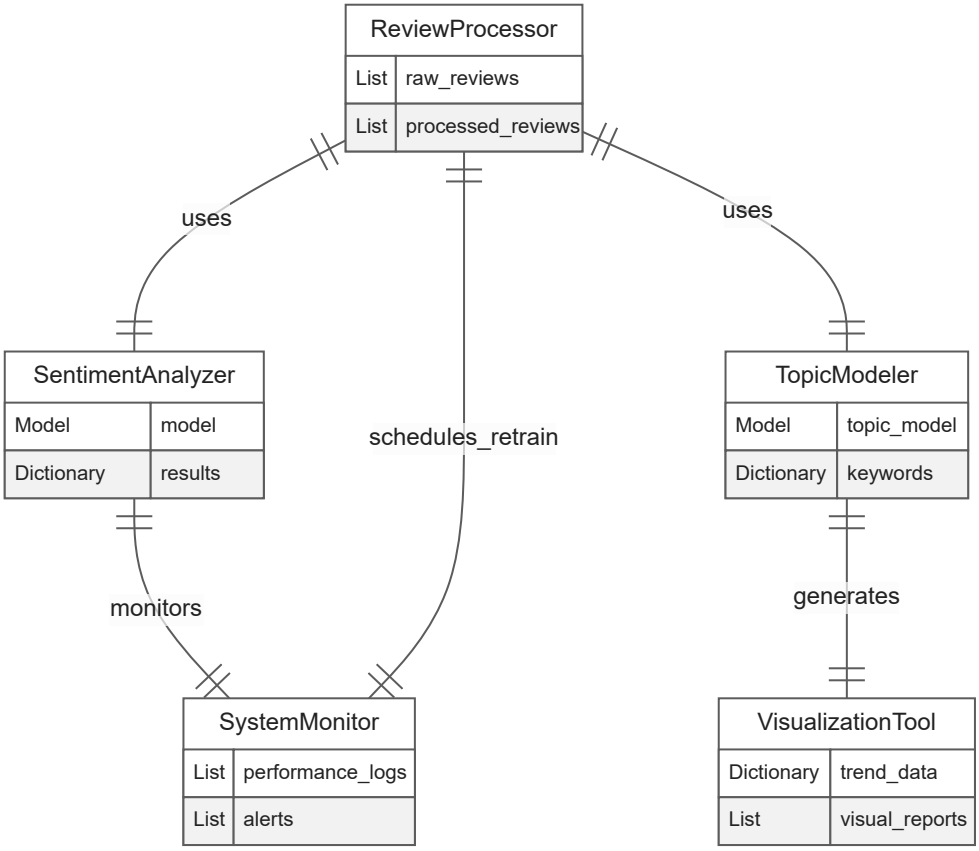
Основные классы системы:

1. **ReviewProcessor** (Обработчик отзывов)
   - **Атрибуты:**
     - `raw_reviews` (список сырых отзывов, полученных с различных источников).
     - `processed_reviews` (список очищенных и нормализованных отзывов).

- **Методы:**
  - `load_reviews(sources)` : загружает отзывы с заданных источников.
  - `clean_reviews()` : выполняет очистку текста (удаление шумов, нормализация).
  - `prepare_reviews()` : выполняет лемматизацию и токенизацию текста.

2. **SentimentAnalyzer** (Анализатор тональности)
   - **Атрибуты:**
     - `model` (обученная модель анализа тональности).
     - `results` (результаты анализа, сгруппированные по категориям).
   - **Методы:**
     - `predict_sentiment(text)` : анализирует тональность конкретного отзыва.
     - `analyze_batch(texts)` : проводит тональный анализ для набора отзывов.
3. **TopicModeler** (Модуль анализа тем)
   - **Атрибуты:**
     - `topic_model` (обученная модель анализа тем).
     - `keywords` (ключевые слова для каждой темы).
   - **Методы:**
     - `generate_topics(reviews)` : выполняет анализ тем по набору отзывов.
     - `update_model(new_data)` : обновляет модель при поступлении новых данных.
4. **VisualizationTool** (Инструмент визуализации)
   - **Атрибуты:**
     - `trend_data` (данные для построения временных трендов).
     - `visual_reports` (готовые визуализации для отчетов).
   - **Методы:**
     - `plot_trends(metric)` : строит графики изменений метрик во времени.
     - `display_topics_chart()` : визуализирует ключевые темы в виде диаграмм.
5. **SystemMonitor** (Система мониторинга)
   - **Атрибуты:**
     - `performance_logs` (логи работы модели).
     - `alerts` (уведомления о сбоях).
     - **Методы:**
     - `check_performance()` : анализирует эффективность работы модели.
     - `trigger_alert(message)` : отправляет уведомления при сбоях или падении качества.
     - `schedule_retrain()` : планирует переобучение модели.

ER-диаграмма системы



BPMN-схема процессов

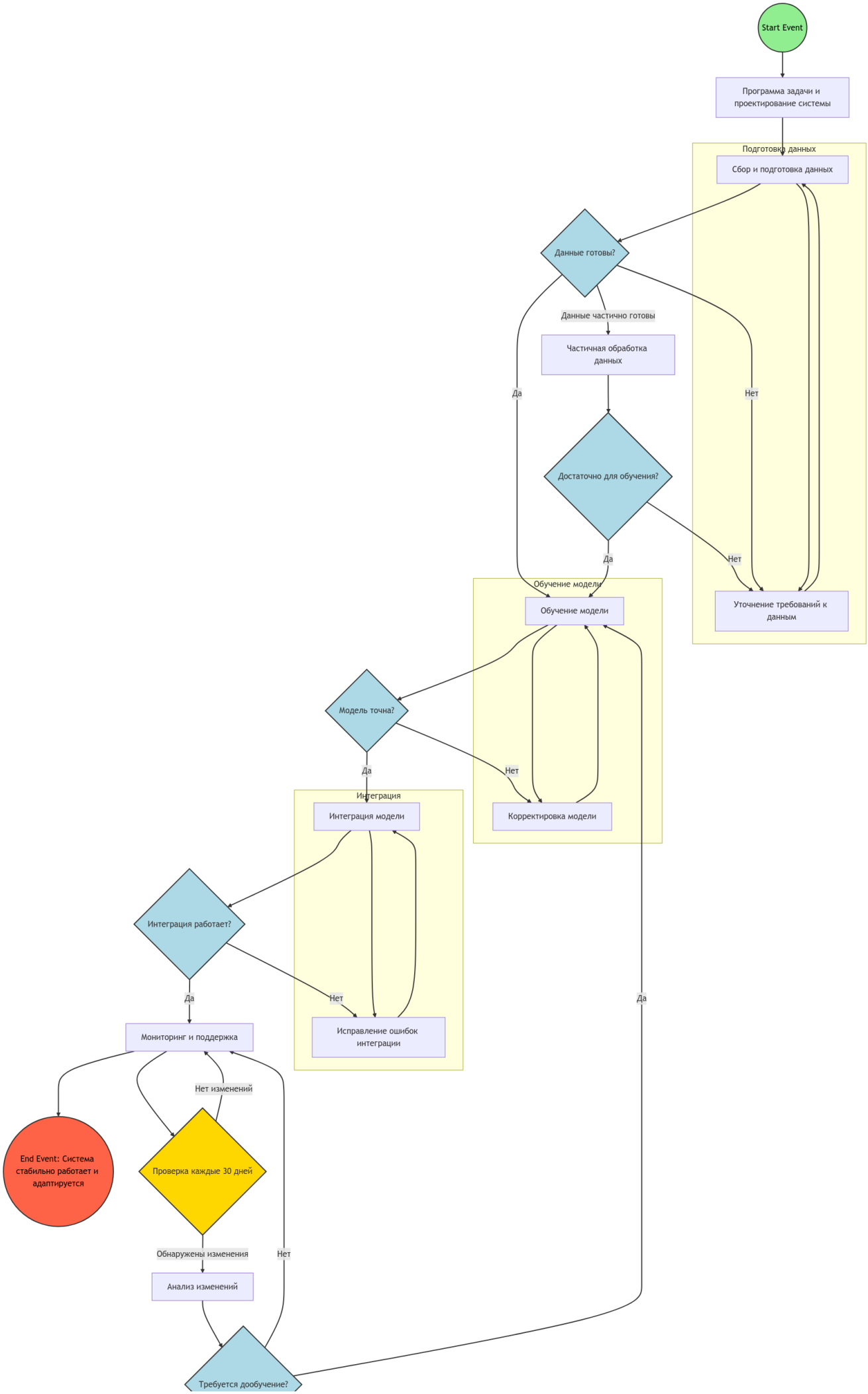
```
flowchart TD

    A((Start Event)) --> B[Программа задачи и проектирование системы]

    B --> C[Сбор и подготовка данных]

    C --> D{Данные готовы?}

    D -- Нет --> C1[Уточнение требований к данным]

    C1 --> C

    D -- Да --> E[Обучение модели]

    E --> F{Модель точна?}

    F -- Нет --> E1[Корректировка модели]

    E1 --> E

    F -- Да --> G[Интеграция модели]
```

```
G --> H{Интеграция работает?}

H -- Нет --> G1[Исправление ошибок интеграции]

G1 --> G

H -- Да --> I[Мониторинг и поддержка]

I --> J((End Event: Система стабильно работает и адаптируется))



%% Добавление параллельных процессов

subgraph Подготовка данных

    C --> C1

end



subgraph Обучение модели

    E --> E1

end



subgraph Интеграция

    G --> G1

end



%% Добавление событий-таймеров

I --> K{Проверка каждые 30 дней}

K -- Нет изменений --> I

K -- Обнаружены изменения --> L[Анализ изменений]

L --> M{Требуется дообучение?}

M -- Да --> E
```

```
M -- Нет --> I

%% Добавление альтернативных путей
D -- Данные частично готовы --> N[Частичная обработка данных]
N --> O{Достаточно для обучения?}
O -- Да --> E
O -- Нет --> C1

%% Улучшение визуализации
style A fill:#90EE90,stroke:#333,stroke-width:2px
style J fill:#FF6347,stroke:#333,stroke-width:2px
style D fill:#ADD8E6,stroke:#333,stroke-width:2px
style F fill:#ADD8E6,stroke:#333,stroke-width:2px
style H fill:#ADD8E6,stroke:#333,stroke-width:2px
style K fill:#FFD700,stroke:#333,stroke-width:2px
style M fill:#ADD8E6,stroke:#333,stroke-width:2px
style O fill:#ADD8E6,stroke:#333,stroke-width:2px

%% Добавление иконок для событий
style A stroke:#333,stroke-width:2px,fill:#90EE90,color:#000
style J stroke:#333,stroke-width:2px,fill:#FF6347,color:#000

%% Улучшение стрелок
linkStyle default stroke:#333,stroke-width:2px
```

```
    linkStyle 0 stroke:#333,stroke-width:2px

    linkStyle 1 stroke:#333,stroke-width:2px

    linkStyle 2 stroke:#333,stroke-width:2px

    linkStyle 3 stroke:#333,stroke-width:2px

    linkStyle 4 stroke:#333,stroke-width:2px

    linkStyle 5 stroke:#333,stroke-width:2px

    linkStyle 6 stroke:#333,stroke-width:2px

    linkStyle 7 stroke:#333,stroke-width:2px

    linkStyle 8 stroke:#333,stroke-width:2px

    linkStyle 9 stroke:#333,stroke-width:2px

    linkStyle 10 stroke:#333,stroke-width:2px

    linkStyle 11 stroke:#333,stroke-width:2px

    linkStyle 12 stroke:#333,stroke-width:2px

    linkStyle 13 stroke:#333,stroke-width:2px
```

В Obsidian не вмещается BPMN-схема целиком. Для более детального просмотра BPMN-схемы можно воспользоваться Mermaid Live Editor.


Start Event
Программа задачи и проектирование системы
Подготовка данных
Сбор и подготовка данных
Данные готовы?
Данные частично готовы
Частичная обработка данных
Достаточно для обучения?
Да
Нет
Да
Нет
Да
Нет
Уточнение требований к данным
Обучение модели
Обучение модели
Модель точна?
Да
Нет
Корректировка модели
Интеграция
Интеграция модели
Интеграция работает?
Да
Нет
Исправление ошибок интеграции
Да
Мониторинг и поддержка
Нет изменений
End Event: Система стабильно работает и адаптируется
Проверка каждые 30 дней
Обнаружены изменения
Нет
Анализ изменений
Требуется дообучение?