

Прототип системы автоматического анализа изменений кода

ИТЕРАЦИЯ 1

Команда:

1. Беннер Виктория 3388
2. Титкова Софья 3388
3. Снигирев Александр 3388
4. Лутфулин Даниил 3388
5. Дубровин Данила 3388
6. Чернуха Владимир 1310

Проблема и решение

1

Проблема

В процессе коллективной разработки или обучения программированию проверка кода занимает значительное время.

Преподаватели вынуждены вручную выявлять повторяющиеся ошибки. Это создает задержки в рабочем процессе и замедляет получение обратной связи.

2

Решение

Разработка сервиса, который автоматически анализирует изменения в коде с помощью локальной языковой модели и предоставляет предварительные рекомендации по исправлению ошибок.

Категории пользователей и их потребности



Разработчик (обучающийся)

заинтересован в получении быстрой проверки своих правок до передачи кода человеку. Это позволяет оперативно устранять недочеты.



Проверяющий (преподаватель)

заинтересован в снижении нагрузки на проверку типовых ошибок и получении аналитики по успеваемости группы/потока и тд.

Сценарии использования

Сценарий 1 — Проверка кода по запросу

- **Основной сценарий:**
 - Пользователь создает Pull Request (PR).
 - Пользователь запускает проверку PR нажатием кнопки.
 - Система считывает файлы и передает их аналитическому модулю.
 - Система публикует комментарии с замечаниями к коду.
- **Альтернативный сценарий:**
 - Если объем кода слишком велик, система сообщает о невозможности анализа из-за превышения лимита символов.
 - Если сервер с моделью недоступен, система выводит сообщение о временном техническом сбое.

Сценарий 2 — Мониторинг качества

- **Основной сценарий:**
 - Администратор открывает панель управления.
 - Администратор выбирает объект анализа: конкретный запрос (PR) или сводку по группе.
 - Система выводит отчет по количеству проверок, числу замечаний и классификатор типов ошибок.
- **Альтернативный сценарий:**
 - При отсутствии данных за выбранный период система уведомляет, что статистика не найдена.

План текущей и следующей итерации

План текущей итерации:

Создание макетов интерфейса: Визуальное проектирование панелей управления.

Вики-страница с информацией по проекту: Подготовка основного раздела документации с описанием целей, архитектуры и сценариев использования.

Создание прототипа серверной части: Разработка базовой логики обработки запросов.

Создание прототипа фронтенд-части: Подготовка структуры к верстке.

Настройка окружения (Docker): Написание Dockerfile и docker-compose для контейнеризации сервисов.

Интеграция: Настройка связи через localtunnel и создание прототипа GitHub Actions.

План следующей итерации :

Инфраструктура и данные:

- Проектирование и инициализация схемы базы данных.
- Настройка системы миграций для управления структурой БД.
- Подготовка инструкции по развертыванию системы.

Backend-разработка:

- **Модуль Gatekeeper:** Прием входящих событий (webhook) и их постановка в очередь брокера.
- **Модуль Consumer:** Обработка задач из очереди, интеграция с LLM и автоматическая публикация результатов в GitHub.

Frontend-разработка:

- Верстка пользовательского интерфейса по утвержденным макетам.

Сборка и запуск:

- Создание Docker-образов для всех компонентов.
- Сборка первого рабочего билда для демонстрации полного цикла работы системы.

Результаты текущей итераций

Результаты текущей итерации — что сделано

Проектирование интерфейсов:

- Разработаны макеты экранов, определяющие логику переходов и состав элементов управления.

Аналитика и сценарии:

- Описан основной сценарий проверки кода и альтернативные пути (реакция системы на технические сбои и отсутствие ошибок).
- Определены ключевые действующие лица: Пользователь и Администратор.

Техническая база:

- Подготовлен прототип фронтенд-части системы.

Сформирован файл `.gitignore` для корректного ведения репозитория.

- Выбран и зафиксирован технологический стек: Python, Go, React/Vue.js, PostgreSQL, RabbitMQ и Docker.
-

Макет интерфейса



Wiki страница

Home

Vladimir Chernukha edited this page 1 hour ago · [3 revisions](#)

Вики страница проекта 15 - Код ревью через llm

[Постановка задачи и требования](#)

[Сценарии использования](#)

+ Add a custom footer

Edit

New page

Pages 3

Find a page or section...

Home

Информация по проекту

Требования и сценарии использ...

Edit

New page

Требования и сценарии использования

3388_Беннер_Виктория_Александровна edited this page now · [4 revisions](#)

Прототип системы автоматического анализа изменений кода с помощью искусственного интеллекта

1. Описание проблемы

В процессе коллективной разработки или обучения программированию проверка кода (рецензирование) занимает значительное время. Преподаватели вынуждены вручную выявлять повторяющиеся ошибки, такие как нарушение стандартов оформления, отсутствие обработки исключений или избыточность логики. Это создает задержки в рабочем процессе и замедляет получение обратной связи.

Решение: Разработка сервиса, который автоматически анализирует изменения в коде с помощью локальной языковой модели и предоставляет предварительные рекомендации по исправлению ошибок.

2. Анализ требований

Функциональные требования:

- Интеграция с GitHub:** Получение уведомлений о новых изменениях (PR) и командах пользователя.
- Интеллектуальный анализ:** Сопоставление кода с базовыми правилами чистого кода и поиск потенциальных ошибок с помощью ИИ.
- Обратная связь в интерфейсе:** Публикация результатов анализа в виде точечных комментариев к строкам кода.
- Статистический учет:** Классификация ошибок и хранение истории проверок для формирования отчетов.
- Сбор оценок:** Механизм подтверждения или опровержения советов ИИ пользователем.

Pages 3

Find a page or section...

Home

Информация по проекту

Требования и сценарии использ...

Прототип системы автоматического анализа изменений кода с помощью искусственного интеллекта

- Описание проблемы
- Анализ требований
Функциональные требования:
Нефункциональные требования:
- Категории пользователей и их потребности
- Макет интерфейса
- Сценарии использования
Сценарий 1: Проверка кода по запросу
Сценарий 2: Мониторинг качества и сбор статистики
- Описание интерфейсов
Интерфейс взаимодействия
Интерфейс мониторинга (Панель преподавателя /