

УДК 004.9

РАЗРАБОТКА ИНСТРУМЕНТОВ ПО СОПРОВОЖДЕНИЮ ПРОЦЕССА РЕЦЕНЗИРОВАНИЯ ПРОГРАММНОГО КОДА РАЗРАБОТЧИКОВ

Троегубов О.А., студент гр. ПИб-212, IV курс

Научный руководитель: Веревкин С.А, старший преподаватель.

Кузбасский государственный технический университет
имени Т.Ф. Горбачева, г. Кемерово

Аннотация: в статье рассматривается важный этап в разработке программного обеспечения – процесс рецензирования программного кода разработчиков (Code review). Представлено описание проекта и набор технологий для его реализации. Описываются реализованные функции и дальнейшие шаги по разработке проекта.

Введение

Прежде чем говорить о процессе рецензирования кода, важно понимать, как организована работа с кодом в командах. Для этого используется система контроля версий, например, Git, и платформы для совместной разработки, такие как GitHub, GitLab, BitBucket.

Также необходимо разобрать следующие понятия:

- **Репозиторий** (Repository) – хранилище кода проекта. Он хранит файлы и историю изменений.
- **Ветка** (Branch) – изолированная версия кода. Обычно выделяют главную ветку для хранения стабильной версии проекта. Для разработки нового функционала создают копию стабильной версии проекта с другим названием ветки.
- **Запрос на изменение** (Pull request) – после окончания разработки создается запрос на добавления изменений в основную ветку. Запрос может иметь несколько статусов: открытый, отклоненный, закрытый.
- **Ревьювер** – разработчик, который проверяет чужой код.

Процесс рецензирования кода – процесс проверки исходного кода перед его слиянием в основную ветку. После окончания этапа разработки выполняемой задачи, один или несколько разработчиков анализируют код коллеги, оставляют замечания, предлагают улучшения. После одобрения код сливается в основную ветку. Ключевые показатели, улучшаемые за счет введение практик рецензирования кода:

- **Качество кода:** обнаружение ошибок во время этапа разработки, соблюдение стандартов команды.
- **Обмен знаниями и опытом.**
- **Экономия времени и ресурсов:** дешевле исправить ошибку на этапе

разработки, чем после релиза.

Описание проекта

Для повышения удобства сопровождения процесса рецензирования кода было предложено создание Telegram бота. Функционал будет реализован с помощью команд бота и внутренней логики приложения [1]. Бот должен иметь следующий функционал:

- Добавление сервера Git для инициирующей команду группы.
- Добавление репозитория из доступных серверов Git у группы для опрашивания новых запросов на изменение.
- Запрос списка открытых запросов на изменение.
- Закрепление ревьювера в запросе на изменение кода для инициирующей команду пользователя из чата. Также уведомление других групп, отслеживающих данный репозиторий, о закреплении этого пользователя.
- Получение статистики действий по запрашиваемому периоду.

Приложение будет также отсыпать уведомление в чат и выполнять следующие действия:

- Опрашивать сервера на наличие новых или измененных запросов на изменение.
- Уведомление ревьюверов о долгом рецензировании кода.
- Снятие ревьювера с этого запроса на изменение, обновление статуса в запросе на изменение.

Набор технологий для реализации

Выбор языка программирования. C#. Современный, кроссплатформенный, строго типизированный и объектно-ориентированный язык программирования. Также C# имеет мощную платформу .NET, имеющую множество библиотек. C#/.NET имеют открытый исходный код [2].

Выбор системы управления базами данных. PostgreSQL. Высокопроизводительная СУБД с открытым исходным кодом. Она является реляционной с поддержкой хранения “гибких моделей данных” (JSON, XML, графы, пары ключ-значение). Имеет высокую производительность, подходит для реализации проектов разных масштабов.

Развертывание системы. Docker. Платформа для контейнеризации приложений, позволяющая упаковывать код и зависимости в изолированные среды. Он обеспечивает единообразие работы на всех этапах разработки, тестирования и развертывания, устраняя проблемы совместимости.

Реализованные функции

На данный момент спроектирована база данных[3], создан прототип интерфейса взаимодействия пользователей с Telegram ботом (рис. 1). Частично реализовано взаимодействие с Git серверами [4], а также следующий функционал:

1. Добавление сервера Git для группы.
2. Добавление репозиториев.
3. Запрос списка открытых запросов на изменение кода.
4. Опрашивание серверов на наличие новых или измененных запросов на изменение для отслеживаемых группой репозиториев.

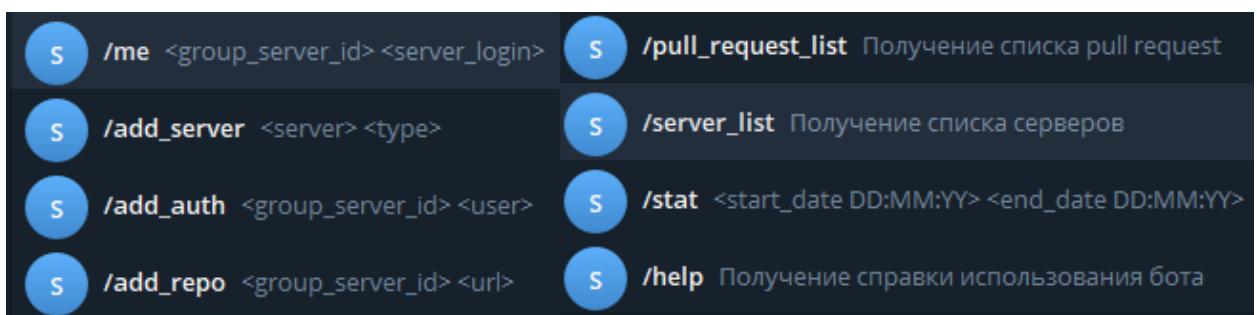


Рисунок №1 – Прототип меню Telegram бота.

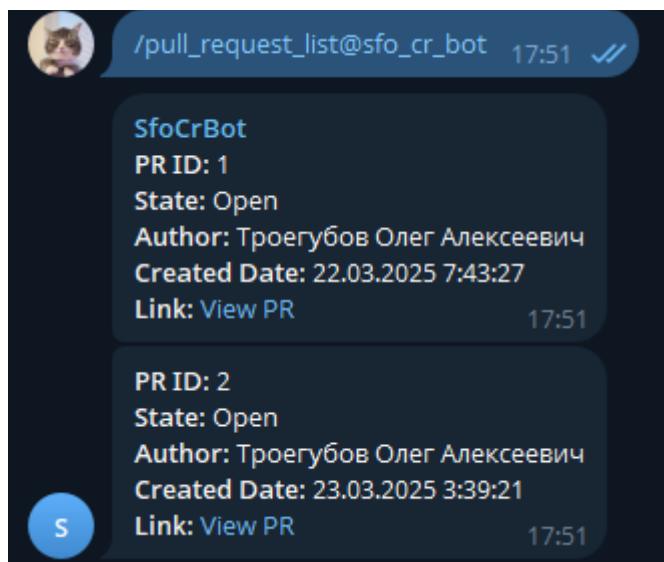


Рисунок №2 – Пример работы команды получения запросов на изменения.

Развитие проекта

Следующим шагом по разработке будет доработка недостающего функционала и улучшение существующего. Также после полной реализации проекта планируется интеграция с системой управления проектами – **Jira**.

Заключение

В данной статье рассмотрен процесс рецензирования программного кода (Code Review) как важный этап разработки программного обеспечения, повышающий качество кода и способствующий обмену знаниями в команде. Описан проект и технологии для его реализации. Реализован прототип бота с базовым функционалом.

Список литературы

1. Документация библиотеки Telegram.Bot. [Электронный ресурс] URL: <https://telegrambots.github.io/book/> (дата обращения: 23.03.2025) - Режим доступа: свободный.
2. Документация языка C#. [Электронный ресурс] – URL: <https://learn.microsoft.com/en-us/dotnet/csharp/> (дата обращения: 23.03.2024) - Режим доступа: свободный.
3. Документация фреймворка Entity Framework для работы с базами данных. [Электронный ресурс] – URL: <https://learn.microsoft.com/en-us/ef/> (дата обращения: 23.03.2024) - Режим доступа: свободный.
4. Документация API Git сервера BitBucket. [Электронный ресурс] – URL: <https://docs.atlassian.com/bitbucket-server/rest/7.21.0/bitbucket-rest.html> (дата обращения: 23.03.2024) - Режим доступа: свободный.