

1. Задачей, которую мы реализовывали, было добавление функционала показа ленты постов. Проект задумывался, как платформа для микроблогинга. Человек, может подписываться на других людей, выкладывать посты и лайкать их. Лента формируется из постов других людей, на которые подписан человек.

2. Это было достаточно сложное задание, требующее хорошей коммуникации и синхронизации между разработчиками. Мы использовали парное программирование для решения этой задачи, так как это позволяло нам быстро обмениваться идеями, дизайнами и кодом, а также сразу же тестировать и исправлять ошибки. Мы попробовали оба метода парного программирования «На равных» и «Пинг-Понг». Каждый из методов имеет свои плюсы и минусы и предназначен для разных задач. Для данной задачи метод «На равных» показался нам более приемлемым, потому что оба разработчика были примерно одинаковы по опыту и знаниям в области разработки, что позволило нам работать более эффективно. Метод «Пинг-Понг» мы использовали в других задачах, в которых не требуется проработка и сложный анализ архитектуры приложения.

3. Реализация функционала показа ленты постов была успешной. Для реализации на стороне FE использовались React и ApolloClient; Node.js, TypeScript для реализации BE. Работать вместе над задачей позволило нам улучшить коммуникацию, обмениваться знаниями и опытом, а также повысить уверенность в качестве кода.

В целом, мы получили положительные впечатления от парного программирования. Особенно понравилось то, что мы смогли на равных обсуждать реализацию каждой функции и принимать решения вместе, что позволило улучшить качество кода и снизить количество ошибок. Мы также научились лучше понимать друг друга и быстро реагировать на запросы другого разработчика.

4. В процессе работы столкнулись с несколькими трудностями при реализации функциональности. Была проблема, что участники реализовывали один и тот же функционал по отдельности, что приводило проблемам и сбоям в работе. Также возникли проблемы с утомляемостью участников. Парное программирование вызывало утомляемость при работе вместе на протяжении длительного времени, что снижало производительность и качество работы.

В качестве решения, сделали выводы, что необходимо сделать разграничение обязанностей, чтобы каждый участник работал над своим участком кода, сохраняя при этом общую цель. Также необходимо производить регулярные перерывы через определенные интервалы времени, чтобы избежать утомления и увеличить продуктивность.

5. Наша команда программистов положительно относится к использованию метода парного программирования в нашей работе. Мы считаем, что парное программирование очень эффективный подход к решению сложных задач в разработке веб-приложений, и может значительно повысить качество кода. Этот подход позволяет команде разработчиков работать вместе, обмениваться знаниями и идеями, и находить более эффективные решения для задач.

Мы уже начали использовать парное программирование, используя метод Пинг-Понг, для тестирования нашего веб-приложения. В процессе тестирования мы применяем как unit-тесты для отдельных компонентов, так и integrational-тесты для проверки взаимодействия между компонентами и их работы вместе. Это позволяет нам более точно определить возможные ошибки в нашем коде и сократить время на отладку.

Кроме того, мы планируем использовать парное программирование для других задач, связанных с разработкой веб-приложений. Мы уверены, что этот метод будет эффективен для решения различных сложных задач, таких как оптимизация

производительности приложения, рефакторинг кода и улучшение общей структуры приложения.

В целом, мы считаем, что использование парного программирования является одним из ключевых факторов, который может значительно улучшить качество и производительность нашего веб-приложения. Мы будем продолжать применять этот метод и искать новые способы его использования в нашей работе.