

УДК 004.4'2

## СРАВНИТЕЛЬНЫЙ АНАЛИЗ ТЕХНОЛОГИЙ УПРАВЛЕНИЯ СОСТОЯНИЕМ В REACT ПРИЛОЖЕНИЯХ: REDUX, MOBX И REACT CONTEXT (REACT HOOKS)

Мелихова К. А. (Университет ИТМО), Томилин Е. С. (Университет ИТМО),  
Кропотов А. С. (Университет ИТМО)

Научный руководитель – доцент, кандидат педагогических наук, Государев И.Б.  
(Университет ИТМО)

**Введение.** Сегодня использование фреймворков является неотъемлемой частью разработки веб-приложений. Фреймворки позволяют разработчикам ускорить и упростить процесс создания приложений, поскольку они предоставляют готовые инструменты и библиотеки, которые можно использовать для решения типовых задач.

Приложение должно иметь механизм для хранения и обработки данных, которые могут изменяться в ходе работы приложения. Контроль состояния является одной из ключевых характеристик, необходимых при разработке веб-приложений. При правильной организации управления состоянием упрощается процесс разработки и обеспечивается масштабируемость приложения, позволяющая эффективно управлять сложностью приложения и уменьшить нагрузку на серверные ресурсы [1].

Для управления состоянием дерева компонентов в веб-приложениях используются специальные библиотеки - стейт-менеджеры. Redux и Mobx являются наиболее популярными решениями в React-приложениях. Они предоставляют разработчикам широкий набор инструментов для управления состоянием. Помимо библиотек популярным подходом является использование React Context & React Hooks [2].

В большинстве исследований и статей по данной тематике часто рассматриваются теоретические аспекты управления состоянием, в то время как конкретные численные данные и практические результаты могут быть ограничены.

Цель работы – оценить производительность различных подходов и инструментов для управления состоянием, а также сравнение их преимуществ и недостатков.

Проведение такого исследования может помочь разработчикам в выборе подходящих технологий и инструментов для управления состоянием в веб-приложениях на React, что может ускорить процесс разработки и обеспечить оптимальную производительность приложения.

**Основная часть.** В процессе разработки веб-приложения могут появляться новые функции, компоненты и библиотеки, которые увеличивают объем кода и сложность архитектуры. Нагруженное приложение может столкнуться с различными проблемами, включая:

- 1) Медленное время компиляции и загрузки
- 2) Сложность отладки
- 3) Медленная производительность
- 4) Перерисовка
- 5) Ошибки и сбои
- 6) Низкая масштабируемость

Правильно организованное управление состоянием может привести к более эффективному использованию ресурсов и уменьшить нагрузку на процессор, а также уменьшить сложность кода и упростить отладку.

В рамках проведенной работы были рассмотрены три различных подхода для управления состоянием в React-приложениях – Redux, MobX, React Context & React Hooks. С целью сравнения эффективности каждого подхода, были реализованы два приложения разной сложности. Первое создано для того, чтобы проверить работу системы управления состоянием в условиях *небольшого количества компонентов* и свойств. Второе приложение – для

проверки эффективности систем в условиях *более сложной структуры* приложения, более высокой нагрузки на систему и более сложной логики взаимодействия компонентов. Для каждого приложения была реализована соответствующая система управления состоянием в рамках трех подходов.

По результатам работ были протестированы и сравнены системы управления состоянием на обоих приложениях, а также проведен эксперимент, направленный на количественную оценку влияния решения на производительность приложений.

**Выводы.** В результате исследования были сделаны следующие выводы: для рассмотренных видов приложений Context API обеспечивает лучшую производительность по сравнению с Mobx и Redux. При этом Mobx, во всех исследованных случаях, быстрее по производительности, чем Redux.

Исследованные/рассмотренные подходы к управлению состоянием предоставляют различные методы тестирования приложений, но Redux и MobX могут быть более удобными для проверки благодаря наличию специализированных инструментов для работы с тестами.

Использование MobX позволяет написать минимальное количество кода для решения поставленных задач, так как нет необходимости использовать шаблонный код, который требуется в Redux. Хотя Mobx уступает Context API по производительности, благодаря перечисленным выше характеристикам, разработка приложений с его использованием будет происходить быстрее, чем с использованием других подходов.

#### **Список использованных источников:**

1. Сукиасян В. М., Придиус Е. С. Современные принципы и подходы к Frontend архитектуре веб-приложений // Наука, техника и образование. – 2019.
2. Pronina D. and Kyrychenko I. Comparison of Redux and React Hooks Methods in Terms of Performance // CEUR Workshop Proceedings. – 2022.