Специфікація вимог до програмного забезпечення для додатку «Priority Queue Manager for React Apps»

Затверджено версією 1.0

Підготовлено Каменєвим Дмитром Вікторовичем

Харківський національний університет радіоелектроніки

15.06.2025

1. Вступ

1.1. Мета

Цей документ визначає функціональні та нефункціональні вимоги до додатку «REACT-PQM» (Priority Queue Manager), призначеного для оптимізації ререндерингу у вебзастосунках на базі бібліотеки React.

1.2. Умовні позначення

Ключові компоненти виділяються жирним, приклади та змінні – курсивом, назви структур, функцій і змінних – у стилі CamelCase.

1.3. Цільова аудиторія

Розробники React-додатків.

1.4. Область проєкту

Додаток призначений для аналізу та оптимізації повторного рендерингу компонентів у React-застосунках шляхом інтелектуального керування чергою оновлень. Для цього реалізовано інструмент Priority Queue Manager (PQM), який порівнює продуктивність оптимізованого рендерингу з базовими методами (наприклад, React.memo).

1.5. Посилання

IEEE Std 830-1998;

ISO/IEC 25010:2011.

2. Загальний опис

2.1. Перспектива продукту

Додаток є веб-утилітою, що працює в браузері, для вимірювання та оптимізації продуктивності рендерингу компонентів у React-додатках.

2.2. Основні функції

Динамічна пріоритизація оновлень компонентів на основі видимості та важливості;

Відстеження видимості компонентів за допомогою IntersectionObserver;

Логування метрик продуктивності (час рендерингу, кількість рендерів, використання пам’яті);

Порівняння продуктивності з базовими методами оптимізації (React.memo, useMemo);

Вивід результатів у консоль для аналізу.

2.3. Користувачі

Усі користувачі мають однаковий функціонал (розробники з доступом до вихідного коду).

2.4. Оточення

React: >=18.0;

TypeScript: >=5.0;

Браузери: Chrome, Firefox, Edge;

ОС: Linux, Windows, macOS.

2.5. Обмеження

Підтримка лише React;

Відсутність збереження результатів у файли;

Обмеження тестування 50 компонентами в одному сценарії.

2.6. Документація

README у репозиторії;

Коментарі у файлах з кодом.

2.7. Припущення

Користувач має базові знання React та TypeScript;

Налаштовано середовище розробки (Node.js, npm).

3. Зовнішні інтерфейси

3.1. Інтерфейс користувача

Веб-інтерфейс: запуск тестових сценаріїв через компонент AppWithManyForms;

Вивід метрик у консоль у текстовому форматі.

3.2. Апаратні інтерфейси

Комп’ютер із сучасним браузером.

3.3. Програмні інтерфейси

Взаємодія з DOM через ReactDOM;

Використання IntersectionObserver API.

3.4. Комунікаційні інтерфейси

Локальне виконання в браузері без мережевих запитів.

4. Особливості системи

4.1. Модулі оптимізації рендерингу

PriorityQueue: клас для керування чергою задач рендерингу;

PQM: основний менеджер для додавання та обробки задач;

useViewportObserver: хук для відстеження видимості компонентів;

usePQMComponent: хук для інтеграції PQM із React-компонентами.

4.2. Проведення тестів

Запуск тестового сценарію з 50 компонентами (AppWithManyForms);

Фіксація метрик: час рендерингу, кількість рендерів, використання пам’яті.

4.3. Вивід результатів

Текстові дані в консоль через logMetrics.

4.4. Налаштування

Можливість налаштувати пріоритети компонентів;

Кількість компонентів у тестовому сценарії (фіксована – 50).

5. Нефункціональні вимоги

5.1. Продуктивність

Точний вимір часу рендерингу та використання пам’яті для кожного компонента.

5.2. Безпека

Відсутність збереження чутливих даних;

Локальне виконання без передачі даних.

5.3. Якість

Відповідність стилю TypeScript/React;

Документовані хуки та класи.

5.4. Надійність

Коректна обробка помилок видимості та рендерингу;

Вивід помилок у консоль.

5.5. Гнучкість

Легке додавання нових компонентів для тестування;

Можливість розширення для інших фреймворків у майбутньому.

Appendix A: Глосарій

PQM – Priority Queue Manager, інструмент для керування рендерингом;

IntersectionObserver – API для відстеження видимості елементів;

React Fiber – архітектура рендерингу в React.

Appendix B: Моделі аналізу

Діаграма послідовності (див. додаток до дипломної роботи);

Архітектурна діаграма (див. додаток до дипломної роботи).

Appendix C: TBD

Додавання підтримки збереження результатів у файли;

Розширення тестування на 100+ компонентів.