

УДК 004.738.5:004.42

<https://doi.org/10.33619/2414-2948/123/18>

РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ВЕБ-ПРИЛОЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ СОВРЕМЕННЫХ ФРОНТЕНД-ТЕХНОЛОГИЙ: СРАВНИТЕЛЬНЫЙ АНАЛИЗ ФРЕЙМВОРКОВ

©Сафронов А. М., Российский технологический университет (РТУ МИРЭА),
г. Москва, Россия, sanchellos27@mail.ru

©Зеленова Ю. И., ORCID: 0000-0002-6979-2443, SPIN-код: 4568-0055, канд. техн. наук,
Российский технологический университет (РТУ МИРЭА),
г. Москва, Россия, zelenova.julie@yandex.ru

DEVELOPMENT OF A WEB APPLICATION USING MODERN FRONTEND TECHNOLOGIES: A COMPARATIVE ANALYSIS OF FRAMEWORKS

©Safronov A., Russian Technological University (RTU MIREA),
Moscow, Russia, sanchellos27@mail.ru

©Zelenova Ju., ORCID: 0000-0002-6979-2443, SPIN-code: 4568-0055, Ph.D., Russian
Technological University (RTU MIREA), Moscow, Russia, zelenova.julie@yandex.ru

Аннотация. Проведен сравнительный анализ современных фронтенд-фреймворков, применяемых при разработке пользовательских интерфейсов веб-приложений в контексте компьютерного дизайна. Пользовательский интерфейс рассматривается как комплексный объект, формируемый на пересечении визуальной композиции, интерактивного поведения и программной архитектуры, что обуславливает необходимость учета как дизайнерских, так и технологических факторов при его проектировании. В условиях роста сложности веб-приложений интерфейс перестает быть статичной визуальной оболочкой и приобретает свойства динамической системы, напрямую зависящей от данных и пользовательских сценариев взаимодействия. В рамках исследования рассматриваются архитектурные принципы, компонентные модели и подходы к управлению состоянием интерфейса на примере наиболее распространенных решений — React, Angular и Vue. Особое внимание уделяется их влиянию на визуальную структуру интерфейса, характер пользовательского взаимодействия и возможности построения устойчивых и масштабируемых дизайн-систем. Показано, что компонентный подход и декларативные модели описания интерфейса способствуют повышению предсказуемости визуального поведения, снижению сложности сопровождения и упрощают масштабирование интерфейсных решений в долгосрочной перспективе. В статье обоснована роль React как наиболее гибкой и масштабируемой технологии для проектирования сложных пользовательских интерфейсов, ориентированных на динамические сценарии взаимодействия. Отмечается, что нейтральность React по отношению к средствам стилизации и его развитая экосистема позволяют эффективно интегрировать современные подходы компьютерного дизайна, включая модульные и атомарные дизайн-системы. Делается вывод о том, что выбор фронтенд-фреймворка является ключевым фактором, определяющим качество пользовательского опыта, архитектурную устойчивость интерфейса и потенциал дальнейшего развития визуальных и интерактивных решений веб-приложений.

Abstract. Presents a comparative analysis of modern frontend frameworks used in the development of web application user interfaces within the context of computer design. The user interface is examined as a complex system formed at the intersection of visual composition, interactive behavior, and software architecture, which requires consideration of both design and



technological factors during the development process. As web applications increase in complexity, the interface evolves from a static visual layer into a dynamic environment directly influenced by data flows and user interaction scenarios. The study analyzes architectural principles, component-based models, and state management approaches using widely adopted solutions such as React, Angular, and Vue. Particular attention is paid to their impact on interface structure, interaction patterns, and the formation of scalable and consistent design systems. It is shown that component-oriented architectures and declarative interface descriptions improve predictability of visual behavior, reduce maintenance complexity, and support long-term scalability of interface solutions. The article substantiates the role of React as the most flexible and scalable technology for designing complex user interfaces focused on dynamic interaction scenarios. React's neutrality toward styling approaches and its extensive ecosystem enable effective integration of contemporary computer design practices, including modular and atomic design systems. It is concluded that the choice of a frontend framework is a key factor influencing user experience quality, architectural stability of the interface, and the potential for further development of visual and interactive solutions in modern web applications.

Ключевые слова: пользовательский интерфейс, веб-приложение, фронтенд-технологии (React, Angular, Vue), компонентная архитектура, виртуальный DOM.

Keywords: user interface, web application, front-end technologies (React, Angular, Vue), component architecture, virtual DOM.

Развитие цифровых продуктов в веб-среде сопровождается устойчивым ростом требований к качеству пользовательского интерфейса, который в современных условиях выступает не только визуальной оболочкой программной системы, но и основным каналом взаимодействия человека с функциональной логикой приложения. В рамках направления «Компьютерный дизайн» пользовательский интерфейс рассматривается как сложный синтетический объект, формируемый на пересечении визуальной композиции, интерактивного поведения и программной архитектуры. Такая постановка задачи обуславливает необходимость анализа не только дизайнерских решений, но и технологических средств, определяющих структуру, гибкость и выразительные возможности интерфейса [1].

Современные фронтенд-технологии сформировались как ответ на усложнение интерфейсных сценариев и рост интерактивной насыщенности веб-приложений. Если на ранних этапах развития веба пользовательский интерфейс представлял собой преимущественно статическую систему гипертекстовых страниц, то в настоящее время он функционирует как динамическая среда, в которой визуальное состояние напрямую зависит от данных, пользовательских действий и контекста использования. Такое смещение парадигмы привело к необходимости применения специализированных фреймворков и библиотек, обеспечивающих формализацию процессов проектирования и реализации интерфейсов [2].

Фронтенд-фреймворки выполняют роль архитектурного каркаса, в рамках которого выстраивается логика пользовательского взаимодействия и визуальная структура приложения. Их использование позволяет перейти от манипуляций отдельными элементами документа к моделированию интерфейса как целостной системы состояний. Для компьютерного дизайна это имеет принципиальное значение, поскольку открывает возможность проектировать интерфейс не как совокупность экранов, а как непрерывное пространство пользовательского опыта, в котором визуальные изменения подчинены внутренней логике приложения [3].

Одной из ключевых характеристик современных фронтенд-фреймворков является компонентный подход, предполагающий декомпозицию интерфейса на автономные, повторно используемые элементы. Каждый компонент объединяет в себе визуальное представление, поведенческую логику и параметры состояния, что обеспечивает целостность и управляемость интерфейсной структуры. С дизайнерской точки зрения компонентная модель способствует формированию устойчивых визуальных паттернов и дизайн-систем, в рамках которых обеспечивается единообразие пользовательского опыта на всех уровнях взаимодействия.

В настоящее время наиболее распространёнными решениями в области фронтенд-разработки являются React, Angular и Vue, каждый из которых реализует собственную концепцию построения пользовательского интерфейса. Их сравнительный анализ позволяет выявить различия не только в технической реализации, но и в подходах к организации визуальной и интерактивной структуры веб-приложения. Для задач компьютерного дизайна такие различия оказываются критически важными, поскольку напрямую влияют на свободу проектирования, масштабируемость интерфейса и возможности его эволюции [4].

Для наглядного сопоставления архитектурных и проектно-дизайнерских характеристик современных фронтенд-фреймворков авторами представлены обобщающая схема (рис. 1) и таблица (табл. 1). Представленная схема дополняет табличный анализ, позволяя визуализировать архитектурные взаимосвязи и логику формирования пользовательского интерфейса, что затруднительно отразить исключительно в табличной форме.

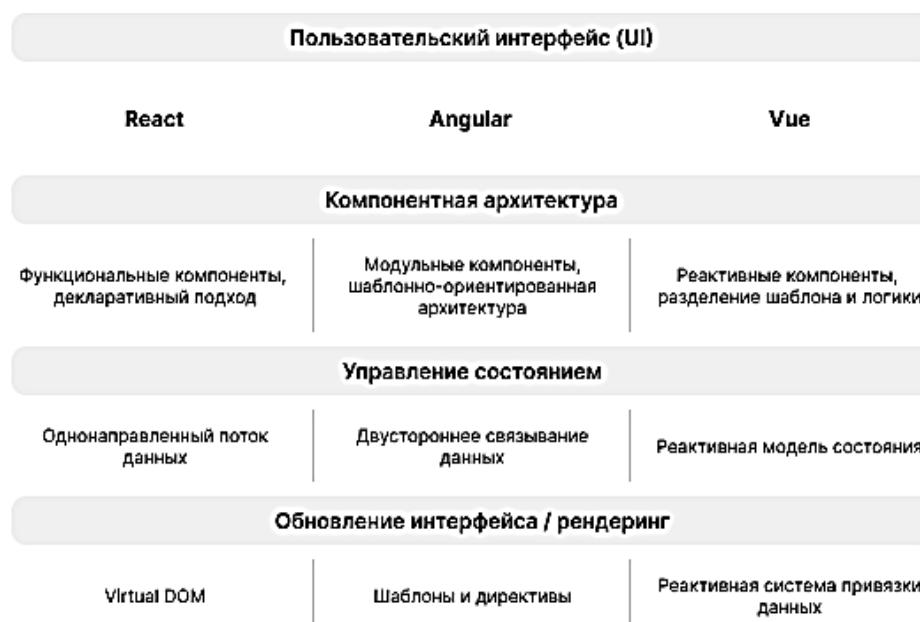


Рисунок. Сравнение архитектурных подходов фронтенд-фреймворков React, Angular и Vue в контексте проектирования пользовательского интерфейса

Представленные в Таблице различия позволяют перейти к более детальному анализу каждого из фреймворков с точки зрения их влияния на архитектуру и визуально-интерактивную структуру пользовательского интерфейса.

Angular представляет собой полнофункциональный фреймворк, ориентированный на строгую архитектурную модель и формализованные правила разработки. Он включает в себя развитую систему шаблонов, механизм двустороннего связывания данных и жестко заданную структуру приложения. Подобный подход обеспечивает высокую степень предсказуемости поведения интерфейса, однако одновременно накладывает ограничения на гибкость

визуального проектирования. В контексте компьютерного дизайна это может приводить к необходимости подстраивать дизайнерские решения под архитектурные требования фреймворка, а не наоборот.

Таблица
СРАВНИТЕЛЬНЫЙ АНАЛИЗ ФРОНТЕНД-ФРЕЙМВОРКОВ
В КОНТЕКСТЕ КОМПЬЮТЕРНОГО ДИЗАЙНА

Критерий анализа	React	Angular	Vue
Архитектурная модель	Библиотека уровня представления с расширяемой экосистемой	Полнофункциональный фреймворк с жёсткой архитектурой	Прогрессивный фреймворк с гибкой структурой
Подход к построению интерфейса	Декларативный, интерфейс как функция состояния	Шаблонно-ориентированный, с двусторонним связыванием данных	Декларативный, с разделением шаблонов и логики
Компонентная архитектура	Высокая степень модульности и повторного использования	Формализованная, строго регламентированная	Компонентная, менее жёстко стандартизированная
Управление состоянием	Однонаправленный поток данных, предсказуемые обновления интерфейса	Двустороннее связывание, повышенная сложность контроля состояния	Реактивная модель, промежуточная сложность
Влияние на дизайн-системы	Максимальная поддержка модульных и атомарных дизайн-систем	Ограниченнная гибкость при нестандартных визуальных решениях	Удобен для базовых дизайн-систем, требует дисциплины в масштабировании
Гибкость визуального проектирования	Очень высокая, не навязывает средств стилизации	Ограничена за счёт архитектурных ограничений	Средняя, зависит от архитектурных соглашений
Масштабируемость интерфейса	Высокая, подходит для сложных интерфейсных систем	Высокая, но с ростом архитектурной сложности	Средняя, требует дополнительных соглашений
Соответствие задачам компьютерного дизайна	Оптимальное для сложных интерактивных и экспериментальных интерфейсов	Подходит для формализованных корпоративных интерфейсов	Эффективен для проектов средней сложности

Vue занимает промежуточное положение между строгими и минималистичными решениями. Его архитектура ориентирована на постепенное внедрение и допускает использование как в небольших интерфейсных модулях, так и в полноценных приложениях. Vue предлагает декларативное описание интерфейса и компонентную модель, при этом сохраняя более традиционное разделение шаблонов, логики и стилей. Такой подход может быть удобен для дизайнеров, ориентированных на классическую модель работы с разметкой и визуальными слоями, однако в крупных проектах он требует дополнительных усилий для поддержания архитектурной целостности интерфейса.

Особое место в современном фронтенд-ландшафте занимает библиотека React, которая фактически сформировала новую парадигму разработки пользовательских интерфейсов. В отличие от полноценных фреймворков, React изначально позиционировался как библиотека для построения интерфейсов, сосредоточенная исключительно на уровне представления. Это

решение привело к высокой гибкости архитектуры и позволило сформировать вокруг React обширную экосистему дополнительных инструментов и библиотек [5].

Ключевая идея React заключается в представлении пользовательского интерфейса как функции от состояния. Визуальное представление определяется текущими данными и автоматически обновляется при их изменении, что исключает необходимость ручного управления элементами документа. Использование виртуального DOM позволяет минимизировать количество операций с реальным деревом документа, обеспечивая высокую производительность даже в условиях сложных интерфейсных сценариев. Для компьютерного дизайна это означает возможность создавать насыщенные интерактивные интерфейсы без потери отзывчивости и визуальной стабильности.

Компонентная архитектура React способствует строгой структуризации интерфейса и облегчает его проектирование на концептуальном уровне. Каждый компонент может рассматриваться как самостоятельный дизайнерский и функциональный объект, обладающий четко определенными входными параметрами и визуальным поведением. Такой подход соответствует принципам модульного дизайна и позволяет выстраивать интерфейсные системы, ориентированные на повторное использование и масштабирование.

Использование JSX, объединяющего описание структуры интерфейса и логики компонента, способствует более тесной интеграции дизайнера и программного мышления. Визуальная структура интерфейса становится неотделимой от данных, которые её определяют, что особенно важно при проектировании адаптивных и контекстно-зависимых интерфейсов. В рамках компьютерного дизайна это позволяет рассматривать интерфейс как динамическую композицию, изменяющуюся в зависимости от пользовательских сценариев.

Важным аспектом React является однодirectional поток данных, который обеспечивает предсказуемость изменений интерфейса. Любое обновление состояния приводит к однозначному пересчету визуального представления, что снижает вероятность несогласованности элементов интерфейса. Для сложных дизайнерских решений, включающих анимацию, переходы и многоуровневые интерактивные сценарии, такая предсказуемость имеет принципиальное значение [6].

В сравнительном контексте React демонстрирует высокую степень адаптивности к различным дизайнерским задачам. Он не навязывает конкретных решений в области стилизации или визуального оформления, что позволяет интегрировать различные подходы к дизайну, включая CSS-in-JS, модульные стили и дизайн-системы. Это делает React особенно привлекательным для проектов, в которых визуальная идентичность и пользовательский опыт играют ключевую роль.

Современные тенденции развития React связаны с расширением возможностей серверного рендеринга и гибридных архитектур, что реализуется в таких фреймворках, как Next.js. Эти решения позволяют оптимизировать загрузку интерфейса и повысить качество пользовательского восприятия за счёт сокращения времени отклика и более плавной инициализации визуальных элементов. В контексте компьютерного дизайна это открывает дополнительные возможности для создания визуально сложных интерфейсов, не перегружающих пользователя техническими задержками.

В рамках сравнительного анализа фронтенд-фреймворков целесообразно выделить совокупность критерииев, позволяющих оценивать их применимость не только с инженерной, но и с проектно-дизайнерской точки зрения. К числу таких критериев относятся степень абстрагирования визуального представления, управляемость интерфейсных состояний, поддержка дизайн-систем, а также возможности масштабирования визуальной структуры без потери целостности пользовательского опыта. Эти параметры непосредственно влияют на

процесс проектирования интерфейса и определяют границы дизайнерской свободы в рамках выбранной технологии.

Существенным фактором является то, каким образом фреймворк поддерживает формирование и развитие дизайн-систем. Современные интерфейсы всё чаще проектируются не как набор отдельных экранов, а как иерархически организованные системы визуальных компонентов, подчинённые единым принципам типографики, цветовой логики и интерактивного поведения. В этом контексте важным становится соответствие компонентной архитектуры фреймворка принципам атомарного и модульного дизайна, при которых каждый элемент интерфейса может быть переиспользован, модифицирован и встроен в различные сценарии взаимодействия.

Проведенный сравнительный анализ показывает, что React в наибольшей степени поддерживает подобный системный подход к проектированию интерфейсов. Его нейтральность по отношению к средствам стилизации и визуального оформления позволяет интегрировать как кастомные дизайн-системы, так и сторонние библиотеки компонентов без нарушения архитектурной логики приложения. Это особенно важно для проектов в области компьютерного дизайна, где визуальная идентичность и экспериментальные решения нередко выходят за рамки стандартных UI-паттернов. Дополнительным преимуществом React является его ориентация на декларативное описание интерфейса, что упрощает концептуальное моделирование пользовательских сценариев. Дизайнерские решения могут быть непосредственно соотнесены с логикой состояния и поведения компонентов, что снижает разрыв между этапами проектирования и реализации. В результате интерфейс перестает быть статичным результатом верстки и рассматривается как динамическая структура, эволюционирующая вместе с функциональными и визуальными требованиями продукта. С позиций компьютерного дизайна такой подход способствует формированию более устойчивого и согласованного пользовательского опыта, поскольку визуальные изменения интерфейса становятся следствием чётко определённых архитектурных правил. Это позволяет не только повысить качество реализации сложных интерактивных решений, но и упростить их дальнейшее развитие и сопровождение в рамках долгосрочных цифровых проектов.

Таким образом, сравнительный анализ современных фронтенд-фреймворков показывает, что выбор технологической основы напрямую влияет на характер пользовательского интерфейса и способы его проектирования. React в настоящее время представляет собой наиболее универсальный инструмент для разработки интерфейсов, ориентированных на сложные интерактивные сценарии и высокие требования к визуальной и архитектурной гибкости. Его концептуальная модель соответствует современным представлениям о пользовательском интерфейсе как динамической системе, что делает React обоснованным и перспективным выбором для дальнейших научных исследований и магистерской диссертации в области компьютерного дизайна.

Список литературы:

1. Бондаренко О. С. Анализ методов обновления DOM в современных веб фреймворках: Virtual DOM и Incremental DOM // Программные системы и вычислительные методы. 2025. №2. С. 35-43.
2. Ратушняк Е. А. Исследование производительности современных клиентских веб-фреймворков // Программные системы и вычислительные методы. 2025. №2.
3. Калугина К. Н. Сравнительный анализ js фреймворков: Vue, React и Angular // Теория и практика экономики и предпринимательства. 2021. С. 285-286.

4. Евстратова О. Д. Сравнительный анализ фреймворков для frontend-разработки // Молодой ученый. 2024. №22 (521). С. 25-26.
5. Чепурной М. П. Сравнительный анализ frontend-фреймворков: React, Angular и Vue.js // Фронтенд-разработка и пользовательские интерфейсы. 2025. <https://s.eduherald.ru/>
6. Ратушняк Е. А. Анализ современных методов оптимизации в React // Программные системы и вычислительные методы. 2025. №3. С. 1-9.

References:

1. Bondarenko, O. S. (2025). Analiz metodov obnovleniya DOM v sovremennykh veb-freimvorkakh: Virtual DOM i Incremental DOM. *Programmnye sistemy i vychislitel'nye metody*, (2), 35-43.
2. Ratushnyak, E. A. (2025). Issledovanie proizvoditel'nosti sovremennykh klientskikh veb-freimvorkov. *Programmnye sistemy i vychislitel'nye metody*, (2). (in Russian).
3. Kalugina, K. N. (2021). Sravnitel'nyi analiz js freimvorkov: Vue, React i Angular. *Teoriya i praktika ekonomiki i predprinimatel'stva*, 285-286. (in Russian).
4. Evstratova, O. D. (2024). Sravnitel'nyi analiz freimvorkov dlya frontend-razrabotki. *Molodoi uchenyi*, (22 (521)), 25-26. (in Russian).
5. Chepurnoi, M. P. (2025). Sravnitel'nyi analiz frontend-freimvorkov: React, Angular i Vue.js. *Frontend-razrabortka i pol'zovatel'skie interfeisy*. (in Russian). <https://s.eduherald.ru/>
6. Ratushnyak, E. A. (2025). Analiz sovremennykh metodov optimizatsii v React. *Programmnye sistemy i vychislitel'nye metody*, (3), 1-9. (in Russian).

Поступила в редакцию
17.12.2025 г.

Принята к публикации
25.12.2025 г.

Ссылка для цитирования:

Сафонов А. М., Зеленова Ю. И. Разработка пользовательского интерфейса веб-приложения с использованием современных фронтенд-технологий: сравнительный анализ фреймворков // Бюллетень науки и практики. 2026. Т. 12. №2. С. 164-170. <https://doi.org/10.33619/2414-2948/123/18>

Cite as (APA):

Safronov, A., & Zelenova, Ju. (2026). Development of a Web Application using Modern Frontend Technologies: a Comparative Analysis of Frameworks. *Bulletin of Science and Practice*, 12(2), 164-170. (in Russian). <https://doi.org/10.33619/2414-2948/123/18>