

УДК 004.4

СРАВНЕНИЕ ПОДХОДОВ К ВЕБ-РАЗРАБОТКЕ: ПРЕИМУЩЕСТВА И ОГРАНИЧЕНИЯ NO-CODE И ТРАДИЦИОННОГО ПРОГРАММИРОВАНИЯ

Пепельшев Д.И., Кулагина Е.С.

ФГАОУ ВО «Уральский федеральный университет имени первого Президента России Б.Н. Ельцина», филиал, Нижний Тагил, e-mail: dmitry.pepelyshev@urfu.ru

Исследование направлено на сопоставительный анализ no-code платформ и традиционных методов программирования при разработке веб-сайтов. В ходе данной работы было проведено комплексное сравнение технологических и эксплуатационных параметров рассматриваемых подходов, что позволило выявить их преимущества и ограничения. Экспериментальная часть исследования включала сравнительный анализ платформы Tilda и традиционного стека технологий (HTML, CSS, JavaScript). Основными критериями оценки стали следующие параметры: производительность, уровень безопасности, кастомизация, масштабирование и перспективы долгосрочной поддержки. Результаты проведенного анализа показывают, что no-code решения обладают рядом существенных технологических ограничений, недостатками данных платформ являются: сниженная производительность из-за генерации избыточного кода, наличие уязвимостей в системе защиты данных, ограниченные возможности для оптимизации. На основании полученных данных можно сделать вывод о том, что no-code технологии не способны полностью заменить профессиональные методы веб-разработки, практическое применение таких платформ целесообразно ограничить созданием простых сайтов и небольших краткосрочных проектов. Для реализации сложных коммерческих веб-сайтов, требующих надежности, высокого уровня безопасности и архитектурной гибкости, традиционные подходы программирования сохраняют неоспоримые преимущества как с точки зрения качества конечного продукта, так и в аспекте долгосрочной поддержки.

Ключевые слова: no-code платформы, Tilda, традиционное программирование, сравнительный анализ, сайт

COMPARISON OF APPROACHES TO WEB DEVELOPMENT: ADVANTAGES AND LIMITATIONS OF NO-CODE AND TRADITIONAL PROGRAMMING

Pepelyshev D.I., Kulagina E.S.

*Ural Federal University named after the first President of Russia B.N. Yeltsin,
Nizhny Tagil branch, Nizhny Tagil, e-mail: dmitry.pepelyshev@urfu.ru*

The research is aimed at a comparative analysis of no-code platforms and traditional programming methods in website development. In the course of this work, a comprehensive comparison of the technological and operational parameters of the considered approaches was carried out, which made it possible to identify their advantages and limitations. The experimental part of the study included a comparative analysis of the Tilda platform and the traditional technology stack (HTML, CSS, JavaScript). The main evaluation criteria were the following parameters: performance, security level, customization, scaling, and prospects for long-term support. The results of the analysis show that no-code solutions have a number of significant technological limitations, the disadvantages of these platforms are: reduced performance due to redundant code generation, vulnerabilities in the data protection system, and limited optimization opportunities. Based on the data obtained, it can be concluded that no-code technologies are not able to completely replace professional web development methods, and it is advisable to limit the practical use of such platforms to creating simple websites and small short-term projects. To implement complex commercial websites that require reliability, high levels of security, and architectural flexibility, traditional programming approaches retain undeniable advantages both in terms of end product quality and long-term support.

Keywords: no-code platforms, Tilda, traditional programming, comparative analysis, website

Введение

В современном мире веб-сайты используются во всех сферах человеческой деятельности, их функциональное разнообразие позволяет решать широкий спектр как профессиональных, так и бытовых задач. На данный момент сайты перестали быть только источниками информации, превратившись в многофункциональные платформы для бизнеса и коммуникации [1, с. 15].

Данная трансформация обусловила появление новых требований к характери-

стикам сайтов. Современные веб-ресурсы должны сочетать в себе адаптивность для различных устройств, интерактивные элементы управления, реализацию возможности интеграции с внешними сервисами. При этом сохраняются и базовые требования к юзабилити: скорость загрузки, интуитивность интерфейса и безопасность данных.

Современные веб-сайты, представляющие собой сложные системы взаимосвязанных страниц с единой структурой, могут создаваться разными способами. Тради-

ционный подход предполагает разработку с использованием HTML, CSS и JavaScript и требует от разработчиков наличия профессиональных знаний и навыков. В качестве альтернативы выступают онлайн-конструкторы сайтов, позволяющие быстро создавать готовые веб-сайты и предлагающие интуитивно понятные интерфейсы с готовыми шаблонами и примерами, что значительно снижает порог входа для непрофессиональных разработчиков [2, с. 55]. Выбор между подходами определяется комплексом ключевых факторов: уровень функциональности разрабатываемого продукта, бюджетные ограничения, временные рамки проекта и техническая подготовка команд разработчиков.

No-code представляет собой инновационную методологию проектирования и разработки веб-приложений, основанную на применении специализированных платформ с графическим интерфейсом [3, с. 184]. Данный подход включает использование интерфейса drag-and-drop, который обеспечивает возможность создания функциональных веб-приложений с помощью применения интерактивной технологии перемещения элементов без написания программного кода [4, с. 396].

Использование no-code платформ для создания веб-сайтов предполагает четкую структуру работы. На начальном этапе разработчик выбирает подходящие шаблоны из обширной библиотеки компонентов. Следующим шагом является этап визуальной кастомизации, на котором каждый элемент интерфейса изменяется и адаптируется под конкретные требования проекта. Завершающим этапом разработки становится наполнение созданной структуры текстовым, графическим и мультимедийным контентом [3, с. 184].

Одним из преимуществ использования no-code решений является их архитектура, значительная часть сложных процессов в которой автоматизирована. Платформа выполняет следующие backend-задачи: управление данными, организация API-взаимодействия, обеспечение базовых механизмов безопасности.

Данный подход существенно снижает порог вхождения в веб-разработку, позволяя создавать функциональные продукты специалистам без глубоких технических знаний. Однако следует отметить, что no-code решения могут иметь ограничения в плане гибкости, обеспечения безопасности и масштабируемости по сравнению с традиционными методами разработки [3, с. 184].

В данной работе была рассмотрена одна из самых популярных существующих no-

code платформ Tilda, которая представляет собой конструктор сайтов, основанный на использовании готовых блоков и визуального редактора. Tilda предоставляет разработчикам веб-сайтов готовые композиционные решения и шаблоны, требующие последующей кастомизации под конкретные запросы пользователя [5, с. 157]. Также следует отметить, что значительная часть функционала, включая специализированные блоки, становится доступной только после активации платной подписки, бесплатная версия позволяет создать один сайт с базовым набором функций, небольшим количеством страниц и ограниченными возможностями хостинга.

Для проектов, требующих нестандартных дизайнерских решений, в Tilda может использоваться Zero Block. Данный инструмент доступен в бесплатной версии конструктора сайтов, он позволяет проектировать и кастомизировать элементы с уникальным дизайном, регулировать параметры анимации и обеспечивать детализированную проработку каждого виджета интерфейса, а также предоставляет возможность настройки адаптации контента для различных устройств [6, с. 132].

Tilda является сбалансированным no-code решением, сочетающим в себе интуитивно понятный интерфейс и набор инструментов для профессиональной верстки, она отлично подходит для быстрого создания простых веб-сайтов. Ключевым преимуществом конструктора сайтов является возможность быстрого запуска визуально привлекательных сайтов без необходимости написания программного кода. Однако существующие ограничения платформы снижают ее применимость для разработки сложных высоконагруженных проектов, требующих глубокой кастомизации, высокого уровня безопасности и масштабируемости.

На данный момент создание профессиональных веб-приложений основывается на трех ключевых технологиях: HTML используется для структурирования страницы, CSS – для оформления внешнего вида, JavaScript – для реализации динамического поведения и интерактивности. Данные технологии образуют фундамент современной веб-разработки, независимо от сложности и назначения создаваемого приложения [7].

HTML (HyperText Markup Language) является стандартным языком разметки, используемым для структурирования содержимого веб-страниц. Данная технология основана на системе тегов – специальных маркеров, определяющих тип и назначение элементов документа. Посредством те-

гов осуществляется разметка компонентов веб-страницы, включая текстовые блоки, мультимедийные элементы, навигационные компоненты, структурные единицы и другие элементы [7]. Современная спецификация HTML5 позволяет расширить базовые возможности языка, дополняя их семантическими тегами и встроенной поддержкой мультимедиа и сохраняя при этом принцип разделения структуры, оформления и поведения веб-страниц.

CSS (Cascading Style Sheets) представляет собой язык описания стилей, предназначенный для визуального оформления веб-страниц. Данная технология обеспечивает возможность детальной настройки внешнего вида HTML-элементов, включая параметры типографики (шрифты, кегль, начертание), цветовые схемы (фон, текстуры), пространственное расположение компонентов (размеры, позиционирование, отступы) и другие параметры [8, с. 121].

Главной отличительной особенностью CSS является реализация концепции разделения содержания и представления веб-страниц. Данная технология позволяет создавать адаптивные интерфейсы, автоматически подстраивающиеся под различные устройства и форматы отображения. Важным преимуществом использования CSS в веб-разработке является архитектура, предполагающая вынесение стилевых правил в отдельные файлы, что обеспечивает эффективное кэширование и существенно упрощает процесс поддержки и обновления дизайна [8, с. 121–122].

Следует отметить, что CSS обладает рядом ограничений. Наиболее значимым является проблема кросс-браузерной совместимости, заключающаяся в том, что различные браузеры могут по-разному интерпретировать одинаковые стилевые правила. Еще одним ограничением является тесная взаимосвязь между CSS-селекторами и HTML-структурой, которая требует синхронного внесения изменений в оба компонента, что усложняет процесс разработки в некоторых сценариях. Несмотря на эти ограничения, CSS остается неотъемлемой частью современной веб-разработки.

JavaScript представляет собой высокоуровневый язык сценариев, выполняемый на стороне клиента и предназначенный для обеспечения интерактивности веб-страниц. Данная технология позволяет динамически изменять содержимое и поведение веб-страниц, реагируя на действия пользователя.

К ключевым преимуществам JavaScript относится универсальность, которая заключается в том, что язык позволяет реализо-

вывать как простые элементы интерфейса, так и сложную клиентскую логику, поддерживая при этом интеграцию с многочисленными API и сторонними сервисами. Также важным достоинством является кроссплатформенность, обеспечивающая выполнение программного кода в большинстве современных браузеров без необходимости адаптации под конкретные устройства или операционные системы. К дополнительным преимуществам можно отнести динамическое обновление контента без перезагрузки страницы и открытую лицензионную модель [9, с. 28].

Основным недостатком JavaScript является сложность написания и поддержки кода. К основным сложностям можно отнести асинхронную природу языка, динамическую типизацию и гибкость синтаксиса, которые могут приводить к трудностям в отладке кода и необходимости написания дополнительных тест-кейсов. Также стоит отметить, что JavaScript выполняется на стороне клиента и имеет уязвимости в виде межсайтовой подделки запросов, js-инъекций, межсайтового скриптинга. Для предотвращения этих угроз необходимо использовать CSP (Content Security Policy), настраивать механизмы аутентификации и авторизации пользователей, использовать современные API и экранировать пользовательский ввод [9, с. 28–29].

Цель исследования – проведение комплексного сравнительного анализа двух подходов к веб-разработке: применение no-code платформы Tilda и традиционного метода программирования на базе HTML, CSS и JavaScript.

Материалы и методы исследования

Для проведения исследования был использован сравнительно-аналитический метод, обеспечивающий всестороннее сопоставление ключевых параметров no-code платформы Tilda и традиционного подхода к веб-разработке на основе HTML, CSS и JavaScript.

Результаты исследования и их обсуждение

Современные веб-сайты могут создаваться с помощью двух абсолютно разных подходов, которые имеют свои преимущества, недостатки, особенности и ограничения. В результате проведенного сравнительного анализа ключевых аспектов каждого из рассматриваемых методов была составлена сравнительная таблица, наглядно показывающая разницу между no-code платформой Tilda и традиционной разработкой на базе HTML, CSS и JavaScript.

Сравнительная характеристика no-code платформы Tilda
и традиционной веб-разработки

Критерий сравнения	No-code платформа Tilda	Традиционная разработка на базе HTML, CSS и JavaScript
Производительность	Средняя (избыточный сгенерированный программный код)	Высокая (оптимизированный программный код)
Безопасность	Базовый уровень	Высокий уровень
Кастомизация	Ограниченная (в рамках возможностей платформы)	Полная (неограниченные возможности)
Масштабируемость	Ограниченная	Высокая
Долгосрочная поддержка	Ограниченная	Неограниченная
Стоимость разработки	Платная подписка для доступа к полному функционалу платформы	Бесплатные технологии
Скорость разработки	Высокая (используются готовые шаблоны и блоки)	Низкая (требуется написание программного кода)
Порог вхождения	Низкий (не требуются навыки программирования)	Высокий (необходимы технические знания и навыки программирования)
Адаптивность	Хорошая (используются встроенные адаптивные шаблоны)	Полная (точный контроль адаптации)
Интеграция с API	Ограниченная	Полноценная
Сферы применения	Лендинги, сайты-визитки, небольшие проекты	Проекты любого уровня сложности и масштаба

Данный анализ позволил выявить технологические возможности, преимущества и ограничения каждого из подходов, а также показал, что no-code платформы и традиционный подход к созданию веб-сайтов сильно отличаются между собой и могут успешно использоваться в зависимости от уровня сложности разрабатываемого проекта, наличия ресурсов и предъявляемых требований.

Заключение

В результате проведенного исследования были определены ключевые различия между no-code решениями и традиционными подходами к веб-разработке. No-code платформы, например Tilda, могут быть успешно применимы для создания простых проектов, для которых важна оперативная разработка рабочей и визуально оформленной версии продукта в кратчайшие сроки и минимальные требования к техническим навыкам разработчиков. Основные преимущества данных платформ заключаются в наличии интуитивно понятного интерфейса и готовой библиотеки шаблонов, а также автоматизации сложных backend-процессов. Технологическими ограничениями no-code решений являются неполные возможности кастомизации при использовании бесплатной подписки, низкая производительность и недостаточный уровень безопасности, а также прямая зависимость от функциональности выбранной платформы.

Традиционные методы программирования на базе HTML, CSS и JavaScript чаще всего используются для реализации сложных коммерческих проектов, но также могут быть применимы и для создания более простых веб-сайтов. К преимуществам данного подхода можно отнести наличие полного контроля над программным кодом, поддержку возможности глубокой оптимизации и кастомизации, обеспечение высокого уровня безопасности и масштабирования. Разработка с помощью традиционных методов программирования требует специальных технических знаний и навыков программирования от разработчиков и длительные сроки для реализации проектов.

Таким образом, каждый из рассматриваемых подходов может быть успешно применен для реализации веб-сайтов. No-code платформы оптимально используются для создания простых маркетинговых лендингов, целевых страниц, интернет-журналов, корпоративных сайтов-визиток и небольших интернет-магазинов. Традиционные методы программирования остаются незаменимыми для реализации сложных коммерческих систем, высоконагруженных проектов и решений, требующих уникальной функциональности, высокого уровня безопасности и масштабируемости.

Выбор подхода к реализации веб-ресурса должен основываться на оценке проектных требований, которые включают в себя сроки реализации, бюджетные огра-

ничения, необходимый уровень функциональности, безопасности и оптимизации, уровень знаний разработчиков. В некоторых случаях рациональным решением может являться применение гибридного подхода, сочетающего в себе преимущества каждого из рассмотренных в исследовании методов.

Список литературы

1. Антонов В.Ф. Анализ характеристик современных фреймворков // Научный альманах. 2020. № 9–2 (71). С. 15–17. URL: <https://www.elibrary.ru/item.asp?id=44120664> (дата обращения: 15.05.2025).
2. Павлушина В.А., Мостяева И.В. Создание и разработка персонального сайта учителя с использованием современных конструкторов сайтов // Информатика и прикладная математика. 2021. № 27. С. 54–57. URL: <https://elibrary.ru/item.asp?id=47398282> (дата обращения: 25.05.2025).
3. Яблокова А.А., Ковалев И.В. Обзор исследований в области применения инструментов проектирования и создания информационных систем на базе стека frontend, backend и fullstack технологий // Перспективы молодежной науки. 2023. Т. 10. С. 183–186. URL: <https://www.elibrary.ru/item.asp?id=54511324> (дата обращения: 15.05.2025).
4. Сулягина Е.А. Технология Drag and Drop как инструмент формирования учебно-познавательной деятельности

на уроках математики // Современные научные исследования и разработки. 2016. № 3 (3). С. 395–397. URL: <https://elibrary.ru/item.asp?id=26464173> (дата обращения: 25.05.2025).

5. Турдубаева Ж.А., Исманов О.М., Саланова А.Ж. Создание сайта автосалона с помощью конструктора сайтов Tilda Publishing // Бюллетень науки и практики. 2025. Т. 11. № 4. С. 156–159. URL: <https://elibrary.ru/item.asp?id=80668802> (дата обращения: 25.05.2025). DOI: 10.33619/2414-2948/113/23.

6. Эркенова М.У., Бытдаева Л.И., Хатуяев Т.А. Особенности использования web-конструктора Tilda при разработке web-проектов для малого бизнеса // Тенденции развития науки и образования. 2022. № 92–6. С. 131–133. URL: <https://elibrary.ru/item.asp?id=50398204> (дата обращения: 25.05.2025). DOI: 10.18411/trnio-12-2022-299.

7. Домбровский Я.А., Ткачев Е.В. Создание веб-приложения средствами HTML, CSS, JavaScript // Дневник науки. 2025. № 3. URL: https://dnevniknauki.ru/images/publications/2025/3/technics/Dombrovsky_Tkachev.pdf (дата обращения: 25.05.2025).

8. Новиков С.А. CSS-каскадные таблицы стилей. Внедрение CSS в html // Информационные технологии и прикладная математика. 2017. С. 121–124. URL: <https://elibrary.ru/item.asp?id=30515921> (дата обращения: 25.05.2025).

9. Галимуллин Н.Р. Javascript: разработка интерактивных веб-страниц с помощью javascript // Экономика и управление: проблемы, решения. 2024. Т. 9. № 9 (150). С. 27–32. URL: <https://elibrary.ru/item.asp?id=71727936> (дата обращения: 25.05.2025). DOI: 10.36871/ek.up.p.r.2024.09.09.004.