

Костенко Вероника Александровна, магистрант,
АНО ВО «Российский новый университет», г. Москва

Золотарев Олег Васильевич,
к.т.н, доцент, заведующий кафедрой
информационных систем в экономике и управлении,
АНО ВО «Российский новый университет», г. Москва

КОМПОНЕНТНЫЙ ПОДХОД В РАЗРАБОТКЕ ИНТЕРФЕЙСОВ: ОТ UI-KIT К ДИЗАЙН-СИСТЕМАМ С ИИ

Аннотация: Рассматривается развитие компонентного подхода в интерфейсах и его усиление с помощью искусственного интеллекта для повышения эффективности, масштабируемости и автоматизации процессов.

Ключевые слова: библиотека компонентов, интерфейс, UI-kit, дизайн-система, ИИ-автоматизация

В условиях быстрорастущего рынка цифровых продуктов качественный интерфейс влияет на успех продукта в условиях высокой конкуренции. Удобные и функциональные приложения имеют более высокий уровень вовлечения, пониженный уровень отказов и сравнительно высокую конверсию. Также качественный интерфейс снижает нагрузку на поддержку, так как пользователи реже встречаются со сложным и непоследовательным функционалом. В итоге вложения в пользовательский опыт окупаются за счет лояльности аудитории и снижения косвенных затрат.

Однако упор на качественный интерфейс не должен идти с ущербом на скорость разработки сервиса. Поэтому популярностью пользуются различные методы оптимизации процесса создания интерфейса: стандарты, фреймворки, инструментов автоматизации рутинных задач процессов прототипирования, разработки и тестирования интерфейсов.

Самым базовым методом ускорения создание интерфейса является внедрение компонентного подхода к дизайну и разработке.

Компонентный подход в дизайне интерфейсов был представлен Брэдом Фростом в 2013 году. Системы компонентов строятся на принципах иерархичности, стандартизации, модульности и переиспользуемости компонентов [1]. Этот подход прошел несколько эволюционных ступеней с точки зрения практической реализации.

Самый базовый уровень компонентного подхода осуществляется с помощью UI-kit. UI-kit это набор компонентов, которые дизайнеры используют при прототипировании интерфейса, а также их базовая структура в коде. UI-kit состоит из стандартизированных графических элементов интерфейса, таких как иллюстрации иконки, кнопки, поля ввода и выпадающие списки, шаблоны структур страниц. Также в UI-kit входят элементы брендинга – шрифты, логотипы, цветовые палитры [2]. В коде компонент UI-kit описывается как визуальная структура и набор стилей.

Логика прописывается уже на этапе создания интерфейса. UI-kit ускоряет создание макетов и обеспечивает визуальную согласованность. Ускоряет перенос макета в код и уменьшает количество ошибок, поскольку базовые элементы уже сверстаны, а цвета и отступы однозначно определены.

UI-киты легко создаются и подходят для быстрого старта, например для создания лендингов и промо сайтов или MVP проектов небольшого масштаба.



Возможные проблемы:

- При обновлении структуры все экземпляры компонента придется менять вручную
- Переиспользование осуществляется через простое копирование
- Возможны конфликты стилей
- Возможно снижение производительности из-за объема стилевых файлов

Частично эти проблемы были решены при создании компонентных библиотек.

Компонентная библиотека – это набор переиспользуемых интерактивных компонентов [3], часто сделанных в формате библиотеки для реактивного фреймворка (Vue, React, Angular). В отличие от компонентов UI-kit в них уже описана логика взаимодействия и обработки событий. Реактивность компонента позволяет гибко управлять его состоянием и синхронизировать его с глобальными состояниями (например, при добавлении товара отображение корзины сразу обновляется). Также такие компоненты изолированы от контекста и при разработке их можно тестировать по отдельности, что повышает надежность использования.

С помощью библиотеки реактивных можно быстро собрать рабочий интерфейс как из конструктора. В интерфейсы, реализованные с помощью компонентных библиотек, легче вносить глобальные изменения. Все модификации, внесенные в логику и структуру компонента автоматически распространятся на все интерфейсы куда подключена библиотека.

Разработка интерфейса на реактивных компонентах подходит для создания веб-приложений, SaaS. В такие интерфейсы проще добавлять новый функционал и настраивать в соответствии с категорией и уровнем доступа пользователя. Для посадочных страниц и промо-сайтов реактивный функционал будет избыточным.

Возможные проблемы:

- Разработчики могут не понимать работу готового компонента и будут делать свои аналоги
- Дизайнеры могут не понимать ограничений в функционале компонентов

Дополнительно стоит упомянуть возможность использования публичной компонентной библиотеки. Поскольку компоненты в реактивных фреймворках приспособлены под тонкую настройку отображения, готовую библиотеку несложно адаптировать под визуальный стиль бренда. Это позволит сэкономить время на разработку и тестирование. Большинство публичных решений уже приведены в соответствии со стандартами доступности, а также поставляются с готовой документацией, где описаны механизмы настройки и продемонстрированы примеры использования. Однако может потребоваться доработка и адаптация библиотеки для осуществления сложной логики или уникальных элементов интерфейса. Если таких доработок необходимо много, лучше создавать свою библиотеку.

Компонентные библиотеки упростили коммуникацию между дизайнерами и разработчиками, а также позволили создавать более сложные приложения. Однако их оказалось недостаточно, для больших экосистем продуктов, где над сервисами работает большое количество разных команд. Потребность в тщательной стандартизации подходов и организации процессов разработки интерфейсов привела к необходимости создания дизайн-систем.

Дизайн-система представляет собой полный набор стандартов, предназначенный для управления проектированием и разработкой интерфейсов в крупномасштабных системах с применением переиспользуемых компонентов [4].

В дополнении к библиотеке компонентов дизайн-система может включать следующую документацию:

- Визуальный язык (палитра, типографика, каталог иллюстраций и иконок, параметры сетки, эффектов и анимаций)



- Гайдлайны использования компонентов и паттернов
- Принципы доступности контента
- Редакционная политика – тональность текстов и правила написания различных сообщений
- Регламенты процессов – как предлагать новые компоненты, как вносить изменения в компоненты
- Документация для дизайнеров – как оформлять макеты; какие компоненты, когда использовать

- Документация для разработчиков – параметры компонентов, правила тестирования

Несложно заметить, что такая четкая регламентация всех аспектов разработки интерфейсов представляет широкие возможности для дальнейшей автоматизации в том числе с использованием машинного обучения и больших лингвистических моделей. Можно выделить несколько перспективных направлений усиления дизайн-системы инструментами ИИ:

- Генерация кода простых интерфейсов по словесному описанию
- Автоматическое тестирование новых и измененных компонентов
- Написание интерфейсных текстов в соответствии с правилами редакционной политики
- Адаптация компонентов одного фреймворка к другому

Однако для перехода на следующий эволюционный уровень разработки интерфейсов с активным участием искусственного интеллекта необходима серьезная подготовительная работа. Эффективное применение ИИ в перечисленных выше задачах требует высокого качества исходных данных. Это подразумевает регулярный сбор, актуализацию и структурирование информации о компонентах, гайдлайнах, пользовательских сценариях и результатах тестирования. Кроме того, развёртывание инфраструктуры на базе ИИ связано с существенными затратами – как технологическими, так и организационными. Необходимо обеспечить не только вычислительные мощности, но и сформировать команды специалистов, способных поддерживать и развивать такие решения.

Список литературы:

1. Frost B. Atomic Design. 2016 [Электронный ресурс]. – URL: <http://atomicdesign.bradfrost.com/> (дата обращения: 20.05.2025)
2. What is a UI Kit? Examples and Why You Should Use One // Untitled UI [Электронный ресурс]. – URL: <https://www.untitledui.com/blog/what-is-a-ui-kit> (дата обращения: 14.05.2025)
3. Component Library vs. Design System // Public UI. 2023 [Электронный ресурс]. – URL: <https://public-ui.github.io/en/blog/2023/06/28> (дата обращения: 10.05.2025)
4. Design Systems // Nielsen Norman Group [Электронный ресурс]. – URL: <https://www.nngroup.com/articles/design-systems-101/> (дата обращения: 14.05.2025)

