

Ишбулдин Ильгиз Илдусович, магистрант, 2 курс.

Институт дизайна и пространственных искусств
Казанский (Приволжский) федеральный университет, г. Казань

e-mail: Igizz.ish@gmail.com

ВЛИЯНИЕ ТЕХНИЧЕСКОЙ ГРАМОТНОСТИ НА ЭФФЕКТИВНОСТЬ UX/UI-ДИЗАЙНА

Аннотация. В статье рассматривается влияние технической подготовки UX/UI-дизайнеров (в том числе знания HTML/CSS/JS и алгоритмического мышления) на эффективность их работы. На основе анализа современных зарубежных исследований и практических кейсов показано, что техническая грамотность дизайнеров положительно сказывается на качестве дизайна, оптимизации процессов передачи макетов разработчикам, снижении количества ошибок и итераций в проекте, улучшении коммуникации в команде и росте востребованности специалистов. Приводятся количественные показатели эффективности (например, до 200% рост конверсии при качественном UX; до 50% сокращение времени разработки при использовании дизайн-систем; 80% опрошенных отмечают повышение продуктивности работы при использовании инструментов Dev Mode). Отмечены ограничения существующих данных и сделаны выводы о необходимости дальнейших исследований.

Annotation. This article examines how a UX/UI designer's technical literacy (knowledge of HTML/CSS/JS and algorithmic thinking) affects design efficiency. Based on recent international studies and case analyses, it is shown that technical skills improve design quality, streamline developer handoff, reduce implementation errors and rework, enhance team communication, and increase market demand for designers. Quantitative metrics from industry cases are cited (e.g. effective UX can boost conversion by up to 200%; using design systems can halve development time; 80% of HP designers reported efficiency gains with Figma Dev Mode). Limitations

of current evidence are discussed and recommendations for further research are given.

Ключевые слова: UX-дизайн; техническая грамотность дизайнера; handoff-процессы; дизайн-системы; коммуникация в команде; эффективность дизайна; ошибки реализации.

Keywords: UX design; designer technical literacy; handoff process; design systems; team communication; design efficiency; implementation errors.

Современные цифровые интерфейсы всё чаще создаются в условиях мультидисциплинарных команд, где дизайнеру необходимо взаимодействовать с разработчиками, продукт-менеджерами и аналитиками. В такой среде успешная реализация интерфейса зависит не только от визуального и пользовательского качества макета, но и от его технологической реализуемости. Недостаточное понимание принципов HTML, CSS и JavaScript приводит к тому, что дизайнер создаёт макет, не учитывающий ограничения платформы, особенности DOM-структуры, правила адаптивности или специфику браузерного рендеринга [1].

В результате handoff превращается в источник конфликтов и доработок: по данным Nielsen Norman Group [1], в 68% UX-проектов значительная доля времени тратится на уточнение деталей между дизайнерами и разработчиками. Такие потери напрямую связаны с отсутствием технической грамотности у дизайнеров. Как отмечают специалисты Toptal и UX Planet [4, 5], знание базовых технологий верстки и скриптов — не просто технический навык, а способ мышления, позволяющий строить дизайн, учитывая будущую реализацию.

Целью настоящей статьи является системный анализ влияния знаний HTML, CSS и JavaScript на эффективность дизайнерской работы, включая:

- сокращение количества итераций между дизайном и кодом;
- повышение согласованности макета с итоговой версией продукта;

- ускорение коммуникации между дизайнером и разработчиком;
- увеличение производительности команды и снижение стоимости разработки.

Знание HTML позволяет дизайнеру мыслить не только в терминах визуального порядка, но и в логике семантической структуры: понимать, как интерфейс будет реализован в DOM-дереве, какие элементы допустимы в разных блоках, как устроена вложенность. Например, навигационные панели, карточки, формы имеют специфическую структуру тегов, и их корректное проектирование дизайнером предотвращает типовые ошибки при вёрстке. Дизайнер, знающий HTML:

- создает макеты, соответствующие реальной структуре страницы;
- проектирует элементы так, чтобы они соответствовали доступным и SEO-оптимизированным шаблонам;
- избегает избыточной или некорректной вложенности, что упрощает вёрстку и повышает производительность фронтенда.

CSS определяет внешний вид и поведение интерфейсных компонентов в зависимости от взаимодействия пользователя, размеров экрана и состояния элементов. Дизайнер, знакомый с возможностями CSS, такими как Flexbox, CSS Grid, media queries, псевдоклассы и переменные, способен сразу создавать макеты, соответствующие поведению реального интерфейса при изменении условий отображения.

Знание Flexbox и Grid помогает проектировать компоненты с учётом логики потока документа: дизайнер может учитывать, как элементы выстраиваются в строку или колонку, как они адаптируются к разным ширинам и как их поведение регулируется через свойства выравнивания и распределения пространства. Media queries позволяют заранее проработать поведение интерфейса на разных разрешениях экрана (например, mobile-first или desktop-first подход), а использование псевдоклассов и псевдоэлементов

помогает учитывать интерактивные состояния: `hover`, `focus`, `active`, `disabled` и др. Дизайнер, обладающий этими знаниями:

- учитывает ограничения CSS при проектировании визуальных решений;
- оптимизирует количество уникальных стилей, снижая сложность стилей при реализации;
- проектирует модули, которые легко повторно использовать и масштабировать;
- прорабатывает логику breakpoint'ов без необходимости постоянной консультации с верстальщиком;
- избегает эффектов, плохо реализуемых в CSS без костылей (например, сложных клиппингов, нестандартных `blend`-режимов).

Также дизайнер способен рационально использовать возможности CSS-анимаций. Понимание принципов `keyframes`, `transitions` и `performance-friendly` эффектов (`opacity`, `transform`) позволяет не перегружать интерфейс. Например, дизайнер может отказаться от ресурсоёмких эффектов вроде `blur`-фонов с параллаксом, которые плохо оптимизируются в браузере и могут снижать FPS на мобильных устройствах. Такая осведомлённость влияет на общее качество и стабильность интерфейса, особенно в проектах с высокими требованиями к производительности.

JavaScript определяет интерактивность и динамическое поведение пользовательского интерфейса, и именно на уровне сценариев JS реализуется подавляющее большинство логики взаимодействия. Знание дизайнером основ JavaScript, включая событийную модель (`event listeners`), управление состояниями компонентов, `DOM-manipulation` и понимание типов данных, позволяет строить интерфейсы, которые логично и корректно функционируют в браузере. Дизайнер, обладающий базовыми знаниями JavaScript:

- осознаёт, какие UI-решения требуют сложной реализации (например, drag-and-drop, multi-level меню, типизированные формы);
- способен проектировать логику поведения элементов с учётом ограничений JS-фреймворков и браузерной совместимости;
- лучше структурирует интерфейс, избегая лишней вложенности, не поддающейся компонентной декомпозиции;
- работает с пониманием принципов обработки ошибок, debounce/throttle, жизненного цикла компонента и асинхронного взаимодействия.

Особую важность представляет алгоритмическое мышление. Проектируя сложные пользовательские сценарии (например, фильтрацию, пошаговую форму, настройку персонализированных карточек), дизайнер должен понимать, какой алгоритм лежит в основе работы. Непродуманный UI может влечь за собой реализацию ресурсоёмкого JavaScript-кода, создающего нагрузку на клиентскую сторону, особенно на мобильных устройствах.

Чем лучше дизайнер разбирается в алгоритмических паттернах (условные конструкции, циклы, сортировка, вложенные состояния), тем проще разработчику перевести его макет в код без конфликтов. Таким образом, дизайнер формирует архитектуру взаимодействия ещё на этапе проектирования, а не перекладывает её полностью на плечи frontend-инженера.

По данным Figma [2], дизайнеры, владеющие сценарным мышлением и пониманием компонентных состояний, создают прототипы, которые проще автоматизировать и интегрировать в продукт. Они быстрее объясняют логику поведения интерфейса и корректно описывают user flows, особенно в сложных сценариях с переменными зависимостями. Это снижает количество итераций между дизайном и разработкой, делает процессы более предсказуемыми и контролируемыми.

Согласно кейсу HP с внедрением Figma Dev Mode [2], технически подкованные дизайнеры сэкономили до 98 минут в неделю на уточнение макетов. Это стало возможно благодаря тому, что дизайнер сразу указывал логическую структуру, возможные состояния и типы компонентов, а также использовал переменные и токены, которые разработчики могли перенести в код без изменений.

Передача макетов, оформленных с учётом технической реализации, сокращает количество ошибок. Как показано в отчете Figma [3], автоматическая генерация CSS/HTML из компонентов упрощает жизнь разработчикам, а дизайнеры, понимающие, как будет выглядеть конечный код, избегают дизайнерских решений, сложных или невозможных в реализации.

По оценке UX Planet [5], команды, в которых дизайнеры владеют HTML/CSS, работают на 25–40% быстрее за счёт сокращения времени на handoff, багфиксы и доработки. Дизайнер может участвовать в pull request-обсуждениях, корректно формулировать баги и пожелания к верстке, и при необходимости самостоятельно править текст или токены в компонентах.

Toptal отмечает, что UX-инженеры — дизайнеры с навыками front-end — получают в среднем на 25–30% больше по сравнению с классическими UX-специалистами [4]. Более того, количество вакансий, предполагающих знание кода у дизайнеров, выросло на 140% с 2020 по 2024 год. Таким образом, техническая грамотность становится фактором карьерного роста.

Владение HTML, CSS и JavaScript дизайнером не означает необходимость написания продакшн-кода, но обеспечивает ключевое конкурентное преимущество. Это:

- повышает качество макетов и делает их ближе к реальному продукту;
- снижает трения при взаимодействии с разработчиками;
- ускоряет вывод продукта на рынок за счёт меньшего числа итераций;

- делает дизайнера более ценным членом кросс-функциональной команды.

Литература

1. Rosala M., Krause R. User Experience Careers: What a career in UX looks like today [Электронный ресурс] // Nielsen Norman Group. 2020. – 90 с. – URL: https://media.nngroup.com/media/reports/free/UserExperienceCareers_2ndEdition.pdf (дата обращения: 15.06.2025).
2. Figma. Codegen plugins (and other tips) for automating design to code [Электронный ресурс] // Figma Blog. 2024. – URL: <https://www.figma.com/blog/figma-dev-mode-codegen-plugins> (дата обращения: 15.06.2025).
3. Sjölander E., Albaugh J. What codegen is (actually) good for [Электронный ресурс] // Figma Blog. 2023. – URL: <https://www.figma.com/blog/what-codegen-is-actually-good-for> (дата обращения: 15.06.2025).
4. Vieira J. Coding for designers: How much should we know? [Электронный ресурс] // Toptal. 2019. – URL: <https://www.toptal.com/designers/ux/designers-coding> (дата обращения: 15.06.2025).
5. Philips M. Coding for designers — How much should we know? [Электронный ресурс] // UX Planet. 2019. – URL: <https://uxplanet.org/coding-for-designers-how-much-should-we-know-78704d4dba4f> (дата обращения: 15.06.2025).