

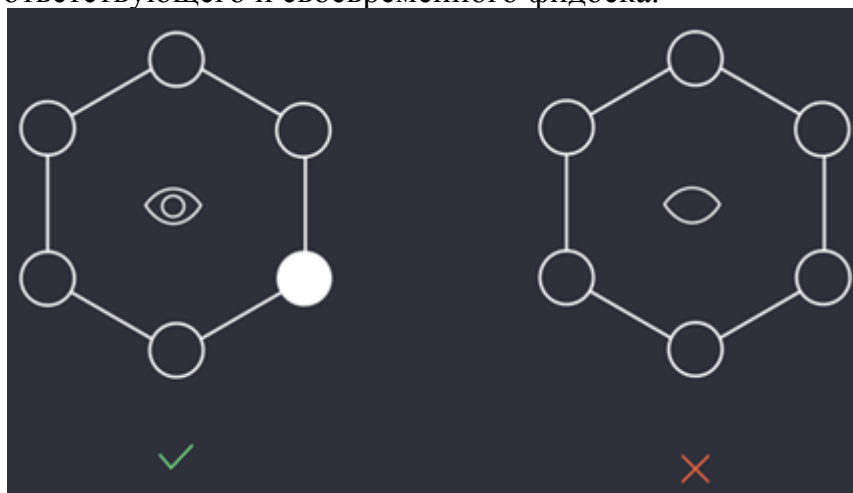
Вопросы разработки интерфейса

«10 эвристик для проектирования интерфейса».

В дизайне пользовательского интерфейса есть правила, которые не стоит нарушать. Их называют «эвристикой» или общими принципами, улучшающими юзабилити интерфейсов. Это повторяемые паттерны, проверенные временем и помогающие пользователям ориентироваться в интерфейсе. Хорошо спроектированный интерфейс всегда учитывает изложенные ниже принципы. В не очень хорошо спроектированном интерфейсе наверняка не хватает одного или нескольких принципов. Если вы планируете быть UI дизайнером, то необходимо знать, что не стоит нарушать и создавать проблемы для пользователей?

1. Видимость состояния системы

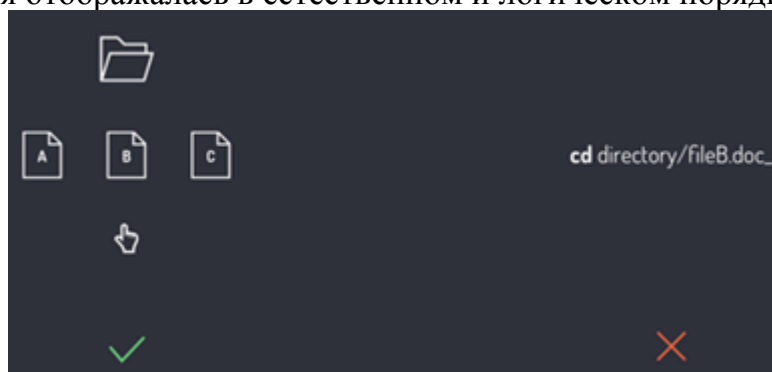
Система всегда должна информировать пользователей о том, что происходит, посредством соответствующего и своевременного фидбека.



Всегда предоставляйте пользователям соответствующую информацию, подсказки и контекст, чтобы они знали свое местоположение в системе. Это позволяет пользователю ощущать контроль и знать, что делать дальше. Товар был добавлен в корзину? Правка была сохранена? Сколько времени займет этот процесс? Каков статус моего заказа? Что сейчас происходит? Всегда отвечайте на подобные вопросы пользователям и никогда не оставляйте их в неведении и не заставляйте гадать.

2. Соответствие между системой и реальным миром

Система должна говорить на языке пользователей, используя знакомые им слова, фразы и концепции, а не системные термины. Следуйте правилам реального мира, чтобы информация отображалась в естественном и логическом порядке.



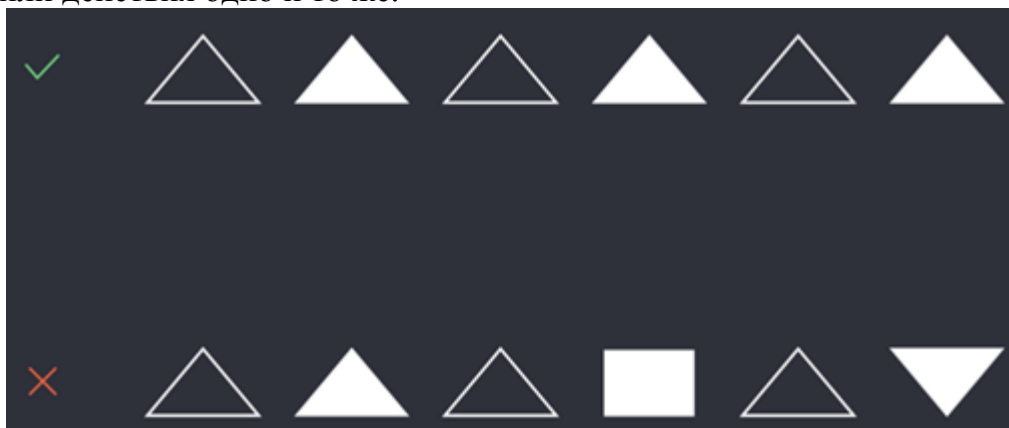
Используйте знакомые слова и язык. Не усложняйте формулировку. Значение слова или иконки на экране должны быть понятны вашей целевой аудитории. Люди

приходят на ваш сайт или в приложение, имея сформированные ментальные модели и опыт, позволяющим им интерпретировать паттерны.

Одно из величайших достижений в технологии произошло с появлением графического пользовательского интерфейса (GUI). До него экран компьютера был ограничен непонятными текстовыми командами, которые было нужно запоминать и повторять всякий раз, когда вы хотели выполнить действие. Потом все изменилось. На экране отображались маленькие изображения папок и файлов и курсор в виде руки. Все это были визуальные символы, которые люди мгновенно понимали. Их не нужно объяснять, потому что они ссылаются на ментальные модели реального мира.

3. Последовательность и стандарты

Пользователи не должны задаваться вопросом, означают ли разные слова, ситуации или действия одно и то же.



Есть два типа последовательности: внутренняя и внешняя. Внутренняя последовательность относится к паттернам на вашем сайте или в приложении. Она может быть простой, например, ссылки одного цвета на всех страницах или одна иконка для одной концепции. Внешняя последовательность относится к соглашениям, применяемым в других программах и системах, используемых большинством людей. Например, корзина для покупок. Большинство людей знакомы с принципом работы корзины покупок. Не нужно изобретать велосипед. В противном случае пользователям будет сложнее узнать, как работает ваша корзина. Сохраняйте последовательность и избавьте пользователей от ненужной путаницы.

4. Контроль и свобода пользователей

Пользователи часто выбирают системные функции по ошибке, и им потребуется четко обозначенный «аварийный выход», чтобы выйти из нежелательного состояния без необходимости проходить расширенный диалог. Добавьте функции отмены и повтора.

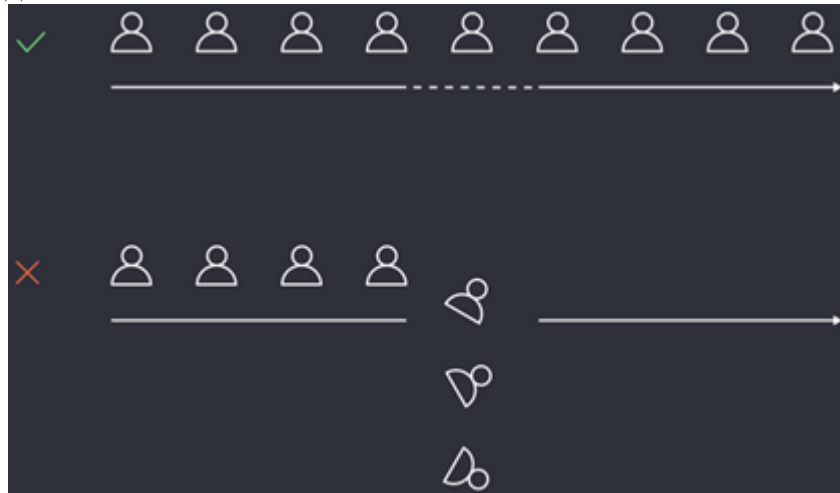


Всегда предоставляйте выход. Никогда не заставляйте пользователей выполнять ненужную функцию, и не заводите их в тупик. Например, если вы проектируете схему

оформления заказа, позвольте пользователям продолжить делать покупки, если они того пожелают. Если они попытались совершить действие в приложении, позвольте отменить действие, если в последнюю минуту они засомневаются.

5. Предотвращение ошибок

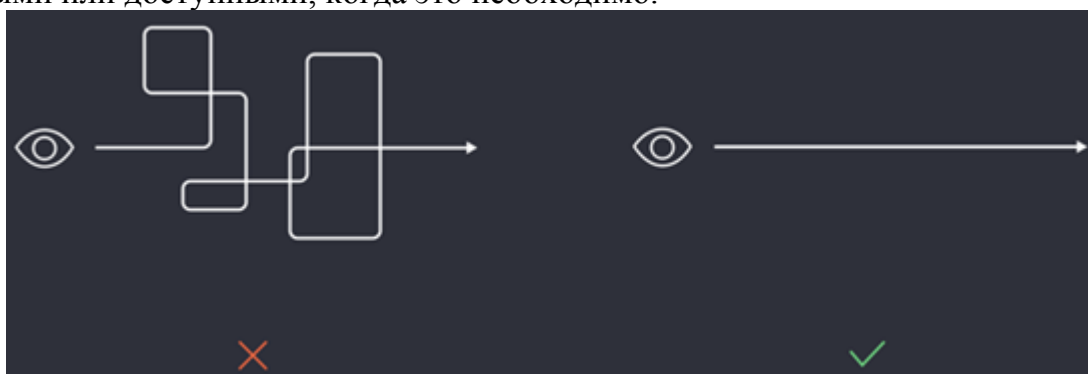
Лучше хорошего сообщения об ошибках только тщательно продуманный дизайн, который в первую очередь предотвращает возникновение проблемы. Либо устраните условия, подверженные ошибкам, либо проверьте их и предоставьте пользователям подтверждения действия.



Когда системные операции критически важны, например, удаление файлов или рассылка письма 1000 получателям, убедитесь, что пользователи знают, что они делают нечто важное. Прежде чем совершить действие, покажите им диалоговое окно подтверждения или предоставьте дополнительную информацию, четко определяющую, что произойдет. Это помешает им двигаться дальше, если они не уверены в своих действиях. Кроме того, это избавит их от сожаления.

6. Пользователи должны узнавать, а не вспоминать

Сведите к минимуму нагрузку на память пользователя, сделав объекты, действия и параметры заметными. Пользователь не должен вспоминать информацию из одной части диалогового окна в другой. Инструкции по использованию системы должны быть заметными или доступными, когда это необходимо.



Одна из целей UI дизайнера – снизить когнитивную нагрузку на пользователей. Психическая память – это огромный ограниченный ресурс. Память работает двумя способами: узнавание и воспоминание. Узнавание – это то, что вам сразу знакомо. Как лицо человека. Вы смотрите на лицо друга и сразу понимаете, что видели его раньше. Механизм воспоминания работает иначе. Это то, что вам нужно извлечь из памяти, например, имя человека. Воспоминание обычно требует больше времени и усилий, потому что разуму нужно обрабатывать больше информации, чтобы расшифровать то, на что он смотрит. С другой стороны, узнавание происходит мгновенно. Мы хотим

больше узнаваемости в интерфейсе и меньше воспоминания. Хороший пример этого принципа – использование для функций универсально узнаваемых кнопок и иконок, например, дом для «Домой» или карандаша для «Редактировать». А если вам нужно спроектировать для интерфейса новые иконки, которые большинство людей никогда раньше не видели, используйте текстовый дескриптор, чтобы объяснить их и уменьшить когнитивную нагрузку.

7. Гибкость и эффективность использования

Ускорители, невидимые для начинающего пользователя, часто могут ускорить взаимодействие для опытного пользователя, потому что система может обслуживать как неопытных, так и опытных пользователей. Разрешите пользователям настраивать частые действия.



Когда в приложении или системе определенные задачи повторяются снова и снова, вы можете сделать взаимодействие более эффективным для пользователей. Например, используйте в мобильном приложении свайп, чтобы сохранить или удалить элементы из списка. Обычный способ удалить элемент – открыть его, а затем нажать кнопку «Удалить». Продвинутый (и более эффективный) способ – просто свайпнуть и мгновенно удалить элемент из списка.

8. Минималистский дизайн и эстетика

Диалоговые окна не должны содержать неактуальную или редко используемую информацию. Каждая дополнительная единица информации в диалоговом окне конкурирует с релевантными единицами информации и снижает их относительную заметность.

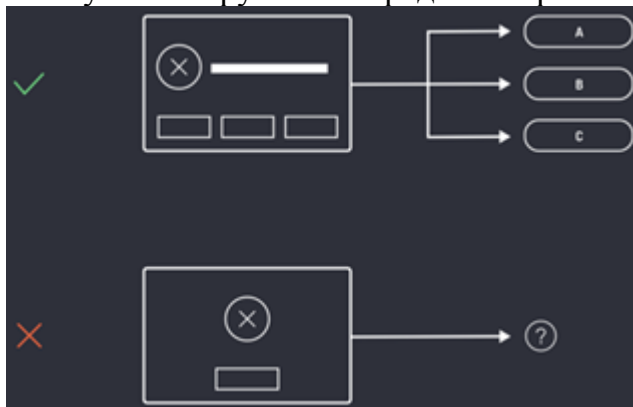


При проектировании для искусства не имеет значения, идем ли мы в сторону барокко и заполняем экран артефактами, текстурами и изображениями. Но при проектировании взаимодействия мы стремимся снизить соотношение сигнал / шум. Это делает интерфейс более понятным для пользователей. Вы можете применить этот принцип, просто уменьшив до минимума контент, отображаемый на экране, будь то

изображения или текст, чтобы пользователь мог не отвлекаясь сосредоточиться на текущей задаче.

9. Помогите пользователям распознавать, диагностировать и устранить ошибки

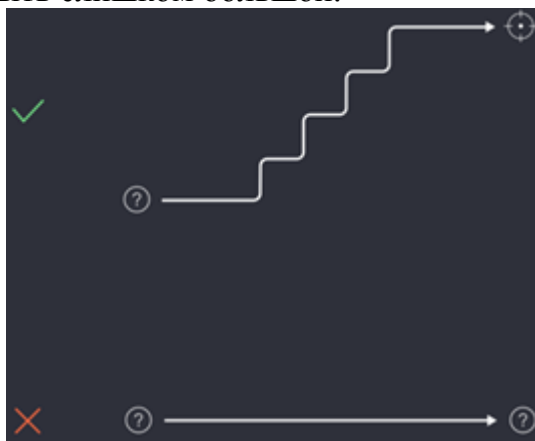
Сообщения об ошибках должны быть изложены простым языком (без кодов), точно указывать на проблему и конструктивно предлагать решение.



Ошибки будут. Это неизбежно. Обязанность UI дизайнера, определить, что произойдет после того, как пользователь обнаружит ошибку. Таким образом, мы можем помочь пользователям, проектируя понятные страницы ошибок и предупреждения, предоставляющие варианты решения проблемы. Например, давайте рассмотрим широко распространенную страницу 404. Мы, как дизайнеры, знаем, что означает страница с ошибкой 404, но обычно пользователи этого не знают. Чтобы помочь им, мы должны перевести код 404 на простой язык, добавив текст, поясняющий, что только что произошло. Например: «Извините, но мы не смогли найти страницу, которую вы искали. Вот несколько страниц с похожим контентом...».

10. Справка и документация

Хотя лучше, если систему можно использовать без документации, пользователю может потребоваться помощь. Подобная информация должна быть удобной для поиска, ориентированной на задачу, содержать список конкретных шагов, которые необходимо выполнить, и не должна быть слишком большой.



Сделайте помощь и справку всегда доступной. Разместите ее на видном месте на верхней панели или в основной области навигации. Когда пользователи сталкиваются с проблемой и не могут легко найти решение, их необходимо направить в раздел, где они смогут его найти. Это может быть страница часто задаваемых вопросов с окном поиска, содержащим возможные предложения и ответы. В случае отсутствия ответа система должна предоставить возможность напрямую связаться со службой поддержки для

получения дополнительной помощи, либо через систему тикетов, либо по электронной почте, либо по телефону.

Для повышения точности оценки сроков и стоимости работ по фронтенд-разработке необходимо выявить основные требования к разработке интерфейса.

Вопросы для выявления основных требований к разработке веб-интерфейса.

1. Какого типа требуется вёрстка?

- Фиксированная. Ширина сайта не изменяется на разных разрешениях экрана.
- Резиновая. Ширина сайта изменяется в зависимости от ширины экрана пользователя, сайт «растягивается».
- Адаптивная (adaptive). При изменении ширины окна браузера интерфейс перестраивается только в определенных точках (например 320 px, 768 px, 1024 px). Промежуточные состояния интерфейса не прорабатываются.
- Отзывчивая (responsive). Уделяется внимание всем возможным размерам экранов. Интерфейс изменяется, когда это необходимо с точки зрения UX/UI, а не в определенных точках.

2. Для каких разрешений нарисованы дизайн-макеты?

Если на сайте не требуется адаптивность, то скорее всего будет достаточно только одного разрешения макетов. Но если необходимо делать адаптивный интерфейс, то из дизайна должно быть понятно, как ведут себя элементы интерфейса на разных разрешениях экрана. В идеале требуются макеты с разрешением 320 px, 768 px, 960 px, 1280 px.

Отсутствие проработанных разрешений повлечет дополнительные трудозатраты на продумывание разработчиком поведения элементов, вопросы заказчику и дизайнеру.

3. Какие браузеры и платформы требуется поддерживать?

Чем больше браузеров и платформ необходимо поддерживать, тем больше требуется времени на тестирование и отладку интерфейса, тем дороже будет работа.

Есть сильно устаревшие версии браузеров, например, Internet Explorer 8, поддержка которых приведет к серьезному, иногда экспоненциальному, росту трудозатрат и сильным изменениям в технологии разработки интерфейса.

Если интерфейс разрабатывается для уже существующего сайта, то разумно будет посмотреть текущую статистику посещений и выбрать поддерживаемые версии браузеров.

Если статистика отсутствует, то можно спрогнозировать версии браузеров исходя из целевой аудитории.

Например, во многих государственных учреждениях установлено устаревшее ПО и потребуется поддержка старых версий браузеров.

Если целевая аудитория «обычная», то можно использовать общемировую статистику и поддерживать только относительно свежие браузеры.

- Chrome, Firefox, Safari, Edge >0,1 % использований;
- Android Chrome, iOS Safari >0,1 % использований;
- Internet Explorer 11.

4. В какой программе разработаны исходники дизайн-макетов?

Adobe Photoshop, Adobe XD, Sketch, Figma...

В зависимости от формата исходных макетов дизайна сайта предъявляются различные требования для работы с ними. Например, Sketch запускается только на

MacOS, поэтому если разработчик работает на Windows, то требуется совершить действия для экспорта макетов в другие приложения.

5. Необходимо ли следовать практике Pixel Perfect при вёрстке макетов?

По умолчанию мы допускаем погрешность до 5 пикселей в размерах блоков отступов и плашек от дизайна. Однако, готовы подогнать вёрстку под макет со 100 % соответствием.

6. Необходимо ли прорабатывать разметку для поисковых систем и социальных сетей?

Для повышения качества позиционирования сайта в выдаче поисковых систем и получения дополнительных возможностей для представления карточки сайта в результатах поиска для доработки HTML-структуры используется стандарт семантической разметки Schema.org.

Подобным же образом решается задача корректной публикации материалов сайта в социальных сетях, для дополнительной разметки HTML-структуры используются протоколы Open Graph, Twitter Card и другие.

7. Какой уровень доступности интерфейса необходимо обеспечить?

Доступность интерфейса для людей с ограниченными возможностями можно прорабатывать с различным уровнем качества. Работа заключается в соблюдении рекомендаций WCAG 2.0 при проектировании и разработки дизайна, написании HTML, CSS, JS кода. Прорабатывается навигация по интерфейсу с клавиатуры и корректность озвучки сайта с помощью скринридера.

Для сайтов органов власти РФ предъявляются дополнительные требования к цветовым схемам, размерам шрифтов и прочему.

8. Какие есть технические требования к разработке интерфейса?

Технологии, инструменты, стиль кода и прочее. Где должна быть организована инфраструктура с репозиторием и дев-сервером.

9. Возможно изучить исходный код проекта?

Если работа подразумевает развитие уже существующего проекта, то необходимо провести аудит имеющейся разработки на предмет определения сложности погружения и внесения изменений в проект.

Контрольные вопросы:

1. Принципы улучшения юзабилити интерфейсов;
2. Основные требования к фронтенд-разработке интерфейса (типы вёрстки подробно).