МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КубГУ»)

Факультет компьютерных технологий и прикладной математики
Кафедра информационных технологий

ДОКЛАД

По теме «UX/UI»

по дисциплине «Спецсеменар»

Работу выполнил студент группы ПМ63 Брезицкий С.О.

Работу принял зав. каф. ИТ Подколзин В.В.

Краснодар-2025

UX — это User Experience (дословно: «опыт пользователя»). То есть это то, какой опыт/впечатление получает пользователь от работы с вашим интерфейсом. Удается ли ему достичь цели и на сколько просто или сложно это сделать.

A UI — это User Interface (дословно «пользовательский интерфейс») — то, как выглядит интерфейс и то, какие физические характеристики приобретает. Определяет, какого цвета будет ваше «изделие», удобно ли будет человеку попадать пальцем в кнопочки, читабельным ли будет текст и тому подобное…

Прототип IT-проекта можно сравнить с макетом здания, который создает архитектор задолго до начала строительных работ. Прототип – это «скелет» будущего продукта, интерактивная модель проекта, с помощью которой прорабатывается положение и поведение всех его элементов.

Прототип позволяет попробовать продукт в действии до его непосредственного создания, проработать интерфейс всех основных экранов, «покликать» кнопки и оценить, на сколько он будет удобен конечным пользователям.

Прототип позволяет

Заказчику:

- понять, насколько полно разработчики понимают идею и учтены ли все его пожелания;
- увидеть продукт глазами будущих пользователей и окончательно определиться с функционалом и интерфейсом;
- обезопасить себя от финансовых и временных затрат в случае несоответствий желаемого и действительного.

Разработчику:

- ставить программистам и дизайнерам четкие задачи;
- определить объем работ, их стоимость и сроки реализации;
- обезопасить себя от неправомерных требований заказчика.

Прототип низкой точности (скетч, бумажный)

Плюсы

- Дешевые, быстро создаются.

- Позволяют итерировать в realtime.
- Могут быть созданы кем угодно, независимо от навыков.
- Позволяют быстро проверить идеи разного уровня абстракции и жизнеспособности.

Минусы

- Могут быть сильно далеки от реальности.
- Почти не подходят для тестирования на реальных пользователях.
- Не подходят для тестирования сложных гипотез.

Прототип средней точности (medium fidelity, wireframe, interaction map, storyboard)

Wireframes — это организация полного функционала конечного продукта в виде структуры с представлением элементов интерфейса и навигации, их взаимодействия друг с другом. Как правило, в wireframes не используются типографика, цвет и любые другие графические элементы оформления, так как основное внимание уделяется функциональности, поведению и содержанию.

Плюсы

- Все еще достаточно быстрый в создании.
- Хороший баланс между ценой создания и получаемым знанием.
- Позволяет проводить полноценные тесты с живой аудиторией.
- Позволяет проверить полноту проектируемого решения. Например, «как закрыть всплывающее окно если нет крестика?».
- Можно использовать как документацию к разработке продукта.

Минусы

- Слабо вовлекает в использование.
- Пользователю необходимо уметь абстрагироваться от визуальных ограничений.

Прототип высокой точности *(high fidelity, interactive)*

Такие прототипы часто ошибочно принимают за конечный продукт, потому что они наиболее близки к готовому продукту. Hi-fi-прототипы лучше всего передают опыт использования продукта и реальные функциональные возможности.

Они идеально подходят для обсуждения сложных частей продукта и полноценных тестов решения на конечных пользователях, хотя могут быть дорогостоящими и отнимать много времени на создание.

Плюсы

- Привычнее всего для пользователя, хорошо вовлекают.
- Лучше всего «продают» идею или решение.
- Подходит для тестирования сложных продуктов (или частей продукта).

Минусы

- Дорогие и долгие в создании.
- Их сложно интерпретировать тем, кто не погружен в продукт, могут увести «не туда» своей привлекательностью.

Карта диалоговых окон представляет собой на высоком уровне абстракции проект того, как будет реализован пользовательский интерфейс.

Цель карты диалоговых окон показать использование пользовательского интерфейса.

На ней отображаются элементы экранов или диалоговых окон системы и путь навигации по ним. В определенный момент времени на карте диалоговых окон только один элемент: диалоговое окно, меню, командная строка или дисплей сенсорного экрана, доступен для ввода пользователем информации или выполнения другого действия.

Пользователь может переходить от одного элемента к другому, если они связаны между собой действием в активной области ввода.

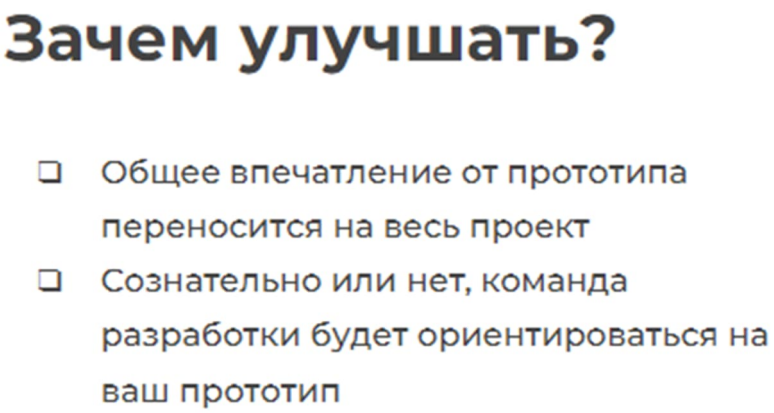
Карта диалоговых окон — это частный случай диаграммы переходов между состояниями. Она очень полезна для того, чтобы разработчики и бизнес-заказчики могли согласовать понимание, как пользователь может взаимодействовать с системой для выполнения задачи.

Количество возможных путей в сложном графическом интерфейсе велико, но оно конечно, и как правило, все возможности известны. Поэтому преимущество от использования карт диалоговых окон состоит в том, что в них

отражается суть взаимодействия пользователя и системы, без деталей макета, которые могут тормозить работу команды. С помощью такой карты можно отследить отсутствующие, неправильные или ненужные переходы, и, соответственно, отсечь неверные требования.

Особенно они полезны на семинарах по сбору информации, касающейся вариантов использования. Карта диалоговых окон — это прекрасный способ представить взаимодействия действующего лица и системы, описываемые вариантом использования.

Карта диалоговых окон позволяет отобразить альтернативные направления в виде ответвлений от нормального поведения. Пользователь, изучающий карту диалоговых окон, может обнаружит недостающее требование. Например, осторожный пользователь может захотеть подтвердить операцию, которая удаляет весь заказ, чтобы избежать непроизвольной потери данных.



Белое пространство

Белым пространством называют пространство между различными элементами пользовательского интерфейса. В буквальном смысле белое пространство не означает белый фон. Это может быть любая текстура или фоновое изображение.

Пустое пространство создает визуальную иерархию между различными элементами интерфейса, что помогает указывать пользователям на стратегические точки взаимодействия. Более того, негативное пространство сокращает информационный беспорядок на странице и делает контент легким на восприятие.

Каким бывает пустое пространство?

1. Микро и макро

Микробелое пространство — пространство между небольшими элементами, такими как буквы, текстовые строки, абзацы, иконки и кнопки.

Макробелое пространство — пространство между более крупными элементами: текстовые столбцы, графика, отступы, поля и т. д.

2. Активное и пассивное (другой способ классификации негативного пространства по способу привлечения внимания)

Активное пространство используется для привлечения внимания пользователей к определенным элементам, например, заголовок, логотип, изображение.

Пассивное пространство между небольшими объектами, которое остается незамеченным пользователем. Оно используется для того, чтобы пользователь мог легко прочитать дизайн и текст.

Белое (пустое) пространство — важнейший элемент в веб-дизайне, которому чаще всего не уделяют должного внимания. При правильном использовании негативное пространство создает эстетику на любом дизайн-макете и влияет на эффективность веб-страницы. Другими словами, оно улучшает ui (пользовательский интерфейс) и ux (пользовательский опыт) страницы.

Эвристики Нильсена

Эвристики Нильсена — это правила, которыми должны пользоваться дизайнеры для проектирования взаимодействия между интерфейсом и пользователями. Эвристики помогают выявить основные проблемы, с которыми пользователи могут столкнуться, и предотвратить их.

Были сформулированы в 1990 году.

1. Видимость статуса системы

Дизайн должен всегда информировать пользователей о том, что происходит в течение разумного периода времени.

Когда люди взаимодействуют с любой системой, она всегда должна обеспечивать немедленную обратную связь. Просто визуальный знак, такой как изменение цвета кнопки, индикатор загрузки или анимация значка может помочь пользователю понять, что происходит, уберегая его от других ненужных взаимодействий.

2. Соответствие между системой и реальным миром

Дизайн должен говорить на языке пользователей. Для интуитивного использования интерфейса его элементы должны быть схожими с теми, что мы уже использовали в жизни.

Крайним примером является скевоморфизм, который передает все детали объектов реального мира в программное обеспечение. В начале эпохи смартфонов он очень помогал людям привыкнуть к новым интерфейсам, сегодня же графика сильно упростилась.

3. Пользовательский контроль и свобода

Пользователи часто выполняют действия по ошибке. Им нужен четко обозначенный «аварийный выход».

Когда людям легко отказаться от процесса или отменить действие, это способствует чувству свободы и уверенности.

4. Последовательность и стандарты

Интерфейс никогда не должен сбивать пользователей с толку использованием разных слов, изображений или действий для одних и тех же понятий или ситуаций.

В основе 4-ой эвристики лежит закон о том, что большую часть времени пользователи проводят в других приложениях, а значит, для их комфортного взаимодействия с системой мы должны следовать гайдлайнам платформ (напр. Apple‘s Human Interface Guidelines и Google‘s Material Design Guidelines) и распространенным дизайн-решениям в нашей отрасли.

Эта эвристика также о согласованности внутри одного интерфейса. Простым языком, если вы используете определенный размер шрифта заголовка или цвет кнопки, то не нужно отходить от заданных параметров в другом месте интерфейса.

5. Предотвращение ошибок

Хорошие сообщения об ошибках важны, но лучший дизайн в первую очередь тщательно предотвращает возникновение проблем.

Используйте ограничения, которые не позволяют пользователю установить неправильное значение (например, когда вы ожидаете число, не разрешайте писать буквы), предложите наиболее распространенные варианты, чтобы облегчить пользователям выбор (например, при поиске), или использовать подтверждения перед деструктивными действиями.

6. Узнавание вместо припоминания

У людей ограничена кратковременная память. Интерфейсы, которые способствуют распознаванию, сокращают количество когнитивных усилий, требуемых от пользователей.

Минимизируйте нагрузку на память пользователя, сделав видимыми элементы, действия и параметры. Пользователь не должен запоминать

информацию из одной части интерфейса в другую. Информация, необходимая для использования дизайна (например, пункты меню), должна быть видна или легкодоступна при необходимости.

Когда в интерфейсе много полей для заполнения, названия полей лучше писать возле каждого поля а не внутри, так как человек может забыть какое поле он заполнял. Также вместо открытого вопроса лучше использовать вопрос с вариантами ответа (по возможности).

7. Гибкость и эффективность использования

Все пользователи уникальны и имеют разные потребности и навыки. Гибкий интерфейс позволяет выполнять одни и те же функции разными методами.

Например, кому-то проще нажать на крестик для закрытия попапа, а кому-то, чтобы не тратить на это время, быстрее нажать в пустое пространство или нажать Esc.

Новички какого-либо ПО будут пользоваться кнопками в интерфейсе, в то время как опытные ребята будут использовать shortcuts (комбинации клавиш на клавиатуре), чтобы ускорить свою работу.

8. Эстетичный и минималистичный дизайн

Интерфейсы не должны содержать неактуальную или редко используемую информацию. Каждая неактуальная единица информации в интерфейсе конкурирует с актуальными единицами информации и уменьшает их видимость.

Минимализм — не только мода последних нескольких лет, но и, безусловно, устойчивая тенденция, цель которой — свести описание предмета только к его необходимым элементам.

Антуан де Сент-Экзюпери "Совершенство достигается не тогда, когда больше нечего добавить, а когда нечего отнять".

9. Помогите пользователям распознавать, диагностировать и устранять ошибки

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

Каждое сообщение об ошибке должно быть максимально явным и точным. Никто не хочет читать расплывчатые сообщения вроде «что-то пошло не так». Изложите, что произошло доступным человеческим языком. Столь же абсурдны сообщения типа «Ошибка класса 372». Дайте пользователю конструктивный совет, что делать дальше. Предложите решение или направьте

пользователя к сотруднику службы поддержки, который сможет разобраться в ситуации. Последнее правило хороших сообщений об ошибках — вежливость. Никогда не обвиняйте пользователя и не подразумевайте, что он глуп.

10. Справочные материалы и документация

Лучше всего, если система не нуждается в дополнительных пояснениях. Однако может потребоваться предоставить документацию (частый пример туториал в приложении), чтобы помочь пользователям понять, как выполнять свои задачи.

Большинство пользователей пропускает обучение, предлагаемое при старте взаимодействия с системой, но это не значит, что оно не понадобится в будущем. Справочные материалы (например FAQ) должны быть легкодоступны и лаконичны. Будьте кратки и перечислите конкретные шаги, которые необходимо выполнить.