**Вопросы контрольного опроса № 1**

1. Сетевые СМИ: основные группы, виды и формы их функционирования.
2. Понятие и определение процесса проектирования. Дайте определение понятию проектирование взаимодействия.
3. Понятие и определение интерфейса. Этапы развития пользовательского интерфейса.
4. Типы пользовательского интерфейса.
5. Назначение и основные функции интерфейсов. Дизайн интерфейса, ориентированный на человека.
6. Общие принципы проектирования пользовательских интерфейсов.
7. Понятие и определение эргономики, ее подразделы.
8. Понятие и определение юзабилити. Юзабильный интерфейс.
9. Юзабилити и эргономика, в чем сходство и в чем различие.
10. Назовите 4 основные критерии качества интерфейса пользователя.
11. Назовите составляющие скорости выполнения работы и дайте краткую характеристику каждой составляющей.
12. Четыре типа человеческих ошибок при работе с интерфейсов.
13. Классификация ошибок при работе пользователя с интерфейсом по уровням их негативного эффекта и обратная связь.
14. Каковы средства обучения работе пользователя с интерфейсом?
15. Понятие ментальной модели.
16. Понятие метафоры. Правила использования метафор.
17. Использование стандартов для обеспечения «понятности» системы. Как может быть достигнута популярность стандарта?
18. Использование аффорданса для обеспечения «понятности» системы. Основные виды аффордансов.
19. Как можно повысить субъективную скорость работы?
20. Какие факторы влияют на субъективное удовлетворение?
21. Отчего зависит субъективное ощущение психологического напряжения пользователя?
22. Каким должно быть сообщение об ошибке??
23. Закон Хика и закон Фиттса. Особенности комплексного применении законов Хика и Фиттса в UX
24. Объяснить принцип работы модели количественного анализа интерфейсов GOMS.
25. Закономерности зрительного восприятия человеком информации.
26. Информационные процессы: память и мышление. Кратковременная и долговременная память.
27. Восприятие и представление информации. Их сходство и различие. Какие факторы влияют на успешное восприятие визуальной информации.
28. Основные свойства внимания.
29. Внимание. Фокус и локус внимания, в чем сходство и в чем различие.
30. Типы внимания: избирательное, фокусированное, распределенное. Когда каждый из них используется при работе с интерфейсом?
31. С помощью чего можно привлечь внимание пользователя?
32. Сколько элементов может запомнить КВП и как можно увеличить их число? Сколько времени хранится в кратковременной памяти информация, ставшая локусом внимания? И когда она из нее исчезает?
33. Кратковременная (КВП) и долговременная память (ДВП). Как информация попадает в КВП и ДВП?
34. Какие свойства интерфейса формируют привычки? Что дают пользователю привычки при работе с интерфейсом?
35. В чем суть гештальт-психологии, когда она возникла и кто был ее основоположником
36. Гештальт-принципы организации восприятия. Факторы, способствующие группировке элементов в целостные гештальты.
37. В чем суть гештальт-принципов близости, непрерывности, замкнутости и сходства и применение их в UX-дизайне.
38. Количественная оценка интерфейса с помощью метода GOMS. Правила расстановки ментального оператора.
39. **Сетевые СМИ: основные группы, виды и формы их функционирования.**

Сетевые СМИ:

• Информационно-новостные порталы

• Группы новостей и списки рассылки

• Читательские форумы при сайтах СМИ и чаты

• Гражданская журналистика (блоги, вики, ютуб)

• Информационно-поисковые системы (ИПС)

• Порталы

• Агентства

• Информационные службы

• Интернет-журналы

• Интернет-газеты

• Интернет-дайджесты.

**2 Понятие и определение процесса проектирования. Дайте определение понятию проектирование взаимодействия.**

Джесс Гаррет в своей книге «Веб-дизайн. Элементы опыта взаимодействия» представляет проектирование опыта взаимодействия в виде 5 уровней.

*Эти пять уровней:*

* стратегии (или идеи);
* набора возможностей;
* структуры;
* компоновки;
* поверхности (или внешнего вида интерфейса).

*Собственно проектирование состоит из следующих этапов:*

1 Определение необходимой функциональности системы

2 Создание пользовательских сценариев

3 Проектирование общей структуры

4 Конструирование отдельных блоков

5 Создание глоссария

***Проектирование******взаимодействия*** *(Interaction Design, IxD )* - выяснение, как система будет вести себя в ответ на действия пользователей.

***Сущность качественного проектирования взаимодействия состоит в том, чтобы позволить пользователям достигать практических целей, не отказываясь от целей личных.***

***Программа, которая не позволяет достичь какой-либо корпоративной или личной цели, потерпит неудачу.***

1. **Понятие и определение интерфейса. Этапы развития пользовательского интерфейса**.

**Интерфейс пользователя** (ПИ) представляет собой совокупность средств и методов, при помощи которых пользователь взаимодействует с различными устройствами и аппаратурой.

**Интерфейс** – только половина во взаимодействии с системой, другая половина – человек, пользователь.

Чтобы интерфейс работал хорошо, нужно знать, что пользователь:

* Воспринимает в интерфейсе
* Думает
* Хочет добиться.

Если интерфейс не изменяется, то можно модифицировать объект, не перестраивая принципы его взаимодействия с другими объектами.

ПИ:

* Интерфейс командной строки – инструкции компьютеру даются путем ввода с клавиатуры текстовых строк
* Графический интерфейс пользователя (или WIMP – интерфейс, Image – образ, Window – окно, Menu – меню, Pointer – указатель) – программные функции представляются графическими элементами экрана
* Естественно-языковой интерфейс (SILK – интерфейс, speech – речь, image – образ, ..) – пользователь разговаривает с программой на родном языке.

1. **Типы пользовательского интерфейса.**

* ***интерфейс командной строки*** – инструкции компьютеру даются путём ввода с клавиатуры текстовых строк (команд);

*В этом виде интерфейса человек подает "команды" компьютеру, а компьютер их выполняет и выдает результат человеку. Командный интерфейс реализован в виде пакетной технологии и технологии командной строки.*

* ***графический интерфейс пользователя*** (или ***WIMP –*** интерфейс: ***Window –***окно***, Image –***образ***, Menu –***меню***, Pointer –***указатель) – программные функции представляются графическими элементами экрана.

*Характерной особенностью этого вида интерфейса является то, что диалог с пользователем ведется не с помощью команд, а с помощью графических образов – меню, окон, других элементов. Хотя и в этом интерфейсе подаются команды машине, но это делается "опосредственно", через графические образы. Этот вид интерфейса реализован на двух уровнях технологий: простой графический интерфейс и "чистый" WIMP – интерфейс.*

* ***естественно–языковой интерфейс*** (или **SILK** – интерфейс: ***Speech*** – речь, ***Image*** – образ, ***Language*** – язык, ***Knowlege*** – знание) – пользователь «разговаривает» с программой на родном ему языке.

*Этот вид интерфейса наиболее приближен к обычной, человеческой форме общения.*

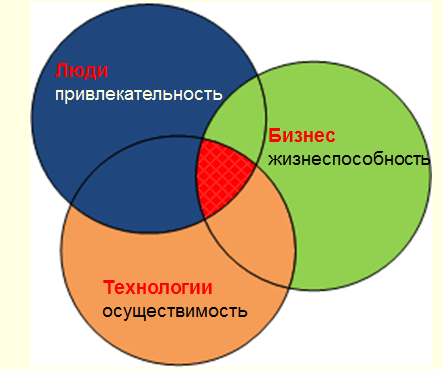
1. **Назначение и основные функции интерфейсов. Дизайн интерфейса, ориентированный на человека.**

**Основная аксиома ПИ: Хороший дизайн ПИ подразумевает, что программа соответствует ожиданиям пользователей о том, как она должна себя вести.** (Все остальное – следствия)

Дизайн – сознательные и интуитивные усилия по созданию значимого порядка.

Дизайн интерфейса позволяет придать определенный вид уже существующему поведению системы.

Дизайн ориентированный на человека должен попасть на перекрытие 3 линз: решения, которые привлекательны, возможны и жизнеспообны.



**6 Общие принципы проектирования пользовательских интерфейсов.**

Для того, чтобы интерфейс был эстетически привлекательным, необходимо, чтобы он был незаметен в процессе его использования.

***Во что бы то ни стало, добивайтесь того, чтобы интерфейс был неощущаем.***

Для этого:

- *Избегайте развязности в изображении Лучше, чтобы он был скромнее.*

- *Избегайте ярких цветов*. Существует очень немного цветов, обладающих и яркостью, и мягкостью (т.е. не бьющих по глазам). На экране их значительно меньше, поскольку в жизни такие цвета обычно моделируются как собственно цветом, так и текстурой, с чем на экране есть проблемы.

- *Избегайте острых углов в изображении.*

*- Старайтесь сделать изображение максимально более легким и воздушным.*

- *Старайтесь добиваться контраста не сменой насыщенности элементов, а расположением пустот*.

- *Старайтесь минимизировать количество констант (тем более, что двух констант обычно хватает на все*). Разумеется, единожды примененных закономерностей необходимо придерживаться во всей системе.

***Стремитесь не столько к красоте интерфейса, сколько к его элегантности***

***Красота понятие относительное***. Для одних красивыми могут считаться только живописные закаты, для других картины художника Кустодиева, а для третьих – комбинация вареных сосисок, зеленого горошка и запотевшей бутылки пива. Это делает красоту вещью не слишком универсальной. Хуже того. Любая красота со временем надоедает и в лучшем случае перестает восприниматься.

Именно поэтому в интерфейсах обычно не место красоте.

***Элегантность и гармония гораздо лучше.*** Во-первых, они не надоедают. Во-вторых, они редко осознается потребителями, обеспечивая неощущаемость. В-третьих – они приносят эстетическое удовольствие независимо от культурного уровня потребителя (так, древнегреческие и слегка менее древние римские здания воспринимаются нами красивыми, несмотря на абсолютную разницу культур и времени). В-четвертых, в производстве они гораздо удобнее красоты, поскольку сравнительно легко ставятся на поток.

Итак, каким образом надо действовать, чтобы добиться элегантности:

*Старайтесь сделать интерфейс максимально насыщенным визуальными закономерностями.* Есть универсальное правило – чем больше закономерностей, тем больше гармонии. Даже самые незначительные закономерности всё равно воспринимаются. Под закономерностью я понимаю любое методически выдерживаемое соответствие свойств у разных объектов, *например, высота кнопок может быть равна удвоенному значению полей диалогового окна*.

- *Всемерно старайтесь использовать модульные сетки*, т.е. привязывайте все объекты к линиям (лучше узлам) воображаемой сетки, которую выдерживайте во всем интерфейсе.

- *Старайтесь привязывать все размеры и координаты* (как минимум пропорции диалоговых окон) к золотому сечению (0.618 х 0.382).

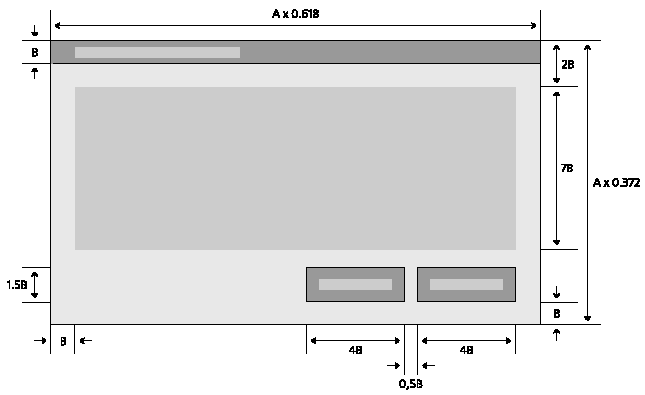


Рис. 10. Визуальный дизайн: использование компонентов

**7 Понятие и определение эргономики, ее подразделы.**

**Эргономика** – научная дисциплина, изучающая взаимодействие человека и других элементов системы, а также сферу деятельности по применению теории принципов данных и методов этой науки для обеспечения благополучия человека и оптимизации общей производительности системы.

Эргнономика изучает действия человека в процессе работы, скорость освоения им новой техники, затраты его энергии, производительность и интенсивность при конкретных видах деятельности.

Современная эргономика подразделяется на:

1. **Микроэргономика** занимается исследованием и проектированием систем «человек–машина». Интерфейсы ПО.

2. **Мидиэргономика** исследует производственные взаимодействия на уровне рабочих мест и производственных задач. Занимается проектирование систем «человек – коллектив», «коллектив–машина».

3. **Макроэргономика** исследует и проектирует систему в целом, учитывая все факторы. Целью является гармоничная, согласованная, надежная работа всей системы и всех элементов системы.

**Целью** эргономического дизайна является преодоление проблем в области человеко-компьютерного взаимодействия и оптимизация человеко-ориентированного интерфейса …

**8 Понятие и определение юзабилити. Юзабильный интерфейс.**

Наиболее часто используемое определение юзабилити:

**ЮЗАБИЛИТИ** – степень **эффективности, продуктивности и удовлетворенности**, с которыми продукт может быть использован определенными пользователями в определенном контексте использования для достижения определенных целей.

Три важных аспекта:

* **Определенные пользователи**: интерфейс всегда создается для конкретной группы людей, хар своими особенностями, знаниями, навыками, ожиданиями, сильными и слабыми сторонами.
* **Определенный контекст использования**: решения, которе подходят для пользователя идеально в одном контексте, мб совершенно неприемлемы в другом.
* **Определенные цели**: ни одно решение вв принципе не мб удачным, если оно не помогает достичь пользователю его целей.

*Что такое юзабильный интерфейс?*

Представители ит-индустрии соревновались между собой вначале за:

* Производительность «железа»
* Потом за функциональность софта
* Теперь за пользовательские качества – юзабилити.

Как пользователь видит интерфейс на экране, так он и воспринимает весь продукт в целом.

Целью практически любого разработчика является привлечение потенциального пользователя.

Пользователю важно получить наиболее позитивные чувства.

Основная цель – доставить наиболее приятные ощущения при взаимном обмене информацией во время работы с нашей системой.

**9 Юзабилити и эргономика, в чем сходство и в чем различие.**

**Эргономика и юзабилити:**

* **Эргономичность** описывает количество затраченных физических сил для работы с сайтом
* **Юзабилити** – общую степень удобства пользования, сумму умственных усилий, требующихся от пользователя для выполнения задачи.

Именно поэтому разработчики, как правило, делают акцент на «юзабельность» сетевых проектов.

Различие между эргономикой и юзабилити – первое ставит больший акцент на технические характеристики. Второе же – на процесс взаимодействия. В основе эргономики стоит сам продукт, в основе юзабилити – пользователь.

**10 Назовите 4 основные критерии качества интерфейса пользователя.**

Существует четыре основных критерия качества любого интерфейса, а именно:

* скорость работы пользователей,
* количество человеческих ошибок,
* скорость обучения,
* субъективное удовлетворение пользователей (подразумевается, что соответствие интерфейса задачам пользователя является неотъемлемым свойством интерфейса).

**11 Назовите составляющие скорости выполнения работы и дайте краткую характеристику каждой составляющей.**

***Непосредственное манипулирование***

Смысл этого метода очень прост. **Пользователь не отдает команды системе, а манипулирует объектами.** Первым популярным применением этого метода была корзина для удаления файлов в компьютерах *Macintosh* (начиная с *Windows 95*, такая корзина стала стандартом и в *Windows* мире, хотя присутствовала она и раньше). Если перетащить в неё пиктограмму файла, этот файл будет фактически стерт.

На самом деле процесс стирания файла, состоит из многих малых, уже не делимых, действий (жестов). При этом для ускорения мыслительной работы пользователя необходимо не только сокращать количество этих жестов, но и делать эти жесты более простыми.

***Применение в интерфейсе эффективных методов при потере фокуса внимания***

При работе с системой, пользователи постоянно отвлекаются.

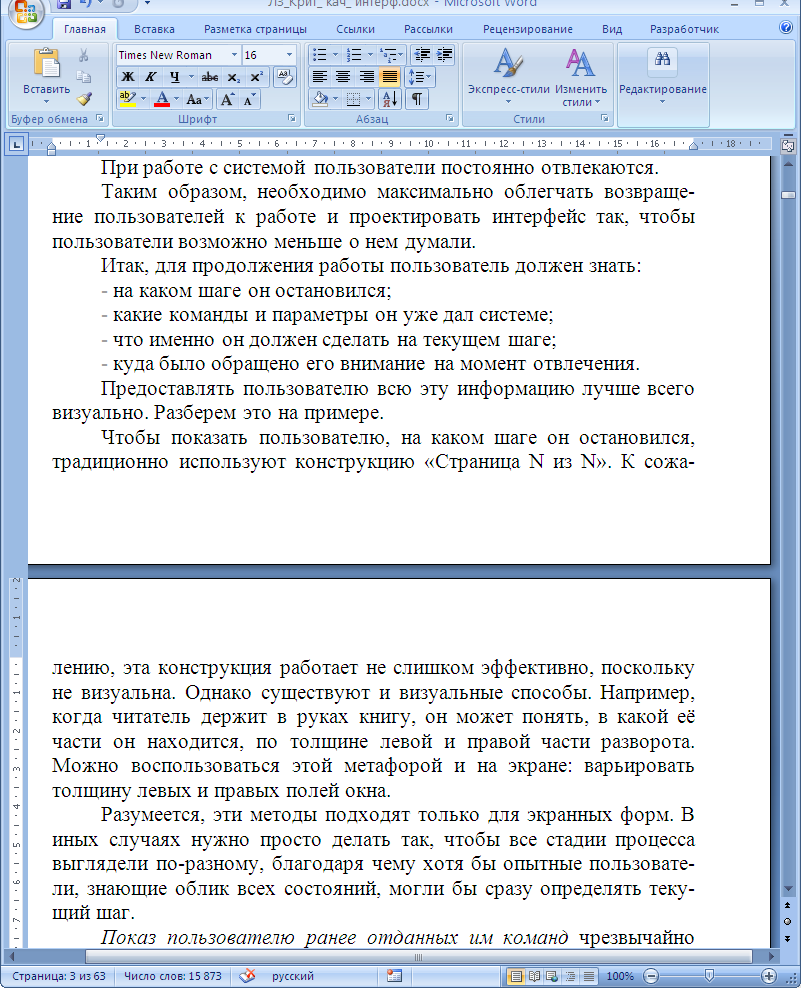
Таким образом, необходимо максимально облегчать возвращение пользователей к работе и проектировать интерфейс так, чтобы пользователи возможно меньше о нем думали.

Предоставлять пользователю всю эту информацию лучше всего визуально.

***Показ пользователю ранее отданных им команд***

Размеры экрана ограничены, так что единственным выходом из этого положения является максимальное облегчение перехода к предыдущим экранам, да и то это работает только с экранными формами.

Чтобы ***показать пользователю, на каком шаге он остановился***, традиционно используют конструкцию «Страница N из N».



И, наконец, четвертый пункт: показ пользователю, ***куда было обращено его внимание на момент отвлечения***. Обычно фокус внимания совпадает с фокусом ввода. Соответственно, нужно делать фокус ввода максимально более заметным, например цветовым кодированием активного элемента. Если количество элементов на экране невелико, пользователь быстро находит активный элемент. Таким образом, просто снизив насыщенность экрана элементами, можно значительно облегчить пользователю возвращение к работе.

**12 Четыре типа человеческих ошибок при работе с интерфейсов.**

***Ошибки, вызванные недостаточным знанием предметной области.*** Теоретически, эти ошибки методологических проблем не вызывают, сравнительно легко исправляясь обучением пользователей. Практически же, роль этих ошибок чрезвычайно велика – никого не удивляет, когда оператора радарной установки перед началом работы оператором долго учат работать, и в то же время все ожидают должного уровня подготовки от пользователей ПО, которых никто никогда ничему целенаправленно не обучал. Еще хуже ситуация с сайтами, у которых даже справочной системы почти никогда не бывает.

***Опечатки*** происходят в двух случаях:

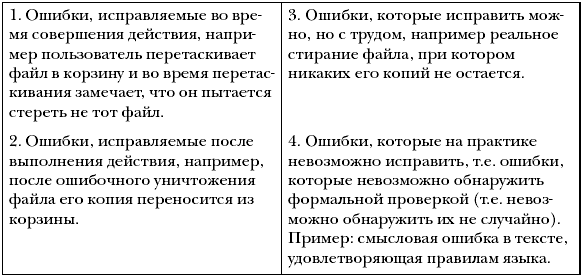
- когда не все внимание уделяется выполнению текущего действия (этот тип ошибок характерен, прежде всего, для опытных пользователей, не проверяющих каждый свой шаг);

- когда в мысленный план выполняемого действия вклинивается фрагмент плана из другого действия (происходит преимущественно в случаях, когда пользователь имеет обдуманное текущее действие и уже обдумывает следующее действие).

***Ошибки, вызванные не считыванием показаний системы***, которые одинаково охотно производят как опытные, так и неопытные пользователи. Первые не считывают показаний системы потому, что у них уже сложилось мнение о текущем состоянии, и они считают излишним его проверять, вторые – потому что они либо забывают считывать показания, либо не знают, что это нужно делать и как.

***Моторные ошибки,*** количество, которых фактически пренебрежимо мало, но к сожалению, не так мало, чтобы вовсе их не учитывать. Сущностью этих ошибок являются ситуации, когда пользователь знает, что он должен сделать, знает, как этого добиться, но не может выполнить действие нормально из-за того, что физические действия, которые нужно выполнить, выполнить трудно. Так, никто не может с первого раза (и со второго тоже) нажать на экранную кнопку размером 1 на 1 пиксель. При увеличении размеров кнопки вероятность ошибки снижается, но почти никогда не достигает нуля. Соответственно, единственным средством избежать этих ошибок является снижение требований к точности движений пользователя.

**13 Классификация ошибок при работе пользователя с интерфейсом по уровням их негативного эффекта и обратная связь.**



**14 Каковы средства обучения работе пользователя с интерфейсом?**

Рассмотрим:

- общая «понятность» системы

- обучающие материалы.

***Понятность системы***

Термин «понятность» включает в себя три составляющих, а именно ментальную модель, метафору, аффорданс и стандарт

***Ментальная модель***. Зачастую, или, точнее, почти всегда, чтобы успешно пользоваться какой-либо системой, человеку необходимо однозначно понимать, как система работает. При этом необязательно точно понимать сущность происходящих в системе процессов, более того, необязательно правильно их понимать. Это понимание сущности системы называется ментальной моделью.

***Метафора***. Разработать пользовательский интерфейс, в котором модель программы соответствует модели пользователя – задача не из легких. Иногда у пользователей просто нет конкретного представления о том, как работает программа и для чего она предназначена. В таком случае вам придется найти способ подсказать им, как функционирует ваша программа. В графических интерфейсах используется метод метафор.

***Аффорданс****.*  В современном значении этого термина аффордансом называется ситуация, при котором объект показывает субъекту способ своего использования своими неотъемлемыми свойствами.

***Стандарт***. Наконец, остался последний, самый мощный, но зато и самый ненадежный способ обучения, а именно стандарт.

Если что-либо нельзя сделать «самопроизвольно» понятным, всегда можно сделать это везде одинаково, чтобы пользователи обучались только один раз.

**Обучающие материалы**

*Что нам нужно и что у нас есть*

**Базовая справка** объясняет пользователю сущность и назначение системы. Обычно должна сработать только один раз. Как правило, не требуется для ПО, зато почти всегда требуется для сайтов.

**Обзорная справка** рекламирует пользователю функции системы. Также обычно срабатывает один раз. Нужна и ПО и сайтам. Поскольку у зрелых систем функциональность обычно очень велика, невозможно добиться того, чтобы пользователи запоминали её за один раз. В этом случае оптимальным вариантом является слежение за действиями пользователя и показ коротких реклам типа «А вы знаете, что…» в случае заранее определенных действий пользователей (примером такого подхода являются помощники в последних версиях MS Office).

**Справка предметной области** отвечает на вопрос «Как сделать хорошо?». Поскольку от пользователей зачастую нельзя рассчитывать знания предметной области, необходимо снабжать их этим знанием на ходу.

**Процедурная справка** отвечает на вопрос «Как это сделать?». В идеале она должна быть максимально более доступна, поскольку если пользователь не найдет нужную информацию быстро, он перестанет искать и так и не научится пользоваться функцией (возможно, никогда).

**Контекстная справка** отвечает на вопросы «Что это делает?» и «Зачем это нужно?».

**Всплывающие подсказки.** Хорошо справляются с ответом на вопросы «Что это такое» и «Зачем это нужно», при условии, что объем ответов сравнительно невелик. Они не занимают пространства экрана и не отвлекают внимания пользователей.

**Справка состояния** отвечает на вопрос «Что происходит в настоящий момент?». Поскольку она требуется именно что в настоящий момент, она не может быть вынесена из интерфейса. В целом это самая непроблематичная для разработчиков система справки, так что в этой книге разбираться она не будет.

**15 Понятие ментальной модели.**

***Ментальная модель***. Зачастую, или, точнее, почти всегда, чтобы успешно пользоваться какой-либо системой, человеку необходимо однозначно понимать, как система работает. При этом необязательно точно понимать сущность происходящих в системе процессов, более того, необязательно правильно их понимать. Это понимание сущности системы называется ментальной моделью.

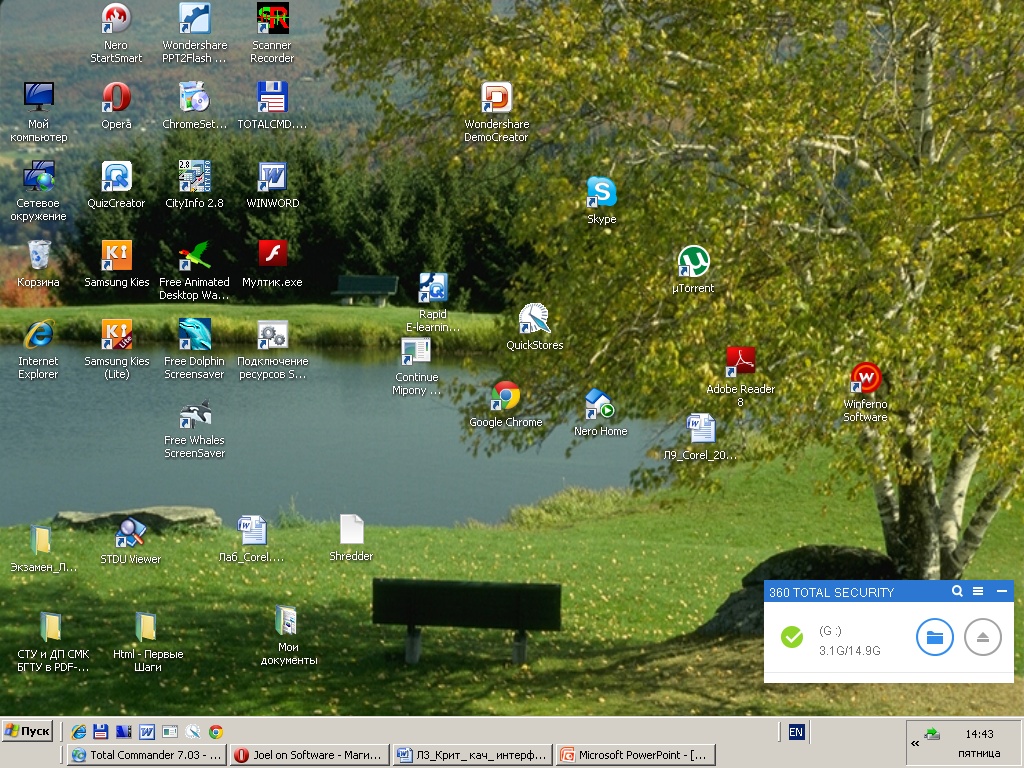
Утюгом никогда не сможет воспользоваться человек, который не знает, что провод от утюга надо воткнуть в розетку. Но, обладая таким знанием, человек может пользоваться утюгом, не зная, сколько энергии утюг потребляет (отсутствие точности), равно как сохраняя искреннюю уверенность, что по проводам, как вода, течёт электричество (отсутствие правильности).

Беда приходит тогда, когда представления человека о системе концептуально не совпадают с реальным устройством системы.

Так, например, человек, привезенный на машине времени из прошлого и никогда не встречавшийся с электричеством, поставит утюг на плиту, отчего прибор непременно сгорит.

**16 Понятие метафоры. Правила использования метафор.**

Разработать пользовательский интерфейс, в котором модель программы соответствует модели пользователя – задача не из легких. Иногда у пользователей просто нет конкретного представления о том, как работает программа и для чего она предназначена. В таком случае вам придется найти способ подсказать им, как функционирует ваша программа. В графических интерфейсах используется метод метафор.

Он позволяет пользователю не создавать новую модель, а воспользоваться готовой моделью, которую он ранее построил по другому поводу.

Самая известная метафора, применяемая и в Windows и в Macintosh – это метафора "десктоп" (рабочий стол). Перед вами маленькие папочки с листочками-файлами внутри, последние можно перемещать из одной папки – в другую. Метафора работает, потому что изображения папок напоминают реальные папки, которые мы используем для хранения и сортировки документов в своих кабинетах.

Еще один пример метафоры в интерфейсе – устройство программ для проигрывания звуков на компьютере. Исторически сложилось, что вся аудиотехника имеет почти одинаковый набор кнопок: несколько кнопок со стрелками (назад/вперед), кнопка с треугольником (воспроизведение), кнопка с двумя дощечками (пауза), кнопка с квадратиком (полная остановка) и красный кружок (запись). Про них нельзя сказать, что они совершенно понятны, но научиться им можно без труда. При этом обычно жизнь складывается так, что сначала человек научается пользоваться этими кнопками на материальных устройствах, а уж потом начинает пользоваться компьютером. Соответственно, при проектировании программы аналогичного назначения разумно скопировать существующую систему маркировки кнопок. Благодаря этому пользователям для использования программы ничему не приходится учиться (и даже не приходится переучиваться, что вдвойне обидно, поскольку полностью отрицает возвращение инвестиций в обучение).

Анализируя опыт применения метафор, можно вывести следующие правила:

*- опасно полностью копировать метафору, достаточно взять из неё самое лучшее;*

*- не обязательно брать метафору из реального мира, её смело можно придумать самому;*

*- эффективнее всего метафорически объяснять значение отдельных объектов: например, для графической программы слои можно представлять как положенные друг на друга листы стекла (этот пример подходит и для предыдущего пункта);*

*- если метафора хоть как-то ограничивает систему, от неё необходимо немедленно отказаться.*

**17 Использование стандартов для обеспечения «понятности» системы. Как может быть достигнута популярность стандарта?**

***Стандарт***. Наконец, остался последний, самый мощный, но зато и самый ненадежный способ обучения, а именно стандарт.

Если что-либо нельзя сделать «самопроизвольно» понятным, всегда можно сделать это везде одинаково, чтобы пользователи обучались только один раз.

Например, кран с горячей водой всегда маркируют красным цветом, а кран с холодной – синим. Частично это соответствует свойствам человеческого восприятия (недаром красный цвет мы называем тёплым, а синий – холодным), но в основном здесь работает привычка.

Основные программы пакета Microsoft Office – Word и Excel – разрабатывались с нуля программистами компании. Другие же были куплены на стороне: FrontPage, например, у Vermeer, или Visio – у Visio. Что у этих программ общего? Дизайн обеих создавался с самого начала так, чтобы они выглядели и работали как приложения Microsoft Office.

Как я уже сказала, стандарт штука мощная, но зато ненадежная. Он очень хорошо работает, когда популярен, в противном случае не работает вовсе.

Популярность стандарта может быть достигнута двумя способами:

* во-первых, он может быть во всех системах,
* во-вторых, он может быть популярен внутри отдельной системы.

Например, стандарт интерфейса MS Windows популярен почти во всех программах для Windows, именно поэтому его нужно придерживаться.

С другой стороны, этот стандарт оставляет неопределенным очень многое (никто да не обнимет необъятного), и это многое в разных системах трактуется по-разному.

**18 Использование аффорданса для обеспечения «понятности» системы. Основные виды аффордансов.**

***Аффорданс****.*  В современном значении этого термина аффордансом называется ситуация, при котором объект показывает субъекту способ своего использования своими неотъемлемыми свойствами.

Например, у двери с ручкой аффорданс к тому, чтобы ее тянули. Но иногда, ручка бывает у двери, которая открывается наружу, соответственно которую надо толкать. Образуется конфликт между аффордансом ручки (тянуть) и функцией двери (открываться толканием). Чтобы нивелировать этот конфликт зачастую вешают табличку с указанием к действию. Более эффективное и простое решение — убрать ручку. Так образуется аффорданс к толканию.

Принцип аффорданса активно используется в компьютерных интерфейсах. Например, выпуклая кнопка сигнализирует нам о том, что ее можно нажать. Дизайн корзины и папок на Рабочем столе, ссылаясь на реальные объекты из жизни, дают понять как их правильно использовать.

Способы передачи аффорданса:

- ***повторение конфигурации объектов конфигурацией элементов управления*** *(этот способ работает хорошо в реальном мире, но не очень хорошо на экране, поскольку предпочтительней непосредственное манипулирование);*

*-* ***видимая принадлежность управляющих элементов объекту;***

*-* ***визуальное совпадение аффордансов экранных объектов с такими же аффордансами объектов реального мира*** *(кнопка в реальном мире предлагает пользователю нажать на неё, псевдотрехмерная кнопка предлагает нажать на неё по аналогии);*

*-* ***изменение свойств объекта при подведении к нему курсора*** *(бледный аналог тактильного исследования).*

**19 Как можно повысить субъективную скорость работы?**

*субъективную скорость работы* можно повысить двумя способами:

- *Заполнение пауз между событиями*. Есть данные о том, что если в периоды ожидания реакции системы пользователям показывается индикатор степени выполнения, субъективная продолжительность паузы существенно снижается. Судя по всему, чем больше информации предъявляется пользователям в паузах, тем меньше субъективное время. С другой стороны, эта информация может вызвать стресс в кратковременной памяти, так что пользоваться этим методом надо осторожно.

- *Разделение крупных действий пользователей на более мелкие*. При этом количество работы увеличивается, но зато субъективная длительность снижается. Плох этот метод тем, что увеличивает усталость.

С другой стороны, повышение объективной скорости работы зачастую способно повысить и субъективную скорость.

**20 Какие факторы влияют на субъективное удовлетворение?**

Все факторы связаны с субъективными ощущениями человека. А именно с субъективным ощущениям:

* эстетики;
* времени работы;
* психологического напряжения;
* собственной глупости;
* самовыражения.

**21 Отчего зависит субъективное ощущение психологического напряжения пользователя?**

Пользователь знает, что во время работы может что-либо испортить. Он может отформатировать жесткий диск, может стереть или испортить нужный файл. И это вызывает у него психологическое напряжение, иначе говоря – стресс.

Пользователь, знающий, **что он *не может* совершить ошибку**, испытывает радость и умиротворение. Чтобы добиться этого, необходимо иметь возможность:

* отмены пользователем своих предыдущих действий, без ограничения количества уровней отмены и типа отменяемых действий.

Задача эта непростая, но зато результат крайне существенен. К сожалению, создание таких систем требует смены модели мышления программистов.

Зачастую более реалистичным решением является давно уже существующая практика:

* прятать опасные для пользователя места интерфейса. Проблема заключается в том, что при этом логично прятать *все* функции, изменяющие данные, например банальная функция автоматической замены, может мгновенно уничтожить текст документа.

Другим фактором, существенно влияющим на субъективное удовлетворение пользователей, является **чувство контроля над системой**.

Существует значительная часть пользователей, для которой использование компьютера не является действием привычным. Для таких пользователей ощущение того, что они не способны контролировать работу компьютера, является сильнейшим источником стресса.

Таким образом, пользователей нужно всемерно снабжать ощущением, что ничего не может произойти, пока этого не захочется самому пользователю. Функции, работающие в автоматическом режиме, но время от времени просыпающиеся и требующие от пользователей реакции, вызывают стресс. В любом случае, стоит всеми силами внушать пользователям мысль, что только явно выраженное действие приводит к ответному действию системы (это, в частности, главный аргумент против ролловеров – пользователь ещё ничего не нажал, а уже что-то произошло).

**22 Каким должно быть сообщение об ошибке??**

Теперь можно рассказать, каким должно быть сообщение об ошибке, тем более, что ничего сложного в создании идеального сообщения нет. Напротив, всё очень просто. Идеальное сообщение об ошибке должно отвечать всего на три вопроса:

- В чем заключается проблема?

- Как исправить эту проблему сейчас?

- Как сделать так, чтобы проблема не повторилась?

Например, попытаемся улучшить уже опоминавшееся ранее сообщение о невозможности перезаписать заблокированный файл.

Итак, старое сообщение об ошибке гласило: «Не удается сохранить файл «D:\Только для чтения.doc». Файл с этим именем уже существует и доступен только для чтения. Сохраните файл под другим именем или в другой папке».

В каких случаях оно появляется?

Оно может появляться, если пользователь попытался сохранить файл на компакт-диске, или же пытается сохранить файл, незадолго перед этим скопировав этот файл с компакт-диска. Случаи, когда файл заблокирован сознательно, в жизни редки, так что их чаще всего можно не учитывать.

Тут возможно несколько не противоречащих друг другу решений:

* можно блокировать возможность что-либо записать на диске, запись на который невозможна. Собственно говоря, запись и так блокируется, но сообщением об ошибке.
* можно просто не показывать диски, на которые нельзя записывать, в окне записи, что эффективнее, поскольку делает ошибку невозможной.
* можно показывать файлы, защищенные от записи, иначе, чем файлы незащищенные. Это будет работать, но тоже неидеально.

Что делать пользователю, который всё-таки хочет перезаписать файл? Сейчас в такой ситуации приходится записывать файл под новым именем, потом стирать старый, а новому давать имя старого. Это и потери времени и ошибочно стертые файлы (лучший способ сделать так, чтобы пользователи не стирали нужные файлы, заключается в том, чтобы лишить пользователей необходимости вообще что-либо стирать в нормальном режиме работы).

Таким образом, сообщение об ошибке должно стать не только сообщением – оно должно позволять разблокировать файлы, разблокировать которые возможно (т.е. записанные не на компакт-диске). Таким образом, получается, что нужно сделать несколько изменений в интерфейсе:

- Диски, на которые ничего нельзя записать, не показываются в диалоговом окне сохранения файлов.

- Заблокированные файлы на остальных дисках показываются иначе, нежели файлы незаблокированные.

- При попытке записать документ поверх заблокированного, появляется сообщение об ошибке примерно такого вида:

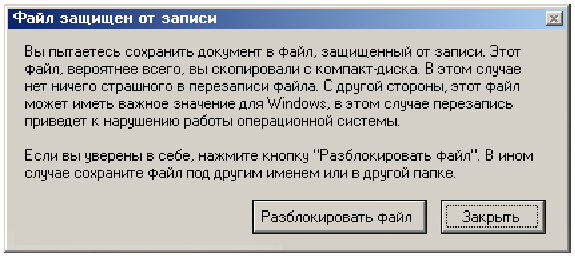


Рис. 15. Улучшенное сообщение об ошибке.

Обратите внимание, что кнопка ***Закрыть*** выбрана по умолчанию, чтобы снизить вероятность перезаписи важных файлов.

Конечно, лучше всего было бы, чтобы ОС сама снимала с копируемых с компакт-диска файлов метку Read Only. Многие проблемы при этом бы исчезли, поскольку защищенными от записи остались только действительно важные для ОС файлы.

**23 Закон Хика и закон Фиттса. Особенности комплексного применении законов Хика и Фиттса в UX**

Еще в 1954 году Поль Фитс (Paul Fitts) сформулировал правило, ставшее известным как Закон Фитса:

*Время достижения цели прямо пропорционально дистанции до цели и обратно пропорционально размеру цели*

***Тдост цели = а +b log2 (D / S + 1) мс,***

Где ***а*** и ***b*** -устанавливаются опытным путем по параметрам производительности человека. Для практического использования можно принять: ***а*** = 50, ***b*** = 150, ***D*** – дистанция от курсора до цели, **S** – размер цели по направлению движения курсора.

Популярно говоря, **лучший способ повысить доступность кнопки заключается в том, чтобы делать её большой и располагать ближе к курсору.**

У этого правила есть два не сразу заметных следствия. Чтобы «бесконечно» ускорить нажатие кнопки, её, во-первых, можно сделать бесконечного размера и, во-вторых, дистанцию до неё можно сделать нулевой.

**Кнопка бесконечного размера.** При подведении курсора к краю экрана он останавливается, даже если движение мыши продолжается.

**Нулевая дистанция до кнопки.** Рассмотрим контекстное меню, вызываемое по нажатию правой кнопки мыши. Оно всегда открывается под курсором, соответственно расстояние до любого его элемента всегда минимально. Именно поэтому контекстное меню является чуть ли не самым быстрым и эффективным элементом.

**Закон Хика** — закон психологии который применим в вебе и проектировании интерфейсов.

Закон говорит о том, что чем меньше элементов меню, тем меньше времени занимает выбор одного из них. А также что одно меню лучше чем два.

**24 Объяснить принцип работы модели количественного анализа интерфейсов GOMS.**

Одним из лучших подходов к количественному анализу моделей интерфейсов является классическая модель *GOMS (the model of goals, objects, methods and selection rules)* - модель целей, объектов, методов и выбор правил.

Эта модель основана на оценке скорости печати.

Время, требуемое для выполнения какой-то задачи системой пользователь-компьютер, является суммой всех временных интервалов, которые потребовались системе на выполнение элементарных жестов, составляющих данную задачу.

Лабораторным путем установлены стандартные средние интервалы для некоторых жестов, выполняемых различными пользователями:

К = 0,2 с – нажатие клавиши;

Р = 1,1 с – указание (на какую-то позицию на экране монитора);

Н = 0,4 с – перемещение (руки с клавиатуры на «мышь» или обратно;

М = 1,35 с – ментальная подготовка – мысленный выбор пользователем своего следующего элементарного действия;

R – ответ (время ожидания ответа компьютера).

Расчет по модели *GOMS.*

Основные правила, позволяющие определить, в какие моменты будут проходить ментальные операции, представлены в табл. 1.

Таблица 1

|  |  |
| --- | --- |
| **Правило 0**  Начальная расстановка операторов **М** | Оператор М устанавливается перед всеми операторами К (нажатие клавиши) и Р, предназначенными для выбора команд, если Р указывает на аргументы этих команд оператор М не ставится |
| **Правило 1**  Удаление ожидаемых операторов **М** | Если оператор, следующий за оператором М ожидаемый с точки зрения оператора, предшествующего М, то этот оператор М может быть удален и последовательность РМК превращается в РК |
| **Правило 2**  Удаление операторов **М** внутри когнитивных единиц | Если строка типа МКМКМК… принадлежит когнитивной единице, то следует удалить все операторы М, кроме первого. Когнитивной единицей является непрерывная последовательность вводимых символов, которые могут образовывать название команды или аргумент. |
| **Правило 3**  Удаление **М** перед последовательными разделителями | Если оператор К означает лишний разделитель, стоящий в конце когнитивной единицы (например разделитель команды, следующий сразу за разделителем аргумента этой команды) то следует удалить оператор М, стоящий перед ним. |
| **Правило 4**  Удаление операторов **М**, которые являются прерывателями команд | Если оператор К является разделителем, стоящим после постоянной строки ( например, название команды или любая последовательность символов, которая каждый раз вводится в неизменном виде), то следует удалить оператор М, стоящий перед ним. Но если оператор К является разделителем для строки аргументов или любой другой изменяемой строки, то М следует сохранить перед ним. |
| **Правило 5**  Удаление перекрывающихся операторов **М** | Любую часть операторов М, которая перекрывает оператор R, означающий задержку, связанную с ожиданием ответа компьютера, учитывать не следует. |

**25 Закономерности зрительного восприятия человеком информации.**

**Гештальт** (от нем. форма, силуэт) – это совокупность принципов визуального восприятия, разработанная немецкими психологами в 1920-е годы. Эти принципы построены на теории о том, что «целое воспринимается больше суммы составляющих его частей».

«Целое – это некая реальность, отличная от суммы своих частей» – Курт Коффка (один из основателей гештальт-психологии).

* ***Близость*.** Это основа строгой группировки содержимого и элементов управления в пользовательском интерфейсе. Так как, размещенные рядом элементы пользователь ассоциирует друг с другом.
* ***Сходство*.** Пользователь ассоциирует элементы друг с другом, если у них совпадает форма, размер, цвет или направление.
* ***Непрерывность*.** Пользователь хочет видеть непрерывные линии и кривые, даже если они сформированы путем выравнивания небольших разнообразных элементов.
* ***Замкнутость*.** Группа элементов воспринимается как простая замкнутая форма или фигура, даже если она не нарисована специально.
* ***Фигура/Фон*.**

**27 Восприятие и представление информации. Их сходство и различие. Какие факторы влияют на успешное восприятие визуальной информации.**

**Восприятие** – это актуальные переживания, то есть те, которые взаимодействует на человека именно в этот момент. Следовательно, восприятие происходит до тех пор, пока какое-либо воздействие продолжается.

**Представление** – воспроизводит то, что действовало на человека когда-то, то есть было отражено в виде восприятия.

*Пример: апельсин. Восприятие: жёлтый овальный., представление: кислый, сочный, ароматный.*

Человек имеет 5 видов чувств: зрение, слух, вкус, обоняние (нюх), осязание (ощупь).

Процесс приёма и усвоения информации получаемой при помощь воздействия предметов и явлений на органы чувства человека называется **восприятие** или **перцепция** (от лат. perceptio).

|  |  |
| --- | --- |
| **Способ восприятия информации** | **Вид информации** |
| С помощью глаз | Зрительная |
| С помощью ушей | Звуковая |
| С помощью языка | Вкусовая |
| С помощью кожи | Тактильная |
| С помощью носа | Обонятельная |

90% информации об окружающем мире человеку даёт зрение, 8% приходится на органы слуха, все остальное – обоняние, осязание, вкус.

На успешное восприятие визуальной информации влияют такие факторы как: размер объекта, цвет объекта, интенсивность объекта, контрастность объекта.

**28 Основные свойства внимания.**

***Объем*** внимания измеряется тем количеством объек­тов, которые воспринимаются одновременно.

Обычно объем внимания зависит от специфически прак­тической деятельности человека, от его жизненного опы­та, от поставленной цели, от особенностей воспринимае­мых объектов. Объединенные по смыслу объекты воспри­нимаются в большем количестве, чем не объединенные. У взрослого человека объем внимания равен 4—6 объектам.

***Концентрация внимания*** есть степень сосредоточения сознания на объекте (объектах).

Чем меньше круг объектов внимания, чем меньше учас­ток воспринимаемой формы, тем концентрированнее вни­мание.

**Устойчивость внимания**не означает сосредоточеннос­ти сознания в течение всего времени на конкретном пред­мете или его отдельной части, стороне. Под устойчивостью понимается общая направленность внимания в процессе деятельности. На устойчивость внимания значительное вли­яние оказывает интерес.

**Отвлекаемость внимания** выражается в колебаниях вни­мания, которые представляют собой периодическое ослаб­ление внимания к конкретному объекту или деятельности.

**Колебания внимания**наблюдаются даже при очень со­средоточенной и напряженной работе, что объясняется непрерывной сменой возбуждения и торможения в коре головного мозга.

**Переключение внимания** состоит в перестройке вни­мания, в переносе его с одного объекта на другой.

Различают переключение внимания преднамеренное (произвольное) и непреднамеренное (непроизвольное). Преднамеренное переключение внимания происходит при изменении характера деятельности, при постановке новых задач в условиях применения новых способов действий. Преднамеренное переключение внимания сопровождается участием волевых усилий человека.

**29 Внимание. Фокус и локус внимания, в чем сходство и в чем различие.**

Внимание это то, что позволяет нам обрабатывать информацию об окружающем нас мире. Мы осознаём вещи только тогда, когда внимаем им, обращаем на них своё внимание.

**Фокус внимания** человека применительно к компьютерным системам – некоторое место на экране, куда направлен его взгляд и где он сознательно сосредоточен.

**Локус внимания** – это некоторое место или область, на которое может быть сосредоточено ваше внимание. В отличие от фокуса, часто обозначающего не только место, но и действие (сфокусировать ваше внимание), локус обозначает только место и переводится с латинского, как место положения или область.

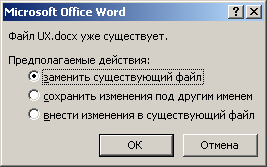
Локус и фокус внимания может быть только один.

**30 Типы внимания: избирательное, фокусированное, распределенное. Когда каждый из них используется при работе с интерфейсом?**

Существует три вида внимания:

1. **Избирательное внимание**. Этот вид внимания иногда называют туннельным вниманием. Оно возникает тогда, когда мы обращаем внимание на стимул или задачу так страстно, что начинаем полностью игнорировать все остальные стимулы и объекты. Программист, занятый написанием кода программы, геймер, бороздящий просторы виртуального пространства, или водитель, полностью сконцентрированный на дороге, все они могут запросто пропустить мимо ушей вопрос, заданный им другим человеком.

При поиске в «зашумленном» интерфейсе некоторой важной информации мы также используем свое избирательное внимание.

**2. Фокусированное внимание**. Это внимание можно назвать более эффективным избирательным вниманием, потому что в данном случае мы целенаправленно перестаем обращать наше внимание на стимулы для того, чтобы завершить задачу. Например, пользователь программы, который сознательно игнорирует уведомление о новом сообщение, мерцающем в углу экрана, для того, чтобы закончить и послать e-mail. В поле фокусированного внимания находится письмо, а остальные стимулы человек намеренно исключил из своего поля зрения.

Например, если пользователь пытается сохранить в MS Word документ с именем уже существующего файла, то выводится модальное предупреждение, которое привлекает внимание пользователя и не позволяет ему отвлекаться на что-либо еще.

**3. Распределенное внимание**. Бывают такие ситуации, когда становится невозможным фокусировать свое внимание на одной задаче из-за того, что другие стимулы начинают отвлекать нас. Например, если мы вдруг услышим, как кто-то разговаривает о нас в то время, пока бы ведем беседу с другими людьми, нам станет трудно удерживать свое внимание исключительно на нашем собственном разговоре. Некоторые компьютерные задачи вынуждают пользователей фокусироваться на нескольких вещах. Это значительно понижает эффективность и продуктивность работы пользователя.

**31 С помощью чего можно привлечь внимание пользователя?**

Заострить внимание на текущем объекте в системе можно с помощью:

• Положения

• Размеров

• Цвета

• Контрастности

• Подсветки

• Указания

• Выделения

• Активации

Платная регистрация на сайте знакомств – пример функций, которая привлекает внимание посетителя для решения бизнес-целей.

**32 Сколько элементов может запомнить КВП и как можно увеличить их число? Сколько времени хранится в кратковременной памяти информация, ставшая локусом внимания? И когда она из нее исчезает?**

Считается, что человеческая память способна запомнить 7±2 элементов. Однако:

* Во-первых, в КВП информация хранится преимущественно в звуковой форме. Поэтому подвергать ограничению следует преимущественно те элементы, которые содержат текст.
* Во-вторых, в память помещается гораздо больше элементов, если они сгруппированы.
* В-третьих, существуют люди неспособные удержать девять значений, а только пять или шесть.

Поэтому, эффективнее считать, что объём КВП равен пяти (шести ил которых один в запасе) элементам.

И запоминание, и извлечение информации из памяти требует усилий. Поэтому необходимо снижать нагрузку на память пользователей, т.е. избегать ситуаций, когда пользователю приходится получать информацию в одном месте, а использовать её в другом.

**33 Кратковременная (КВП) и долговременная память (ДВП). Как информация попадает в КВП и ДВП?**

Чтобы что-либо попало в КВП пользователя:

* Пользователь должен заметить (самое важное в интерфейсе должно быть наиболее заметным) и счесть полезным лично для себя;
* Смена содержимого происходит при появлении новых стимулов (нельзя допускать, чтобы пользователь отвлекался, поскольку новые стимулы при отвлечении стирают содержимое КВП).

Считается, что информация попадает в ДВП в трёх случаях:

* При повторении (если системой придётся пользоваться часто, пользователи ей обучаются).
* При глубокой семантической обработке (**семантика** – свойство, определяющее смысл информации как соответствие сигнала реальному миру). Если пользователь долго мучается, стараясь понять, как работает система, он запомнит её надолго.
* При наличии сильного эмоционального шока.

**34 Какие свойства интерфейса формируют привычки? Что дают пользователю привычки при работе с интерфейсом?**

Чтобы интерфейс формировал привычки, он должен быть:

* простым
* часто использоваться

Привычки высвобождают внимание.

Человек с высвобожденным вниманием устаёт меньше и меньше напрягается.

**35 В чем суть гештальт-психологии, когда она возникла и кто был ее основоположником**

**Гештальт** (от нем. форма, силуэт) – это совокупность принципов визуального восприятия, разработанная немецкими психологами в 1920-е годы. Эти принципы построены на теории о том, что «целое воспринимается больше суммы составляющих его частей».

«Целое – это некая реальность, отличная от суммы своих частей» – Курт Коффка (один из основателей гештальт-психологии).

Приведенная цитата кратко описывает суть гештальта. Люди видят группу объектов, как нечто целое, прежде чем воспринимают эту же группу, как состоящую из элементов. Мы видим целое, как что-то большее, чем сумма частей, и даже тогда, когда детали являются отдельными субъектами, мы визуально группируем их в целое.

**36 Гештальт-принципы организации восприятия. Факторы, способствующие группировке элементов в целостные гештальты.**

**Генштальт** (нем. Gestalt – форма, образ, структура) – пространственно-наглядная форма воспринимаемых предметов, чьи существенные свойства нельзя понять путём суммирования свойств их частей.

Согласно генштальт-психологии, свойства целого отличны от свойств его элементов. На основе этого мнения генштальт-психологии разработали несколько принципов, объясняющих особенности организации восприятия.

* **Близость.**

Элементы, которые близки друг к другу в пространстве или во времени кажутся нам объединёнными в группы, и мы стремимся воспринимать их совместно.

* **Непрерывность.**

В нашем восприятии существует тенденция следования в направлении, позволяющем связывать наблюдаемые элементы в непрерывную последовательность или придать им определённую ориентацию.

* **Сходство.**

Подобные элементы воспринимаются нами совместно, образуя замкнутые группы.

* **Замкнутость.**

В нашем восприятии существует тенденция завершения незаконченных предметов и заполнения пустых промежутков.

* **Простота.**

В любых условиях мы стремимся видеть фигуры настолько завершёнными, насколько это возможно; в генштальт-психологии это свойство поучило название «прегнантная форма».

Прегнантный генштальт должен быть симметричным, простым и неизменным и не может быть упрощён или упорядочен каким-либо иным образом.

На изображении олимпийских колец мы отчётливо видим именно кольца, а не совокупность фигур более сложной формы. Согласно этому закону, реальность организована наипростейшим образом из всех доступны.

* **Фигура-фон**

Мы стремимся организовывать наше восприятие таким образом, чтобы видеть объект (фигуру) и задний план (фон), на котором она проявляется. При этом фигура представляется наиболее заметной и явнее выделяется на общем фоне изображения.

**37 В чем суть гештальт-принципов близости, непрерывности, замкнутости и сходства и применение их в UX-дизайне.**

36 ответ

**38 Количественная оценка интерфейса с помощью метода GOMS. Правила расстановки ментального оператора.**

Модель GOMS (the model of goals, objects, methods and selection rules) – модель целей, объектов, методов и наборов правил.

Время, требуемое для выполнения какой-то задачи системой пользователь-компьютер, является суммой всех временных интервалов, которые потребовались системе на выполнение элементарных жестов, составляющих данную задачу.

Таблица 2

|  |  |
| --- | --- |
| **Правило 0**  Начальная расстановка операторов **Д** | Операторы **Д** следует устанавливать перед всеми операторами **К** и **М** (нажатие клавиши), также перед всеми операторами **P**, предназначенными для выбора команд. Но перед операторами **P**, предназначенными для указания на аргументы этих команд, ставить оператор **Д** не следует. |
| **Правило 1**  Удаление ожидаемых операторов **Д** | Если оператор, следующий за оператором **Д**, является полностью ожидаемым с точки зрения оператора, предшествующего **Д**, то этот оператор **Д** может быть удален.  Если пользователь перемещает мышь с намерением нажать на ее кнопку по достижении цели движения, то в соответствии с этим правилом следует удалить оператор **Д**, установленный по **правилу** **0**. Так последовательность действий **P Д К** преобразуется в **P К**. |
| **Правило 2**  Удаление операторов **Д** внутри когнитивных единиц | Если строка **Д К Д К Д К…** принадлежит когнитивной единице, то следует удалить все операторы **Д**, кроме первого. Когнитивной единицей является непрерывная последовательность вводимых символов, которые образуют название команды или аргумент. Например *Y*, *перемещать*, *4564.23 –* это когнитивные единицы. |
| **Правило 3**  Удаление операторов **Д** перед последовательными разделителями | Если оператор **К** означает лишний разделитель, стоящий в конце когнитивной единицы (например, разделитель команды, следующий сразу за разделителем аргумента этой команды), то следует удалить оператор **Д** , стоящий перед ним. |
| **Правило 4**  Удаление операторов **Д**, которые являются прерывателями команд | Если оператор **К** является разделителем, стоящим после постоянной строки (например, название команды или любая последовательность символов, которая каждый раз вводится в неизменном виде), то следует удалить оператор **Д**, стоящий перед ним. (Добавление разделителя станет привычным действием, и поэтому разделитель станет частью строки и не будет требовать специального оператора **Д**.) Но если оператор **К** является разделителем строки аргументов или любой другой изменяемой строки, то оператор **Д** следует сохранить перед ним. |
| **Правило 5**  Удаление перекрывающих операторов **Д** | Любую часть оператора **Д**, которая перекрывает оператор **R**, означающий задержку, связанную с ожиданием ответа компьютера, учитывать не следует. |

В этих правилах под ***строкой*** понимается некоторая последовательность символов.

***Разделителем*** считается символ, которым обозначено начало или конец значимого фрагмента текста, такого как, например, слово естественного языка или телефонный номер. Так, пробел является разделителем для большинства слов, а точка используется в конце предложений для разделения. В качестве разделителей могут выступать скобки для ограничений пояснений или замечаний и т. д.

Если для выполнения команды требуется дополнительная информация, она называется здесь ***аргументом*** данной команды.

Правила GOMS позволяют определить время, необходимое пользователю для выполнения любой, четко сформулированной задачи, для которой данный интерфейс предусмотрен. Однако этого метода недостаточно, чтобы оценить насколько быстро должен работать интерфейс – его производительность. Рассмотрим подробнее **критерий эффективности** **Раскина**, который оценивает информационную производительность интерфейса.