

Человеко-машинное взаимодействие

Лекция 3

Мерзлякова Екатерина Юрьевна

к.т.н. доцент ПМиК

Проблемно-центрированная разработка интерфейса

- анализ задач и пользователей;
- выбор репрезентативных задач;
- заимствование;
- черновое описание дизайна;
- обдумывание дизайна;
- создание макета или прототипа;
- тестирование дизайна с пользователями;
- итерирование;
- реализация;
- отслеживание эксплуатации;
- изменение дизайна.

Анализ задач и пользователей



Кто и зачем собирается использовать разрабатываемую систему?

- уровень знаний пользователя
- область деятельности
- общие характеристики

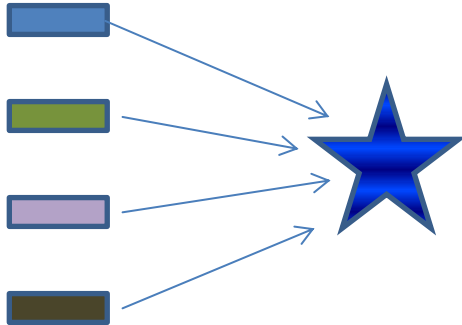
- Система должна запрашивать от пользователя информацию в порядке, который кажется ему естественным,
- Система должна давать возможность простой коррекции ошибок при вводе данных,
- Выбранные для организации интерфейса аппаратные средства должны вписываться в рабочую среду пользователя и быть эргономичными для его действий.

Выбор репрезентативных задач



- задачи, которые пользователи описали разработчикам
- реальные задачи, с которыми сталкиваются пользователи
- задачи должны достаточно полно покрывать всю функциональность системы
- смесь простых и более сложных задач

Заимствование



найти существующие интерфейсы, с помощью которых пользователи могут выполнить требуемую работу, и затем строить идеи новой системы на их базе

- Какие программы ваши пользователи используют сейчас?
- Найти существующие интерфейсы, с помощью которых пользователи могут выполнить требуемую работу, и затем строить идеи новой системы на их базе
- Чаще всего наилучшим вариантом будет придерживаться правилам старой системы

Черновое описание дизайна



Черновое (грубое) описание разрабатываемой вами системы должно быть положено на бумагу (обязательно). Это позволяет задуматься о многих вещах. Но это описание не следует оформлять в виде компьютерной программы (пока), даже если вы умеете пользоваться какими-либо системами автоматизации разработки. Такие системы вынуждают вас прикрепляться к конкретным решениям, которые ещё слишком рано делать.

Обдумывание дизайна



- ✓ Стоимость построения законченного пользовательского интерфейса и его тестирование с достаточным количеством пользователей для выявления всех проблем очень высока.
- ✓ Существует несколько структурных подходов, которые можно использовать, чтобы исследовать сильные и слабые стороны интерфейса до его программного воплощения.

CWT - позволяет находить места в дизайне, где пользователь может делать ошибки.

GOMS - оценка трудоёмкости выполнения задач по времени и выявление задач, требующих слишком много шагов.

Создание макета или прототипа



Тестирование дизайна с пользователями

«думать вслух»



Итерирование



Тестирование
проанализировать
результаты тестирования,
соизмеряя **стоимость**
корректировок с
серьёзностью возникших
проблем, затем доработать
интерфейс и протестировать
его снова.

цель тестирования состоит не
в том, чтобы доказать
правильность интерфейса, а в
том, чтобы улучшить его

Построение интерфейса системы



пользовательский интерфейс часто занимает более половины кода коммерческого продукта

Отслеживание дизайна системы



**Программный
продукт**

команда разработчиков не должна быть
изолирована от всей остальной
деятельности, связанной с
функционированием системы

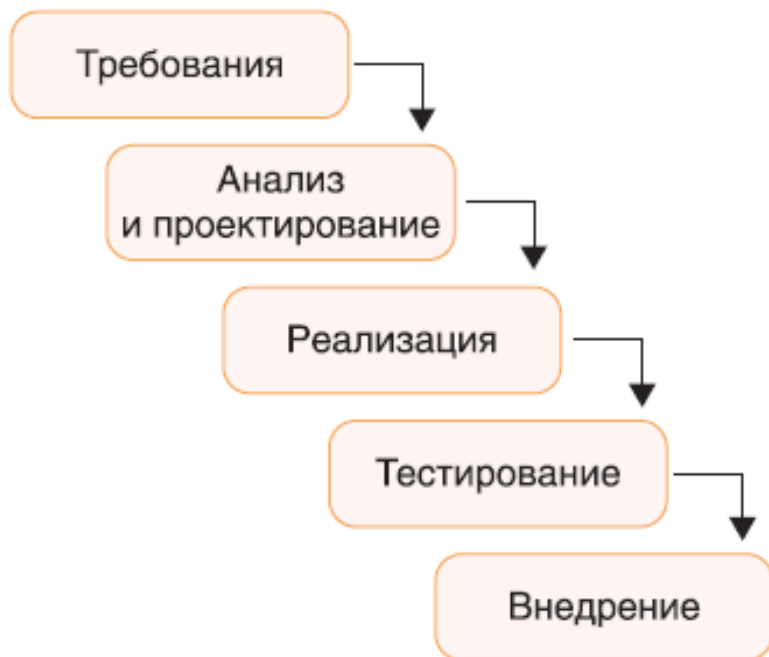
Изменение дизайна



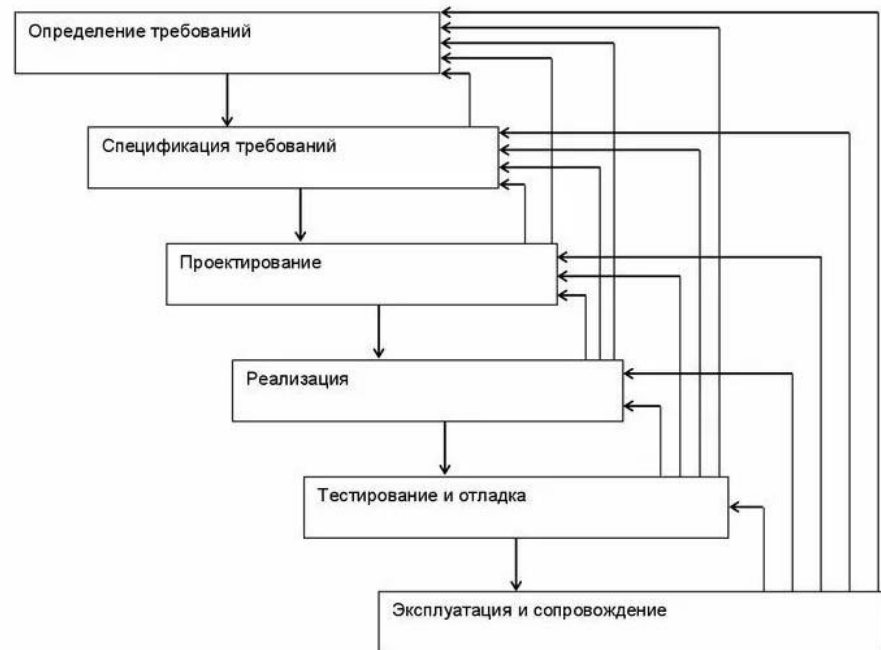
Независимо от того, насколько удачно система была спроектирована первоначально, с большой вероятностью она будет **терять адекватность** с течением лет. Меняются и задачи и пользователи. **Приёмы** работы меняются из-за нового оборудования и программных продуктов. Пользователи приобретают **новые навыки** и ожидаемые реакции. Разработчики должны стоять вровень с этими изменениями, не только отслеживая состояние той **рабочей среды**, для которой была предназначена их система, но и **развитие** всего общества, технологий и методов, потребностей.

Управление процессом разработки

Метод водопада



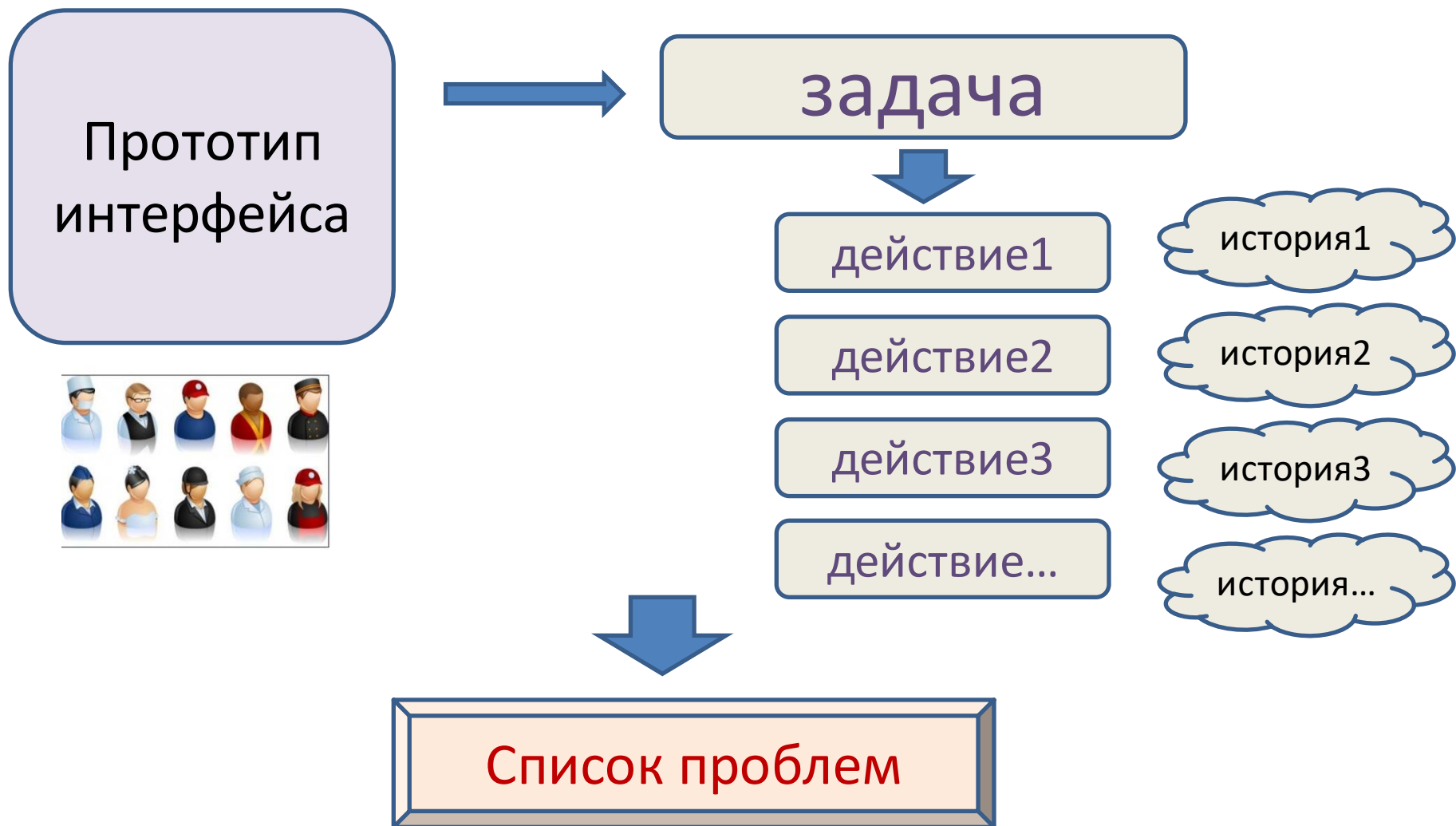
Проблемно-центрированный подход



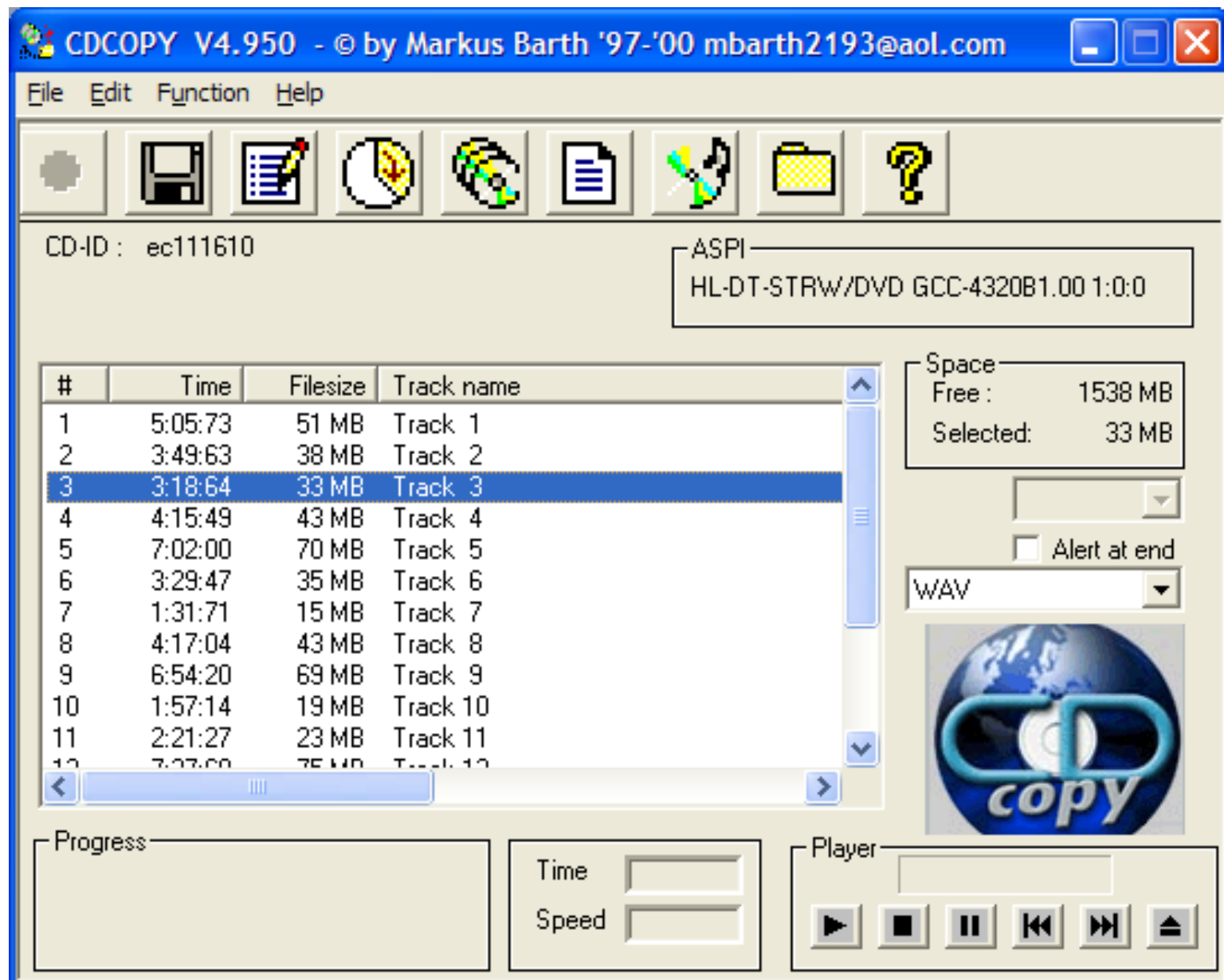
Требования практичности

- ✓ Требования практичности – это целевые значения для таких характеристик, как скорость выполнения репрезентативных задач и допустимое количество ошибок.
- ✓ Эти показатели могут использоваться, чтобы мотивировать разработчиков и обосновывать решения по распределению ресурсов.
- ✓ Целевые значения могут быть выбраны так, чтобы побить конкурентов или обеспечить функциональные нужды для хорошо определённых задач.

CWT-анализ интерфейса

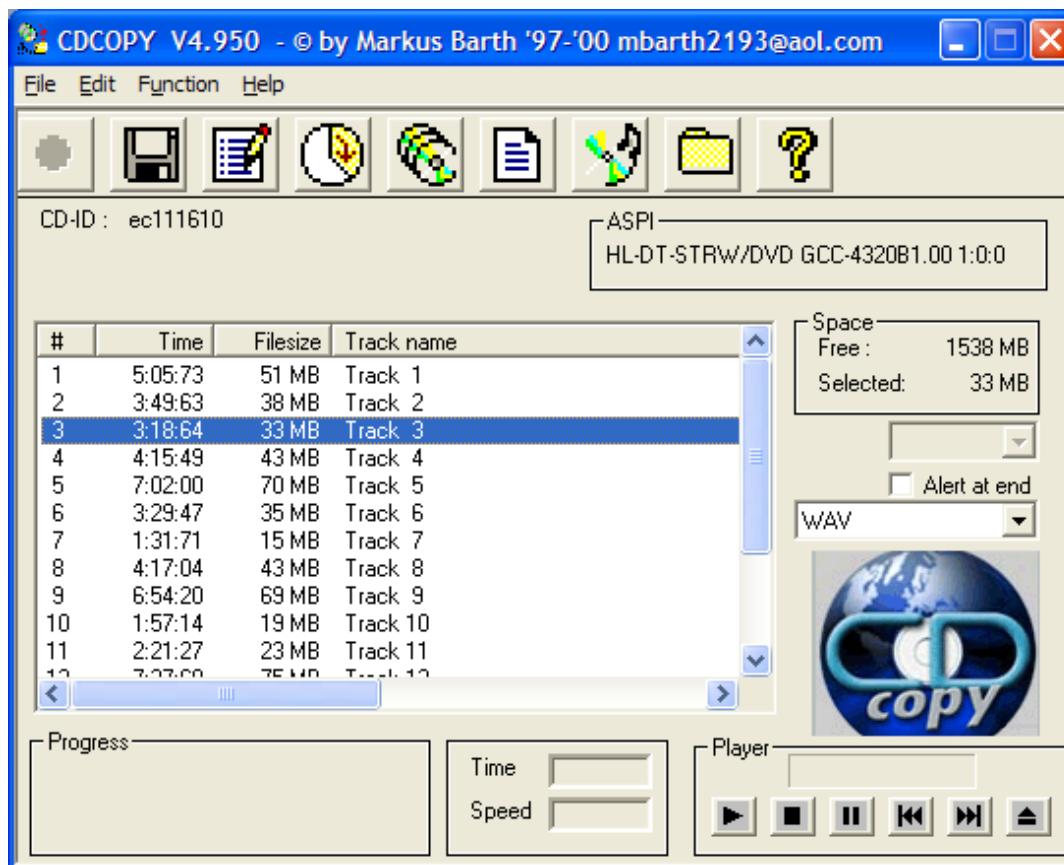


CWT-анализ интерфейса



CWT-анализ интерфейса

1. Загрузить аудио диск в устройство чтения компакт-дисков компьютера;
2. Запустить программу CDCopy;
3. В появившемся списке треков выбрать трек № 3;
4. В списке форматов файлов выбрать MP3(MPEG 1 Lay. 3);
5. Нажать кнопку "Start copying".



Что дает CWT-анализ интерфейса

- Он может поставить под сомнение ваши первоначальные и не вполне обоснованные предположения о том, как мыслит пользователь (вначале поставить диск, а потом запустить программу, или наоборот? кнопка с изображением дискеты означает копирование или сохранение?).
- Он может выявлять элементы управления, которые очевидны для разработчика, но могут быть скрыты от пользователя (список форматов выходного файла).
- Он может выявлять затруднения с надписями и подсказками (неудачное предупреждение "No media present").
- Он может обнаруживать неадекватную обратную связь, что может заставить пользователя сомневаться в результате и повторять всё с начала, хотя всё было сделано правильно (отсутствие индикации пути, по которому сохраняется файл).
- Он может показывать недостатки в текущем описании интерфейса (слова "Start copying" вместо графического изображения кнопки в руководстве пользователя).



Рекомендации при выполнении CWT-анализа

- ✓ Будут ли пользователи пытаться произвести тот или иной эффект, который даёт действие?
- ✓ Видят ли пользователи элемент управления (кнопку, меню, переключатель и т.д.) для осуществления действия?
- ✓ Если пользователи нашли элемент управления, поймут ли они, что он производит тот эффект, который им нужен?
- ✓ После того как действие сделано, будет ли понятен пользователям тот отклик, который они получают, чтобы перейти к следующему действию с уверенностью?

Анализ GOMS

Операции в GOMS – это элементарные действия, которые нельзя разложить на более мелкие.

Шаг метода: нажатие на кнопку

- визуально определить местонахождение кнопки (мыслительная операция);
- навести на кнопку указатель мыши (внешняя операция);
- щелкнуть кнопкой мыши (внешняя операция).

Анализ GOMS

K – нажатие клавиши;

B – клик кнопкой мыши;

P – наведение указателя мыши;

R – ожидание ответной реакции компьютера;

H – перенос руки с клавиатуры на мышь или наоборот;

D – проведение с помощью мыши прямой линии (например, выделение или прокрутка текста);

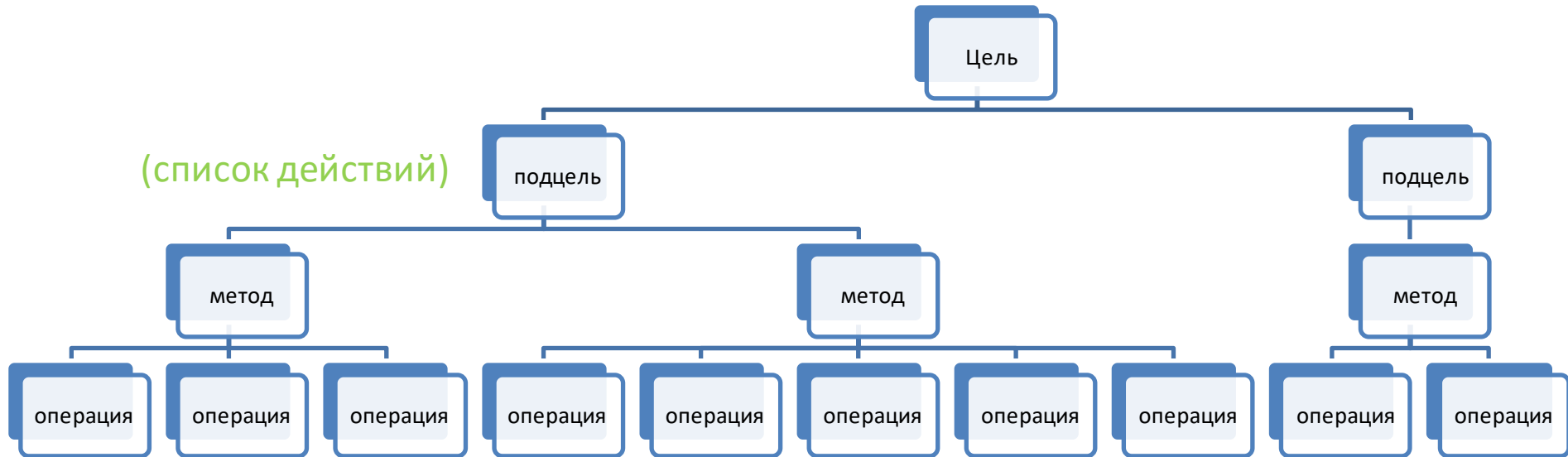
M – мыслительная подготовка (к осуществлению одной из перечисленных операций).

Анализ GOMS

К	0.2 с
В	0.2 с
Р	1.1 с
Р	0.25 с (или больше)
Н	0.4 с
Д	2 с (или больше)
М	1.35 с

Анализ GOMS

(репрезентативная задача)



(элементарные действия пользователя)

Анализ GOMS

Пример: Возьмём в качестве исследуемой программы Microsoft Word 2003. Пусть поставлена **цель**: напечатать (вставить в текст) уравнение $x^2 + x - 1 = 0$ с помощью редактора формул (перед печатью уравнения набирался текст).

Анализ GOMS

Для выполнения цели сформулируем три подцели:

1. Вызвать редактор формул через меню "Вставка | Объект... | Microsoft Equation 3.0".
2. Напечатать формулу в редакторе.
3. Выйти из редактора и подготовиться к продолжению набора текста.

Анализ GOMS

Теперь опишем методы для каждой подцели:

1. Вызвать редактор формул через меню "Вставка | Объект... | Microsoft Equation 3.0".

- 1.1. Войти в меню "Вставка"

- 1.2. Выбрать пункт "Объект..."

- 1.3. Выбрать объект "Microsoft Equation 3.0" путём скроллинга списка типов объектов.

Анализ GOMS

2. Напечатать формулу в редакторе.

2.1. Напечатать символ x .

2.2. В окне "Формула" выбрать объект 

2.3. В выпадающем меню объектов выбрать 

2.4. Напечатать символ 2.

2.5. Нажать клавишу \rightarrow .

2.6. Напечатать " $+ x - 1 = 0$ ".

Анализ GOMS

3. Выйти из редактора и подготовиться к продолжению набора текста.

3.1. Нажать клавишу Esc.

3.2. Нажать клавишу End (иногда при выходе из редактора формула остаётся выделенной, поэтому необходимо снять выделение).

Анализ GOMS

Теперь распишем каждый метод с точностью до операции (повторим для наглядности название подцелей и методов):

1. Вызвать редактор формул через меню "Вставка | Объект... | Microsoft Equation 3.0".

1.1. Войти в меню "Вставка"

Н (переместить руку на мышь, т.к. мы набирали текст)

Р (переместить указатель мыши)

В (клик мыши)

Анализ GOMS

1.2. Выбрать пункт "Объект..."

PB

1.3. Выбрать объект "Microsoft Equation 3.0" путём скроллинга списка типов объектов.

PB (перемещение указателя и фиксация мыши на элементе управления скроллинга)

D(3.0) (скроллинг вниз и поиск нужной строки, экспериментальная оценка времени)

PBV (установка указателя и двойной щелчок)

R(0.8) (ожидание запуска редактора формул)

Анализ GOMS

2. Напечатать формулу в редакторе.

2.1. Напечатать символ x .

H (перемещение руки на клавиатуру)

K (печать x)

2.2. В окне "Формула" выбрать объект 

H (перемещение руки на мышшь)

PV

Анализ GOMS

2.3. В выпадающем меню объектов выбрать



PB

2.4. Напечатать символ 2.

H

K

2.5. Нажать клавишу →.

K

2.6. Напечатать "+ x – 1 = 0".

7K (нажать Shift, печатать + (отпустить Shift),
печатать оставшиеся символы)

Анализ GOMS

3. Выйти из редактора и подготовиться к продолжению набора текста.

3.1. Нажать клавишу Esc.

K

R(0.8) (ожидание выхода из редактора формул и переход в текстовый режим)

3.2. Нажать клавишу End.

K

Анализ GOMS

HPBPBPD(3.0)PBPR(0.8)HKHPBPBHK7KKR(0.8)K

MHPBPBPD(3.0)PBPR(0.8) MHK MHPBPB MHK MK M7K MKR(0.8) MK

$8M = 10.8, 4H = 1.6, 6P = 6.6, 7B = 1.4, D(3.0) = 3.0, 2R(0.8) = 1.6, 12K = 2.4.$

Общий итог: 27.4 с.

Анализ GOMS

Пример: Возьмём в качестве исследуемой программы текстовый редактор. Пусть поставлена **цель:** напечатать (вставить в текст) уравнение $x^2 + x - 1 = 0$ с помощью редактора формул (перед печатью уравнения набирался текст).

8М, 10Н, 3Р, 6В, 20К.

Оценка среднего времени решения задачи составляет 23.3 с.

Анализ GOMS

МНРВРВРВД(3.0)РВВР(0.8) 10.95 с.

МНРВР(0.8) 3.85 с.

Анализ GOMS

Время нажатия на клавишу:

K = 0.2 с (естественной последовательности)

K = 0.5 с (случайной последовательности)

K = 0.75 с (сложные коды)

извлечения простого элемента знания из долговременной памяти

1.2 с

извлечения простого элемента знания из краткосрочной памяти

0.6 с

Золотые правила построения интерфейсов

Правила Нильсена-Молиха (Nielsen, Molich)

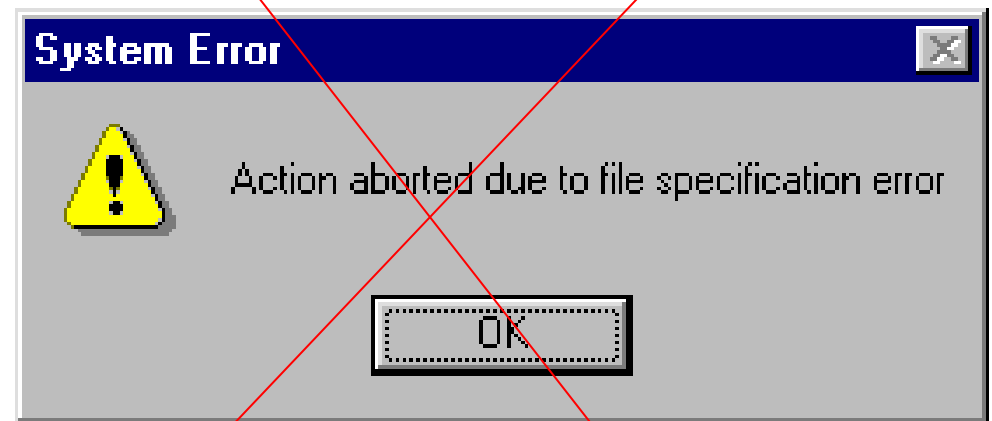
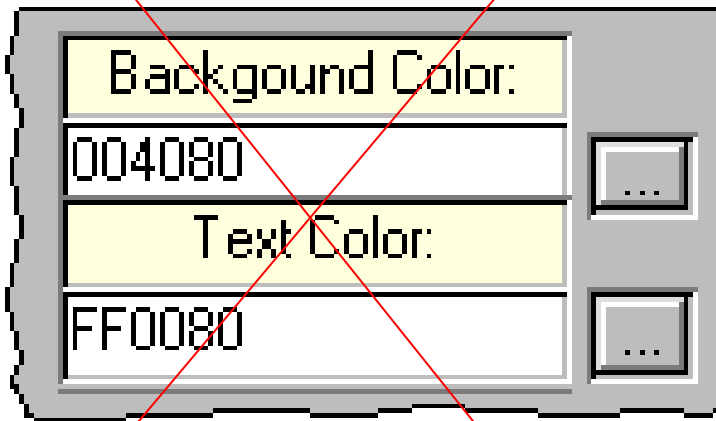
1. Простой и естественный диалог:

- не должно присутствовать не относящейся к теме или редко используемой информации ;
- "лучше меньше да лучше»;
- информация, которая выводится на экран, должна появляться в порядке, соответствующем ожиданиям пользователя

Правила Нильсена-Молиха (Nielsen, Molich)

2. Говорите на языке пользователя:

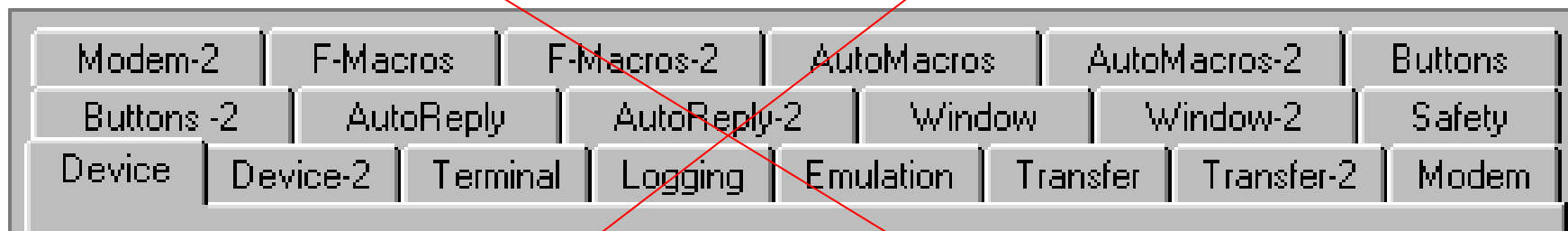
- используйте слова и понятия из мира пользователя;
- не используйте специфических инженерных терминов.



Правила Нильсена-Молиха (Nielsen, Molich)

3. Минимизируйте загрузку памяти пользователя:

- не заставляйте пользователя помнить вещи от одного действия к следующему;
- оставляйте информацию на экране до тех пор, пока она не перестанет быть нужной;
- хорошим стилем считается делать только один ряд закладок.



Правила Нильсена-Молиха (Nielsen, Molich)

4. Будьте последовательны:

- у пользователей должна быть возможность изучить действия в одной части системы и применить их снова, чтобы получить похожие результаты в других местах.

5. Обеспечьте обратную связь:

- дайте пользователю возможность видеть, какой эффект оказывают его действия на систему.

Правила Нильсена-Молиха (Nielsen, Molich)

6. Обеспечьте хорошо обозначенные выходы:

- Если пользователь попадает в часть системы, которая его не интересует, у него всегда должна быть возможность быстро выйти оттуда, ничего не повредив.

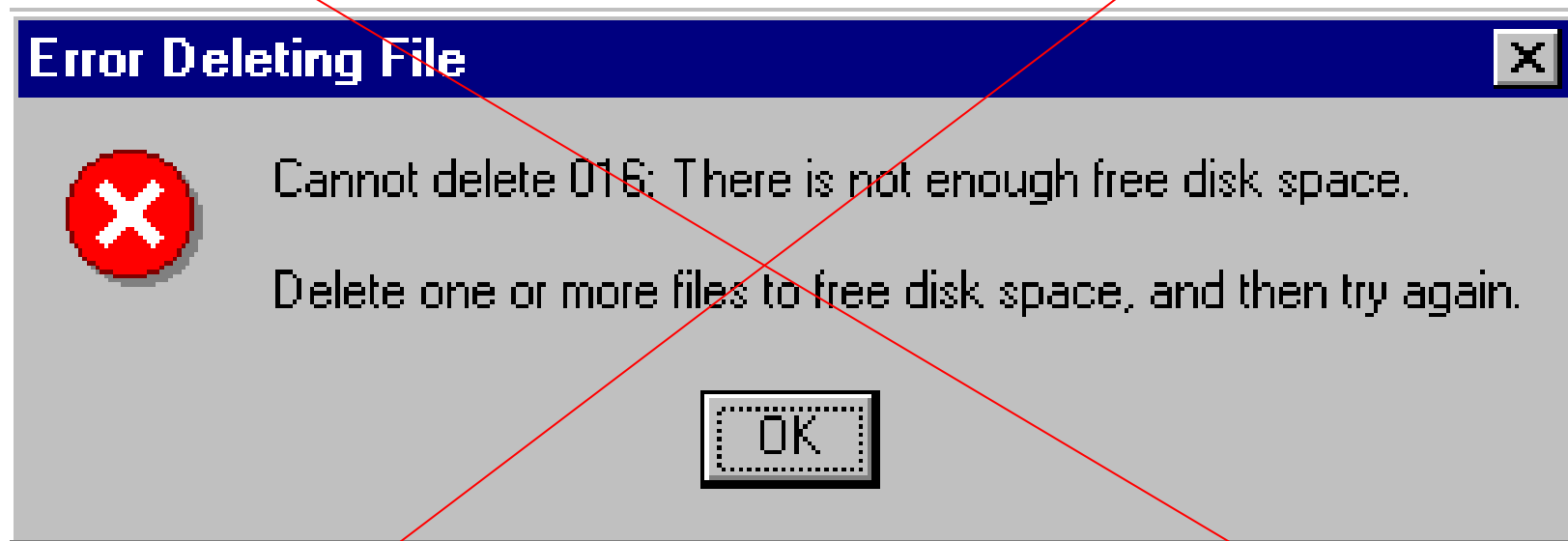
7. Обеспечьте быстрые клавиши и ярлыки:

- элементы быстрого доступа могут помочь опытным пользователям избегать длинных диалогов и информационных сообщений, которые им не нужны.

Правила Нильсена-Молиха (Nielsen, Molich)

6. Хорошие сообщения об ошибках.

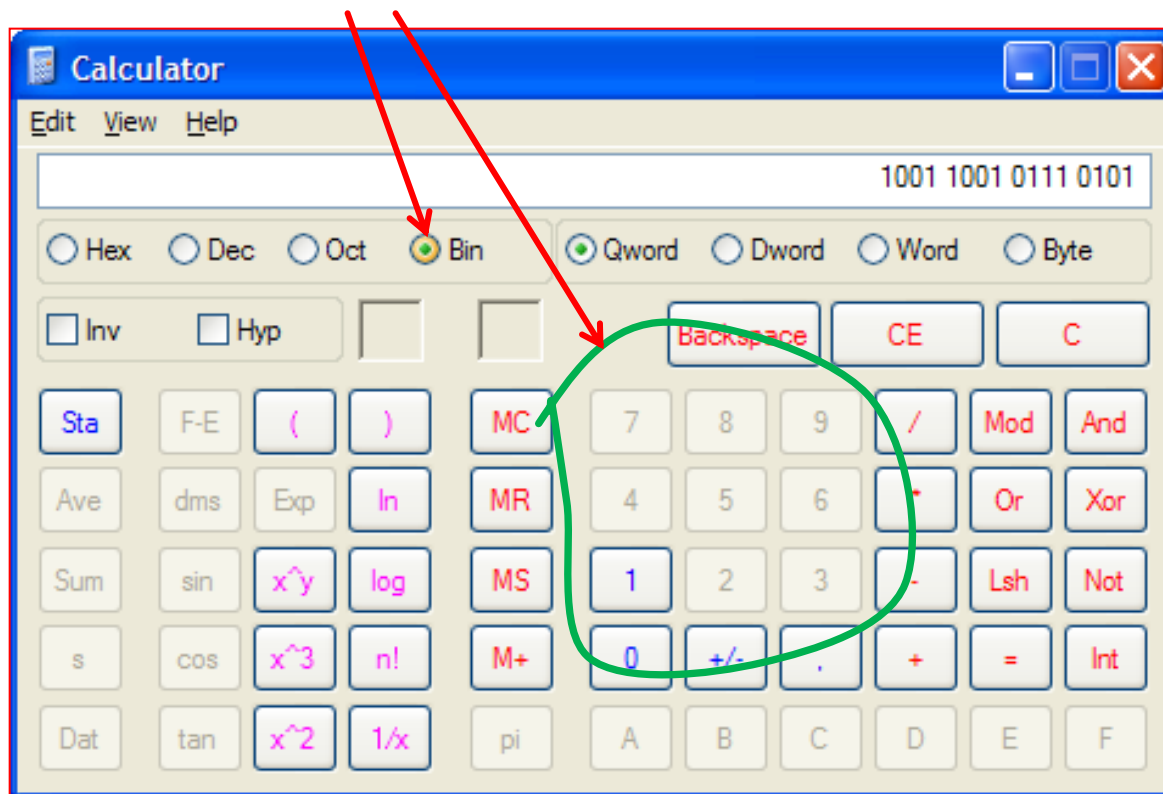
- Хорошее сообщение об ошибке помогает пользователю понять, в чём проблема и как это исправить.



Правила Нильсена-Молиха (Nielsen, Molich)

6. Предотвращайте ошибки

Всегда, когда вы пишете сообщение об ошибке, вы должны спросить себя, можно ли избежать этой ошибки?



Золотые правила построения интерфейсов

Принципы организации графического интерфейса

1. Принцип кластеризации:

- организуйте экран в виде визуально разделённых блоков с похожими элементами управления, предпочтительно с названием для каждого блока.;
- подобные команды должны быть в одном меню: это позволяет им быть визуально близко и идти под одним заголовком;
- команды, относящиеся к некоторой конкретной области функциональности, могут также быть показаны в диалоговых боксах, опять таки в визуально определяемых блоках.

Принципы организации графического интерфейса

2. Принцип "видимость отражает полезность" :

- делайте часто используемые элементы управления заметными, видимыми и легко доступными;
- прячьте или сжимайте редко используемые элементы;

Принципы организации графического интерфейса

3. Принцип интеллектуальной последовательности:

- используйте похожие экраны для похожих функций.;
- экраны не должны выглядеть одинаково, если в действительности они должны отражать совершенно другие вещи;
- предупреждение о критической ошибке в системе реального времени должно иметь вид, значительно отличающийся от экрана помощи или информационного сообщения.

Принципы организации графического интерфейса

4. Принцип "цвет как приложение":

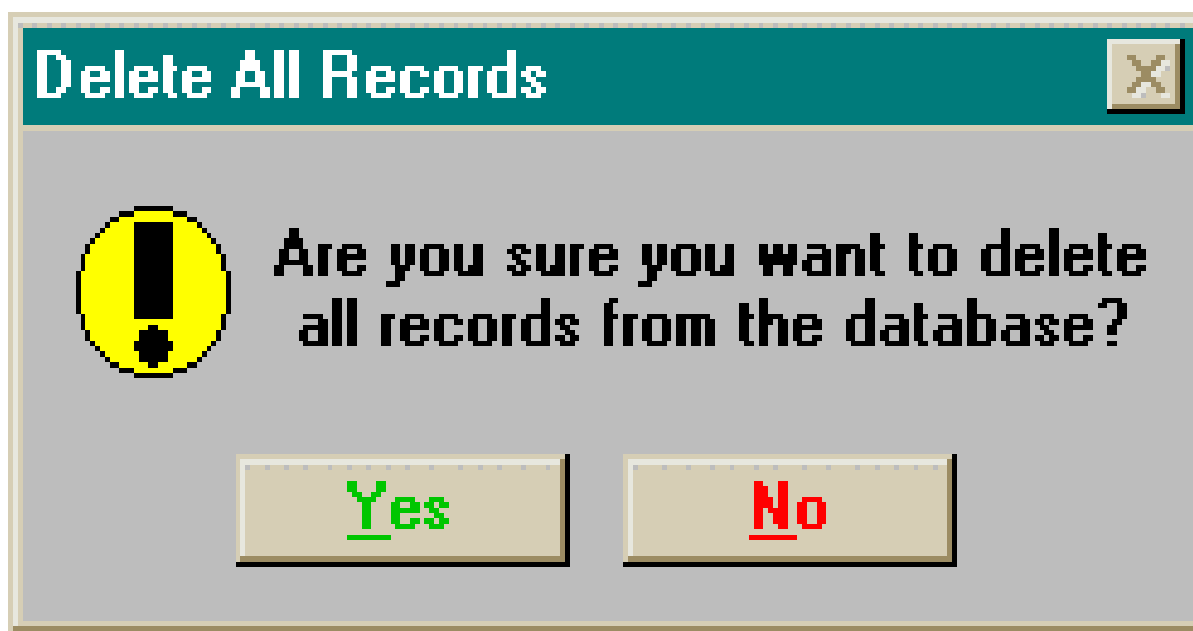
- не полагайтесь на цвет как носитель информации;
- используйте цвет умеренно, чтобы лишь акцентировать информацию, передаваемую другими средствами;
- обязательно дайте дополнительные ключи для пользователей, не способных воспринимать изменения цвета;
- помните, что многие пользователи могут, и часто это делают, изменить цвет окон, подсветок и других системных объектов. Стройте ваш продукт так, чтобы он работал с пользователем, а не боролся с ним.

Принципы организации графического интерфейса

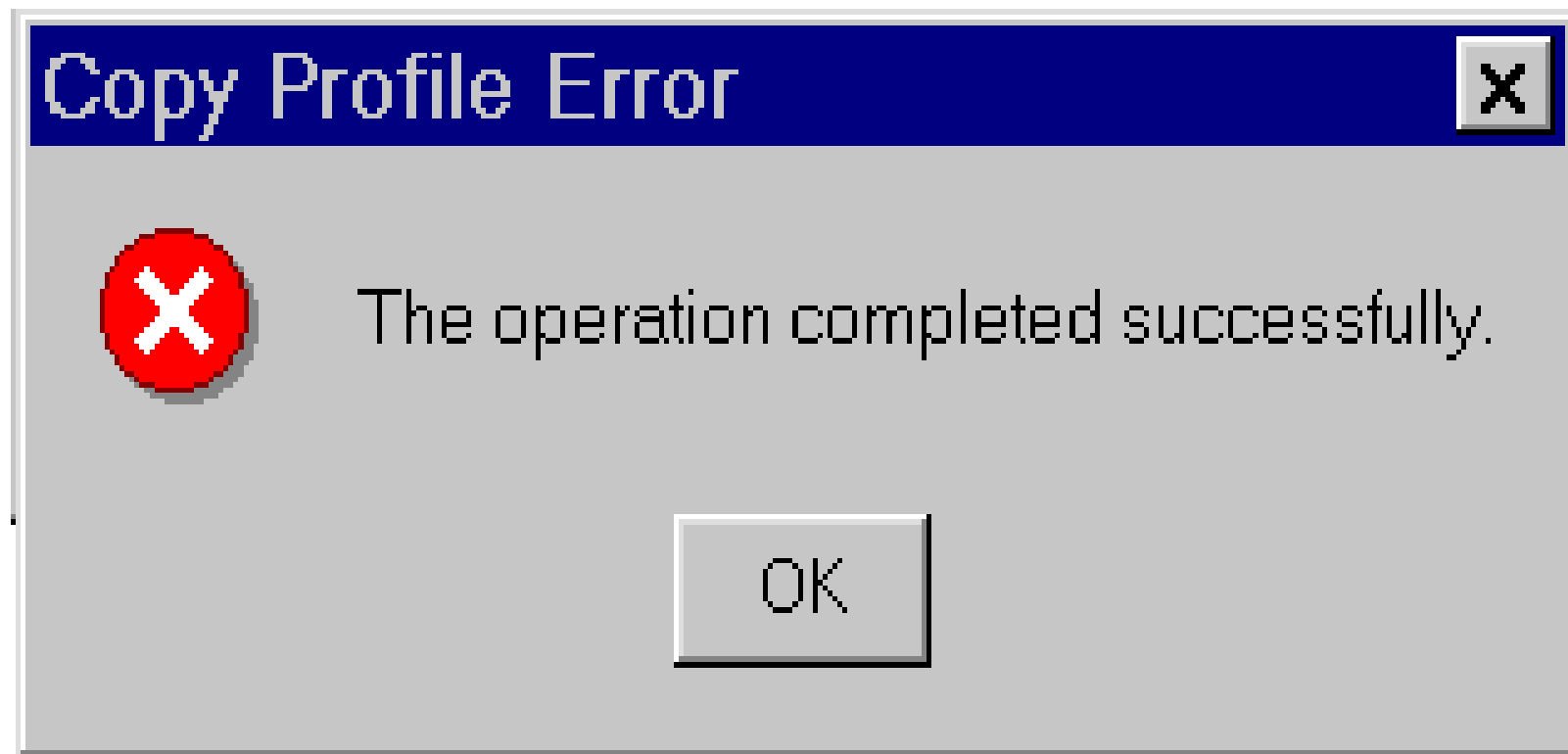
5. Принцип уменьшения беспорядка:

- не помещайте на экран слишком много всего;
- не пытайтесь наделить каждое меню собственным шрифтом или работать с большим набором размеров. Как правило, пользователи заметят не столько различия, сколько беспорядок.

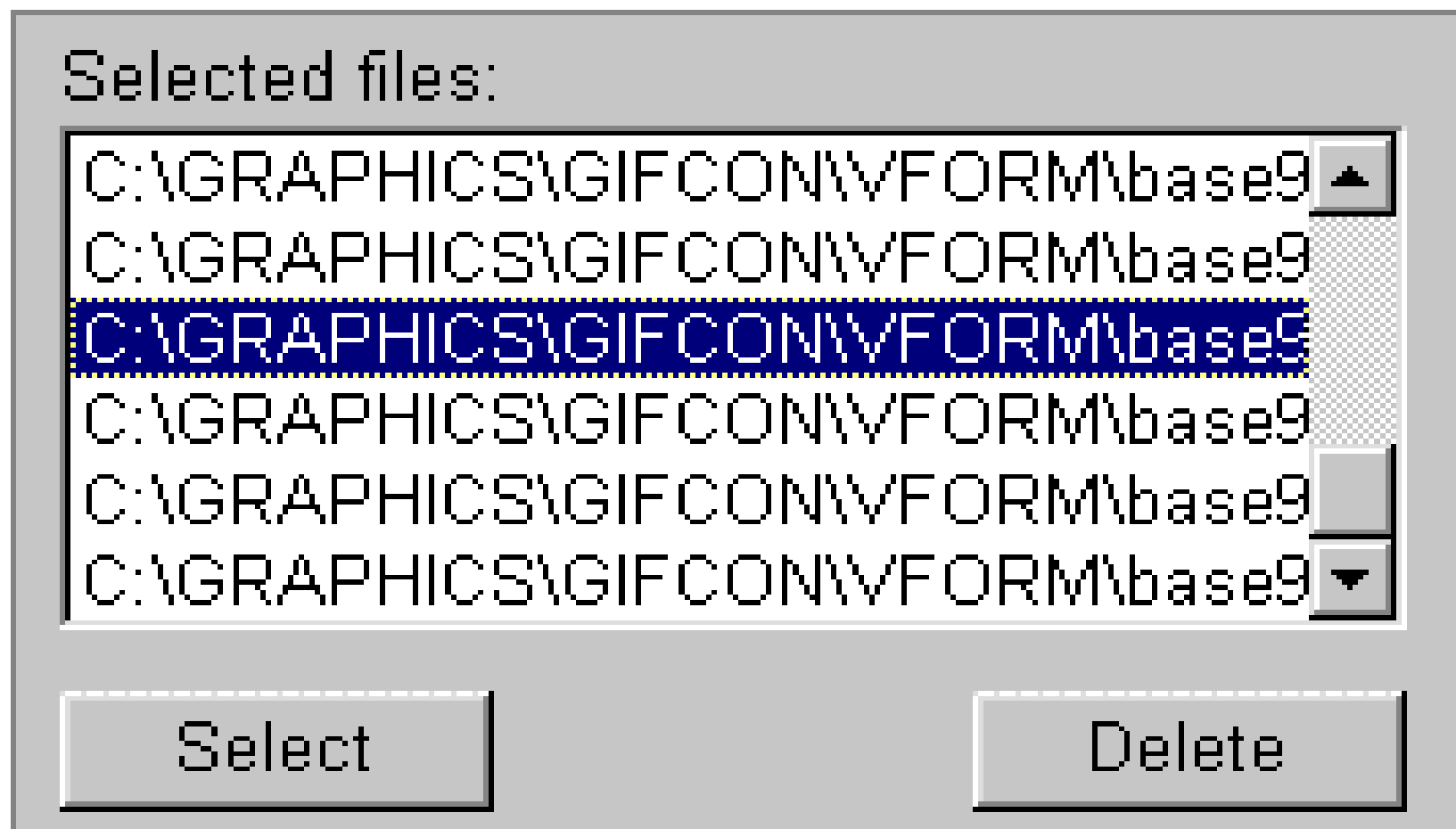
Как НЕ НАДО делать



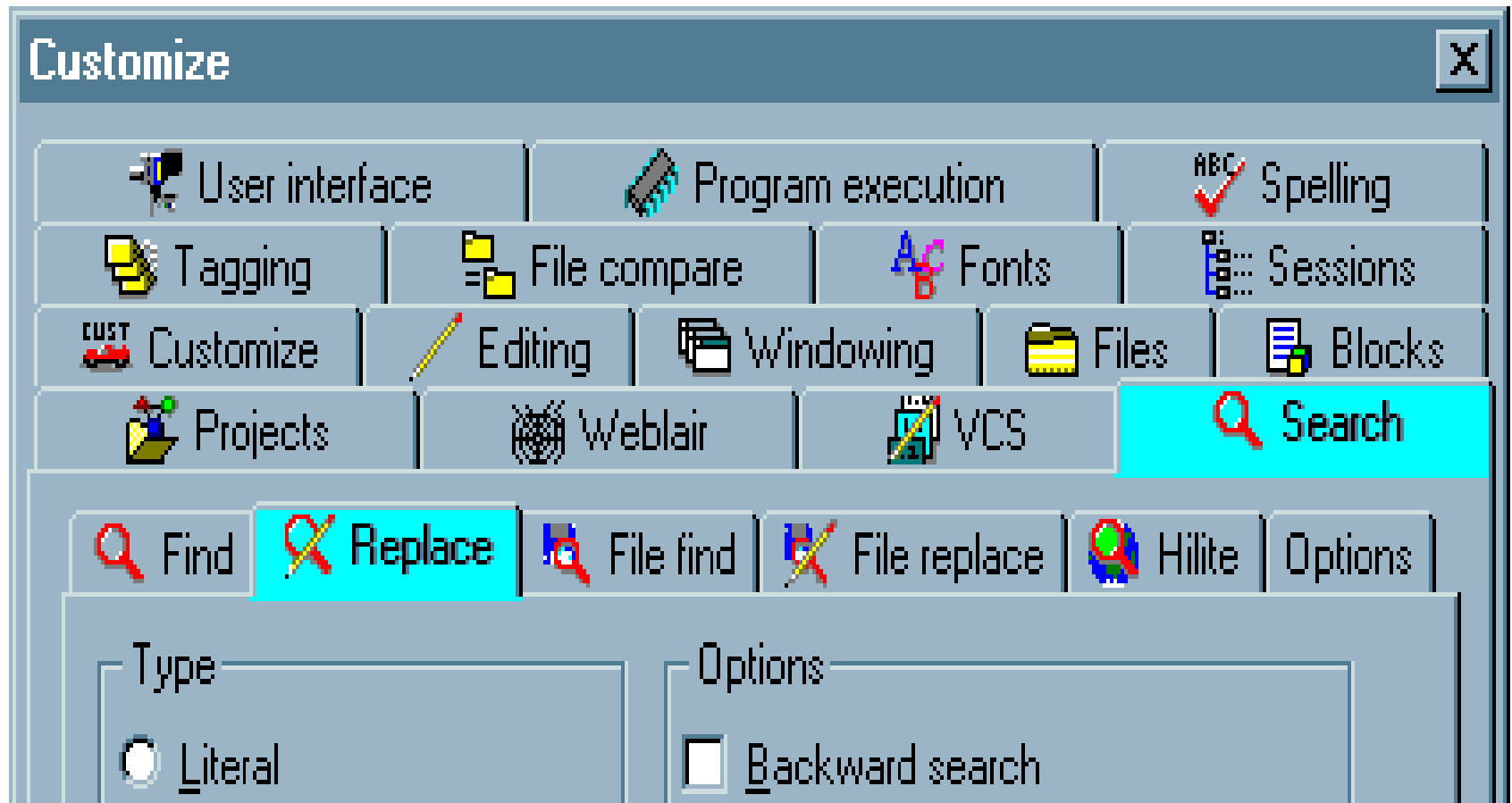
Как НЕ НАДО делать



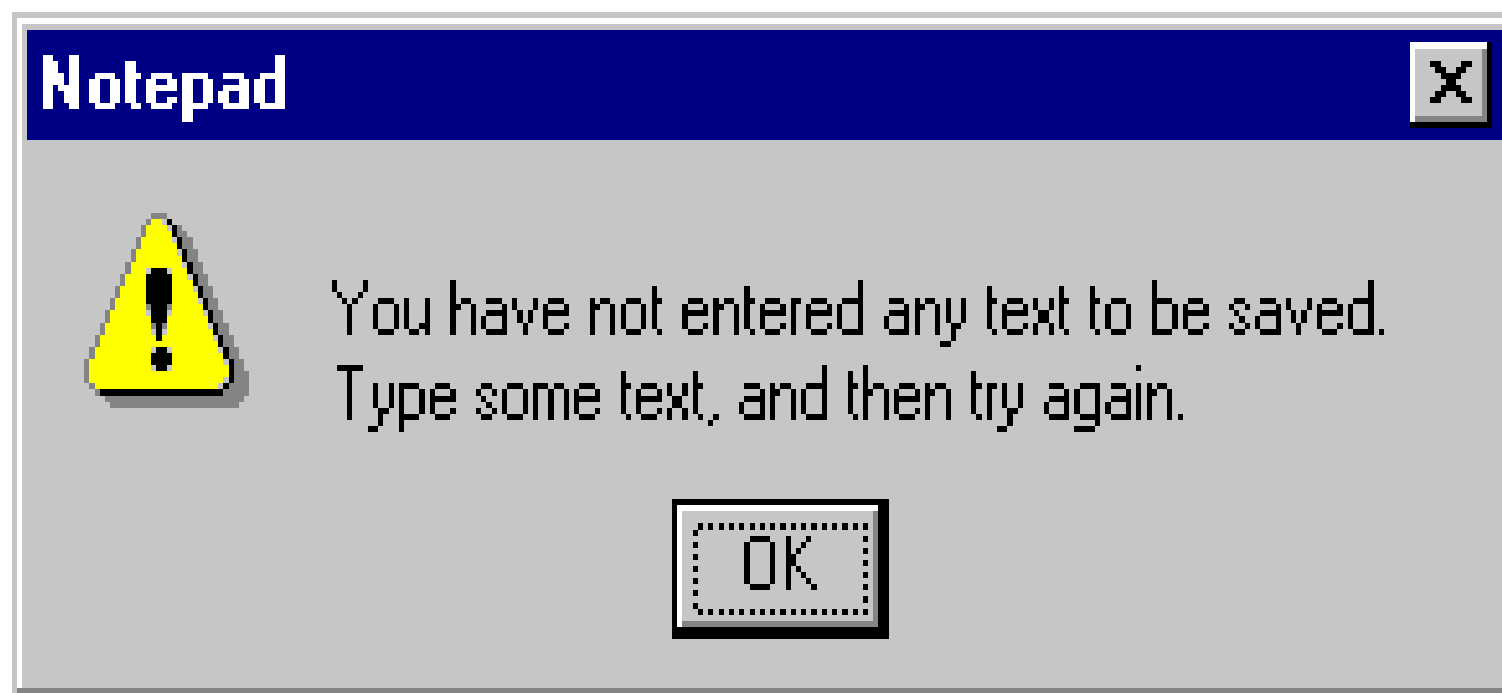
Как НЕ НАДО делать



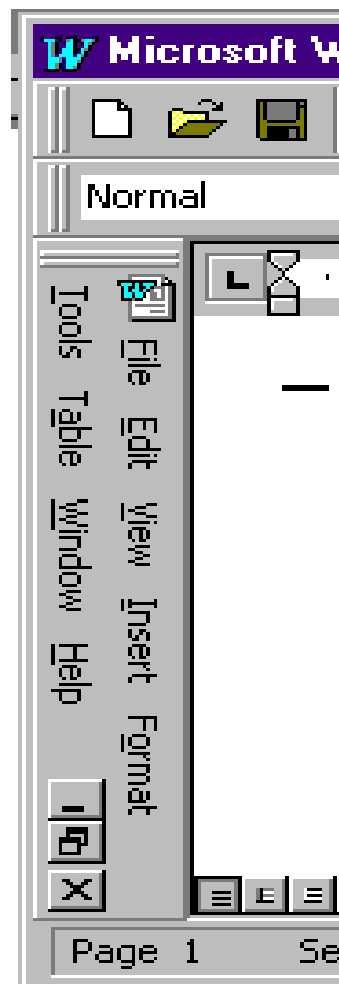
Как НЕ НАДО делать



Как НЕ НАДО делать



Как НЕ НАДО делать



Как НЕ НАДО делать

Whenever your local SMS Administrator sends you an actual software Package, the SMS Package Command Manager will appear (usually at network login time) displaying the available Package(s). The following screenshots display scenes similar to what you will see when you receive an actual SMS Package.

To start the demonstration, click the "CLICK HERE" button of the screen.

Как НЕ НАДО делать



Как НЕ НАДО делать

Form Title -- (appears above URL in most browsers and is used by WWW search engines)		Background Color:
Q&D Software Development Order Desk		FFFBF0
Form Heading -- (appears at top of Web page in bold type)		Text Color:
Q&D Software Development Order Desk		000080
E-Mail responses to (will not appear on page)	Alternate (for mailto forms only)	Background Graphic
dversch@q-d.com		
Text to appear in Submit button	Text to appear in Reset button	<input type="radio"/> Mailto
Send Order	Clear Form	<input checked="" type="radio"/> CGI
Scrolling Status Bar Message (max length = 200 characters)		
WebMania 1.5b with Image Map Wizard is here!!		
<< Prev Tab		Next Tab >>

Как НЕ НАДО делать

