



# Usability Testing

# Содержание

- Понятие Usability
- Понятие пользовательского интерфейса
- Основы человеческого восприятия
- Usability в жизненном цикле проекта
- Проектирование UI
- Usability тестирование
- Литература

# Содержание

- Стандарты, регламентирующие качество программного обеспечения (ПО)
- Usability, как атрибут качества, его характеристики
- Определение Usability
- Понятие интерфейса. Свойства современного интерфейса. Предназначение интерфейса
- Стандартизация и согласованность пользовательского интерфейса
- Usability в жизненном цикле проекта
- Основы когнитивной психологии
- Основы проектирования пользовательского интерфейса
- Методы Usability: проектирование, прототипирование, usability тестирование, экспертная оценка usability продукта

## **В рамках курса предполагается проведение практических занятий по:**

- созданию документа GUI Specification
- разработке метафоры для реализации UI Системы
- созданию прототипа GUI Системы методом бумажного прототипа
- проведению анкетирования удовлетворенности пользователя интерфейса
- тестированию usability Системы методом контрольного списка интерфейса
- проведению оценки usability Системы методом контрольного списка интерфейса

# 1. Понятие Usability

- Стандарты, регламентирующие качество ПО
- Usability, как атрибут качества
- Эргономические показатели качества продукта
- Вопросы, которые решает Usability

# Usability, как атрибут качества



**Согласно стандарту ISO 9126 (ГОСТ Р ИСО / МЭК 9126-93) — «Информационная технология. Оценка программного продукта. Характеристики качества и руководство по их применению», п.4.3 одним из атрибутом качества программного продукта является:**

***«Usability - способность программного продукта быть понятным, освоенным, использованным и привлекательным для пользователя, в том случае, когда он используется в определенных условиях»***

# Стандарты, определяющие Usability

**Согласно стандарту ISO 9241-11:**

*«Usability- степень, в которой продукт может быть использован определенными пользователями для достижения поставленных целей эффективно, экономично и с удовольствием в заданном контексте использования»*

Человеко-центрированный подход характеризуется: вовлечением пользователей и ясным пониманием их требований; соответствующим распределением функций между пользователями и технологией; итерациями дизайнерских решений – ISO 13407

# Что такое usability

## Согласно Бену Шнайдеру:

Показатель количества ошибок, скорости взаимодействия с продуктом, скорости обучения навыкам взаимодействия и субъективной удовлетворенности определенных пользователей продукта, достигающих определенных целей/мотивов в Контексте использования



# Пример контекста использования

Программный продукт будет использоваться оператором call center.  
Какие особенности среды следует учесть ?



**(!)** Usability не может быть измерен без учета контекста использования.  
Из дословного перевода следует, «usability» - пригодность продукта к  
использованию, а использование – это ***процесс взаимодействия  
пользователя, системы, объекта и среды деятельности.***

# Usability начинается с философии

*Андрей Туполев: «Эта машина летать не будет» , - в ответ на показанные молодым конструктором чертежи нового самолета.*

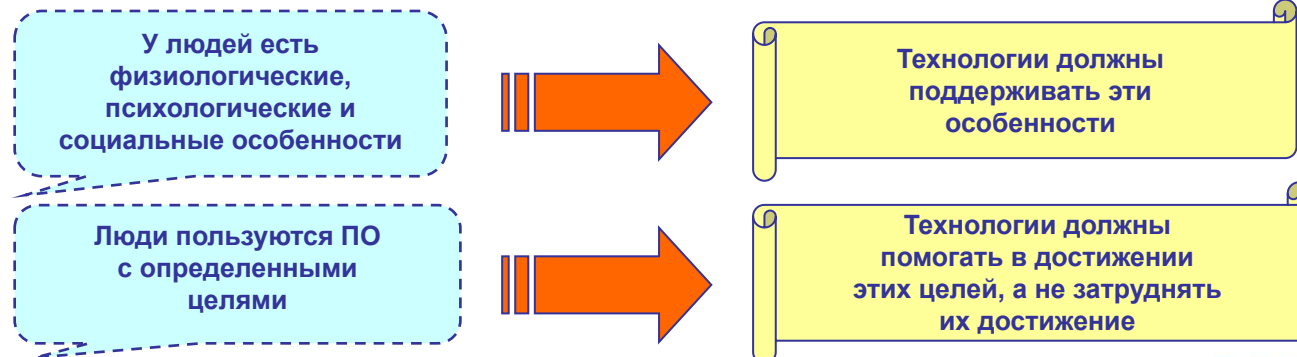
*«Почему? - удивился тот. - Я все просчитал - сопротивление, аэродинамику...»*

*«Потому что она некрасивая» - ответил Туполев*

*«If the user can't use it, it doesn't work»- Susan Dray, President of Dray & Associates, Inc., usability consultant*

Usability начинается с убеждения, связанного с четырьмя моментами:

- ✓ usability означает фокусировку на пользователях
- ✓ люди используют эффективные продукты
- ✓ пользователи – занятые люди, выполняющие свои задания
- ✓ пользователи решают, когда продукт прост в использовании



# Определения Usability

**Usability** - это:

- **степень удобства** чего-либо (ПО)
- **свойство** эффективного в использовании **интерфейса** программного продукта
- **мера качества пользовательского опыта**, приобретенного при взаимодействии с продуктом или системой
- **раздел эргономики**, посвящённый разработке ПО. Эргономика делает упор на физиологическом удобстве, а usability рассматривает психологические аспекты в большей степени, чем эргономика
- **подход к разработке продукта**, который включает проходящую через весь производственный цикл обратную связь с пользователем для минимизации затрат и для производства продукта, отвечающего нуждам пользователей

# Два аспекта Usability ПО

- **В Usability присутствуют два аспекта:**
  - С одной стороны это методика контроля и обеспечения качества, которая помогает вам узнать, что в вашей системе работает на пользователя, а что – нет
  - С другой стороны, это система понятий, которая призвана обеспечить власть человека над тем, что он сам же и создал

*Основной принцип: «Не люди для системы, а система для людей»*

# Определение Usability в свете человеческого фактора

- **Определение Usability, как системы веры в определенные права человека:**
  - ✓ Право человека быть выше технологий
  - ✓ Право человека на управление
  - ✓ Право человека на простоту
  - ✓ Право человека на свое время



*По исследованиям, 30% посетителей не ждут окончания загрузки web-страницы, если это время превышает 8 секунд, а 70% посетителей "уходят раздраженными", если это время превышает 12 секунд*

*«Your user is just one click away from your competitor»*

# Главный закон usability



“НЕ ЗАСТАВЛЯЙТЕ МЕНЯ ДУМАТЬ!”

**IT-GOST.RU**

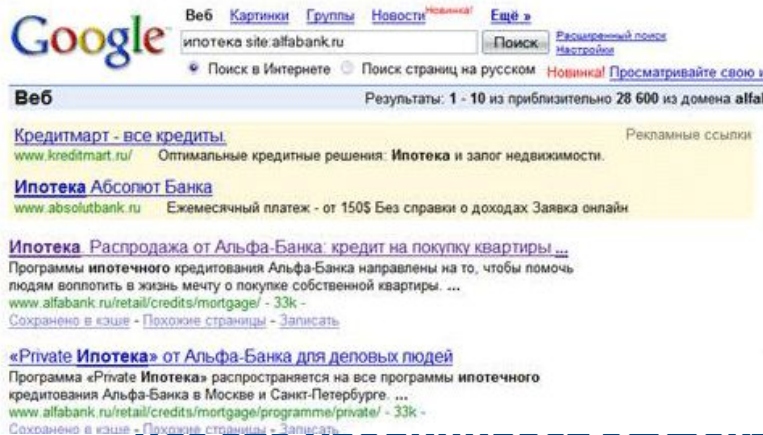
# Характеристики атрибута качества - Usability



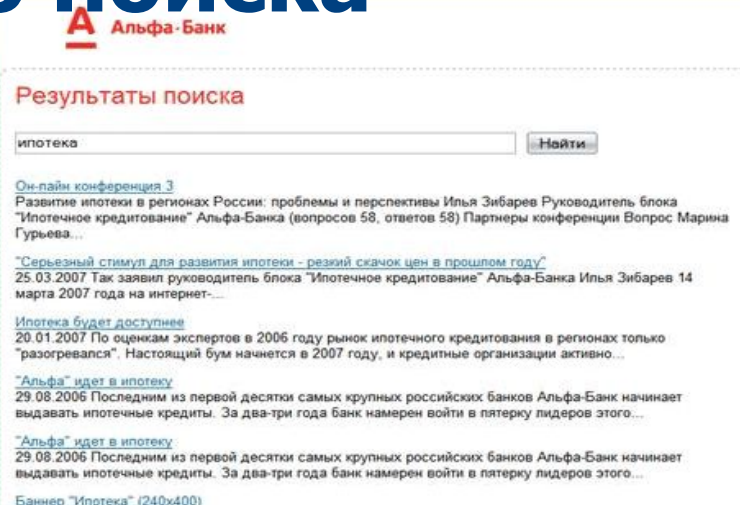
- **Usability** – это *качественный признак*, который определяет удобство интерфейса и легкость в его использовании. *Несколько минут работы дает субъективное мнение о сайте: «удобно» или «неудобно»*
- **Количественные характеристики Usability:**
  - **Обучаемость** - время, которое тратит неподготовленный пользователь на обучение правилам пользования Вашим интерфейсом
  - **Эффективность** - показатель того, насколько быстро и эффективно пользователь может выполнять свои задачи, используя Ваш интерфейс. *(пример не эффективного поиска)*
  - **Запоминаемость** - фактор, определяющий то, насколько быстро пользователь сможет вспомнить принципы работы с Вашей системой, если ему придется столкнуться с ней через определенный промежуток времени
  - **Ошибки** - количество ошибок, которые пользователь может совершить, используя Вашу систему
  - **Удовлетворенность пользователя** от использования Вашего интерфейса

# Повышение эффективности на примере эффективного поиска

- Результат поиска по запросу «Ипотека» на сайте [www.alfabank.ru](http://www.alfabank.ru) – получили совершенно ненужную информацию:
- Если использовать встраиваемую форму поисковой системы, то первый же результат – вся информация об ипотеке Альфа-банка!



**Все это увеличивает эффективность поиска, а следовательно, эффективность работы сайта (страницы системы)**



- Поиск по ключевым словам, по первому вхождению, использование маски
- Расширенный поиск, задание запросов с использованием логических операций
- Использование вертикального поиска (искать не по всему сайту, а ограничить поиск какой-либо одной сферой : Finance, Images, News, Video etc.)



# Эргономические цели и показатели качества программного продукта (ПП)

- С точки зрения эргономики, самое важное в программе — создать такой пользовательский интерфейс, который сделает работу эффективной и производительной, а также обеспечит удовлетворенность пользователя от работы с программой

## Цели:

- **Эффективность работы** - означает обеспечение точности, функциональной полноты и завершенности при выполнении производственных заданий на рабочем месте пользователя
- **Производительность работы** отражает объем затраченных ресурсов при выполнении задачи, как вычислительных, так и психофизиологических
- **Удовлетворенность пользователя от работы** тесно связана с комфортностью его взаимодействия с приложением, и способствует сохранению профессиональных кадров на предприятии Заказчика за счет привлекательности работы на данном рабочем месте

# Показатели качества ПП для эффективности

- Создание UI должно быть нацелено на показатели эффективности:
  - **Точность работы** определяется тем, какой процент ошибок совершил пользователь в процессе работы с ПП
  - **Функциональная полнота** - определяется через процент применения отдельных функций в ПП
  - **Завершенность работы** описывает степень исполнения производственной задачи средним пользователем за определенный срок, долю неудовлетворенных заявок, процент незаконченных результатов, а также число пользователей, которые выполнили задание в фиксированные сроки

Не надо бояться сложности системы, надо избегать такого интерфейса, который не соответствует алгоритму решения пользовательских задач

# Показатели качества ПП для производительности работы

- Дизайн UI должен обеспечивать минимизацию усилий пользователя при выполнении работы и приводить к:
  - сокращению длительности операций чтения, редактирования и поиска информации
  - уменьшению времени навигации и выбора команды
  - повышению общей продуктивности пользователя, заключающейся в объеме обработанных данных за определенный период времени
  - увеличению длительности устойчивой работы пользователя и др.

Сокращение непроизводительных затрат и усилий пользователя - важная составляющая качества программного продукта

# Показатели качества ПП для удовлетворенности пользователя

- Высокая удовлетворенность от работы достигается в случае:
  - Прозрачной для пользователя навигации и целевой ориентации в программе. Главное, чтобы было понятно, куда идем, и какую операцию программа после этого шага произведет
  - Ясности и четкости понимания пользователем текстов и значения икон. В программе должны быть те слова и графические образы, которые пользователь знает или обязан знать по характеру его работы
  - Быстроты обучения при работе с программой, для чего необходимо использовать преимущественно стандартные элементы взаимодействия, их традиционное или общепринятое их расположение
  - Наличия вспомогательных средств поддержки пользователя (поисковых, справочных, нормативных), в том числе и для принятия решения в неопределенной ситуации (ввод по умолчанию, обход «зависания» процессов и др.).

Удобный интерфейс помогает пользователю справиться с усталостью и напряжением при работе в условиях высокой ответственности за результат

# Какие вопросы решает Usability

## Внедрение Usability в процесс разработки ПО и раннее его использование решает следующие проблемы:

- Полученный результат равен ожидаемому, т.к. заказчик «видит» систему ещё до начала основной разработки
- Достигнут приемлемый показатель эффективности использования
- Увеличивается конкурентоспособность продукта
- Уменьшается нагрузка на службу поддержки
- Лёгкость дальнейшего развития системы
- Увеличению степени удовлетворения пользователей и Заказчика



*«Успех вашего проекта — это счастливые пользователи»*

*84% интернет-пользователей  
считают Интернет необходимым.*

*Примерно такова же потребность в e-mail.*

*85% используют Интернет каждый день*

# Выгода добавления Usability в проект



- **Увеличении продуктивности:** Среднестатистический программный продукт имеет 40 недостатков дизайна, ослабляющие его использование. Продуктивность возрастает на 4-9% ежегодно, если программный продукт спроектирован с usability
- **Увеличении объема продаж:** На объединение «железа», ПО, сетей и людей для поддержки и обучения в 1996 было затрачено 500 миллиардов долларов в США и более 1 триллиона долларов в мире
- **Уменьшении затрат на обучение пользователей :** Изучено, что время обучения новых пользователей на стандартном ПК - 21 час, а на наиболее удобном компьютере – всего лишь 11 часов
- **Сокращении времени производства:** Статистика 1995 года: 9-16% программных проектов закончены вовремя и по бюджету. Когда ПП завершены, они имеют только 42% изначально планируемых свойств. Более 30% ПП остановлены перед завершением из-за не отвечающего требованиям пользовательского дизайна. Реализация usability-технологии демонстрирует сокращение цикла производства продукта на 33-50%
- **Сокращение на переделку продукта:** Каждый доллар, вложенный в человеко-центрированный подход, возвращается в размере от 2 до 100 долларов. 80% затрат происходит на ремонтной фазе жизненного цикла проекта

- **Опыт взаимодействия (User Experience, UX)** - ощущение и реакция человека, вследствие использования или предполагаемого использования продукта, системы или услуги ( ISO 9241-210).
- **Проектирование взаимодействия (Interaction Design)** - дисциплина, направленная на определение поведения систем и продуктов, с которыми взаимодействует пользователь.

Проектирование взаимодействия является человеко-ориентированным методом проектирования пользовательских интерфейсов.

## **2. Понятие пользовательского интерфейса**

- Определение пользовательского интерфейса(UI)
- Этапы развития и усовершенствования UI
- Принципы и свойства современного UI
- Согласованность



# Пользовательский интерфейс. Зарождение исследований

- Началом интерактивных вычислений и, следовательно, *исследований человеко-машинного интерфейса* принято считать 1956 г., когда на конференции UNESCO по обработке информации К. Стрейчи предложил режим разделения времени при решении задач на компьютерах
- Разработка любого прикладного программного обеспечения, как правило, подразумевает создание пользовательского интерфейса (UI)
- Рост мощности компьютеров привел к необходимости увеличения затрат на диалоговую компоненту программного обеспечения



# Понятие пользовательского интерфейса (UI)

- **Пользовательский интерфейс** – это совокупность программных и аппаратных средств, обеспечивающих взаимодействие пользователя с компьютером.
- Иными словами, **пользовательский интерфейс** - способ, которым вы выполняете какую-либо задачу с помощью какого-либо продукта, а именно совершаемые вами действия и то, что вы получаете в ответ.
  - Основу такого взаимодействия составляют диалоги
  - Под диалогами понимают регламентированный обмен информацией между человеком и компьютером, осуществляемый в реальном масштабе времени и направленный на совместное решение конкретной задачи
  - Каждый диалог состоит из отдельных процессов ввода/вывода, которые физически обеспечивают связь пользователя и компьютера
  - Обмен информацией осуществляется передачей сообщения

# Актеры в «театре» UI

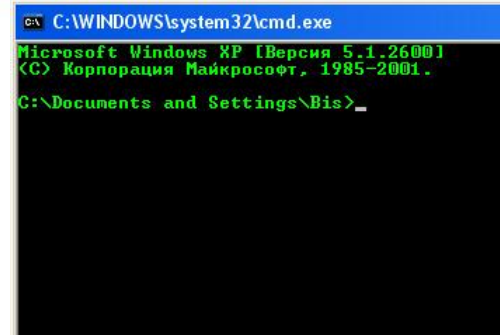
- **Пользовательский интерфейс включает в себя три понятия:**
  - общение пользователя с приложением
  - общение приложения с пользователем
  - язык общения
- **Пользователь генерирует сообщения следующих типов:**
  - запрос информации
  - запрос помощи
  - запрос операции или функции
  - ввод или изменение информации
- **От приложения пользователь получает:**
  - подсказки или справки
  - информационные сообщения, требующие ответа
  - приказы, требующие действия
  - сообщения об ошибках и другую информацию
- **Язык общения определяется разработчиком программного приложения**



# Этапы развития и совершенствования UI

## ■ (1) Интерфейс командной строки

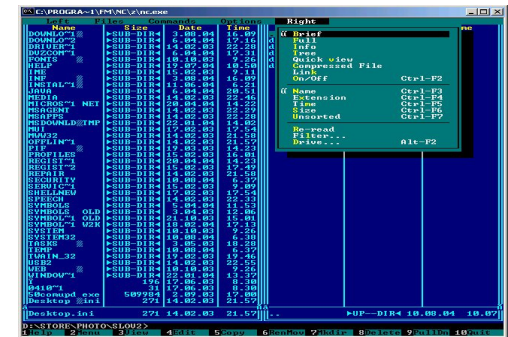
- многочисленность команд
- отсутствие стандарта для приложения, что ограничивало круг его применения



## ■ (2) Интерфейс с иерархическим меню

Были предприняты попытки упростить командный интерфейс

*Пример: создание Norton Commander (NC) – возможность выбора команд из списка (меню)*



## ■ (3) Современный графический интерфейс

Современный интерфейс реализует технологию

WIMP = "window, icon, menu, pointing device" (окно, значок, меню, манипулятор)



# Принципы современного UI

- Применение мыши
- Выбор команд из меню
- Предоставление программам отдельных окон
- Использование для обозначения программ образов в виде пиктограмм
- Документирование
- Включение встроенных коммуникационных возможностей

# Свойства современного UI

- Наглядность
- Простота
- Конкретность
- Удобство
- Богатство возможностей

*Например, удобство интерфейса и богатство возможностей делают Windows оптимальной системой для повседневной работы*

# Предназначение UI

- Одной из важнейших функций интерфейса является формирование у пользователя одинаковой реакции на одинаковые действия приложений, их **согласованность**
- Единообразии интерфейсов сводит к минимуму процесс обучения работе с любым приложением

**Концепция согласованности** состоит в том, что при работе с компьютером у пользователя формируется система ожидания одинаковых реакций на одинаковые действия

# Пример несогласованного интерфейса

- В пакете электронной почты America Online для получения бланка нового письма вместо привычного сочетания клавиш **Ctrl+N** используется сочетание клавиш **Ctrl+M** (от «mail»).
- Нетрадиционная комбинация горячих клавиш привела к снижению скорости обучения и увеличению числа ошибок.





# Стандартизация и согласованность UI

- Стандартизация и согласованность интерфейса экономят время пользователя и разработчика
- Согласование должно быть выполнено в трех аспектах:
  - *физическом*, который относится к техническим средствам (физическое отображение)
  - *синтаксическом*, который определяет последовательность и порядок появления элементов на экране (язык общения) и последовательность запросов (язык действий)
  - *семантическом*, который обусловлен значениями элементов, составляющих интерфейс

# Согласованность в вопросе Ok/Cancel

Какой способ размещения кнопок Ok/Cancel выбрать для проектируемого приложения?



# OK–Cancel или Cancel–OK?

## *Парадигма OK–Cancel:*

OK является более часто используемой кнопкой, а так как природный порядок чтения западных стран слева-направо, то она будет достигнута взглядом быстрее, находясь слева.

Также, при таком расположении OK будет быстрее достигнута пользователями, использующими клавиатуру (и клавишу tab) в качестве манипулятора.

## *Парадигма Cancel-OK:*

OK способствует продвижению процесса вперед, потому должна быть размещена на месте кнопки Next в паре Previous/Next, то есть справа.

Лучшим решением будет использовать тот подход, который регламентирован стандартом платформы, для которую разрабатывается приложение

# Преимущества стандартизации и согласованности UI



- **Для пользователя:**
  - уменьшается время изучения
  - а затем использования системы
  - сокращается число ошибок
  - появляется чувство комфортности и уверенности
- **Для разработчика:**
  - позволяет выделить общие блоки интерфейса
  - стандартизировать отдельные элементы и правила взаимодействия с ними
  - сократить время проектирования новой системы

# Простое должно оставаться простым



Насколько труднее установить время на электронных наручных часах с четырьмя кнопками, чем выполнить *то же самое* действие на механической модели часов ?



Наблюдается тенденция неоправданного усложнения современных интерфейсов. Простота использования - одно из конкурентных преимуществ сегодня.



## 3. Основы человеческого восприятия

- Когнитивная психология
- Локус внимания
- Формирование привычек
- Кратковременная и долговременная память

- **Когнитивная психология** - раздел психологии, изучающий когнитивные, то есть познавательные процессы человеческого сознания.
- **Характеристики процесса человеческого познания:**
  - Ощущение и восприятие
  - Обучение и запоминание
  - Внимание и речь
  - Принятие решений

(!) Прежде чем пытаться учесть различия между отдельными пользователями следует сосредоточить внимание на том, что является общим для всех людей с точки зрения требований к интерфейсу, на то, что справедливо для каждого из нас вне зависимости от пола, возраста, культурного происхождения и уровня компьютерной грамотности.



# Модель обработки информации человеком



- Зрительная и слуховая информация, передаваемая пользователю прикладным ПО, должна находиться в воспринимаемом им диапазоне
- Из сенсорных буферов данные передаются в кратковременную память, где включаются процессы распознавания образов, направленные на устранение неопределенности, определение контекста

# Локус<sup>1</sup> внимания

- Когда вы находитесь в бодрствующем и сознательном состоянии, вашим локусом внимания является какая-то деталь или объект окружающего мира или идея, о которой вы думаете.
- Из всех объектов или явлений окружающего мира, которые вы воспринимаете с помощью своих чувств или воображения, в каждый момент времени вы можете сконцентрироваться только на **ОДНОМ**.
- **Когда некоторая информация становится локусом внимания, она перемещается в кратковременную память. Там она будет храниться в течение 10 секунд.**

**Вывод №1:** при смене фокуса (напр., на сообщение об ошибке) предмет фокуса теряется вместе с содержимым краткосрочной памяти, что снижает продуктивность.

**Вывод №2:** если операция выполняется более 10 секунд, внимание пользователя полностью рассеивается. В идеале длительные процессы должны проходить в фоновом режиме, позволяя пользователю заняться другой работой.

# Формирование привычек

- Любая последовательность действий, которую вы регулярно выполняете, становится, в конце концов, *автоматичной*. Набор действий, составляющих последовательность, становится как бы одним действием.
- **Неизбежность формирования привычек имеет свои следствия:**
- Например, при выполнении необратимого действия (напр., удаление файла), ПП задают вопрос: «Вы уверены?» с вариантами Y («Да»), либо N («Нет») в качестве ответа. Так как ошибки случаются редко, вы обычно отвечаете Y («Да») на любую команду.
- Из-за постоянного повторения ввод Y вскоре становится привычным действием и превращается в часть действия по удалению файлов.
- **В результате вы, не останавливаясь и не проверяя собственное намерение, вводите Y.** Таким образом, запрос компьютерной системы, предназначенный служить в качестве меры безопасности, из-за привычки становится бесполезным и только усложняет обычный процесс удаления файлов.

**Вывод:** не существует идеального способа подтверждения операции.  
Хороший интерфейс предоставляет возможность отката операций.

# Кратковременная память

- **В кратковременной памяти храниться контекст локуса внимания.**
- Вся обработка поступающей информации производится в кратковременной памяти, в этом кратковременная память сходна с ОЗУ в компьютерах. Особенности:
  - ✓ запоминается автоматически
  - ✓ для извлечения требуются очень небольшие усилия
  - ✓ объем сравнительно невелик ( $7 \pm 2$  элемента)
  - ✓ информация храниться преимущественно в звуковой форме
  - ✓ помещается гораздо больше элементов, если они сгруппированы
  - ✓ удерживается повторением
  - ✓ легко теряется за счет отвлечения или недостатка внимания

**Вывод №1:** количество элементов, предоставляемых пользователю для одновременного обозрения необходимо минимизировать.

**Вывод №2:** важную звуковую информацию стоит повторять дважды.

**Вывод №3:** ограничение  $7 \pm 2$  в большей степени касается пунктов меню, содержащих текста, чем, например, групп икон.

# Долговременная память

- **Работа с долговременной памятью человека** (информация, относящаяся к прошлому опыту).
- Особенности:
  - ✓ запоминание и извлечение требуют некоторых усилий
  - ✓ зависит от индивидуальной интерпретации
  - ✓ отличается огромным объемом
  - ✓ организуется иерархически за счет прикладывания усилий
  - ✓ осознается быстрее, чем вспоминается
  - ✓ повторение и семантический разбор помогают запоминанию



Легко ли запомнить число  $e=2,7118281828$  ?  
А если так:  $e=2.71$  1828 1828, где 1828 – день рождения Льва Толстого.

# Психофизиологические и морально-эстетические принципы построения UI

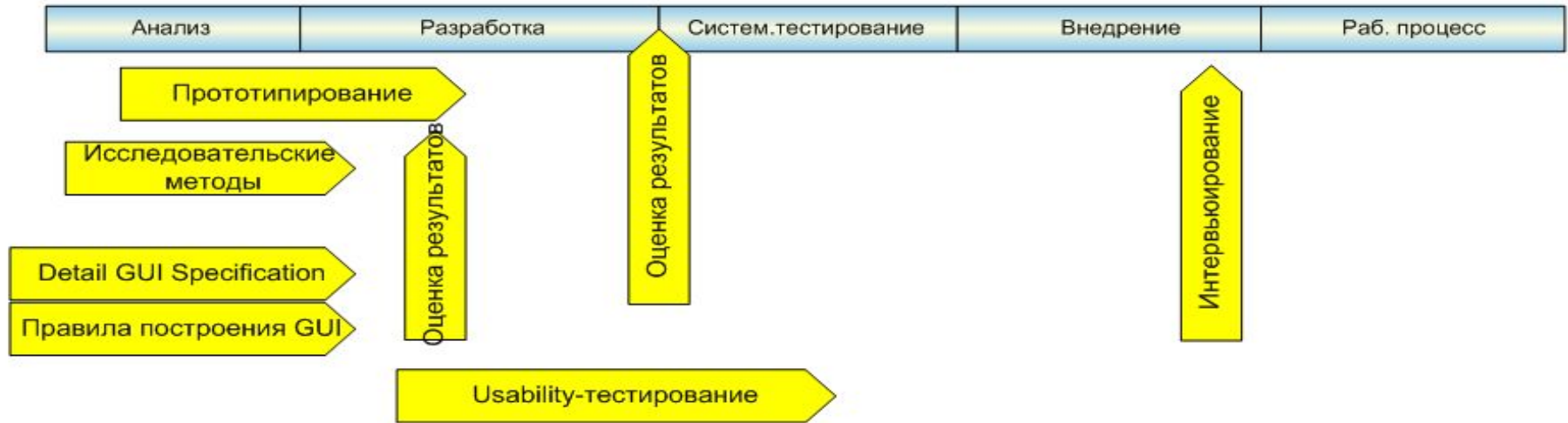
- **Взгляд человека избирателен - мы видим то, на что мотивированы.**  
 Кнопки диалогов в интерфейсе должны четко сигнализировать, в какой момент их нажимать. Эти сигналы крайне слабы, если кнопки выглядят одинаково. *Пример того, что разнообразие не нарушает эстетического нарушения: пульт дистанционного управления (приятно смотреться и легки в использовании): кнопки разных размеров, надпись на кнопке «Play» короткая и напоминает пиктограмму*
- **Синхронизация точки внимания пользователя и точки активности системы.** *Пример: элемент линейка прокрутки находится в противоречии с принципом психологии восприятия: у человека может быть только одна точка активного внимания. При использовании же линейки прокрутки приходится смотреть в две совершенно различные точки – на прокручиваемое изображение (не пора ли остановиться) и на линейку. Другие решения проблемы синхронизации точки взаимодействия: подсвечивание активного окна, подмигивание текстового курсора. Поиск курсора мыши пользователь до сих пор осуществляет подергиванием мыши*
- **Сохранять баланс между интерактивными возможностями программы и сложностью ее изобразительного ряда.** Простая программа не имеет права сложно управляться, это очевидно, но она и не имеет права на слишком изощренную графику

## 4. Usability в жизненном цикле проекта

- Usability в жизненном цикле проекта: активности и артефакты
- Группы usability активностей
- Этапы процесса проектирования UI
- Проектные роли, связанные с Usability

Практическое задание по созданию документа GUI Specification

# Usability в жизненном цикле проекта



Разработка usability интерфейса – это всегда итерационный процесс.

Каждый этап жизненного цикла проекта должен включать соответствующие usability активности.

Usability подразумевает подход к разработке ПП, основанный на получение обратной связи от пользователя **на протяжении всего производственного цикла.**



# Usability в жизненном цикле проекта: активности и артефакты

Project Initiation	Analysis & Requirements	Architecture Design	Development & Test	Acceptance & Post Release
Context analysis	Questionnaires	Static prototyping	Rapid prototyping	User surveys
Competitor analysis	Interviews	Heuristic evaluation	Heuristic evaluation	Post release testing
Evaluating existing systems	Observations		Usability testing	
	Focus Groups			
	Static prototyping			
<b>Project Management</b>				
Management Docs	Usability Requirements Set, GUI Specifications	GUI Specifications, Clickable Prototype, Usability Expert Report, Usability Testing Report	Codable Prototype, Boundary Classes Model, Usability Expert Report, Usability Testing Report	Usability Report, Usability CheckList

# Основные группы usability-активностей



- Проектирование и прототипирование UI
- Usability-исследования
- Usability-тестирование
- Экспертная оценка usability ПО, usability-консалтинг

# Проектирование и прототипирование GUI

Project Initiation	Analysis & Requirements	Architecture Design	Development & Test	Acceptance & Post Release
Context analysis	Questionnaires	<b>Static prototyping</b>	<b>Rapid prototyping</b>	User surveys
Competitor analysis	Interviews	Heuristic evaluation	Heuristic evaluation	Post release testing
Evaluating existing systems	Observations		Usability testing	
	Focus Groups			
	<b>Static prototyping</b>			

- **Цель:** Как можно раньше получить наглядное представление о том, как будет выглядеть и функционировать конечный продукт
- **Назначение:**
  - Раннее тестирование с пользователями: можно получить огромное количество информации о том, насколько наш дизайн соответствует запросам
  - Более точное определение трудоёмкости, планирование разработки: на ранней стадии выявить скрытые проблемы и заложить их решение в проект, раньше поменять набор функций, заранее изменить архитектуру интерфейса

# Usability исследования

Project Initiation	Analysis & Requirements	Architecture Design	Development & Test	Acceptance & Post Release
Context analysis	<b>Questionnaires</b>	Static prototyping	Rapid prototyping	<b>Interviews</b>
Competitor analysis	<b>Interviews</b>	Heuristic evaluation	Heuristic evaluation	Post release testing
Evaluating existing systems	<b>Observations</b>		Usability testing	
	<b>Focus Groups</b>			
	Static prototyping			

- Usability исследования** – набор usability методов, которые подразумевают сбор информации через опрос пользователей про их взгляды и мнения, такие как интервью, анкетирование и фокус группы.

# Usability-тестирование

Project Initiation	Analysis & Requirements	Architecture Design	Development & Test	Acceptance & Post Release
Context analysis	Questionnaires	Static prototyping	Rapid prototyping	User surveys
Competitor analysis	Interviews	Heuristic evaluation	Heuristic evaluation	<b>Post release testing</b>
Evaluating existing systems	Observations		<b>Usability testing</b>	
	Focus Groups			
	Static prototyping			

- **Цель:**

Как можно раньше выявить узкие места системы и внести необходимые коррективы

# Экспертная оценка usability ПО, usability-консалтинг

Project Initiation	Analysis & Requirements	Architecture Design	Development & Test	Acceptance & Post Release
Context analysis	Questionnaires	Static prototyping	Rapid prototyping	Interviews
Competitor analysis	Interviews	<b>Heuristic evaluation</b>	<b>Heuristic evaluation</b>	Post release testing
Evaluating existing systems	Observations		Usability testing	
	Focus Groups			
	Static prototyping			

- **Цель:**

Поиск и анализ потенциальных и реальных проблем в GUI системы

# Этапы процесса проектирования UI

## Анализ

- Анализ бизнес-процессов и деятельности пользователей
- Формулировка требований, выбор показателей оценки пользовательского интерфейса

## Проектирование

- Формализация информации в виде диаграмм бизнес-процессов и сценариев
- Построение пользовательской модели данных, привязка объектов к ролям и формирование рабочих мест
- Выработка концепции
- Прототипирование интерфейса

## Реализация

- Уточнение прототипов
- Имплементация UI в коде, создание тестовой версии
- Тестирование с пользователями

## Внедрение

- Получение обратной связи от пользователей и ее анализ

# Проектные роли, связанные с Usability



- **Проектировщик интерфейсов (UI-Designer)**
- **Дизайнер (Painter)**
- **Usability-тестировщик (Usability-tester)**
  - Умение анализировать факторы, влияющие на удовлетворённость пользователей
  - Навык написания сценариев использования
  - Глубокое знание принципов работы, взаимной совместимости и назначения возможно большего числа типов интерфейсных элементов
  - Умение проводить качественно и быстро usability-тестирование
  - Представление о проблематике пользовательских интерфейсов
  - Исключительная коммуникабельность, тщательность и педантичность. Развитая эмпатия (сопереживание собеседнику), честность, непредвзятость



# Сбор требований UI в рамках обеспечения эффективности

- **Какие вопросы следует осветить:**

- ✓ *Какая информация необходима пользователю для решения задачи?*
- ✓ *Какую информацию пользователь может игнорировать (не учитывать)?*
- ✓ *Совместно с пользователем разделить всю информацию на сигнальную, отображаемую, редактируемую, поисковую и результирующую.*
- ✓ *Какие решения пользователю необходимо принимать в процессе работы с программой?*
- ✓ *Может ли пользователь совершать несколько различных действий (решать несколько задач) одновременно?*
- ✓ *Какие типовые операции использует пользователь при решении задачи?*
- ✓ *Что произойдет, если пользователь будет действовать не по предписанному Вами алгоритму, пропуская те или иные шаги или обходя их?*

- **Необходимо понимать модель данных**

- Тщательно продумать и осознать сценарий взаимодействия программы с пользователем, приведя его к оптимальному относительно рассмотренных показателей
- Реализовать UI в соответствии с этой системой

# Сбор требований UI в рамках обеспечения производительности

- Для оценки продуктивности используются соответствующие показатели, проверяемые специалистами по эргономике в процессе usability тестирования рабочего прототипа
- Формирование таких показателей происходит в процессе определения требований к UI при изучении следующих вопросов:
  - ✓ *Что от пользователя требуется в первую очередь?*
  - ✓ *Сколько информации, требующей обработки, поступает пользователю за период времени?*
  - ✓ *Каковы требования к точности и скорости ввода информации?*
  - ✓ *На какие операции пользователь тратит больше всего времени?*
  - ✓ *Чем мы можем облегчить работу пользователя при решении типовых задач?*

# Создание документа GUI Specification



**GUI Specification** - это материализация проекта программного продукта в документальной форме. Она описывает и показывает действия пользователей, а также вид и поведение ПО в специфических ситуациях

## Цель документа:

- Описать основные принципы построения интерфейсов, руководствуясь принципами usability
  - Документом необходимо руководствоваться разработчикам UI для соблюдения целостности и удобства разрабатываемого приложения
  - Документ может постоянно модифицироваться из-за корректировок заказчика и согласования дизайнерских решений

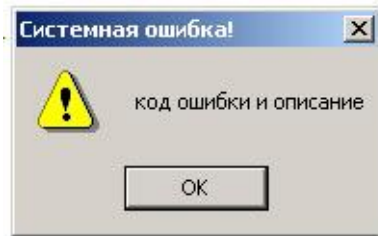
# Структура документа GUI Specification (1/4)

1. Описание способов перехода по элементам формы
2. Описание формата вывода логических и системных ошибок

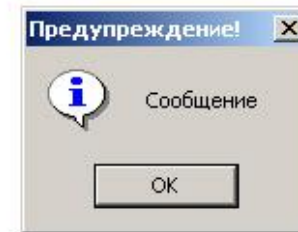
*Пример решения:*

Все тексты ошибок (системных, логических) должны быть вынесены в отдельный ресурсный файл и иметь общий стиль оформления

**Вид системной ошибки:**



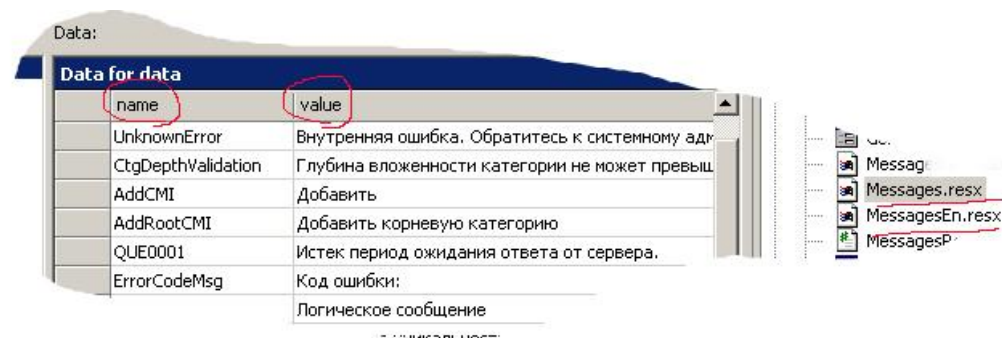
**Вид логической ошибки:**



3. Описание текста ошибок

*Пример решения:*

Тексты системных и логических ошибок находятся в ресурсных файлах

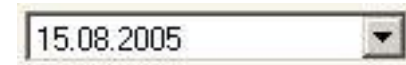


# Структура документа GUI Specification (2/4)

## 4. Описание формата полей

Примеры решения:

a) Описание полей формата даты:



15.08.2005

Представление поля в зависимости от локали;  
Представление поля, если оно необязательное

b) Представление обязательных для ввода полей:

\* Ответ на вопрос:



c) Описание реакции системы при вводе максимально допустимого значения в поле:

Примеры решения: дальнейший ввод блокируется, видимая длина поля определяется общим дизайном конкретного экрана

d) Нередактируемые поля

Например, цвет фона таких полей:

RGB: 240;232;211



ID

e) Вид, цвет и размер полей, меток, кнопок

width: 20 px

color: white

BorderStyle: FixedSimple



\* Название (Рус)



f) DataGrid: подсветка выделенных записей, возможность выделения нескольких записей, изменение размеров колонок при изменении размеров экрана

# Структура документа GUI Specification (3/4)

## 5. Начальное позиционирование:

Описание положения указателя при вызове модального окна и возврате в родительское

## 6. Способ генерации ID

Например: описать, что ID формируется уникальным целым числом без нулей перед самим значением

## 7. Цветовое и шрифтовое решения, общий стиль оформления оконных интерфейсов

Фон всех окон: RGB 248, 244, 233

Верхняя титульная строка:

RGB 240, 232, 211/width:24px

ButtonPanel: RGB 240, 232, 211/Size 40px

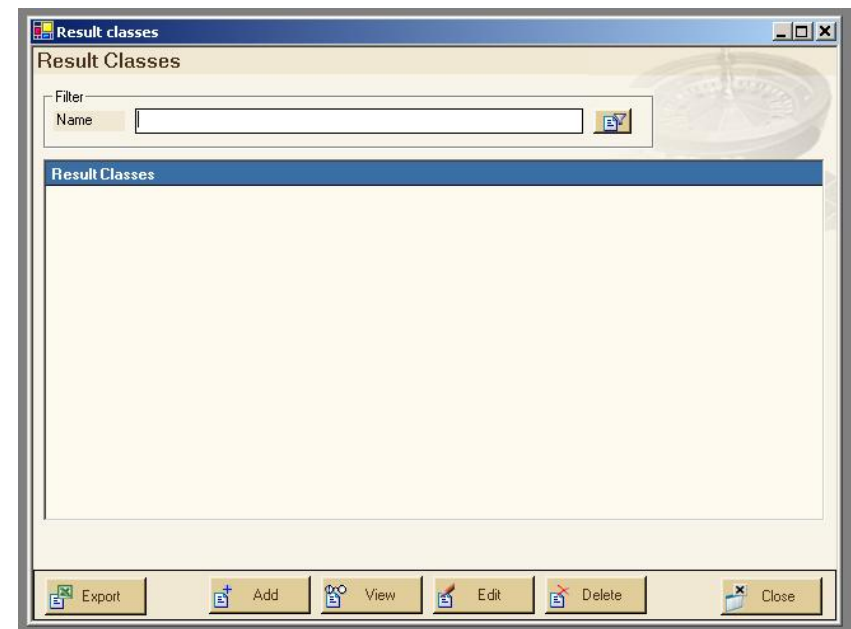
Шрифт: MS San Serif

поля: 8.25 цвет: black

Заголовки: 13, цвет: RGB 71, 47, 23

Иконка на подложке:

рулетка – временно, в последующем там будет логотип компании заказчика

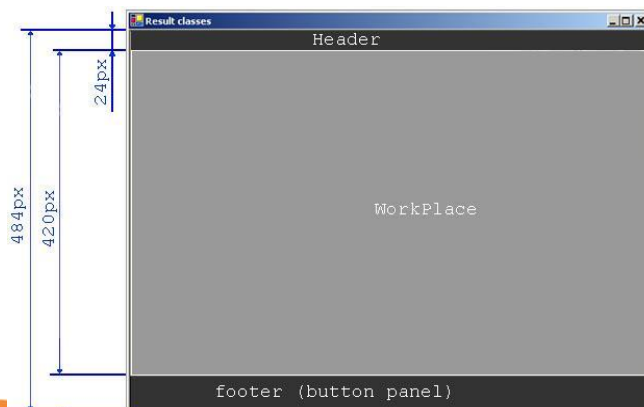
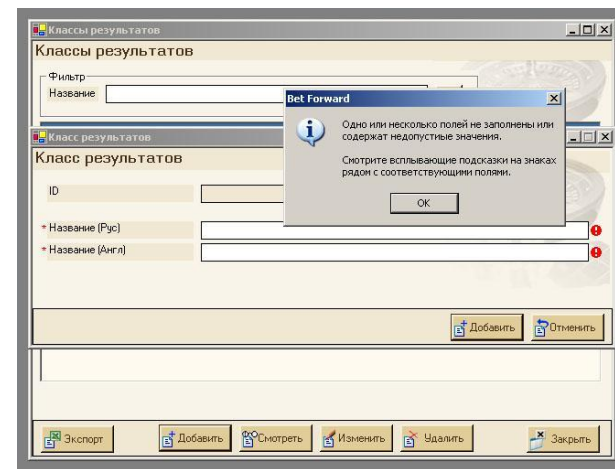


# Структура документа GUI Specification (4/4)

## 8. Отступы между элементами формы и размеры рабочей области окон

*Пример решения:* Левая и правая граница рабочей области имеют отступ 13px; Вертикальный отступ между элементами: 5 px; Кнопки привязаны к правому краю и масштабируются в этой же привязке, центрированы по высоте ButtonPanel

## 9. Обязательные поля и вывод предупреждений о некорректно введённой информации (не заполнены обязательные поля, введены недопустимые значения)



## 10. Общая структура оконных интерфейсов, размеры шрифтов

# Практическое занятие (1)



## По созданию документа GUI Specification

### Задание:

#### Выдать слушателям:

1. В качестве образца структуры документа - распечатку документа «GUISpecification\_4TST019.doc»
2. Распечатку скриншота реализации одной из основных форм Системы – «Список групп» - документ «СписокГрупп.doc»
3. Фрагменты документов 150-04, 260-04, описывающих спецификацию и функциональность элементов управления формы «Список групп»:
  - а) Документ «150-04-(Список Групп).doc»
  - б) Документ «260-04-(СписокГрупп).doc»



# Выполнение практического задания (исходные данные)

## Описание:

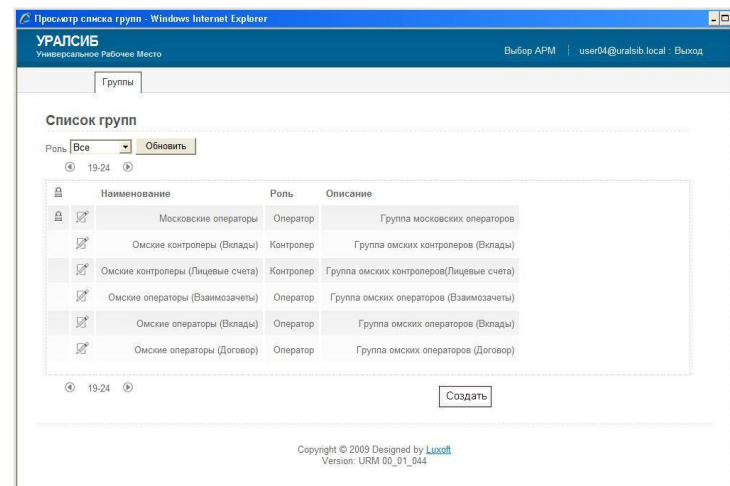
- ❑ Ознакомиться с содержанием документов 150-04, 260-04
- ❑ Рассмотреть скриншот реализованной формы «Список групп»
- ❑ Приступить к выполнению задания (см. след. слайд)

## **Примечание:**

- *Доступ к списку групп есть только у пользователя с правами «Администратор безопасности».*
- *Если вдруг в процессе выполнения операции работы с группой/ списком групп **пользователю изменили права** – он перестал быть Администратором безопасности, то Система должна выдать сообщение*

*(см. п. 1.3.4. Альтернативные направления документа «150-04-(Список Групп).doc»)*

*Слушателям нужно учесть этот факт , придумать и описать вид и формат данного сообщения*



Скриншот веб-интерфейса «Список групп» в браузере Internet Explorer. В заголовке страницы «УРАЛСИБ» и «Универсальное Рабочее Место». В строке меню «Выбор АРМ» и «user04@uralsib.local - Выход». В центре экрана таблица «Список групп» с колонками «Наименование», «Роль» и «Описание». В строке «Роль» выбрано «Все» и кнопка «Обновить». В строке «19:24» и «19:24».

Наименование	Роль	Описание
Московские операторы	Оператор	Группа московских операторов
Омские контролеры (Вклады)	Контролер	Группа омских контролеров (Вклады)
Омские контролеры (Лицевые счета)	Контролер	Группа омских контролеров(Лицевые счета)
Омские операторы (Взаимозачеты)	Оператор	Группа омских операторов (Взаимозачеты)
Омские операторы (Вклады)	Оператор	Группа омских операторов (Вклады)
Омские операторы (Договор)	Оператор	Группа омских операторов (Договор)

Создать

Copyright © 2009 Designed by Luxoft  
Version: URM\_00\_01\_044

# Выполнение практического задания



## Слушатели должны:

1. Попытаться создать структуру документа GUI Specification в части описания интерфейса для формы «Список групп», основываясь на входных материалах для задания
2. По возможности набросать разделы документа, не требующих уточнения
3. Составить список вопросов, касающихся описания графического интерфейса, требующих уточнения

Время выполнения задания - 20 минут

Время обсуждения выполненного задания – 10 минут

## 5. Проектирование UI

- Стандарты фирмы IBM о принципах Usability
- Метафора. Концептуальная модель
- Правила дизайна форм
- Принципы Usability в построении страниц Web приложения
- Решение проблем проектирования интерфейса для МУ

Практическое задание по разработке метафоры для реализации UI

Практическое задание по созданию прототипа GUI Системы методом бумажного прототипа (приложение Web)

Практическое задание по созданию прототипа GUI Системы методом бумажного прототипа (приложение Windows)

# Стандарты фирмы IBM о принципах Usability



Разработка UI состоит из проектирования **панелей** и **диалога**. Панель приложения разделена на три части: **место действий**, **тело панели**, **область функционирования клавиш**:

- Использование **меню действий** (выпадающего меню) – наглядно, эти действия могут быть запрошены пользователем установкой курсора, вводом. Многоуровневая система выпадающих меню (оптимальное число уровней - три)
- **Тело панели** содержит элементы тела панели: разделители областей, идентификатор и заголовок панели, инструкцию, заголовки столбца, группы, поля; указатель протяжки; область сообщений и команд; поля ввода и выбора
- **Область функциональных клавиш** – необязательная часть, показывающая соответствие клавиш и действий, которые выполняются при их нажатии
- Работа на панели основано на принципе **«объект-действие»**. Пользователь выбирает объект, затем производит действия с этим объектом, что минимизирует число режимов, упрощает и ускоряет обучение работе с приложениями и создает для пользователя комфорт
- **Диалог** ведется в первичном окне, затем продолжается во вторичном. «Всплывающие окна» призваны улучшить диалог. Диалог состоит из запросов на обработку информации и навигации по приложению. Унифицированные действия диалога: "отказ", "команда", "ввод", "выход", "подсказка", "применить", "извлечение", "регенерация"

# Стандарты фирмы IBM.

## Элементы экрана



Минимальные единицы панели называются *элементами* тела панели:

- **Разделители.** В качестве разделителей могут использоваться цветные границы, линии, простые строки или столбцы, заголовки столбцов
- **Заголовок панели** сообщает пользователю о том, какая информация содержится в теле панели. Если другие области тела панели должны протягиваться, то заголовок образует самостоятельную область и не протягивается
- **Инструкция** сообщает пользователю, что нужно сделать и как продолжить работу
- **Поле выбора.** Когда пользователь выбрал объект, приложение отмечает это при помощи цвета, подсветки или символа, размещаемого перед выбранным объектом. Цвет и подсветка называются *выделением*, а символ - *указателем выбора*. Различают поля *однозначного, многозначного и расширенного выбора*
- **Курсор выбора** может быть в виде контура, линейки, подчеркивания, изменения цвета
- **Указатель протяжки** используется в тех случаях, когда выполняется скроллинг, обозначается стрелками, специальной линейкой или текстовыми указателями
- **Поле ввода, заголовок поля и др.**

# Стандарты фирмы IBM. Рекомендуемая палитра



- При первоначальном отображении панели каждый элемент должен иметь свои яркость и цвет
- По мере углубления диалога для показа текущего состояния объекта, с которым работает пользователь, цвета и эффекты выделения могут изменяться
- Рекомендуется:
  - Панель в первичных и вторичных окнах, за исключение панели «Справка» - белая
  - Панель в окне «Справка» – синяя
  - Панель во всплывающих окнах нечетного уровня – голубая, а четного уровня – белая
  - Ошибки выделяются красным
  - Предупреждения об ошибках – желтые
  - Критические сообщения – красные

# Метафора как отправная точка проектирование UI

- Придумать метафору для реализуемого UI
  - Обстановка на экране и способы взаимодействия должны апеллировать к ситуации, хорошо знакомой пользователю.
  - **Метафорические интерфейсы** полагаются на интуитивные связи, которые пользователь устанавливает между визуальными элементами интерфейса и его функциональностью.
  - Пользователи воспринимают картинку и предположительно могут расширить ее до понимания предназначения объекта.



*Например,* оконный интерфейс задумывался как метафора рабочего стола с документами

# Преимущества использования метафоры

*Преимущества* такого подхода:

- Пользователю легче понимать и интерпретировать изображение на экране
- Ему не нужно каждый раз заглядывать в руководство, чтобы узнать, как выполняется то или иное действие. По крайней мере некоторые действия должны "естественно" следовать из метафоры
- У пользователя возникает чувство психологического комфорта, характерного для встречи с чем-то хорошо знакомым



# Проблемы, возникающие при использовании метафоры



- **Проблемы** такого подхода:
  - Метафоры плохо "масштабируются". Метафора, хорошо работающая для простого случая в простой программе часто перестает работать, как только задача усложняется
  - Метафоры не ответственны за то, как их понимают. Иногда метафору поймут интуитивно, иногда нет
  - Метафору приходится "подправлять", т.к. взаимодействия проходят не в реальном мире, а с помощью искусственных приспособлений - экран, мышь и клавиатура
  - Возможности мира внутри компьютера обычно шире возможностей физического мира, и это может с успехом использоваться для более мощного интерфейса . Поэтому с **осторожностью** используйте метафору для интерфейса в целом, т.к. **это ограничивает возможности наших программ!**

# Неудачные примеры использования метафоры

Практически каждый аспект интерфейса MagicCar (интерфейса портативного коммуникатора середины 90-х) опирается на метафоры.



Все элементы, используемые пользователем, расположены на столе и имеют внешний вид настольных аксессуаров, документов и папок.

Чтобы запустить пасьянс, вы проходите по коридору до дверей игровой комнаты.

Вы входите в здание, чтобы запустить приложение и видите коридор с дверями, обозначающими функции. Интуитивно понятно назначение всех функций, но чтобы запустить другое приложение, вы должны вернуться на улицу.



Минус такого подхода – как только пользователь освоился с программой, **метафора начинает мешать навигации.**

Дополнительный минус - MagicCar была **медленной**, устанавливалась только на устройства с большим объемом памяти и по цене была выше аналогов

# Неудачные примеры использования метафоры

## *Портфель в Windows OS*

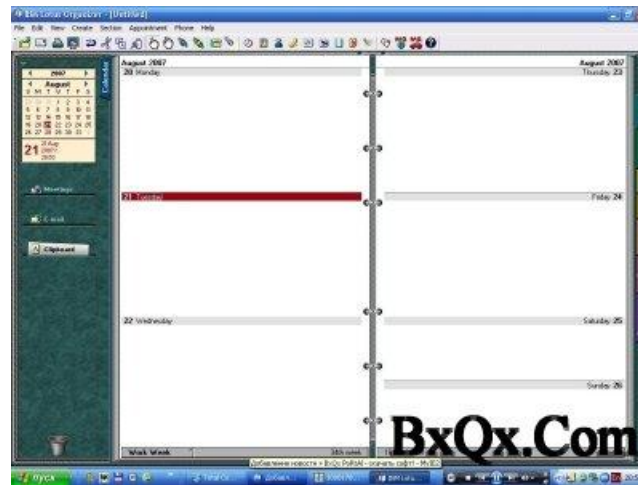
Предполагалось, что это тот самый портфель, в который вы складываете рабочие документы, чтобы поработать с ними дома. Изменения сделанные дома вы принесете обратно, и они синхронизируются с имеющимися версиями файлов.

В современных версиях Windows OS возможна синхронизация по сети, но в Windows 98 файлы все равно приходилось копировать на дискету - Портфель не получил признания пользователей, а метафора выглядела несколько странной.



# Пример удачной метафоры UI

- В качестве примера удачной метафоры в интерфейсе можно привести Lotus Organizer, внешний вид которого напоминает привычный еженедельник, функции которого и выполняет этот продукт:



## Рекомендация:

- Используйте метафору, если найдете, но не делайте свой интерфейс "заточенным" под какую-нибудь стандартную метафору

- **Используйте не метафору, а идиомы**

Кнопки, выпадающие списки и полосы прокрутки - это то, что пользователь узнает автоматически. При этом они не являются метафорами, т.к. не существует их соответствия с объектами реального мира.

- Для освоения идиомы, как правило, достаточно однажды ее изучить



*Например,* окно с изменяющимся размером. Ему нет аналога в реальном мире, но мы его легко узнаем и легко с ним обращаемся

# Практическое занятие (2)

## По разработке метафоры для реализации UI Системы

### Задание:

Цель: Пробудить у слушателей творческий, неординарный подход к процессу проектирования пользовательского интерфейса

### Выдать слушателям:

1. Краткое описание концепции Системы с кратким описанием функциональности одной из компонент Системы (см. след.слайд)

### Слушатели должны:

1. Разработать метафору для реализации пользовательского интерфейса
2. Попытаться объяснить, почему выбрана именно такая метафора
3. Продумать, какие элементы управления могут быть использованы в рамках придуманной метафоры

Время выполнения задания - 10 минут,

Время обсуждения выполненного задания – 10 минут

# Выполнение практического задания



Предлагается придумать метафору, которая «ляжет» в основу интерфейса следующих систем:

1. Приложения, работающего с расписанием самолетов, выбор рейса (время отправления, прибытия). Справка: просмотр условий перелета и провоза багажа
2. Системы для воспроизведения музыкальных и видео файлов
3. Файловая система
4. Системы для ввода карточки пациента и печати ее на принтер
5. Системы коммуникаций посредством сети Intranet (почта, чат, видео чат, телефонные переговоры)

# Разработка концептуального дизайна интерфейса

## Что такое концептуальный дизайн интерфейса?

- Разработать в рамках придуманной метафоры систему интерфейсных элементов, своего рода алфавит взаимодействия, изучив который пользователь сможет легко делать то, что ему нужно
- Найти изящный способ изображения как отдельных элементов так и их групп
- Выбрать общий изобразительный стиль, который был бы легко узнаваем и приятен для глаз

## Какие вопросы должна решить разработка концепции?

- Давать представление об интерфейсе, давать возможность валидации (согласования с заказчиком)
- Позволять оценивать реализуемость концепции
- Выделить основные типы экранов (ключевые)
- Определить паттерн в интерфейсе - фрагмент интерфейса, который будет использоваться на нескольких ключевых экранах
- Создать модель навигации по типам экранов
- Создать пользовательскую модель данных



**Прототипирование** – один из методов проектирования пользовательского интерфейса

**Прототип** – это материализация построенного проекта с использованием его предполагаемой платформы реализации, включая оборудование, ОС, языки и средства реализации

## Требования к прототипу:

- Эстетичность и привлекательность прототипа
- Максимально дешев
- Минимальные временные затраты
- Максимально проработан
- Лёгкость модификации и обновления

## Цель:

- Как можно раньше получить наглядное представление о том, как будет выглядеть и функционировать конечный продукт
- Получить feedback от Заказчика
- ✓ На ранних стадиях проекта очень эффективно использовать метод тестирования с помощью *бумажных прототипов*. В 100 раз дешевле внести изменения до того, как код был написан, чем после того, как он уже готов

# Методы прототипирования

- В зависимости от временных рамок, проектных вопросов, квалификации специалистов, а также имеющихся в распоряжении средств и методов представления проекта может быть *раскадровка* или *макет на бумаге*
- Различные типы моделей и прототипов ориентированы на различные вопросы глубины и широты представления возможностей UI или системы. Различают следующие методы прототипирования:
  - ✓ Беглое (rapid) на бумаге
  - ✓ Повторное использование прототипов (reusable)
  - ✓ Функциональное (functional)
  - ✓ Модульное (modular)
  - ✓ Горизонтальное (horizontal) – прорабатываются все пункты меню – раскрывающиеся экраны
  - ✓ Вертикальное (vertical) - прорабатывается один пункт меню до самого низа
  - ✓ Прототипирование с низкой точностью
  - ✓ Прототипирование с высокой точностью

# Прототипирование. Разработка Templates (проволочных схем)

- Указывать сразу все известные точные данные на Template. Чем точнее описан функционал страницы, тем точнее он может быть сделан
- Хорошо сделанный Template должен задавать информационную иерархию элементов на странице, выделять важные и скрывать ненужные
- Различимость всех элементов залог того, что дизайнер изобразит их на эскизе, а разработчик внесет в код
- Аккуратность выполнения Template
- Помимо самого графического изображения Template требуется также четко описывать пояснение к каждому пункту
- Несмотря на точность и четкость Template должен оставлять свободу выбора. Например, выпадающие списки, радио-кнопки и пр. элементы интерфейса изображать похожими, но не реальными Windows-элементами



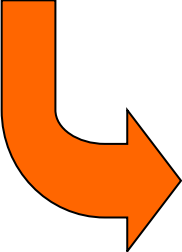
# Пример проектирования и прототипирования GUI

Типы событий

Название  Обновить

ID	Название	

Экспорт в Excel



Название экрана

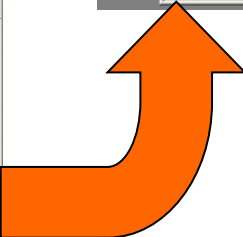
Название экрана

Фильтр  
Название:

Название датагрида

Экспорт в

Статус



Категории

Категория

ID

Название (Рус)

Название (Англ)

Сокращение (Рус)

Сокращение (Англ)

Дата начала

Дата окончания

Страна

Категория-родитель

Корневая категория

Ключевые слова

Архив

Добавить Отменить

- **Этапы проработки пользовательского интерфейса:**  
от общего к деталям  
отдельным формам

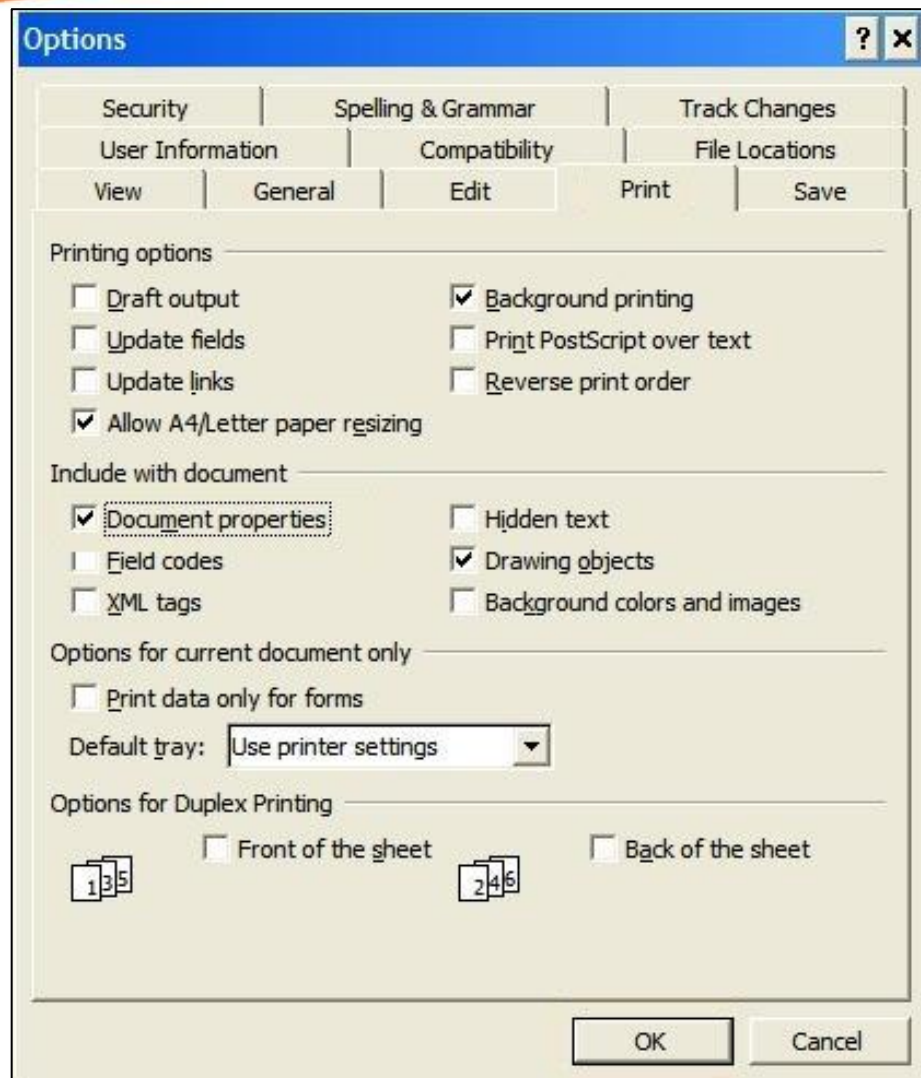
## ■ Принцип 1 (не только о дизайне форм)

Необходимо **учитывать язык GUI:**

- Подчеркнутое можно кликнуть
- Серое сейчас неважно и недоступно
- ...

Учитывайте язык GUI – те идиомы, которые есть у пользователя в опыте от предыдущего общения с компьютерными программами.

# Правила дизайна форм (2/20)



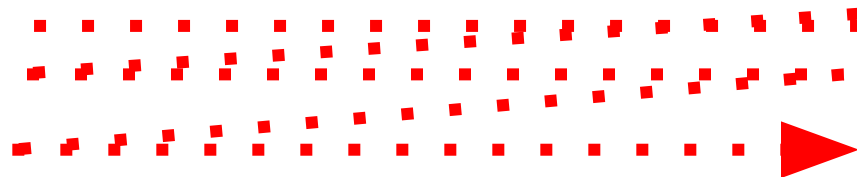
Окно Параметры в Word - пример злоупотребления идиомой диалоговых окон с вкладками.

Проблема в том, что пользователю приходится проделывать большую работу, чтобы найти тот единственный параметр, который требуется изменить.

У всех идиом есть практические ограничения.

## ■ Принцип 2

- Данные в форме должны располагаться в **наиболее вероятном порядке** их ввода, а также чтения
- Расположение элементов должно быть **согласовано с задачами** пользователя - не заставляйте пользователя перепрыгивать из раздела в раздел
- Этот порядок определяется нашими привычками к чтению – **слева направо, сверху вниз:**



# Правила дизайна форм (4/20)

Search for Group Policy Objects

1 Search for GPOs in this domain: microsoft.com

Search for Group Policy objects (GPOs) that meet all of these criteria:

Object	Condition	Value
GPO Name	Contains	Default
Computer Configuration	Contains	Security

5 Find now

Stop

New search

Search this object: [ ]

2 Condition: [ ] 3 Add

Value: [ ] Remove

Search results:

GPO	Domain	GUID
Default Domain Controller Policy	microsoft.com	{6AC1786C-016F-11D2-945F-...}
Default Domain Policy	microsoft.com	{31B2F340-016D-11D2-945F-...}

6 Save results 7

Found 2 items

Search for Group Policy Objects

Search for GPOs in this domain: microsoft.com

Add search criteria to the list below:

Search this object: [ ]

2 Condition: [ ] 3 Add

Value: [ ]

All search criteria:

Object	Condition	Value	Remove
GPO Name	Contains	Default	5 Search
Computer Configuration	Contains	Security	5

Stop search

Search results:

GPO	Domain	GUID
Default Domain Controllers Policy	microsoft.com	{6AC1786C-016F-11D2-945F-...}
Default Domain Policy	microsoft.com	{31B2F340-016D-11D2-945F-...}

6 Save results 7

Clear Close

Found 2 items



## ■ Принцип 3

- При вводе данных в форму ввода: если пользователь собирается ввести в базу данных 10000 записей, вероятно, он **не хочет подтверждать ввод каждой записи**
- В форме ввода данных необходимо **максимально использовать свободное пространство**, поскольку открытие и закрытие дополнительных форм существенно замедляет работу
- Если возможно, выполняйте **добавление и редактирование записей в одной и той же форме**, тогда пользователю не придется осваивать несколько методов доступа к одним и тем же данным
- Всегда назначайте **клавиатурные эквиваленты команд**; **не требуйте обязательного использования мыши** – это ускоряет процесс ввода данных

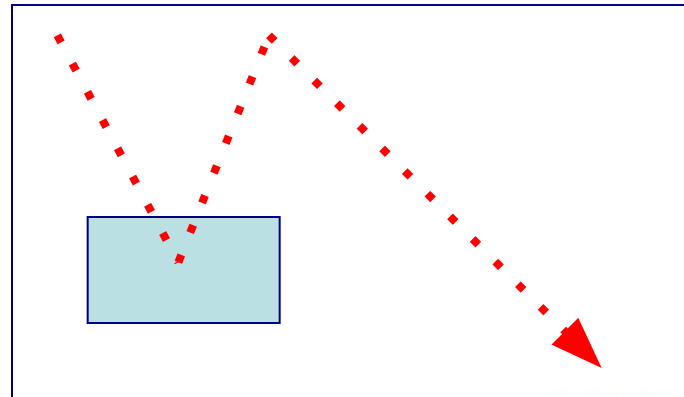
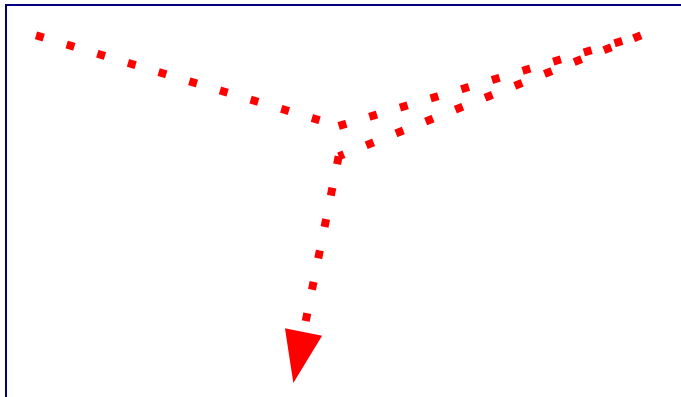
## ■ Принцип 4

Форма должна быстро сканироваться **взглядом**:

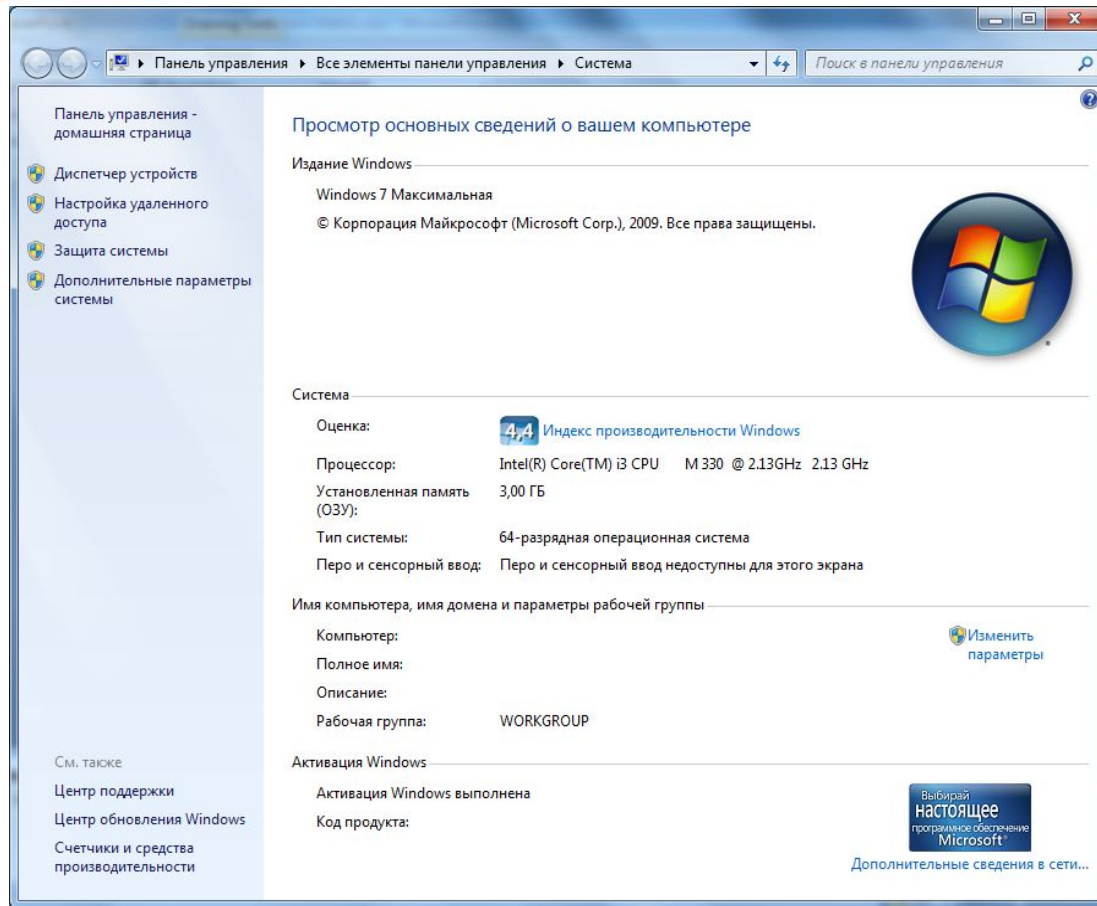
- Достаточное количество пустого пространства
- Линия просмотра

## ■ Принцип 5

**Линию просмотра** можно изменить введением новых центров масс



# Правила дизайна форм (7/20)

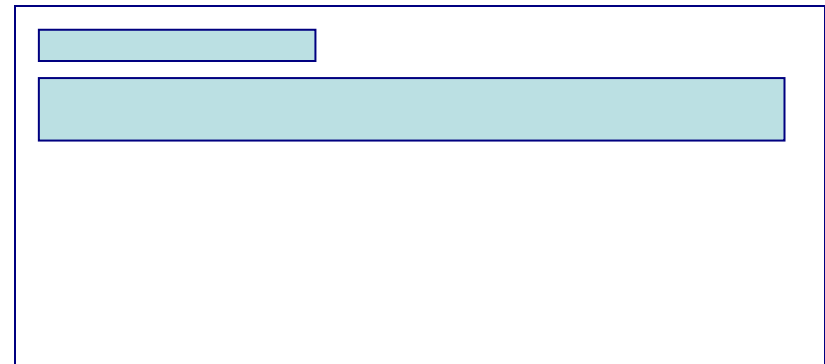


Пример формы, которая хорошо сканируется ВЗГЛЯДОМ.

Используется комбинация пустого пространства и разделителей.

## ■ Принцип 6

- Самое частотное/важное должно **находиться в начале формы, быть на виду, легко доступными**, а менее важная - прятаться где-нибудь в меню
- Интерфейс программы должен быть построен вокруг объектов, с которыми манипулирует пользователь, и отражать состояние текущего объекта
- *Хороший пример – панели управления в Corel Draw 8.0. Они постоянно меняются в зависимости от того, с каким объектом в данный момент работает пользователь*

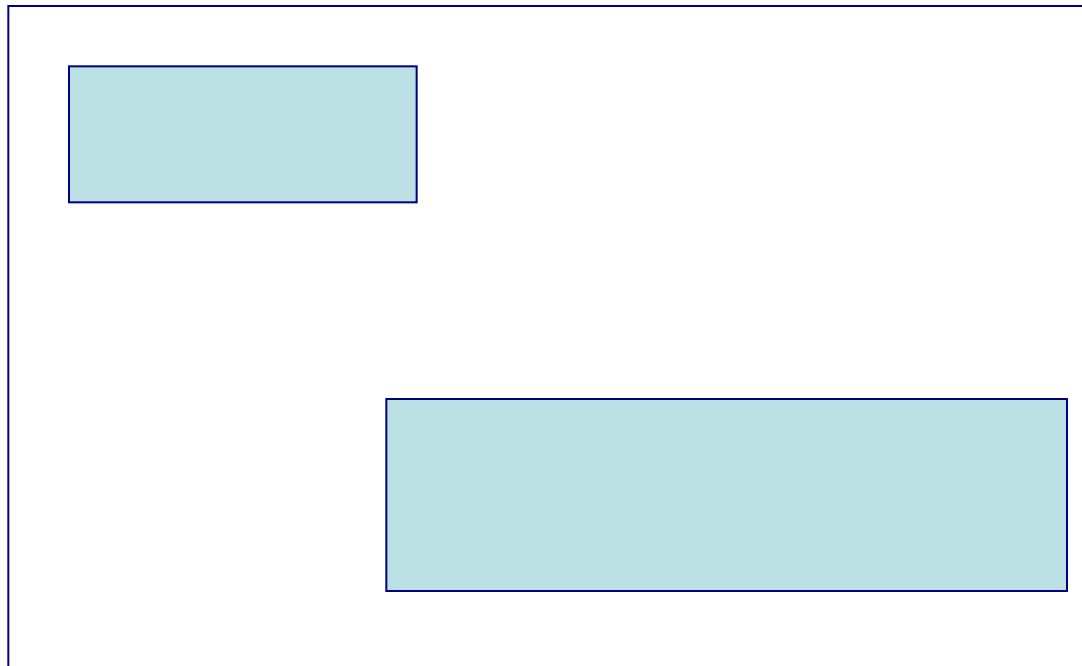


## ■ Принцип 7

- Не заставляйте пользователя **выполнять лишнюю работу**
- Другими словами, если информация, содержащаяся в полях со 2-го по 10-е, необходима только, когда первое поле имеет определенное значение, не нужно заставлять пользователя заполнять все поля подряд
- В то же время, не ставьте работу формы в зависимость от содержимого отдельных полей. В противном случае это может существенно замедлить работу пользователя

- **Принцип 8**

**Центры масс** формы должны соответствовать частотности/важности элементов в них. Самое важное должно находиться в центрах масс



## ■ Принцип 9

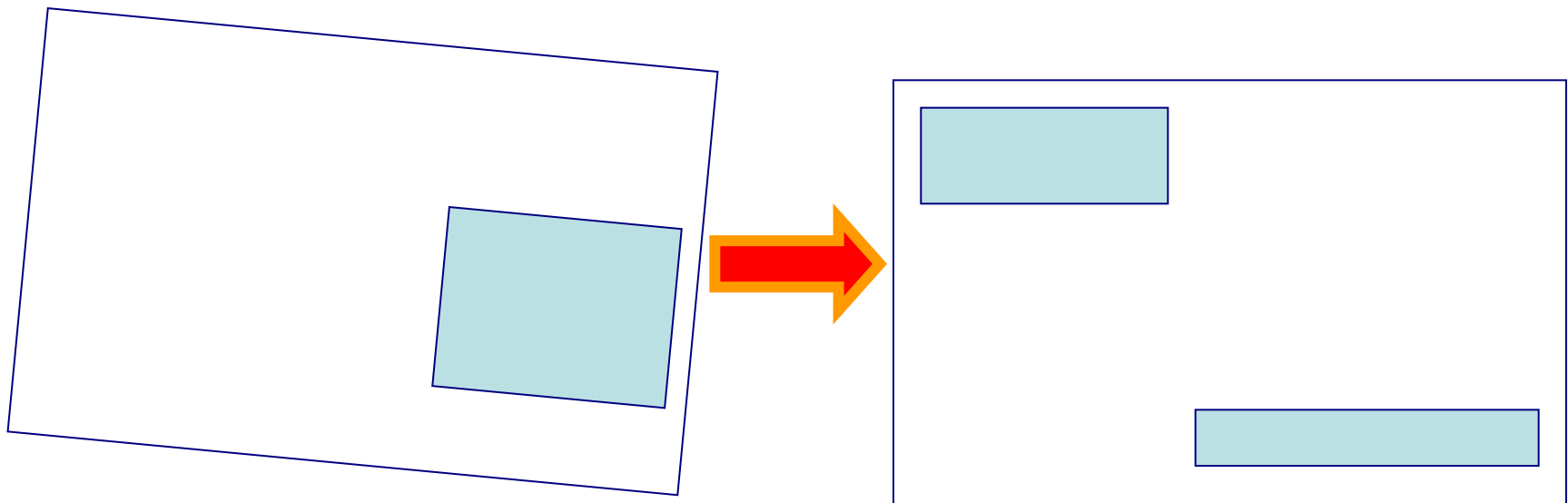
Принцип «Бритвы Оккама» или «KISS» (*Keep It Simple, Stupid*) или (*Keep It Short and Simple*)

"Не множить сущности без надобности". На языке интерфейсов это означает, что:

- любая задача должна решаться минимальным числом действий
- логика этих действий должна быть очевидной для пользователя
- движения курсора и даже глаз пользователя должны быть оптимизированы
- Например, метод **drag'n'drop** - перетаски-и-оставь – хорошая иллюстрация этого принципа. Это абсолютно естественное действие, выполняемое одним движением мыши, с великолепной оптимизацией движений курсора и глаз

- **Принцип 10**

Центры масс должны быть сбалансированы, иначе форма начнет заваливаться набок



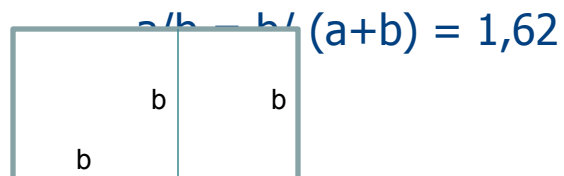


## ■ Принцип 11

### Принцип «Собора Василия Блаженного»



- Использование ограниченного числа основных цветов в программе - не более трех, исключая, конечно, оттенки. *Хороший пример - стильный дизайн кнопок в Netscape Communicator 4.0.*
- Золотое сечение, 1:1,62 - самая приятная для глаза пропорция, ею насквозь пронизаны чертежи собора. Применимо для дизайна интерфейсов

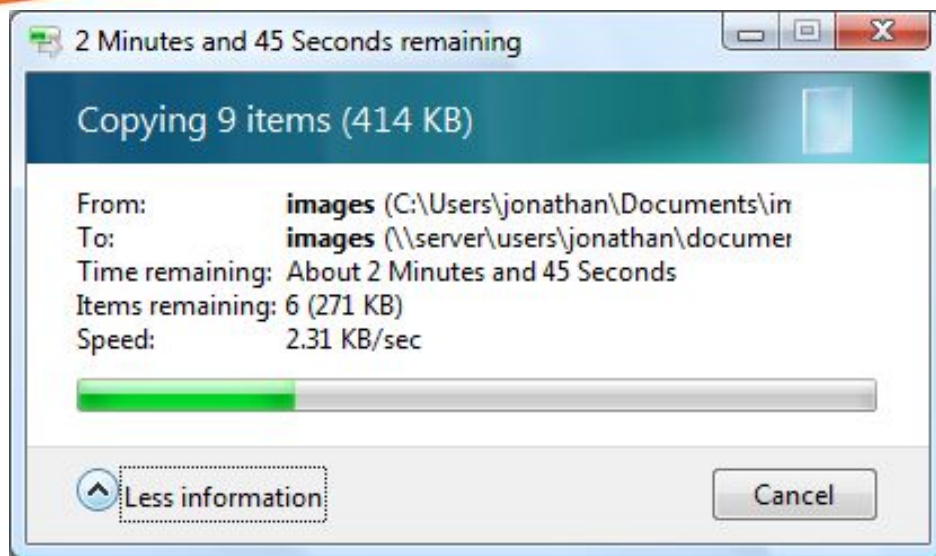


- **Принцип 12**

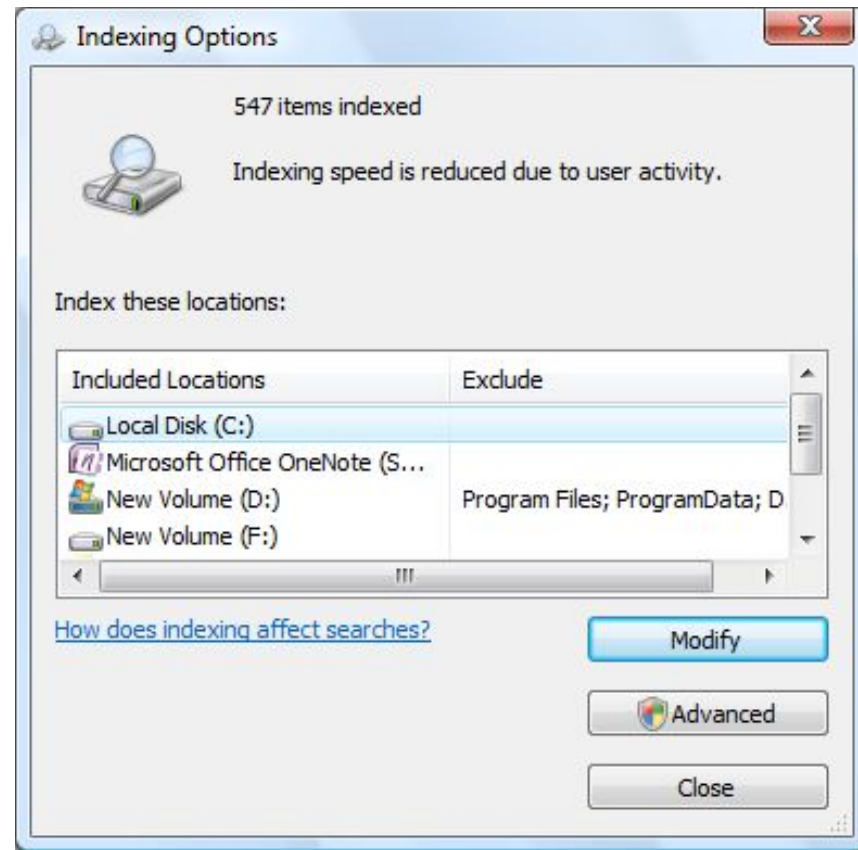
Размер полей должен быть не больше и не меньше максимального объема вводимых в них значений

Имя:	<input type="text"/>
Фамилия:	<input type="text"/>
Дата рождения:	<input type="text"/>

# Правила дизайна форм (15/20)



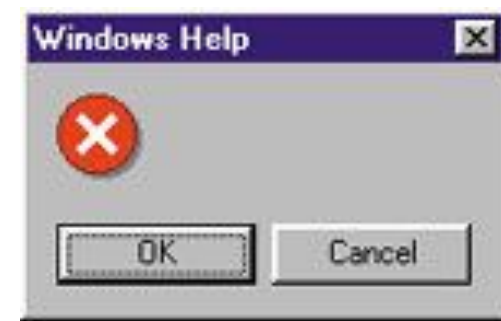
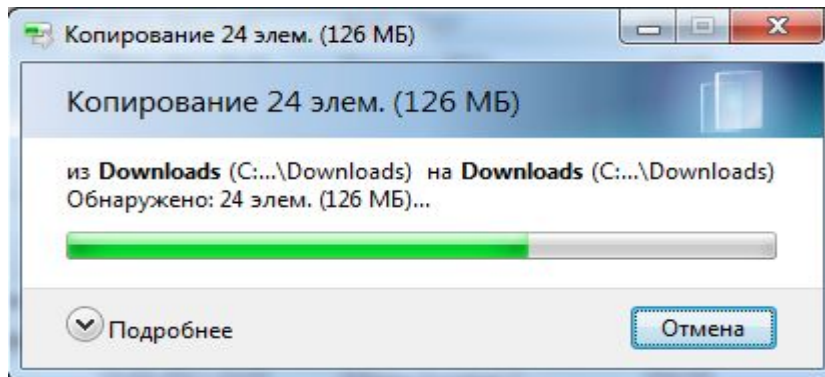
Пути источника и получателя обрезаны.



Размер окна не соответствует  
содержимому

## Принцип 13 - Обратная связь

- Пользователь должен всегда видеть, чем сейчас занимается программа (использование индикатора процесса) или к чему привело его действие.



- При длительных процессах используйте вывод графики, сопровождающие звуки, шум жесткого диска – это отвлекает пользователя и он не ощущает ожидания
- Используйте заметную, но ненавязчивую обратную связь с пользователем. *Хороший пример - работа редактора программного кода Visual Basic, который проверяет правильность написания переменных и констант*

## ■ Принцип 14

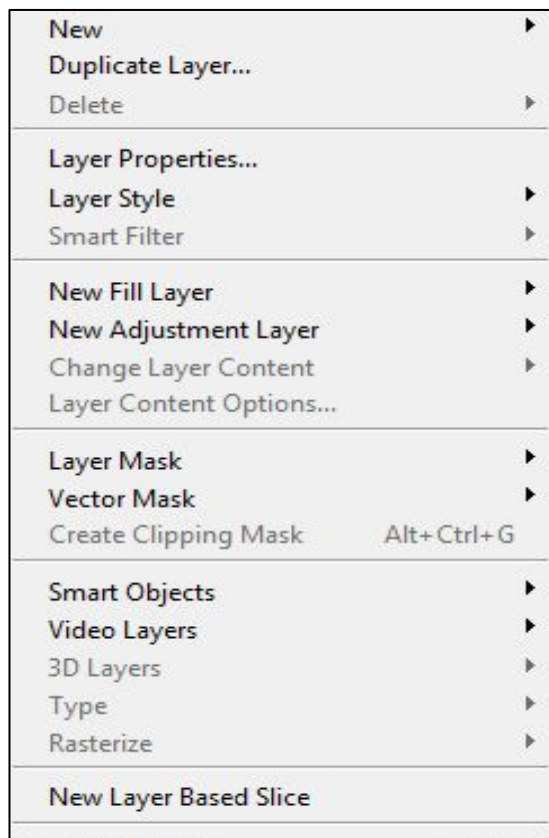
- Блоки полей должны разделяться либо рамками группировки, либо линиями, либо (лучше, но сложнее всего) отступами

- Количество полей в блоках – кошелек Миллера.

Кошелек Миллера - это наша с вами кратковременная память. За один раз в нее вмещается семь плюс/минус две сущности  $7 \pm 2 = \min 5 \max 9$

Соответственно, группировать сущности в программе (пункты меню, закладки, опции на этих закладках и т. п.) желательно с учетом этого правила - то есть не более семи в группе, в крайнем случае - девяти

# Правила дизайна форм (18/20)



Выпадающее меню Layer программы Adobe Photoshop.

Слева – с разделителями, справа – без.

- **Принцип 15**

Должна быть возможность прочтения формы вслух

Настройка схемы "Power4 Gear"

При питании:  от сети  от батарей

Отключение дисплея:

Отключение дисков:

---

Ждущий режим:

## ■ Принцип 16

Способы расположения названий и выравнивания полей:

Имя:	<input type="text"/>
Фамилия:	<input type="text"/>
Дата рождения:	<input type="text"/>

Имя:	<input type="text"/>
Фамилия:	<input type="text"/>
Дата рождения:	<input type="text"/>

Имя:	<input type="text"/>
Фамилия:	<input type="text"/>
Дата рождения:	<input type="text"/>



# Специфика Usability в построении страниц Web приложения



*«Your user is just one click away from your competitor»*

- Usability web сайтов и страниц web приложений тесно связано с проектированием навигации и поиска
- Поиск – один из самых востребованных интернет-сервисов
- Проектирование эффективного поиска – путь продвижения сайтов в поисковых машинах

# Принципы Usability в построении страниц Web приложения (1/8)

- **Принцип 1**

Сайт должен давать четкое представление о его цели и содержании

## **Рекомендации:**

Разместить на первой странице текст, содержащий главные ключевые слова, которым отвечает сайт (приложение), описывающие его цель

**Пример:** Если на главной странице написано только «Эффективный офис» и заголовки новостей «Завершен очередной проект для компании ХХХ», то посетитель вряд ли быстро поймет, чем это сайт может быть ему полезен, и куда идти дальше

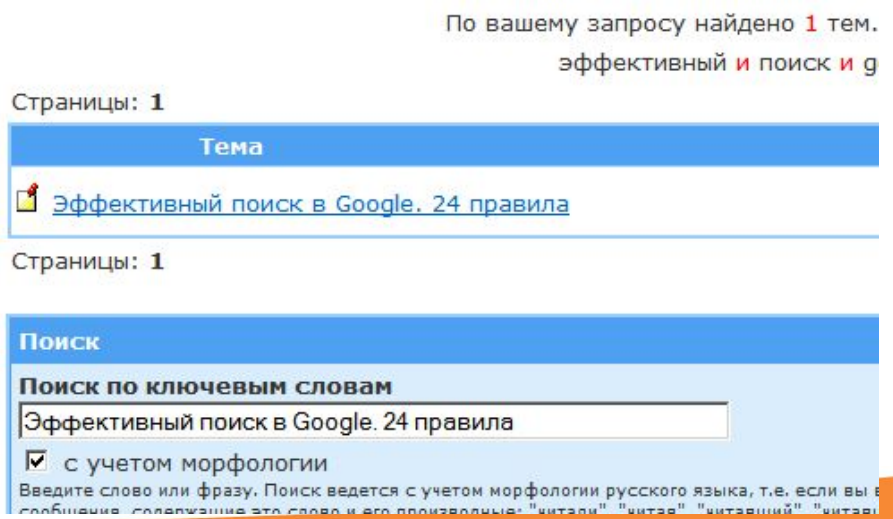
# Принципы Usability в построении страниц Web приложения (2/8)

## ■ Принцип 2

«Говорящие» титулы – титул должен отражать контент страницы и содержать ключевые слова

- В поиске – что в большом (поисковые машины), что в локальном (по сайту) титул расценивается как название страницы
- Не делать слишком длинные титулы с высокой концентрацией ключевых слов. Поисковые машины обрезают длину титулов до 70-80 символов титула. Достаточно 5-6 слов для описания содержимого страницы

*Пример: Если задать в запросе поиска содержимое титула страницы – поисковик выдаст в результате поиска эту страницу в числе первых*



# Принципы Usability в построении страниц Web приложения (3/8)

## ■ Принцип 3

Поддерживать адресную структуру сайта:

- Ссылки делают web удобной средой для поиска информации и перемещения от одного документа к другому
- Внешняя ссылка не должна вести на устаревший адрес – в лучшем случае посетитель попадет на страницу с сообщением об ошибке

## **Рекомендации:**

- Проверить актуальность ссылок
- При редизайне определить, куда ведут ссылки, перенаправить и людей, и поисковые машины на новый контент
- Стараться не менять адресацию страниц без насущной необходимости

# Принципы Usability в построении страниц Web приложения (4/8)

## ■ Принцип 4

Использовать альтернативный текст для иллюстраций:

- Когда пользователь отключает графику в своих браузерах, функциональность сайта не должна страдать – иллюстрации должны заменяться на текст
- Подписи к иллюстрациям также полезны для поисковых машин

## ■ Принцип 5

Чрезмерное использование Flash в навигации почти всегда является ошибкой и затрудняет работу системы:

Сайты, полностью сделанные на Flash:

- не могут ранжироваться поисковыми машинами так, как обычные HTML сайты
- имеют сложности при работе с закладками, выделении ссылок, увеличении размера текста, поиске на странице

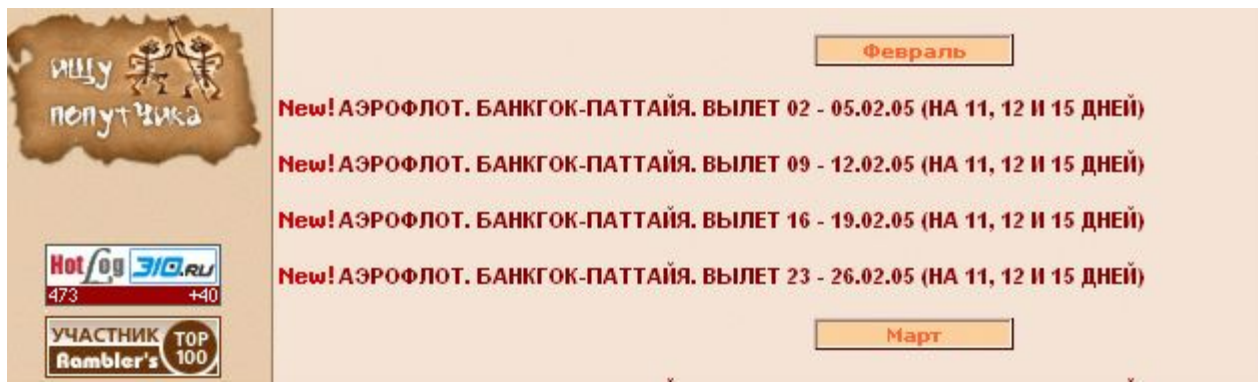
# Принципы Usability в построении страниц Web приложения (5/8)

## ■ Принцип 6

Основной контент должен быть доступен в HTML-формате:

- неудобно, когда для получения важной информации приходится запускать дополнительную программу, особенно, если она долго грузится. Еще хуже, если у вас этой программы нет совсем

*Пример, когда ссылки ведут на файлы в формате MS Excel*

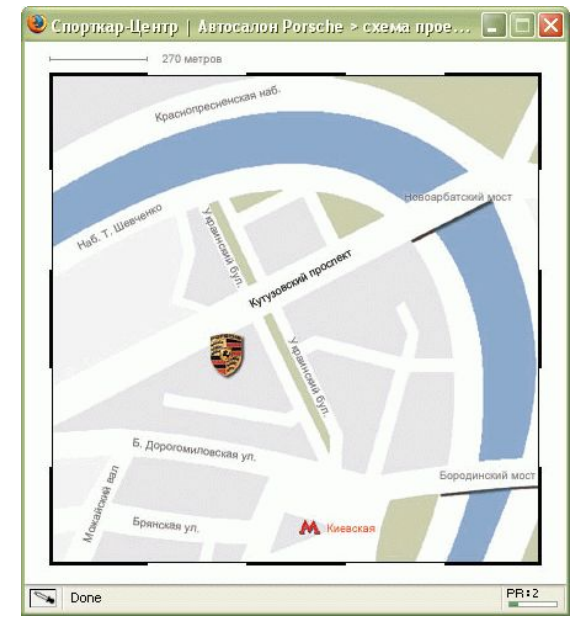


# Принципы Usability в построении страниц Web приложения (6/8)

## ■ Принцип 7

Мало на странице разместить рисунок, например, со схемой проезда - надо позаботиться, чтобы у пользователя была возможность распечатать ее в данном интерфейсе

*На приведенном примере нет ни кнопок, ни меню с пунктом «Печать»*



## ■ Принцип 8

■ Не использовать слишком длинные адреса страниц

**Пример:** Слишком длинный адрес разбивается почтовым клиентом на две части, и получаются письма с битыми ссылками, которые потом приходится склеивать, чтобы перейти по прямой ссылке. Это неудобно

■ Использовать «плоские» адреса, которые не бьются:

**Пример:** <http://www.rian.ru/investigations/20050203/17346581.html>

# Принципы Usability в построении страниц Web приложения (7/8)



## ■ Принцип 9

Не должно быть плохо размеченного и плохо структурированного текста, текст не должен состоять из длинных абзацев - его трудно воспринимать

### Рекомендации:

- Активнее, чем в обычном тексте использовать разбиение на разделы и абзацы
- Делать выделения различного вида и – что, разумеется, недоступно в печати – выделять гиперссылки (посещенные и не посещенные ссылки должны отличаться по своему оформлению)
- Ключевые для данной страницы словосочетания должны быть выделены шрифтом либо вынесены в заголовки – в Интернете внимательно читают редко, чаще сканируют страницу глазами

**Пример:** Мелкий подчеркнутый текст, большая ширина колонки, фиксированный размер шрифта, малый контраст между цветом шрифта и фоном затрудняет читаемость и работу Web-приложения

- Пути развития текстильной и легкой промышленности (Директор, февраль, 2004, Лебедев В.В., Фомченков Л.Н., Шамис И.А.)
- Службы призваны возродить торговые перевозки, чтобы преодолеть инертность в торговле
- ДОСТОИНСТВА ДВОЙСТВЕННОГО УСТАНОВЛЕНИЯ ЦЕН НА ПРИРОДНЫЙ ГАЗ РОССИИ (Дэвид Таро и Питер Томсон, Всемирный Банк)
- Пресс-релиз заседания директоров организаций ООН в Париже (25-26 апреля 2003 г.)
- Для укрепления многосторонней торговой системы важно устранить квоты ([www.wto.org](http://www.wto.org), 05.05.2003)
- Выступление Супачи Панитчакди на министерском заседании ОЭСР ([www.oecd.org](http://www.oecd.org), 07.05.2003)
- Присоединение России к ВТО: анализ прецедентов в алюминиевой промышленности. Автор: Олег Устенко
- Оценки социальных и экономических последствий присоединения России к ВТО
- Международно-правовые аспекты защиты интеллектуальной собственности ВТО и ВОИС
- Годовой Отчет Генерального Директора ВТО на английском языке (15 ноября 2002г.)
- Министерская декларация, принятая 14 ноября 2001 года в Дохе
- Отчет ВТО за 2002 год
- Исследование "Неоодолеваемые последствия присоединения России к ВТО"
- АНАЛИТИЧЕСКИЕ СТАТЬИ О ВТО И СТРАНАХ С ПЕРЕХОДНОЙ ЭКОНОМИКОЙ (В Т.Ч. РОССИИ) (на английском языке).
- Концепция развития автомобильной промышленности России
- Китай в ВТО: Либеральный протекционизм (Александр Салицкий, доктор экономических наук, ведущий научный сотрудник ИМЭМО РАН)
- Соглашение по торговым аспектам прав интеллектуальной собственности (ТРИПС): меры по обеспечению соблюдения прав интеллектуальной собственности



# Принципы Usability в построении страниц Web приложения (8/8)

Другие принципы Usability в построении страниц Web приложения:

- Не нужно делать сайты на фреймах:
  - Есть проблемы при переходе из поисковиков в дочерний фрейм – он как правило остается без навигации
  - Нет возможность поставит закладку и внешнюю ссылку
- На страницах должны быть «хлебные крошки»: заголовок страницы, строка состояния, подсказки - помогают посетителю определить положение страницы и легко вернуться на уровень выше
- Визуальный шум должен быть сведен к минимуму – иначе у пользователя теряется концентрация внимания
- Страница должна быстро грузиться – посетители быстро теряют терпение
- Внешние ссылки должны вести вглубь сайта – если даете ссылку на статью, неправильно заставляя пользователя искать ее самостоятельно
- Должна обрабатываться ошибка 404 (страница отсутствует) – пользователь должен знать, что произошло

# Проблемы на этапе разработки прототипа GUI, варианты решения (1/7)

- **Учет особенностей устройств ввода/вывода информации, используемых пользователем, например:**
  - размер экрана монитора
  - разрешение экрана
  - цветовая палитра
  - характеристики звуковой (качество воспроизведения речи) и видеокарты (скорость вывода при анимации)
  - вид мыши (с роликом или без)
  - тип клавиатуры ("прямая", "косая")
  - необходимость дополнительного оборудования (штрих-декодера, светового пера сенсорного экрана и др.)
- **Решение**
  - Проводить usability исследования: интервью и фокус-группы, работа с опросниками

# Проблемы на этапе разработки прототипа GUI, варианты решения (2/7)

- **Выбор технологии и методов ведения диалога программы с пользователем:**
  - степень учета ситуации (контекстные подсказки, меню дальнейших событий или объектов, запоминание типичных путей диалога)
  - соответствие ожиданиям пользователя (предсказание, предобработка, предформатирование)
  - устойчивость, терпимость к ошибкам пользователя путем исправления типичных ошибок
  - дублирование вручную отдельных функций системы и дополнительные контрольные процедуры работы отдельных режимов
  - Настройка UI на различный уровень подготовки пользователя, степень адаптивности UI под предпочтения пользователя
- **Решения:**
  - Анализ сценариев взаимодействия пользователя с системой
  - Использование методов usability исследования

# Проблемы на этапе разработки прототипа GUI, варианты решения (3/7)

- При размещении информации и управляющих элементов в поле экрана возникает задача оптимального расположения максимально возможного объема информации
- Решения:
  - Логическая увязка данных в зависимости от алгоритма работы пользователя, а не ориентация на структуру и последовательность физических таблиц данных
  - Нахождение компромисса между желанием вывести много записей одновременно и/или сразу увидеть детальную информацию по каждой из них
  - Выделение важной информации на экране, определение основных и вспомогательных блоков информации
  - Определение статических полей на экране, а также полей, где информация периодически изменяется
  - Применение принципов гармонии при компоновке экрана (симметрия, баланса масс, соблюдение пропорций, сочетание цветов)

# Проблемы на этапе разработки прототипа GUI, варианты решения (4/7)

- **Формирование обратной связи между пользователем и приложением**
- **Решения:**
  - Показ актуального состояния системы, режима работы системы (автономного, штатного, защищенного и пр.) и режима взаимодействия (*например, отображение, редактирование или поиск данных*)
  - Вывод отдельных, важных для рабочей операции данных и показателей
  - Отражение действий пользователя (нажатия клавиш, запуск процесса, динамика выполнения процесса, получение ожидаемого и иного результата)
  - Ясность и информативность сообщений системы

# Проблемы на этапе разработки прототипа GUI, варианты решения (5/7)

- **Проектирование панелей меню и инструментов (toolbars) и выбор пунктов в них**
- **Решения:**
  - Следуйте стандартным соглашениям о расположении пунктов меню принятым в Windows File, Edit, View, и т.д.
  - Группируйте пункты меню в логическом порядке и по содержанию
  - Для группировки пунктов в раскрывающихся меню используйте разделительные линии
  - Фиксированная позиция панелей на экране
  - Ограничение на ширину списка выборов и шагов (глубины) меню
  - Использование привычных названий, широко распространенных иконопиктограмм, традиционных икон-символов и аккуратное введение сокращений
  - Размещение наиболее часто используемых пунктов (обычно в начале списка)
  - Избегайте избыточных меню

# Проблемы на этапе разработки прототипа GUI, варианты решения (6/7)

- **Разработка средств ориентации и навигации**
- **Решения:**
  - Легкость определения своего местонахождения и указание направления следования
  - Удобный переход от обобщенного взгляда до конкретных деталей (варьирование степени детализации рассматриваемых объектов)
  - Быстрый поиск в списке или таблице
  - Указание на дополнительно существующую информацию и способ ее получения
  - Использование средств листания и прокрутки

# Проблемы на этапе разработки прототипа GUI, варианты решения (7/7)

- **Создание форм для ввода данных**
- **Решения:**
  - Использование одного или нескольких механизмов ввода в рамках режима (клавиатура, мышь, штрих-декодер, световое перо, др.)
  - Определение способов ввода данных (таблицы, списки, простая форма, меню и пр.)
  - Минимизация объема ввода (гибко поддерживайте формат телефонных номеров, номеров кредитных карточек и пр.)
  - Устанавливайте курсор автоматически на первом поле формы
  - Выделение редактируемых обязательных и необязательных, а также нередитуемых полей
  - Сделать обязательными лишь те поля, которые точно нужны пользователю
  - Использование механизмов быстрого ввода (по умолчанию, сокращения, с продолжением, функцию автозаполнения и пр.)
  - Выделение введенной или отредактированной информации



# Проектирование war/web-интерфейсов мобильных устройств (МУ)

- Под проектированием интерфейса здесь понимается интерфейс разрабатываемого сайта, системы, приложения, программы для использования на МУ
- МУ - сотовый телефон, смартфон, тачскрин-телефон и др.
- Основные проблемы при проектирования интерфейса для МУ:
  - Пользователь выполняет много задач одновременно
  - Экран МУ редко находится в условиях идеального освещения
  - Малоразмерный экран
  - Параметры экранов пользователей значительно варьируются
  - Малоразмерный кэш и cookies
  - Ограниченные возможности ввода
  - Медленная загрузка данных



# Решение проблем проектирования интерфейса для МУ (1/4)



- **Пользователь выполняет много задач одновременно, в движении**
  - Программа должна быть терпима к случайным ошибкам пользователя – например, случайный, неправильный ввод возможен за счёт произвольных и случайных толчков при движении
  - Должен быть чётко обозначен контекст выполняемой в приложении операции – в первую очередь за счёт коротких, но информативных заголовков, корректного и короткого именованя ссылок и командных кнопок
  - В интерфейсе не должно быть ничего лишнего, что отвлекало бы внимание пользователя или запутывало бы его в принятии решений
  - Не следует экономить на памяти пользователя – критическая информация, важная для выполнения задачи, должна быть доступной
- **Малоразмерный кэш и cookies**
  - Страницы с объёмным содержимым (длинные списки и тексты) следует разбивать на несколько страниц
  - Система навигации должна минимальной, но соответствовать текущим потребностям (выполняемой задаче) пользователя

# Решение проблем проектирования интерфейса для МУ (2/4)



- **Экран МУ редко находится в условиях идеального освещения**
  - Следует особое внимание уделять соблюдению наивысшего контраста текста: располагать тёмный текст на светлом фоне, или наоборот – светлый текст на тёмном фоне (наивысший контраст даёт отношение чёрный/белый)
  - Желательно избегать ярких фоновых заливок использования графических обоев, затрудняющих восприятие текстовой информации. Если всё-таки таковые используются, то следует применять двуцветные шрифты (чёрные символы с белой окантовкой)
  - Не применяйте синий и пурпурный (лилово-фиолетовый) цвет в оформлении текста во избежание перепутывания такого оформления с гиперссылками, особенно на устройствах, в которых ссылки не подчёркиваются
  - По возможности следует применять крупный шрифт или обеспечить пользователю возможность изменять размер шрифта
- **Параметры экранов пользователей значительно варьируются**
  - Следует применять вёрстку без привязки к размерам экрана, так, чтобы данные корректно отображались на экранах разного размера
  - Цветовое оформление должно адекватно отображаться на экранах с различной глубиной цвета, в т.ч. и монохромных

# Решение проблем проектирования интерфейса для МУ (3/4)

- **Малоразмерный экран**

- Информация должна быть представлена в сжатом, но информативном виде
- Лишних и второстепенных деталей не должно присутствовать в интерфейсе
- Следует использовать принцип «вложенности», т.е. давать информацию порциями
- Выводы, итоговые и критические данные, ключевые слова следует помещать в начало страницы
- В таблицах отображать только самые важные столбцы

- **Медленная загрузка данных**

- Минимизировать объём единомоментно загружаемых страниц (оптимизировать код и изображения, использовать небольшие картинки только там, где это необходимо, разбивать объёмные страницы на несколько страниц, и т.д.)
- Большие изображения следует предварять миниатюрами с явным указанием размеров и объёма загружаемой картинки
- Ссылки на любой загружаемый мобильный контент следует специфицировать в явном виде (тип, объём, размер, и т.д.)
- Реклама должна быть явным образом и конситентно обозначена

# Решение проблем проектирования интерфейса для МУ (4/4)



- **Ограниченные возможности ввода (неудобно, медленно и чревато ошибками)**
  - Свести количество нажатий клавиш к минимуму. *Например, следует использовать как можно более короткие URL-адреса*
  - Следует заменять поля ввода другими контролами, которые не требуют длительного ввода: разворачивающимися списками или радиогруппами
  - Количество полей, обязательных для заполнения, должно быть сведено к минимуму, необязательные для заполнения поля убрать вообще либо вынести на отдельную страницу
  - Максимальное количество полей в заполняемых пользователем формах должно быть заполнено значениями по умолчанию
  - Заполнять поля формы по умолчанию, восстанавливать заполненную пользователем форму, применять технику автозаполнения
  - Облегчить навигацию, использовать клавиши доступа (быстрого перехода), обеспечивать возврат к предыдущему экрану (Back), особенно там, где это может помочь исправить некорректный пользовательский ввод

# Практическое занятие (3)



## По созданию прототипа GUI Системы методом бумажного прототипа (приложение Windows)

### Задание:

#### Описание:

1. Оператор страховой компании для оформления страховки по системе КАСКО может ввести новый контракт или отредактировать старый контракт клиента
2. В контракте выполняется предварительный расчет суммы страховки
3. В интерфейсе по каждому контракту должна быть возможность для ввода, редактирования и просмотра следующей информации:
  - a. Данных страхователя. Страхователем м.б. и физическое лицо (ФИО, место работы, должность, документ, параметры документа, адрес, стаж, возраст водителей), и юридическое (Организация, ИНН, свидетельство о регистрации, параметры свидетельства, юр. адрес, кол-во водителей, минимальный стаж, возраст)
  - b. Данных транспортного средства (модель, тип двигателя, год выпуска, первоначальная стоимость, пробег, ущерб, место хранения, ПС)
  - c. Данные по страховке (дата выдачи полиса, период страхования, система страхования)

# Выполнение практического задания



## Слушатели должны:

Спроектировать пользовательский интерфейс приложения, которое позволяет оператору производить навигацию по контрактам клиентов, вводить, редактировать, просматривать котировки для страхования транспортного средства клиентов

## Примечание

Кроме того, приложение должно давать возможность:

- a. Просматривать сумму рассчитанной страховой суммы исходя из заданных начальных условий страхования
- b. Распечатывать страховой полис с рассчитанной суммой и условиями страхования, данными страхователя и его транспортного средства
- c. Пользовательский интерфейс Системы должен соответствовать принципам и концепции Windows GUI

Время выполнения задания - 20 минут,

Время обсуждения выполненного задания – 10 минут

# Практическое занятие (4)

## По созданию прототипа GUI Системы методом бумажного прототипа (приложение Web)

### Задание:

Описание: Система Web для учета сети торговых точек:

1. В системе есть справочник торговых точек. Каждая торговая точка описывается параметрами (код, наименование, город, логотип)
2. В данной сети действуют три типовых конфигурации, каждой из которых присущи свои регламентирующие документы: шаблоны и справочники
3. Параметры шаблона: наименование, файл, дата модификации; параметры справочника: название, комментарии
4. Каждая точка продаж может относиться к одной из конфигураций. Любой из операторов может включить точку продаж, которая еще не принадлежит ни к одной из конфигураций в любую из конфигураций. Точка продаж может быть включена только в одну конфигурацию. К одной конфигурации может принадлежать несколько точек продаж
5. После внесения изменений в конфигурацию (справочники, шаблоны, точки продаж) оператор должен опубликовать их для общего просмотра. До этого другой оператор с данной конфигурацией работать не может (конфигурация помечается, как заблокированная)



# Выполнение практического задания



## Слушатели должны:

Спроектировать пользовательский интерфейс приложения, которое позволяет оператору:

1. Производить навигацию по справочнику торговых точек и конфигураций
2. Добавлять новую (открывшуюся) торговую точку, изменять ее параметры. Опубликовывать добавленную точку в общий справочник торговых точек для всеобщего просмотра, удалять точку из справочника
3. Просматривать, редактировать и опубликовывать изменения в конфигурациях торговых точек (изменения в шаблонах и справочниках, изменения в составе торговых точек). Шаблон можно редактировать, удалить, загрузить новый из файла. Справочник можно изменить и удалить

Время выполнения задания - 20 минут,

Время обсуждения выполненного задания – 10 минут

## 6. Usability тестирование

- Usability исследование
- Usability тестирование
- Экспертная оценка
- Путеводитель в мире методов usability-тестирования

Практическое задание - тестирование usability Системы методом контрольного списка интерфейса

# Следует отличать

- Usability исследование
- Usability тестирование
- Экспертная оценка



# Цели проведения usability-исследования



## Цели:

- Получить ответы на поставленные вопросы, заинтересовать пользователей, выявить профили
- Наблюдение за поведением пользователей в их рабочей среде, сбор информации о потребностях пользователей, выявление проблемных ситуаций
- Сбор информации, предпочтений
- Сбор требований, выявление конкретных групп пользователей

# Методы usability-исследования



- Интервьюирование
- Анкетирование
- Ведение дневника
- Фокус-группы
- Полевые исследования и этнографические методы
- Протоколирование реального использования
- Работа с опросниками

# Пример usability-исследования мобильных версий сайтов



При проведении usability-исследований мобильных версий сайтов были использованы следующие методы usability исследования:

- **Ведение дневника**

- Участники в течении недели заносили в дневник все действия, которые они выполняли с помощью своих МУ.
- Затем, в конце дня они заполняли опросники для сбора более подробных данных

- **Задания:**

- Найдите погоду на сегодня
- Найдите, что будет идти по BBC TV сегодня в 20:00
- Узнайте, не откладывается ли сегодня 12-часовой рейс Мюнхен-Лондон

- **Исследования в фокус-группах**

- В usability-тестировании приняли участие 48 чел. Со своими телефонами. Соотношение мужчин/женщин 50/50. Возраст – от 20 до 48 лет.
- 33 чел. проживали на территории США, 15чел. – в Лондоне

- **Мульти-платформенная проверка**

- Был протестирован дизайн 20 сайтов с использованием 6 различных телефонов: 3 разных смартфона, 2 тачфона и мобильные телефоны

# Выводы Usability исследования мобильных версий сайтов

## Выводы:

- Эксперимент с дневниками позволил наблюдать поведение пользователей **в естественной среде** и на значительном отрезке времени
- Ожидалось, что в Лондоне будут более высокие показатели, т.к. в Великобритании сильнее традиции использования мобильных сервисов, но показатели эффективности использования оказались везде низкие. **Нужно использовать мобильные версии сайтов:** пример, когда участники тестирования пользовались версиями сайтов, разработанными специально для МУ, то средний процент успешно выполненных операций увеличивался на 15-20% в сравнении, если использовался обычная версия сайта
- Выяснено, что даже при наличии мобильной версии сайта пользователи не находят ее. **Нужно определять устройства пользователя, и автоматически переадресовывать на мобильную версию сайта**
- **Чем больше экран МУ, тем выше средний показатель успешно выполненных задач:** обычные телефоны – 38%, смартфоны – 55%, тачскрин-телефоны – 75%
- При сравнении результатов исследований с прошлыми, выяснено, что пользователи стали тратить на выполнение тех же заданий на 38% времени больше, чем в 2000 году. Это объясняется тем, что пользователи были ограничены узким набором сайтов. Сегодня **набор сервисов и возможностей расширился**, и у пользователей стало больше свободы выбора

# Исследование желанности визуального дизайна

- **Исследование желанности визуального дизайна** – это совокупность количественных и качественных подходов, позволяющих оценить предпочтения пользователей, относящихся к эстетике и визуальной привлекательности
- Дизайн может оказать эмоциональное влияние очень быстро
- Эмоциональный отклик на первое впечатление пользователя от дизайна переносится на восприятие таких качеств продукта или приложения, как **полезность, удобство и надежность**
  - *Например, если пользователь имеет **положительное** первое впечатление от эстетики дизайна, они с большей вероятностью не заметят, а если и заметят, то забудут **низкий уровень удобства** или **ограниченную функциональность***
  - *С **отрицательным** первым впечатлением, пользователи с большей вероятностью найдут **проблемы во взаимодействии**, даже если удобство использования продукта в общем хорошее и продукт предлагает реальную ценность для пользователя*



# Методы измерения желанности (удовлетворенности пользователя)

- Интервью
- Анкетирование
- Сопоставление трех
- Быстрые тесты памяти
- Наблюдение за эмоциональными реакциями
- Измерение физиологических показателей

# Интервьюирование (1/2)

Интервью – это разговор специалиста и респондента по заранее разработанному плану

## ■ Подготовка

- Подготовка тем для обсуждения
- Выстроить последовательность из тем
- Подготовить вопросы (+уточняющие вопросы)

## ■ Проведение

- Вводная фраза (представление, цели, разогрев)
- Побуждающая фраза
- «Тело» интервью
- Закрывающая фраза (суммирование полученной информации)

## ■ Обработка результатов



Плюсы: гибкость, расширение социальных связей.  
Минусы: дороговизна, возможность искажения.

# Интервьюирование (2/2)

## Рекомендации:

- Избегать намеков на оценку и наводящих вопросов (вместо «Нравится ли вам этот экран?» - «Что вы думаете об этом экране?»)
- Быть готовым объяснить каждый вопрос
- Фокусировать человека на его личном опыте
- Почаще задавать вопрос «Почему?»
- Не смотреть на часы
- Слушать, а не говорить
- Не обобщать раньше времени
- По возможности, проводить там, где происходит реальное взаимодействие пользователя с продуктом
- Не слушать рекомендаций пользователя из области UI, лучше поощрять пользователя рассказывать истории

- Анкеты привлекают простотой и легкостью применения при работе с большими объемами информации
- Многие из опросников включают значительное число вопросов о взаимодействии и требуют от участников иметь определенный опыт взаимодействия с сайтом или приложением, поэтому не подходят для быстрого сравнения визуального дизайна статического макета
- Надежных анкет нет. Есть вполне работоспособные анкеты:
  - Анкета по словам
  - Формальная анкета

# Анкетирование по словам

- Респонденту предъявляется лист бумаги с набором случайно подобранных прилагательных, одна половина которых скорее позитивна, вторая – негативна
- Респонденту предлагается подчеркнуть слова, которые, на его взгляд, применимы к продукту
- После того, как анкета заполнена, подсчитывается разница между числом негативных и позитивных терминов

**Пример анкеты:** Устаревший – Эффективный – Нечеткий – Неудобный – Замусоренный – Тусклый – Яркий – Чистый – Прямой – Ясный – Непоследовательный – Неуправляемый – Привлекательный – Стандартный – Управляемый – Хороший – Интуитивный – Веселый – Любительский – Неэффективный – Опасный – Скучный – Радостный – Безопасный – Жесткий – Раздражающий – Треугольный – Неприятный – Комфортабельный – Холодный – Умный – Беспольный – Халтурный – Теплый – Светлый – Последовательный – Загадочный – Качественный – Интересный – Ненадежный – Гибкий – Красивый – Некрасивый – Непривлекательный – Полезный – Глупый – Запутанный – Удобный – Понятный – Непредсказуемый – Четкий – Тяжелый – Современный – Легкий – Дружественный – Нестандартный – Плохой – Надежный – Сложный – Простой – Темный – Профессиональный – Медленный – Круглый – Печальный – Недружественный – Предсказуемый – Непонятный – Быстрый – Головоломный – Грустный – Приятный

# Формальная анкета

Данный вид анкет нужно адаптировать под конкретный проект

Пример:

	1	2	3	4	5
Во время выполнения заданий я ошибался	Нет				Да
Система способна делать все, что мне нужно и даже больше	Нет				Да
Система работает достаточно быстро	Нет				Да
Мне нравится внешний вид интерфейса	Нет				Да
Я чувствую, что если я лучше изучу систему, я смогу делать в ней вещи, о которых сейчас даже и не подозреваю	Нет				Да
Систему можно легко настроить под мои нужды	Нет				Да
Начать работу было легко; я не столкнулся с существенными трудностями	Нет				Да
Всякий раз, когда я ошибался, я с легкостью замечал и исправлял свою ошибку	Нет				Да
Я доволен своей скоростью работы	Нет				Да
Во время выполнения заданий я чувствовал себя вполне уверенно	Нет				Да
В любой момент времени я понимал, что должен сделать дальше	Нет				Да
Система представляется мне полезной, я бы с удовольствием использовал бы её для решения моих задач	Нет				Да

**Результаты нужно подсчитывать по следующему алгоритму:** центральное значение дает ноль баллов, крайние значения дают либо  $-2$  балла (левый вариант ответа), либо  $+2$  балла (правый вариант), промежуточные значения либо  $-1$  либо,  $+1$  балл соответственно. Сумма баллов является сравниваемым значением

# Сопоставление трех

- Участникам исследования предоставляется три варианта дизайна и просят идентифицировать два, отличающихся от третьего, и объяснить, почему они отличаются
- **Цель:**
  - Выявить атрибуты, которые участники исследований и целевая аудитория могли бы использовать для сравнения нескольких альтернатив, не подвергаясь влиянию со стороны исследователей
  - Помогает понять, какие аспекты важны для целевой аудитории при сравнении разных дизайнов
- **Недостатки:**
  - Сложный при проведении исследования с большим числом участников
  - Кроме того, результаты сложно представить заинтересованным лицам в таком виде (например, в виде таблиц), чтобы они помогли выбрать лучший вариант дизайна

# Быстрые тесты памяти

- Исследователи показывают на мгновение (на очень короткий момент) участникам пользовательский интерфейс, затем убирают его
- После они опрашивают участников: что они запомнили из пользовательского интерфейса за время недолгой демонстрации
- Так как участники ограничены во взаимодействии с сайтом или приложением, то, теоретически, они рассказывают вам о мимолетном впечатлении, застывшем в их памяти
- Метод полезен для выявления впечатлений:
  - от главной страницы и других стартовых страницах
  - при оценке структуры и информационного дизайна



# Наблюдение за эмоциональными реакциями

- **Принцип:**
  - Улыбнулся – ставим плюс
  - Выругался или поморщился – ставим минус
  - Количество и знак реакций и составляет искомую величину показателя
- **Недостатки:**
  - ✓ Непонятно, как подсчитывать реакции разной силы
  - ✓ Субъективная оценка и субъективное понятие эмоциональной реакции
  - ✓ Наблюдается при этом только операционная удовлетворенность, и не учитывается воспринимаемая удовлетворенность, что важнее

# Измерение физиологических показателей

## Методы измерений:

- *Электроэнцефалография (ЭЭГ)* измеряет активность разных участков мозга, которую можно сопоставить с определенной эмоциональной реакцией
- *Электромиография (ЭМГ)* измеряет мышечную активность, которая соотносится с уровнями возбуждения
- *Электродермическая активность (ЭДА)* измеряет активность потовых желез, которые, как считается, связаны с возбуждением и волнением
- *Уровень кровяного давления* измеряет расширение кровяных сосудов, которое, в свою очередь, связано с возбуждением
- *Расширение зрачков* появляется в связи с возбуждением и умственной рабочей нагрузкой
- *Измерение дыхания* может свидетельствовать об отрицательной валентности или возбуждении

## Проведение исследований:

- *В ходе исследований разные датчики отслеживают физиологические показатели, в то время как исследователи показывают участникам разные дизайны*
- *Изменение одного или более показателей свидетельствуют о определенном эмоциональном отклике*

**Недостатки:** Затратный метод по месту проведения, времени и бюджету

# Практическое занятие (5)



## Проведение анкетирования удовлетворенности пользователя интерфейса

### Задание:

#### Описание:

1. Каждый из слушателей должен выбрать один из нескольких web-сайтов или приложений, например:
  - <http://www.google.ru/> (рассматривать, в том числе, настройку поиска)
  - <http://www.yandex.ru/> (рассматривать, в том числе, настройку поиска)
  - <http://promis.luxoft.com/default.aspx> (заполнение тайм-шотов)
  - <https://sentinel2.luxoft.com/sen/> (Wiki -страницу проекта)

- другое, предложенное самим слушателем и хорошо ему знакомое
2. Проанализировать работу приложения/ сайта с точки зрения удовлетворенности потенциального пользователя

# Выполнение практического задания



## Слушатели должны:

1. По аналогии с представленным на слайде 115 фрагментом анкеты создать список вопросов для анкетирования удовлетворенности пользователя при работе с сайтом/ приложением
2. Разбиться по парам и опробовать работоспособность анкеты на респонденте

Время выполнения задания - 15 минут,

Время работы в парах – 10 минут,

Обсуждение выявленных при анкетировании результатов – 10 минут

# Определение usability-тестирования

## Варианты определения usability-тестирования:

- **Usability-тестированием** является любой эксперимент, направленный на измерение качества интерфейса или же поиск конкретных проблем в нем.
- **Usability-тестирование** представляет собой постановку экспериментов с целью выявления специфичной информации, касающейся дизайна исследуемого продукта

Usability-тестирование не может сделать из плохого продукта хороший продукт, оно всего лишь делает продукт лучше

# Usability-тестирование на разных этапах проектирования UI

## Анализ

- Количественное usability-тестирование аналогов и предыдущих версий продукта для определения объективных критериев качества пользовательского интерфейса. Этот метод тяжело применять в случае, если пользовательская деятельность автоматизируется впервые, либо есть другие неопределённые факторы (например, неизвестны временные задержки, определяемые реакцией оборудования).

## Проектирование

- Экспресс-тестирование быстрых прототипов для получения объективного подтверждения правильности выбора того или иного интерфейсного решения – сравнительная оценка альтернативных интерфейсных решений, либо подтверждение факта того, что выбранное интерфейсное решение не вызывает серьёзных проблем в применении

## Тестирование

- Полномасштабное usability-тестирование для определения проблемных областей в пользовательском интерфейсе а также для определения того, достигаются ли заданные критерии качества в новом интерфейсе

# Цели usability-тестирования



- Понять, насколько плохо или хорошо работает интерфейс
  - что может либо побудить улучшить его,
  - либо, если он уже достаточно хорош, успокоиться
  - в любом случае достигается польза
- Сравнить качество старого и нового интерфейсов, и тем самым дать обоснование изменениям или внедрению
- Найти и опознать проблематичные фрагменты интерфейса, а при достаточном объеме выборки также и оценить их частотность

Первое тестирование крупных систем всегда показывает, что интерфейс работает гораздо хуже, чем думает его владелец или создатель, и гораздо лучше, чем изначально уверен тестер

# Что измерять при тестировании? (объекты тестирования)

Все измеряемые показатели сводятся к набору из пяти базовых характеристик:

- **Скорость работы пользователя.** *Метрики:* длительность выполнения операции; количество команд, исполняемых при выполнении операции, длительность поиска сведений в документации; количество команд, более эффективных, чем использованные пользователем; снижение производительности при длительной работе
- **Ошибки.** *Метрики:* процент операций, вызвавших ошибку; среднее число ошибок на операцию у опытных пользователей, количество ошибок, не обнаруженных и не исправленных пользователями
- **Обучаемость навыкам работы с системой.** *Метрики:* количество и частота обращений к справочной системе; длительность периода между началом использования системы и точкой, в которой скорость работы/количество ошибок пользователей перестает расти; разница в количестве ошибок/скорости работы у пользователей с опытом использования системы и без такого опыта
- **Субъективная удовлетворенность пользователя.** *Метрики:* подсчет эмоциональных реакций респондента – сложность в оценке (реакции разной силы и субъективная оценка)
- **Сохранение навыков работы с системой.** *Метрики:* разница в скорости работы/количестве ошибок у пользователя после часа работы с системой и у того же пользователя в начале использования системы после длительного перерыва



# Что нужно для usability-тестирования?

- **Респонденты.** Определить общие требования к респондентам (опыт работы и уровень компьютерной грамотности, возраст, пол, уровень эмоциональной открытости); выбор респондентов не из целевых групп опасен: выявит несуществующие и не выявит существующие проблемы
- **Метод тестирования** (будут рассмотрены далее)
- **Тестовые сценарии** – это тестируемый аспект системы
  - Адекватно подобранные тестовые сценарии являются важнейшей предпосылкой качества тестирования
  - Тестовый сценарий состоит из тестовых заданий и признаков успешности выполнения задачи
  - Все задания должны быть реальными, выбирать только важные задачи
  - Тестовые задачи должны характеризоваться следующими свойствами: *Однозначность, Полнота, Краткость, Отсутствие подсказок*
- **Рабочее место для теста и отлаженный метод фиксации материала**
- **Протестированный тест**

# Методы фиксации результатов тестирования

- Веб-камера
- Микрофон
- Программа записи содержимого экрана
- Спортивный секундомер
- **Тестовое задание в виде распечатки** (каждый шаг на отдельном листе, чтобы респондент не мог забежать вперед и прочитать то, что еще не выполнил; на первом листе вводная форма с рекомендациями и предупреждениями пользователю, описанием цели тестирования и поведения пользователя во время тестирования)
- **Для анкетирования – распечатанные анкеты**
- **Для фиксации параметров, находясь рядом с респондентом – планшет с бумагой и ручкой**

# Недостатки использования usability-тестирования

- Usability-тестирование часто приравнивают ко всей usability-технологии. Это не так
- Usability-тестирование есть метод необходимый, но вовсе не достаточный для создания качественного программного продукта
- Этот метод не всегда эффективный, т.к.
  - с одной стороны, сам по себе достаточно дорог (по людским, временным и материальным ресурсам),
  - с другой стороны, на этапах анализа и проектирования используется лишь как вспомогательный,
  - а на этапе тестирования выявленные концептуальные проблемы устранять крайне тяжело (уже потрачены серьёзные ресурсы на создание этих проблем, и может уже не хватать ресурсов на их устранение)

Как упростить тестирование, чтобы сделать его доступным?

# Пути снижения трудоемкости usability-тестирования

**Резко снизить трудоемкость, а значит, и себестоимость usability-тестирования позволяют следующие подходы:**

- Некоторое упрощение понятие usability – оперировать только понятиями:
  - Эффективность
  - Успешность
  - Удовлетворенность
- Для измерения показателя эффективности продукта ограничиться измерением показателей:
  - Скорость работы пользователя
  - Скорость обучения
  - Количество человеческих ошибок
- Отказ от сбора количественных данных

## Несколько фактов о тестировании:

- Если Вы хотите получить хороший сайт, то его нужно тестировать
- Лучше протестировать одного пользователя, чем не тестировать ни одного
- Лучше протестировать одного пользователя в начале разработки, чем 50 пользователей в конце
- Важность привлечения репрезентативных пользователей преувеличена
- Важность тестирования не в том, чтобы что-то доказать или опровергнуть, а в том, чтобы собрать достаточно информации для получения критической оценки
- Тестирование – это повторяющийся процесс
- Лучшим средством тестирования удовлетворенности является живая реакция аудитории

# Основные положения и тезисы

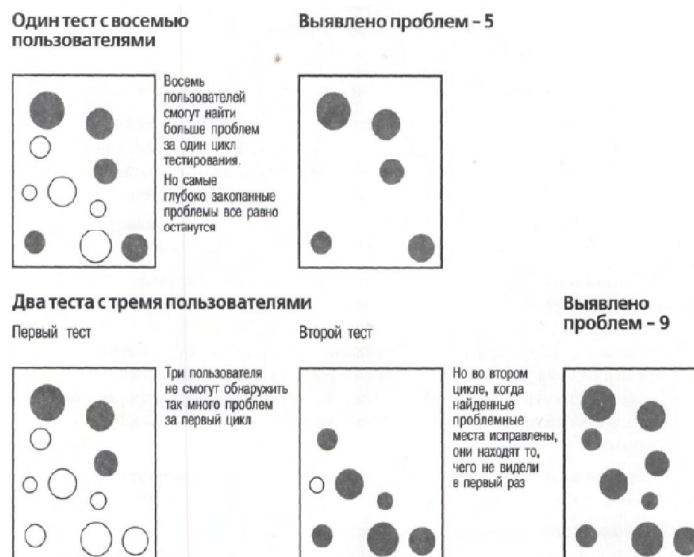
## Сверхэкономное usability-тестирование

Определитель	Традиционное тестирование	Сверхэкономное тестирование
Число пользователей на тест	Обычно от 8-ми, чтобы оправдать запланированные расходы	3 или 4
Способ отбора пользователей	Тщательный отбор в соответствии с целевой аудиторией	Хватайте первых попавшихся людей. Подойдет почти каждый, кто хоть как-то умеет пользоваться Интернетом
Место проведения тестирования	Специальная лаборатория с комнатой наблюдения и зеркальным стеклом	Любой офис или комната
Кто проводит тестирование	Опытный специалист по usability	Любое разумное существо с терпением
Предварительное планирование	Требуется предварительно (за несколько недель) зарезервировать лабораторию и отобрать пользователей	Тестирование можно проводить почти в любое время с минимальным предварительным планированием
Подготовка	Необходимо составить, обсудить и утвердить протокол тестирования, группу респондентов	Надо только решить, что вы будете демонстрировать пользователям (респондентам)
Периодичность и объем тестирования	Если вы не располагаете огромными средствами, то проводится одно тестирование сразу по всем вопросам почти в конце разработки проекта	Периодически проводятся небольшие тесты на протяжении всего процесса разработки
Стоимость	От 5 000 до 15 000 (и более) долларов	Около 300 (от 50 до 100 каждому респонденту + 20 на видеопленку) долларов
Что происходит после тестирования	Через неделю после тестирования появляются 20-страничный письменный отчет. Затем проводится собрание всей рабочей группы для обсуждения результатов тестирования	Сразу после тестирования каждый наблюдатель пишет 1-страничное резюме, которое можно обсудить в тот же день

# Основные положения и тезисы

## Выводы:

- Лучше меньше респондентов, но больше циклов тестирования, и начинать тестирование следует как можно раньше



- В большинстве случаев в качестве респондентов могут быть ЛЮБЫЕ пользователи, необязательно из целевой аудитории

# Тестирование usability ПО (1/3)



- **Кто проводит?**

Может проводиться только группой тестирования компании-разработчика

- **Методы**

- **Типовые методики** (контрольные списки и т.д.) - обеспечивают принцип «необходимого минимума», но не дают оценки того, насколько удобно конечному пользователю работать с системой:

- **Общие**

- ПРИМЕР: контрольный список интерфейса*

- **Частные** (разрабатываются на основе типового контрольного списка с учетом специфики приложения)

- ПРИМЕР: контрольный список интерфейса web-сайта*

- **Usability-тестирование на основе пользовательских вариантов использования** - помогает понять, насколько пользователю удобно работать с системой




# Тестирование usability ПО (2/3)

## Контрольный список интерфейса

- Позволяет провести достаточно формальное и технологичное тестирование качества реализации UI
- Позволяет обеспечить соблюдение принципов удобства использования при проектировании на разных этапах проекта
  - **Требования к конкретным элементам управления**
    - Кнопки
    - Поля ввода
    - Списки
    - Чекбоксы и радиокнопки
  - **Взаимодействие**
    - Системные сообщения и обработка ошибок
    - Клавиатура
  - **Визуал**
    - Индикация
    - Пиктограммы
  - **Окна**
    - Строка статуса
    - Меню
    - Контекстное меню
    - Структура интерфейсных форм
    - Формы ввода
  - **Текст**

# Тестирование usability ПО (3/3)

## Пример того, как может быть описан раздел контрольного списка интерфейса web-сайта «Устойчивость к ошибкам»

- Архитектура и навигация
  - Планировка и дизайн
  - Содержание
  - Формы и взаимодействие
  - Графика
  - Цвета
  - Оформление текста
  - **Устойчивость к ошибкам** 
  - Платформа и особенности реализации
- Должен ли пользователь что-нибудь запоминать, переходя между страницами?
  - Возникает ли предупреждение при попытке совершения необратимых или дорогостоящих действий?
  - Можно ли отменить рискованные или дорогостоящие действия?
  - Перехватываются ли возникающие ошибки локально, без обращения к серверу?
  - Содержат ли страницы с сообщением о возникших ошибках полезную информацию?
  - Содержат ли страницы с пустыми результатами поиска советы по расширению условий поиска?
  - Существует ли система помощи (справки)?
  - Структурирована ли помощь по задачам пользователя? Объясняет ли она пользователю, как совершить то, или иное действие?
  - Система помощи контекстно-зависимая?
- В описании разделов критично прописывать используемую платформу (например, тип и версию браузера), разрешения, размеры используемого пользователем монитора, особенности реализации (использование Flash, Flex, др.

# Контрольный список интерфейса

## Требования к конкретным элементам управления

### ■ Кнопки

- Все кнопки, запускающие действия, имеют текст в инфинитивной форме глагола (пример: искать), а не другую часть речи либо форму глагола (пример: готово). Давать кнопке текст «ОК» можно, только если какой-либо глагол не вмещается
  - **Примечание: ОК, Отмена – вполне допустимо.**
- Кликабельный размер кнопок совпадает с их видимым или логическим размером
- Между кнопками, стоящими рядом, должно быть пустое пространство, щелчок по которому не отрабатывается
- Нет разных состояний кнопок, которые выглядят одинаково
- Недоступные команды не исчезают с экрана, а становятся заблокированными
  - **Примечание: не всегда (состояние рабочей области vs. отсутствие прав)**
- Частотные кнопки снабжены не только текстом, но и пиктограммами; редко используемые кнопки - только текстовыми подписями.
- В модальных диалоговых окнах нет кнопок Применить.
  - **Примечание: Есть исключения: например, выбор стилового оформления**

# Контрольный список интерфейса



## Требования к конкретным элементам управления

### ■ Поля ввода

- В полях ввода уже стоят наиболее вероятные значения

#### Примечание: Не всегда...

- Если в поле вводится численное значение, границы диапазона выводятся во всплывающей подсказке
- Если в поле вводится численное значение из ограниченного диапазона, поле снабжено крутилкой (Spinner)
- Длина полей не меньше, и, по возможности, не больше, длины вводимых в них данных
- Если поле предназначено для ввода заметного количества текста, оно многострочное
- Многострочные поля имеют максимально возможную высоту; нет резервов для их увеличения

# Контрольный список интерфейса

## Требования к конкретным элементам управления

### ■ Списки

- В списках уже стоят наиболее вероятные значения
- Если список содержит более 50 элементов, используется фильтр или режим поиска
- Не использовать короткие списки (менее пяти элементов); такие списки представлены как группы радиокнопок или чекбоксов
- Ширина списков не меньше ширины входящих в них элементов
  - **Примечание:** Очень частая ошибка – делать горизонтальную прокрутку
- Элементы списка отсортированы; либо структурно, т.е. по общим признакам, либо по алфавиту, либо по частотности (только списки меньше 7 элементов)
- Если в списке более 50 отсортированных по алфавиту элементов, первыми тремя элементами являются наиболее частотные элементы. Они также повторяются на своих алфавитных местах
- Многострочные списки множественного выбора снабжены чекбоксами возле каждого элемента (списки старого стиля отсутствуют)
- Многострочные списки имеют высоту не менее 4 строк
- Если есть свободное место, используются расширенные комбобоксы, а не однострочные

# Контрольный список интерфейса



## Требования к конкретным элементам управления

- **Чекбоксы и радиобаттоны**
  - Если чекбоксов в группе больше 10, вводится дополнительный, выставляющий/снимающий все чекбоксы
  - Внутри группы радиокнопок одна обязательно установлена по умолчанию
  - Чекбоксы и радиокнопки внутри своих групп расставлены по вертикали
  - Если в окне, помимо терминационных кнопок, есть только набор радиокнопок, двойной щелчок по радиокнопке устанавливает ее и закрывает окно

# Контрольный список интерфейса

## Взаимодействие

### ■ Системные сообщения и обработка ошибок

- В формах ввода проверка корректности вводимых значений выполняется прямо во время ввода; если вводятся некорректные данные, система сразу сообщает об этом пользователю, не дожидаясь момента, когда пользователь завершит ввод данных во всей форме
- Сообщения о некорректности введенных данных показываются рядом с элементом управления, данные в котором некорректны
- Текст сообщений о некорректности введенных данных не говорит, что, дескать, совершена ошибка, напротив, он только информирует пользователя, данные какого типа и формата приемлемы
- Текст сообщений о проблемах состоит из трех частей: в первой кратко описывается проблема, во второй части - как ее решить, в третьей - описывается, как не допускать возникновения этой проблемы в дальнейшем
- Статусные сообщения («Синхронизация успешно завершена») выводятся только в строке статуса

# Контрольный список интерфейса

## Взаимодействие

### ■ Клавиатура

- В формах ввода нажатие табуляции ведет к правильной последовательности перемещения по форме
- Обработка формы запускается не только по нажатию на терминационную кнопку, но и по нажатию клавиши Enter на последнем поле этой формы
- Для наиболее частотных элементов управления (включая меню) установлены клавиши быстрого вызова
- Каждому пункту меню назначены ALT-комбинации (выделены подчеркиванием)
- ALT-комбинации и горячие клавиши стандартные
- Если горячих клавиш больше 40, в интерфейсе есть способ их изменить
- По нажатию клавиши Tab переход от элемента к элементу внутри формы осуществляется сверху вниз, слева направо



# Контрольный список интерфейса



## Визуал

- Направление теней во всех элементах управления должно быть одинаковым: снизу справа
- **Индикация**
  - Индикация цветом не является единственной; если она используется, система снабжена и другой индикацией
- **Пиктограммы**
  - В группах пиктограмм нет пиктограмм, по цвету и форме сходных между собой
  - Нет пиктограмм со стандартными значениями, но нестандартными сюжетами
  - В пиктограммах нет текста
  - В наборах пиктограмм пиктограммы одного значения, но разных размеров используют одни и те же особенности и/или сюжет

# Контрольный список интерфейса



## Окна

- На растягивающихся окнах есть индикатор растягиваемости
- Заголовки окон соответствуют названиям элементов, при помощи которых окна были вызваны. Если окно вызывается элементом, не имеющим явного названия, в заголовке окна отражается название экранной формы
- Тип окна (модальное, немодальное, возможность минимизации/максимизации) был выбран осознанно, в соответствии с задачами пользователей
- В диалоговых окнах отсутствуют меню или инструментальные панели
- Кнопки «Применить» используются только в окнах-палитрах (вместо кнопок «ОК»)

# Контрольный список интерфейса



## Окна

### ■ Строка статуса

- В строке статуса выводится только информация о текущем состоянии системы и кнопки (не выглядящие кнопками) для функций, предназначенных только опытным пользователям
- Индикаторы выполнения выводятся в строке статуса. Исключение: окна-мастера, в них индикаторы выполнения можно выводить внутри самих окон

### ■ Меню

- Первая буква в названии пунктов меню - заглавная
- Все пункты меню первого уровня активизируют раскрывающиеся меню
- Элементы, открывающие вложенные меню, выглядят иначе, чем терминальные элементы. Например, используется символ троеточия для обозначения пунктов меню, активизирующих вложенные окна
- Используются не более двух подуровней меню
- Если в меню есть пиктограммы, ими снабжены только самые частотные элементы
- Помещайте на панель инструментов часто используемые команды меню

# Контрольный список интерфейса



## Окна

### ■ Контекстные меню

- На всех объектах, видимых в интерфейсе, есть специфичное для каждого объекта контекстное меню
- В контекстных меню не более 10 элементов
- В контекстных меню элементы отсортированы по убыванию частоты их использования
- Все элементы контекстных меню присутствуют и в других фрагментах интерфейса; нет команд, вызываемых только из контекстных меню

### ■ Формы ввода

- Во всех формах, служащих для сбора информации, есть пункты «Другое» и «Не применимо» или подобный
- Все поля, обязательные для заполнения, помечены, и есть соответствующее пояснение
- Во всех формах, служащих для сбора информации, есть описание целей сбора данных, объясняется, что с этими данными будет сделано и что не будет

# Контрольный список интерфейса



## Окна

- **Структура интерфейсных форм**
  - В группах интерактивных элементов (поля форм, элементы меню и т. п.) этих элементов не больше семи
  - Кнопка «Отмена» всегда самая правая
  - Многостраничные формы имеют указание на то, что они многостраничные; пользователь всегда видит количество оставшихся экранов (пример: «Экран x из y»)
  - Если в форме есть несколько кнопок, одна является кнопкой по умолчанию. Если кнопка в форме только одна, она не может быть кнопкой по умолчанию. Опасные для пользователя кнопки не являются кнопками по умолчанию
  - Если в окне есть свободное место, наиболее частотная кнопка больше остальных
  - Кнопки находятся в секции, на которую они оказывают непосредственное воздействие
  - Кнопки, управляющие окном, расположены либо снизу в ряд, либо справа в колонку

# Контрольный список интерфейса

## Окна

- **Структура интерфейсных форм (продолжение)**
  - Кнопки, относящиеся ко всему блоку вкладок, расположены за пределами блока
  - Если окно или вкладка имеет автоматически пополняемое содержимое, например, в нем перечислены входящие сообщения, в названии элемента интерфейса, который открывает окно или вкладку, выводится число объектов в этом окне и отдельно число новых объектов
  - Пункты меню и кнопки, инициирующие другие действия пользователя, обозначены в конце многоточием (...). Примеры: элемент «Сохранить как...» требует многоточия, т.к. пользователь должен выбрать название файла, а элемент «О программе» многоточия не требует, т.к. на открываемом окне нет самостоятельных интерфейсных элементов
  - Подписи к интерфейсным элементам размещены единообразно
  - Недоступные в данный момент интерфейсные элементы заблокированы, а не скрыты

# Контрольный список интерфейса

## Окна

### ■ Текст

- На все главные интерфейсные элементы повешены всплывающие подсказки, текст которых отражает результат использования этих элементов
- В интерфейсе отсутствуют жаргонизмы
- В интерфейсе отсутствуют отрицательные формулировки (например, чекбокс «Не показывать примечания» неприемлем, взамен него нужно выводить чекбокс «Показывать примечания»)
- Ни один элемент не называется по-разному в разных местах (интерфейсный глоссарий не просто сделан в явной форме, но и выверен)
- В тексте всех подтверждений дается наименование объекта, над которым совершается подтверждаемое действие
- Для улучшения удобочитаемости длинные числа разбиваются неразрывным пробелом по три цифры: 1 234 567

# Контрольный список интерфейса

## Окна

### ■ Текст (продолжение)

- Каждый элемент списка содержит на конце точку или начинается с прописной буквы по следующему правилу: «Текст всех элементов начинается со строчной буквы. Все элементы оканчиваются по последней букве слова без каких-либо знаков препинания, кроме последнего, который оканчивается точкой. Исключение: если хоть один элемент списка содержит более одного предложения, все элементы начинаются с заглавной буквы и заканчиваются точкой.»
- Любому списку предшествует, по меньшей мере, один абзац текста.
- В таблицах все столбцы с цифрами выравниваются по правому краю.
- Точка в конце фразы отсутствует в заголовке (если он отделен от текста), в конце подписи под рисунком и в таблице.
- Подписи к интерфейсным элементам начинаются с прописной буквы и заканчиваются двоеточием





# Контрольный список по распространенным ошибкам

## Scroll

Частая ошибка – скроллинг элемента TreeControl

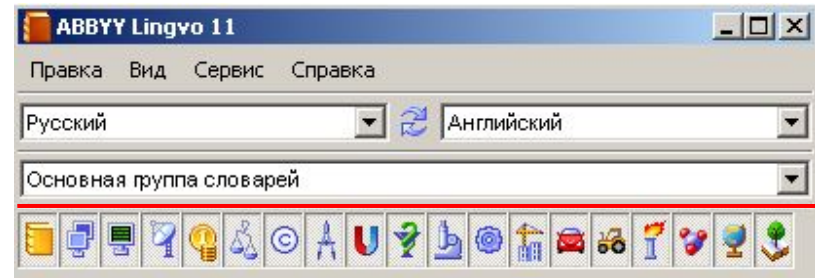
### ■ Навигация

- «Где я?» и «Куда мне отсюда идти?»

Частая ошибка – нет «bread crumbs» («хлебные крошки») и/или не выделен текущий пункт меню

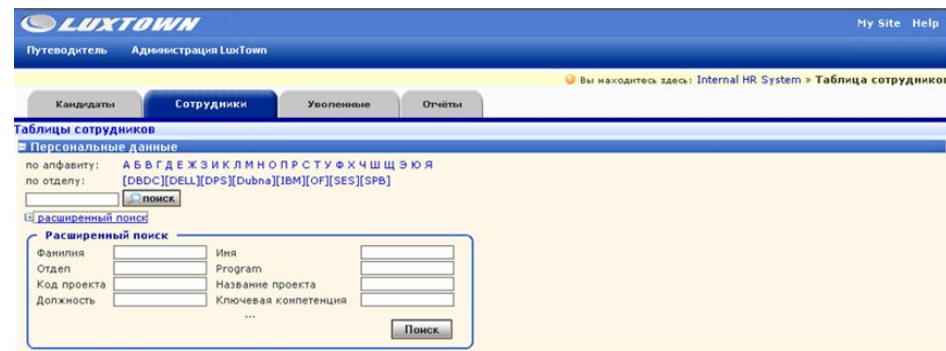
- Количество кнопок (7 - 10), разбивка на группы (информация и инструменты)

*Пример большого количества кнопок - ABBYY Lingvo, пример разбиения номера телефона на блоки*



### ■ Текст

- Контрастность текста и фона
- Наличие полей, отступов
- Правила набора текстов
  - &nbsp; (неразрывные пробелы)
  - Текстовые выделения
  - Кавычки



# Контрольный список по распространенным ошибкам

## ■ Правила набора текстов

- &nbsp;
- И.&nbsp;И.&nbsp;Иванов
- Г-н&nbsp;Иванов
- Г.&nbsp;Москва
- 200&nbsp;кг
- ...



Создать элемент   Изменить элемент   Удалить элемент	
Полное имя	Петров Иван Иванович
Фамилия	Петров
Имя	Иван
Отчество	Иванович
Фамилия И.О.	Петров И. И.
Дата рождения	7/30/1964
Пол	Мужской
Местонахождение	Российский
Организация	Государственная дума
Должность	Спикер
Адрес	г. Москва

## ■ Текстовые выделения

- Предпочтительнее использовать *курсив* для выделения слов, а не **полужирный** шрифт.
- Знаки препинания вокруг выделенного фрагмента должны быть набраны основным начертанием (исключение - кавычки).

## ■ Кавычки

- **RU:** «Французские» (внутри - „немецкие“)
- **ENG:** “English” (внутри: "... '...' ...")

# Работа с респондентом (1/9)



## ■ Кто проводит?

Может проводиться группой тестирования компании-разработчика с обязательным привлечением сторонних пользователей.

## ■ Виды тестирования

- «Что здесь?» тестирование
- Выполнение заданий
- Количественные и качественные методы

## ■ Методы

- Основные (в основном организационные):
  - **пассивное наблюдение за выполнением тестовых заданий (метод теневого наблюдения)** - предназначен для поиска проблемных фрагментов и для сбора количественных данных
  - **поток сознания (метод «мышления вслух», метод совместного исследования)** – то же, что пассивное наблюдение, респондента просят давать комментарии, которые позже анализируются
  - **активное вмешательство (метод вопросов и ответов)** – метод близкий к фокусированному интервью – предназначен для сбора качественных данных
- Дополнительные (в основном психологические):
  - **слежение за движениями значков и др.**

## ■ Измерения

- **Основной принцип – «НЕТ количеству!».** В основном, usability-тестирование должно быть направлено на сбор качественных данных (понять, что именно плохо и как это исправить)
- **Что именно измерять?**
  - Основными с точки зрения эффективности метриками являются:
    - **скорость работы пользователя**
    - **скорость обучения**
    - **количество человеческих ошибок**
  - А также:
    - **субъективная удовлетворенность пользователя**
    - **сохранение навыков работы с системой**

# Работа с респондентом (3/9)

- **Организация usability-тестирования с пользователем**

- **Кто?**

- Кто проводит тестирование и кто наблюдает за тестированием

- **Где?**

- Место проведения тестирования

- **Что и Когда?**

- Таблица, что и когда тестировать

- **Как?**

- Как проводить тестирование и как интерпретировать его результаты

## КТО И ГДЕ?



Субъект тестирования (А) сидит напротив монитора (В). Ассистент (С) говорит субъекту (А), что делать, и задает ему вопросы. Видекамера (D), работающая от беличьего колеса (Е), направлена на монитор (В) и фиксирует то, что видит субъект

В это время по кабелю (F) сигнал от видекамеры передается на телевизор (G), находящийся в соседней комнате, где находятся наблюдатели (H) из группы разработчиков

# Работа с респондентом (5/9)

**ЧТО**

**И**

**КОГДА?**

Определи-тель	Планирова-ние	Наброс-ки	Разработка страниц	Прототип	Первая рабочая версия	Тестиро-вание коллег
Что тестировать	Сайты-конкуренты	Набросок начальной страницы, названия сервисов и категорий верхнего уровня	Начальная страница, шаблоны для страниц второго уровня. Шаблоны страниц основного содержания	Все, что уже функционирует	Все, что уже функционирует	Любая страница
Формат	«Живой сайт»	Бумага	Бумага	HTML-прототип	«Живой сайт»	HTML-страница
Как тестировать	Что очень нравится пользователям? Насколько это полезно в их жизни? Что работает хорошо? Насколько трудно выполнять задания?	Понимают ли пользователи, в чем идея и назначение сайта? Это то, что им нужно?	Понимают ли пользователи, в чем идея и назначение сайта? Понятна ли им навигация сайта? Легко ли найти на сайте то, что нужно пользователям?	Пользователи все еще понимают, в чем идея и назначение сайта? Удастся ли пользователю выполнять задания?	Пользователи все еще понимают, в чем идея и назначение сайта? Удастся ли пользователю выполнять задания?	Удастся ли пользователю выполнять задания?
Длительность теста	1 час	15-20 мин.	15-20 мин.	45-60 мин.	1 час	5 мин на каждую страницу
Количество тестов	1	1-3	1-3	1-3	1-3	1 на каждую страницу
ИТОГО: 13 тестов x 3 пользователя x 1 тест x \$100 на пользователя = \$3 900						





# Работа с респондентом (7/9)



- **Ввод респондента в задачу:** Объяснить правила и цель тестирования; важность участия его – респондента в тестировании; предупредить, что все проблемы - это проблемы системы; предупредить, что вынуждены записывать все его действия
- **Выяснить у респондента его ожидания от системы:** выслушать, но не принимать ко вниманию – до просмотра интерфейса это в основном его фантазии
- **Тестирование интерфейса**
- **Выяснить, насколько оправдались ожидания респондента после работы с интерфейсом:** показанный ему интерфейс может побудить его сформулировать требования, которых он раньше не осознавал
- **Завершение теста:** задать накопившиеся вопросы; дать заполнить анкеты; поблагодарить и расплатиться за участие; если респондент хорош – спросить, можно ли привлекать его для будущего тестирования
- **Анализ результатов и подготовка отчета**

# Работа с респондентом (8/9)

- **Никогда** не извиняйтесь за несовершенство тестируемой системы
- **Никогда** не говорите «Мы потом это исправим»
- **Никого** не обвиняйте в том, что интерфейс плох («Разработчики, конечно, придурки, и создали нечто несуразное, но мы с вами это щас исправим»)
- **Никогда** не называйте процесс тестирования «пользовательским тестированием» – респондент решит, что тестируют его, и будет бояться. Тестируется интерфейс
- **Никогда** не прерывайте респондента. Даже если он говорит нечто неактуальное, дайте ему полностью выговориться и только тогда задавайте свои вопросы
- **Никогда** не формируйте поведение респондента. Все ваши слова должны быть подчеркнута нейтральными:
  - Не стоит задавать вопросы с единственным вариантом ответа. *Лучше задавать такие вопросы – «Простой или сложный у системы интерфейс?»*
  - Часто респонденты сами задают вам вопросы, стараясь избежать необходимости принимать решение самому. *Лучший ответ – встречный вопрос: Респондент: «Я справляюсь?» – «А как вы сами думаете?»*  
*Респондент: «Я выполнил задание правильно?» - «А как вы сами считаете?»*

# Работа с респондентом (9/9)

- Обычно тестирование на прототипах проводят по **методу активного вмешательства**. Для этой цели подойдет **обычный прототип** – можно объяснить все респонденту. При этом получить какие-либо количественные данные не получится
- Чтобы получить **количественные данные**, нужно создать **тестовый прототип**, на котором респондент может выполнить тестовые задания
- ✓ **Обычный прототип:**
  - на каждом изображении экрана будут представлены все возможные при разных обстоятельствах фрагменты интерфейса
  - вместо реальных данных будут выводиться обобщенные данные с пометками о том, какой у них возможен максимальный и минимальный объем
- ✓ **Нефункциональный тестовый прототип:**
  - интерфейс будет представлен так, как он отображается в любой момент выполнения тестовых заданий. Например: если на каком-либо экране по тестовому заданию производится редактирование данных, то нужно изобразить состояние интерфейса до, во время и после редактирования

**Недостаток:** Высокая трудоемкость – обычно создавать приходится оба прототипа

**Выход:** Создание функциональных тестовых (Clickable/ Codable Prototype) прототипов

# Анализ результатов и подготовка отчета



- Когда начинать анализ
- Как анализировать действия респондента
- Анализ результатов и подготовка отчета

# Когда начинать анализ

- **Анализ во время теста**

**Достоинства:**

- Позволяет сэкономить время на этапе анализа
- Дает наиболее непосредственное впечатление от теста, что позволяет увидеть проблемы, не замечаемые никаким другим способом

**Недостатки:**

- Не позволяет фиксировать более одного эргономического показателя за раз, да и то на практике можно замерять только скорость работы пользователя и число человеческих ошибок (правда, это самые востребованные показатели)
- Возможен только при значительном опыте usability-специалиста

- **Анализ после тестирования** позволяет внимательно и вдумчиво проанализировать материал, независимо от количества и сущности измеряемых показателей. Кроме того, он легко масштабируется на сколько угодно исполнителей

- **Оптимальная стратегия:**

- Начинать анализ во время теста
- Иногда на этом можно ограничиться
- Затем можно проанализировать еще и видео-протоколы тестирования

## Ошибки

- Ошибка может появиться просто по невнимательности, но анализировать надо все. Исключить ситуацию - причина ошибки в ошибке тестового задания
- Если ошибка критическая – приводит к другим ошибкам, то фрагмент должен быть переделан: нужно предпринять шаги по устранению неоднозначности, добавить подсказок и т.п.
- Если ошибка некритическая, то исправлять ее нужно, если понимаете, почему ошибка произошла. В любом случае – ошибку зафиксировать

## Замедления выполнения задания

- Респондент без видимых причин приостановился (пытается понять, что ему нужно делать дальше)
- Вероятно, интерфейс недостаточно самоочевиден или однозначен. Проблему необходимо исправлять

## Комментарии респондентов

Советы и пожелания респондентов не обязательно принимать, т.к. респонденты:

- ✓ *не обладают всей необходимой информацией о системе*
- ✓ *ничего не смыслят в проектировании интерфейсов*
- ✓ *их мотивация существенно отличается от необходимой*

Нужно задуматься, зачем нужно то, что они предлагают – значит, есть потребность

# Анализ результатов

- Количественные данные практически не способны показать *сущность* проблемы, они показывают только *количество проблем*
- Количественные данные могут быть использованы для сравнения старого и нового интерфейсов и указать, где выявлены самые значительные проблемы
- Если при тесте собираются количественные данные, вам необходимо провести тест на всех респондентах, прежде чем вы сможете начать править интерфейс
- Вносить исправления лучше немедленно:
  - Если исправление не сработало, то есть шанс исправить до конца теста
  - Ваши впечатления о тесте будут еще остры, а чем острее впечатления, тем глубже будут исправления



*Отчет служит цели коммуникации с Заказчиком.  
Краткий, не содержащий воды (и в то же время полный ) отчет всегда  
предпочтительнее*

## Структура отчета:

- **Резюме :**
  - Цель – похвалить интерфейс, кратко, что нужно сделать, чтобы его улучшить
  - Указать номер версии продукта
  - Указать, кто и когда проводил тест
- **Основные проблемы (интерфейсные проблемы, проявляющиеся по всему интерфейсу)**
- **Частные проблемы (проблемы, проявляющиеся на отдельных экранах)**
- **Количественные данные (если они собирались)**
- **Приложение 1. Методика эксперимента и условия теста**
- **Приложение 2. Описание тестовых сценариев**
- **Приложение 3. Описание респондентов**

# Представление проблем в отчете



- Все выявленные проблемы нужно описывать в порядке убывания их важности
- Проиллюстрировать проблему собранными количественными данными
- Указать, у каких респондентов проблема проявилась
- Укажите рекомендации, основанные на экспертной оценки (если таковая предусматривается) – см. слайды далее
- Описывая проблему приводите скриншоты, указывая на них проблемные моменты
- При описании соблюдать правила:
  - В отчете использовать официальный язык, без жаргонизмов, фамильярности и разговорного стиля
  - Недопустимы отступления от стандартного глоссария (как правило, это глоссарий Microsoft)
  - Интерфейсные объекты нужно именовать так же, как они именуется в системе, недопустимы ни сокращения, ни синонимы
  - Использовать английские наименования (писать - не «поиск на Яндексe», а «поиск на Yandex.ru»)
  - Проверить орфографию





# Оформление количественных данных

- Для каждой метрики нужно свое представление
- Для каждого типа данных создайте отдельную таблицу, при этом в первую таблицу раздела складировать данные по всем метрикам
- Если можно нарисовать график, рисуйте график – поможет понять, почему результаты именно таковы
- При описании респондента нужно ограничиться именами

*Пример приведения данных по количеству человеческих ошибок:  
В каждой ячейке приведено количество человеческих ошибок (крупные и незначительные отдельно)*

Количество ошибок \ Респонденты	1	2	3	4	5	Среднее
Иван П.	3/4	1/13	1/4	0/4	1/2	1.2/5.4
Петр Б.	2/5	0/4	0	2/12	1/3	1/4.8
Елена С.	3/4	3/9	0/4	1/8	1/2	1.6/5.4
Марина А.	3/5	2/9	2/1	4/19	2/4	2.6/7.6
Среднее	2.8/4.5	1.5/8.8	0.8/2.3	1.8/10.8	1.3/2.8	

## Легенда:

-  -- задания, от выполнения которых респонденты отказались
-  -- задания, которые респонденты не смогли выполнить полностью
-  -- задания, выполненные неправильно
-  -- задания, которые не удалось выполнить по техническим причинам (сервер лежал)

# Практическое занятие (6)



## Тестирование usability Системы методом контрольного списка интерфейса

### Задание:

#### Выдать слушателям:

1. Распечатку вопросов, которые могут входить в контрольный список интерфейса – «AddChecklist.doc»
2. Помочь слушателям определиться с выбором сайта/ приложения, которое они хотели бы протестировать на usability, например:

<http://www.google.ru/>

<http://www.yandex.ru/>

<http://promis.luxoft.com/default.aspx>

<https://sentinel2.luxoft.com/sen/>

- другое, предложенное самим слушателем и хорошо ему знакомое

# Выполнение практического задания



## Слушатели должны:

1. Выбрать понравившийся им сайт/ приложение, которое они хотели бы протестировать на usability, например:
2. Проанализировать работу приложения/ сайта с точки зрения удовлетворенности потенциального пользователя
3. Определиться с разделами и вопросами контрольного списка, по которым для выбранного сайта/ приложения проводить тестирование
4. Провести usability-тестирование выбранного сайта/ приложения

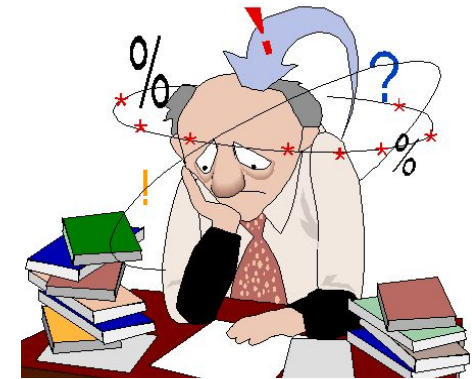
Время выполнения задания (анализ работы приложения/ сайта и составление вопросов контрольного списка) – 20 минут,

Обсуждение выявленных при тестировании результатов – 10 минут

- **Экспертной оценкой** (ЭО) называется определение проблем интерфейса через его оценку профессиональным дизайнером интерфейса или usability-специалистом (**экспертом**)
- Эксперт разглядывает интерфейс и записывает найденные проблемы
- Т.к. рассматривается эксперимент не на реальности, то ЭО не может считаться usability-тестированием
- ЭО дает результаты, в некоторых отношениях более интересные, нежели тестирование
- Если тестирование показывает только плохие фрагменты интерфейса, то с помощью ЭО можно выявить недостаточно хорошие

# Качества эксперта

- Эксперт хорошо знает стандарты на интерфейс, как писанные, так и неписанные. Сами по себе нарушения стандартов не плохи и не хороши, но если они не оправданы, т.е. появились случайно, их лучше устранить
- Эксперт заведомо внимателен и, что гораздо важнее, смотрит на интерфейс свежим взглядом. Это позволяет ему находить множество мелочей, балансирующих между багом и проблемой интерфейса
- Эксперт располагает опытом, позволяющим ему безошибочно определять проблемы, которые проявлялись в его профессиональной деятельности раньше. Если какое-то решение не сработало в прошлом, оно, по видимости, не сработает и в будущем
- Эксперт владеет формальными методами оценки интерфейса, способными даже без тестирования выявить неэффективные решения



# Недостатки экспертной оценки



- Часть проблем, выявленных при ЭО, проблемами, собственно, не является – синдром «мне так показалось». Если плохо проведенное тестирование не показывает всех проблем, то плохо проведенная оценка (что еще хуже) показывает проблемы несуществующие
- ЭО неспособна выявить уникальные проблемы конкретного интерфейса (они уникальны, а значит, эксперт с ними еще не сталкивался)
- Применение формальных методов анализа интерфейса улучшает результат, но также значительно увеличивает стоимость ЭО
- Экспертная оценка обладает обманчивой легкостью, делающей ее особенно опасной в неопытных руках. Выполнять ее может только эксперт и никто другой
- Количество выявленных проблем при ЭО зависит не только от опыта экспертов, но и от их количества, из-за чего заказывать работу приходится у нескольких разных исполнителей, что сложнее с финансовой и организационной стороны



# Виды экспертной оценки

- Эксперт разглядывает интерфейс и записывает найденные проблемы
- Чтобы поиск проблем был результативным, нужен сколько-нибудь формальный подход:
  - Проверка по контрольному списку
  - Эвристическая оценка
  - Мысленная прогонка по интерфейсу

# Проверка по контрольному списку

- Ближе всего к формальному тестированию качества
- Составляется список произвольных требований, после чего интерфейс проверяется на соответствие этим требованиям
- Как правило, в контрольный список попадают требования из стандартов и руководств по проектированию интерфейсов
- Качество проверки по контрольному списку напрямую зависит от качества самого списка

## Недостатки:

- Чем более полон и детален контрольный список, тем дольше производится проверка
- Контрольный список, чтобы быть совершенно надежным, не имеет права содержать сколько-нибудь размытых требований

	оценка				комменты
	1	2	3	4	
Удобство навигации					
В любой момент Вам ясно в какой части системы Вы находитесь.					
В любой момент Вам ясно какое действие производится.					
Если какое либо действие было сделано ошибочно, то система позволяет Вам вернуться и исправить ошибку.					
Все опции системы легко доступны и их поиск для Вас не занимает для Вас много времени.					
Все инструкции системы Вам понятны и просты в исполнении.					
Используемый фон и форматирование экранов позволяют Вам легко ориентироваться по системе.					
Приложение помогает Вам фокусировать внимание на текущей задаче.				x	
Элементы навигации системы имеют понятные для Вас названия и Вы можете легко догадаться о действиях, последующих активизации этих элементов.				x	
Табуляторы и пункты меню расположены в порядке выполняемых функций, важности и частоте использования.				x	



*Например, требование «На растягивающихся окнах есть индикатор растягиваемости» совершенно четкое. А вот требование «Таблицы легко читаются» может быть понятно неоднозначно*

# Эвристическая оценка

- В любой момент времени система показывает, что с ней происходит
- В любой момент пользователь контролирует систему, а не наоборот. Любую команду можно отменить или повторить
- В любой момент времени система выглядит и функционирует единообразным и стандартным способом
- Интерфейс системы препятствует появлению и позволяет исправлять человеческие ошибки
- В любой момент времени интерфейс показывает объекты и команды сам, не требуя от пользователя вспоминать их
- В интерфейсе есть методы ускорения работы, предназначенные для опытных пользователей и не мешающие пользователям неопытным
- Справка доступна в любой момент времени

## Недостатки:

- Оценку должны проводить от трех до пяти экспертов, иначе она не будет сколько-нибудь результативной. Оценку должны проводить эксперты
- Никак не учитывает деятельности пользователей, в лучшем случае она покрывает выполняемые пользователями операции, поэтому не выявляет структурные проблемы интерфейса, что важнее локальных

# Мысленная прогонка по интерфейсу



- Если исходить из того, что интерфейс предназначен для использования функций, можно проверить, как эти функции вызываются и используются
- Если просто проговорить словами, как работают интерфейсы всех функций, становится понятно, какие из них неоправданно подавлены, а какие работают недостаточно хорошо
- Для этого метода также необходим опыт эксперта

## **Недостатки:**

- Метод дает наилучшие результаты, если функции не просто проговариваются, а записываются. Запись требует очень много времени, что не позволяет использовать этот метод всегда и всюду

# Практическое занятие (7)



## Проведение оценки usability Системы методом контрольного списка интерфейса

### Задание:

#### Выдать слушателям:

1. Распечатку альтернативного контрольного списка «Alternate Usability Checklist.xls»
2. Помочь слушателям определиться с выбором сайта/ приложения, которое они хотели бы оценить на привлекательность и визуальный дизайн, например:

<http://market.yandex.ru/>

<http://www.spbstu.ru/index.asp>

<http://www.cstore.ru/>

<http://www.stock-center.ru/>

2. Провести usability-оценку выбранного сайта/ приложения, используя альтернативный контрольный список

# Выполнение практического задания



## Слушатели должны:

1. Выбрать понравившийся им сайт/ приложение, которое они хотели бы оценить на привлекательность и визуальный дизайн
2. Провести usability-оценку выбранного сайта/ приложения, используя альтернативный контрольный список

Время выполнения задания - 20 минут,

Обсуждение выявленных при оценке результатов – 10 минут

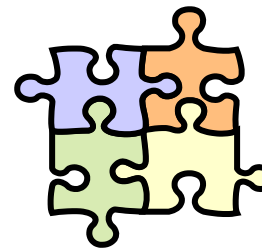
# Путеводитель в мире методов usability-тестирования

## Методы тестирования

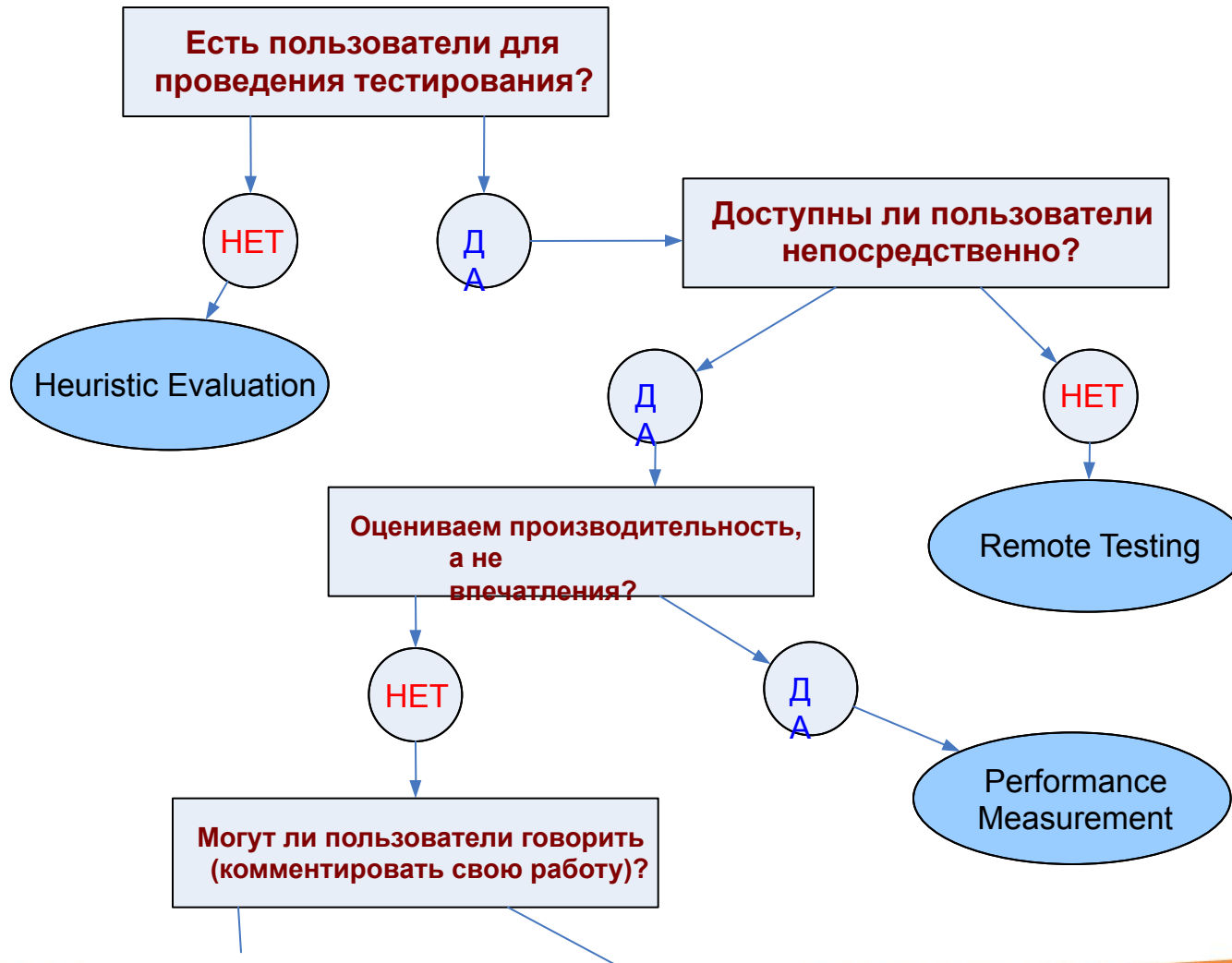
- Heuristic Evaluation (Эвристическая оценка)
- Remote Testing (Удаленное тестирование)
- Co-discovery Learning (Метод открытого обучения)
- Coaching Method (Коуч метод)
- Performance Measurement (Измерение эффективности)
- Question-asking Protocol  
(Тестирование методом активного вмешательства)
- Retrospective Testing (Ретроспективное тестирование)
- Shadowing Method (Метод теневого тестирования)
- Teaching Method (Обучающий метод)
- Thinking Aloud Protocol (Пассивное наблюдение + комментарии респондента)

## Вопросы определителя

- Есть пользователи для проведения тестирования?
- Доступны ли пользователи непосредственно?
- Оцениваем производительность, а не впечатления людей?
- Могут ли пользователи говорить (комментировать свою работу)?
- Может ли другой пользователь комментировать работу пользователя, выполняющего тестовые задания?
- Является ли тестер экспертом в предметной области?



# Определитель (1/3)





# Эвристическая оценка (Heuristic Evaluation)

- Независимые эксперты ищут потенциальные проблемы в UI
- Оценка идет не по сценариям, а по критериям
- После окончания работы всех экспертов проводится анализ результатов

## Критерии:

- Наглядное представление состояния системы
- Связь системы с реальным миром
- Управляемость и свобода действий
- Консистентность и стандарты
- Предотвращение ошибок
- Узнавание против запоминания
- Гибкость и эффективность использования
- Эстетика и минимализм дизайна
- Предвидение, диагностика и восстановление после ошибок
- Справочная система и документация

# Удаленное тестирование (Remote Testing)

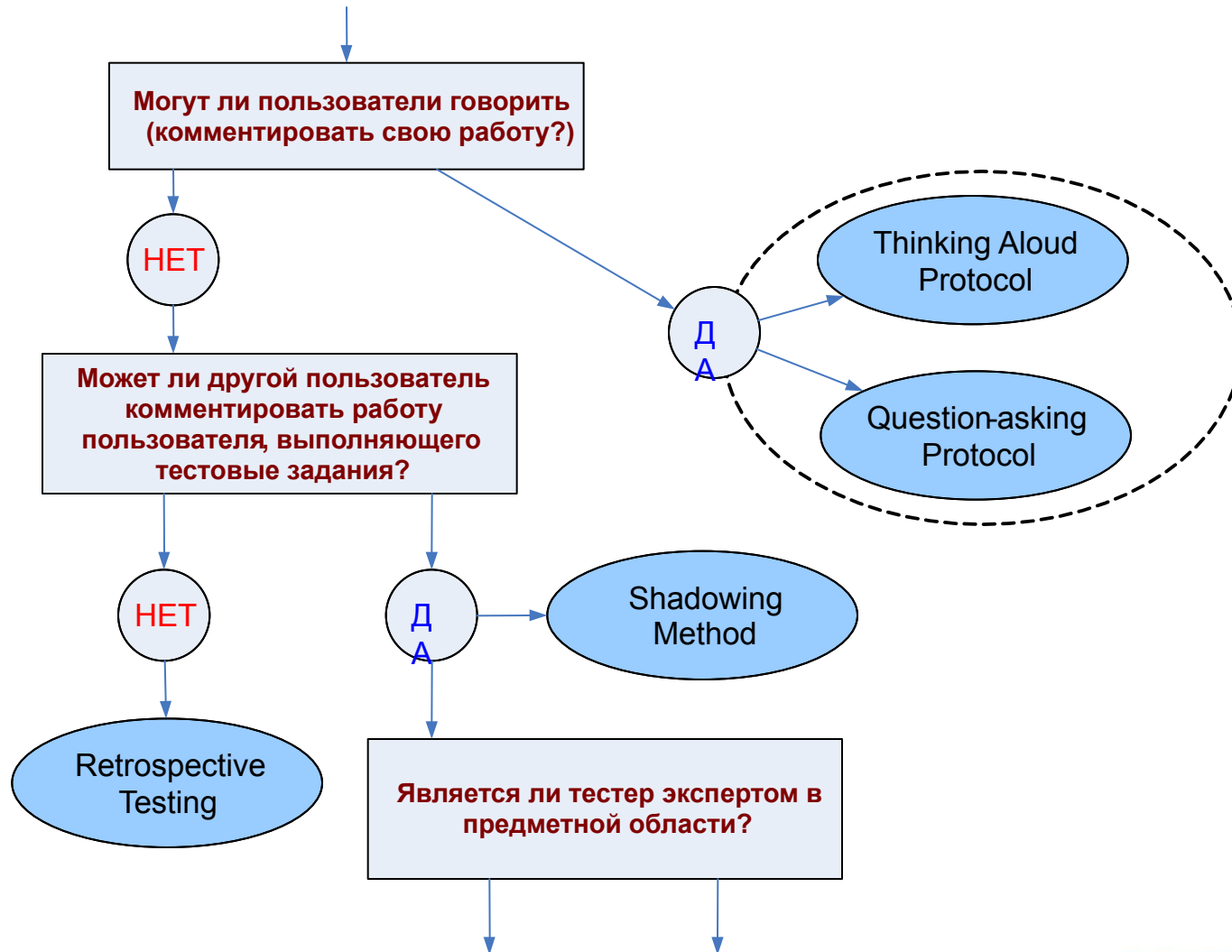


- **Тестирующий и пользователь (респондент) работают отдельно (в разных местах и/или в разное время)**
- **Необходимы технологии:**
  - Журналирование действий пользователя
  - Видеокамеры и голосовая связь или запись
  - Средства удаленного воспроизведения экрана пользователя

# Измерение эффективности использования (Performance Measurement)

- **Определить цели тестирования в терминах критериев Usability**
- **Выбрать наиболее важные показатели**
- **Определить как эти показатели будут оцениваться количественно, например:**
  - Время, затраченное на выполнение задачи
  - Количество задач разного типа, которые можно выполнить в ограниченное время
  - Отношение успешных и ошибочных действий
  - Время, потраченное на восстановление после ошибки
  - Количество пользовательских ошибок
  - и т.д.

# Определитель (2/3)



# Thinking Aloud Protocol and Question-asking Protocol

- **Thinking Aloud Protocol** (Пассивное наблюдение + комментарии респондента)
  - **Critical response** – пользователь должен озвучивать выполнение только заранее определенных задач
  - **Periodic report** – в случае, если задача трудна сама по себе, пользователь рассказывает о её выполнении после окончания
- **Question-asking Protocol** (Тестирование методом активного вмешательства)
  - Помимо того, что пользователи озвучивают свою работу, им задаются заранее подготовленные вопросы о продукте. Например, «Как Вы могли бы послать email из этой программы?»

**Цель этих методов:** понять, насколько пользователь осознанно выполняет свою работу

# Ретроспективное тестирование (Retrospective Testing)



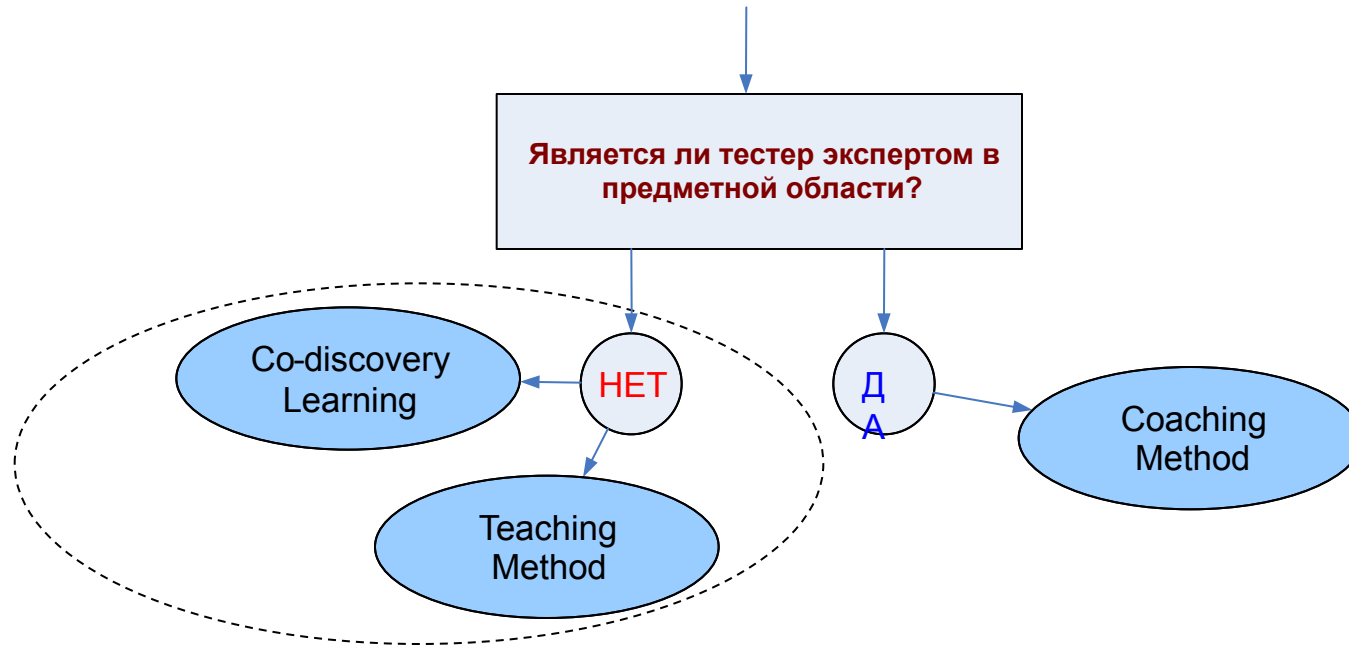
- Пользователи обсуждают с usability-тестером видеозапись тестирования
- Отвечают на вопросы, возникшие у тестера во время тестирования
- Пользователь объясняет, что он делал, чего пытался добиться по видеозаписи

# Метод теневого тестирования (Shadowing Method)



- Один пользователь выполняет задания
- Второй (эксперт в предметной области) комментирует usability-тестеру его работу

# Определитель (3/3)





# Co-discovery Learning and Teaching Method



- **Co-discovery Learning** (Метод открытого обучения)
  - Задания выполняют сразу два пользователя, помогающие друг другу
  - Пользователи должны быть знакомы и иметь предварительный опыт совместной работы
- **Teaching Method** (Обучающий метод)
  - Один пользователь осваивает продукт
  - А потом объясняет новичку, как им пользоваться

**Цель этих методов:** выявить проблемы обучением

# Метод открытого обучения

- Один тестер выступает в роли тренера, обучающего пользователя
- Второй тестер наблюдает за общением первого и пользователя
- Цель – выявить трудности обучения продукту и проблемы документации

## 7. Литература

- Для начинающих
- Дизайн и проектирование интерфейса
- Usability тестирование
- Web usability
- Ссылки

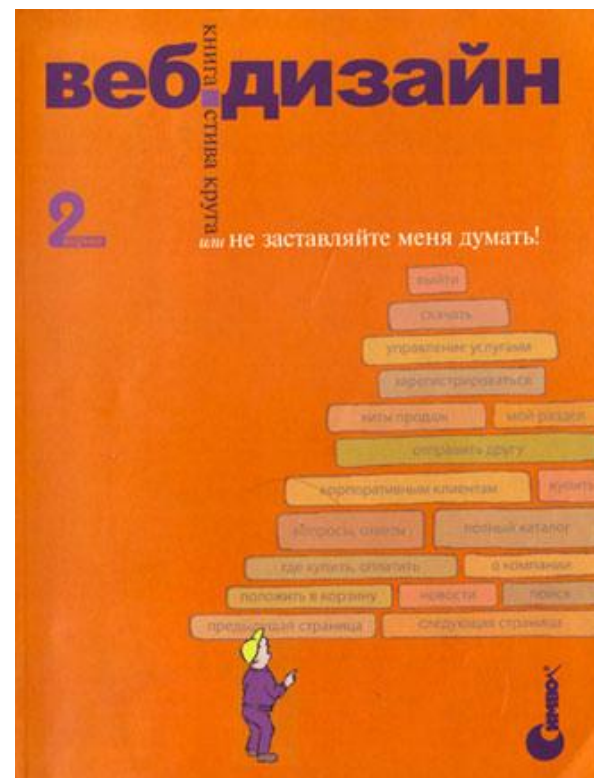
# Для начинающих



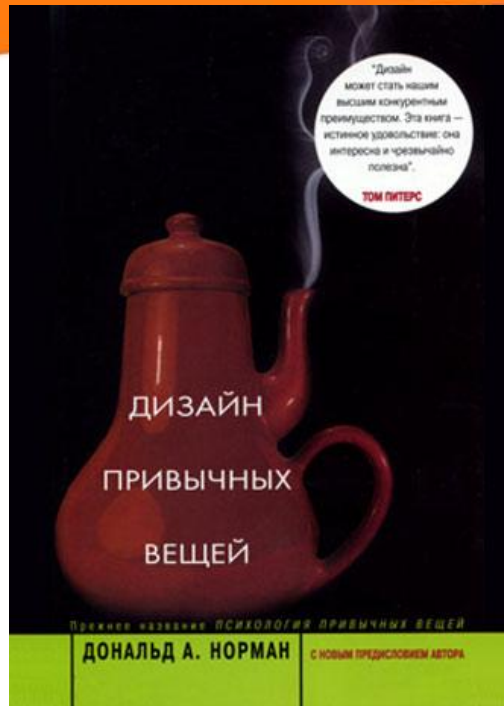
**Влад В. Головач**  
**«Дизайн пользовательского интерфейса. Искусство мыть слона»**

**Стив Круг «Веб-дизайн или не заставляйте меня думать»**

*«Очень простая в изложении книга о том, насколько сложными (и сложными неоправданно) для пользователей могут быть многие варианты веб-дизайна» Влад Головач*



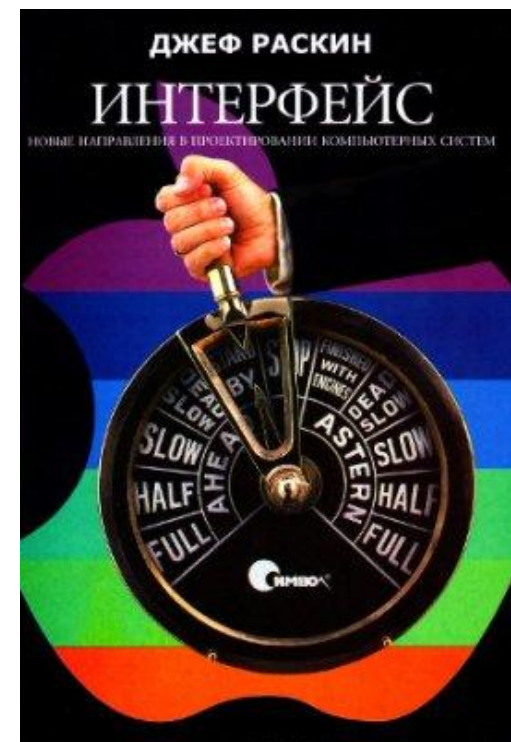
# Дизайн и проектирование интерфейсов



Дональд Норман «Дизайн привычных вещей»  
*«Никогда больше вы не посмотрите прежним взглядом на дверные ручки» Стив Круг*

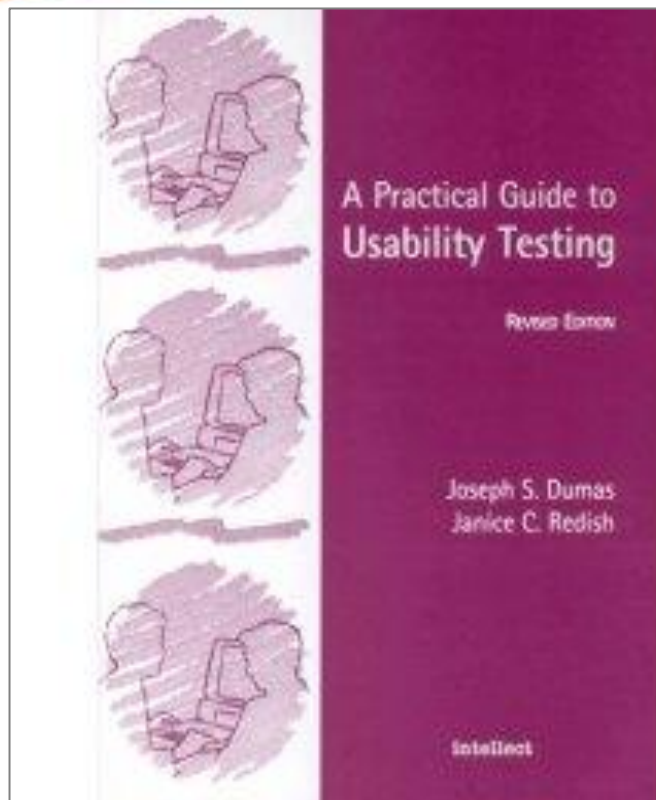


Алан Купер «Об интерфейсе. Основы проектирования взаимодействия»  
*«Пожалуй, самая главная книга о дизайне пользовательских интерфейсов» Влад Головач*

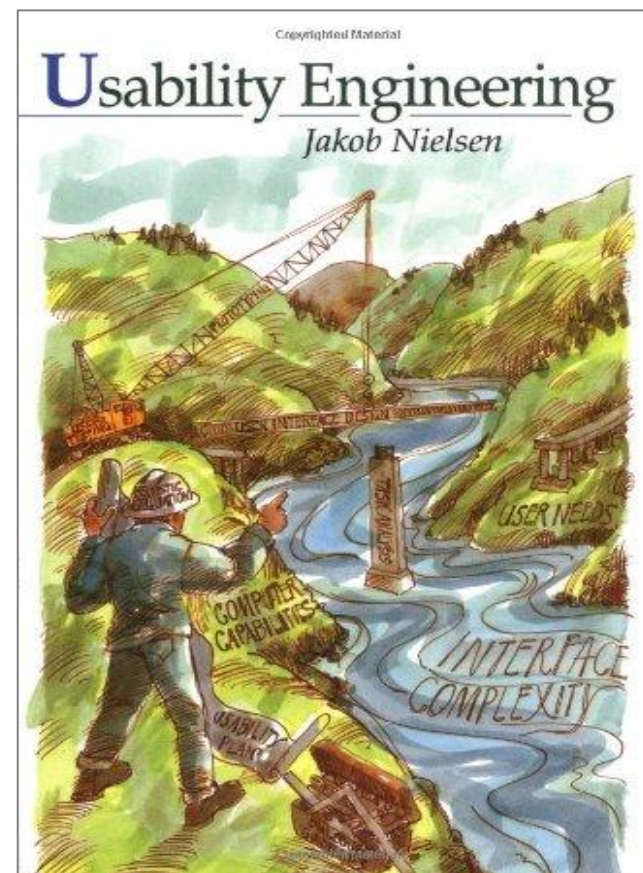


Джеф Раскин  
«Интерфейс»

# Usability тестирование

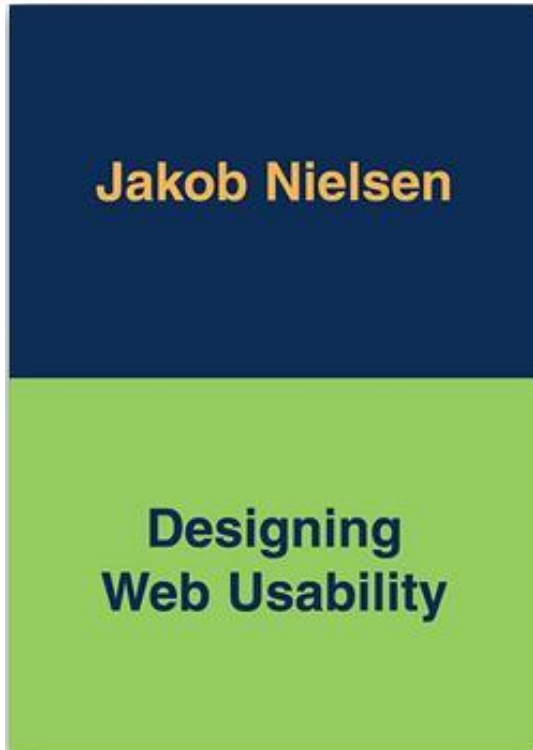


Joseph Dumas, Дженис Редиш  
«A Practical Guide to Usability Testing»

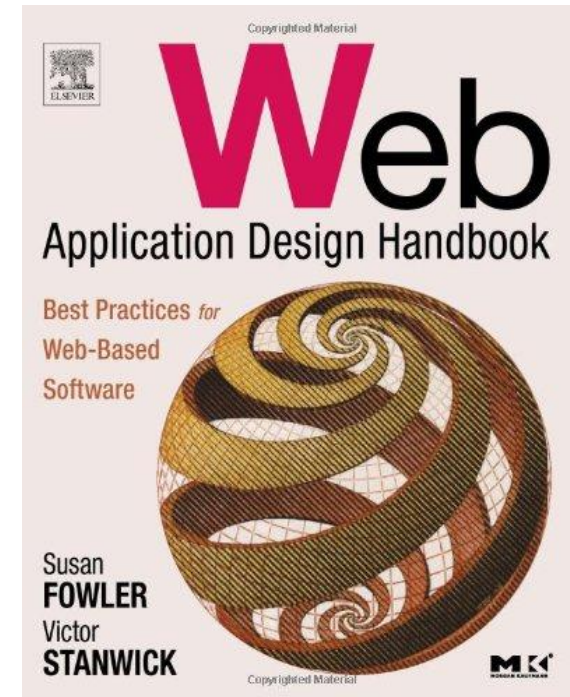


Jakob Nielsen «Usability Engineering»

# Web usability



**Jakob Nielsen**  
**«Designing Web Usability: The Practice of Simplicity»**



**Susan Fowler, Victor Stanwick «Web Application Design Handbook: Best Practices for Web-Based Software»**

- <http://www.usability.ru/>
- <http://www.usability.by/>
- <http://usethics.ru/lib/>
- <http://www.usabilityprofessionals.ru/>
- <http://www.webmascon.com/topics/development/22a.asp>
- <http://www.it-online.ru/projects/usabilitylab/>
- <http://usabilitylab.ru/usability/>
- [http://sigchi.ru/index\\_ru.htm](http://sigchi.ru/index_ru.htm)



# Материалы для практического использования

1. UI Specification. Windows client – «UISpecification\_4TST019.doc» (материал к презентации)
2. Контрольный список интерфейса (презентация, слайды 136-152)
3. Контрольный список интерфейса web-сайта (презентация, слайды 82-89, 135)
4. Контрольный список по дизайну форм (презентация, слайды 67-80), дополнительный контрольный список - «AddChecklist.doc» (материал к презентации)
5. Альтернативный контрольный список – «Alternate Usability Checklist.xls» (материал к презентации)
6. Usability Expert Review – пример отчета по экспертной оценке usability - «LoanSTAR Usability Review.pdf» (материал к презентации)



**Спасибо за внимание!**

**Вопросы?**