

Workshop: Evaluate User Interface

- When to Test
- What to Test
- How to Test

Presented by: Mr. Shimon Rothschild
Email: Shimon@Crims.org
Phone (310) 651-8200



Workshop Plan

In this workshop you will learn:

- Appropriate UI evaluation techniques
- Checklist to evaluate the UI
- Ideas to fix a UI



Workshop Plan

This will not cover:

- Functional testing
- GUI standards
- Industry specific concerns

Main Message



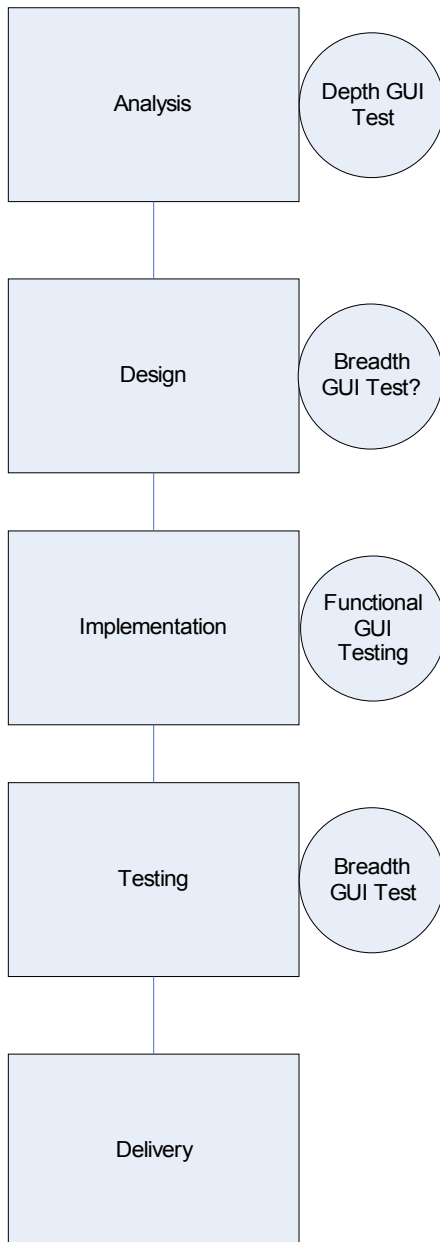
UI testing is a long term commitment to your users. The challenge of UI validation is to do the right tests at the right time with the right people.

Tales From the Trenches



\$2,000 misses \$100,000 and \$20,000 saves ?

When to Test



Section One

- Depth testing (use case validation)
- Functional testing (as specified)
- Breadth testing (look and feel)



Depth Test Prerequisites and Participants

Prerequisites

- Use cases or 'scenarios'
- Wire frames (screen mock ups)

Who

- Use case authors
- Wire frame designers
- 'Users' to test the activity

Depth Test

What

- Use case based
- Focus on primary flow
- Identify errors and continue

How

- Group step through
- Individual then compare notes

When

- Before analysis finalized

Test according to a script

Functional Testing

What

- Test according to the specification
- “Traditional” QA testing

How

- Create test scripts based on specifications

When

- After the application is written

QA can provide more with the check list

Breadth Test

What

- UI specification based
- Domain based
- User testing and UI Expert

How

- Self guided exploration by evaluators
- Use case step through optional








When

- During requirements (Debatable)
- During Beta testing

Free flow navigation

Summary of Types

Comparison of Test Types

Problem Type/ Method	Reoccurring problem	Consistent behavior	Usability Problem	Cost
Depth				
Breadth			?	

Duplicate error rate is 10-30%

Challenges

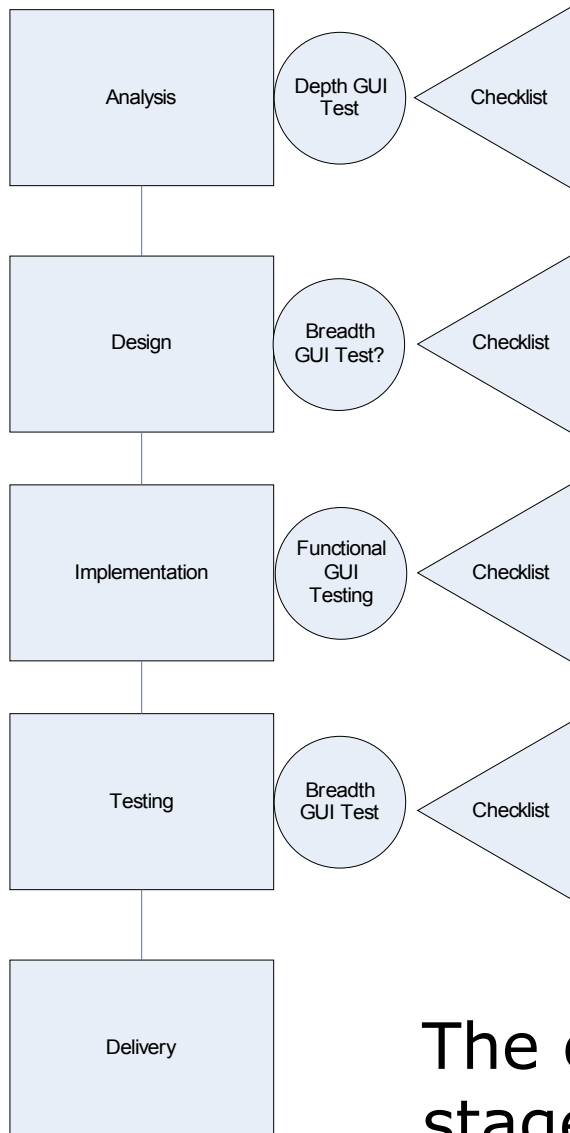
- Reliability of findings
- Effectiveness in finding
- Acceptance by the organization
- Users are not designers and not a definitive source

Applicable to depth and breadth testing

What to Test Checklist

Section Two

- Simple and natural language
- Speak the USER'S language
- Minimize user's memory load
- Consistency
- Feedback
- Clearly marked exits
- Shortcuts
- Precise, useful error messages
- Prevent errors
- Help and documentation



What to Test

The checklist should be applied at every stage in the development process

Main Message



Use this check list in all phases of development and UI testing

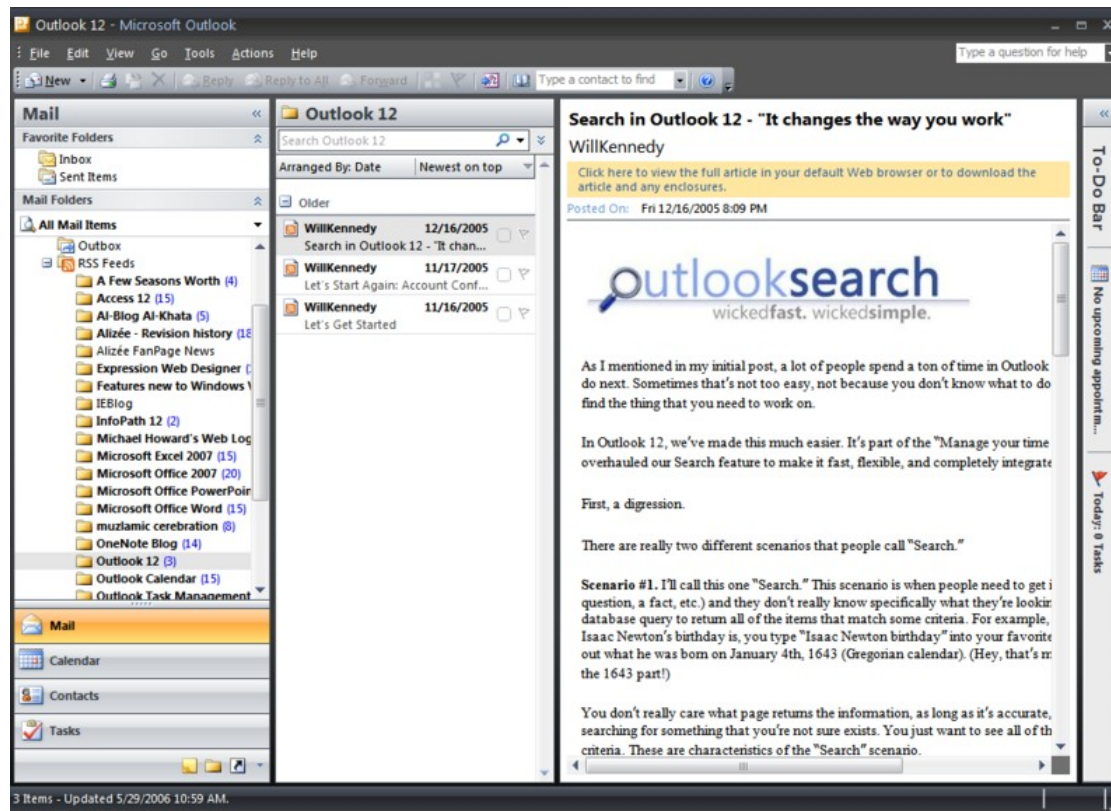
Tales From the Trenches



Hospital's order-entry system -22 ways to accidentally kill a patient

Simple and Natural Language

Text is clear, unambiguous and intuitive



Speak the Users Language

Use language specific to the domain
(example:

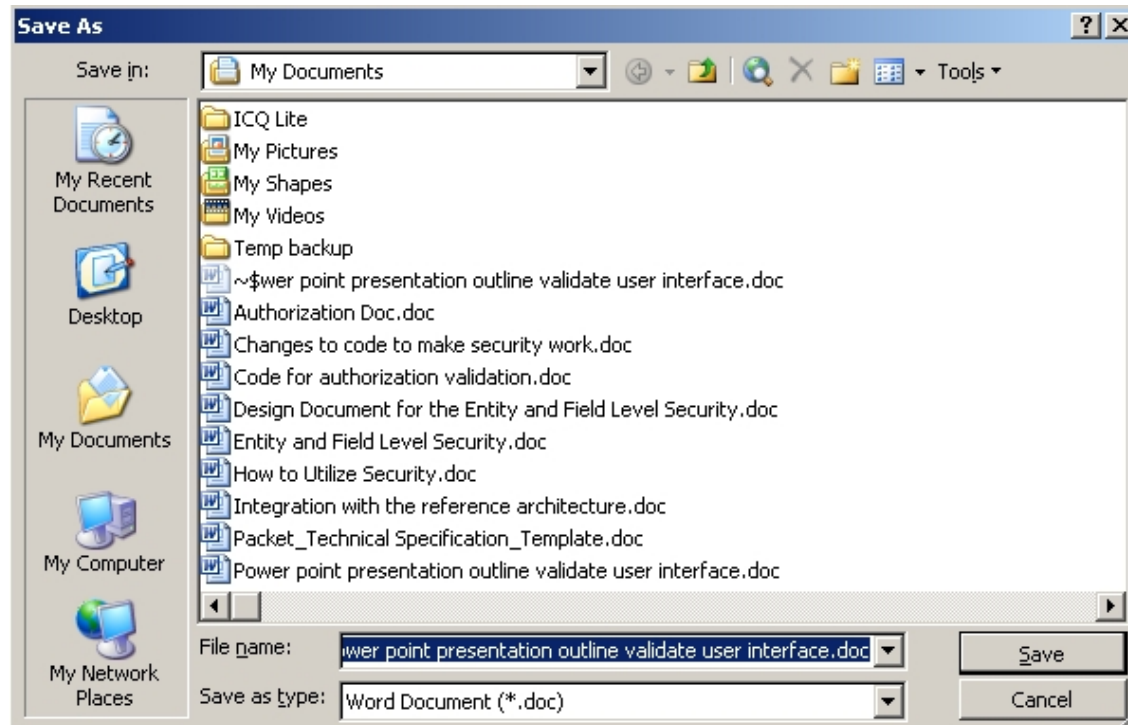
Finance sees: expense **account**

Manager sees: budgeted **activity**

Programmer sees: $\text{cost} = \text{rate} * \text{time};$

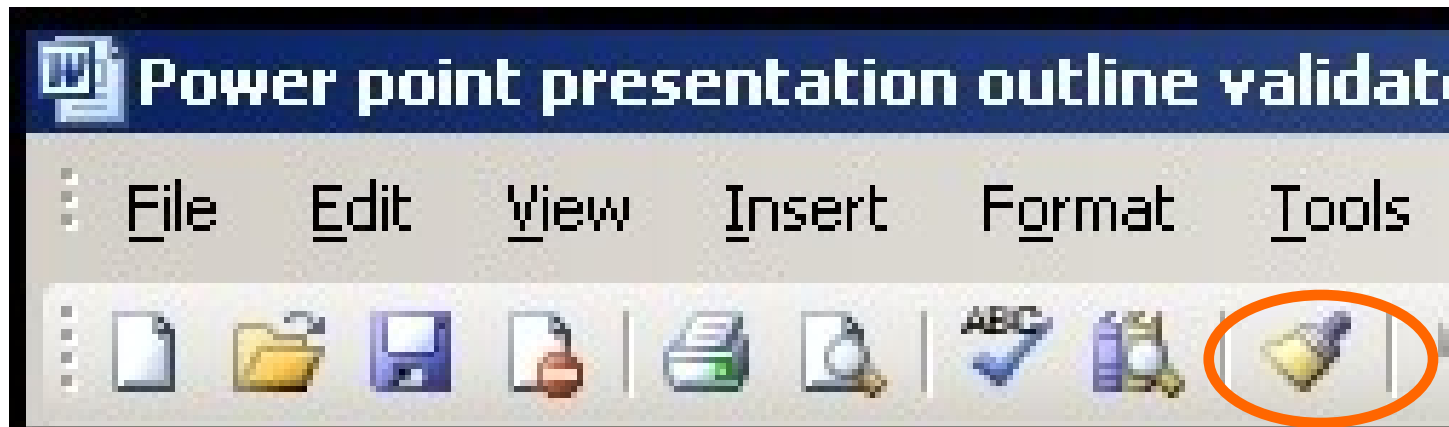
Minimize User's Memory Load

Do not force the user to remember



Consistency

Once a behavior is learned it should be applicable in similar situations.



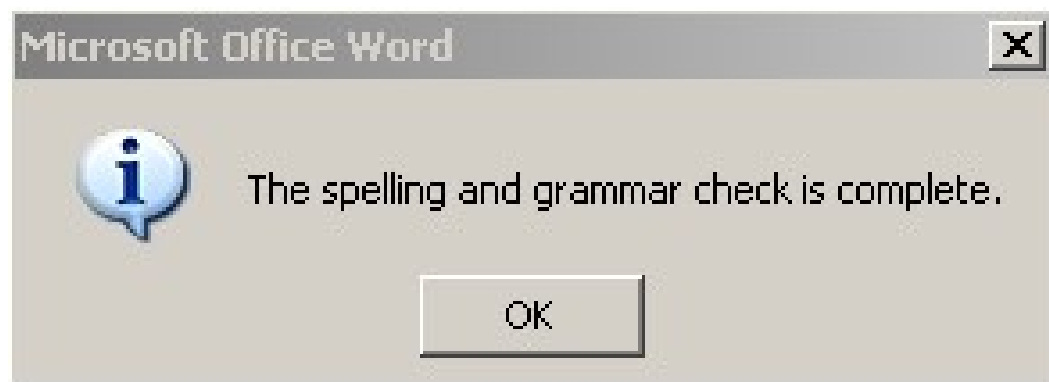
Feedback

Users expect feedback:

Human - human dialog

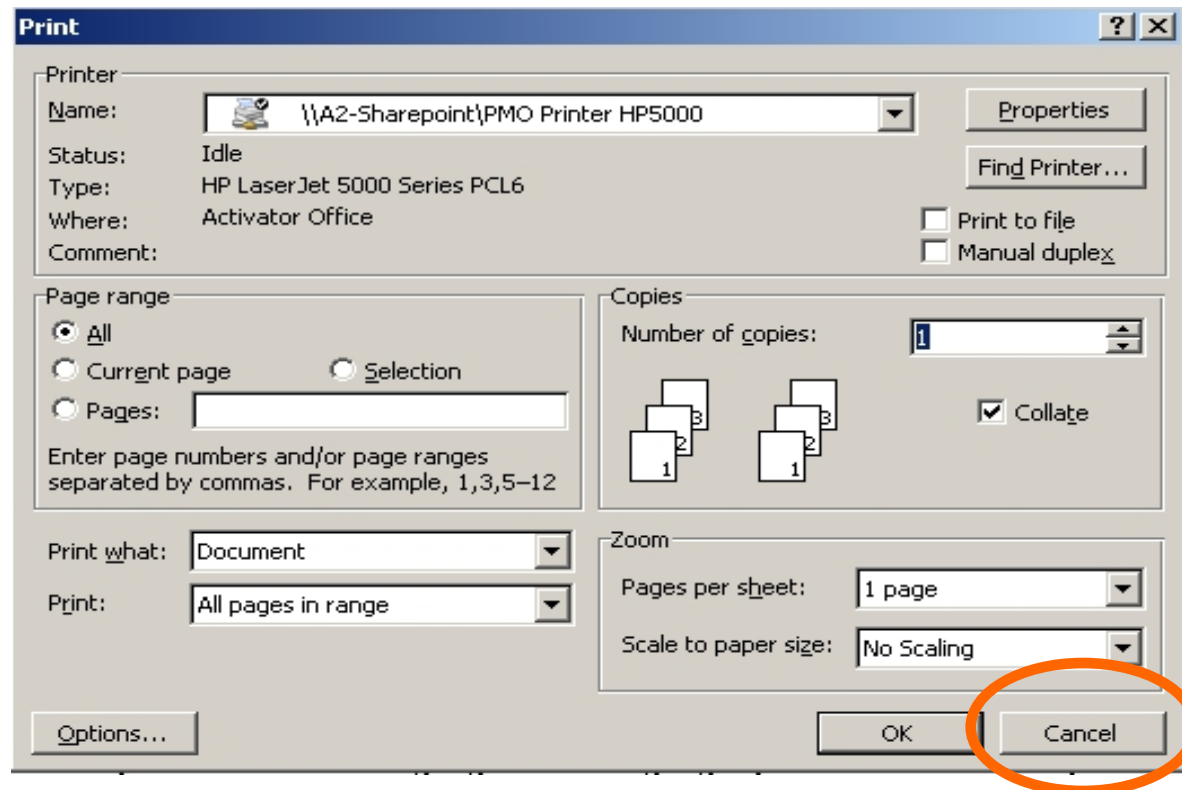
Human - system dialog

Explicit or self evident.



Clearly Marked Exits

Return to a previous condition.

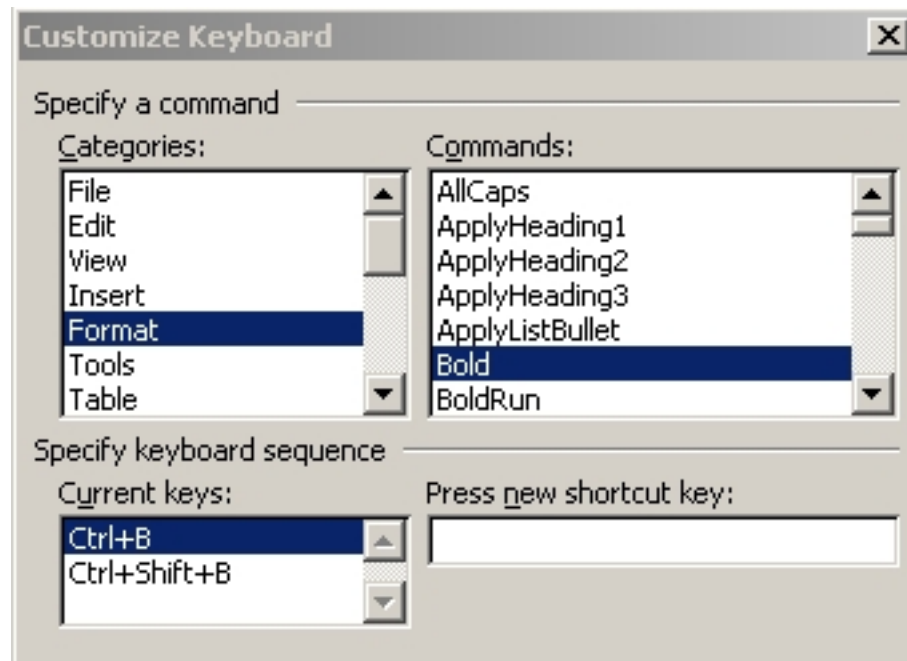


Short Cuts

Two types of users:

- New/ casual users
- Power users

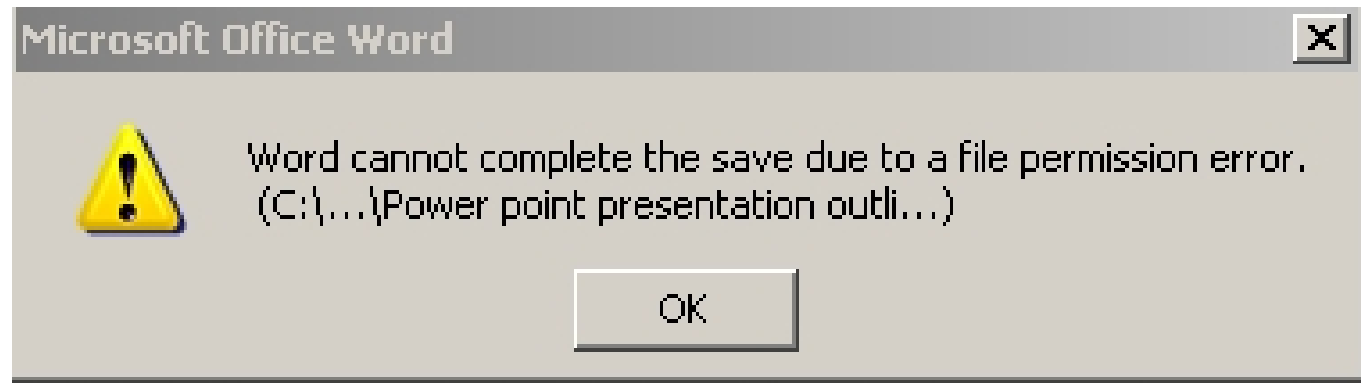
Provide short cuts for the power users



Precise and Constructive Error Messages

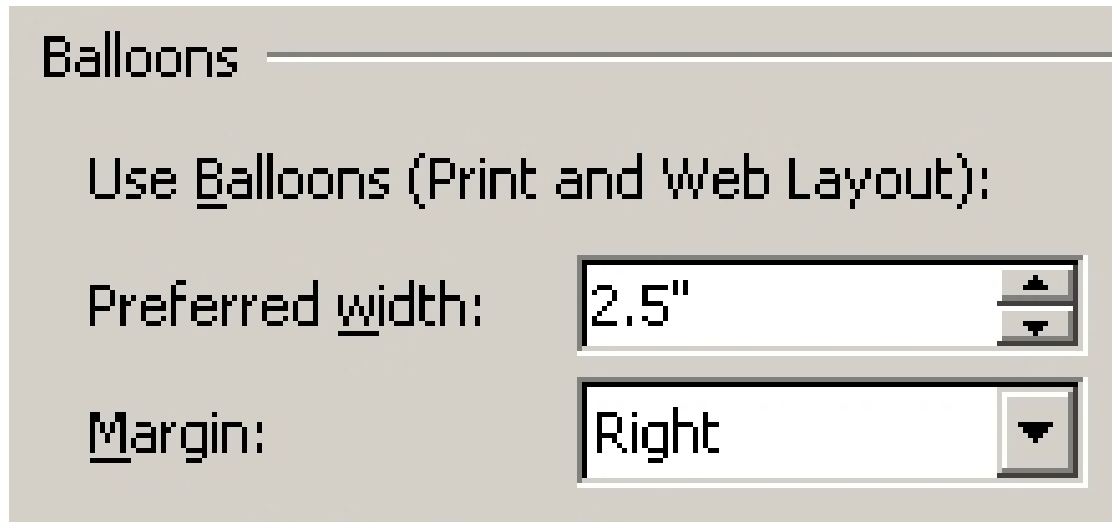
Inform the user why.

Indicate the remedy to complete request.



Prevent Errors

Use the correct component and filters to guide acceptable user input



Balloons

Use Balloons (Print and Web Layout):

Preferred width:

Margin:



Help and Documentation

If documentation is necessary it should be topical to the problem at hand.

Context sensitive help and a useful search capability



How to Test

Section Three

- Testing Techniques
- Evaluating and Scoring
- Recommending Solutions

Main Message



No one test methodology is sufficient. Use multiple evaluation techniques for maximum coverage and to validate results.

Tales From the Trenches



Hitachi broke the rules

Test Techniques

Five Techniques

Depth

Breadth

- Heuristic Evaluation
- Cognitive Walk Through
- Usability Testing
- Guidelines Testing
- Consistency Inspection

Heuristic Evaluation

Individuals independently 'test' the application and come together to share results

Optionally:

- Observer to record and 'help' tester.
- Use case driven
- Multiple iterations to become familiar
- More testers is better 4 is optimal ROI
- Usability specialist to judge results

Cognitive Walk Through

Focus is on how long for user to 'learn' the application.

Use case driven


- Uncover mismatch between user and designer concept of the task
- Individual or group 'walk through'
- Document errors then proceed

Usability Testing

Formal process

Similar to cognitive walk through but
focused on the 6 step process.

4. Planning
5. Kickoff (Define test path)
6. Preparation (Testing)
7. Meeting (Evaluation of tests)
8. Design (Propose solutions)
9. Assess (Evaluate process)



Guidelines and Consistency Testing

Guidelines for behavior across the application.
Consistent behavior for similar actions.

Testing requires a high degree of expertise.

Evaluating Scoring

Considerations

- Severity
- Content
- Scope

Problem impact based on score

Severity

- Impede user progress
- Necessity to fix

Example: Unnecessary dialogs to complete a task: print

6. Click Print button
7. Dialog Print setup (orientation, margins...)
8. Dialog Print preview
9. Dialog Printer selection
10. Dialog Print

Content

- Consistent or conflict with other implementations
- Recurring problem or only until learned

Example: Cancel button sometimes closes dialog and sometimes only clears text boxes

Scope

- Context is task based or independent
- Unique to instance or across multiple instances

Example: Right click mouse button sometimes displays a context menu. Is the problem local or pervasive?



Recommending Solutions

Problem Type

- Hard to achieve objective
- Unclear next action
- Action did not achieve desired effect
- User does not see progress



Achieve Objective

- Eliminate the action
- Prompt user
- Modify task

Example: Create a wizard to guide user

Next Action

- More obvious control
- Use prompt to indicate action
- Utilize 'intuitive' method of discovery

Examples:

7. Fill in form, and want to save data, button should say 'Save', not 'Ok'.
8. Cursor automatically moves to next field to complete

Desired Effect

User selects an action and it does not achieve the desired affect.

- Use terminology user recognizes
- Re-label 'incorrect' options to be obviously the incorrect option

Example: Check box with a "not" condition

Progress



Provide appropriate feedback

The user wants to know that the request has been accomplished.

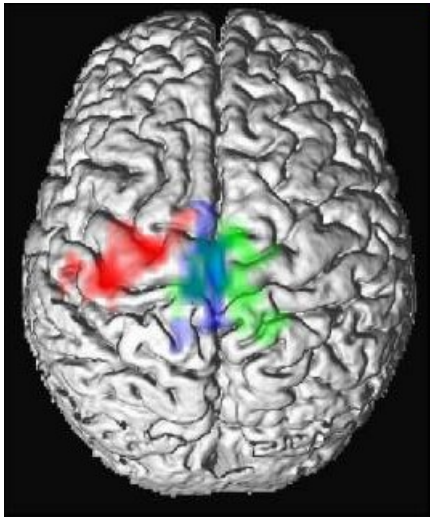
Additional Resources

Human/ Computer Interaction
<http://sigchi.org/>

Microsoft (MSDN Search GUI Design)
[http://msdn.microsoft.com/library/
default.asp?url=/library/en-us/
dnwui/html/iuiguidelines.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwui/html/iuiguidelines.asp)

SAP
<http://www.sapdesignguild.org/index.asp>

User Interface



Using the senses...
– Until direct to the
neural system!

Presented by: Mr. Shimon Rothschild
Email: Shimon@Crims.org
Phone (310) 651-8200