

i213 User Interface Design and Development

Professor Tapan Parikh
School of Information, UC Berkeley

October 14

Formative Evaluation

FORMATIVE vs SUMMATIVE EVALUATION

Formative evaluation - Discover usability problems as part of an iterative design process. Goal is to uncover as many problems as possible.

Summative evaluation - Assess the usability of a prototype, or compare alternatives. Goal is a reliable, statistically valid comparison

THINKING ALOUD

“Having a test subject use the system while continuously thinking aloud”

Useful for formative evaluation

Understand how users view the system by externalizing their thought process

Generates a lot of qualitative data from relatively small number of users

Focus on what the user is concretely doing and saying



EXAMPLE PROMPTS

“Please keep talking.”

“Tell me what you are thinking.”

“Tell me what you are trying to do.”

“Are you looking for something? What?”

“What did you expect to happen just now?”

“What do you mean by that?”

GETTING USERS TO OPEN UP

Thinking aloud can be unnatural and awkward



Requires prompting by the experimenter to ensure that the user continues to externalize their thought process

May slow them down and affect performance

Planning next semester's classes

POINTS TO REMEMBER

Do not make value judgments

User: “This is really confusing here.”

Tester: “Yeah, you’re right. It is.” (BAD)

Tester: “Okay, I’ll make a note of that.” (GOOD)

Video or audio record (with user’s permission), and/or take good notes

Screen captures / Eye tracking

When the user is thinking hard, don’t disturb them with a prompt - wait!

THINK ALOUD VARIANTS

Co-Discovery: Two users work together

- Can spur more conversation
- Needs 2x more users

Retrospective: Think aloud after the fact, reviewing a video recording

- Doesn't disturb the user during the task
- User may forget some thoughts, reactions

Coaching: Expert coach guides user, answering questions

- Identify training, help and documentation needs

HEURISTIC EVALUATION

A cheap and effective way to find usability problems

A small set of expert evaluators “examine the interface and judge its compliance with recognized usability principles”

“Discount” usability testing - find problems earlier and relatively cheaply, without involving real users

WHAT HEURISTICS?

Recommended books provide a number of high-level and low-level design guidelines:

Jakob Nielsen, Usability Engineering

Donald Norman, Design of Everyday Things

Jeff Johnson, GUI Bloopers

Other heuristics can be provided by your own intuition, common sense, user research

We will use Nielsen's list from Usability Engineering

NIELSON'S HEURISTICS?

- Simple and Natural Dialog
- Speak the User's Language
- Minimize User Memory Load
- Consistency
- Feedback
- Clearly Marked Exits
- Shortcuts
- Good Error Messages
- Prevent Errors
- Help and Documentation

SIMPLE AND NATURAL DIALOG

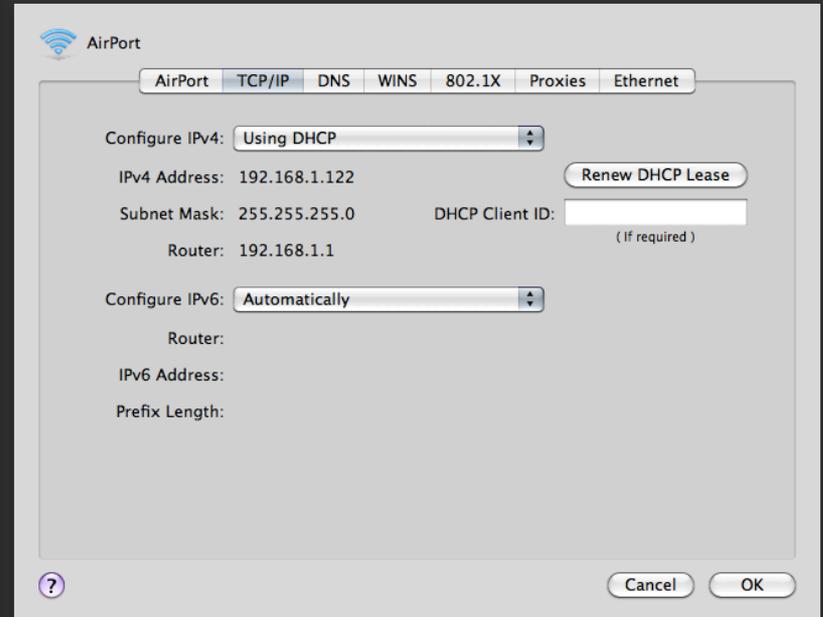
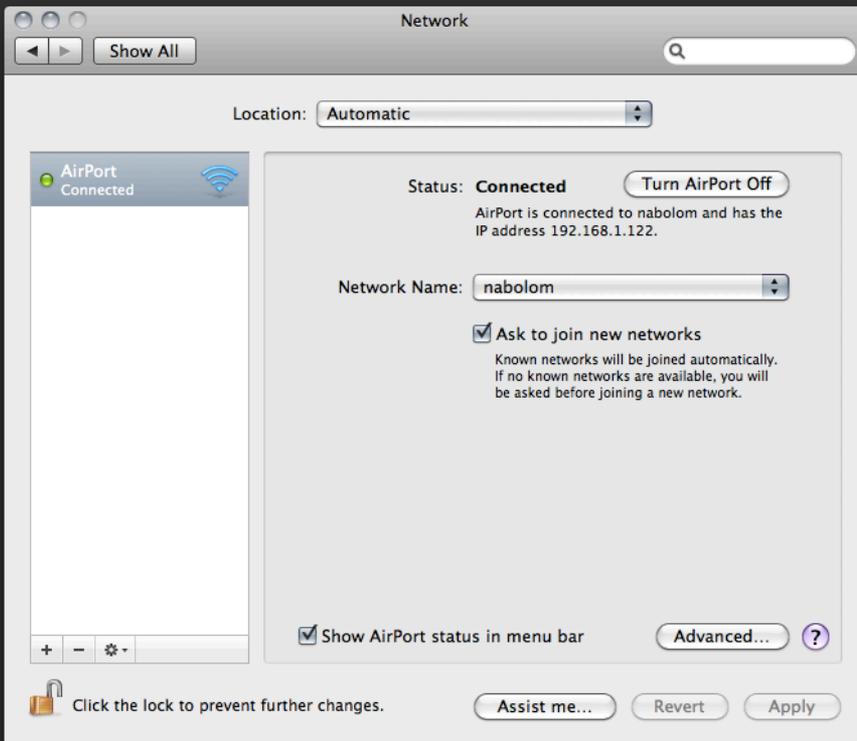
Match the user's task

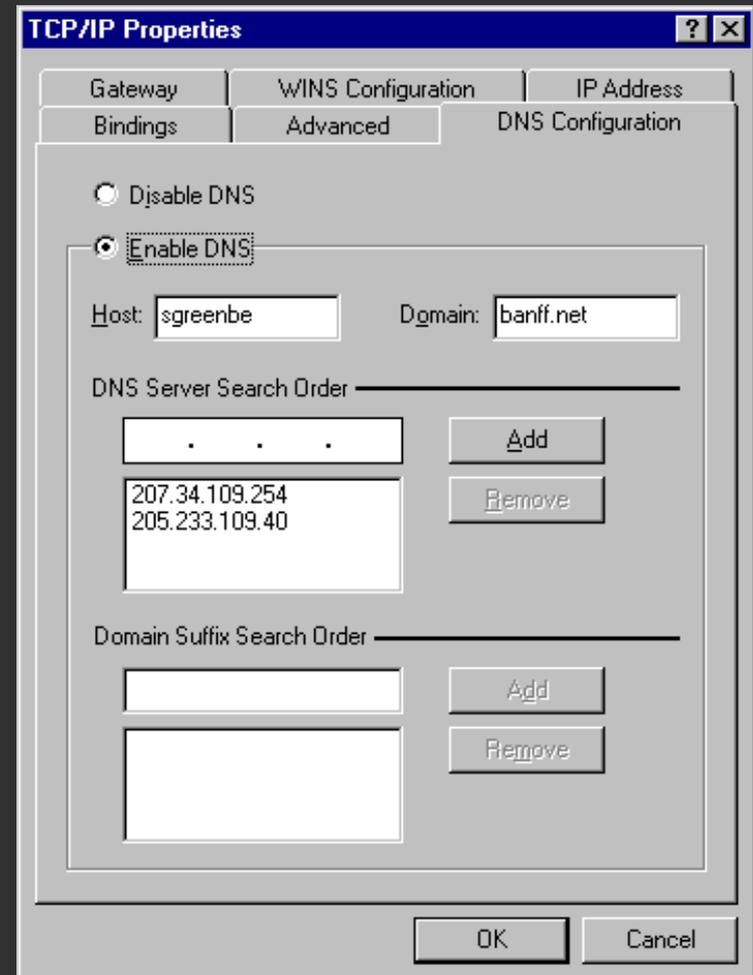
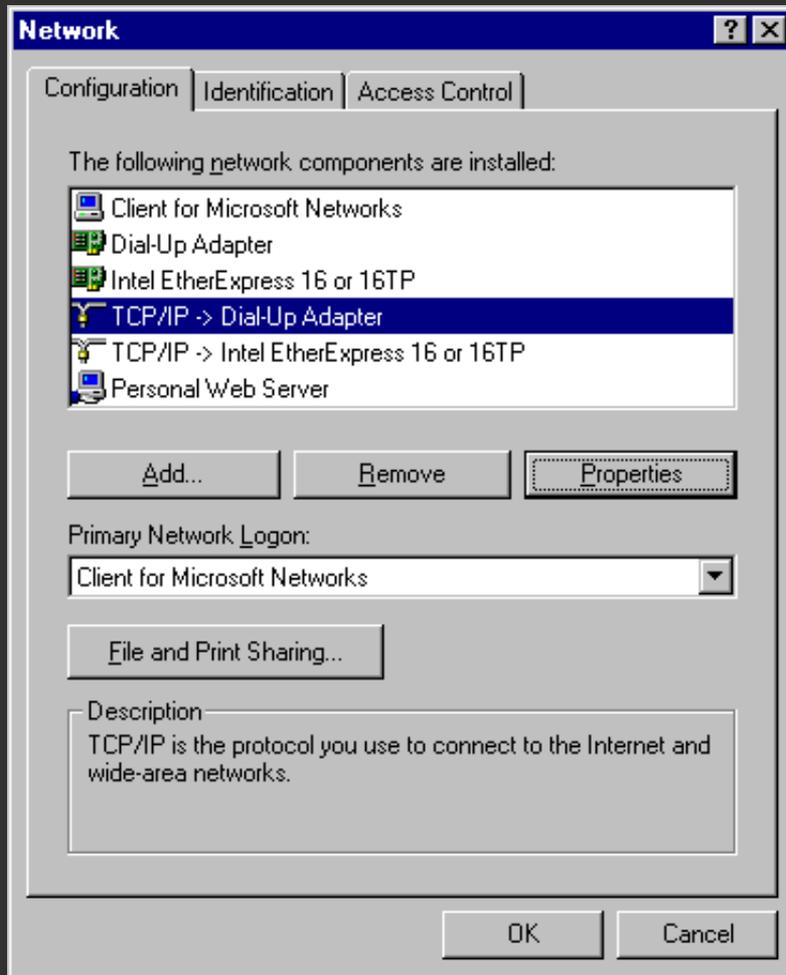
Minimize navigation

Present exactly the information the user needs, when she needs it

Use good graphic design

Less is more





SPEAK THE USER'S LANGUAGE

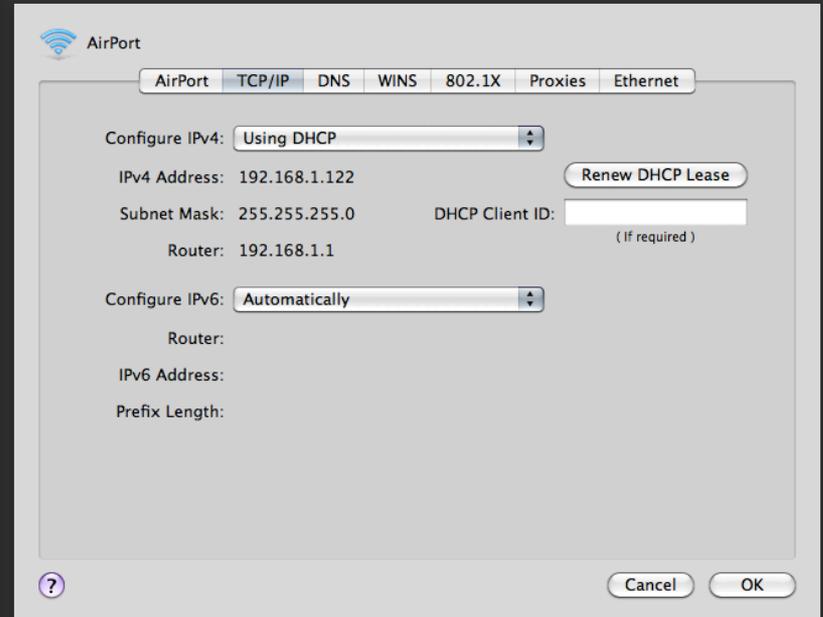
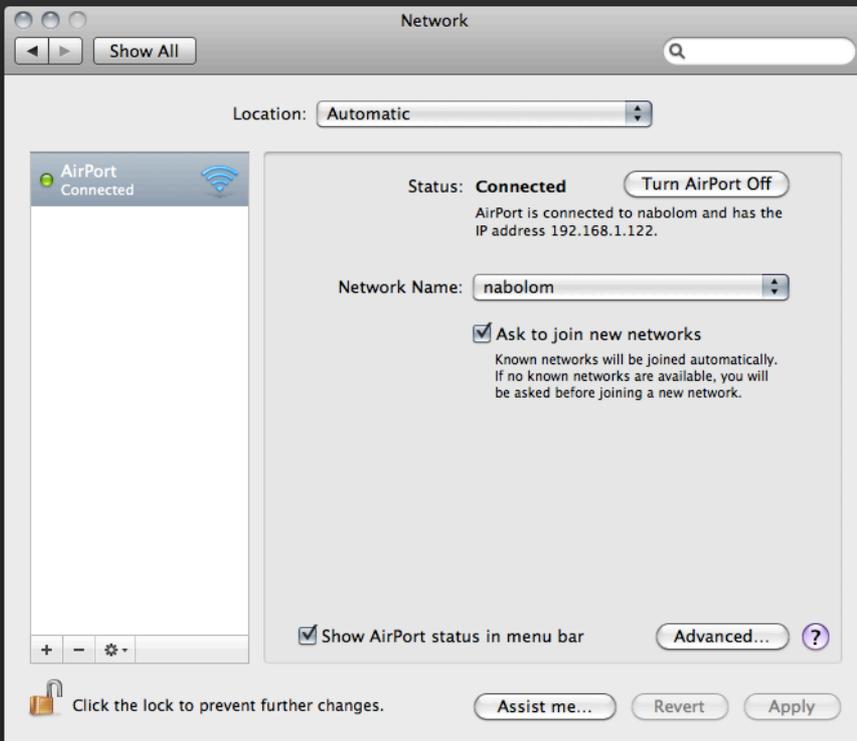
Use the same terms the user would

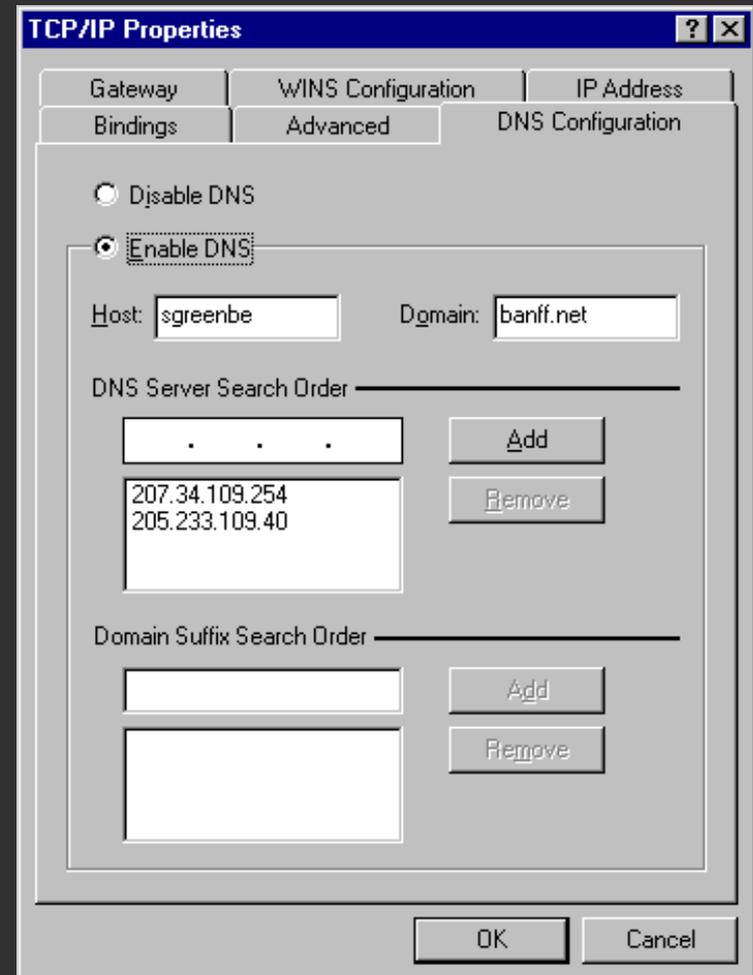
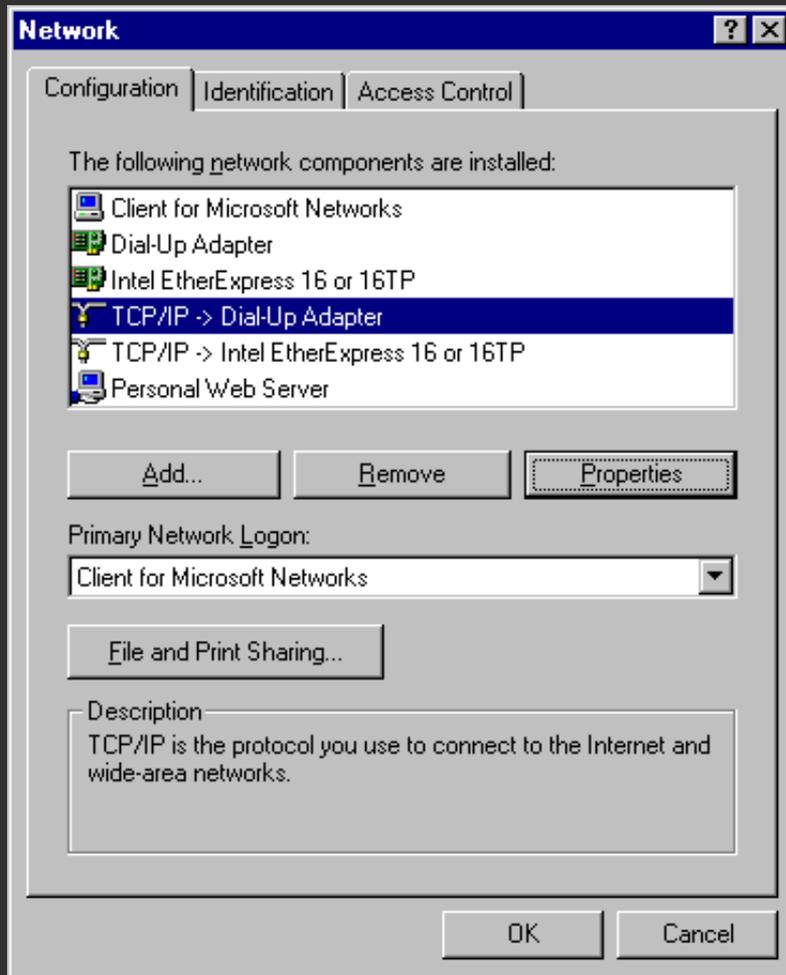
Avoid unusual word meanings

Support synonyms and aliases

Don't impose naming conventions

Understand users and how they view their domain





MINIMIZE USER MEMORY LOAD

Recognize rather than Recall

Edit rather than Enter

Choose rather than Input

Provide a small number of basic commands

Form1

Date:

Month Day Year

May 22 1997

Month Day Year

May 22 1997

Appointment

General Attendees Notes Planner

When

Start: 8:30AM Wed 5 /14 /97

End: 4:30PM Wed 5 /14 /97 All day

Description:

Smart Technology Ser

Where:

May 1997						
S	M	T	W	T	F	S
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

GA Telnet unix.andrew.cmu.edu

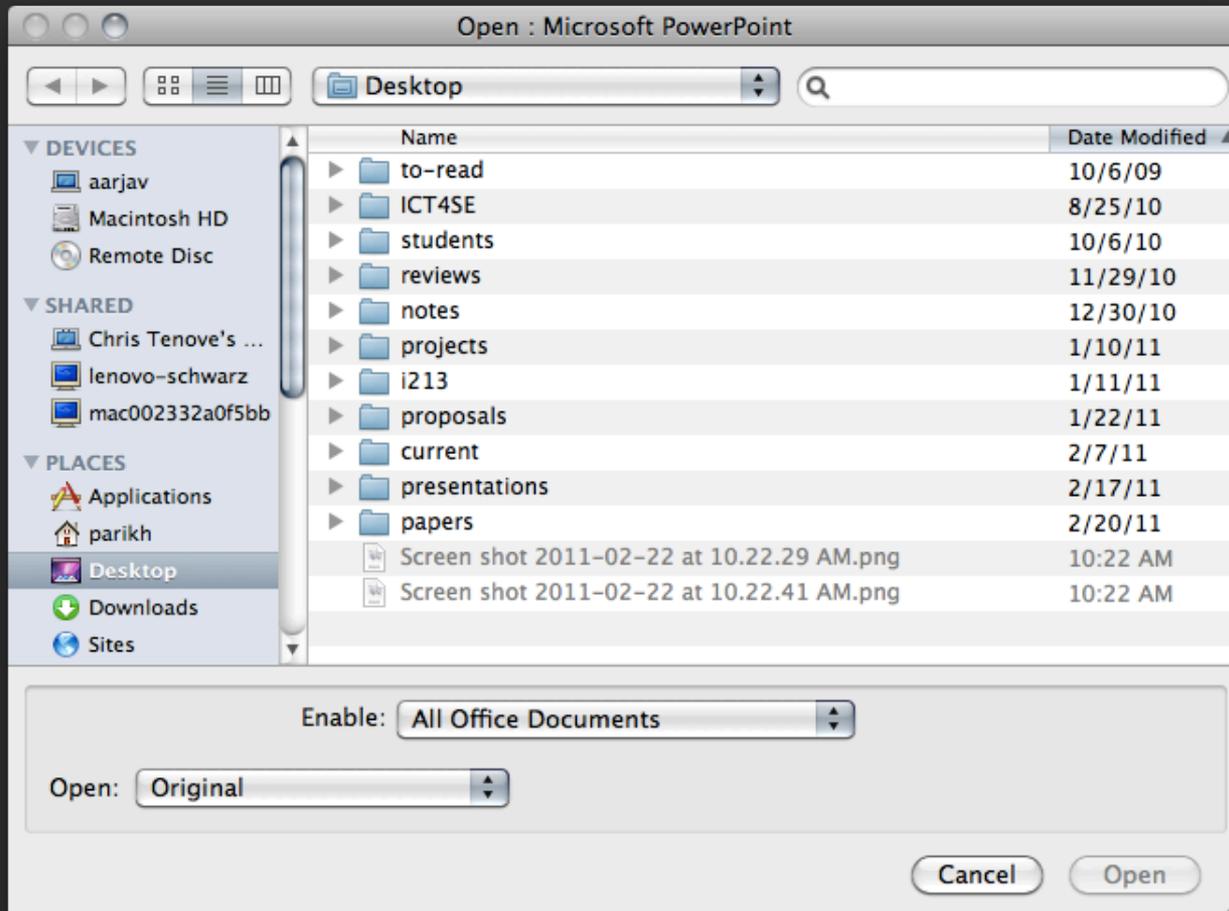
```
% ls
#cpfa.sas#
#guatamalan.lst#
#splot-pointbypoint.txt#
cholest.lst
cholest.sas7bdat
cholest2.lst
cpfa.sas
cpfa.sas~
guatamalan.lst
guatemalan.sas
guatemalan.sas~
intrinsic.lst
intrinsic.sas
intrinsic.sas~
iron.sas
iron.sas~
iron1.lst
planetdist.sas
planetdist.sas~
planetsumstat.sas
planetsumstat.sas~
pollen.sas
pollen.sas~
pollen1.lst
pollen2.lst
pollen3.lst
pollen4.lst
pollen5.lst
pollen6.lst
solar.lst
solarnlog.lst
splot-46-betlab.txt
splot-46-betlab.txt~
splot-46-pointbypoint
splot-46-pointbypoint
splot-pointbypoint.txt
splot-pointbypoint.txt~
splotpp.sas
splotpp2
splotpp2.sas
splotpp2.sas~
splotpp3
splotpp3.sas
splotpp4.sas
veggies.sas
veggies.sas~
%
```

CONSISTENCY

Ensure that the same action always has the same effect (avoid modes)

Present the same information in the same location

Follow established standards and conventions



PROVIDE FEEDBACK

Continuously inform the user about what is going on

Restate and rephrase user input

Provide warnings for irreversible actions

Give informative feedback even if the system fails

PROVIDE FEEDBACK

what mode am I in now?

what did I select?



how is the system interpreting my actions?

WAITING

Provide a progress indicator for any operation longer than ten seconds

Reassure the user system hasn't crashed

Indicate how long user has to wait

Provide something to look at

If can't provide specific progress, use generic "working" indicator like the spinning ball in Mac OS X

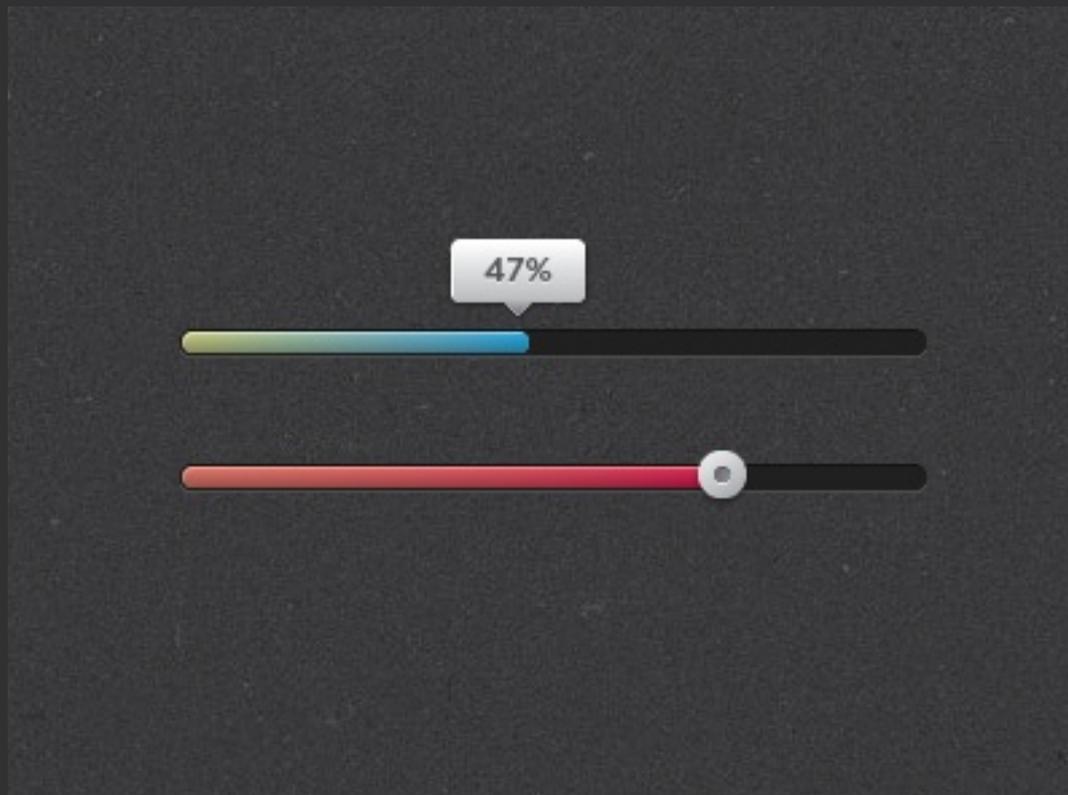
FIRST STEP

Enter your email address:

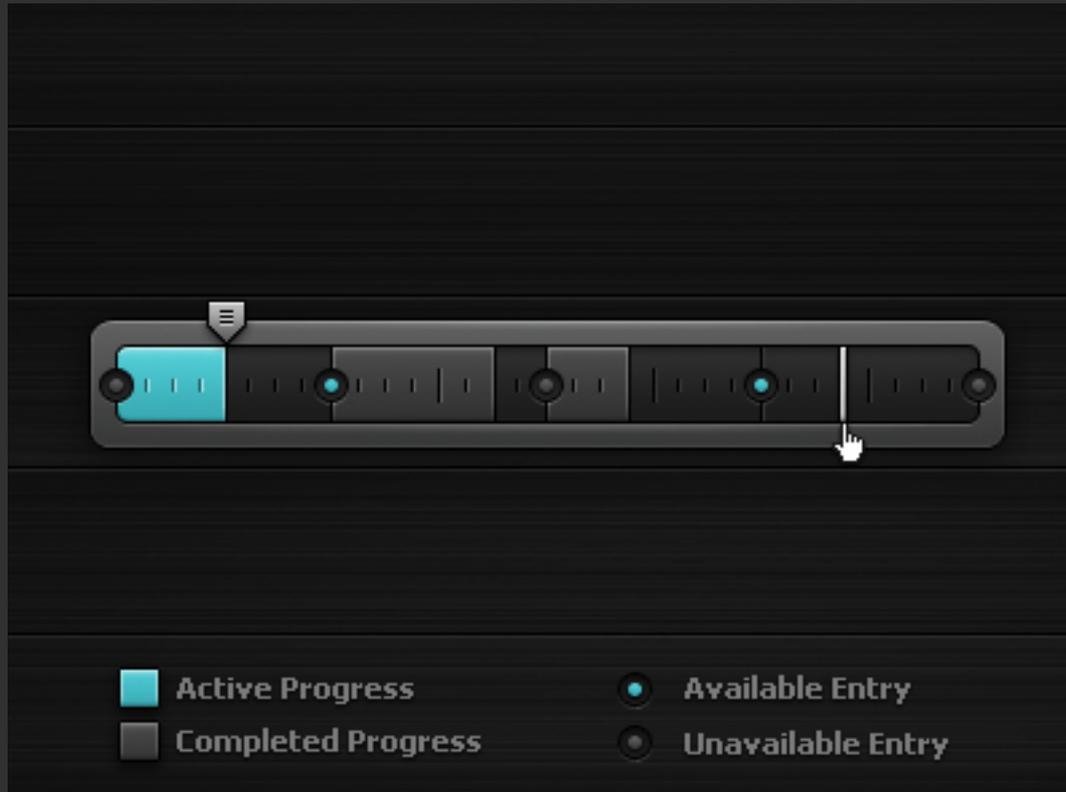
GO

• • •

The form is a white circle with a blue progress bar at the top and a blue 'GO' button on the right. It contains the text 'FIRST STEP', a prompt to enter an email address, an email input field with an envelope icon, and three dots at the bottom.



360 degree tour progress location bar



RESPONSE TIMES

0.1 second - perceived as instantaneous

1 second - user's flow of thought stays uninterrupted, but delay noticed

10 seconds - limit for keeping user's attention focused on the dialog

> 10 seconds - user will want to perform other tasks while waiting

CLEARLY MARKED EXITS

Don't "trap" the user

Provide an easy way out of trouble

Encourage exploratory learning

Mechanisms:

- Cancel

- Undo, Revert, Back

- Interrupt

- Exit

Adobe Illustrator



You are saving this document in Adobe Illustrator 9.0 format. Saving this document in an older format may disable some editing features when the document is read back in.

Yes

No

Now installing files, please wait...



Writing: E:\DRAWLT\SAMPLES\S-06-20.VLM

Percent Copied: 0%  100%

Press [Esc] To Abort

SHORT CUTS

Allow expert users to go fast

Avoid GUI operations

Mechanisms:

- Keyboard shortcuts

- Macros, scripts

- Type ahead

- Bookmarks, History

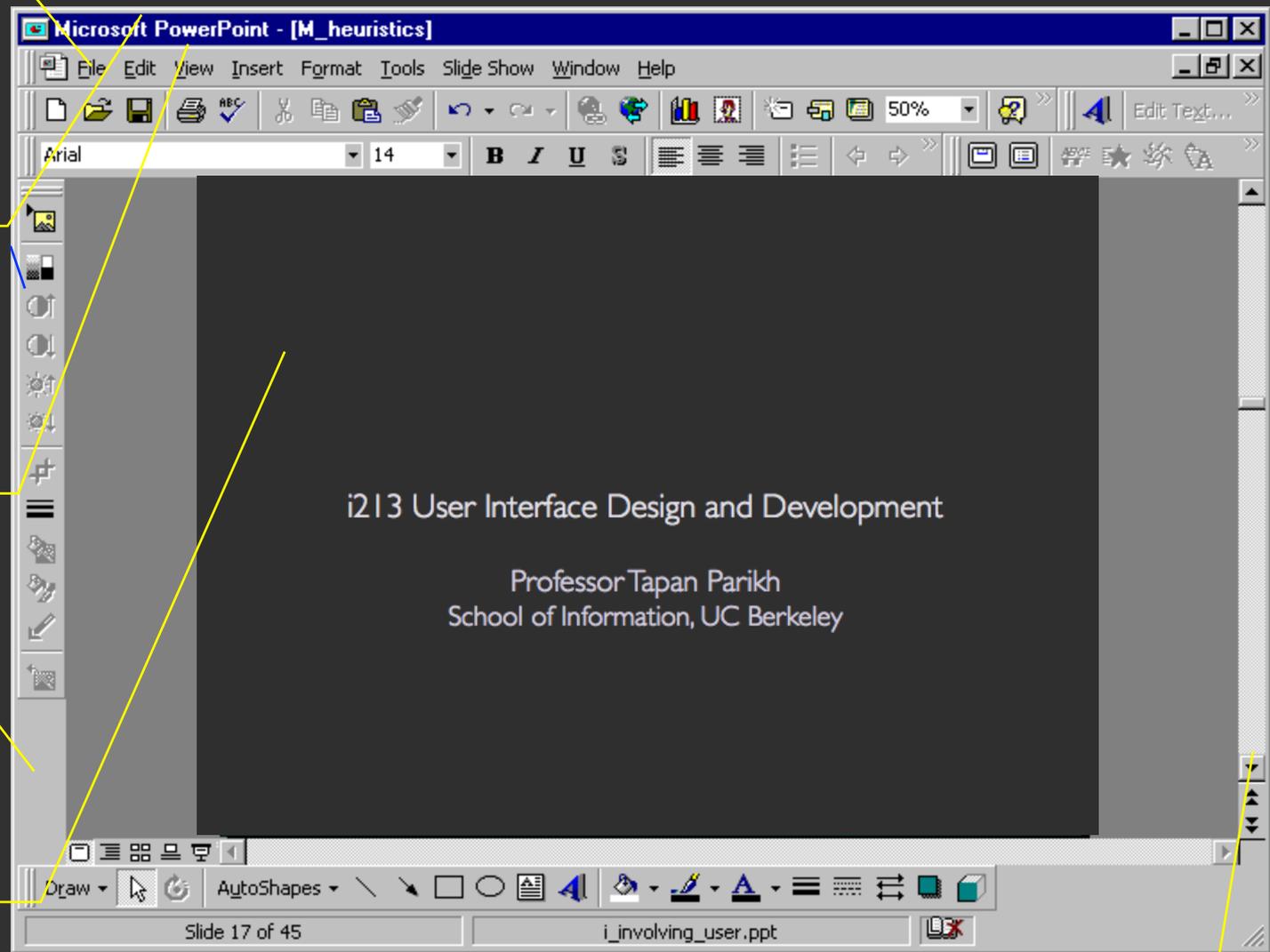
Keyboard accelerators for menus

Customizable toolbars and palettes for frequent actions

Split menu, with recently used fonts on top

Double-click raises toolbar dialog box

Double-click raises object-specific menu



Scrolling controls for page-sized increments

GOOD ERROR MESSAGES

Phrased in clear language

Avoid obscure codes

Precisely indicate the problem

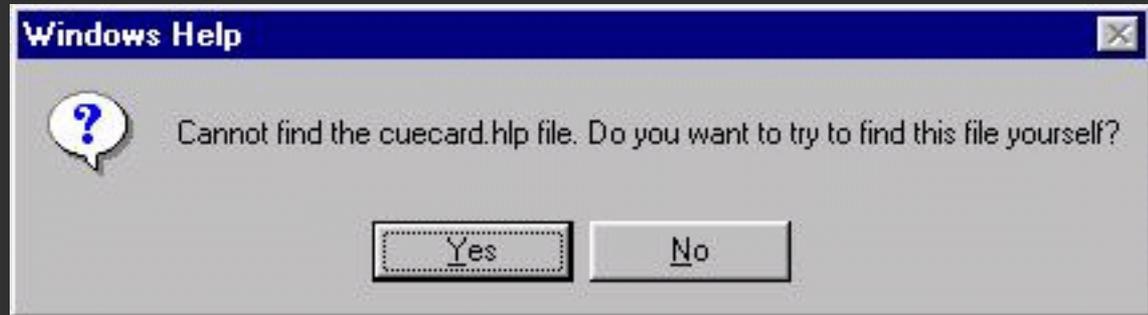
Restate user input

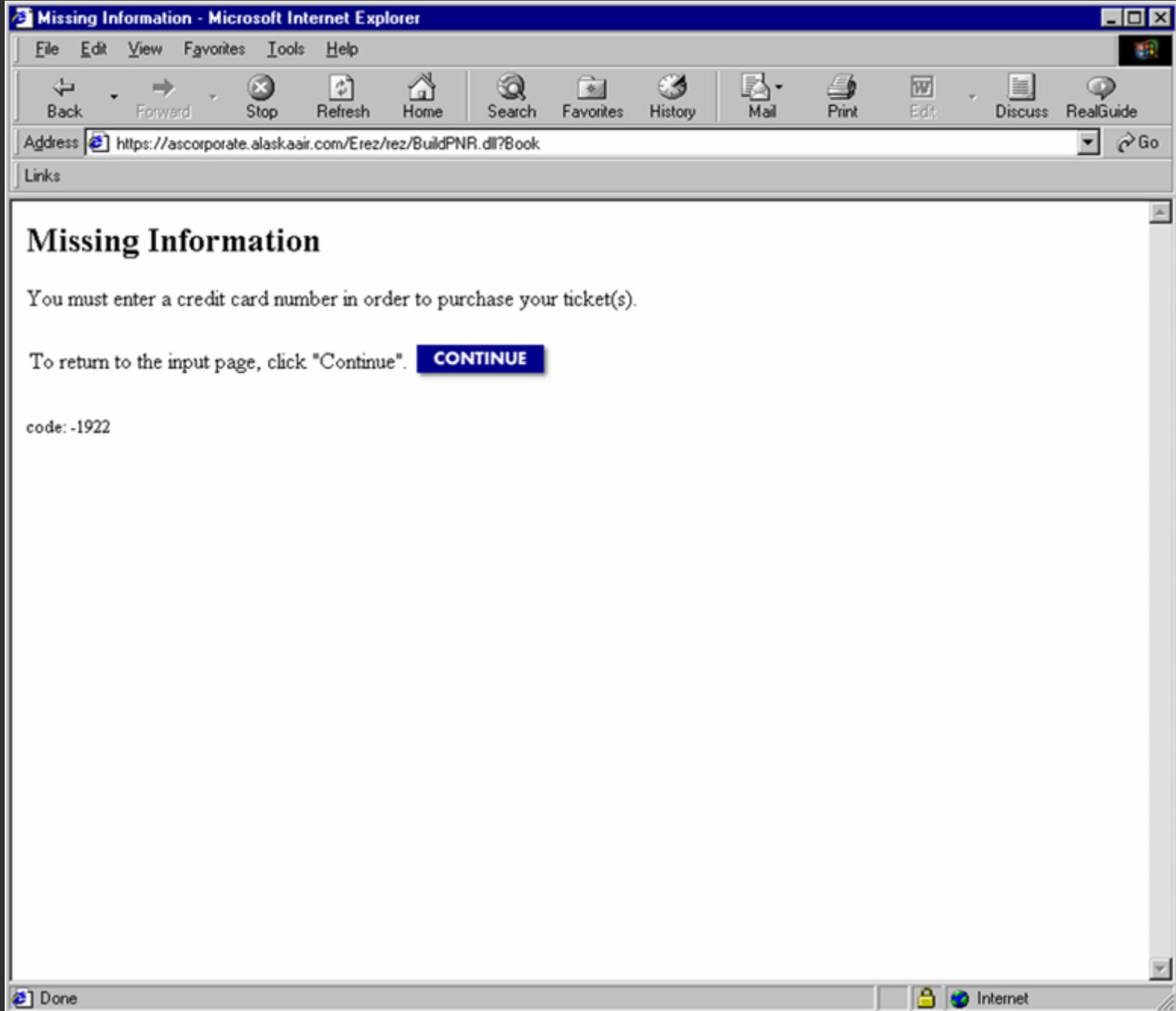
Do not blame the user

Constructively suggest a solution

Opportunity to help user in time of need

BAD





GOOD?

Applicant Tracking System - Printer Error



The Applicant Tracking System seems to have lost communication with the printer.

To solve this problem, make sure that the printer is switched on, and try to print again.

If printing still fails, try wiggling the cable that runs between the computer and the printer. Make sure the cable is connected securely at both ends, and try to print again.

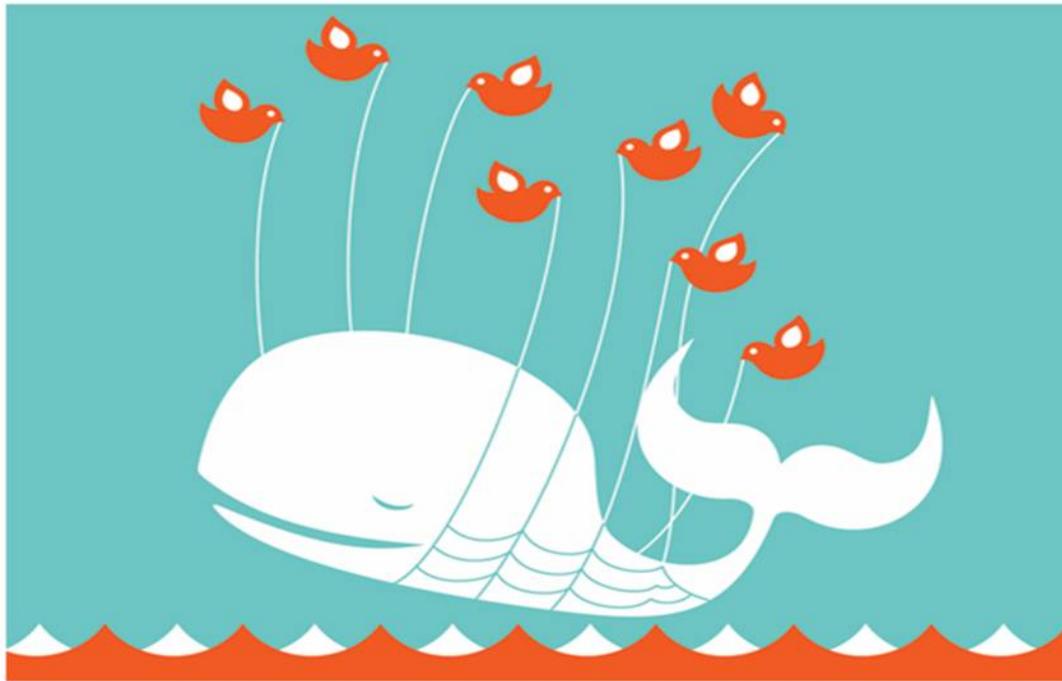
If the program still fails to print properly, please call Joe Grant at (212) 555-1212, and tell him that the program is reporting Error ATSPR35 at line 31 in module PRNFNC.

OK

twitter

Twitter is currently down for Unplanned maintenance.

We expect to be back in about an hour. Thanks for your patience.



© 2008 Twitter [Status](#) [Blog](#) [Help](#)

PREVENT ERRORS

Bounds-checking

Select rather than Enter

Judicious use of confirmation screens

Avoid modes, unless they are clearly visible or require action to maintain

PREVENT ERRORS

October	23	2005
November	24	2006
December	25	2007
January	26	2008
February	27	2009

Appointment

General | Attendees | Notes | Planner

When

Start: 8:30 AM Wed 5 /14 /97

End: 4:30 PM Wed 5 /14 /97

All day

Description:

Smart Technology Ser

Where:

May 1997

S	M	T	W	T	F	S
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

HELP AND DOCUMENTATION

Easy to search

Task-oriented

List concrete steps

Provide context-specific help

Shouldn't be too large

Is not a substitute for good design

HELP AND DOCUMENTATION



KINDS OF HELP

Tour / Demo

Tutorials

User Guide / Reference manual

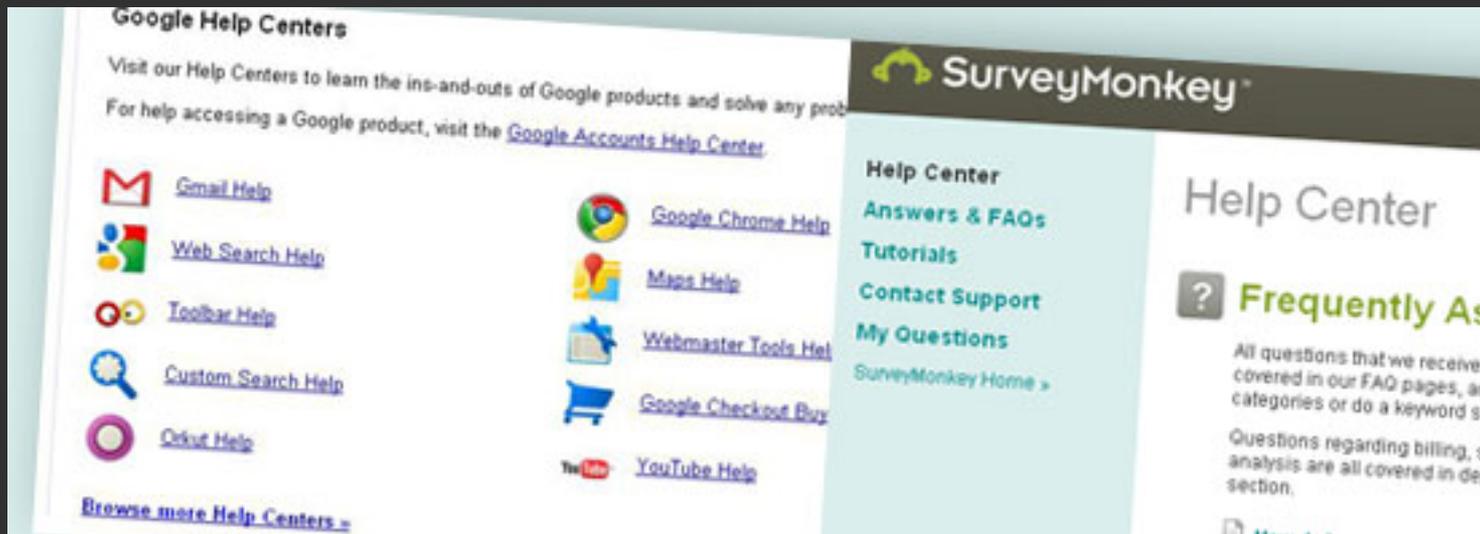
Searchable index

Tooltips, Balloon Help

Reference cards

Keyboard templates

HELP AND DOCUMENTATION



HEURISTIC EVALUATION

CONDUCTING A HEURISTIC EVALUATION

Can use hi-fi or lo-fi prototype

Each session should last 1-2 hours

Evaluator should go through the interface several times, with specific tasks in mind

- First pass: overall feel and scope, identify obvious violations
- Second pass: focus on specific elements

CONDUCTING A HEURISTIC EVALUATION

3-5 evaluators are enough to uncover most important problems

Each evaluator should inspect the interface alone (to reduce bias)

After the session, the evaluators aggregate observations

Output is a list of usability problems

CONDUCTING A HEURISTIC EVALUATION

If the system is intended to be “walk up and use”, then evaluators should be provided with minimal help

If the system requires training, then evaluators should be trained and given an example scenario

Evaluator can be helped after they have made an attempt and articulated their difficulties

CONDUCTING A HEURISTIC EVALUATION

Pre-evaluation training

Evaluation

Severity / Fixability rating

Debriefing

SEVERITY RATINGS

Provided by each evaluator

Based on *frequency, impact, persistence*

Combined into a single numeric index

Average taken across evaluators

Allows for prioritization of fixes

SEVERITY RATINGS

0: don't agree that this is a problem

1: cosmetic problem

2: minor problem

3: major problem; important to fix

4: catastrophe; imperative to fix

DEBRIEFING

Conducted with evaluators, observers, and development team

Discuss characteristics of UI

Suggest improvements to address major usability problems

Dev team provides fixability ratings (if it exists)

Make it a brainstorming session

FIXABILITY

Describes how easy each problem would be to fix

Requires some technical knowledge of system & platform

Allows for estimating “cost-benefit”

Can provide possible fix as guidance to development team

FIXABILITY

0: Impossible to Fix

1: Nearly Impossible to Fix

2: Difficult to Fix

3: Easy to Fix

4: Trivial to Fix

A list of problems with heuristics, severity, fixability and possible fixes

Evaluator: John T. Doe							
Date: January 1, 2008							
System: Nokia Mobile Phone Model #9999							
<i>Number</i>	<i>Heuristic</i>	<i>Location</i>	<i>Description</i>	<i>Severity</i>	<i>Fixability</i>	<i>Sum</i>	<i>Possible Fix</i>
1	Visibility of system status	Home screen	The home screen does not portray any information about battery power remaining, making it hard for users to tell how much power they have left.	3	3	6	Display a battery life indicator on the home screen.
2	User control and freedom	Screen for writing a text message	Once you are on the screen for writing a text message, you cannot leave without sending the message. Users need a way to get out if they decide not to send a message.	3	2	5	Allow the CLR button to always move the user back one screen no matter where they are.

For next time

Low fidelity paper prototypes due

In class formative evaluation