Administrative Stuff

Assignment 1 can be uploaded as a zip or e-mailed to adpeer@cs.wisc.edu and irene@cs.wisc.edu

Project 1 instructions are posted on the website under "Assignments"

Processbook Template is available, can use extra pages

Mid-semester evaluations

How does sketching tie in to creating work models for CIs?

It provides a tool to create simple but effective models based [on] the inquiries

Show[s] how the project would address the users' need and problems in a clear and visually friendly way

Sketching will help quickly represent our ideas about the users' work models

Sketching helps give a visual outline of the work model

It's easier to draw sonething then to write it out and it gives good insight into what actually was seen. gives us visuals that we wouldn't have normally seen but shows what the user did and how

If there is a disagreement about the flow of the model it can be quickly and cheaply changed.

Sketching allows one to quickly visualize information, interactions, and ideas about the outcomes from a contextual inquiry without the time and economic costs of other methods, like prototyping.

Usability Evaluation Basics

Irene Rae
Computer Sciences
University of Wisconsin–Madison

CS-570 Introduction to Human-Computer Interaction

What is usability?

Some Definitions

"Usability is a quality attribute that assesses how easy user interfaces are to use."

"The word 'usability' also refers to methods for improving ease-of-use during the design process."

— Jacob Nielsen

5 Components

According to Nielsen

Learnability	How easy is it for users to accomplish basic tasks the first time they encounter the design?
Efficiency	Once users have learned the design, how quickly can they perform tasks?
Memorability	When users return to the design after a period of not using it, how easily can they reestablish proficiency?
Errors	How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
Satisfaction	How pleasant is it to use the design?

More Definitions

Usability

How well a system satisfies its intended use.

Usefulness

How much value a system offers to its user.

Desirability

How desirable the system is for its user.

Why Things Are Not Usable

According to Rubin & Chisnell

Development focuses on the machine or system.

Target audiences change and adapt.

Designing usable products is difficult.

Team specialists don’t always work in integrated ways.

Design and implementation don’t always match.

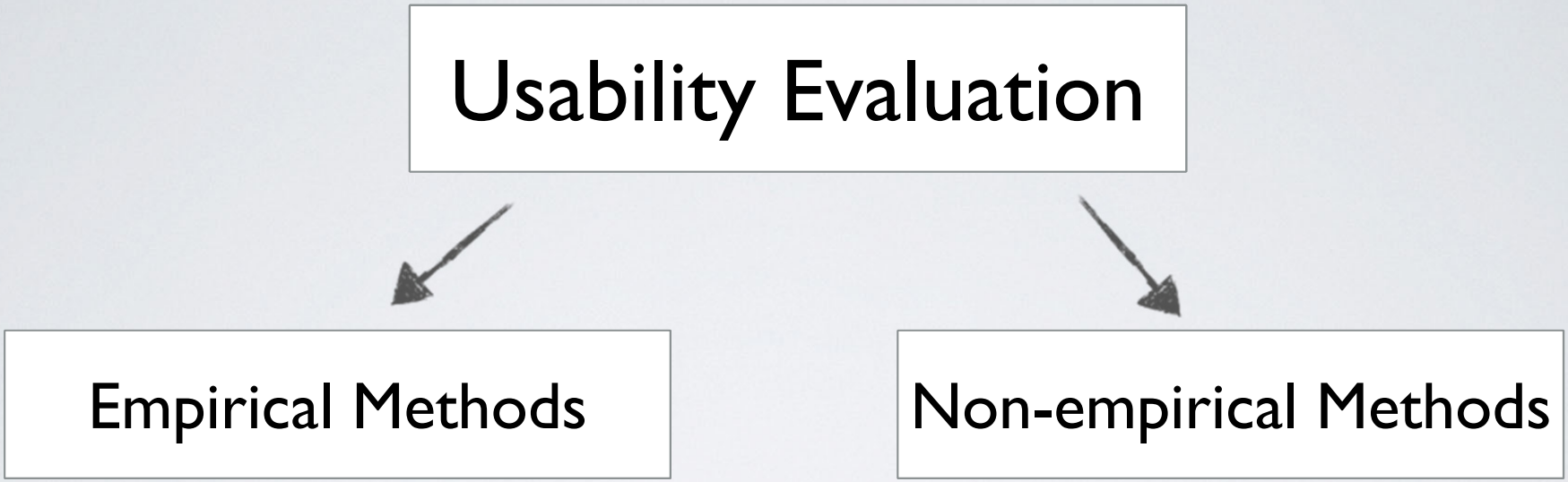
How To Make Them Usable

According to Rubin & Chisnell

Early focus on users and their tasks

Evaluation and measurement of product usage

Iterative design



Usability Testing

Performance Evaluation

Behavior Evaluation

Measuring Satisfaction

Cognitive modeling

Heuristic evaluation

Cognitive walkthrough

Usability Testing

Why Do Usability Tests?

Informing design

Eliminating design problems and user frustration

Create data for communication in the design team

Remove expert blind spot

Improving profitability

5 Characteristics

According to Dumas and Redish (1999)

1. The goal is to improve usability.
2. The participants represent real users.
3. The participants do real tasks.
4. You observe and record what the participants do and say.
5. You analyze the data, diagnose the real problems, and recommend changes to fix the problems.

Participants?
Is this a Scientific Experiment?

NOT a Scientific Experiment

Experimental control is not as necessary.

Data measurement is not as precise.

Changes can be made mid-test to explore alternatives.

The exact same test does not have to be repeated precisely for each user.

The number of participants is fewer in usability testing than in scientific experiments.

Hypotheses and inferential statistics are used less often.

Research Method	Research	Usability testing
Experimental design	Isolate and understand specific phenomena with the goal of generalization to other problems	Find and fix flaws in a specific interface, not goal of generalization
Experimental design	A larger number of participants is required	A small number of participants can be utilized
Ethnography	Observe to understand the context of people, groups, and organizations	Observe to understand where in the interface users are having problems
Ethnography	Researcher participation is encouraged	Researcher participation is not encouraged in any way
Ethnography	Longer-term research method	Short-term testing method
Ethnography and experimental design	Used to understand problems or answer research questions	Used in systems and interface development
Ethnography and experimental design	Used in earlier stages, often separate from (or only partially related to) the interface development process	Typically takes place in later stages, after interfaces (or prototypes) have been developed
Ethnography and experimental design	Used for understanding problems	Used for evaluating solutions

Questions?

When To Test

Formative vs. Summative

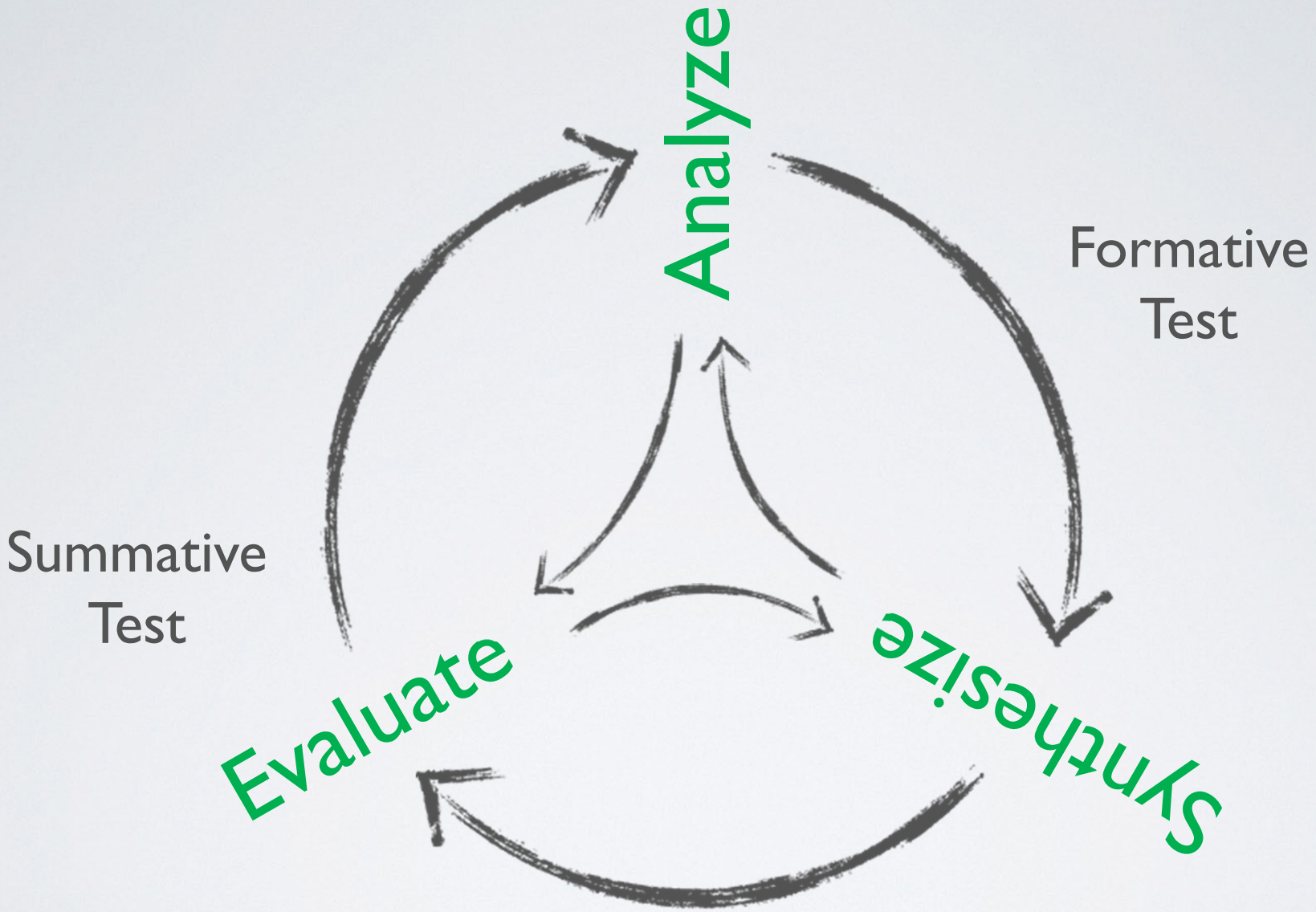
Formative tests help “form” a design.

- They are quick and dirty and run during many stages of development.

Summative tests are run “at the sum” of the project.

- They give final results about a more advanced prototype.
- Usually more formal and thorough than a formative test.
- Might lead to a report that you pass to a product management team.

TopHat Question


Analyze
Formative
Test
Synthesize
Evaluate
Summative
Test

The Usability Test Process

Stages of a Usability Test

According to Rubin & Chisnell

1. Develop the test plan
2. Set up the test environment
3. Find and select participants
4. Prepare test materials
5. Conduct the test sessions
6. Debrief the participants
7. Analyze data and observations
8. Report findings and recommendations

Parts of a Test Plan

According to Rubin & Chisnell

1. Purpose, goals, and objectives of the test
2. Research questions
3. Participant characteristics
4. Method (test design)
5. Task list
6. Test environment, equipment, and logistics
7. Test moderator role
8. Data to be collected and evaluation measures
9. Report contents and presentation

Test Plan

Methodology

This usability study will be somewhat exploratory but will also gather assessment data about the effectiveness of www.H.com. Participants will fall into three groups by the starting point they use to perform the main task, which is to reserve a room. We will collect data about error and success rates as well as qualitative data about participants' experiences using the site.

We will use a between-subjects design

In this between-subjects study, each participant will work through one task path (in a within-subjects study, each participant would try all paths in counterbalanced order). I will conduct up to 30 individual 45-minute usability study sessions. Each participant will perform one of three major task "paths" using www.H.com. I'll use 15 minutes of each session to explain the session to the participant, review basic background information with the participant, and then conduct a post-test debriefing interview. During the middle 30 minutes of the session, participants will work to reserve a room at an H property in a major U.S. city.

Session outline and timing

The test sessions will be 45 minutes long. I will use 15 minutes of each session for pre-test introductions and post-test debriefing interviews. The sessions will take place at Shugoll Research in Bethesda.

Pre-test arrangements

Have the participant:

- Review and sign nondisclosures and recording permissions.
- Fill out a background questionnaire (with the same questions as the screener).

Introduction to the session (2 minutes)

Discuss:

- Participant's experience with usability studies and focus groups.
- Importance of their involvement in the study.
- Moderator's role.
- Room configuration, recording systems, observers, etc.
- The protocol for the rest of the session.
- Thinking aloud.

Background interview (3 minutes)

Discuss the participant's:

- Experiences booking their own travel.
- Reasons for booking their own travel.

Tasks (30 minutes)

Participants will start at one of three points to reserve a room at an H hotel in a major U.S. city where H has multiple properties.

Post-test debriefing (10 minutes)

- Ask broad questions to collect preference and other qualitative data.
- Follow up on any particular problems that came up for the participant.

Introducing the Test

"Today we are interested in learning about X. That's where you come in!"

"I did not develop X. I just want to know what the problems are with X."

"I will give you tasks to perform with X. Do your best."

"It is X being tested here, not you."

"It is essential that you think out loud while working with X. Tell me constantly what you are thinking, looking for, wondering, confused about, surprised, and so on. If you stop talking, I will prompt you to talk."

"I will not be able to answer your questions when you start using X. Do you have any questions now?"

Running the Test

Give the opening scenario.

Then begin assigning tasks in order.

Observe task time, errors, breakdowns, workarounds, confusions, and success/failure.

Make notes, video-record, audio-record.

Task-based Usability

Set up an overall scenario.

Prescribe tasks for users fitting the scenario.

Ask users to perform these during the test.

- Improvisation is okay — just watch the time!

Creating the Tasks

With a short paragraph, set the stage.

> "We are interested in improving people's ability to save, update, and use contacts in their mobile phones."

Then write a numbered-list of tasks.

1. Try to find the contacts list in the phone.
2. View the contact information for John Smith.
3. Change John Smith's number to end in a "6".
4. …

Tasks should be realistic, doable, ethical, clear, succinct, and be focused on uncertain aspects of the design.

Chain tasks so that one leads naturally to the next.

Think-aloud 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?"

Allowing Them to Stray

If you build extra time into your tests, you can allow users to stray a bit as they work.

They should stay on task.

But they might wander down a rabbit hole.

This can yield good data, but takes time.

Eventually, you may have to interrupt and prompt them to find their way back. If they can't, help them, and note a major failure.

TopHat Question

Answering Questions

Basically, you really, really shouldn't.

You wouldn't be there "in real life."

You want to see if they can figure it out.

You want to see how hard it is.

You want to see how catastrophic the outcome is if they keep struggling.

Answering users' questions for help ruins your data and contaminates it.

Types of Data

Objective measurements;

- Things that can be measured objectively the same by everyone.
- Time, errors, confusions, breakdowns, workarounds, successes, and failures.

Behavioral measurements;

- Mostly your observations.
- Notes about where, when, why and how the above things occurred.

Subjective measurements;

- Users' comments and feedback.
- Often a questionnaire is used at the end.

Debriefing

Give them more detail about what you were interested in discovering with their help.

Answer any questions they might have.

- Now you can show them how to accomplish the tasks that they had failures on.

Thank them for their time.

Pay them!

Analyzing Test Data

Gather notes, measures, video, audio, screen captures, and so on for each participant.

Distill main findings in a document.

Single data format is best.

Look for frequent or recurring problems.

Look for particularly *catastrophic* problems;

Even if they happened for just one user.

Assign severity and fixability ratings.

Questions?

Non-Task Based Testing

Tasks vs. Open Exploration

Task-based usability tests are easiest to control, focus, and direct.

Open exploration tests can also be used to understand people's initial experiences.

"Take 10 minutes to explore the system. Do whatever you like. I only ask that you keep thinking out loud so that I know what you're trying to do, what you find interesting, and what confusions you have."

What are the tradeoffs here?

Co-discovery

Two users work together and talk to each other out loud to accomplish assigned tasks.

- Observers can learn a lot from hearing two people talk to each other.

Downsides?

- Costs more — have to pay 2x.
- No-shows are more problematic.
- Group effects come in — one disruptive user ruins two.

How many users should you test with?

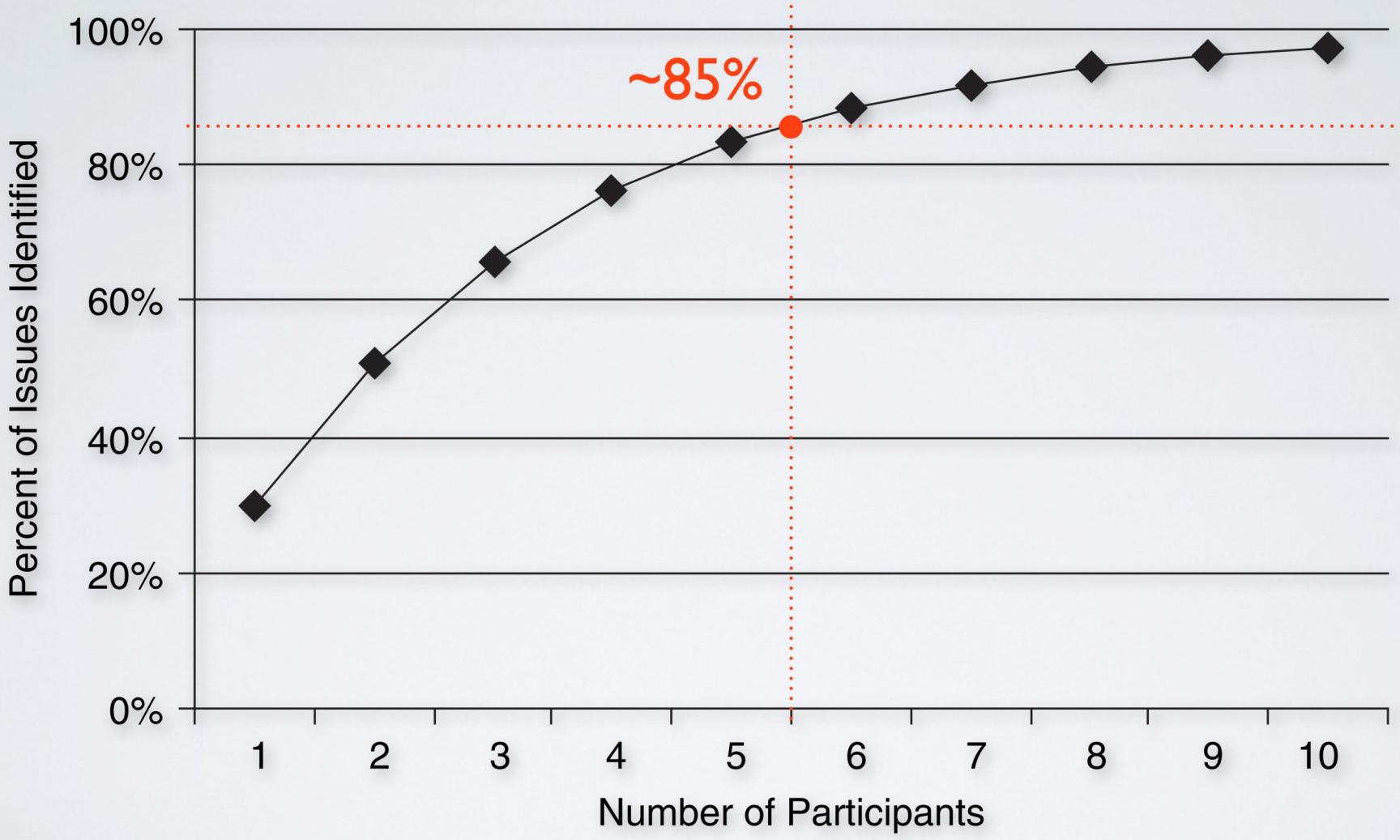
How Many Users?

$$N(1-(1-L)^n)$$

N = total number of usability problems discovered with a single user

L = proportion of usability problems discovered with a single user

n = number of users



* Tullis & Albert

Rules of Thumb

Fifteen users allow you to find all usability problems

Five users allow you to find 85% of problems

"You add more and more users, you learn less and less..."

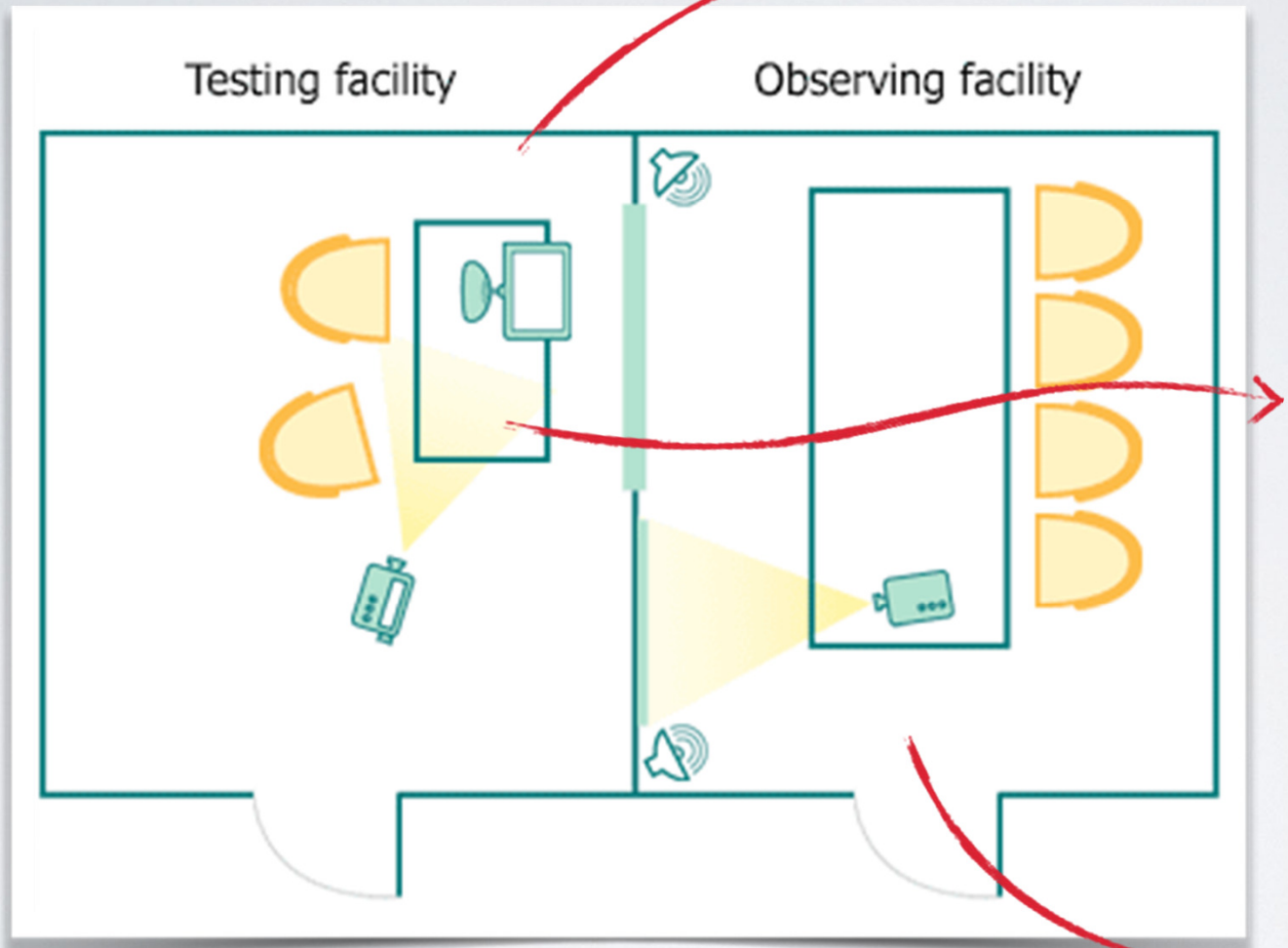
Use your budget to run 3 tests with 5 users rather than 1 test with 15 users

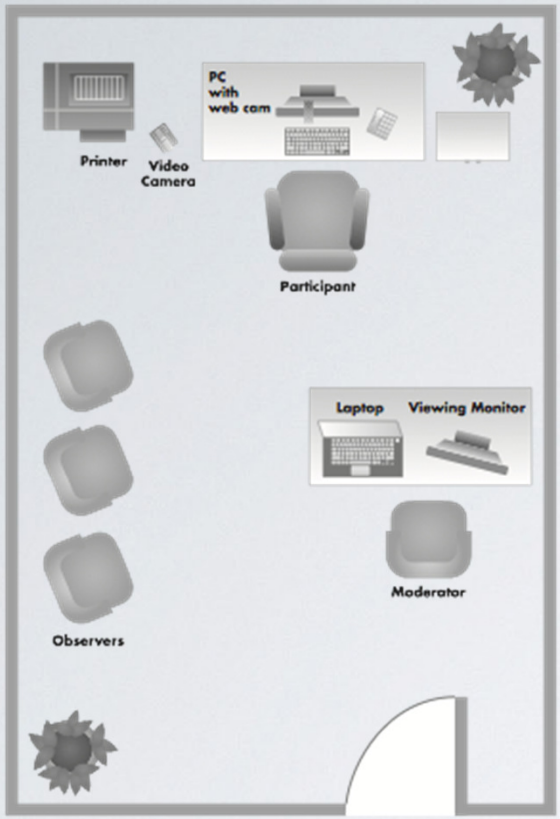
Iterative design and testing might uncover all remaining problems

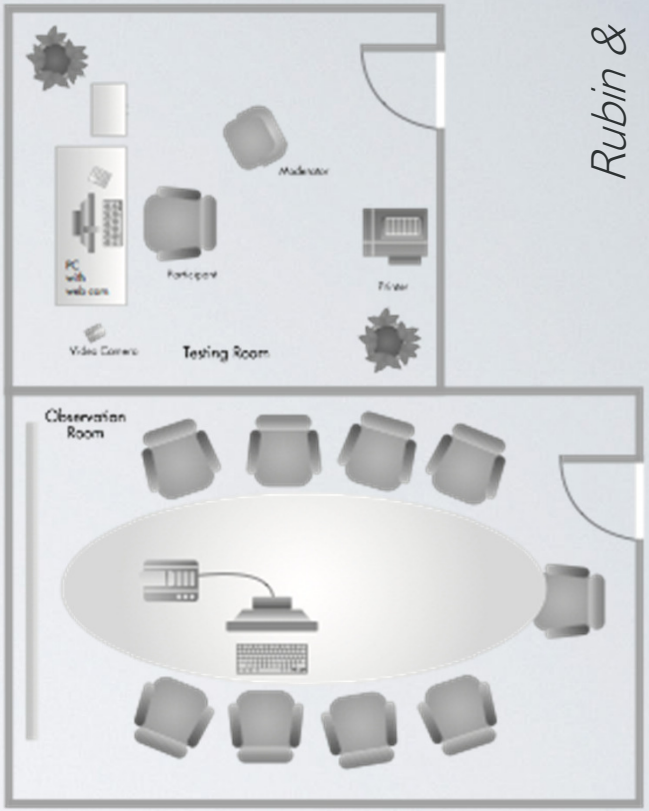
Questions?

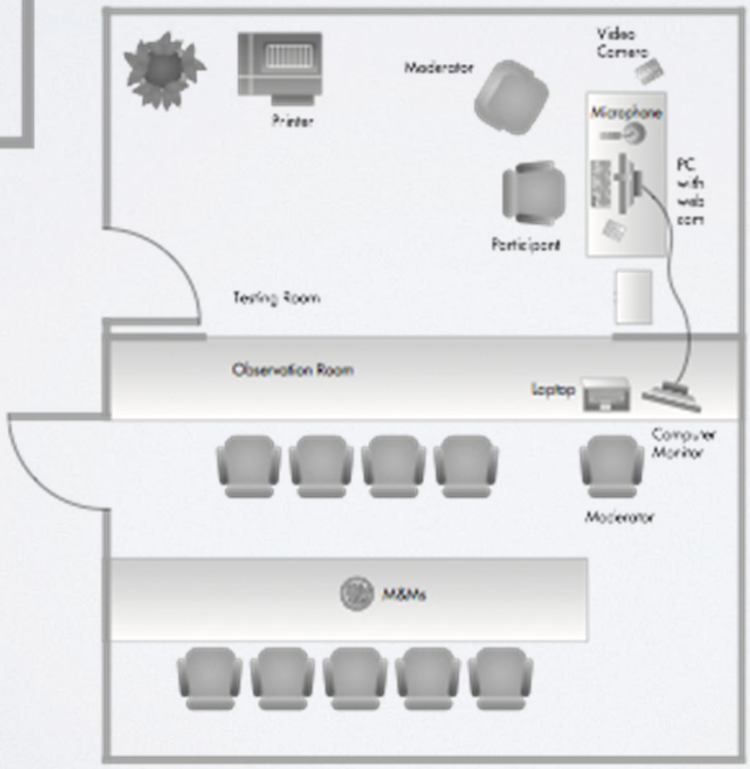
Where do I do testing?

Usability Labs




Printer
Video Camera
PC with web cam
Participant
Laptop
Viewing Monitor
Moderator
Observers


Moderator
Participant
Printer
PC with web cam
Video Camera
Testing Room
Observation Room


Printer
Moderator
Video Camera
Microphone
PC with web cam
Participant
Testing Room
Observation Room
Laptop
Computer Monitor
Moderator
M&Ms

Test Moderators

According to Rubin & Chisnell

Grounding in the basics of user-centered design

Quick learner, good memory, long attention span

Instant rapport with participants

Comfortable with ambiguity, flexible

Empathic “people person”

“Big picture” thinker

Good communicator, organizer, and coordinator

Moderating Problems

According to Rubin & Chisnell

Leading rather than enabling

Too involved with the act of data collection

Acting too knowledgeable

Too rigid with the test plan

Not relating well to each participant

Jumping to conclusions

Limitations of Testing

According to Rubin & Chisnell

Testing is always an artificial situation.

Test results do not prove that a product works.

Participants are rarely fully representative of the target user population.

Testing is not always the best technique to use.

Questions?

Think-aloud Protocol

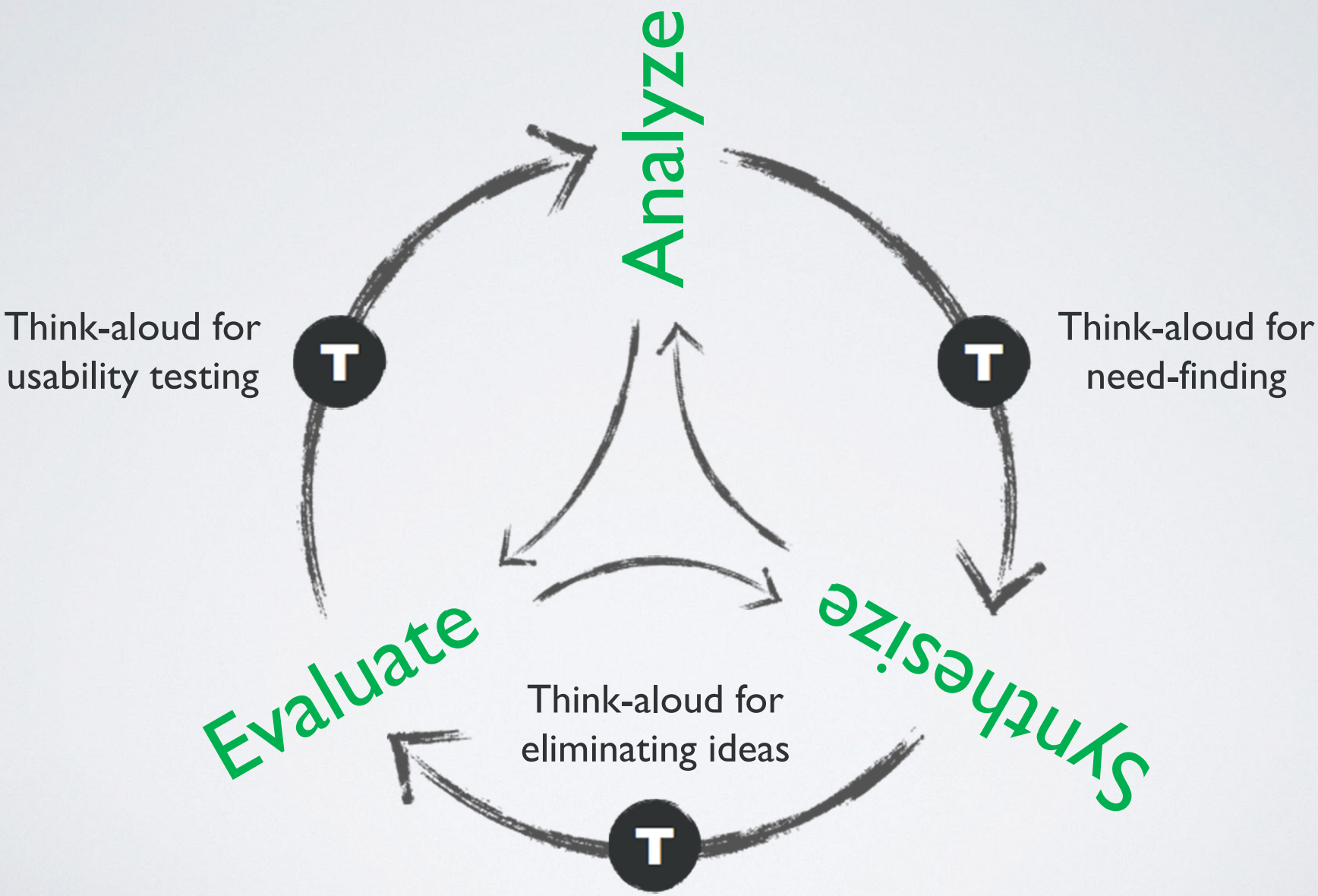
Think-aloud Protocol

Think-alouds serve as the basis for understanding users' cognition and perception during testing.

Used by Ericsson and Simon (1984) to learn about people's cognitive processes for doing things.

Herb Simon — Turing Award, 1975; Nobel Prize 1978

THE DESIGN PROCESS



In-class Exercise

Think-aloud Exercise

I need two volunteers.

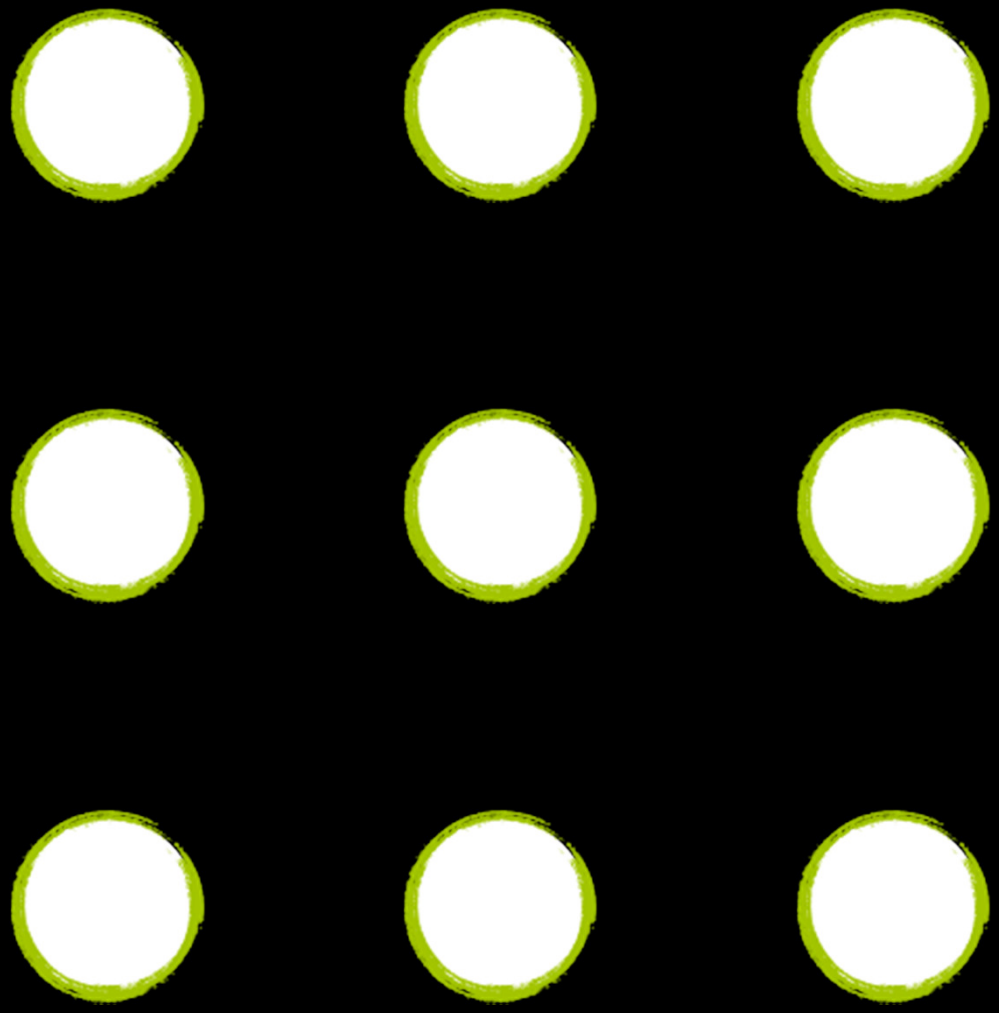
One is user.

Not familiar with the 9-dot problem.

Another is tester.

Tester runs the think-aloud on the user.

Keep them talking!



Connect all 9 dots using 4 or less
straight lines without lifting your pen.

Running the Test

Introduce the test;

"The interface is being tested, not you."

"I didn't design or build this; I just want to know what the problems are."

Prompt them to continually think-aloud.

Observe task times, errors, confusions, breakdowns, workarounds, and success/failure.

Points to Remember

Keep prompting user to talk while performing.

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), or take good notes. Screen captures can also be useful.

Think-aloud Prompts

"Please keep talking."

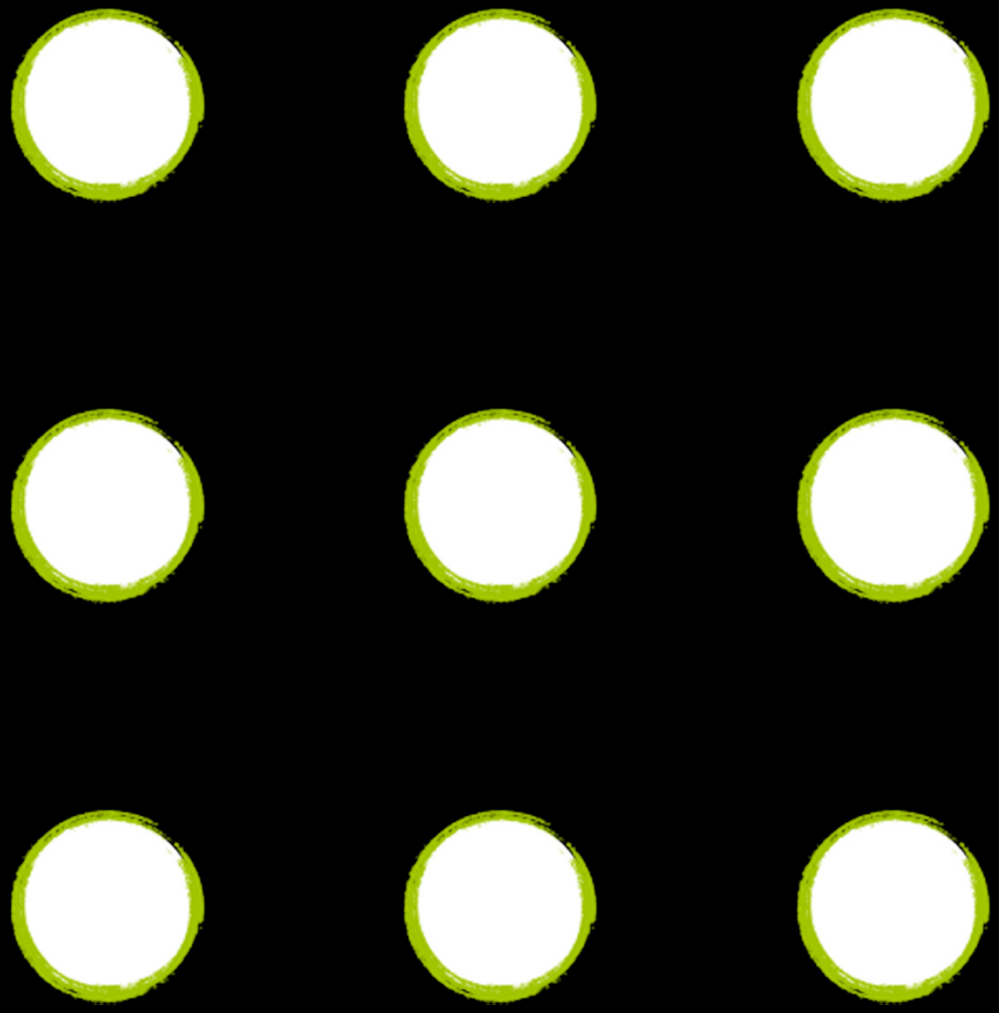
"Tell me what you are thinking."

"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?"



Connect all 9 dots using 4 or less
straight lines without lifting your pen.

How'd They Do?

User:

- Was that hard to think-aloud?
- Did thinking-aloud make it harder?

Tester:

- Was it hard to get them to think-aloud?
- Did you worry about being disruptive?

Insight Problems

The only major problem with thinking-aloud as a technique;

When people are trying to figure something out, and it requires "insight," thinking-aloud can be very disruptive.

Therefore, if your participant is really baffled and thinking hard, it might *not* be the best time say "keep talking."

Instead, wait until a natural break, and then ask them, "What were you thinking just there?"

Retrospective Think-aloud

For insight problems, video- and audio-tape sessions, and then do the think-aloud after-the-fact while watching the video with the user.

Do it right after the tasks are done.

Thanks!

Irene Rae
Computer Sciences
University of Wisconsin–Madison

CS-570 Introduction to Human-Computer Interaction