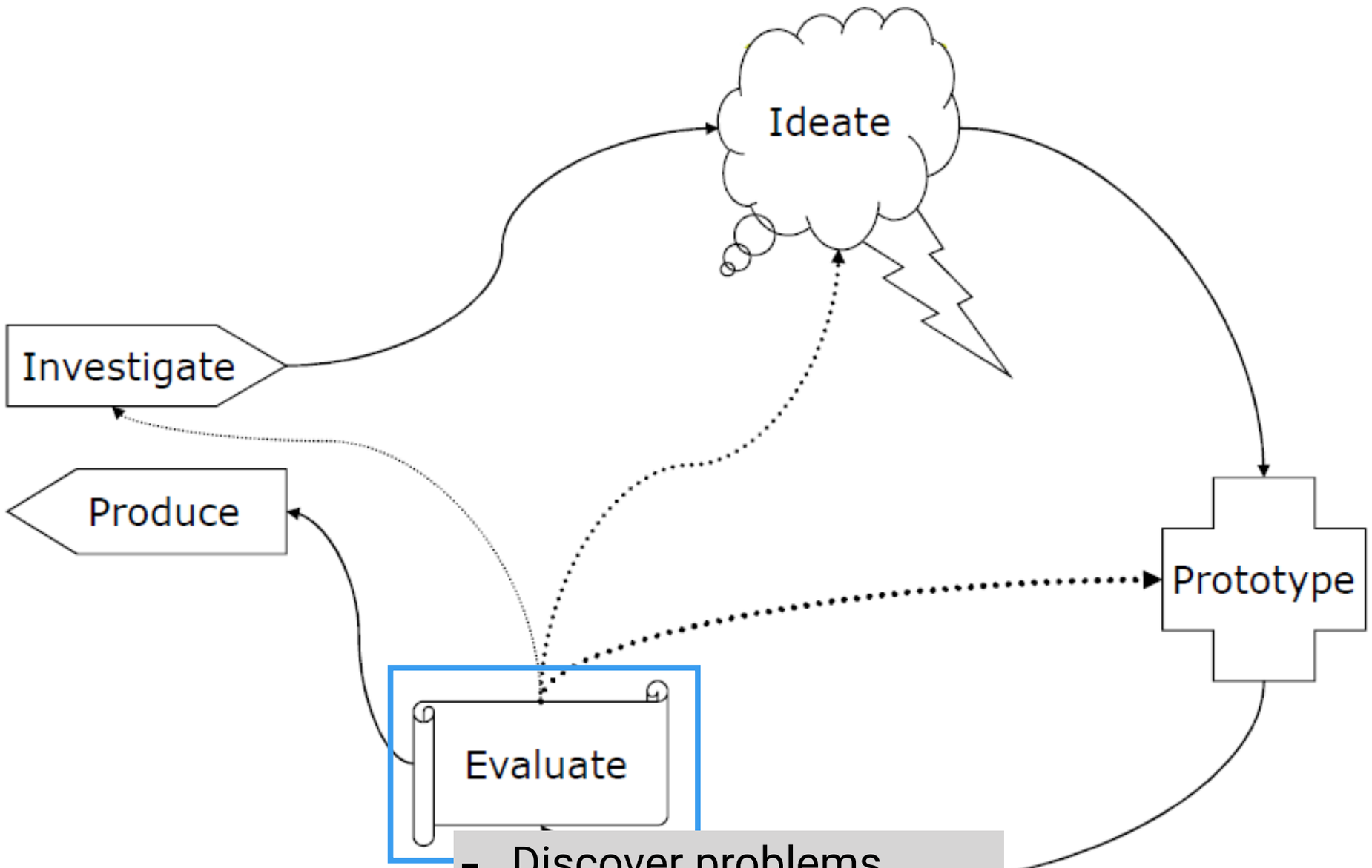
Human-Computer Interaction

CPSC 481 - Winter 2019

Usability Evaluation

With acknowledgements to Tony Tang

Evaluate



- Discover problems
- Assess progress
- Determine next steps

Why evaluation?

- Automated processes can find bugs, but not usability issues
- Evaluation gives you a way to move forward
  - What needs to be fixed, added, removed?
- Answers to two questions:
  - Did we build the right thing?
  - Did we build the thing right?

Which evaluation method to choose?

- Time
- Cost
- Required number of specialists
- Required number of users
- Physical environment configuration
- Equipments

In most organizations, you have three major options

- **“Inspection (Expert) Evaluations”**
  - Task Centered System Evaluation; Heuristic Evaluation; Guideline Review
- **Usability Test**
  - Formal method of evaluation that asks (potential) users to complete tasks
- **Field Deployment**
  - Give a prototype to users in the field, and watch their usage/ask for feedback

Within an organizational context

- **Reviews with stakeholders**
  - Usually, fairly cursory as a presentation / part of a meeting
  - General flow, look/layout/feel
  - Useful for: getting people on board
- **Test with users**
  - See whether it actually works with real people
  - Looking for the problems that people encounter
  - In organizations with poor design culture: part of “quality assurance” (aka “testing”)

Inspection Evaluation

- Who evaluates?
  - Usability specialist
  - Software development consultants specialized in a particular interface style
- Inspection methods
  - Heuristic Evaluation
  - Guideline Review
  - Cognitive Walkthrough

“User” Testing

~~“User”~~ Testing
Usability Test

- A usability test is a “formal” method for evaluating whether a design is learnable, efficient, memorable, can reduce errors, and meets users’ expectations.

- **Users are not being evaluated**

- The design is being evaluated

Usability Test: Essentially

Usability Test: Essentially

- Bring in real users

Usability Test: Essentially

- Bring in real users
- Have them complete tasks with your design, while you watch **with your entire team**

Usability Test: Essentially

- Bring in real users
- Have them complete tasks with your design, while you watch **with your entire team**
- Use a think-aloud protocol, so you can “hear what they are thinking”

Usability Test: Essentially

- Bring in real users
- Have them complete tasks with your design, while you watch **with your entire team**
- Use a think-aloud protocol, so you can “hear what they are thinking”
- Measure
  - Task completion, task time
  - Satisfaction, problem points, etc.

Usability Test: Essentially

- Bring in real users
- Have them complete tasks with your design, while you watch **with your entire team**
- Use a think-aloud protocol, so you can “hear what they are thinking”
- Measure
  - Task completion, task time
  - Satisfaction, problem points, etc.
- Identify problems (major ones | minor ones)

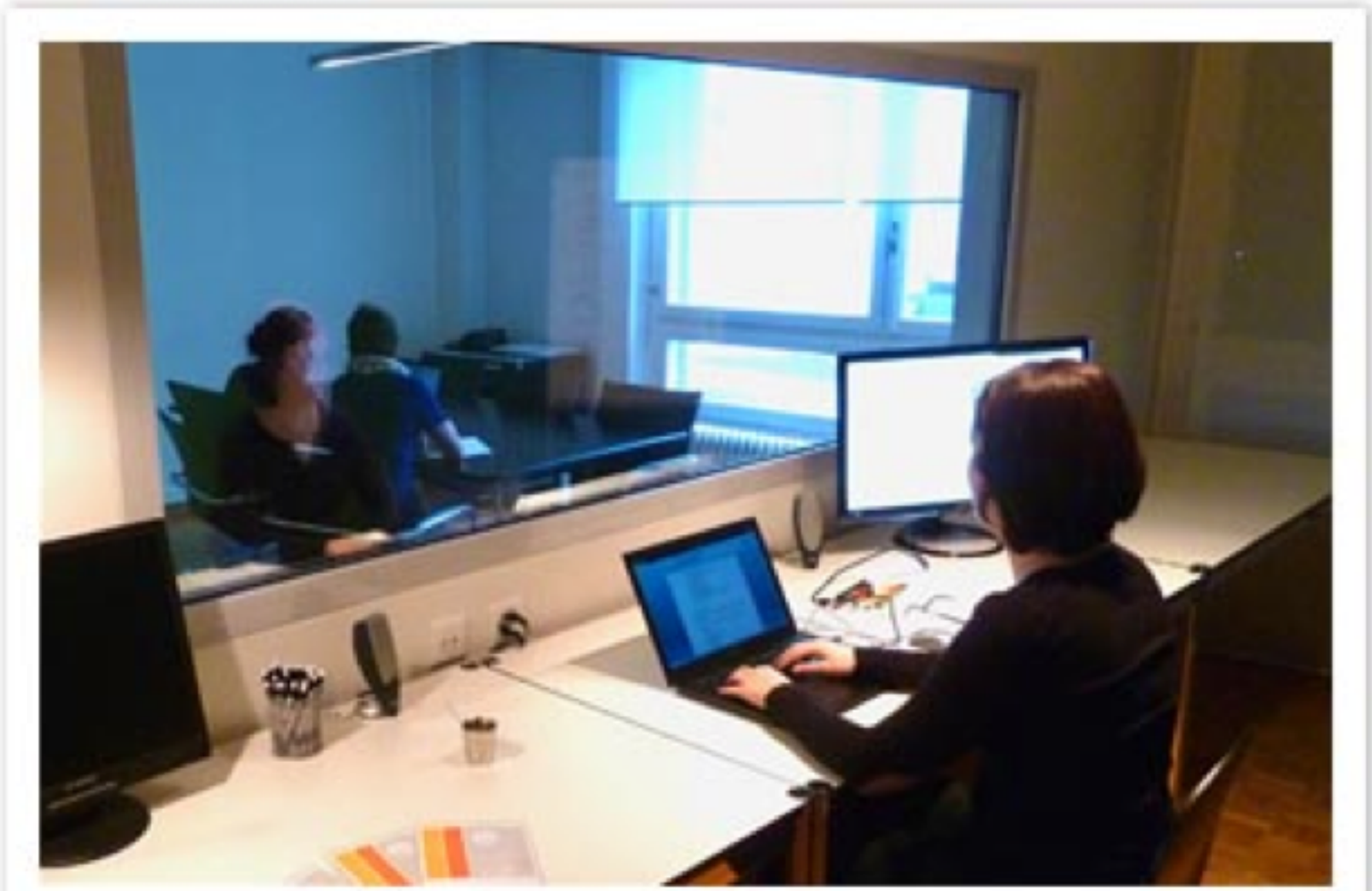
Usability Test: Essentially

- Bring in real users
- Have them complete tasks with your design, while you watch **with your entire team**
- Use a think-aloud protocol, so you can “hear what they are thinking”
- Measure
  - Task completion, task time
  - Satisfaction, problem points, etc.
- Identify problems (major ones | minor ones)
- Provide design suggestions to design/engineering team

Usability Test: Essentially

- Bring in real users
- Have them complete tasks with your design, while you watch **with your entire team**
- Use a think-aloud protocol, so you can “hear what they are thinking”
- Measure
  - Task completion, task time
  - Satisfaction, problem points, etc.
- Identify problems (major ones | minor ones)
- Provide design suggestions to design/engineering team
- Iterate on the design, repeat

Usability Test: Environments



Usability Test: Environments

Usability Test: Environments

Usability Test: Environments

Usability Test: Environments

- Best environment depends on pragmatic considerations, as well as what you're looking for

Usability Test: Environments

- Best environment depends on pragmatic considerations, as well as what you're looking for
  - Do you want your whole team to be able to view?

Usability Test: Environments

- Best environment depends on pragmatic considerations, as well as what you're looking for
  - Do you want your whole team to be able to view?
  - Do you want to be able to review a test?

Usability Test: Environments

- Best environment depends on pragmatic considerations, as well as what you're looking for
  - Do you want your whole team to be able to view?
  - Do you want to be able to review a test?
  - Are interruptions important?

Usability Test: Environments

- Best environment depends on pragmatic considerations, as well as what you're looking for
  - Do you want your whole team to be able to view?
  - Do you want to be able to review a test?
  - Are interruptions important?
  - Repeat use systems, or one-time use systems?

Things you care about in a usability test (aspects of usability)

Things you care about in a usability test (aspects of usability)

- **Learnability/Discoverability:** How easy is it for users to accomplish basic tasks the first time they encounter the design?

Things you care about in a usability test (aspects of usability)

- **Learnability/Discoverability:** 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 the tasks?

Things you care about in a usability test (aspects of usability)

- **Learnability/Discoverability:** 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 the tasks?
- **Memorability:** When users return to a design after a period of not using it, how easily can they reestablish proficiency?

Things you care about in a usability test (aspects of usability)

- **Learnability/Discoverability:** 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 the tasks?
- **Memorability:** When users return to a design after a period of not using it, how easily can they reestablish proficiency?
- **Errors:** How many errors do users make, where are these errors occurring, and how easy is it to recover from these errors?

Things you care about in a usability test (aspects of usability)

- **Learnability/Discoverability:** 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 the tasks?
- **Memorability:** When users return to a design after a period of not using it, how easily can they reestablish proficiency?
- **Errors:** How many errors do users make, where are these errors occurring, and how easy is it to recover from these errors?
- **Satisfaction:** How pleasant is it to use?

Corel Paper Prototype Test

- http://www.youtube.com/watch?v=ppnRQD06ggY

What tasks should we test?

What tasks should we test?

- Tasks are surprisingly challenging to “get right.” A lot of this depends on context.

What tasks should we test?

- Tasks are surprisingly challenging to “get right.” A lot of this depends on context.

- TCSD gives you a way of thinking about this: specifically, focus on **user goals** rather than **system functionality**.

What tasks should we test?

- Tasks are surprisingly challenging to “get right.” A lot of this depends on context.

- TCSD gives you a way of thinking about this: specifically, focus on **user goals** rather than **system functionality**.

- Keywords to good task selection: specific, concrete user goals that describe a complete job (or interaction)

Usability Tasks

- Again, depends a lot on what you're looking for
  - **Specific**: does a task flow work?
  - **Broad**: does your language match the user's mental model/language?

Usability Tasks

- Again, depends a lot on what you're looking for
  - **Specific**: does a task flow work?
  - **Broad**: does your language match the user's mental model/ language?

- Consider "the context of use"
  - What would someone need to do with your tool?
  - Under what circumstances would they be in?
    - (relaxed *vs*. under pressure; non-interrupted *vs.* interrupted constantly)

Usability Tasks
Netflix

- “Rate a few movies”

Usability Tasks
Netflix

- “Rate a few movies”
- “It’s a Friday night, and you’re looking for a movie to watch. What do you do?”

Usability Tasks
Netflix

- “Rate a few movies”
- “It’s a Friday night, and you’re looking for a movie to watch. What do you do?”
- “You’re about to watch `Batman 3’, but want to watch the first two, first. How do you do this?”

Usability Tasks
Netflix

- “Rate a few movies”
- “It’s a Friday night, and you’re looking for a movie to watch. What do you do?”
- “You’re about to watch `Batman 3’, but want to watch the first two, first. How do you do this?”
- “You want to watch Batman 1 through Netflix in your living room with your xbox. How do you set that up?”

Usability Tasks
Netflix

- “Rate a few movies”
- “It’s a Friday night, and you’re looking for a movie to watch. What do you do?”
- “You’re about to watch `Batman 3’, but want to watch the first two, first. How do you do this?”
- “You want to watch Batman 1 through Netflix in your living room with your xbox. How do you set that up?”
- “What do you think about the site?”

Usability Testing Metrics

Usability Testing Metrics

- **Performance**
  - Task success, time on task, errors, efficiency

Usability Testing Metrics

- **Performance**
  - Task success, time on task, errors, efficiency
- **Issue Metrics**
  - Identify issue, issue severity

Usability Testing Metrics

- **Performance**
  - Task success, time on task, errors, efficiency
- **Issue Metrics**
  - Identify issue, issue severity
- **Behavioural**
  - Observe verbal behaviour, issue severity

Usability Testing Metrics

- **Performance**
  - Task success, time on task, errors, efficiency
- **Issue Metrics**
  - Identify issue, issue severity
- **Behavioural**
  - Observe verbal behaviour, issue severity
- **Self-reported**
  - Ease, satisfaction, clarity, comprehension, etc.

Usability Testing: Users

Usability Testing: Users

- **Who?**
  - Depends on your needs
  - Goal: get the people that will be using it, or people that represent those that will be using it

Usability Testing: Users

- **Who?**
  - Depends on your needs
  - Goal: get the people that will be using it, or people that represent those that will be using it
- **How many?**
  - Considerable debate in the community
    - Rule of thumb: ~5

Usability Tests: How many users?

Number of usability problems found with **n** users is described by

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

Where:

- N = total number of usability problems
- L = proportion of problems discovered on 1 user
- Typically, L = 31%

Usability Tests: How many users?

- Main argument: If you have 15 people, it’s better to test 3 designs with 5 users each, rather than one design with 15 people.

Number of usability problems found with **n** users is described by

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

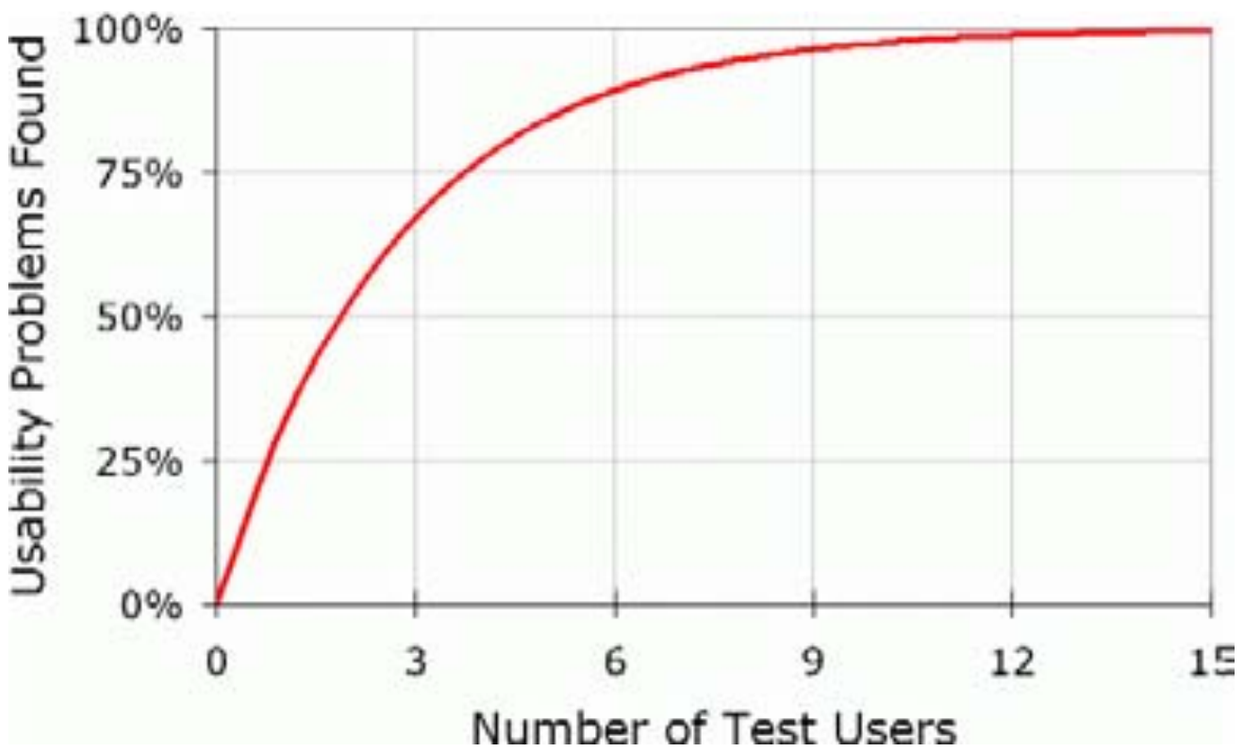
Where:

- N = total number of usability problems
- L = proportion of problems discovered on 1 user
- Typically, L = 31%

Usability Tests: How many users?

- Main argument: If you have 15 people, it's better to test 3 designs with 5 users each, rather than one design with 15 people.

Nielsen & Landauer (1993)



Making sense of your data

- Look for:
  - Big obvious problems
  - Error trends
  - Trends in comments

Making sense of your data

- Look for:
  - Big obvious problems
  - Error trends
  - Trends in comments
- Group issues in terms of severity/priority
  - 1: must fix/brick wall
  - 2: should fix/okay to wait
  - 3: okay as is/could be improved

Making sense of your data

- Affinity diagraming

Making sense of your data

- Discussion with others who watched with you

Usability Testing: Providing Feedback

- Based on your list of issues, provide a small handful of suggestions on how to address the issue

- Depending on the part of the design cycle you are in (early, middle, late), these should be bigger or smaller suggestions

- Provide video “**proof**” of people encountering issues

Three Basic Usability Test Protocols

- Think-Aloud Protocol

- Co-Discovery Protocol

- Conceptual Model Extraction

Think-aloud protocol

Think-aloud protocol

- As participants complete a task, you ask them to report
  - what they are thinking
  - what they are feeling
  - rationale for their actions and decisions
- **Idea:** rather than interpret their actions/lack of action, you can actually understand why they are doing what they are doing

Think-aloud protocol

- *What's weird*:

- People are not normally used to saying things out loud as they work.

- They may also be embarrassed to say things out loud.

Co-discovery Learning protocol

Co-discovery Learning protocol

- **Main idea**: remove the awkwardness of think-aloud

- Two people sit down to complete tasks
- Only one person is allowed to touch the interface
- Monitor their conversation

- **Variation**: use a semi-knowledgable “coach” and a novice (only the novice gets to touch the design)

Conceptual Model Extraction

Conceptual Model Extraction

- Show the design, but don’t say how it works

- Ask the user to explain
  - function of each element
  - how they would perform a particular task

Conceptual Model Extraction

- Initial conceptual model (before they use it)
- Formative conceptual model (after they've used it)
- *Good for*: eliciting a user's understanding before and after use
- *Bad for*: understanding exploration and learning

Acknowledgements

- Tony Tang
- Lora Oehlberg
- Ehud Sharlin
- Frank Maurer
- Saul Greenberg

Course information

- Website
  - GitHub Pages https://silvadasilva.github.io/CPSC481-2019W/en/#!index.md
- Communications
  - Slack https://cpsc481-2019w.slack.com/
- Readings and Slides
  - Posted online at the main website