Introduction to HCI
Fall 2021

Evaluation
Cognitive Walkthrough

Mahmood Jasim
UMass Amherst

nmahyar@umass.edu
https://people.cs.umass.edu/~mjasim/

Logistics

- Milestone 2 Presentation due Nov 3
- Milestone 2 in-class presentation on Nov 4
  - Same procedure as before
- Midterm grades
- Milestone 2 participation
  - 10 points for details on who participated in what component of the milestone
  - Every participant will have to individually write their contributions

Learning goals

- Explain discount usability evaluation methods
- Outline the general procedure for conducting a cognitive walkthrough
- Describe the pros/cons of cognitive walkthroughs

Discount usability testing

- Cheap (thus ‘discount’)
  - No special labs or equipment needed
  - Doesn’t need to involve users directly
  - The more careful you are, the better it gets
- Fast
  - On order of 1 day to apply
  - Standard usability testing may take a week
- Easy to use
  - Can be taught in hours

Types of discount methods

- Cognitive walkthrough: “mental model”
  - Assesses “exploratory learning stage”
  - What mental model does the system image facilitate?
  - Done by non-experts and/or domain experts

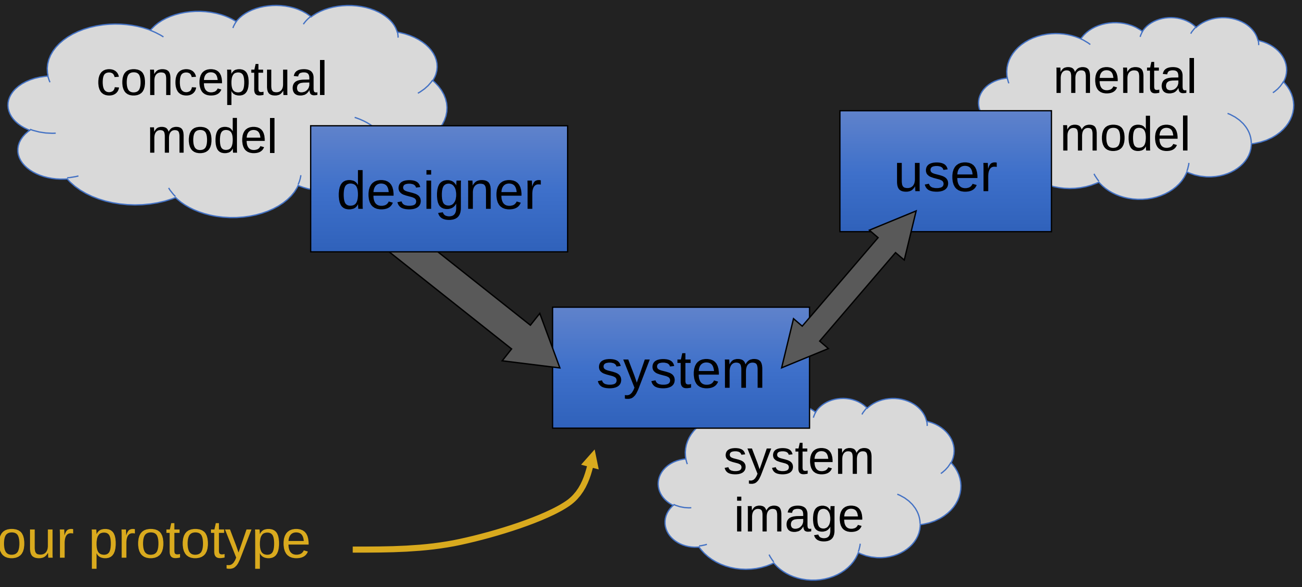
- Heuristic evaluation: “fine tune”
  - Fine-tunes the interface (hi-fi prototypes; deployed systems)
  - HCI professionals apply a list of heuristics while simulating task execution
  - Targets broader use range (including expert)

What is a cognitive walkthrough?

- Cognitive walkthroughs are used to evaluate a product’s usability
- Test conceptual model/interface support for mental models though task examples: task + design = scenario
- Use a “walkthrough” to evaluate a scenario

Cognitive Walkthrough and mental model

- Assessing…
  - Is the conceptual model an effective one?
  - Does the interface design communicate the conceptual model?
  - How well does it support forming a good mental model?



i.e. your prototype

Cognitive walkthrough for exploratory learning

- What for: developing / debugging an interface, without accessing users (which is expensive)

- Tests: how well
    - interface design
    - underlying conceptual model aligns with/sets up the user’s mental model

- Not for: assessing performance at highly skilled, frequently performed tasks; or finding radically new approaches

How to conduct a cognitive walkthrough evaluation?

- Start: with a scenario
  task examples + design = scenario

- 1) break task down into user actions (expected system response)

- 2) perform each step ON the existing interface and ask:
  - Q1: will the user know what to do?
  - Q2: will the user see how to do the action?
  - Q3: will the user correctly understand the system response?

- 3) if you locate a problem, mark it & pretend it has been repaired; then go on to next step.

Cognitive walkthrough

- Possible outputs:
  - Loci & sources of confusion, errors, dead ends
  - Estimates of success rates, error recovery
  - performance/speed less evident
  - Helps to figure out what activity sequences could or should be

- What's required:
  - Task examples: design-independent descriptions of tasks that representative users will want to perform
  - A prototype to provide a design

- Who does it: [theoretically] anyone – usually design team members or expert outside analysts.
  - Can use real users . . . but this makes it a lot less 'discount'

Basic steps

- Step I. Generate “correct”, intended steps to complete a task
- Select a task to be performed and write down all the ‘user actions’, and expected “system responses”
  - ” (a) can they find correct sequence(s) in current version?
    use high-level directives: correct user action = ‘’enter amount of food for pet feeder to dispense”
  - (b) are there mental-model problems even if they use exactly the right sequence?
    get very specific: correct user action = “type ‘36g’ into the text entry box in the middle of the screen

Basic steps

- Step II. Carry out steps, simulating the mindset of your intended user, and note your success OR failure on a log sheet.

- for each step:
  - Q1: ask yourself if user knows what to do?
    are they trying to produce this effect? do they have enough info? etc.
  - Q2: explore – will the user see how to do the step?
    look for the needed action? is it visible? it is obvious how to perform the step?
  - Q3: interpret – will the user correctly understand the system response?
    Is the feedback understandable? Will the interpretation be correct?

- Note: even with an error, user may have progressed if error became apparent. Distinguish this from when user is left with a misunderstanding.

Two approaches to instructing person(s) doing CW

- Approach (a): participant follows the pre-prepared steps and assess according to expected actions/system response
  - at each step, assess using the questions usually best you can do with a paper/low-fidelity prototype (unless it is very complete, has many paths)
  - approach you will probably want to use at this point in your project

- Approach (b): give the CW participant ONLY the higher-level directive(s).
  - e.g., “create an event note with the following attributes. . . ”
  - more exploratory; still use Q1-3 to assess for each step they take
  - BUT - the steps he/she takes might diverge from the list you made – note them down on another action-list sheet. These points should trigger further analysis
  - usually most effective higher fidelity prototypes or released systems

What kinds of problems should I record?

- In a CW you may note many kinds of problems, for example:
  - Problems with particular steps
  - Problems moving between steps
  - Larger problems that involve lots of steps
  - Larger problems that hint at deeper problems with conceptual model/design
  - Small problems that might only apply to unusual users
  - Other kinds of problems that just become apparent while using interface, etc.

- Make note of these as appropriate
  - If you do a lot of CW, you may develop your own template for noting problems that works for you

How do I become good at doing CWs?

- 1. When you’re new to CWs, it’s easy to assume the user will know what to do if YOU know what to do
  - force yourself to imagine what the user might not know

- 2. When asking the questions at each step:
  - really think about what the user could be thinking. . .
  - consider the impact of misconceptions or mistakes that they could have made earlier!

- 3. Perform lots of them!
  - you’ll get better at figuring out what to focus on with practice

What do I do after the CW?

- CWs can be done in teams or individually
  - aggregate and discuss problems
    - possibly found over more than one CW
  - prioritize problems based on severity, likelihood
  - THEN:
  - iterate and fix as required
    - decide on which you can/will address
    - iterate on conceptual model and/or interface design
  - OR write up a report/recommendations → design team
    - if you’re not the one(s) doing the designing

In-class activity

- An example system, CommunityPulse - (https://communitypulse.cs.umass.edu):

- A visual analytic system that utilizes text analysis to extract important topics, emotions and sentiments from community comments and enables civic leaders to explore the comments at multiple levels of granularity.

In-class activity

- Consider these steps from a user’s task
  - Decides to use CommunityPulse
  - Sorts based on excited comments
  - Sorts based on angry comments
  - Sorts based on negative comments
  - Selects the proposal with the most angry comments
  - Goes back to the overview page
  - Selects two top proposals with the largest number of comments

In-class activity

- Work in groups
- https://tinyurl.com/4y6rv5y5
- Write the names of the team members who are present
- Ask yourselves each of Q1-Q3;
    - If answer is NO for any questions:
    - Write down the problem (possible solutions if you have ideas)
    - THEN assume it’s fixed; go on to next step

Optional Reading

- Cognitive Walkthrough
  - https://drive.google.com/file/d/1KboyukZK9QF6_Cm4vDAJ7pmzrfe6Hzoq/view?usp=sharing