Analytical & Empirical Evaluation

Informatics 132

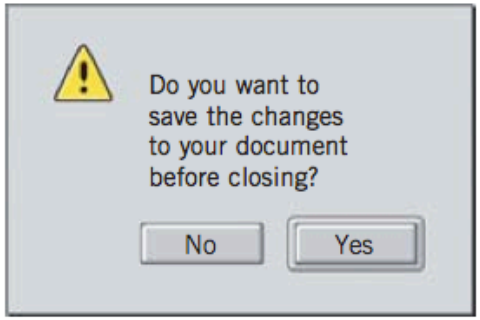
5/22/2012

Errors

An action or omission of action yielding an unintended result.

Most accidents are thought to be caused by what is referred to as *human error*, yet most accidents are actually due to design errors rather than errors of human operation. An understanding of the causes of errors suggests specific design strategies that can greatly reduce their frequency and severity. There are two basic types of errors: slips and mistakes.[1]

Action



CAUSES Changes to repetitive tasks or habits

SOLUTIONS Provide clear and distinctive feedback
Use confirmations for critical tasks
Consider constraints, affordances, and mappings

◂EXAMPLE Confirmations are useful for disrupting behaviors and verifying intent

Knowledge

How to use a fire extinguisher

P pull the pin
A aim the hose at the flame
S squeeze the trigger
S sweep hose from side to side

CAUSES Lack of knowledge and poor communication

SOLUTIONS Use memory and decision aids
Standardize naming and operational conventions
Train using case studies and simulations

◂EXAMPLE Memory mnemonics are useful strategies for remembering critical information in emergency situations

TODAY

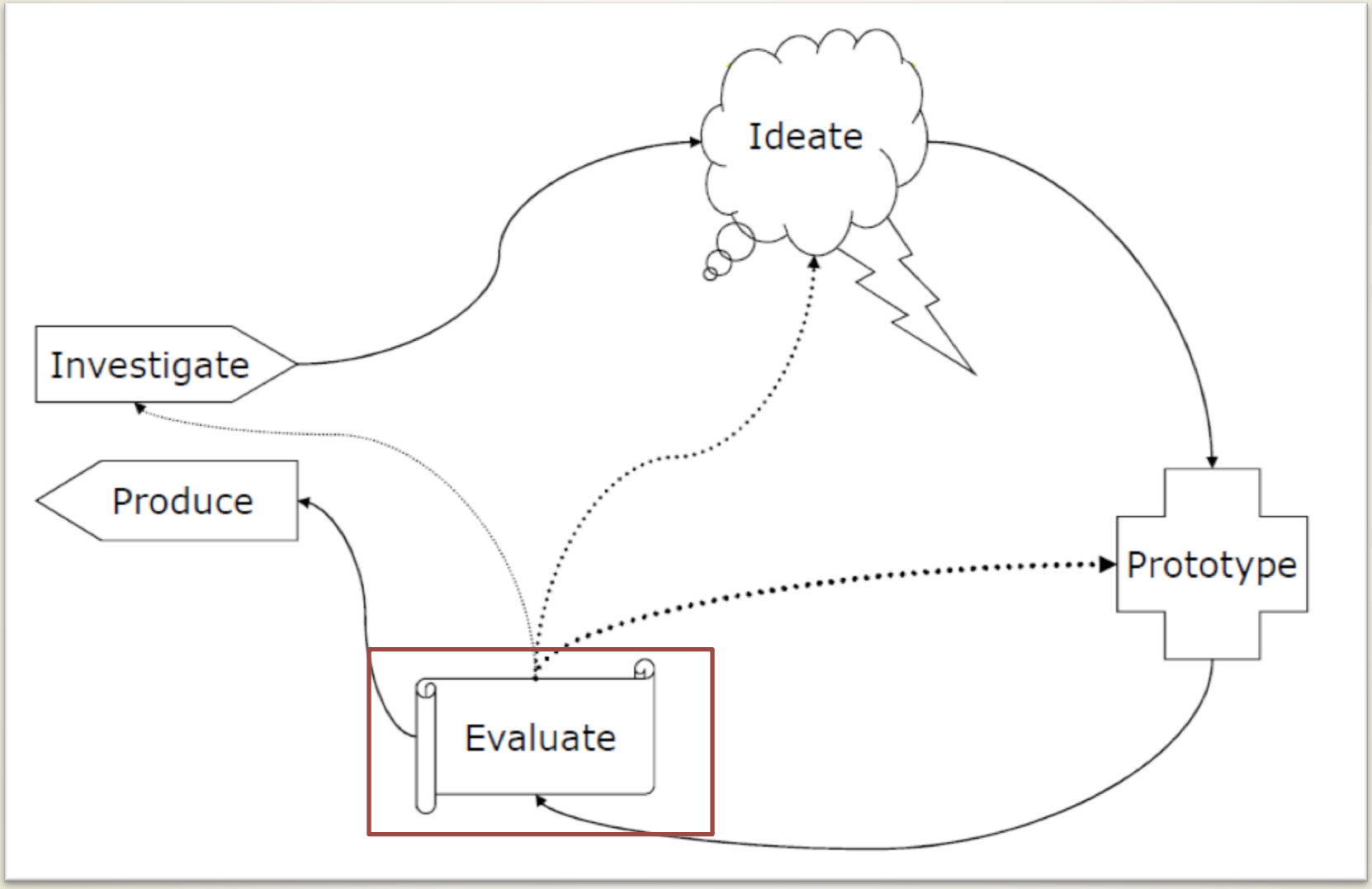
- Evaluation
- Due: A3 – Paper Prototyping

UPCOMING

- Friday:
  Group Project Time
- Monday:
  Memorial Day, No Class
- Wednesday:
  HCI in the Real World – Future Trends

Evaluation Overview

In the Design Process…



Many Approaches to Evaluation

Usability goals:

- Effectiveness
- Efficiency
- Safety
- Utility
- Learnability
- Memorability

User experience goals:

- Satisfying
- Pleasurable
- Rewarding
- Fun
- Provocative
- …

Consider your project. What are your usability goals? What are your user experience goals? How would you define and operationalize these goals?

Why and Where to Evaluate

- Why?
  - Feedback on design directions and ideas
  - Discover major issues
  - (Help to) resolve disagreements
- Where?
  - In laboratory (controlled)
  - In natural settings (uncontrolled)

When to Evaluate

- Early design of an artifact
- Evaluation of working prototype
- Refining or maintaining a product
- Competitive comparison between two products
- Exploring an new design concept
- Demonstrate performance for a procurement contract

Types of Evaluation

- Analytic (design judgment – users not involved)
  - Often called “discount evaluations”
  - Standards enforcement
  - Heuristic evaluations
  - Cognitive walkthroughs
- Empirical (involves users)
  - Usability testing
  - Field studies
  - Click-through studies

Analytical Evaluation

Analytical Evaluation

- Cognitive Walkthrough
  - Have experts analyze your prototype in a detailed way to understand how users will understand it
  - Best for understanding novel use, not expert use
  - http://en.wikipedia.org/wiki/Cognitive_walkthrough
- Heuristic Evaluation

Cognitive Walkthrough

- Uses a small number of HCI experts to evaluate a design for ease of learning, especially via exploration
- Analogy to code walkthrough (Polson, Lewis, et al. at UC Boulder)

Cognitive Walkthrough

- Requires prototype or fairly detailed description
- Requires a description of the user task to be analyzed
- Requires a complete, written list of actions necessary to complete the task
- Requires an indication of who the users are and their assumed knowledge

Procedure

- Define required inputs
- Walk through action sequences for task
- Record critical information & obtain believability story

Inputs

(1) Select Interaction Task

- Task should be a one that would be common or typical for a potential user
- Should be representative of what users would want to do with the system

(2) Define interaction action sequence

- Tasks should be broken down until any further division yields obvious subtasks
- E.g., type “run” at prompt NOT type “r”; type “u” etc.

Inputs

- (3) Identify the users
  - Educate the HCI experts on the domain knowledge, experience, and characteristics of the user
  - Give evaluators a perspective from which to evaluate the system
- (4) Prototype
  - Need not be functional but…
  - Must be at a level of detail where any action necessary to complete the task is defined

Doing the walkthrough

- Address each step of task sequence in turn
- Formulate a believability story
  - Answer 4 questions
  - Include justification for each answer based on the interface, knowledge of HCI, and understanding of users

Question 1: Will the user be trying to produce whatever effect the action has?

- Common supporting evidence
  - It is part of their original task
  - They have experience using the system
  - The system tells them to do it
- No supporting evidence?
  - Construct a failure story
  - Why would the user notbe trying to do this?

Question 2: Will the user be able to notice that the correct action is available?

- Common supporting evidence
  - Known through experience
  - Visible device, such as a button
  - Visible representation of an action, such as a menu entry
- No supporting evidence?
  - Why would the user not notice such an action is available?

Question 3: Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?

- Common supporting evidence
  - Based on past experience with similar interactions
  - The interface provides a prompt or label that connects the action to what he/she is trying to do
  - All other actions look wrong
- If not, why not?

Question 4: After the action is taken, will the user understand the feedback given?

- Common supporting evidence
  - Past experience with similar interactions
  - Recognizing a connection between the system response and what the user was trying to do
- If not, why not?

Analytical Evaluation

- Cognitive Walkthrough
- Heuristic Evaluation
  - Have usability experts go through your prototype to uncover common usability problems

Heuristic Evaluation

- Developed by Jakob Nielsen
- Helps find usability problems in a UI design
- Small set (3-5) of evaluators examine UI
  - independently check for compliance with usability principles (“heuristics”)
  - different evaluators will find different problems
  - evaluators only communicate afterwards
    - findings are then aggregated
- Can perform on working UI or on sketches

Heuristic Evaluation Process

- Evaluators go through UI several times
  - inspect various dialogue elements
  - compare with list of usability principles
  - consider other principles/results that come to mind
- Usability principles
  - Nielsen's "heuristics"
  - Supplementary list of category-specific heuristics
    - competitive analysis & user testing of existing products
- Use violations to redesign/fix problems

Heuristics (Nielsen, 1994)

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation

More info here:
http://www.nngroup.com/articles/ten-usability-heuristics/

Phases of Heuristic Evaluation

1. Pre-evaluation training
   - give evaluators needed domain knowledge & information on the scenario
2. Evaluation
   - individuals evaluates UI & makes list of problems
3. Severity rating
   - determine how severe each problem is
4. Aggregation
   - group meets & aggregates problems (w/ ratings)
5. Debriefing
   - discuss the outcome with design team

How to Perform Evaluation

- At least two passes for each evaluator (3-5 people)
  - first to get feel for flow and scope of system
  - second to focus on specific elements
- If system is walk-up-and-use or evaluators are domain experts, no assistance needed
  - otherwise might supply evaluators with scenarios
- Each evaluator produces list of problems
  - explain why with reference to heuristic or other information
  - be specific & list each problem separately

Example Errors from Evaluators

- Can't copy info from one window to another
  - Violates: “Minimize the user’s memory load” (H6)
  - Fix: allow copying
- Typography uses different fonts in 3 dialog boxes
  - Violates “Consistency and standards” (H4)
  - slows users down
  - probably wouldn’t be found by user testing
  - Fix: pick a single format for entire interface

Severity Rating

- Used to allocate resources to fix problems
- Estimates of need for more usability efforts
- Combination of
  - frequency
  - impact
  - persistence (one time or repeating)
- Should be calculated after all evals. are in
- Should be done independently by all judges

Severity Ratings (cont.)

0 - don't agree that this is a usability problem

1 - cosmetic problem

2 - minor usability problem

3 - major usability problem; important to fix

4 - usability catastrophe; imperative to fix

Debriefing

- Conduct with evaluators, observers, and development team members
- Discuss general characteristics of UI
- Suggest potential improvements to address major usability problems
- Dev. team rates how hard things are to fix
- Make it a brainstorming session
  - little criticism until end of session

Severity Ratings Example

#22 – Inconsistent “Save” Terminology

[H4 Consistency] [Severity 3]

Description: The interface used the string "Save" on the first screen for saving the user's file, but used the string "Write file" on the second screen. Users may be confused by this different terminology for the same function.

HE vs. User Testing

- HE is much faster
  - 1-2 hours each evaluator vs. days-weeks
- HE doesn't require interpreting user's actions
- User testing is far more accurate (by def.)
  - takes into account actual users and tasks
  - HE may miss problems & find "false positives"
- Good to alternate between HE & user testing
  - Find different problems
  - Don't waste participants

Empirical Evaluation

Empirical Evaluations

- Usability Evaluations
  - Typically in lab
  - Tests usability metrics
  - Earlier in design process
- Field Studies
  - Out in the “real world”
  - Tests user experience metrics
  - Later in the design process

Usability Evaluations

- Testing Plans
- Usability Lab Studies

- Example: how do we evaluate this site?
  - http://historywired.si.edu/index.html

Usability Test Plan

- Objectives
- User profile
- Method
- Task list
- Evaluation measures

- From Rubin, J. (1994). Handbook of Usability Testing. New York: Wiley

A. Test Objectives

Create very specific objectives for your evaluation

Poor examples:

- Can users identify trends?
- Is the user-interface usable?

Good examples:

- Can users employ the slider associated with the timeline to identify outlying dates?
- Can users select filters and select colors so that the relationship between X & Y is readily seen?
- Can users find material more quickly in the visual or textual version of the table of contents?

B. User Profile

- Enumerate attributes for your target users and select users that meet the profile
- Can be based on people who fit your persona types
- Example:
  - Age: 25-30
  - Gender: 50% men; 50% women
  - Computer skills: Daily use of a web browser
  - Background: Intro class in statistics (STATS-101)
  - Interests: Track stocks online

For your projects …

- What are good test objectives?
- What is a good user profile?

C. Method

- Many different approaches to structuring a test design

- The ‘best’ approach depends on
  - Resources (time & money)
  - Objectives of the study

D. Task List

- A detailed list of tasks
- Each task
  - Description:
    What you prompt users with?
  - Machine state:
    Where users begin from?
  - Successful completion:
    When is the task completed?

Task Unit

- Description:
  - Name three important events that took place in the 1770s in America
- Machine state:
  - Timeline is set to the 1980s
  - Article on space shuttle is being shown
- Successful completion:
  - Participants verbally reports the names of three events by using the timeline

For your project …

- What is a good task unit?
- Description
- Starting machine state
- Successful completion

E. Evaluation Measures

- Quantitative count data
  - Time, errors, confusions, breakdowns, workarounds, success/failure.
- Your observations
  - Notes about where, when, why and how the above things occurred.
- Users' comments and feedback
  - Often a questionnaire is used at the end
  - User quotes "I LOVE it, except when it crashes"

For your projects …

- What could be good evaluation measures?

Running the Test

- Introduce the test
  - "The interface is being tested, not you."
  - "I didn't design or build this; I was just asked to find out what the problems are."
- Prompt them to continually think-aloud
- Observe task times, errors, confusions, breakdowns, workarounds, and success/failure
  - Make notes, video-record, audio-record

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.

Answering a User's Questions

- Basically, you really, really shouldn't
  - You (the researcher) 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 them
  - Suggestion: “Why don't you try something else?”

Being a Good Moderator

- Spend almost all your time listening, observing carefully, and planning what to say (or not say) next
- 'Encourage' participants in a neutral fashion
- When people become quiet say
  - "Can you keep talking?"

Think Aloud Prompts

- “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?”

Debrief

- Tell them more details about what you were interested in discovering. Emphasize their contribution.
- Answer any questions they have
- Now you can show them how to accomplish tasks that they had failures on
- Thank them for their time
- Pay them $$! :)

Human Subject Ethics

Being in a user test can be uncomfortable for some

Guidelines:

- Acknowledge that that system is being tested, not the participant (remind repeatedly)
- Tell the participant that she is free to leave at any time
- Reveal who is watching & what is being recorded
- Do not report results such that a participant is identified
- Avoid telling the participant that he is making mistakes or doing things wrong
- Acknowledge participants efforts but in a neutral fashion

Bottom line: Treat people with great respect

Tips for Usability Evaluations

- Keep it simple
- Keep your objectives specific
- Be consistent with all participants
  - Create a script and follow it carefully
- Conduct a pilot test to uncover problems
- Have detailed plan for analyzing the data

Field Studies

- Give users a functional prototype of your system and let them use naturally for a set amount of time
  - Also called "*in situ*" studies, "real world deployments," or studies "in the wild"

Considerations

- How long? / “The Novelty Effect”
- How many people?
- How to recruit?
- How to retain participants?
- Experimental or exploratory?
- What data to collect?

How long? / The Novelty Effect

- Any new technology will get the most use when it first is introduced, then interest wanes
- How long?
  - Nathan Eagle (MIT) estimates it is about 2 weeks
  - May be longer
- Field deployments should last long enough for the novelty effect to wear off to understand more realistic use

Participants

- The more the better, but think about your resources
- Try to recruit as diverse of sample as possible
  - Think about recruiting proportional to your personas
  - If experimental, may want to recruit homogenous sample to reduce variables
- Recruitment – internet ads, word of mouth, “snowball” sampling
  - Consider offering payment to attract and retain

Experimental or Exploratory?

- Comparing new product against an old one can be very powerful
  - Your new product: experimental
  - Existing product: control
  - “Participants using product Y preferred it over product X 9 times out of 10”
- Exploratory: just give out your product and see what happens
  - Often better in initial stage evaluation

What data to collect?

- Pre- and post- evaluation interviews and surveys
  - Depending on length of study, consider mid-study interviews as well
- Log usage data (if possible)
  - Automatically, with a computer script
  - Time stamped
- Diary entries after each use
  - Can automatically be prompted after usage

Summary: Empirical Evaluations

- Usability Evaluations
  - Typically in lab
  - Tests usability metrics
  - Earlier in design process
- Field Studies
  - Out in the “real world”
  - Tests user experience metrics
  - Later in the design process