User-Centered Design

A couple notes about the course….

- First assignment due next Monday
  - Shouldn’t be too much work
  - All the details at the course site!
  - Submit via Gradescope!
- Second assignment due the following Monday
  - Will require setting up a Zoom call with someone outside the course! (A target user.)
  - Encourage you to schedule that for next Wednesday or Thursday
    - …and thus encourage you to send the invite today!
    - We’ll discuss this more in class

Plan for today

- Reading reflection
  - 15 minutes
- Discuss Assignment 2
  - 5 minutes
- Plan for Assignment 2, with partners
  - 20 minutes
- Break
  - 10 minutes
- Lecture
  - 30 minutes

Reading Reflection

Discuss in groups

- The reading broke usability down into subgoals:
  - effective to use (effectiveness)
  - efficient to use (efficiency)
  - safe to use (safety)
  - having good utility (utility)
  - easy to learn (learnability)
  - easy to remember how to use (memorability)
- How many of these had you thought of before? How many were new to you (as usability goals)?
- For the ones that were new to you, how would you define a metric that measures the subgoal for a PL task? Or a programming tools task?
- Brainstorm examples of “dark patterns” in PL.

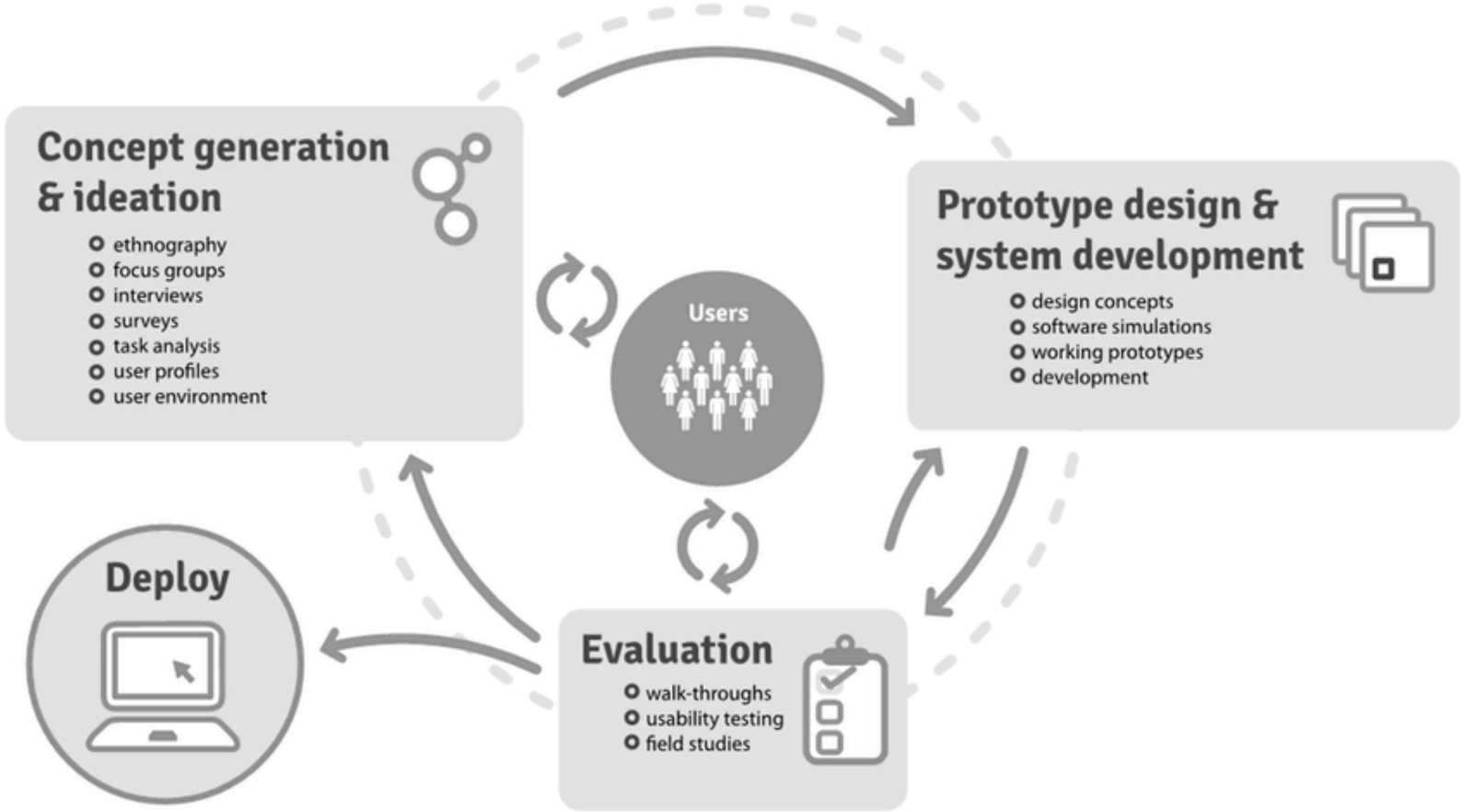
Need Finding Assignment

- I know we haven't started studying need finding techniques yet!
- But you're going to need to schedule a meeting or Zoom call with a non-classmate, and that can take a while, so I encourage you to do it now. I recommend scheduling it for sometime between Wednesday and Friday of next week.
  - That way you should…
    - …have already read the need finding readings
    - …still have time to think about the results before the writeup is due the following Monday
- Details: see the website
- This is a partner assignment, so find a partner now!
  - This is just a HW partner, not your partner for the final project!
- I highly encourage you to have reached out to someone about scheduling a call by the end of class session! Or at least by the end of this week.

Poll

Before this week's readings, I thought applying HCI to PL design meant doing user studies to evaluate PLs after they're done.

Most important slide today



Visual from McCurdie et al. 2012

Table 6. Ranking of Importance and Frequency of Most Commonly Used UCD Methods

	Ranking						
	1	**2**	**3**	**4**	**5**	**Average Ranking**	**Frequency**
Field studies (include contextual inquiry)	12	6	5	2	1	2.00	28
User requirements analysis	3	3	0	0	1	2.00	7
Iterative design	17	21	9	5	2	2.15	65
Usability evaluation	12	8	10	7	1	2.39	43
Task analysis	6	8	6	7	1	2.61	34
Focus groups	5	2	2	1	4	2.79	16
Formal heuristic evaluation	3	2	5	2	2	2.86	15
User interviews	2	0	3	4	0	3.00	11
Prototype without user testing	1	3	5	4	1	3.07	15
Surveys	0	2	2	1	1	3.17	9
Informal expert review	4	6	3	10	6	3.28	31
Card sorting	0	1	1	0	1	3.33	5
Participatory design	1	0	1	2	1	3.40	7
No code/too sketchy to be categorized							64

A Survey of User-Centered Design Practice, Vredenburg et al.

Think-Pair-Share

Define *user-centered design*

Now define *human-centered design*

Human-Centered vs. User Centered Take 1

General human characteristics vs.
characteristics of a particular audience of users

Human-Centered vs. User Centered Take 2

"The paper reviews recent approaches to user-centered IS design and concludes that these methods are targeted at the closure of technology-centered problems, rather than the investigation of suitable changes to a system of human activity supported by technology."

- HUMAN-CENTERED VS. USER-CENTERED APPROACHES TO INFORMATION SYSTEM DESIGN, Gasson

Human-Centered vs. User Centered Take 2

"The pap
design ar
the closu
the invest

- HUMAN-CENTERED VS. USER-CENTERED APPROACHES TO INFORM

"…questions the traditional interpretation of human-centeredness found in the HCI and IS literatures, as the production of a usable system design. The author critiques a number of recent developments in human-centered design methods, to examine the extent to which their focus on stakeholders as simply users of technology limits the extent to which they can support organizational work.

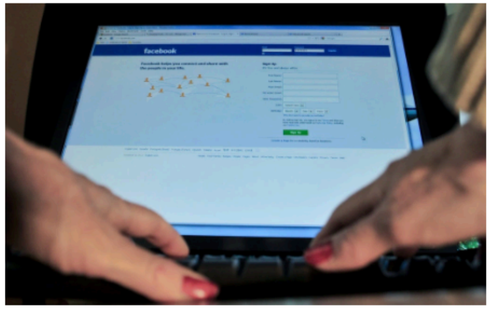
"The discourse of Interaction Design starts with a concept of "the computer" (or computer-based technology) and only then considers the context of the human-computer interaction. This has the effect of moving the design model back to the historically unitary focus of HCI: a single technology user, moving towards closure of a single, task-related problem, in isolation from the social world of work that surrounds them. Interaction is thus reduced to interface. "

The New York Times

Facebook Tinkers With Users' Emotions in News Feed Experiment, Stirring Outcry

By Vindu Goel

June 29, 2014



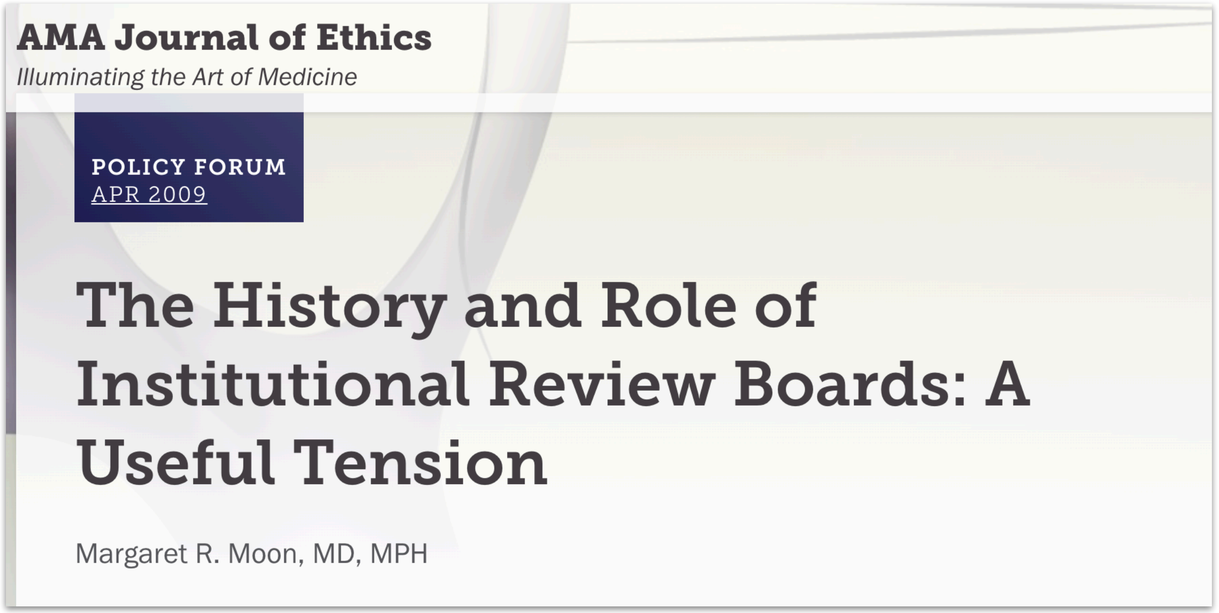
Facebook revealed that it had altered the news feeds of over half a million users in its study.
Karen Bleier/Agence France-Presse — Getty Images

To Facebook, we are all lab rats.

Facebook routinely adjusts its users' news feeds — testing out the number of ads they see or the size of photos that appear — often without their knowledge. It is all for the purpose, the company says, of creating a more alluring and useful product.

But last week, Facebook revealed that it had manipulated the news feeds of over half a million randomly selected users to change the

Serious ethics considerations both in user studies and in the broader design process.



AMA Journal of Ethics
Illuminating the Art of Medicine

POLICY FORUM
APR 2009

The History and Role of Institutional Review Boards: A Useful Tension

Margaret R. Moon, MD, MPH

The history of human-subjects research is replete with horrid examples of what happens when investigators fail to respect humans as ends in themselves. Even after the Nuremberg trials exposed the Nazi war crimes and the Nuremberg Code provided a clear statement of standards for research on human subjects, unethical research programs continued to be designed and conducted [2]. In the United States, the Willowbrook study of hepatitis transmission in a hospital for mentally impaired children, Tuskegee Syphilis Study, Fernald State School trials using radioactive minerals in impaired children, and Jewish Chronic Disease Hospital case in which chronically ill patients were injected with cancer cells to monitor rejection, are infamous examples of egregiously unethical research designed and conducted long after the Nuremberg Code was in place. In each of these studies, investigators were confident that the ends of research justified the means.

The National Research Act of 1974, passed in response to growing concern about the ethics violations in research, created the National Commission for the Protection of Human Subjects of Biomedical and Behavioral Research. The Belmont Report of 1974 was the commission's summary of the ethical principles that form the basis of acceptable human-subjects research, and the three foundational Belmont principles were:

Respect for persons. This principle includes both respect for the autonomy of human subjects and the importance of protecting vulnerable individuals.

Belmont Report Principles

Respect for persons. This principle includes both respect for the autonomy of human subjects and the importance of protecting vulnerable individuals.

Beneficence. More than just promotion of well-being, the duty of beneficence requires that research maximize the benefit-to-harm ratio for individual subjects and for the research program as a whole.

Justice. Justice in research focuses on the duty to assign the burden and benefits of research fairly.

IRB Review

In case you're thinking of publishing the work associated with your final project, it may already be time to start thinking about this

- For non-risky study designs (including much but not all of the work in PL+HCI)
  - "Exempt" status (doesn't mean not writing it up!)
  - At Berkeley, approx. 2 week review times

…but how should we observe users?

Method	Phases supported	Key benefits	Challenges and limitations
Interviews	Requirements, Creation	Gathers open-ended qualitative data from experts	Depends on skill of interviewer and selection of participants; results may not generalize
Surveys	Requirements, Creation	Assesses opinions among a broad audience; can generalize interview results	Output is subjective; may not reflect reality
Corpus studies	Requirements, Creation	Assesses incidence of problems or applicability of solutions in a large dataset	Depends on appropriate datasets and efficient methods of analysis
Natural programming	Requirements, Creation	Obtains insights from people without biasing them toward preferred solutions	Data may be biased toward participants' prior experiences
Rapid prototyping	Requirements, Creation	Facilitates efficient design space exploration	Lack of fidelity in prototypes may hide faults
Programming language theory	Requirements, Creation, Evaluation	Ensures sound designs	High cost; applying formal methods too early may limit ability to iterate, but applying too late can waste time on unsound approaches
Software engineering theory	Requirements, Creation, Evaluation	Improves practicality of designs	Unclear how to prioritize recommendations when they conflict
Qualitative user studies	Requirements, Creation, Evaluation	High-bandwidth method to obtain insight on user behavior when using systems	Results may not generalize; Results depend on skills of experimenter and participants
Case studies	Evaluation	Tests applicability of systems to real-world cases; allows in-depth explorations of real-world difficulties	Requires finding appropriate cases; generalizability may be limited
Expert evaluation	Evaluation	Benefit from experience acquired by experts	Biased by opinions of experts, which may not reflect real-world implications of the design
Performance evaluation	Evaluation	Reproducible way of assessing performance	Results depend heavily on selection of test suite
User experiments	Evaluation	Quantitative comparison of human performance across systems	Results may not generalize to non-trivial tasks, other kinds of participants, expert users, long-term use, or use on large systems
Formalism and proof	Requirements, Creation, Evaluation	Provides definitive evidence of safety	Results are limited to the specific theorems proven

Interdisciplinary programming language design, Coblenz et al.

…but how should we observe users?

Expressivity analysis (the standard "usability" eval for much of the history of PL)

Method	Phases supported	Key benefits	Challenges and limitations
Interviews	Requirements, Creation	Gathers open-ended qualitative data from experts	Depends on skill of interviewer and selection of participants; results may not generalize
Surveys	Requirements, Creation	Assesses opinions among a broad audience; can generalize interview results	Output is subjective; may not reflect reality
Corpus studies	Requirements, Creation	Assesses incidence of problems or applicability of solutions in a large dataset	Depends on appropriate datasets and efficient methods of analysis
Natural programming	Requirements, Creation	Obtains insights from people without biasing them toward preferred solutions	Data may be biased toward participants' prior experiences
Rapid prototyping	Requirements, Creation	Facilitates efficient design space exploration	Lack of fidelity in prototypes may hide faults
Programming language theory	Requirements, Creation, Evaluation	Ensures sound designs	High cost; applying formal methods too early may limit ability to iterate, but applying too late can waste time on unsound approaches
Software engineering theory	Requirements, Creation, Evaluation	Improves practicality of designs	Unclear how to prioritize recommendations when they conflict
Qualitative user studies	Requirements, Creation, Evaluation	High-bandwidth method to obtain sight on user behavior when using tems	
Case studies	Evaluation	Tests applicability of systems to real-world cases; allows in-depth explorations of real-world difficulties	Requires finding appropriate cases; generalizability may be limited
Expert evaluation	Evaluation	Benefit from experience acquired by experts	Biased by opinions of experts, which may not reflect real-world implications of the design
Performance evaluation	Evaluation	Reproducible way of assessing performance	
User experiments	Evaluation	Quantitative comparison of human formance across systems	other kinds of participants, expert users, long-term use, or use on large systems
Formalism and proof	Requirements, Creation, Evaluation	Provides definitive evidence of safety	Results are limited to the specific theorems proven

Why?

Never appropriate to make usability claims here

Interdisciplinary programming language design, Coblenz et al.

List not exhaustive. E.g., Observation

…but how should we observe users?

Method	Phases supported	Key benefits	Challenges and limitations
Interviews	Requirements, Creation	Gathers open-ended qualitative data from experts	Depends on skill of interviewer and selection of participants; results may not generalize
Surveys	Requirements, Creation	Assesses opinions among a broad audience; can generalize interview results	Output is subjective; may not reflect reality
Corpus studies	Requirements, Creation	Assesses incidence of problems or applicability of solutions in a large dataset	Depends on appropriate datasets and efficient methods of analysis
Natural programming	Requirements, Creation	Obtains insights from people without biasing them toward preferred solutions	Data may be biased toward participants' prior experiences
Rapid prototyping	Requirements, Creation	Facilitates efficient design space exploration	Lack of fidelity in prototypes may hide faults
Programming language theory	Requirements, Creation, Evaluation	Ensures sound designs	High cost; applying formal methods too early may limit ability to iterate, but applying too late can waste time on unsound approaches
Software engineering theory	Requirements, Creation, Evaluation	Improves practicality of designs	Unclear how to prioritize recommendations when they conflict
Qualitative user studies	Requirements, Creation, Evaluation	High-bandwidth method to obtain insight on user behavior when using systems	Results may not generalize; Results depend on skills of experimenter and participants
Case studies	Evaluation	Tests applicability of systems to real-world cases; allows in-depth explorations of real-world difficulties	Requires finding appropriate cases; generalizability may be limited
Expert evaluation	Evaluation	Benefit from experience acquired by experts	Biased by opinions of experts, which may not reflect real-world implications of the design
Performance evaluation	Evaluation	Reproducible way of assessing performance	Results depend heavily on selection of test suite
User experiments	Evaluation	Quantitative comparison of human performance across systems	Results may not generalize to non-trivial tasks, other kinds of participants, expert users, long-term use, or use on large systems
Formalism and proof	Requirements, Creation, Evaluation	Provides definitive evidence of safety	Results are limited to the specific theorems proven

Interdisciplinary programming language design, Coblenz et al.

Useful generalizations, but take tables like this with a grain of salt!

Method	Phases supported	Key benefits	Challenges and limitations
Interviews	Requirements,	Gathers open-ended qualitative data	Depends on skill of interviewer and selection of participants; results may not generalize
Surveys		lts	Output is subjective; may not reflect reality
Corpus studies	Requirements, Creation	Assesses incidence of problems or applicability of solutions in a large dataset	Depends on appropriate datasets and efficient methods of analysis
Natural programming		out	Data may be biased toward participants' prior experiences
Rapid prototyping	Requirements, Creation	Facilitates efficient design space exploration	Lack of fidelity in prototypes may hide faults
Programming language theory	Requirements, Creation, Evaluation	Ensures sound designs	High cost; applying formal methods too early may limit ability to iterate, but applying too late can waste time on unsound approaches
Software engineering theory	Requirements, Creation, Evaluation	Improves practicality of designs	Unclear how to prioritize recommendations when they conflict
Qualitative user studies	Requirements, Creation, Evaluation	High-bandwidth method to obtain insight on user behavior when using systems	Results may not generalize; Results depend on skills of experimenter and participants
Case studies	Evaluation	Tests applicability of systems to real-world cases; allows in-depth explorations of real-world difficulties	Requires finding appropriate cases; generalizability may be limited
Expert evaluation	Evaluation	Benefit from experience acquired by experts	Biased by opinions of experts, which may not reflect real-world implications of the design
Performance evaluation	Evaluation	Reproducible way of assessing performance	Results depend heavily on selection of test suite
User experiments	Evaluation	Quantitative comparison of human performance across systems	Results may not generalize to non-trivial tasks, other kinds of participants, expert users, long-term use, or use on large systems
Formalism and proof	Requirements, Creation, Evaluation	Provides definitive evidence of safety	Results are limited to the specific theorems proven

Could you design a survey to produce objective measures?

Could you remove programming education confound?

Interdisciplinary programming language design, Coblenz et al.

Need Finding Assignment

- I know we haven't started studying need finding techniques yet!
- But you're going to need to schedule a meeting or Zoom call with a non-classmate, and that can take a while, so I encourage you to do it now. I recommend scheduling it for sometime between Wednesday and Friday of next week.
  - That way you should…
    - …have already read the need finding readings
    - …still have time to think about the results before the writeup is due the following Monday
- Details: see the website
- This is a partner assignment, so find a partner now!
  - This is just a HW partner, not your partner for the final project!
- I highly encourage you to have reached out to someone about scheduling a call by the end of class session! Or at least by the end of this week.