Needfinding: Lessons for Programming Language Design

Veronica A. Rivera (varivera@stanford.edu)

Postdoctoral Researcher in CS

CS 343s: April 10, 2024

Who I am and what I do

Research on digital safety of marginalized and vulnerable populations

- Harms & protective behaviors of people who engage in algorithmically-mediated offline interactions (gig workers, sex workers, online daters)
- S&P advice given to people in the majority world
- How to design tools/online spaces to support post-harm support for survivors

HARVEY MUDD COLLEGE

BS in Computer Science and Math

UC SANTA CRUZ

PhD in Computational Media (HCI)

HAI Stanford University Human-Centered Artificial Intelligence

Stanford | McCoy Family Center for Ethics in Society SCHOOL OF HUMANITIES & SCIENCES

Working with philosophers to create ethics curriculum for undergraduate CS courses

What is design?

"[Design is] a plan for arranging elements in such a way as to best accomplish a particular **purpose**." Charles Eames

Core design skills

To **frame**, or reframe, the problem and objective

To create and **envision** alternatives

To **select** from those alternatives

Understanding people

To **synthesize** a solution from all the relevant constraints

Synthesis

To visualize and **prototype** the intended solution

Iterative prototyping

Bill Moggridge

Design thinking

Empathize
(Needfinding)

Research users' needs to understand the problem and set aside your assumptions about the world

Define

Analyze and synthesize information about users to define the core problems. Create personas to keep your approach human-centered

Ideate

Brainstorm alternative ways to view the problem and identify innovate solutions

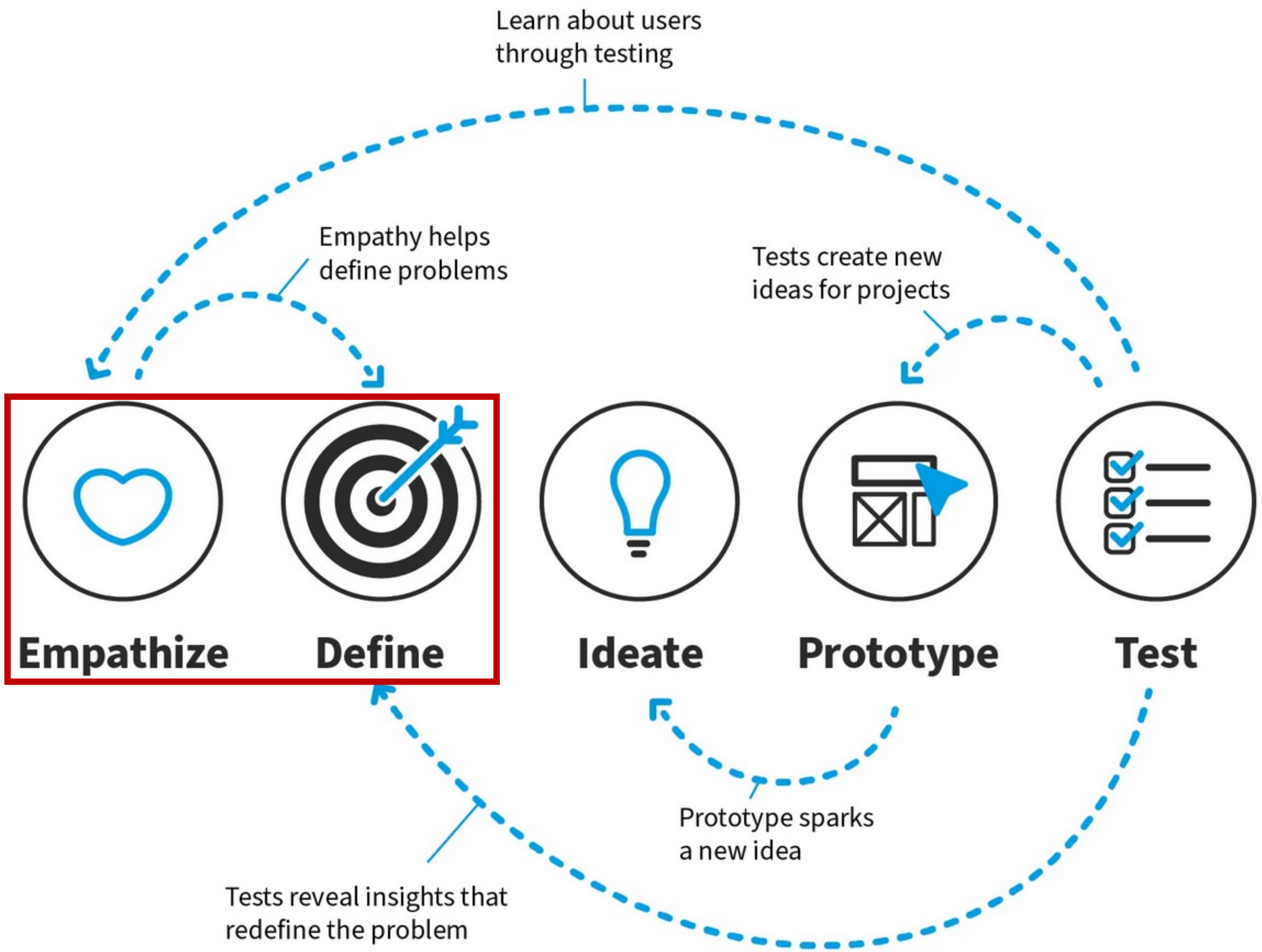
Prototype

Identify the best possible solution for your problem and create quick "drafts" – scaled down, inexpensive versions of a product

Test

Try out your prototype with real users. This might lead to new insights that lead you to reframe the problem & solutions

https://www.interaction-design.org/literature/topics/design-thinking.


Learn about users through testing
Empathy helps define problems
Tests create new ideas for projects
Empathize
Define
Ideate
Prototype
Test
Prototype sparks a new idea
Tests reveal insights that redefine the problem

https://www.interaction-design.org/literature/topics/design-thinking.

Why understand users?

- You are not everyone, you are not the only user
- Engineers often make default choices that leave people out:
  - Automobile crash test dummies are designed to match the anatomy of 70kg adult man, excluding much of the population.
  - Face recognition technologies fail to account for diverse race and gender profiles
  - Assumption that smartphones have only a single user, which may not hold for parents sharing devices with their children or people in lower-income or non-US contexts
    - Security and privacy advice for users often rests on this assumption

Understanding users is important for developing PLs

- Want to limit burden on users
- Understand user behavior that could introduce security flaws
- Reduce programmers' cognitive load when using your language
- Create something people want to use that solves a problem

Discussion (5 minutes): Turn to your neighbor and brainstorm some potential usability limitations in some of the programming languages you use. Be ready to share your responses

Many ways to collect information to understand users

Observe

- **Ethnography**
- **Think aloud**
- Content Analysis

Inquire

- **Qualitative**
  - **Interviews**
  - Focus groups
  - Diary studies
- Quantitative
  - Surveys

Experiment

- Surveys
- In-lab or in the field
  - Behavioral economics
  - A/B tests

Agenda

- What (Empathize/Needfinding)
  - Tasks
  - Needs
- How (Approach/skills)
  - Observations
  - Interviews
- Synthesis (Define)
  - Personas – understanding/summarizing what you learn about people
  - How might we questions – frame a brainstorm

What

What are we trying to learn about people?

- What (Empathize/Needfinding)
  - **Tasks**
  - Needs
- How (Approach/skills)
  - Observations
  - Interviews
- Synthesis (Define)
  - Personas – understanding/summarizing what you learn about people
  - How might we questions – frame a brainstorm

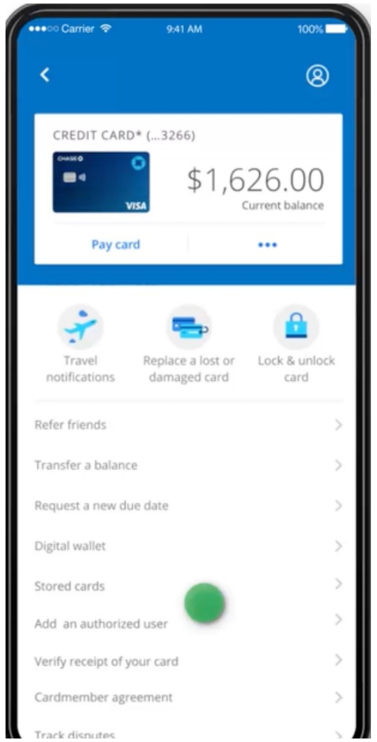
Two things we might be interested in when needfinding:

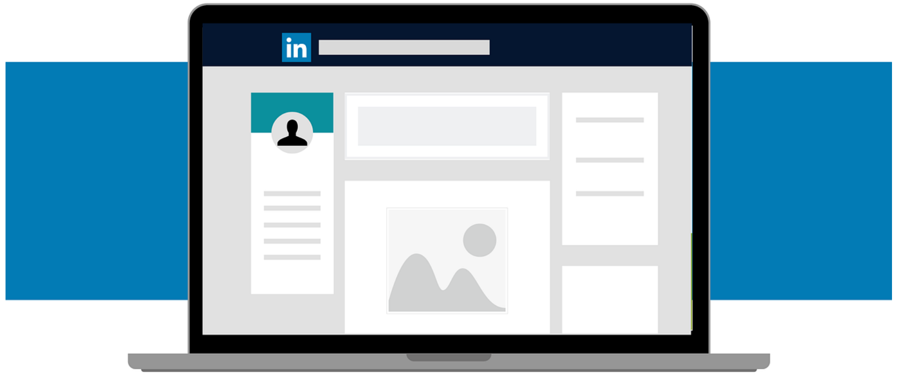
Understanding how people use an existing product/piece of software

Uncovering broad needs; eliciting new perspectives

Task analysis

- A user experience (UX) research method for mapping out how users complete a specific task using a product or software
- Helps identify challenges users might experience when carrying out a task





Task analysis

1. Who is going to use the system?
2. What tasks do they now perform? What tasks are desired?
3. How are the tasks learned? Where are the tasks performed?
4. What is the relationship between people & data? What other tools do people have?
5. How do people communicate with each other? How often are the tasks performed?
6. What are the time constraints on the tasks? What happens when things go wrong?

What can task analysis help with?

- Compare different design alternatives
- Consider how features of a system/interface work together
- Identify specific parts of a system/interface that are challenging to use and could be improved

- What (Empathize/Needfinding)
  - Tasks
  - **Needs**
- How (Approach/skills)
  - Observations
  - Interviews
- Synthesis (Define)
  - Personas – understanding/summarizing what you learn about people
  - How might we questions – frame a brainstorm

Broad needfinding

- What does the user need?
- What are their goals?
- How do they use technology to achieve those goals?
- What challenges do they face in trying to achieve those goals with technology?

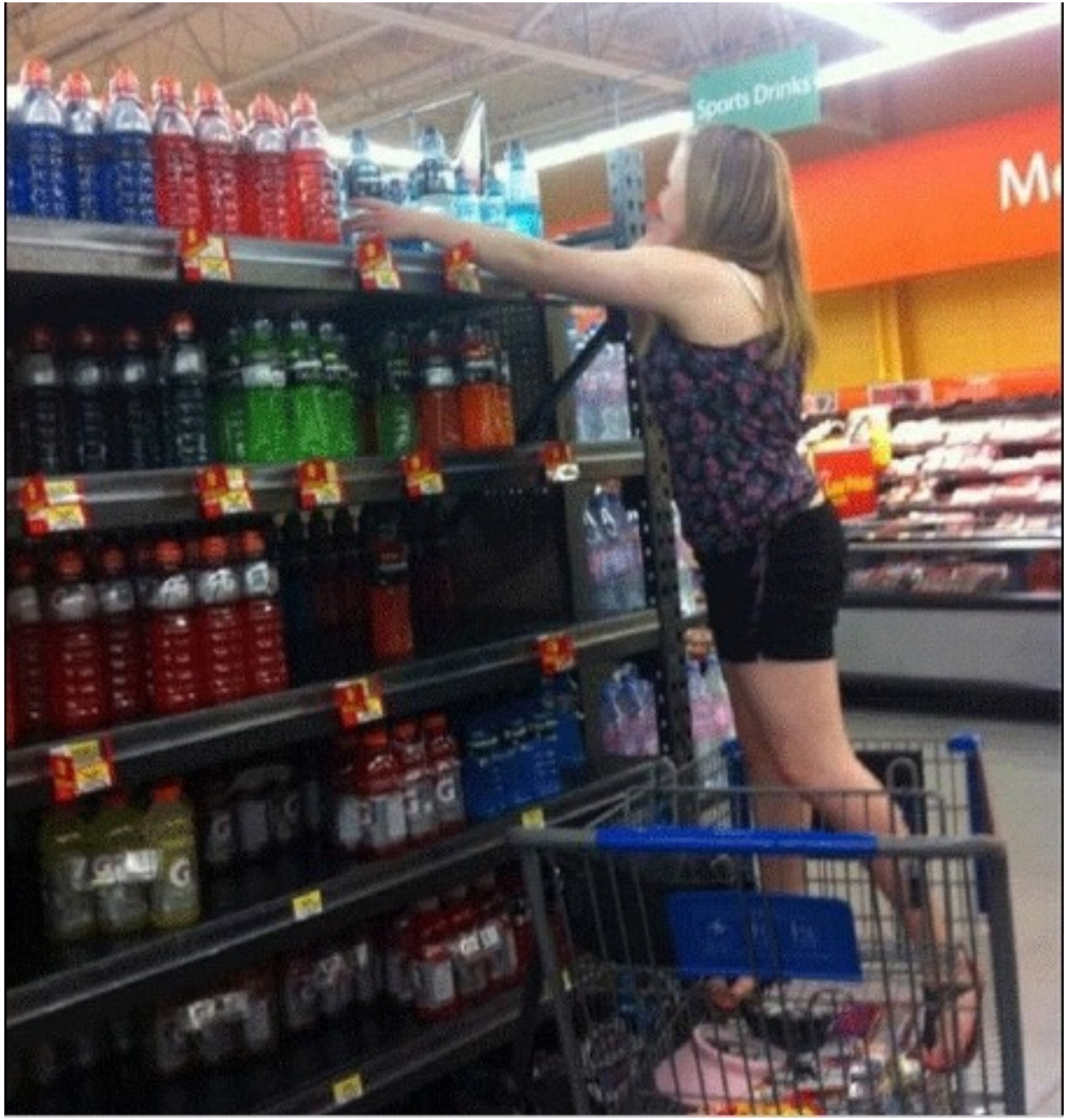
Easier said than done. People are not cognizant of their needs. This is called functional fixedness

Identifying needs is challenging

- Beware of skimming the surface
    - This is the biggest needfinding failure among students and designers/engineers

- Symptoms
    - Describing only what is visible. Such conclusions as the most obvious ones to draw
    - Assuming the tasks are fixed, rather than the needs
    - Recommendations are local tweaks to the environment

Adapted from Michael Bernstein's CS 247 slides:


Sports Drinks

Needs are verbs, not nouns

- Nouns assume the solution:
  - “She needs a ladder”
- Verbs open up many possible solutions:
  - “She needs to grab all her items before leaving”

Adapted from Michael Bernstein’s CS 247 slides:
https://hcicourses.stanford.edu/cs247/2016/slides/07-needfinding.pdf

Observation vs. Interpretation

- A common error is to mix up *what you see* with *what you interpret*
- Start with what you see (observation):
  - What’s the environment or activity that’s framing this behavior
  - What’s out of frame that might be important
- Then interpret, why are you seeing what you see?

Adapted from Michael Bernstein’s CS 247 slides:
https://hcicourses.stanford.edu/cs247/2016/slides/07-needfinding.pdf

Interpretation

- Ask yourself *why* you think something happened
    - Suggest a reason
    - Ask yourself why that reason exists and matters
    - Recurse…
- Aim to produce needs

Adapted from Michael Bernstein's CS 247 slides:
https://hcicourses.stanford.edu/cs247/2016/slides/07-needfinding.pdf

Achieving breadth

- Often needfinding results in observations that focus on what people **see** and **do**
- Make sure you also consider what they **think** and **feel**

Adapted from Michael Bernstein's CS 247 slides:
https://hcicourses.stanford.edu/cs247/2016/slides/07-needfinding.pdf

http://www.pommietravels.com/wp-content/uploads/2011/12/Rockerfeller-Ice-Rink.jpg

How

How we learn things about people?

- What (Empathize/Needfinding)
  - Tasks
  - Needs
- How (Approach/skills)
  - **Observations**
  - Interviews
- Synthesis (Define)
  - Personas – understanding/summarizing what you learn about people
  - How might we questions – frame a brainstorm

Observation

A research method, rooted in the social sciences, that is used to understand how users interact with a product or system by **closely watching how people use the product or interface in a real-world or controlled environment**.

https://www.interaction-design.org/literature/article/how-to-conduct-user-observations

Observation

Controlled setting

Example: Think aloud

- Takes placed in a lab
- Structured: Researcher has a list of things they want to observe
  - How users login to a platform
  - Number of mistakes users make when completing a task

Real-world setting

Example: Ethnography

- Done in the field
- Researchers might go live with the people they want to learn about
- Very useful for understanding how people behave in their natural environment
- Much less structure & higher cost (time)

- What (Empathize/Needfinding)
  - Tasks
  - Needs
- How (Approach/skills)
  - Observations
  - **Interviews**
- Synthesis (Define)
  - Personas – understanding/summarizing what you learn about people
  - How might we questions – frame a brainstorm

Interviews

Interviews involve asking participants a set of evaluation questions and hearing the participants' point of view in their own words.

Interviews vary in structure/precision

- **Informal unstructured interviews**
  - Typically happen "in the field"
- **General topic interviews**
  - You have a list of topics you want to cover, but no specifically phrased questions
- **Semi-to-fully structured interviews**
  - You have an interview protocol consisting of specifically phrased questions.
  - In a semi-structured interview you can paraphrase a little + adjust order based on the conversation
  - In a fully-structured interview you follow the protocol exactly

Interviews are useful…

When there is not much existing work on a given problem, so you wouldn't know

- what to ask (or what answer choices to give) on a survey
- What interventions to design, implement and test in an A/B test

You want to understand the "why" – quantitative data has a harder time giving you the why

When you want to delve into depth about a topic, an experience, a program

When reading and writing skills are limited

https://www.nixdell.com/classes/HCI-Spring-2018/Lecture4.pdf

How To Do Interviews

- Draft your protocol
- Practice
- Recruit
- Interview
- Analyze
- Report

What to write questions about

- Consider your research questions. Outline the broad areas of knowledge that are relevant to answering these questions
- Create a series of sub-questions tied to each research question / area of knowledge to fill the gaps and get the data you need to answer those research questions + move to the next step of your research process.

Examples: Broad areas of knowledge

Research question: How do people protect themselves from online hate and harassment when using social media?

- Types of social media people use
- Types of online hate and harassment they experience
- Steps people take to protect themselves
- Tools/technologies they use to protect themselves
- Limitations of those technologies

Types of questions

- Direct questions:
  - Do you find it difficult to avoid online harassment?
  - Are you happy with existing mechanisms to protect against online harassment?
- Indirect questions:
  - What has your experience been using X, Y, and Z tools to protect against online harassment?
  - How have those experienced shaped your use of the Internet?
- Structuring questions:
  - 'I would now like to move on to a different topic'.
- Follow-up questions: getting the interviewee to elaborate his/her answer
  - 'Could you say some more about that?'; 'What do you mean by that . . .?'
- Probing questions: following up what has been said through direct questioning.
- Specifying questions:
  - 'What did you do then?'
  - 'How did your strategies change based on X event?'
- Interpreting questions: make sure you understand what they mean
  - Is it fair to say that you don't feel moderators of online communities understand the types of harassment you've experienced so they're not well-equipped to remove content?

Adapted from: https://www.nixdell.com/classes/HCI-Spring-2018/Lecture4.pdf

Key element of interviews: Follow ups + Probes

Probing is the process of asking follow-up questions to **dig deeper** in order to obtain useful, meaningful information

Example:

"What do you like best about the CS program at Stanford?"

Response: "I like everything."

Probe 1: "What one thing really stands out?"

Response: "My professor in the algorithms class."

Probe 2: "What did you like about that professor?"

Response : "I liked that he had famous people come talk to us."

Probe 3: "Really? Tell me more...?"

Response : "It was interesting to hear their perspectives. I heard some things I hadn't considered before."

Probe 4: "What is one thing that you learned from them?"

Adapted from https://www.nixdell.com/classes/HCI-Spring-2018/Lecture4.pdf

Good Question Practices

The best questions are those which elicit the longest answers from the respondent. Do not ask questions that can be answered with one word.

- Anchored to experiences
  - Helps with memory
  - The point of a qualitative interview is to let the respondent tell their own story on their own terms
- Open Ended
  - Ask "how" questions rather than "why" questions to get stories of process rather than acceptable "accounts" of behavior. "How did you come to join this group . . .?"
  - THIS IS NOT A SURVEY! The guide acts as a prompt, reminding you of necessary topics to cover, questions to ask and areas to probe. As such, it should be simple so that your primary focus can stay on the respondent. It's best to memorize your guide!
- Ask one question at a time
- Adapt to the participants' language
- Simple, easy, short questions; no jargon
- Be neutral, consider couching language for sensitive topics
  - "**People have many different things they do in their free time**. Could you tell me a bit about how you use substances in your free time?"

https://sociology.fas.harvard.edu/files/sociology/files/interview_strategies.pdf
https://nsuworks.nova.edu/tqr/vol15/iss3/19/

“Bad” Question Practices

- Leading
- Impossible
    - Beyond expertise
        - Functional fixedness
- Use very very rarely
    - Closed
        - Binary
        - Counts / likerts
    - Hypothetical

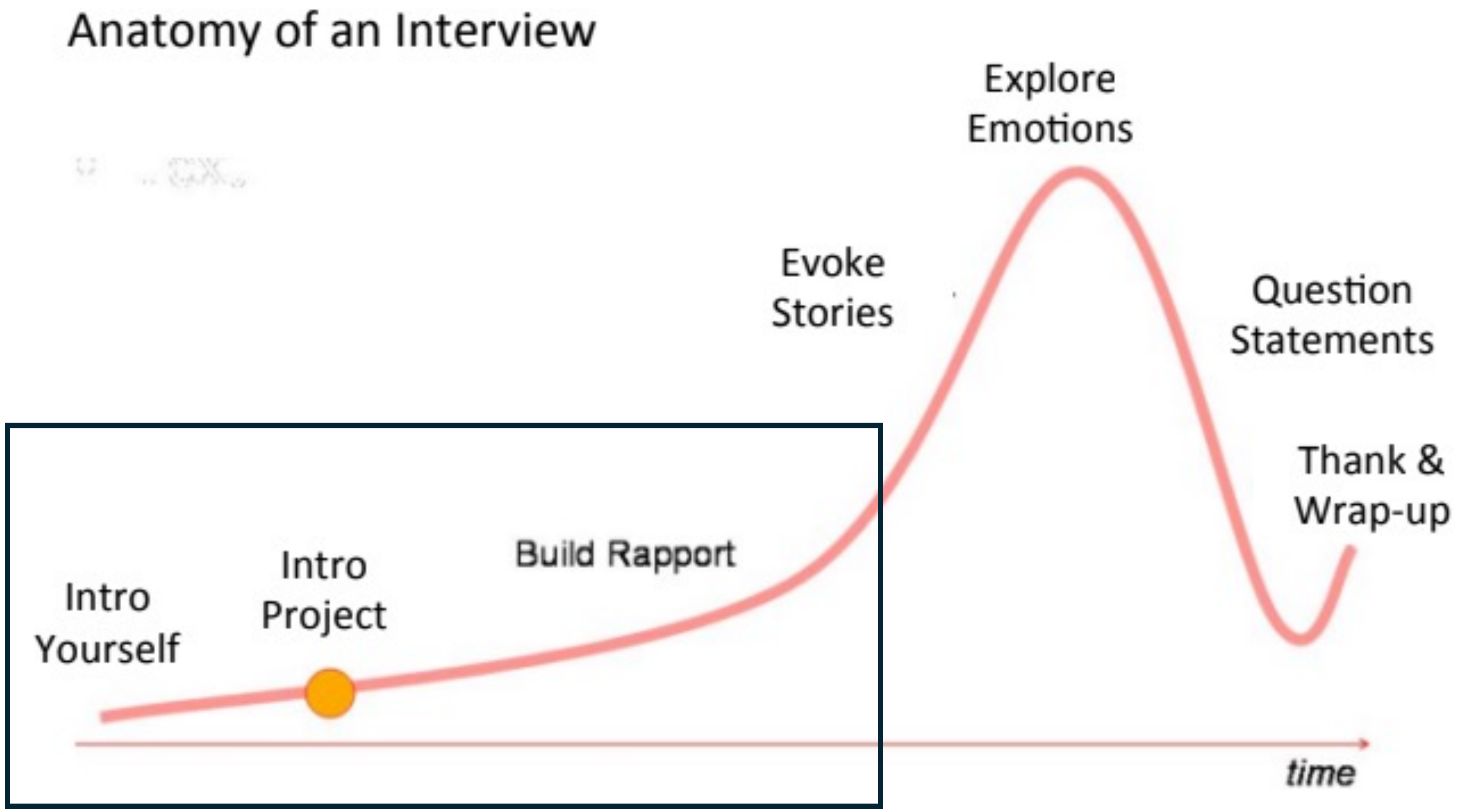
Activity (10 minutes): Interview protocol

How people interact with technology and what challenges they experience.

Today, spend 5 minutes

1. writing out broad areas of knowledge you want to learn about from your interviewee
2. different types of questions you might ask to learn about those areas of knowledge

Then, you’ll get 5 minutes to share your protocol with the person sitting next to you. I’ll let you know when to move on to this.



Slide courtesy of Prof. Ben Wiedermann for CS 111 at HMC

Structuring Interviews

- Give context on the interview (without priming)
- Warm the participant up before launching into sensitive questions
- Provide transition between major topics, e.g., "we've been talking about (some topic) and now I'd like to move on to (another topic);"

10 things to keep in mind while interviewing

1. Be polite
    1. Greet the interviewee in a culturally appropriate way
    2. Explain the purpose of the interview
    3. Give them an opportunity to ask you questions about the study and their participation
    4. If recording, ask for their permission to begin recording
    5. Thank the interviewee at the end and ask them again if they have any questions
2. Be considerate
    1. Try to stay on time.
3. Be personable – friendly conversation helps break the ice
    1. In your interview, be unconditionally positive
4. Be engaged
    1. Follow the flow of conversation
    2. Listen!! Don't only look at your interview guide
    3. Ask followup questions
    4. Pick up phrases the interviewee has used and use them for your questions

https://www.nixdell.com/classes/HCI-Spring-2018/Lecture4.pdf
https://sociology.fas.harvard.edu/files/sociology/files/interview_strategies.pdf

10 things to keep in mind while interviewing

5. Be gentle
    1. Let people finish their thought. Don’t be afraid of silence
6. Be balanced
    1. Don’t talk too much!
    2. Never answer a question for the interviewee or give them an example answer
7. Be sensitive
    1. Treat interviewees as your equal. Never pass judgment on anything they say
8. Be open
    1. Be flexible and respond to what the interviewee thinks is important
9. Know how to steer the interview
    1. Be prepared & know what you want to find out
10. Be critical
    1. Gently challenge inconsistencies in interviewees responses to understand more

Final reminders about interviews

- Your job is to follow, not lead
- Hold off on solutions and answers
- Don’t be afraid of silence

Synthesis

How do we make sense of data about people to build something useful that meets their needs?

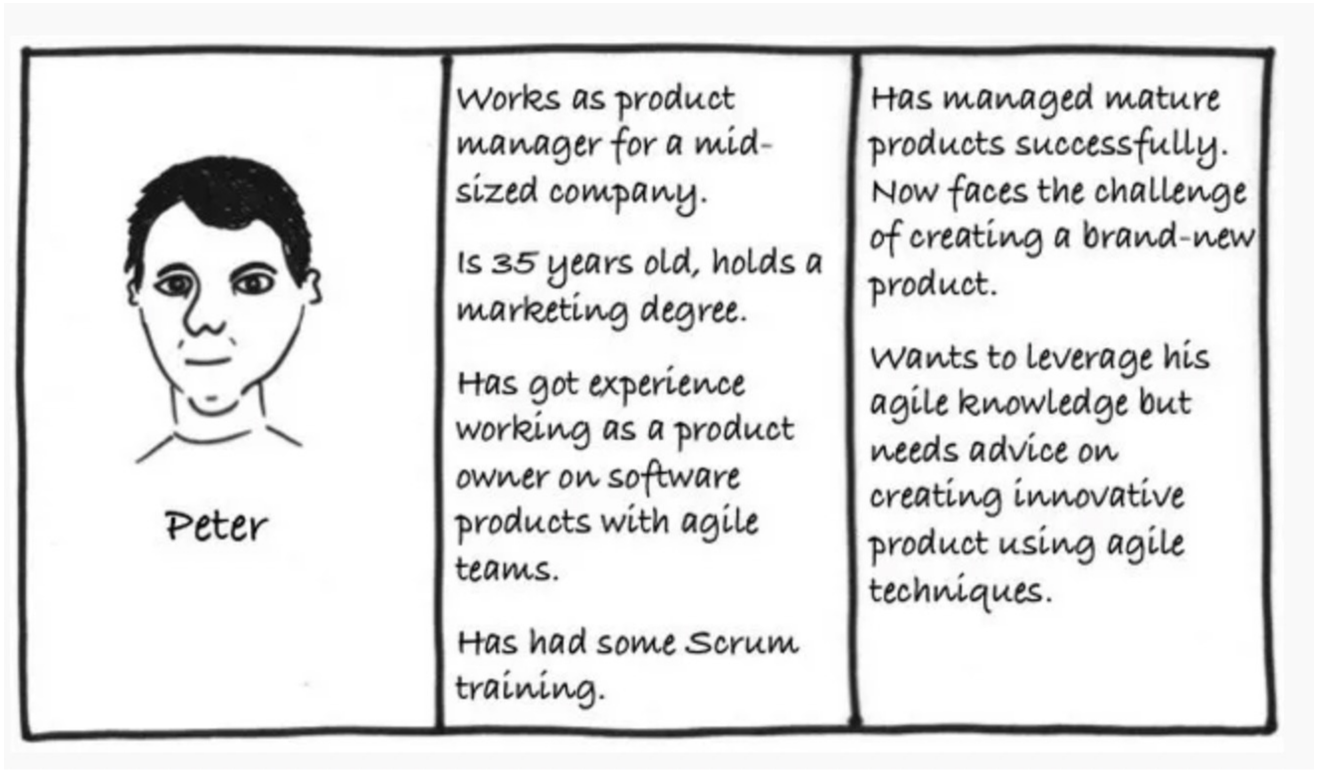
- What (Empathize/Needfinding)
  - Tasks
  - Needs
- How (Approach/skills)
  - Observations
  - Interviews
- Synthesis (Define)
  - **Personas** – understanding/summarizing what you learn about people
  - How might we questions – frame a brainstorm

User personas

- Not every person on a product team has time or should directly talk to “users”
  - (And, you don’t always need new research!)
    - It’s quite a burden to get asked about your experience over and over and over again
- **A persona is a fictional yet realistic representation of a specific user or user group that helps designers and developers understand their target users’ needs, goals, and behaviors**
  - “use of abstract representations of users originated in marketing” [1]
  - “Personas were originally proposed in the late 1990s by Cooper as an effective and efficient way of engaging designers and developers with the characteristics and needs of their users” [2]

https://dl.acm.org/doi/pdf/10.1145/997078.997089
https://dl.acm.org/doi/pdf/10.1145/3419249.3420135

User personas



Peter

Works as product manager for a mid-sized company.

Is 35 years old, holds a marketing degree.

Has got experience working as a product owner on software products with agile teams.

Has had some Scrum training.

Has managed mature products successfully. Now faces the challenge of creating a brand-new product.

Wants to leverage his agile knowledge but needs advice on creating innovative product using agile techniques.

A fictional representation of a user that includes details about their:

- Education
- Lifestyle
- Interests
- Age
- Values
- Goals
- Needs
- Limitations
- Desires
- Attitudes
- Patterns of behaviors

https://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them

Personas as proxies in design

We can’t talk to everyone we’re designing for, so personas are used as a proxies in design

Major function of a persona is to enable designers and engineers to break free of the tendency to design for themselves

Personas Case Study: Rust

- The situation
  - Who: the Rust language "async foundations" team
    - A distributed group of (mostly) volunteers
  - When: 2021
  - What: long-term planning for Rust async
- What did they do?
  - They ***made personas***!
  - https://nikomatsakis.github.io/wg-async-foundations/vision/characters.html

Personas Practice (10 minutes, in pairs)

1. Choose a programming language
2. Create **two** personas to guide future changes
   - Occupation (HS student, college student, researcher?, developer?, manager? non-dev worker?)
   - Educational background?
   - Experience with the language?
   - Future goals?
   - Think outside the box!
   - Try to find two personas that are meaningfully different and “span” the space
3. Summarize the personas on whiteboards
4. Next, we’ll share them

Inspired by: https://github.com/uwcse440/web-cse440-sp19/blob/master/slides/CSE440-08-TaskAnalysis.pdf

Persona Discussion (5 minutes)

- With another pair
- Take turns explaining your personas
- Critique the personas
    - Which personas seem left out?
        - (this is unavoidable---you only made two!)
    - Is there an adequate level of detail?
    - Do you feel like you have a good understanding of the user?
    - Would it be easy/useful to refer to this persona while designing a system or feature?
        - What is an example language feature that you might imagine one of the personas having an opinion about?

Inspired by https://github.com/uwcse440/web-cse440-sp19/blob/master/slides/CSE440-08-TaskAnalysis.pdf

- What (Empathize/Needfinding)
  - Tasks
  - Needs
- How (Approach/skills)
  - Observations
  - Interviews
- Synthesis (Define)
  - Personas – understanding/summarizing what you learn about people
  - **How might we questions** – frame a brainstorm

“How might we…?”

“How might we…?” statements help brainstorm potential problem spaces and can generate lots of creative ideas

- *How might we* ensure more people pay their taxes before the deadline?
- *How might we* help employees stay productive and healthy when working from home?
- *How might we* make customers feel that their information is safe and secure when creating an account?

https://www.nngroup.com/articles/how-might-we-questions/

The Goldilocks of How Might We

- A good “How might we” question is:
  - Not so broad that it is inapproachable
    - How might we help people organize all their digital media?
  - Not so narrow that it suggests a solution
    - How might we help people retrieve their favorite digital media with just a click
  - In a happy middle ground
    - How might we help weekend extreme sports enthusiasts organize their digital media?

Adapted from Michael Bernstein’s CS 247 slides:
https://hcicourses.stanford.edu/cs247/2016/slides/07-needfinding.pdf

Case study: D and Rust

- Goal: to optimize both performance and safety
  - **HMW 1:** How might we make garbage collection performant?
    - Presumes a solution that in this timeline was not ultimately adopted (too strict). If people had asked this question they would not have come up with Rust. (Too narrow)
  - **HMW 2:** How might we ensure C++ is memory safe?
    - Would have led to a bunch of static analysis tools for C++ probably wouldn't have been very impactful. This also wouldn't have led to Rust. (too broad)
      - HMW 3: **How might we write code that has type control over memory management but is also memory safe? → leads to Rust**
      - HMW 4: **How might we test whether some C++ code might be memory unsafe? → leads to Valgrind**

Recap

- What (Empathize/Needfinding)
  - Tasks
  - Needs
- How (Approach/skills)
  - Observations
  - Interviews
- Synthesis (Define)
  - Personas – understanding/summarizing what you learn about people
  - How might we questions – frame a brainstorm