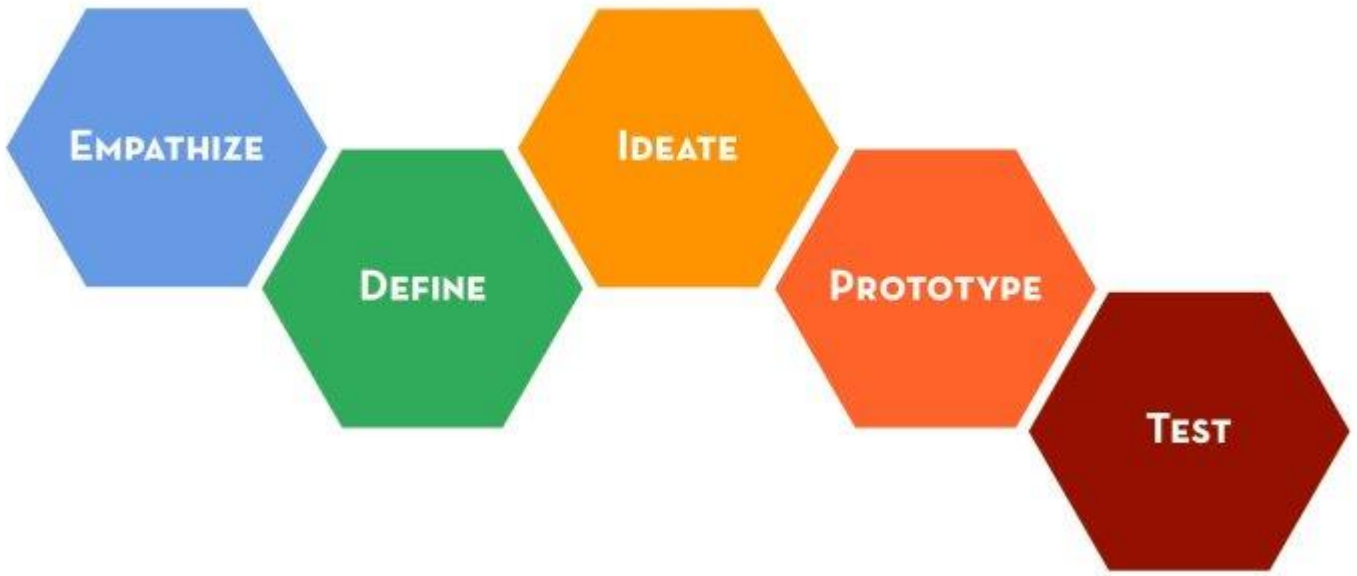
Introduction to Human-Computer Interaction

4. Design Thinking Process

Instructor: Jaemin Jo (조재민, jmjo@skku.edu)
Interactive Data Computing Lab (*IDCLab*),
College of Computing and Informatics,
Sungkyunkwan University

Design Thinking Process

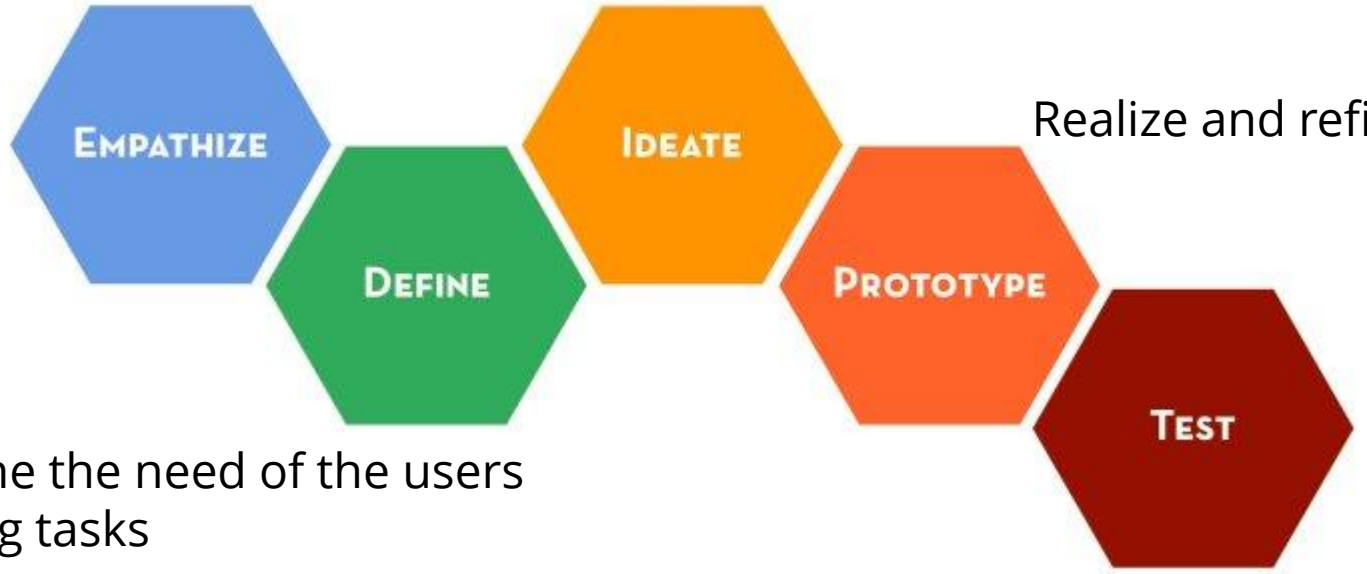
- Design thinking is a non-linear, iterative process that teams use to understand users, challenge assumptions, redefine problems and create innovative solutions to prototype and test.
  - A framework for user-centered design



Design Thinking Process

Explore; getting to know your users
Understanding users

Develop a list of promising ideas

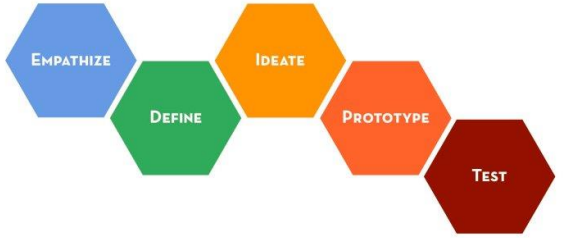


Realize and refine the ideas

Find and define the need of the users
Understanding tasks

Observe users' reception

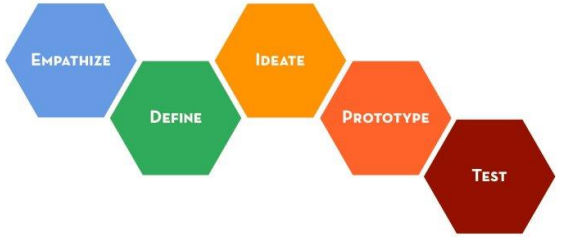
1. Empathize



- Understand people within the context of your design challenge.
  - How and why do the users do certain things?
  - Sometimes, called needfinding or user analysis

- **Observe**: View users and their behavior in the context of their lives
- **Engage**: Interviewing, but it should really feel more like a conversation
- **Watch and Listen**: Combine observation and engagement
  - Ask someone to show you how they complete a task
  - Have them physically go through the steps
  - Ask them to vocalize what's going through their mind as they perform a task

2. Define

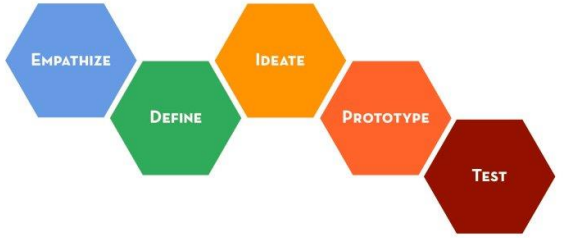


- Define the challenge you are taking on, based on what you have learned about your user and about the context.

- Craft a meaningful and actionable statement

- Develop an understanding of the users
- Synthesize and select a limited set of needs
- Work to express insights you developed

2. Define

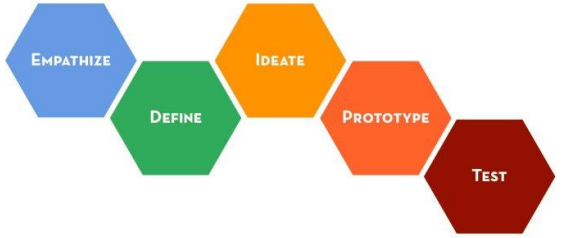


- Point-of-view statement (POV)

- **(USER)** needs a way to **(NEEDS)** because **(INSIGHT)**.

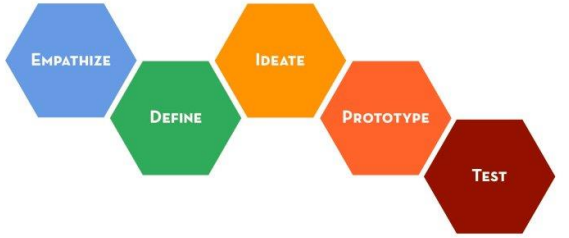
- **USER**: An adult person who lives in a city
- **NEED**: To use a car for 10-60 minute trips 1-4 times per week
- **INSIGHT**: The user would not want to own his own car as it would be too expensive compared to his needs.

2. Define



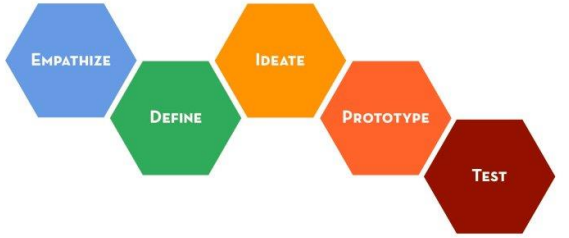
- Overworked, busy Mom with 3 kids seeks help with grocery shopping, to keep kids happy and have them learn from the experience.

2. Define



- Provides focus and frames the problem
- Inspires your team
- Informs criteria for evaluating competing ideas
- Empowers your team to make decisions independently in parallel
- Captures the hearts and minds of people you meet
- Saves you from the impossible task of developing concepts that are all things to all people

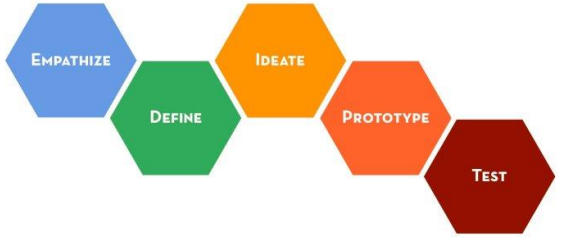
3. Ideate



- Concentrate on idea generation
- Transit from identifying problems to creating solutions for the users.
  - Brainstorming, building, mind mapping, ...

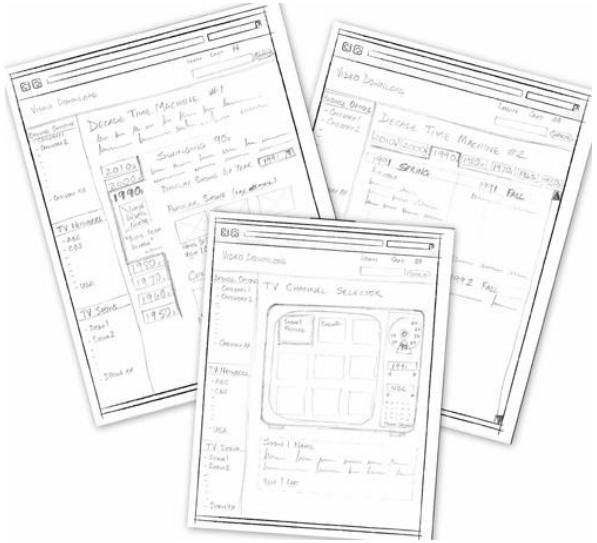
- **Keep an open mind:** No ideas are bad ideas at this point; everything is on the table. Ideas will be judged on feasibility later, but now is the time to push the boundaries and consider all possibilities.

4. Prototype

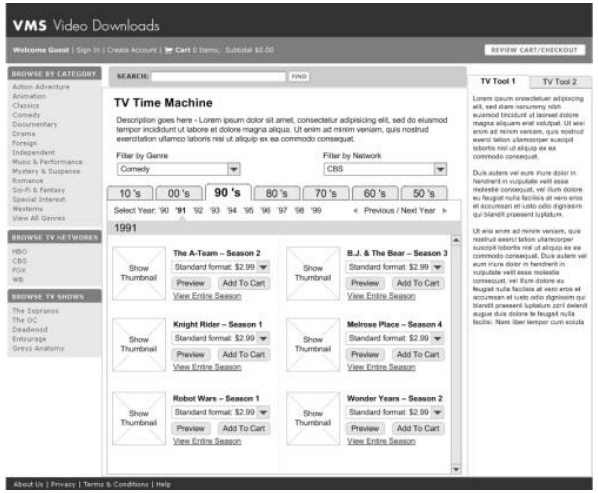


- Iterative generation of artifacts (programs, visual designs, …) intended to answer questions that get you closer to your final solution
- “Build to think and test to learn”

Paper sketches
(low fidelity)

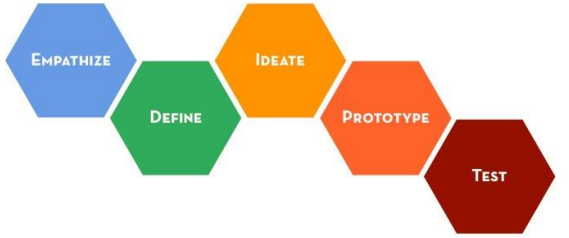


Wireframing
(med fidelity)



Coding
(high fidelity)

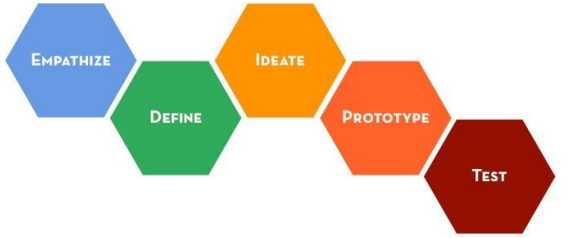
4. Prototype



- Why do we prototype?

- To ideate and problem-solve. Build to think.
- To communicate.
  - If a picture is worth a thousand words, a prototype is worth a thousand pictures.
- To start a conversation.
  - Your interactions with users are often richer when centered around a conversation piece.
- To fail quickly and cheaply.
- To test possibilities.

5. Testing



- Solicit feedback about the prototypes you have created from the users.
- Another opportunity to understand the users

- Not simply asking whether or not the users like your solution.
- Continue to ask “Why?”

- Show don’t tell
- Ask users to compare multiple prototypes

Empathize

- Feel what the users actually feel

- “You are not the customer”
  - Easy to think of self as a typical customer (designer ego bias)

- You **NEED** to meet the actual users.
  - If you want to design a new product, the first thing you should do is to find people who will user the product.
  - This often takes long.

Who are the Users?

- Those who interact directly with the product
- Those who manage direct users
- Those who receive output from the product
- Those who make the purchasing decision
- Those who use competitors' product
- **Primary users**: people who use the product directly
- **Secondary users**: people who are affected by the product, or who influence its development

What are their Needs?

- Needfinding

- The most important thing about need finding in design thinking is that we look without knowing what we are looking for

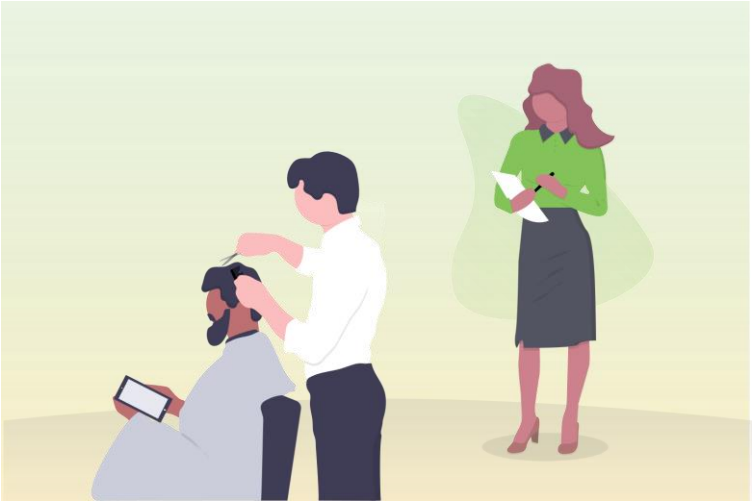
- How?
  - Observation
  - Contextual Inquiry
  - Diary studies
  - Surveys
  - Interviews
  - …

Observation

- Observe users and their behavior
- **Field studies:** directly observing users in their context (usual work or home environment)
- **Controlled studies:** directly observing users in a controlled setting, performing special tasks
- Video recording/performance logging (indirect)

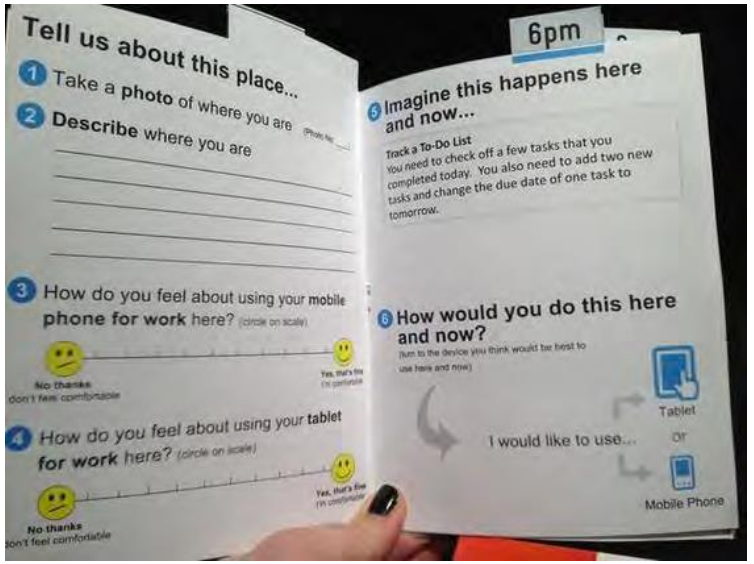
Contextual Inquiry

- Master/Apprentice model
  - The user does the work and talks aloud.
  - Apprentice (you) makes observations and interrupts with questions.
- Good for short and regular tasks
  - You cannot stay with them for a month (longitudinal study)
  - Hard to get in “rare” situations
    - How doctors treat people struck by lightening



Diary Studies

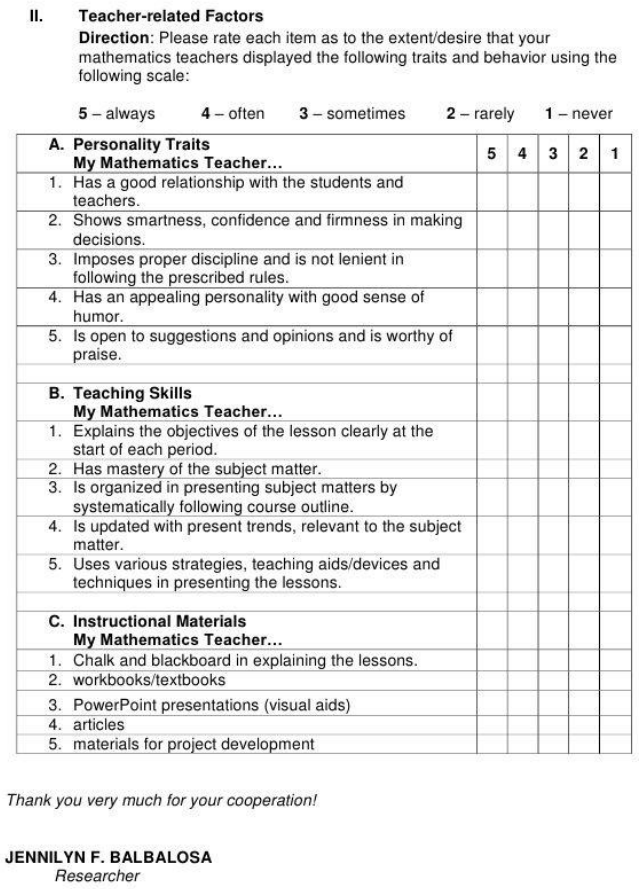
- Participants capture the experience themselves at a regular interval.
  - How productive are you now?
- Longitudinal, qualitative
- Journals, cameras, voice, video
  - Easy capture tools are important.
  - Smartphone + notification
- During a lecture? While driving?



Kate Needham on Pinterest

Survey

- A series of carefully structured questions given to a group of people
- Easy to recruit participants
  - You can get many perspectives rapidly.
- Screening questions either qualify or disqualify respondents from taking your survey
- Provide progress updates through survey
- Ask at least one open ended question



II. Teacher-related Factors

Direction: Please rate each item as to the extent/desire that your mathematics teachers displayed the following traits and behavior using the following scale:

5 – always **4** – often **3** – sometimes **2** – rarely **1** – never

	5	4	3	2	1
A. Personality Traits My Mathematics Teacher…					
1. Has a good relationship with the students and teachers.					
2. Shows smartness, confidence and firmness in making decisions.					
3. Imposes proper discipline and is not lenient in following the prescribed rules.					
4. Has an appealing personality with good sense of humor.					
5. Is open to suggestions and opinions and is worthy of praise.					
B. Teaching Skills My Mathematics Teacher…					
1. Explains the objectives of the lesson clearly at the start of each period.					
2. Has mastery of the subject matter.					
3. Is organized in presenting subject matters by systematically following course outline.					
4. Is updated with present trends, relevant to the subject matter.					
5. Uses various strategies, teaching aids/devices and techniques in presenting the lessons.					
C. Instructional Materials My Mathematics Teacher…					
1. Chalk and blackboard in explaining the lessons.					
2. workbooks/textbooks					
3. PowerPoint presentations (visual aids)					
4. articles					
5. materials for project development					

Thank you very much for your cooperation!

JENNILYN F. BALBALOSA
Researcher

Types of Interviews

- **Structured interview**: using pre-determined questions to ask in a set way
  - Inflexible

- **Semi-structured interview**:
  - Start with set topics for discussion and follow up based on the answers
    - You should “drive” the interview well.
  - Useful for gathering requirements and understanding users’ opinions further in depth
  - One of the most widely used data collection methods in UI/UX design
  - We will just call this “an interview”.

Slide adapted from Sunyoung Kim

Pros/Cons of Interview

- **Pros:**
  - “Deep” rich data
  - Open-ended and exploratory
  - More economical in some cases than observing the user

- **Cons:**
  - Time-consuming (conducting + analyzing)
  - Overwhelming amount of data (hard to digest all)
  - Can be biased by interviewer
  - Users can be inconsistent
    - what they say might be different from what they actually do

Slide adapted from Sunyoung Kim

How Many Users? Who?

- As many as you can afford
- Stop interviewing if no new things are found out.
- 10-15 is a good number

- Make sure to choose representative users.
  - Who are the representative users?

Slide adapted from Sunyoung Kim

Formulating Questions

1. Start by defining broader themes
   - Think through what you are trying to get out of the interviews
   - Think about themes you are trying to uncover, not specific questions just yet

- Examples:
  - “Why do people shop online?”
  - “How do people shop online?”
  - “For your customers, what is the difference between online and offline shopping?”

Slide adapted from Sunyoung Kim

Formulating Questions

2. Break down your questions to make them answerable

- Examples:

“Why do people shop online?”

To:

- “What types of product do you buy online?”
- “What types of product do you avoid buying online? Why?”
- “What do you like the most and the least about the checkout process?”

Slide adapted from Sunyoung Kim

Formulating Questions

3. Prioritize open-ended questions
    - Give users some room to elaborate their answers, as opposed to making super binary questions.

- Examples:
    - “What was the last thing you bought online?” vs.
    - “Tell me about the last time you bought something online.”

Slide adapted from Sunyoung Kim

Get Specific

- Ask for tasks, roles and details of tasks
- Ask about specific moment in the past: walk through tasks “yesterday”
- Ask for examples!!!

- Examples:
  - “What goes through your head when an online purchase fails?” vs.
  - “Tell me what went through your head the last time you tried to buy something online and the purchase failed.”

Slide adapted from Sunyoung Kim

Don’t Ask

- How often they do things (they simply do not know enough)
- How much they like things on an absolute scale. If you ask, they will “guestimate”
- Questions that you can easily predict the answers
- Questions that will influence the answer (**leading questions**)
  - Don’t ask leading questions
  - “**How anxious** do you feel when an online purchase can’t be completed successfully?”
  - “Try to remember the last time an online purchase couldn’t be completed for some reason. How did you feel then?”

Slide adapted from Sunyoung Kim

Don’t Ask

- What they would do/like/want; new features
- Anything that makes people imagine hypothetical situations

- Because they will focus on functionality they currently understand
- Because they do not know what is possible

Slide adapted from Sunyoung Kim

Tips for Questions

- Leading Question: “How angry do you usually feel when an online transaction fails to go through successfully?”
- Non-leading Question: “Recall a time when an online transaction failed to go through successfully. How did you feel?”
- Even More Non-leading Question: “Recall a time when an online transaction failed to go through successfully. What did you think or feel?”

Slide adapted from Sunyoung Kim

Tips for Questions

- Ask about specific incidents in the past, when possible.
- Speculative Question Based on Imagination: “Tell me what goes on in your head when an online transaction fails to go through successfully.”
- Context-based Question Grounded in Reality: “Tell me what went on in your head the last time an online transaction fails to go through successfully.”

Slide adapted from Sunyoung Kim

Tips for Questions

- You can ask more than one question to find out the same thing.
- Q1: “What are some reasons why you shop online?”
- Q2: “What are some things that will make you decide not to buy from an online store?”

Slide adapted from Sunyoung Kim

Interview Examples

- A: My only frustration is when the server goes down, everything will just freeze
- Q-: That makes me feel real nervous when that happens to me
- Q+: What do you do then?

- A: Yeah, is that stuff I've been working on really going to be there when it comes back up?
- Q-: I know, I would like some kind of indication when everything freezes up, so you know that your work isn't gone.
- Q+: What's your reaction when that happens

Slide adapted from Sunyoung Kim

Interview Examples

- Ask unbiased questions
- Q-: Which of the desktop applications do you use regularly? It looks like file manager, calendar manager…

- Avoid presenting options
- A: I use calendar manager and mail tool constantly
- Q-: And do you use that to set up the appointments with the people that you work with or do you keep track on your own?
- Q+: What do you use calendar manager for?

Slide adapted from Sunyoung Kim

Interview Script

1. Introduction
   - Introduce yourself, goal, agreement for recording, background, …
2. Warm up questions
   - Occupation, hobbies, “casual” questions
3. Main body questions
4. Wrap up
   - Thank them for their time and contribution

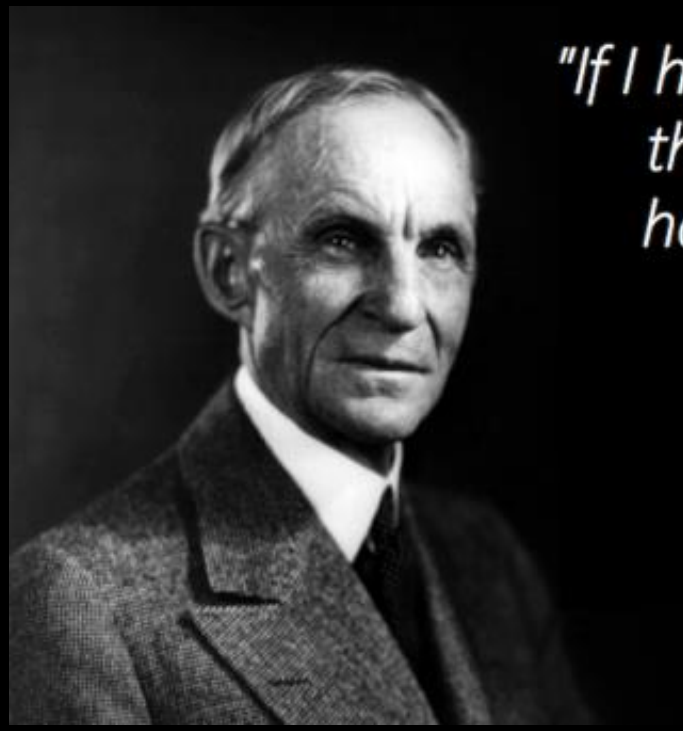
Slide adapted from Sunyoung Kim

Interview Tips

- Do not simply transcribe answers. You should digest their answers in the middle of interviews.
- Look for workarounds & hacks.
- “Errors” are a goldmine.
- Do not talk too much. The interviewee should speak most of the time.
- Be ready to hear something new and be changed by it.
- Be curious.
  - Ask why, ask for an example, clarify terms, ask for step-by-step actions
- Have a “beginners’ mindset”.

But, in practice…

- Users often don’t know what’s possible.
- Users often can’t articulate clearly what they need to achieve their goals.
- Users often can’t provide enough details for you to proceed to the next step.



"If I had asked people what they wanted, the would have said faster horses"

Henry Ford

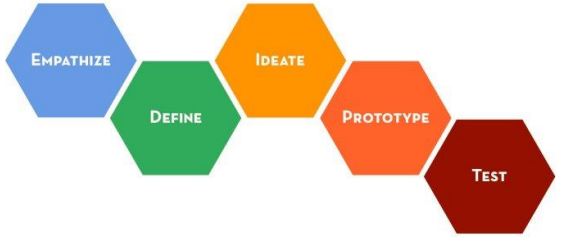
"A lot of times, people don't know what they want until you show it to them."

— Steve Jobs

But, in practice…

- This is why the design process should be iterative.
- We will revisit this stage with a tangible prototype later.
- With the “visible” prototype in the second round, the users will be able to give more feedback/comments and even realize their real needs.

Task Analysis



- One goal of the “Define” stage is to have concrete description on the (current) way in which people perform their tasks to achieve their goal.

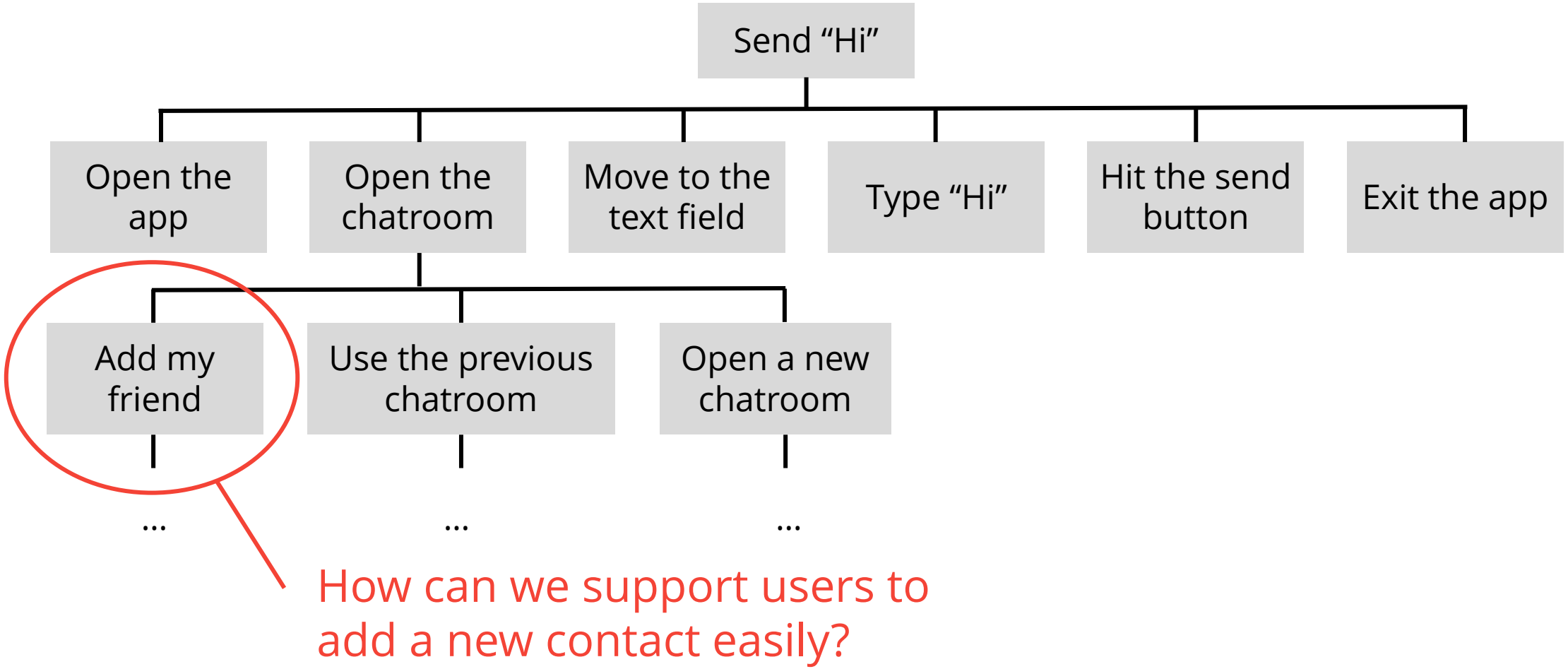
- **Outcome**: a diagram explaining the steps that a user takes.
  - List the actions and describe each.
  - Identify additional/unnecessary actions.
  - Give the user support for those actions.

Task Analysis Example

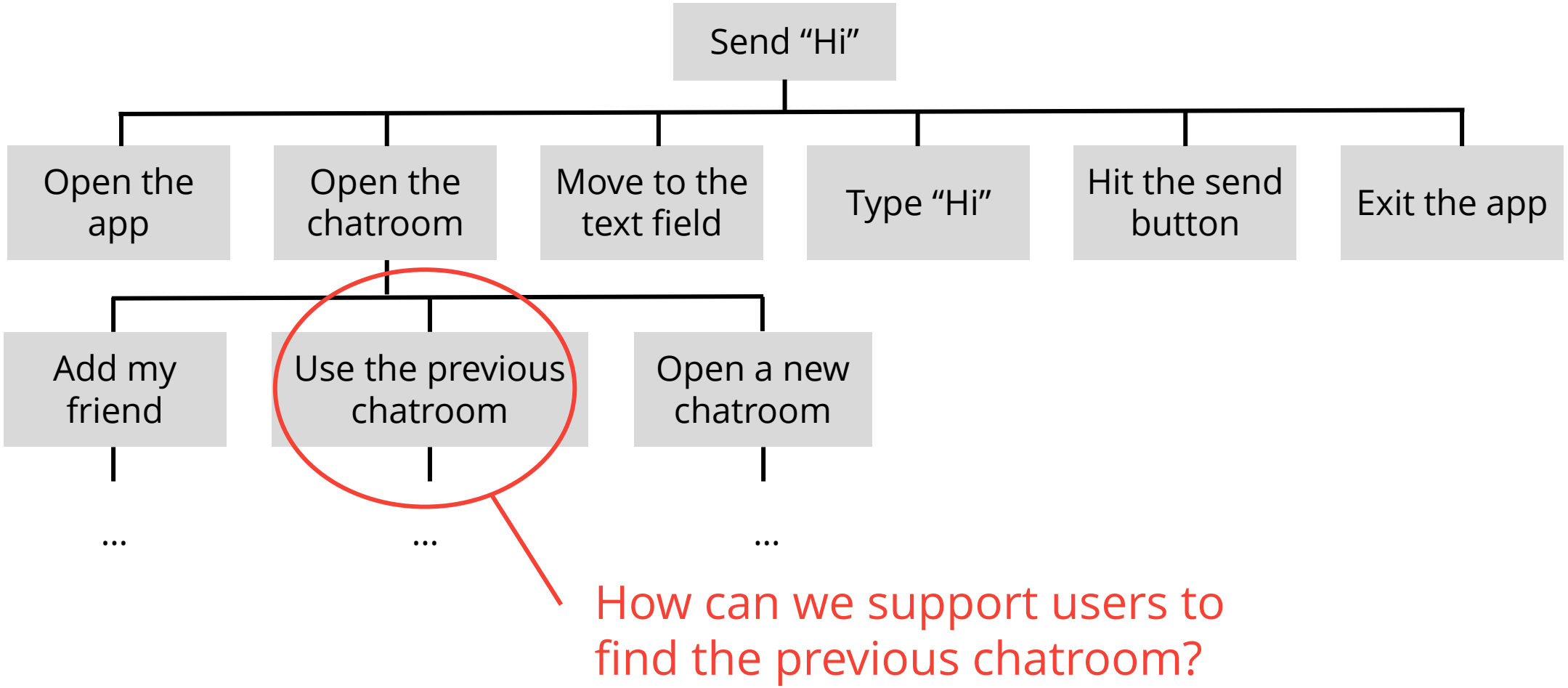
- Goal: Send “Hi” to my friend on KaKaoTalk

- **Tasks:**
- Open KaKaoTalk.
- Open the chatroom where my friend is.
  - If my friend has not been added in my contact list, …
  - If there is a chatroom already opened, …
  - If there is no chatroom, …
- Move to the text input field.
- Type “Hi”.
- Hit the send button.
- Exit KaKaoTalk.

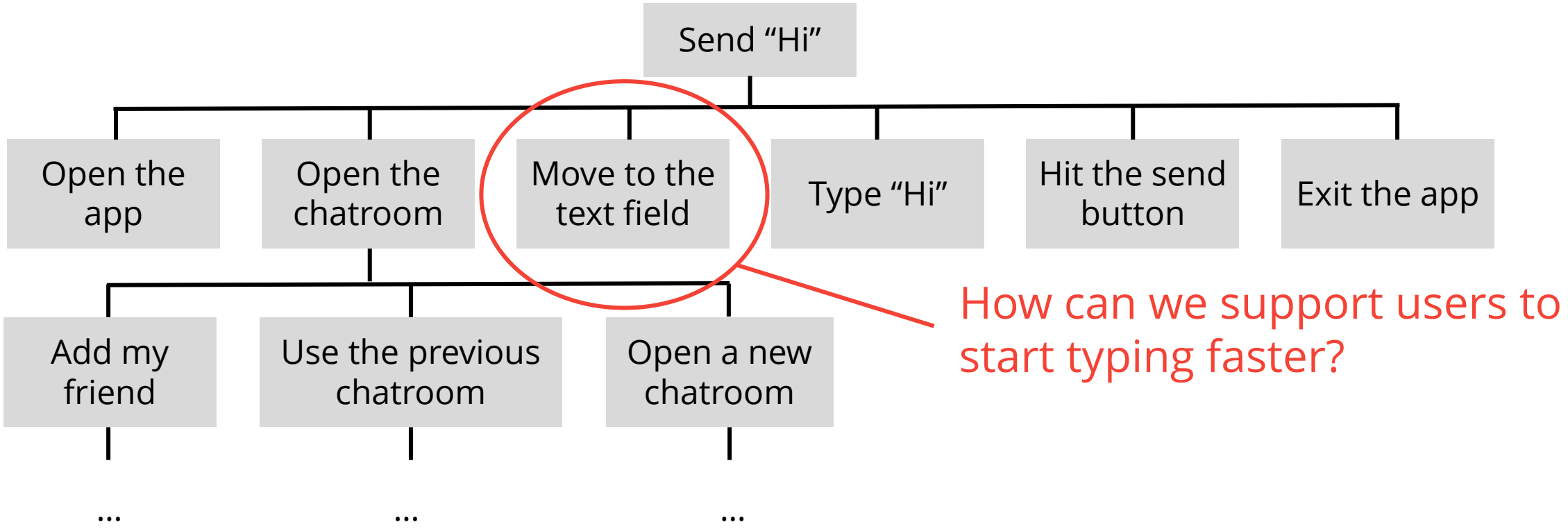
Task Analysis Example



Task Analysis Example



Task Analysis Example



Focus of Task Analysis

- When collecting data for task analysis, focus on:

- **Trigger**: What prompts users to start their task?
- **Desired Outcome**: How users will know when the task is complete?
- **Base Knowledge**: What will the users be expected to know when starting the task?
- **Required Knowledge**: What the users actually need to know in order to complete the task?
- **Artifacts**: What tools or information do the users utilize during the course of the task?

Task Analysis: Steps

1. Identify the task to be analyzed: What is the user's goal and motivation for achieving it?
2. Break this goal (high-level task) down into subtasks: You should have around 4–8 subtasks after this process. If you have more, then it means that your identified goal is too high-level and possibly too abstract
3. Collect data to identify tasks/subtasks that the user performs.
4. Draw a layered task diagram of each subtask and ensure it is complete.

Task Analysis: Questions

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