TASK EXAMPLES & ESTABLISHING REQUIREMENTS

CPSC 544: FUNDAMENTALS IN DESIGNING INTERACTIVE COMPUTATIONAL TECHNOLOGY FOR PEOPLE

COMING UP

This week (Oct 9, Class C10; includes Thanksgiving holiday)

- **Tues 10/10**
  - Researcher Journal #7

Coming Up – back to 2 lectures/week!

- **Sun 10/15**: next team deliverable
  - Tasks, Requirements and Personas report
  - Researcher Journal #8

- **Wed 10/18**
  - Draft conceptual models for team walkthroughs (in-class activity)

LEARNING GOALS

Part 1: Tasks

- Learn about tasks examples, task analysis and task dependencies
- Understand the relationship between **personas**, **scenarios**, **use cases, tasks, task examples** and **requirements**
- Practice breaking down an activity (task analysis) to arrive at requirements

Part 2: Requirements

- Define different types of requirements
- Consider ways in which we can evaluate task outcomes, i.e., have we satisfied requirements

OVERVIEW: DEFINING REQUIREMENTS
"Tasks" as a path towards requirements

Getting from **what people need to do (activities) …**
To **defining the system needed to support those activities (requirements)**

- **People**: Identify all users & other stakeholders who do or will perform the activity: groups, capabilities, motives,
- **Tasks**: Identify (and model) the human activity which the proposed interactive system will support: task, goals, conditions, current problems and strengths
- **Support**: Set focus and levels
  - Usability; constraints on the product's performance

What we've been working on

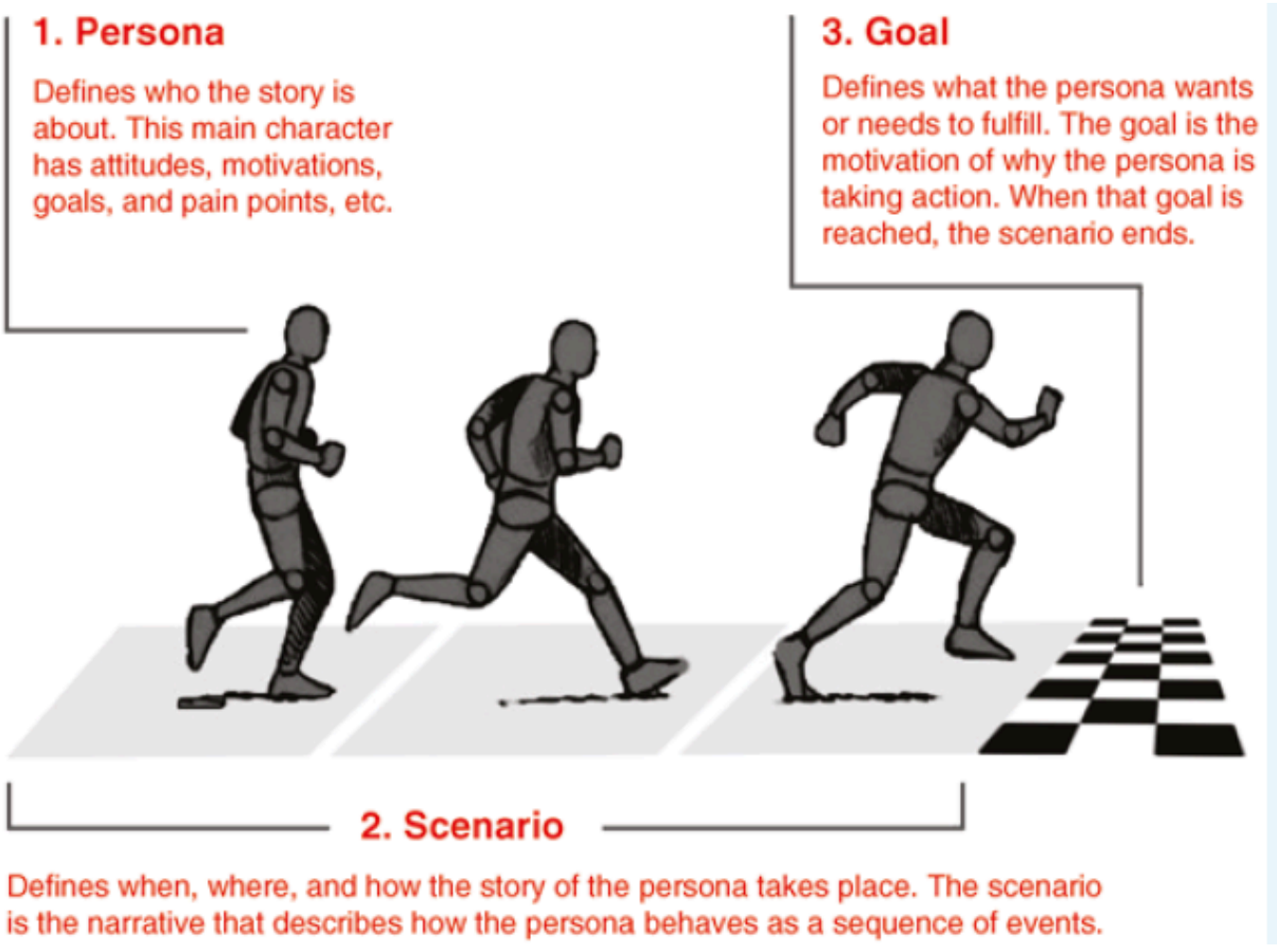
Today

A FEW DEFINITIONS TO START

SCENARIOS

- "Describes **human activities or tasks in a story** that allows exploration and discussion of contexts, needs, and requirements" (Sharp et al., 2019, p. 408)
- Can we wrapped around the *persona* to give more detail about user activities
- Include *context, activities, user experience and usability* goals
- Can explore how a user would react in a new situation/adapt to a new technology

See examples of scenarios in the Examples of Task Examples reading for today...



Source: http://www.smashingmagazine.com/2014/08/06/a-closer-look-at-personas-part-1/. Reproduced with permission of Smashing Magazine; as cited in Sharp et al., 2019, p. 415

USE CASES More general than scenarios – 3 components

Essential use cases specify:

(1) user intention
(2) user actions (steps),
(3) product responsibility (steps)

Does not specify the actual interaction

For the **task of retrieving an international visa**,

	User Intention	**System Responsibility**
Task:	find visa requirements supply required information	request destination and nationality obtain appropriate visa information
Steps: see how they stop short of specifying how the system implements them. ***System independent***	*Normal course:* ***typical set of steps***	1. Product asks for the name of the destination country 2. User provides the country's name 3. Product checks that the country is valid.
	Alternative course: ***atypical set of steps***	4. If the country name is invalid 4.1 The product provides an error message 4.2 The product returns to Step 1

DISCUSS: Difference between goals, activity, use case, scenario?

Then we'll move on to Tasks!

TASK EXAMPLES

IDENTIFY (AND MODEL) HUMAN ACTIVITY

What outputs might you have at the **end of modeling the human activity**?

- **User goals** (why they are doing the activity / tasks)
- Task **descriptions**, task **examples**
- Task **models**; normal steps and process; common breakdowns

TASK EXAMPLES

- Concrete, detailed examples of tasks people perform or want to perform that your system should support
- Categories:
  - Routine
  - Infrequent but important
  - Infrequent and incidental
- **Task examples should be design-independent!**
  - They describe *what* the user does /needs to do at a higher level, but leave flexible the details of *how.*
  - You should be able to "plug in" different interface designs to the task example, and "run them" like a computer program on a dataset, to compare how well the different designs support the task.

IDENTIFYING TASKS

- Field data, research literature, popular media, own experience
  - Describe expected set of users, and expected set of tasks
  - May need to verify or modify your assumptions later as you walk people through the early prototype
- If there are no real users or tasks…
  - Think again, there probably are!
    - Jeff Hawkins, inventor of the Palm Pilot (first popular handheld personal organizer), was said to have carried a small block of wood around in his shirt pocket … As various everyday situations arose, he would take out the block of wood and imagine how he would use the device.
  - …the same technique can be used to evoke a response from expected end-users.

[1]*see Sato and Salvador, Interactions 6(5)*

GROCERY LIST TASK EXAMPLE

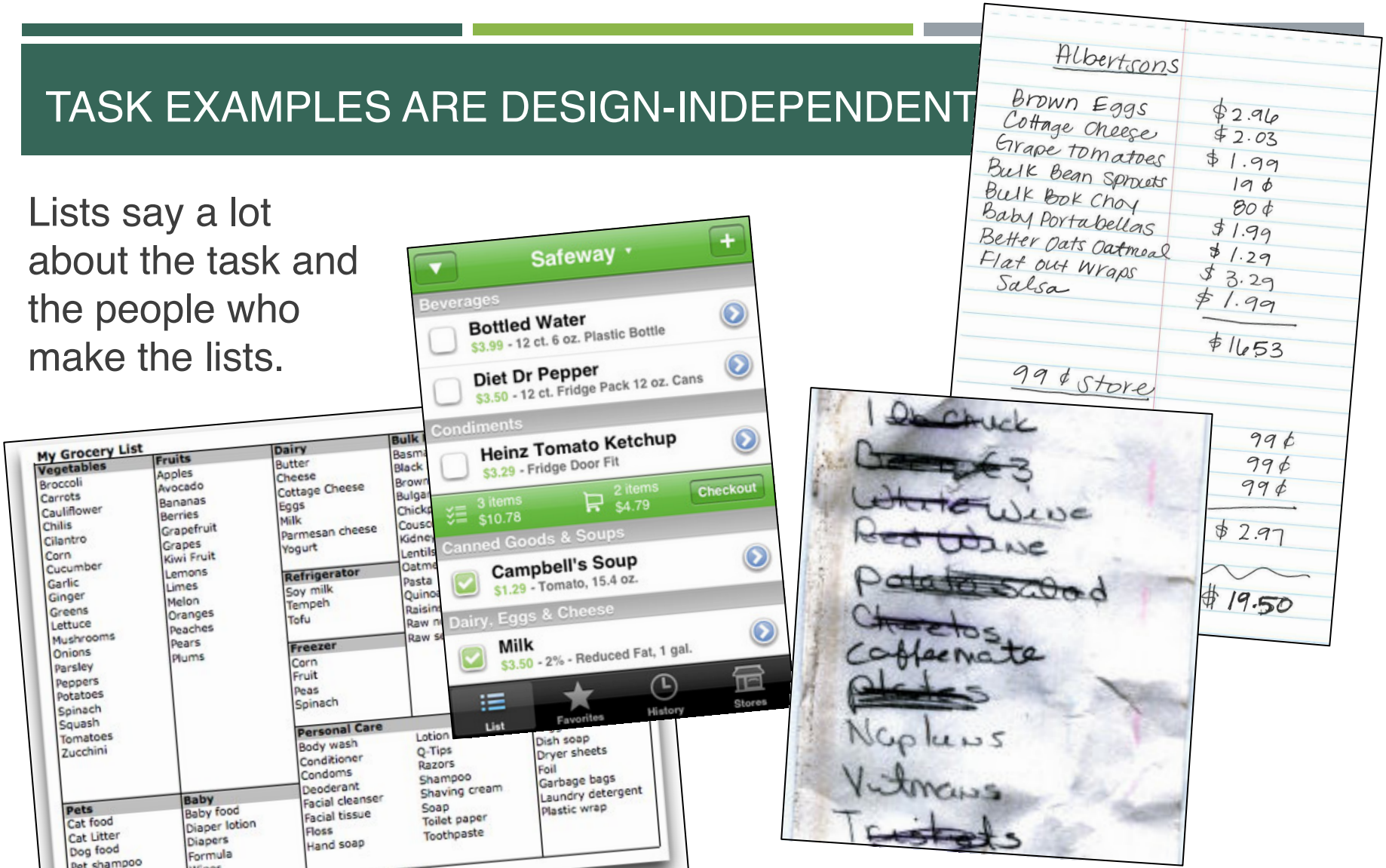
- Vik is doing the weekly menu planning for her family of 4. She chooses a set of recipes that suit the season, available prep time, current individual dietary eccentricities and her own preference at that moment.
- Many of this week's choices are regulars. She creates a shopping list of ingredients, ordered by where they can be found in the grocery store. Her partner Cam, who does the actual shopping and is more familiar with the store, supplies "feedback" on any errors she makes.
- When a recipe requires an ingredient that was already needed for an earlier day's meal, it is not added again. After getting through the week's meals, she adds a few regular items like milk, bread, cereal and juice. After Cam has left with the list, she realizes she's forgotten to check the pantry for staples like flour and rice.

This is the style of task example you will write for the Ideate milestone.

Observe what detail it includes, and also what it **excludes - design independence**!

TASK EXAMPLES ARE DESIGN-INDEPENDENT

Lists say a lot about the task and the people who make the lists.



LISTS ARE JUST ONE WAY PEOPLE VARY in how they do meal planning

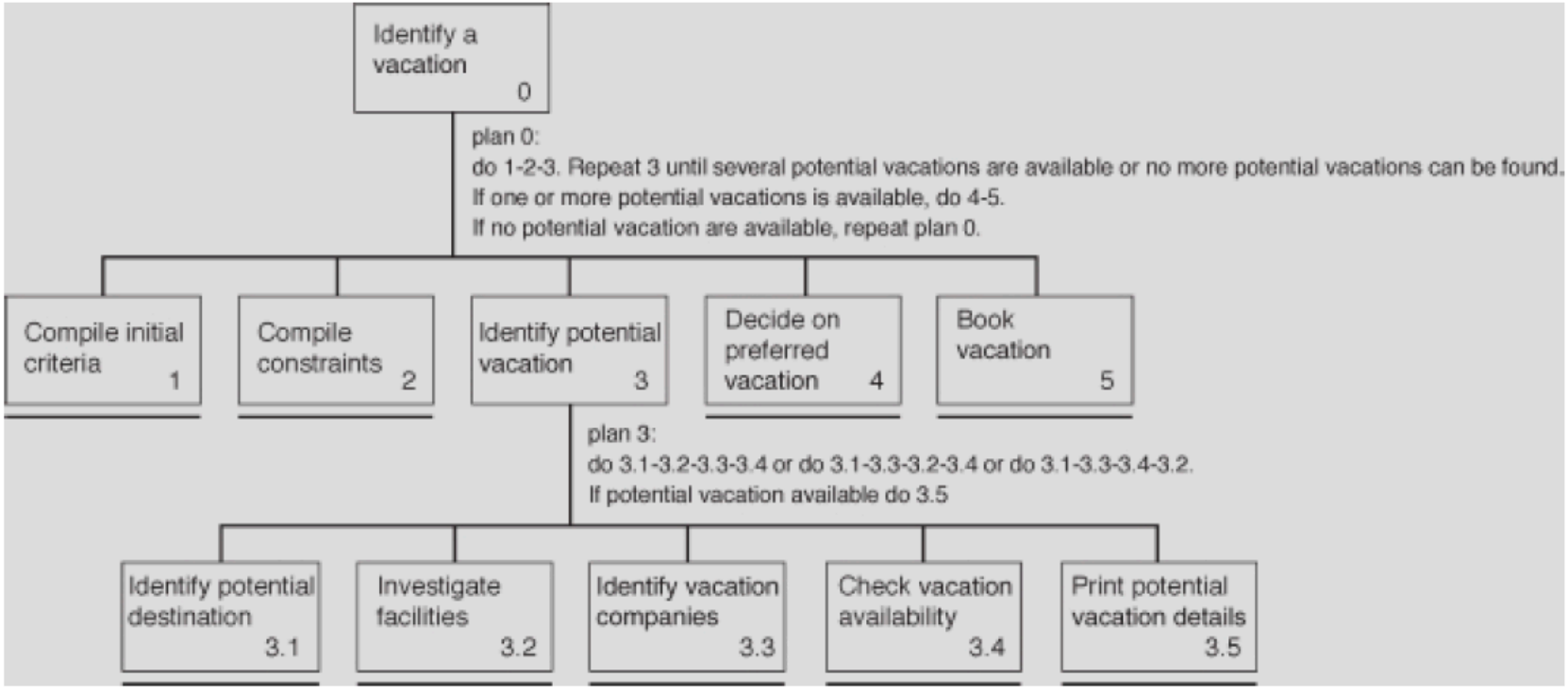
- Plan ahead, or last minute??
- Alone, or as a family?
- Does list maker also shop?
- Improvisation allowed?

TASK ANALYSIS: WHAT IS THE SITUATION OF CONCERN?

- The context for the design problem
  - Something is wrong that we want to change, or
  - Something could be improved upon
- Requires a **course of action** that will result in a **change** that **resolves the situation of concern**
  - ➔ this course of action (i.e. the solution) is what needs to be specified in requirements.

Situation of concern becomes clarified through data collection (Empathize) and other means (e.g., literature review) into a concrete **Problem Definition**.

Task Analysis: an umbrella term that covers techniques for **investigating** and **articulating** cognitive processes and physical actions, at a high level of abstraction and at an useful level of detail



Compare this kind of task description to a task example.
It abstracts Steps, Dependencies; is still design-independent

Preece, Sharp & Rogers. (2015). *Interaction design: beyond human-computer interaction*. John Wiley & Sons.

EXAMPLE: SCHEDULING MEETINGS

- Informal problem definition (situation of concern):
  - Hard to learn everyone's schedule & find a common free time
  - Participants respond slowly or incompletely to request
  - Complicated to respond in adequate detail
  - Individual schedules change → time no longer available
  - Shared calendars: privacy and system incompatibility
- Result: too much iteration; non-convergence
- Course of action:
  - Ideas?

ACTIVITY – PART I [10 MIN]

- Activity goal: practice breaking down and analyzing a human activity to start to generate requirements.

  Human activity: scheduling meetings
  Mission: schedule a meeting between project team members
- What steps are involved in this task?
  - Begin by stating **the goals for the steps**.
  - Remember design/interface independence! These goals should apply for … Doodle poll, email coordination, in meeting, etc.
- How can these steps go wrong?
- Create a diagram of this task to help answer these questions

SCHEDULING EXAMPLE - TAKEUP

Goals of individual sub-tasks (steps of the overall activity)?

- ???

Detailed steps involved for one of these?

- ???

How could these steps go wrong?

- ???

SCHEDULING EXAMPLE, CONT.

NEXT, break one of these down
(many possible ways)

- Find common empty spaces in calenda
  1. Ask all to communicate avail during a block; OR suggest times, get respor
  2. Examine manually or automatically
  3. Find common openings, if any
  4. If no, iterate with different time blocks or suggestions

some possible task goals:

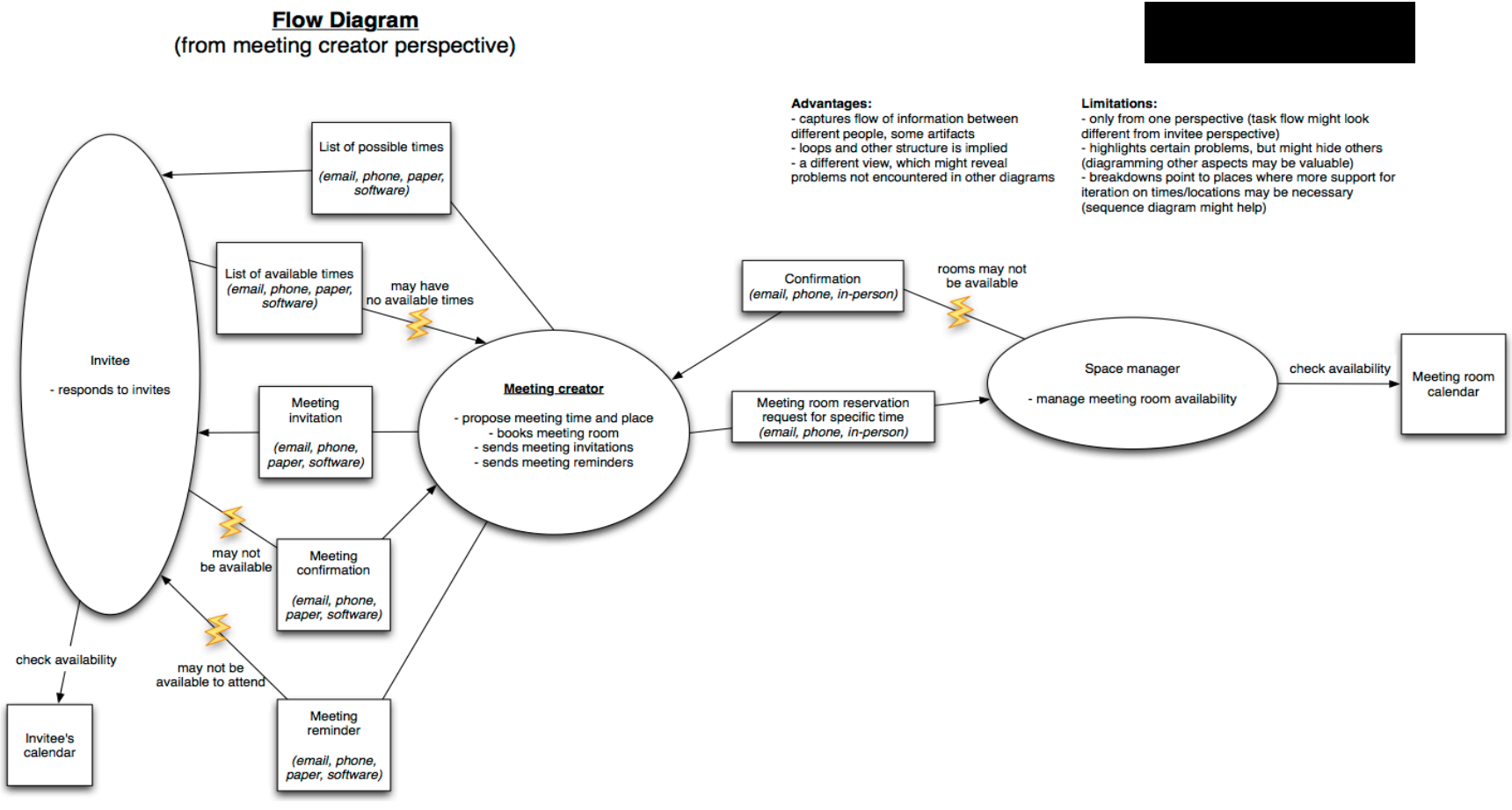
- identify who needs to be at meeting
- find common empty spaces in calendars
- identify a subset of empty spaces to suggest
- choose one » tell everyone
- receive confirmation that everyone still available
- if no, iterate
- identify location

How might this go wrong?

- what if people respond very slowly?
- what if people respond incompletely?
- what if there are no solutions?

ANALYZING TASK EXAMPLES

A POSSIBLE FLOW DIAGRAM OF MEETING TASK



DEPENDENCIES AMONG TASKS: TASK OBJECTS

- "**Task objects**" are **resources** required by tasks
  - Artifacts (files, lists, databases)
  - People (special expertise, authority, or knowledge)
  - Other processes, equipment or tasks
- Low-level subtasks typically focus on a single resource
  - The subtask is dependent on that resource (cannot be accomplished without it)
  - Once resource is available the task can be completed
- Higher-level / composite subtasks have multiple **dependencies**
  - The focus shifts as different subtasks are performed
  - Activity is suspended when resources are not available

*What are some **task objects** from scheduling model (last slide)?*

SIGNS OF TASK DEPENDENCIES?

- Joint use of task objects by different tasks
  - E.g., constrained access to shared files or databases
- Communication between people
  - May be direct (1:1 text or phone call) or indirect (group email)
- Synchronization with real-world physical and mechanical processes
  - E.g., need to meet in a physical space, or access some equipment
- Suspension
  - Blocking when resources (information, people, real-world processes) are not available

ACTIVITY – PART 2: TASK DEPENDENCIES [10 MIN]

Consider the Meeting Task Flow Diagram from earlier slide:

- What "**task objects**" (i.e. resources, parts of process, even things generated by the process) might be required to to carry out this task?
- Are there **dependencies** on/among these objects that could lead to conflicts or breakdowns?

1. Work in pairs [5min]
2. Work out all together, on screen [5min]

TAKEUP

OBJECTS for Scheduling task?

Other signs of task dependencies?

EXAMPLE

OBJECTS for Scheduling task?

Conflicts in their use **suggest dependencies**…

- Calendars
- Communication mechanisms (email, phone, cooler)
- “Leader” – meeting leader, secretary, program

Other signs of task dependencies?

- Can’t find time until have heard from all participants
- Participants can’t give feedback on times until told their choices

REQUIREMENTS

WHAT IS A REQUIREMENT?

- "...a statement about an intended product that specifies what it is expected to do or how it will perform"
  - Sharp et al.,2019, p. 387
- Can be operationally defined and measured
- Types of requirements:
  - Functional → what the product will do (its reason for existing)
    - E.g., ATM (cash withdrawal machine): allows people to withdraw money from account
  - Non-functional (aka "constraints") → characteristics of the product
    - ATM: permits this to take place in a safe and weather-protected environment

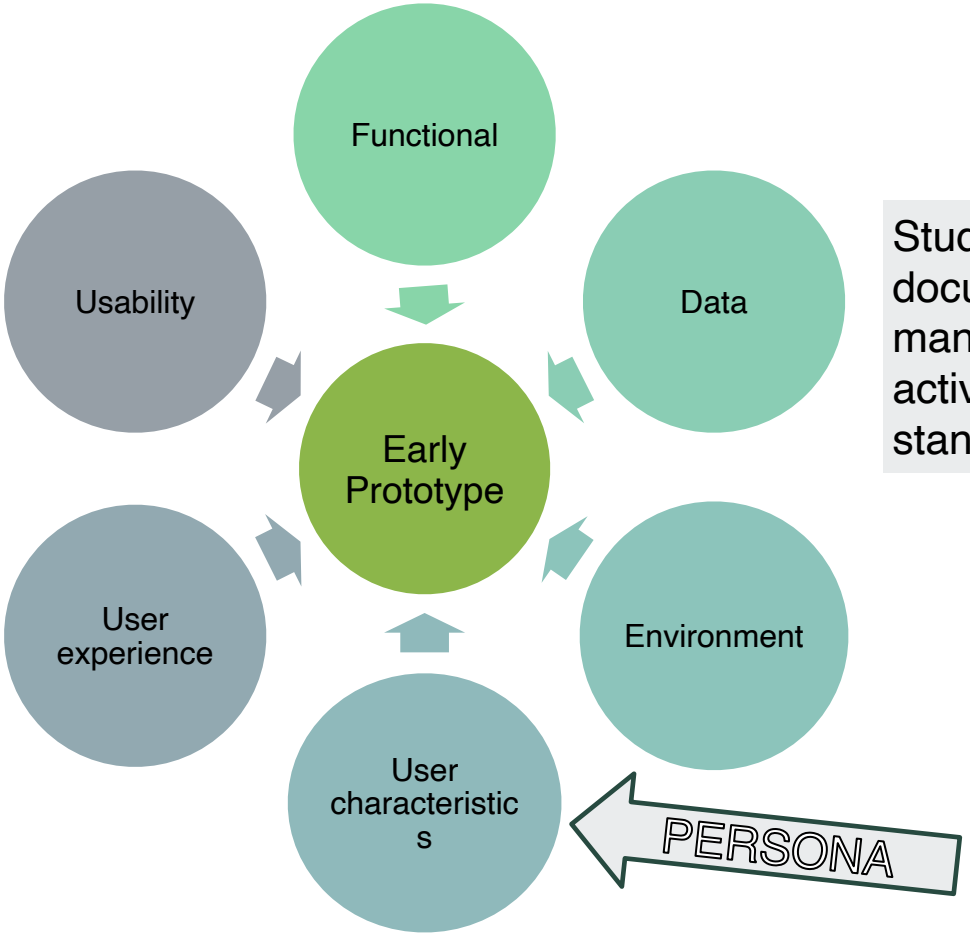
SIX COMMON REQUIREMENT TYPES

1. **Functional** requirements
2. **Data** requirements:
   - "type, volatility, size/amount, persistence, accuracy, and value of required data"
3. **Environmental** requirements or context of use
   - Physical environment
   - Social environment
   - Organizational environment
   - Technological environment
4. **User** characteristics: attributes of/user profile
   - i.e., individual differences
5. **Usability** goals, i.e., pragmatic
6. **User experience** goals; i.e., hedonic

GATHERING & REFINING REQUIREMENTS: where do they come from?

Walking through an existing system or prototype with users

Research with/ about users: interviews, observations, focus groups, diaries, previous studies



Studying documentation: manuals, activity logs, standards

THREE STEPS TO REQUIREMENTS

Usually in parallel

1. Identify all users & other stakeholders who do or will be involved with the activity: groups, capabilities, motives, needs
   - Describe (e.g., in personas);
   - Identify priority users / stakeholders
2. Identify (and model) the human activity which the proposed interactive system will support: task, goals, conditions, current problems and strengths
   - Describe in task examples, analysis, models

Following 1+2

3. Set focus and levels of support
   - Usability
   - Constraints on product's performance (how much it needs to do)

ACTIVITY – PART 3 – IF TIME: USER NEEDS [10 MIN]

- Identify potential users in this situation (are they all the same? different?)
- Brainstorm a list of general human needs or needs specific to these users that could apply to this task?

SCHEDULING EXAMPLE

The USER

examples of **general** needs:

- ???

examples of **specific** needs:

- ??

SCHEDULING EXAMPLE

OBJECTS for Scheduling task?
conflicts in their use suggest dependencies…

- Calendars
- Comm. Mech. (email, phone, cooler)
- "leader" – mtg leader, secretary

activity

Other signs of task dependencies?

- Can't find time until heard from all participants
- Participants can't supply time until told when to look

The USER

Examples of **general** needs:

- Social / cultural environments
  Are people more comfortable with email, telephone or just running into each other?
- Are some users more overwhelmed with information than others?

Examples of **specific** needs (recall "behavioral variables", from Personas)

- Frequency of use of computers/mobile devices; attentiveness to email, texting, other communication forms?
- Is there variation in how responsive they are?
- Do they have control over their time?
  i.e. how much freedom to decide what meetings they participate in?

CHOOSING USABILITY & USER EXPERIENCE TARGETS

- Choice of usability metrics affects the solution:
  - prioritize most important facets based on design goals:
    e.g. is speed most important, or is it very bad to make errors?
  - ease of learning can be important, especially for novices
- Levels of performance need to be quantified:
  - must know baseline performance first (pre-redesign)
  - then establish realistic target levels
  - make sure we can measure the changes → iterate

METRICS: HOW DO WE KNOW WE HAVE SUCCEEDED?

- Quantitative metrics: measured (countable) indicators of people's use of the interface:
  - Speed of performance
  - Incidence of errors
  - Ease of learning the system
  - User satisfaction
- Qualitative metrics: descriptive accounts that shed light on quantitative measures:
  - e.g. stories about user impressions and frustrations, and their change pre- and post design; critical incidents; product "buzz" …

PUTTING IT ALL TOGETHER: STATING REQUIREMENTS

One approach – list out and then prioritize each of:

- Supported activities (tasks and steps)
  - Tasks and processes involved that support the activity.
- User(s). (could be Personas)
  - Who does the task and what are their characteristic
- Level of support
  - What usability & user experience properties are im

locating a jointly available meeting time

-people with tight schedules who need to participate in meetings of 3+ participants
-people with “frequent” online access

-users can provide all requested information with 1 min of their time
-require no iteration
-respect privacy (e.g. posting shared calendars)

FORM OF THE SOLUTION (FINALLY, THE DESIGN!)

- Design can start by adding **constraints**
  - cost (time, money, expertise)
  - compatibility with specific hardware or software
  - market pressures (standards, “look and feel”)
  - many of these don’t come from the designers!
- **Multiple levels in the description** (and in the prototypes)
  - the social/cultural/physical environment
  - the user interface
  - the application software
  - the operating system
  - system resources (storage, networking, peripherals)

OTHER FACTORS IN CHOOSING A SOLUTION

- Existing intellectual property
  - technology owned or licensed by the organization
  - unique skills or knowledge in the organization
  - market share or reputation
- Innovation
  - technology becomes obsolete quickly
  - R&D requires time and effort
  - often incremental improvements are good enough
  - significant changes may be required sometimes