ESTABLISHING REQUIREMENTS
TASK CENTERED DESIGN

690A- Advanced Methods in HCI

Prof. Narges Mahyar

Slides from Prof. Joanna McGrenere and Dr. Leila Aflatoony
Includes slides from Prof. Karon MacLean and Jessica Dawson

TODAY

- Task centered design [40 min]
  - Task description: task example and analysis
  - Requirements and metrics
- In class activity [20 min]
  - Task analysis and description
- Discussion of reading [10 min]

LEARNING GOALS

- define and give examples of different types of requirements.

- compare/contrast: task description, need, problem statement, requirement, specifications, metrics, provide proper examples.

- give examples of HCI techniques that are suitable / helpful for setting requirements.

- be able to identify appropriate metrics for a given requirement (and outline what features good metrics have).

- explain 3 steps for requirements generation.
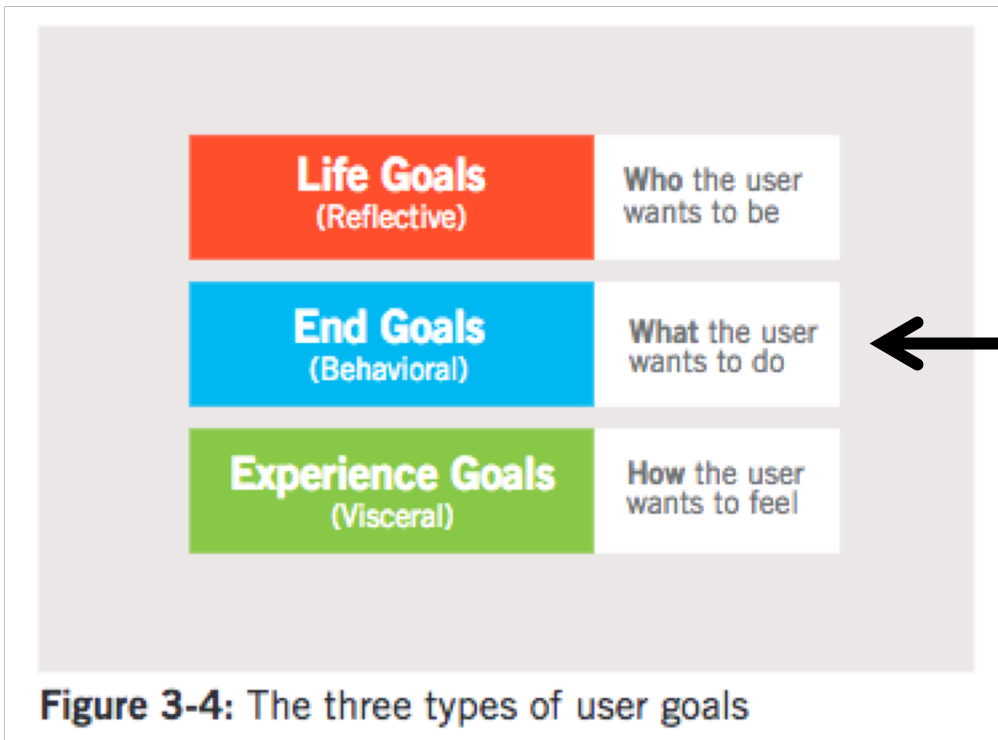
TASK CENTERED SYSTEM DESIGN

vs.

The User
a pretend person who
will mold themselves to
fit your system

Mary
a real person with real
constraints trying to get
her job done

TASKS ARE CENTRAL TO HCI

Recall Cooper's 3 levels of goal:

Figure 3-4: The three types of user goals

Tasks!

Classic HCI is all about tasks.

TASK DESCRIPTION:
TASK *EXAMPLES* WORK TOGETHER WITH TASK *ANALYSIS*

in HCI, establishing requirements typically begins with establishing tasks:

- ***task examples*** *describe tasks and (to some extent) users*

together with design prototypes, task examples are also a good way to evaluate initial designs at low cost and effort.

USERs -> this term we are using PERSONAS

So in this lecture, when you see User, think Persona.

TASK EXAMPLES

articulate concrete, detailed examples of tasks users perform or want to perform that your system should support

it is useful to categorize task examples:

- routine
- infrequent but important
- infrequent and incidental

Why?

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 preferences 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 Cameron, 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 incremented. After getting through the week's meals, she adds a few regular items like milk, bread, cereal and juice. After Cameron has left with the list, she realizes she's forgotten to check the pantry for staples like flour and rice.

KEY POINT TO NOTE

Task examples are interface independent, or as independent as possible.

What does that mean?

PEOPLE MAKE MANY KINDS OF GROCERY LISTS

lists say a lot about the task and the people who make the lists.
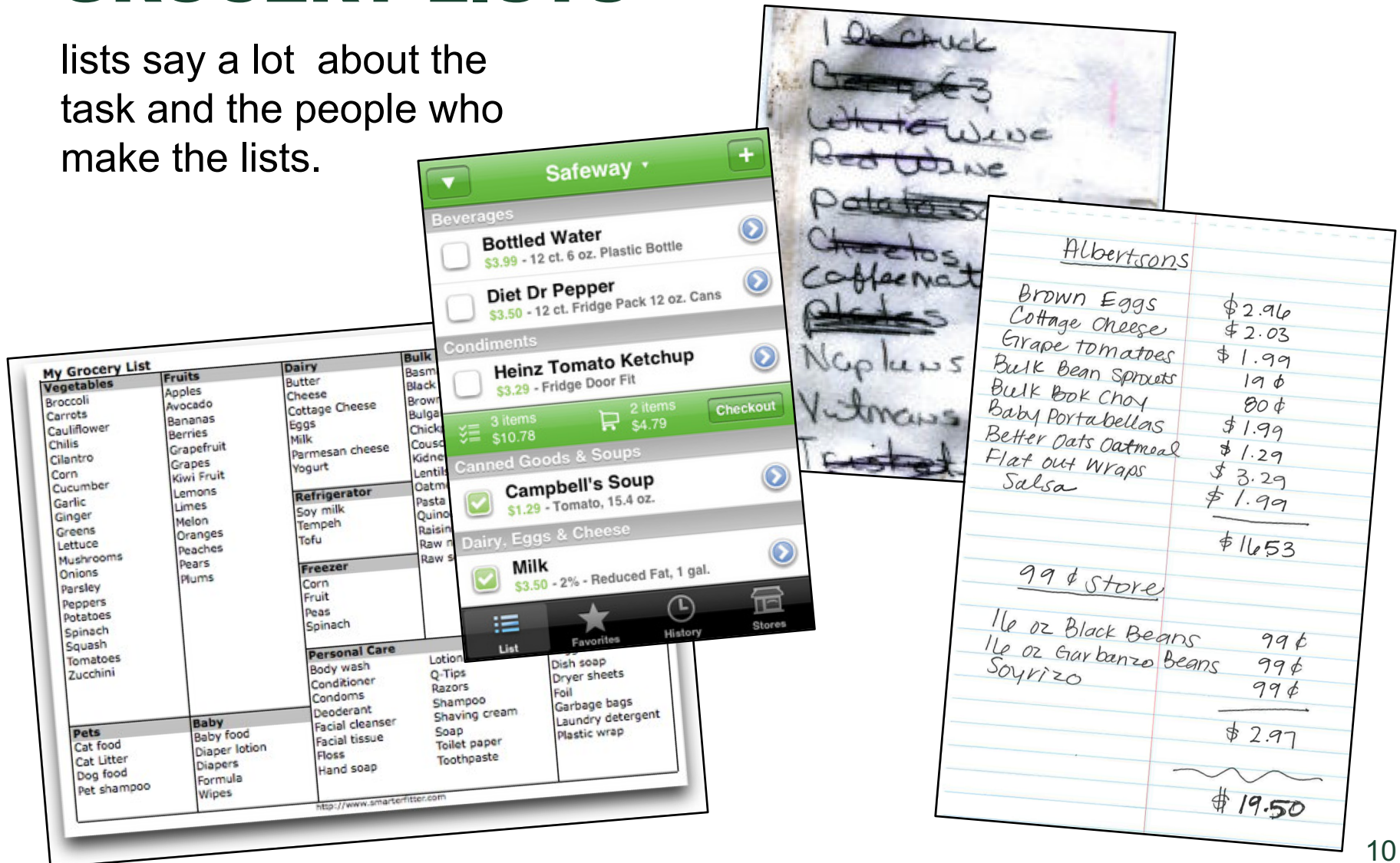
AND LISTS ARE JUST ONE OF THE WAYS THAT PEOPLE VARY IN HOW THEY DO THEIR MEAL PLANNING.

- plan ahead, or last minute??
- alone, or as a family?
- does list maker also shop?
- improvisation allowed?

RICH INFORMATION COLLECTED FOR "EMPATHIZE" …

… is used for creating personas AND for generating task examples.

Big Picture – WHEN do task examples and requirements happen?

GOALS

Understand USERS:
- who they are
- their key tasks

Understand DESIGN:
- design space and risks
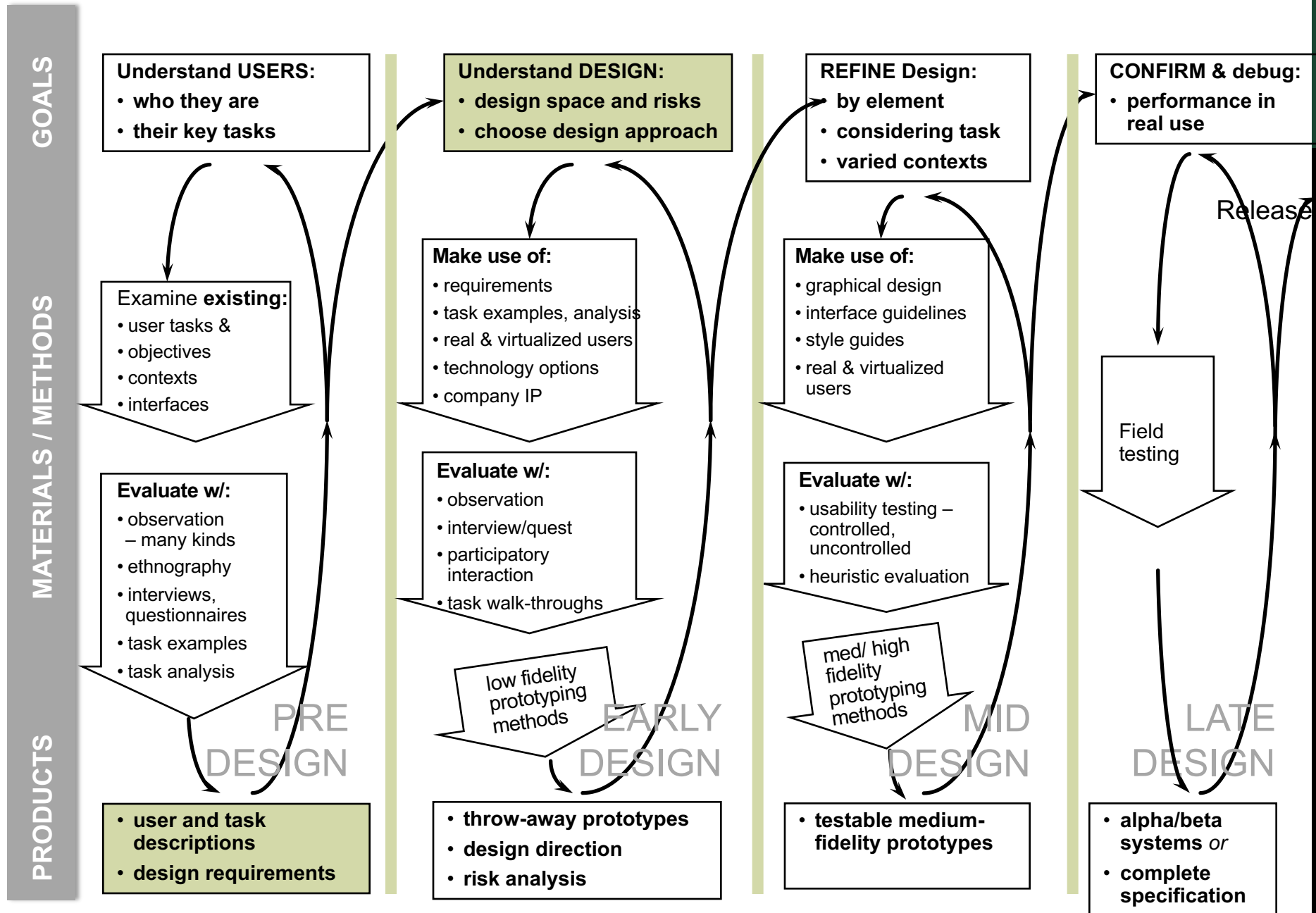- choose design approach

REFINE Design:
- by element
- considering task
- varied contexts

CONFIRM & debug:
- performance in real use

Release

MATERIALS / METHODS

Examine **existing:**
- user tasks &
- objectives
- contexts
- interfaces

Make use of:
- requirements
- task examples, analysis
- real & virtualized users
- technology options
- company IP

Make use of:
- graphical design
- interface guidelines
- style guides
- real & virtualized users

Field testing

Evaluate w/:
- observation – many kinds
- ethnography
- interviews, questionnaires
- task examples
- task analysis

Evaluate w/:
- observation
- interview/quest
- participatory interaction
- task walk-throughs

Evaluate w/:
- usability testing – controlled, uncontrolled
- heuristic evaluation

low fidelity prototyping methods

med/ high fidelity prototyping methods

PRE DESIGN

EARLY DESIGN

MID DESIGN

LATE DESIGN

PRODUCTS

- **user and task descriptions**
- **design requirements**

- **throw-away prototypes**
- **design direction**
- **risk analysis**

- **testable medium-fidelity prototypes**

- **alpha/beta systems** *or*
- **complete specification**

WHAT ARE REQUIREMENTS?

Two types of requirements…

1. functional requirements: what the interface must do

- **usefulness** -- scope, features…

2. non-functional requirements: *constraints* that development must live in:

- delivery time, maximum cost, delivery platform, supportability, sustainability…

- **usability / user experience**: what it should be like to use (a primary focus of HCI design)

ALL: clear, specific, defined in a MEASURABLE way

"USABILITY" FOCUS IS RARELY INDEPENDENT

meeting usability requirements will/should often influence functional requirements. e.g.

- needed *speed* of response from system to user
  → implementation platform

- desired user context – e.g. on-the-go
  → must work on mobile platform

- use in distracted environment – e.g. hospital tool
  → voice output, speech input
  → visual display requirements to support memory chunking

FUNCTIONAL VS. USABILITY REQUIREMENTS:

The following list of requirements includes functional, usability and user experience requirements that could be defined for a movie theatre website. Which requirement is the best example of a *functional requirement?*

a)   users must be able to buy electronic tickets in less than 3 distinct steps.

b)   users must feel that the buying of electronic tickets is easy.

c)   users must be able to buy electronic tickets using the website.

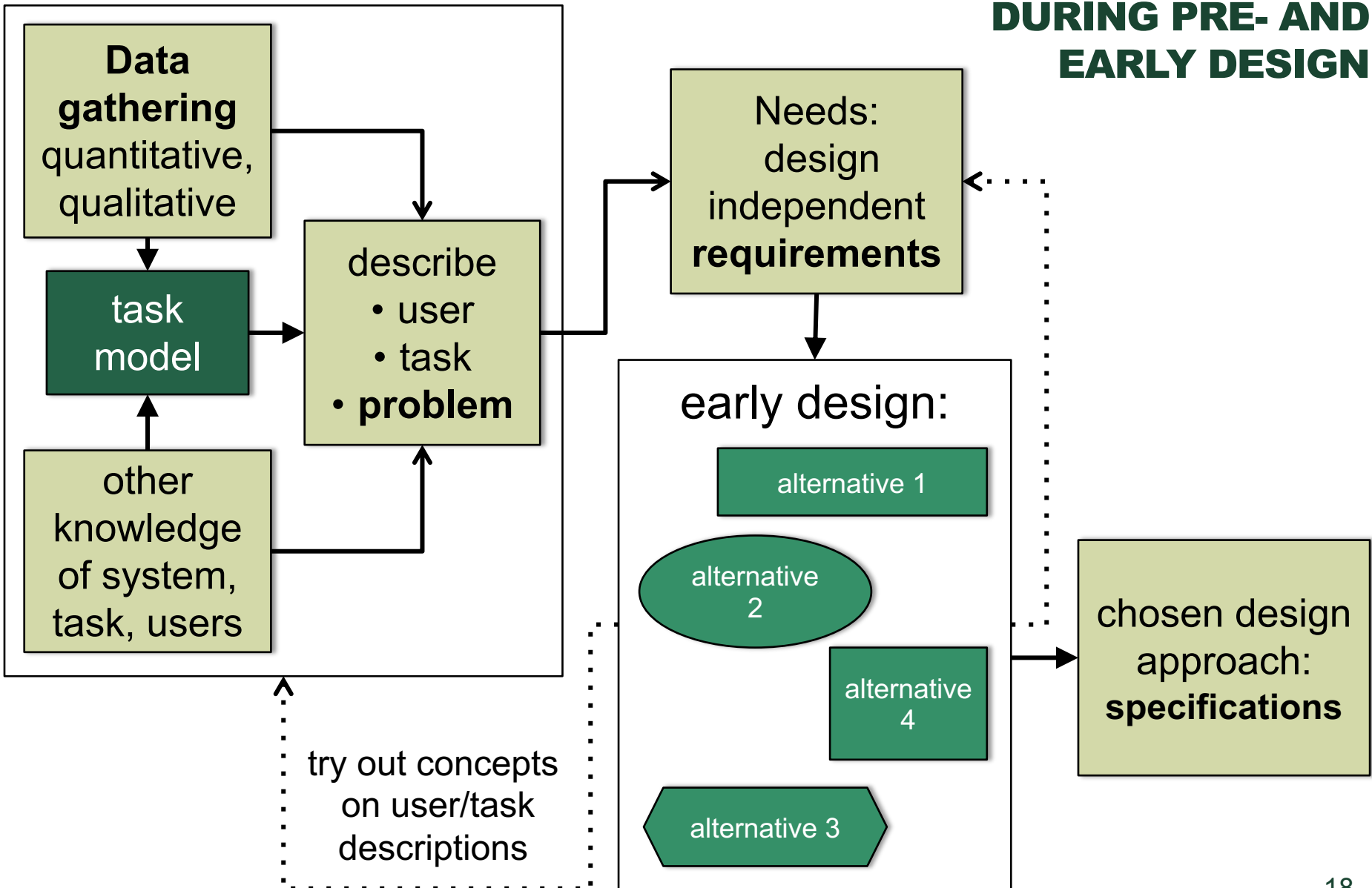d)   users must be able to learn how to buy electronic tickets on their first attempt.

functional: usefulness not usability
specifies a feature or capability

CAN REQUIREMENTS CHANGE?

- **requirements should be stable**
  if based on good data

- **but not rigid**
  they may shift over time, in particular as design reality dictates
  what is possible / feasible given other constraints.

Data gathering quantitative, qualitative

task model

other knowledge of system, task, users

describe
- user
- task
- **problem**

Needs: design independent **requirements**

early design:

alternative 1

alternative 2

alternative 4

alternative 3

try out concepts on user/task descriptions

chosen design approach: **specifications**

THREE STEPS TO REQUIREMENTS

1. **identify (and model) the human activity**
   which the proposed interactive system will support:
   task, goals, conditions; current problems and strengths

2. **identify all the users & other stakeholders**
   who perform or will perform the activity:
   groups, capabilities, motives, needs

3. **set focus and levels of support**
   which the system will provide (the system's usability):
   constraints on the product's performance, to support specific user-stakeholders you have targeted.

IDENTIFY (AND MODEL) THE HUMAN ACTIVITY

what outputs might you have at the end of this?

- goals
- task descriptions, task examples
- task models; normal steps and process; common breakdowns

EXAMPLE: SCHEDULING MEETINGS

Informal **problem definition:**

- 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-convergent

course of action:

- ideas?

Online scheduler is one obvious one which has taken hold; others?

21

ACTIVITY – PART I [10 MIN]

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

human activity: <u>scheduling meetings</u>
task: schedule a meeting between project team members.

1. what steps are involved in this task?

   - try stating goals for the steps.
   - remember *design/interface independence!*
     these goals should apply for …
     Doodle poll, email coordination, in meeting, etc.

2. how can these steps go wrong?

Create a diagram of this task to help answer these questions

22

SCHEDULING EXAMPLE, MY LIST…

some possible task goals:
- identify who needs to be @ meeting
- find common empty spaces in calendars
- identify a subset of empty spaces to suggest
- choose one » tell everyone
- receive confirmation that everyone still avail
- if no, iterate
- identify location

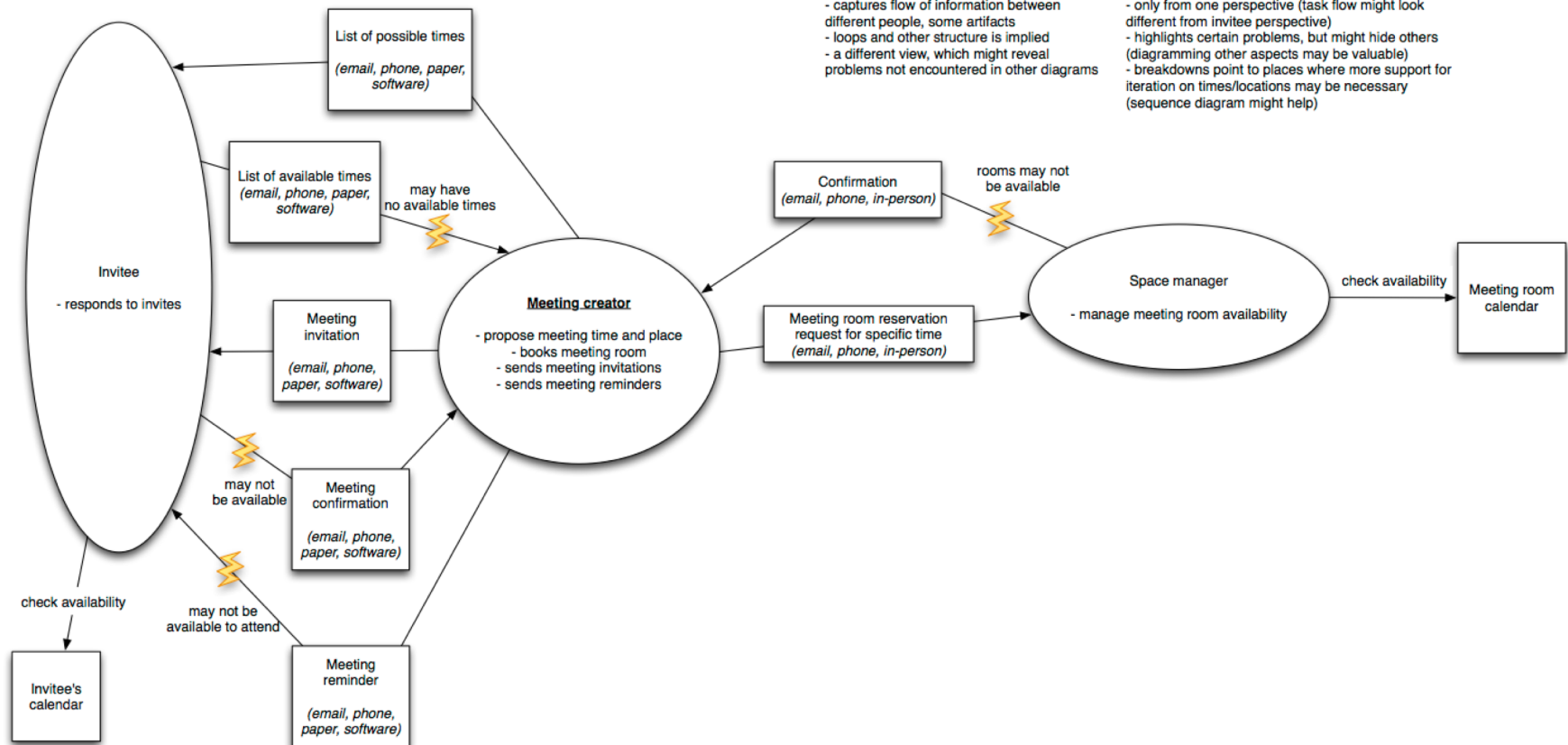*NEXT, break one of these down (many possible ways)*

- **find common empty spaces in calendars:**
  1. ask all to communicate avail during a block; OR suggest times, get responses
  2. examine, manually or automatically
  3. find common openings, if any
  4. if no, iterate with different time blocks or suggestions

How might this go wrong?
- what if people respond very slowly?
- what if people respond incompletely?
- what if there are no solutions?

A FLOW DIAGRAM OF MEETING TASK

Flow Diagram
(from meeting creator perspective)

Advantages:
- captures flow of information between different people, some artifacts
- loops and other structure is implied
- a different view, which might reveal problems not encountered in other diagrams

Limitations:
- only from one perspective (task flow might look different from invitee perspective)
- highlights certain problems, but might hide others (diagramming other aspects may be valuable)
- breakdowns point to places where more support for iteration on times/locations may be necessary (sequence diagram might help)

List of possible times

(email, phone, paper, software)

List of available times
(email, phone, paper, software)

may have no available times

Invitee
- responds to invites

Meeting invitation
(email, phone, paper, software)

Meeting creator
- propose meeting time and place
- books meeting room
- sends meeting invitations
- sends meeting reminders

Confirmation
(email, phone, in-person)

rooms may not be available

Space manager
- manage meeting room availability

check availability

Meeting room calendar

Meeting room reservation request for specific time
(email, phone, in-person)

may not be available

Meeting confirmation
(email, phone, paper, software)

check availability

Invitee's calendar

may not be available to attend

Meeting reminder
(email, phone, paper, software)

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 tasks typically focus on a single resource

- task cannot be accomplished without the resource
- once resource is available the task can be completed

(higher-level / composite) tasks have multiple dependencies

- the focus shifts as different sub-tasks are performed
- activity is suspended when resources are not available

SIGNS OF TASK DEPENDENCIES?

- joint use of task objects by different tasks
  e.g. access to shared files or databases

- communication between people
  may be direct (phone call) or indirect (memo)

- synchronization
  with real-world physical and mechanical processes

- suspension
  blocking when resources (information, people, real-world processes) are not available

ACTIVITY – PART 2
TASK DEPENDENCIES [5 MIN]

1. What "task objects" (i.e. resources, parts of process, even things generated by the process) might be required to meet this task?

2. Are there dependencies on these objects that could lead to conflicts or breakdowns?

Use your diagram to identify, or draw an entirely new diagram if required!

SCHEDULING EXAMPLE, MY LIST

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

THREE STEPS TO REQUIREMENTS

1. **identify (and model) the human activity**
    which the proposed interactive system will support:
    task, goals, conditions; current problems and strengths

2. **identify all the users & other stakeholders**
    who do or will perform the activity:
    groups, capabilities, motives, needs

3. **set focus and levels of support**
    which the system will provide (the system's usability):
    constraints on the product's performance, to support specific user-stakeholders you have targeted.

THE USER

need to understand general human needs:

- physical and cognitive abilities
- social and cultural environments
- use **models** of human behavior to test ideas

need to understand specific human needs:

- individuals have different skills and requirements
- they have responsibilities and authority in organizations
- they are expected to have a certain level of training
- they have specific access to tools and resources

Do not assume the user is "like you", or "normal"

ACTIVITY – PART 3
USER NEEDS [10 MIN]

1. Identify potential users in this situation (are they all the same? different?)

2. Brainstorm a list of general human needs or needs specific to these users that could apply to this task?

For developing Personas – What are the relevant variables? What can their range of values be?

SCHEDULING EXAMPLE, MY LIST

OBJECTS for Scheduling task?
conflicts in their use suggest dependencies…
- calendars
- Comm. Mech. (email, phone, cooler)
- "leader" – mtg leader, secretary, program

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, *more than they can humanly process*?

examples of **specific** needs:

- do all have laptops? are some reliant on mobile devices?

- is there variation in how responsive they are?

- do they have control over their time – i.e. are they permitted to decide what meetings they should / should not go to?

THREE STEPS TO REQUIREMENTS

1. **identify (and model) the human activity**
    which the proposed interactive system will support:
    task, goals, conditions; current problems and strengths

2. **identify all the users & other stakeholders**
    who do or will perform the activity:
    groups, capabilities, motives, needs

3. **set focus and levels of support**
    which the system will provide (the system's usability):
    constraints on the product's performance, to support specific user-stakeholders you have targeted.

FOCUS: WHICH USERS WILL YOU SUPPORT?

usually the intersection of

- greatest need
  and thus opportunity for business

plus

- feasibility
  we can identify a way to help

For example: you might choose NOT to support student users who **do not have** mobile devices.

Why?

How does this relate do the different types of Personas?

34

METRICS: HOW WE KNOW IF 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:

- really important
- e.g. *stories* about user impressions and frustrations, and their *change* pre- and post design;
critical incidents; product "buzz" …

CHOOSING USABILITY 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**

PUTTING IT ALL TOGETHER
STATING REQUIREMENTS

no single right way to write requirements; lots of companies have specific methods, tools, etc.

one approach – list out and then prioritize each of:

    1. supported activities (tasks and steps)

        tasks and processes involved that support the activity.

    2. user(s)

        who does the task and what are their characteristics?

    3. level of support

        what usability properties are important?

putting it all together
example – meeting scheduling

1. **supported task**
   - locating a jointly available meeting time
2. **users**
   - people with tight schedules who need to participate in meetings of 3 or more participants
   - people with "frequent" online access
3. **level of support**
   - users can provide all requested information with **1 minute** of their time
   - require no iteration
   - respect privacy (e.g. posting shared calendars)

WHERE DOES TECHNOLOGY COME IN?

tools support tasks

- new tools should **improve performance** of a task
- tools are often **specific** to the tasks they support
- tools must be acceptable / desirable to users

systems support processes

- systems have to support **links** between tasks
- often tasks are **automated** using technology
- tasks have to be **supported** in a consistent manner
- desirable to **reduce dependencies**
- desirable to **reduce task complexity**

one way (of many) to use your task description…help you find solutions!

DISCUSSION ON REQUIREMENT READINGS [10 MIN]

Get into group of 3-4 answering the following questions:

- What surprised you? or
- What you disagreed with?
- Others?

ON DECK…

Next class (Tuesday) …

- Readings (as posted) and researcher journal

- Third project milestone: Ideate

  - due on March 18th

EXTRA SLIDES

THE FORM OF THE SOLUTION
(FINALLY, THE DESIGN!!):

But, design starts 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 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