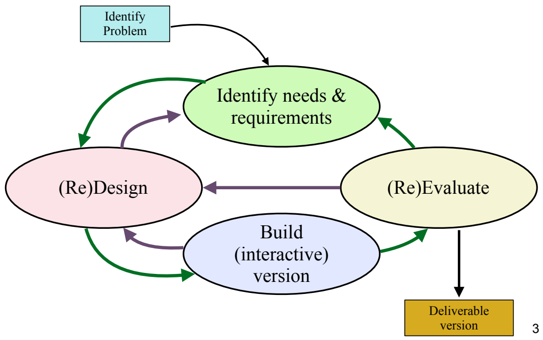
CPEN441: user interface design

needs and requirements

when we’re done today…

- know **four basic components** of a problem definition
- be able to create a **hierarchic task description**
- understand why we need **quantitative usability metrics** and give examples
- be able to create a **one-sentence problem statement**
- have awareness of other approaches to problem definition
- begin first steps of Design

PRS simple interaction design model



needs and problem definition

define the problem we are solving so we can systematically solve it

important for project proposals!

Forms of problem statements

- one-line problem statement
- problem statements appear in different forms
  - 5 steps identification by Business Online Learning
    - https://www.youtube.com/watch?v=7dKmvkg6H8Y
  - There is even an app to help with forming problem statement
    - ex.. RPM Academy
    - https://www.youtube.com/watch?v=yEZf_ZOg9Sw
- Key is to make sure you create one
- Talking to/Observing people is critical!

the situation of concern:

is the **context** for the design problem -
something is wrong *that we want to change,* ***or***
something could be improved upon

→ some **course of action** is required

- this will result in a **change** that resolves the situation

to identify this action, we must identify a possible **causal link:**

- a **human activity** that technology can support
- **improved performance** in this activity will affect the situation of concern

example: scheduling meetings

- 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

➡ what's the result?

- course of action:
  - ideas?

fundamental components of a problem definition

1. identify the **human activity**
   which the proposed interactive system will support;
2. identify the people (**users**)
   who will perform the activity;
3. set the **levels of support**
   which the system will provide (the system's **usability**)
4. select the basic **form of solution**
   to employ in the design problem

Identify the human activity

- Observe
- Interview
- Analyze existing tools
- Try it for yourself/team
- Anything that helps you understand what people do

Document

describing the human activity

Goals and subgoals

- what people want to achieve
- tangible outcomes

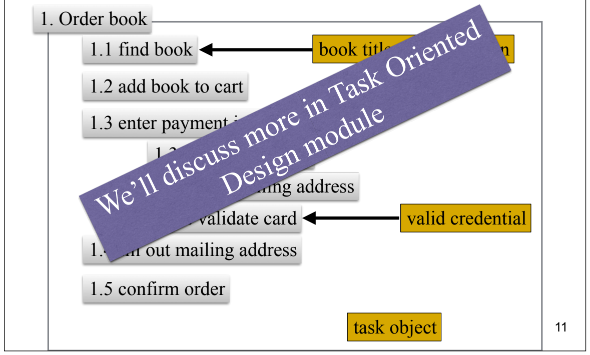
Tasks and Subtasks

- basic unit of human activity
- made up of a sequence of steps (usually linear)
- has a specific, single goal

Processes

- a linked set of tasks
- tasks may be performed in serial or parallel
- the process has one or more higher-level goals
- tasks in a process depend on varying resources
- dependencies often exist among the task

Example HTD



We’ll discuss more in Task Oriented Design module

fundamental components of a problem definition

1. identify the **human activity**
   which the proposed interactive system will support
2. identify the people, or **users**
   who will perform the activity
3. set the **levels of support**
   which the system will provide (the system’s **usability**)
4. select the basic **form of solution**
   to employ in the design problem

the user

need to understand **general** human:

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

need to understand **specific** human:

- 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

fundamental components of a problem definition

1. identify the **human activity**
   which the proposed interactive system will support;
2. identify the people, or **users**
   who will perform the activity;
3. set the **levels of support**
   which the system will provide (the system's **usability**)
4. select the basic **form of solution**
   to employ in the design problem

usability: some quantitative metrics
(how we know if we have succeeded)

here are some examples:

- speed of performance (objective)
- incidence of errors (objective)
- ease of learning the system (objective?)
- user satisfaction (subjective)

objective quantitative usability metrics

- performance
- speed
  - how long it takes to perform an activity
  - how many people it requires to complete
- error rate
  - how often will errors happen
  - how critical will they be to success

obj/sub quantitative usability metrics

- learning to use the system
  - how much time will it take
  - what background is required
  - what tools are available to help
  - how much speedup do experts obtain
- satisfaction
  - do users ‘like’ the system
  - would they recommend it to their friends/co-workers?
  - e.g., quantitative subjective measures like questionnaires based on technology acceptance model (TAM)

deriving more quantitative usability metrics

- recovery from errors
  - can users fix mistakes
  - how difficult is it
- retention of learned skills
  - do users continue to perform well after breaks
- ability to customize
  - can users control how they use the system
- ease of reorganization of activities
  - can the system be used to do other things
  - does the system support changing needs

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 should be **quantified:**

- must know **baseline performance** first (pre-design)
- then establish **realistic target levels**
- make sure we can **measure** the changes

fundamental components of a problem definition

1. identify the **human activity**
   which the proposed interactive system will support;
2. identify the people, or **users**
   who will perform the activity;
3. set the **levels of support**
   which the system will provide (the system’s **usability**)
4. select the basic **form of solution**
   to employ in the design problem

the form of the solution

design is the process of **adding constraints**

- cost (time, money, expertise)
- compatibility with specific hardware or software
- market pressures (standards, “look and feel”)

multiple levels in the description

- the social/cultural/physical environment
- the user interface
- the application software
- the operating system
- system resources (storage, networking, peripherals)

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

fundamental components of a problem definition

1. identify the **human activity**
   which the proposed interactive system will support;
2. identify the people, or **users**
   who will perform the activity;
3. set the **levels of support**
   which the system will provide (the system's **usability**)
4. select the basic **form of solution**
   to employ in the design problem

the one-line problem statement

supported activity

- what tasks or processes are involved and how do they support the activity?

user(s)

- who does the activity and what are their characteristics?

level of support

- what usability factors will we consider important?

form of solution

- what technologies can we employ and how can they be combined into a system?

example: scheduling meetings

- supported activity
  - locating a jointly available meeting time
- users
  - people with tight schedules who need to participate in meetings of 3 or more participants
  - people with "frequent" online access
- Level of support?

Who are these "people"?

Level of support

which of these apply to the scheduling system?

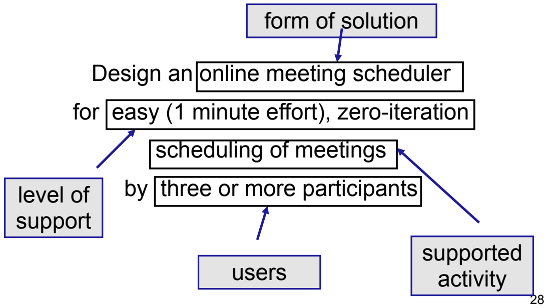
- speed of performance
- incidence of errors
- ease of learning the system
- user satisfaction
- recovery from errors
- retention of learned skills
- ability to customize
- ease of reorganization of activities

example: scheduling meetings

- supported activity
  - locating a jointly available meeting time
- users
  - people with tight schedules who need to participate in meetings of 3 or more participants
  - people with "frequent" online access
- 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)
- form of solution
  - online meeting scheduler

where does this come from?

the **one-sentence** problem statement
scheduling meetings



Design an online meeting scheduler
for easy (1 minute effort), zero-iteration
scheduling of meetings
by three or more participants

where does technology come in?

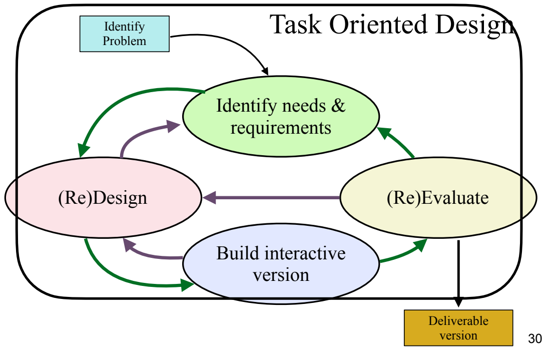
tools support tasks

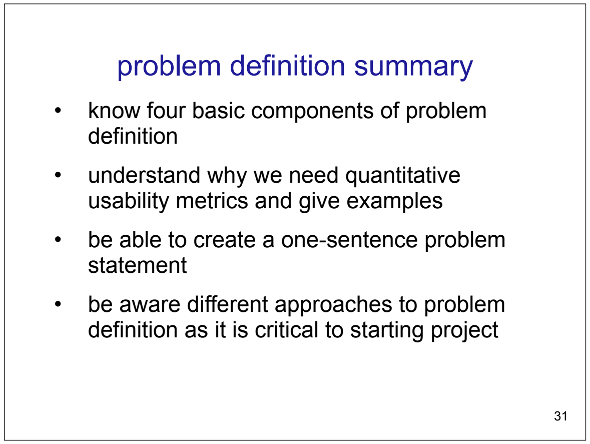
- new tools should **improve performance** of a task
- tools are often **specific** to the tasks they support
- tools must be acceptable / desirable to users

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**

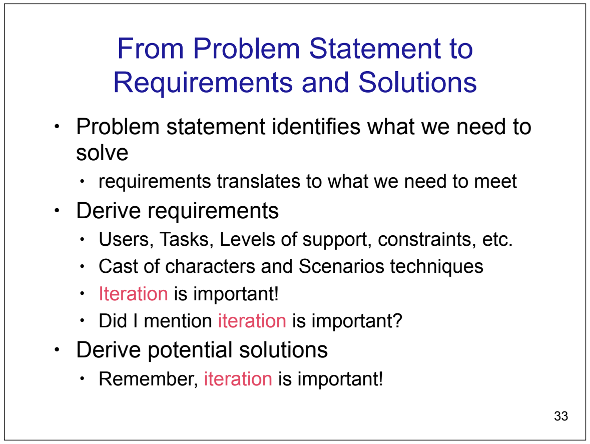
Where does design process fit?





problem definition summary

- know four basic components of problem definition
- understand why we need quantitative usability metrics and give examples
- be able to create a one-sentence problem statement
- be aware different approaches to problem definition as it is critical to starting project



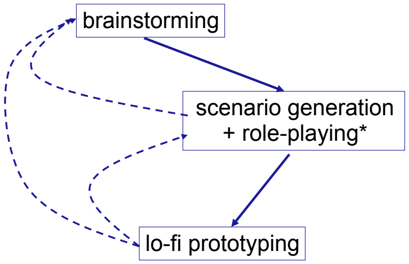
From Problem Statement to Requirements and Solutions

- Problem statement identifies what we need to solve
  - requirements translates to what we need to meet
- Derive requirements
  - Users, Tasks, Levels of support, constraints, etc.
  - Cast of characters and Scenarios techniques
  - Iteration is important!
  - Did I mention iteration is important?
- Derive potential solutions
  - Remember, iteration is important!

Design: concept generation and requirements

- Once we have problem we need to start to solve it:
  - require concept generation
  - requirement definitions
- practice two methods for concept generation & initial design processes

typical concept generation sequence for requirements



* Refer to PRC for excellent discussion plus more techniques.

characteristics of these methods

- Goals
  - **explore space broadly**
  - **gain insight** into possible design directions
  - provide **rich descriptions**
- Characteristics
  - low-overhead
  - low-cost
  - augment requirements documents to reflect actual user needs
  - can use **with** or **without** direct involvement with users
    - value is related to how well you know them (i.e. during problem definition stages)
  - useful at all stages of design

How to Brainstorm

- Group of 3-5 people

- Start with a list of brainstorm topics
- Use a facilitator & note taker
- Postpone and withhold your judgment of ideas: **never criticize**
- Encourage **wild and exaggerated ideas**
- **Quantity counts** at this stage, not quality
- Switch **topics** when the popcorn slows down
- **Build** on the ideas put forward by others
- Every person and every idea has **equal worth**

More on Brainstorming

- Creating ideas is challenging
  - many strategies to help
    - Ed Muzio:
      - https://www.youtube.com/watch?v=9K8W4ooygUU
    - 6 creative ways to brainstorm from Vertical Measures:
      - https://www.youtube.com/watch?v=yAidvTKX6xM
- no absolute rules other than:
  - don’t criticize/evaluate
- can be used whenever you are ‘stuck’

“Cast of characters/Personas”

- develop a set of individual representative users who span the possibilities of those you expect
  - ➡try to base them on people you have studied and/or have real experience with
- give them:
  - ➡ names, personalities, jobs, hobbies, lifestyles, needs and interests
- imagine each member of your cast using your interface, and what he/she would want from it
- helpful to role play with your team

“scenarios”

- informal narrative of that expands what, why and how of activities and tasks that people are doing
  - ➡usually tied to personas
- provide a specific example of a person progressing through the problem space
- provides a common story that whole team uses to communicate how people solve specific problems
- has expanded to be a common strategy for feature requirements in s/w eng (i.e. stories)
- use role playing to help articulate scenarios

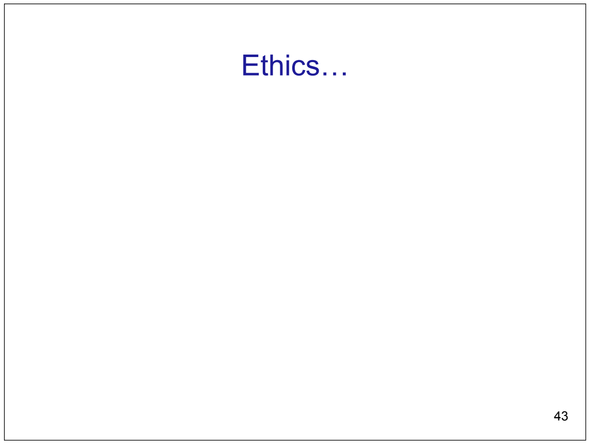
summary

one form of problem definition

be aware of a typical concept generation sequence

have practiced two methods for concept generation & early design

Practice Exercises/Tutorials



Ethics…