INTRODUCTION TO USER EXPERIENCE DESIGN PROCESSES IN SOFTWARE

CST315 – Fall 2015 Revision

Arizona State University

UX Design Principles

User-Centered Design

Techniques and Tools

Some notes from Sommerville 9th edition

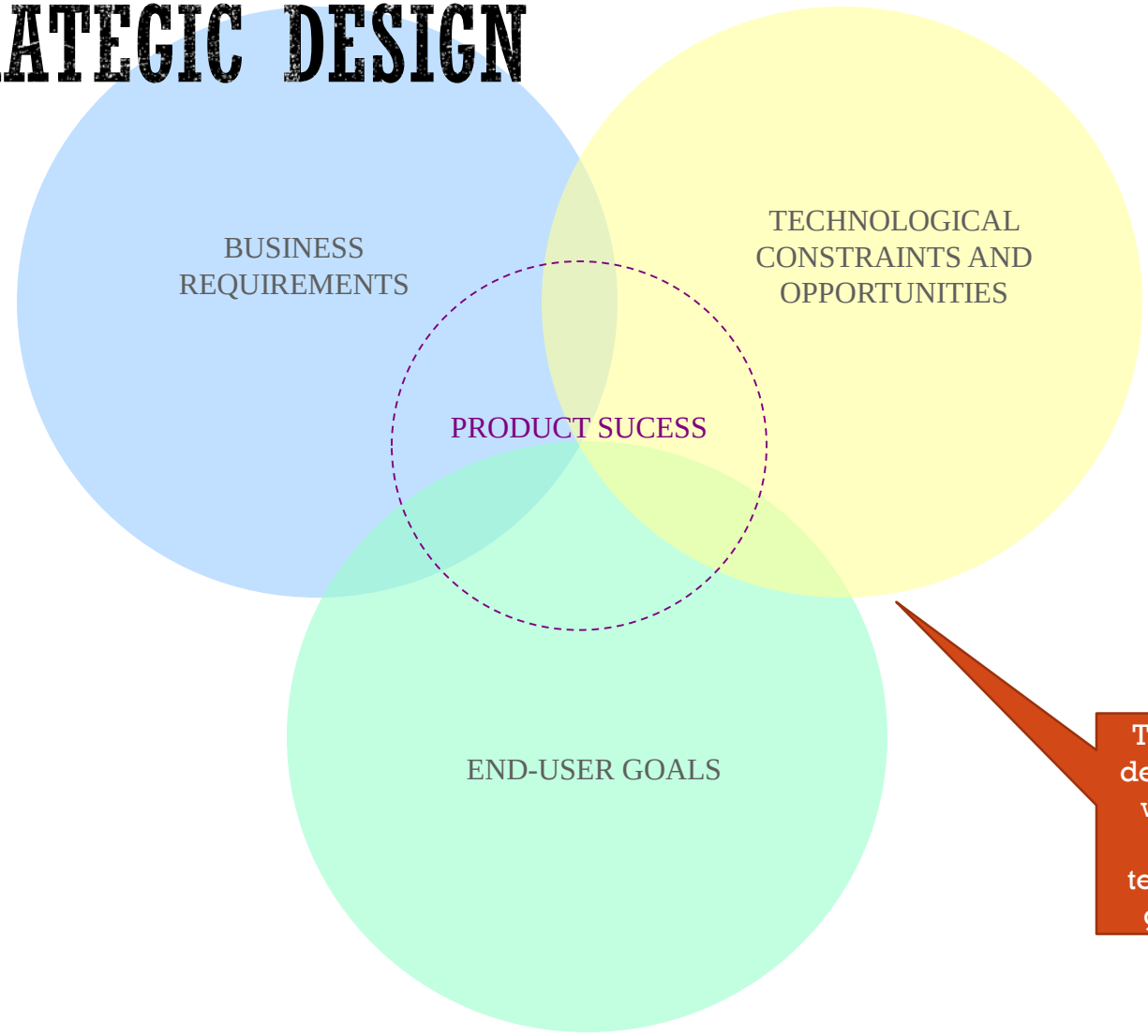
WHAT IS USER CENTERED DESIGN?

We will take a standards approach in this module – hence the preparation material.

We'll focus on four aspects:

PRINCIPLES Understand Target End-Users Prototype and Test Design with Target End-Users	PROCESS GUIDELINES ISO 13407 ISO 9241-11
STRATEGIC DESIGN Answer the "What" to Design	TACTICAL DESIGN Answer the "How" to Design It

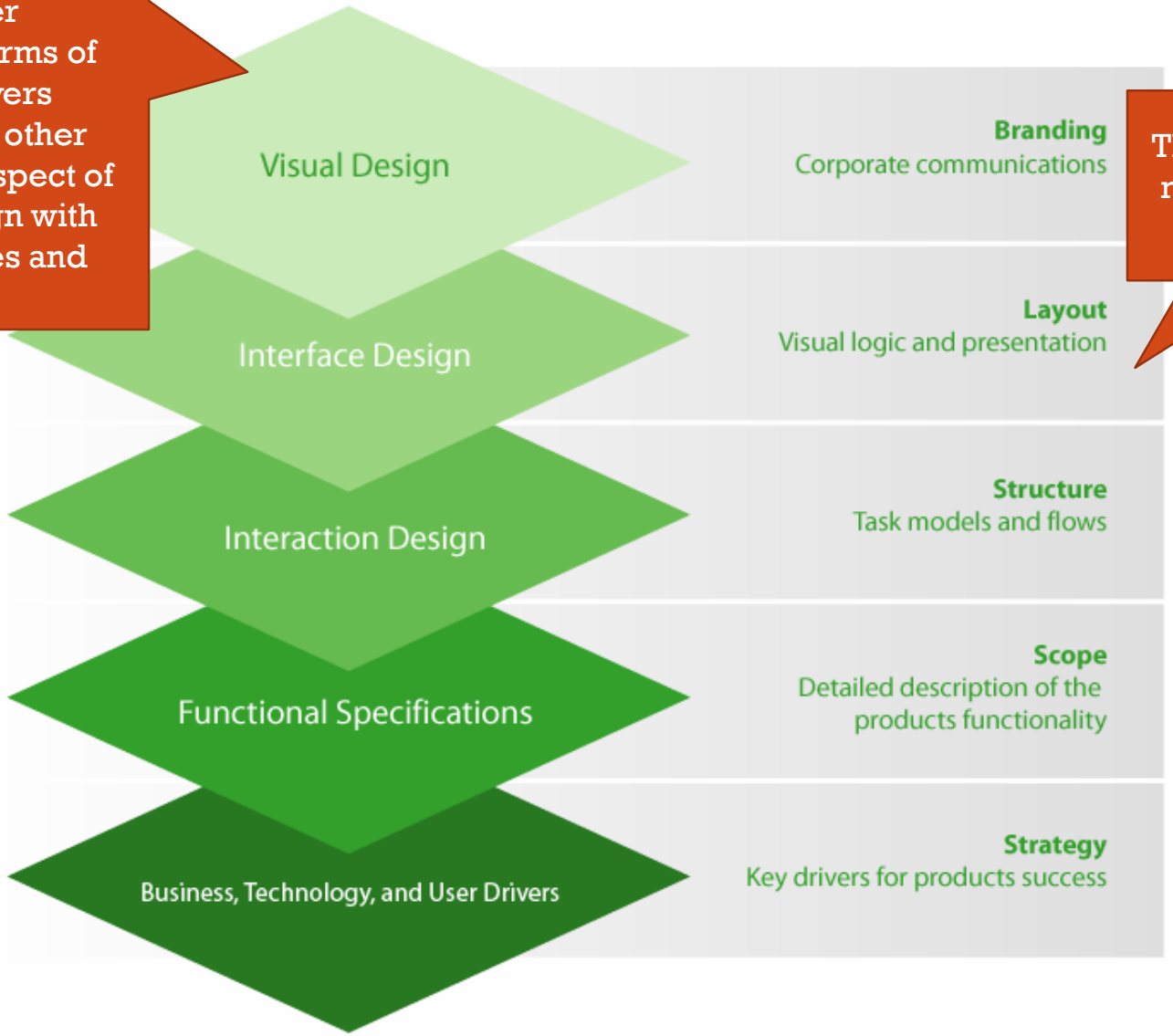
STRATEGIC DESIGN



To be successful in designing a product, we must consider requirements, technology and use goals, in parallel.

ELEMENTS OF THE USER EXPERIENCE

We can think of a design underlying a user experience in terms of layers, where layers depend on each other and each is an aspect of the overall design with specific purposes and techniques.



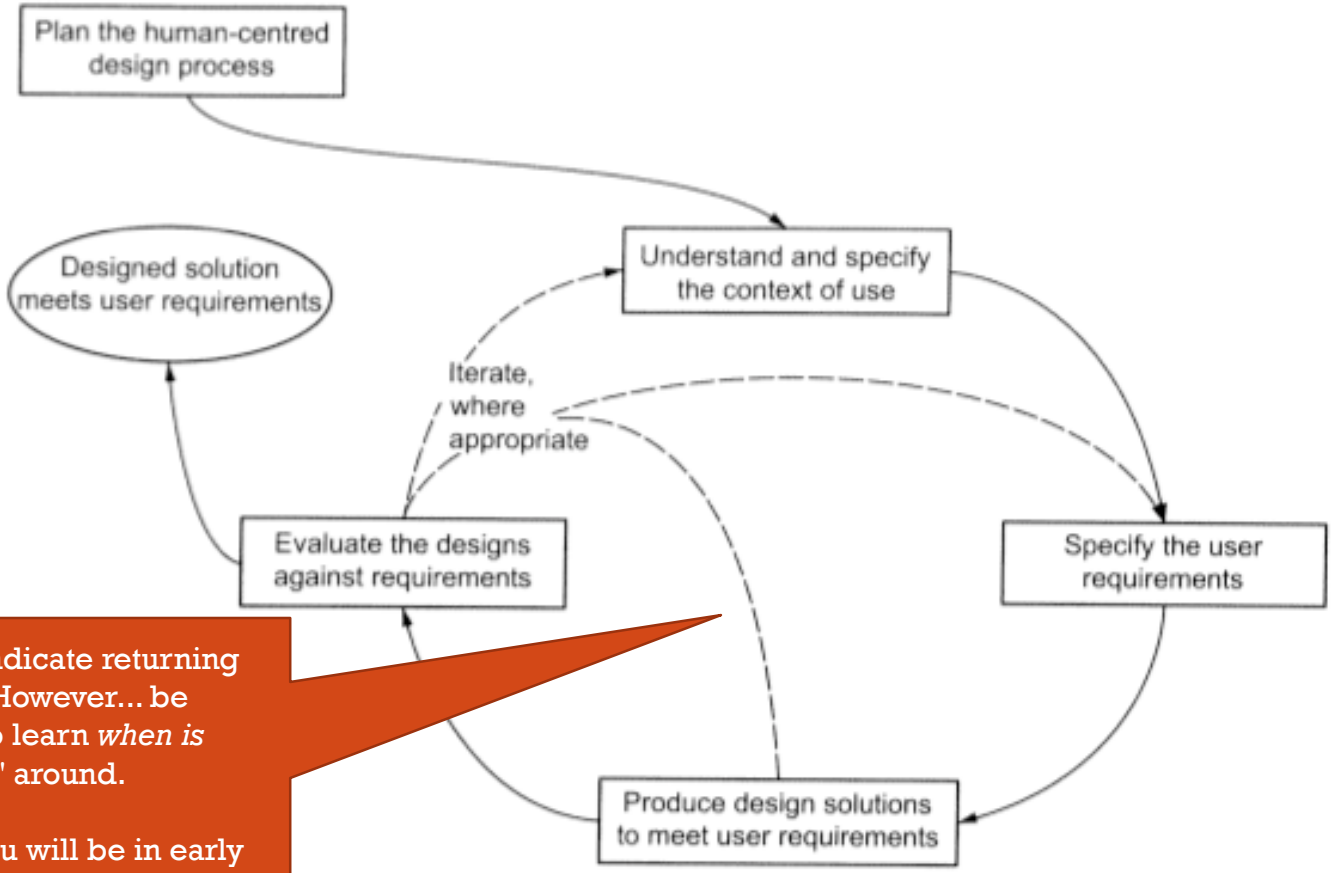
The top 3 will drive much of the next 2 weeks of class.

ISO 9241 DEFINITION OF USABILITY

- **Usability**: The extent to which a product can be used by specified users to achieve specified goals with *effectiveness*, *efficiency* and *satisfaction* in a specified *context of use*.
  - *Effectiveness*: the accuracy and completeness with which users achieve specified goals
  - *Efficiency*: the resources expended in relation to the accuracy and completeness with which users achieve goals
  - *Satisfaction*: freedom from discomfort, and positive attitude to the use of the product
  - *Context* of use: characteristics of the users, tasks and the organizational and physical environments
- ISO 9241 also defines a *Goal* as an intended outcome, and *Tasks* as activities to achieve a goal
  - Example (Jokela et al.): "90 % users achieve the goal (Es) in less than 1 minute (Ey) with an average satisfaction rating '6' (S) when users are novice ones (U), and they want to have a desired sum of cash withdrawn (G) with any bank machine (Cx)."

Es = effectiveness U = user
Ey = efficiency G = user goal
S = satisfaction Et = environment

ISO 13407 UCD PROCESS



The dashed lines indicate returning to a previous step. However... be careful! You need to learn *when is appropriate* to 'skip' around.

For your project, you will be in early iterations, where everything is still evolving, and will almost certainly need to execute all of the steps.

PLANNING IN UCD

1.
Plan

- UCD values information gleaned from collaboration:
  - What activities will you perform at steps 2-5?
  - When will you perform these activities?
  - Who will conduct the process?

Don't forget, we should also have some 'buy-in' from the team on the use of UCD.

- From the spec, the plan should identify:
  - a) the human-centred design process activities for steps 2-5;
  - b) procedures for integrating these activities with other system development activities, e.g. analysis, design, testing;
  - c) the individuals and the organization(s) responsible for the human-centred design activities and the range of skills and viewpoints they provide;
  - d) effective procedures for establishing feedback and communication on human-centred design activities as they affect other design activities, and methods for documenting these activities;
  - e) appropriate milestones for human-centred activities integrated into the overall design and development process;
  - f) suitable timescales to allow feedback, and possible design changes, to be incorporated into the project schedule.

UNDERSTAND THE CONTEXT OF USE

Context is very important – without it we could choose to do **anything**!

- The context of use description should
  - a) specify the range of intended users, tasks and environments in sufficient detail to support design activity;
  - b) be derived from suitable sources;
  - c) be confirmed by the users or if they are not available, by those representing their interests in the process;
  - d) be adequately documented;
  - e) be made available to the design team at appropriate times and in appropriate forms to support design activities.

- The specification suggests that the context of use document (wiki, knowledge base) should be a *working document* that is iterated upon as you gain more understanding.

DEFINE USER AND BUSINESS REQUIREMENTS

The specification's guidance on this step may not be useful.
User-centric requirements can be identified through a breakdown:

Users

- Who are they?
- What are their work environments like?
- How experienced are they with the technologies?
- What are their mental models and vocabulary?
- What are their personal characteristics?
- What are their cultural differences?
- What are their motivational differences?
- *Techniques we might use: Personas, Scenarios*

3. Specify User Requirements

Goals

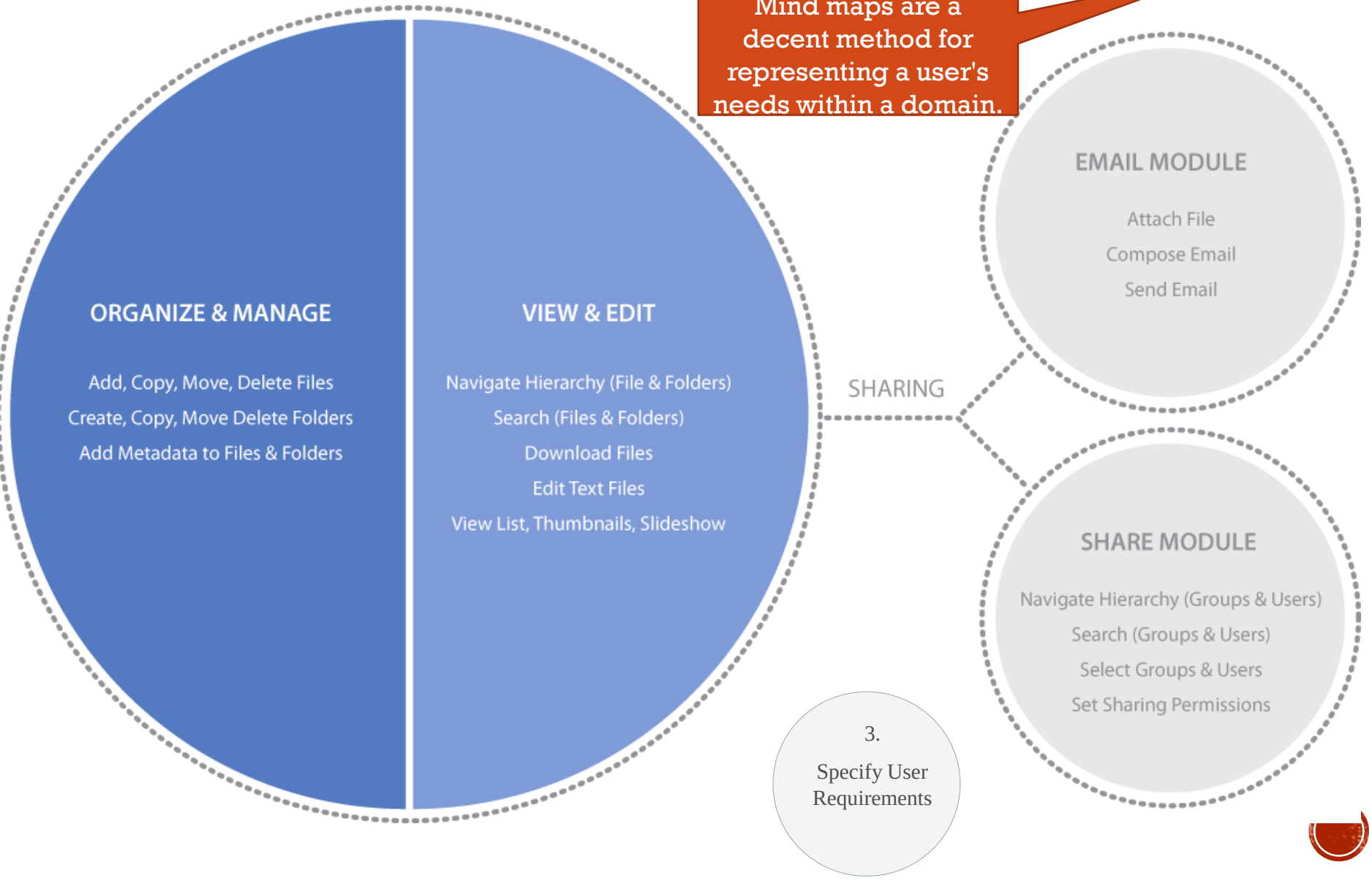
- The main thing users are trying to achieve
- *Techniques we might use: Interviews, Questionnaires, Reverse Engineer*

Tasks

- How users do things to achieve their Goals
- Business process analysis, Job analysis, Task sequences,
- *Techniques we might use: Observation, Task hierarchies, Research*

EXAMPLE: CONCEPTUAL TASK MODEL (MIND MAP)

Mind maps are a decent method for representing a user's needs within a domain.



3.

Specify User Requirements

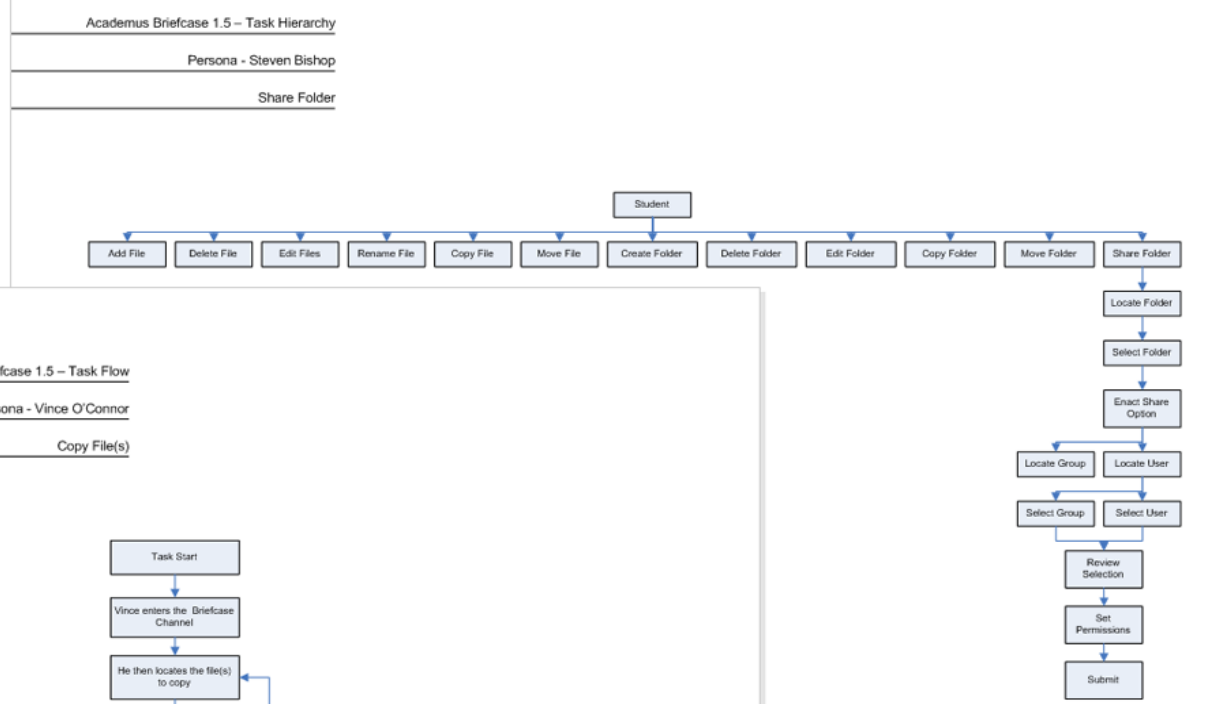
EXAMPLE : TASK ANALYSIS

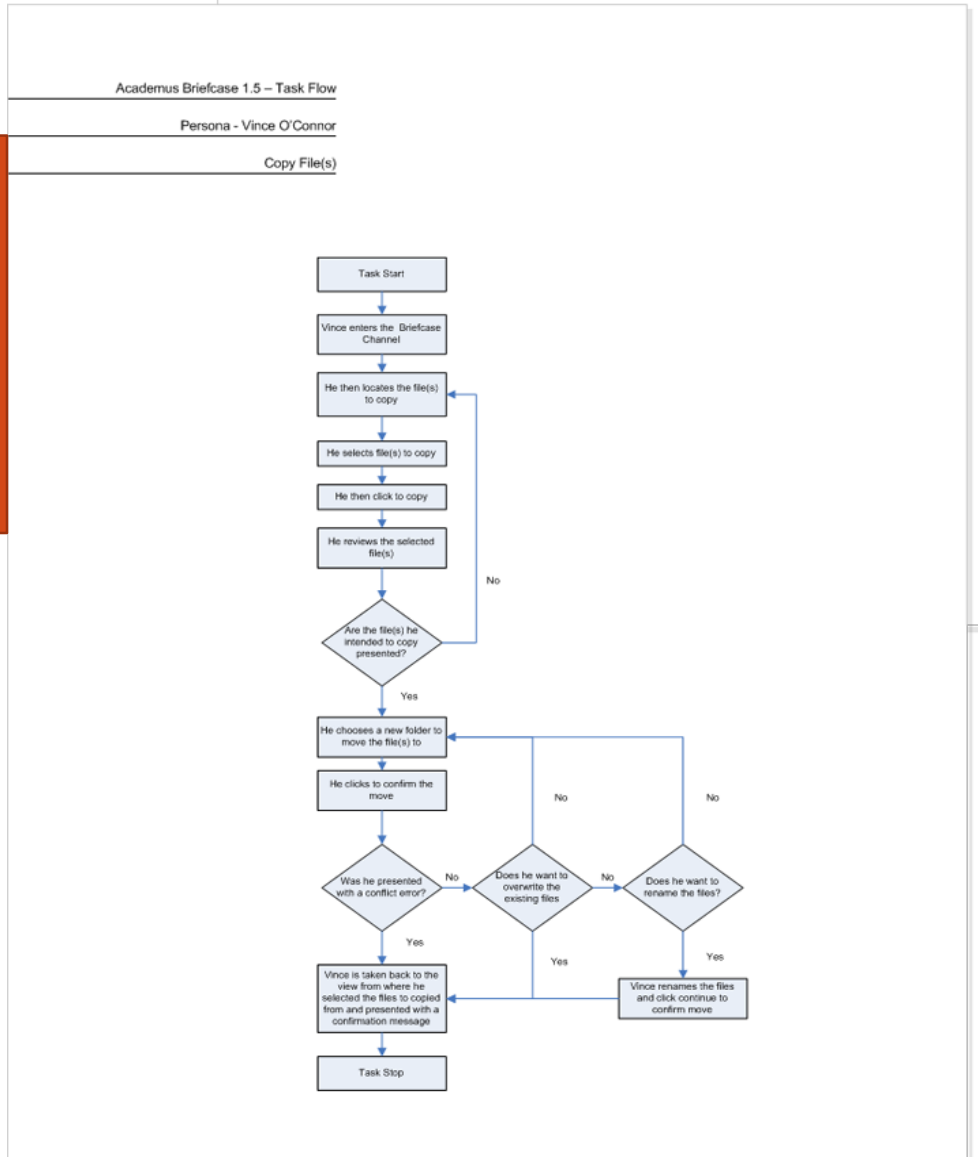
We can also breakdown a user's personal problem solving 'workflow' as a flowchart style diagram.

More on this later when we talk about Hierarchical Task Networks (HTNs).

3.

Specify User Requirements





PRODUCE DESIGN SOLUTIONS

- From the spec, this step involves the following activities:
  - a) *use existing knowledge* to develop design proposals with multi-disciplinary input;
  - b) make the design solutions *more concrete* using simulations, models, mock-ups, etc.;
  - c) present the design solutions to users and *allow them to perform tasks*
  - d) alter the design in *response to the user feedback* and iterate this process until the human-centred design goals are met;
  - e) manage the *iteration* of design solutions.

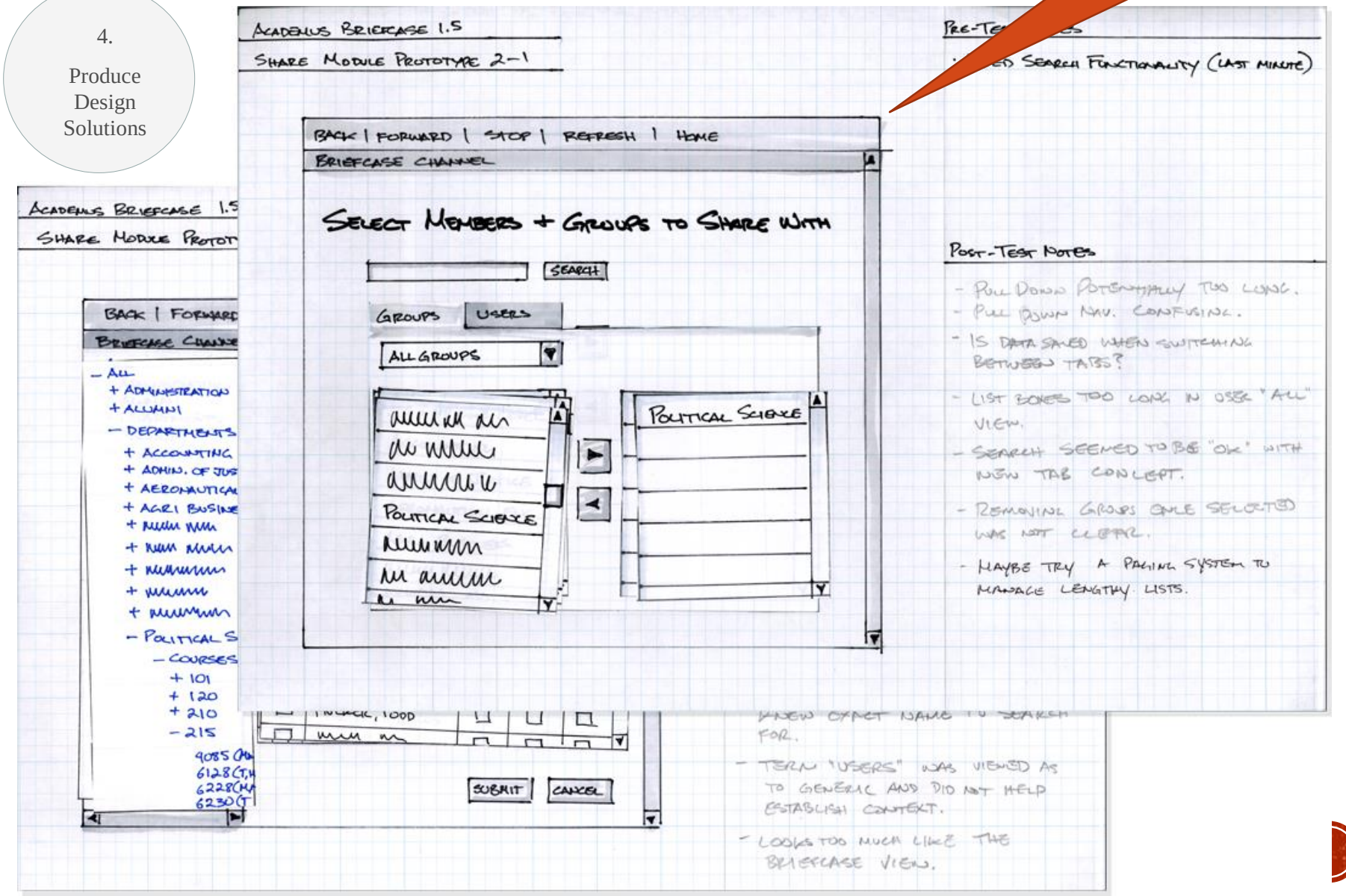
- *According to the spec, the benefits of prototyping (step b above):*
- a) make design decisions *more explicit* (this enables members of the design team to communicate with each other early in the process);
- b) allow designers to *explore several design concepts* before they settle on one;
- c) make it possible to *incorporate user feedback* into the design *early*
- d) make it possible to *evaluate* several iterations of a design & alternative designs
- e) *improve the quality and completeness* of the functional design specification.

TECHNIQUE: PAPER PROTOTYPES

Quickly sketching out a design on paper is the simplest way to build a prototype!

4.

Produce Design Solutions

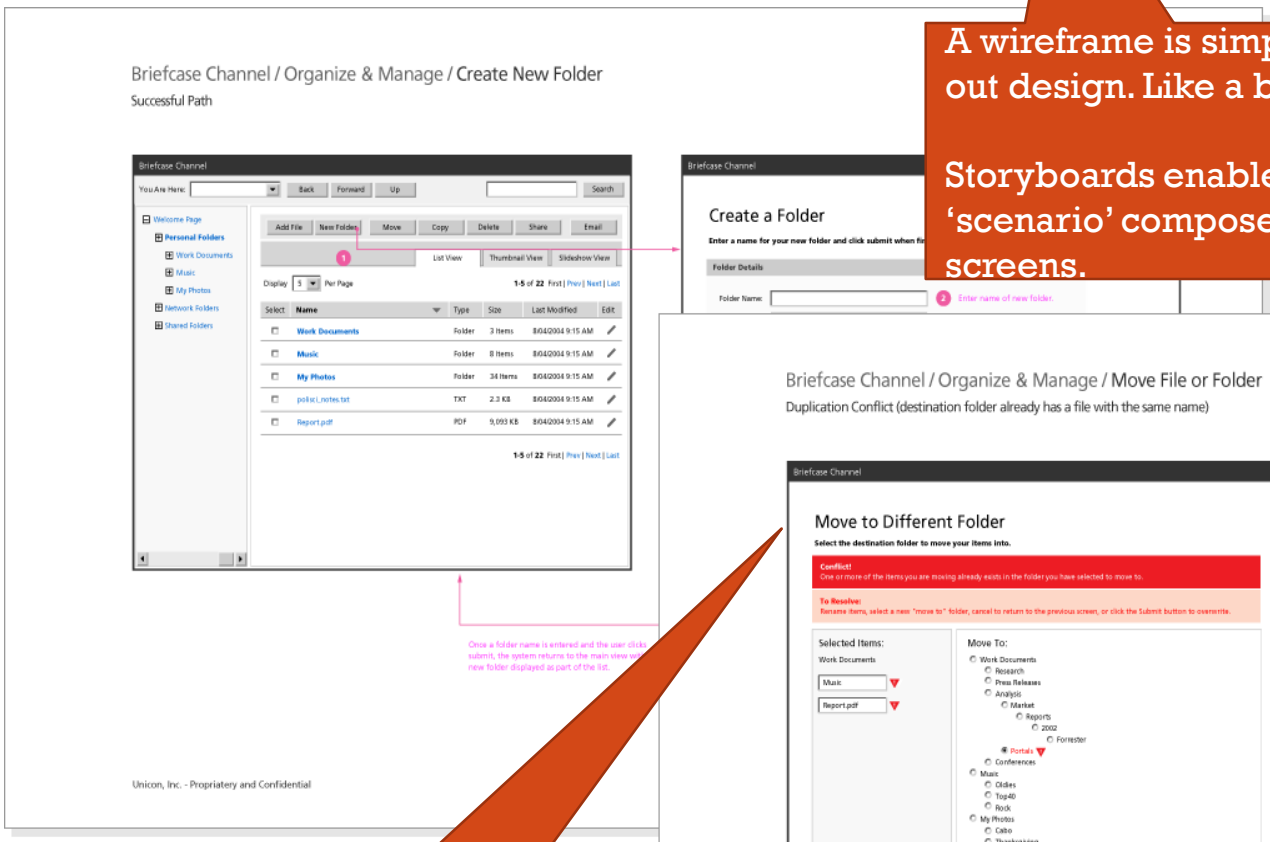


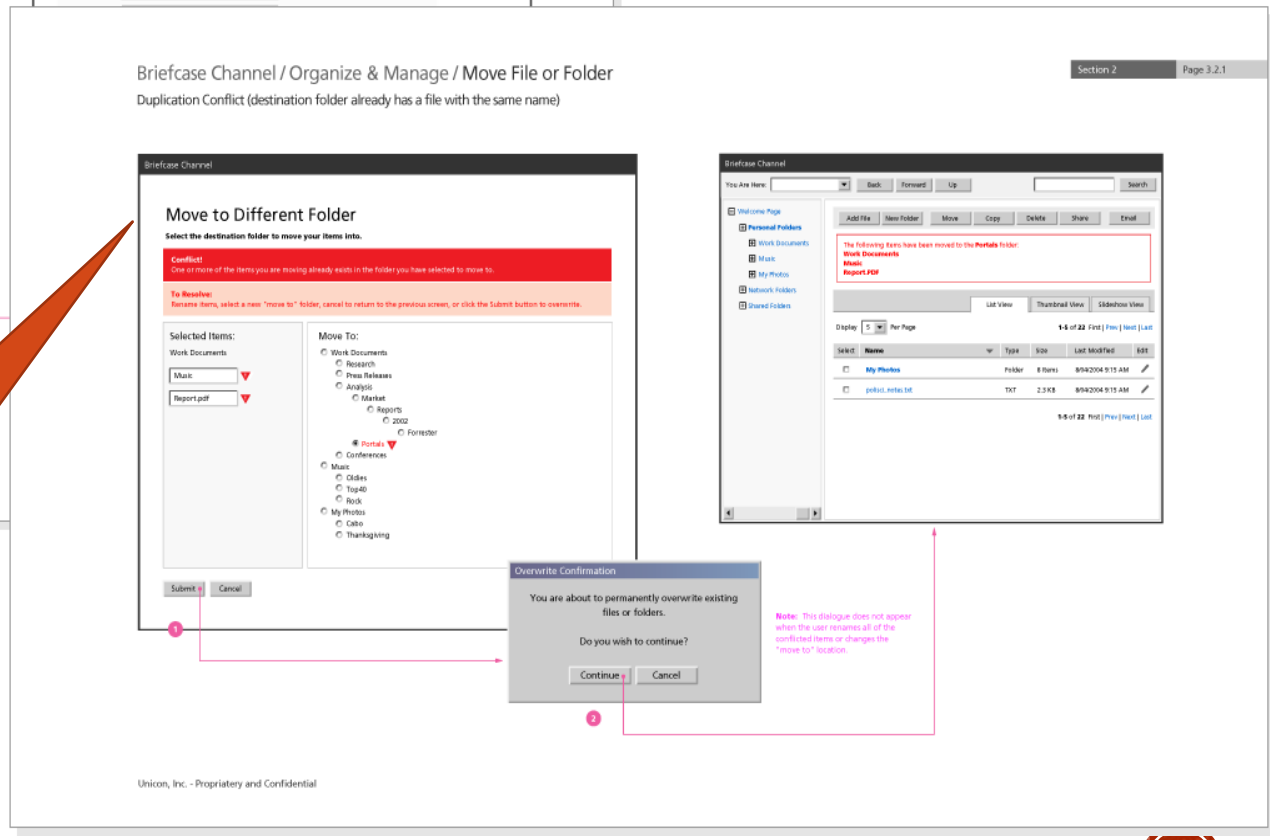
TECHNIQUE: WIREFRAMES / STORYBOARDS

A wireframe is simply a blocked out design. Like a blueprint.

Storyboards enable an actual 'scenario' composed of multiple screens.

4. Produce Design Solutions





Once we have a paper (i.e., 'low fidelity') prototype, we can start to refine it.

Tools like Pencil allow us to construct electronic prototypes with interactivity.

STORYBOARDING / PROTOTYPES

4.

Produce Design Solutions

- Modality:
  - 2-way
  - Participants are end users
  - Provides a structure for individual or group interaction
    - Storyboarding more conducive to small group interaction
- How-to:
  - Develop functionality based on vague requirements
    - Throw-away code!
  - Present to end user for direct feedback
  - Robustness of prototype needs only to be "sufficient to facilitate effective user feedback"
  - Technology base is chosen based on RAD, not based on the non-functional requirements

The code for your prototype is **not** intended to be used in the final product!

We will explore Storyboarding and Prototyping in detail in Interaction Design.

STORYBOARDING / PROTOTYPES

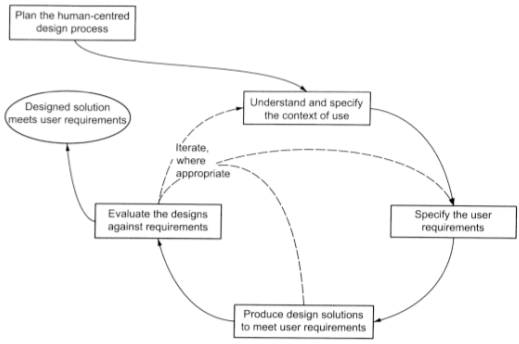
- Pros:
  - Making the solution "visible" provides you a precise means of agreeing on things with the user
  - May also facilitate your design and test cases
- Cons:
  - Cost to develop (need a RAD framework)
  - May pigeon-hole user into early requirements commitments
  - May pigeon-hole developers into early design commitments
  - Throw-away solution becomes a Big Ball Of Mud (BBOM)

DESIGN EVALUATION (OR, TEST PROTOTYPE)

- Evaluation (in any context), is only effective when there is a clearly defined goal for the evaluation
  - Consider two examples
    1. SER215/SER216: software testing classes last year
    2. How you are evaluated (graded)

5. Evaluate Designs

- But it is not just *summative evaluation* (go or no-go)
  - We use the process to find out more about our design ideas
  - From this perspective, we may elicit *formative* feedback
    - We aren't ready early on to know if we are "right"
    - We are still gathering information
    - So we may demonstrate several design ideas in one cycle
    - As we prune out bad ideas, we try to then "get it right"



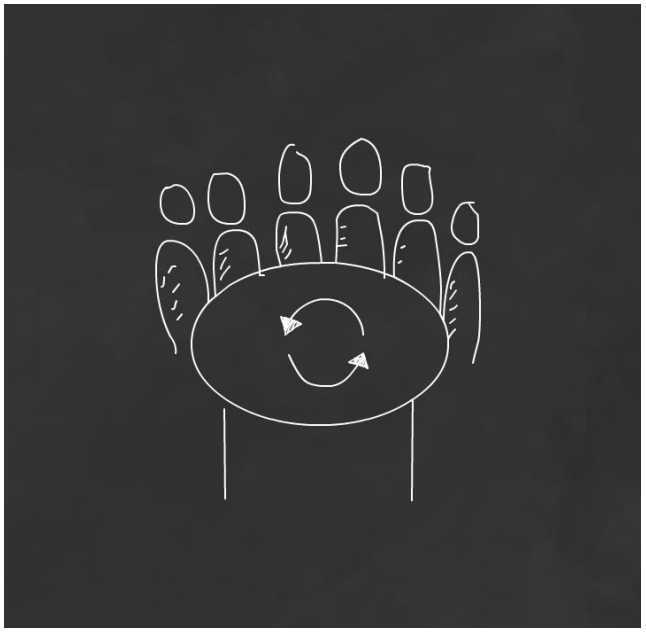
EXAMPLE METHOD: JOINT APPLICATION DESIGN

5. Evaluate Designs

- Modality
  - Multi-way: customers, users, designers, and experts
  - A cross between group meetings and prototypes
  - Similar to evolutionary style of concurrent development, except the stakeholders are part of the development team
- How-to:
  - Carefully assemble a team
  - **Ensure roles are blurred – everyone is a peer and everyone's opinions are important. Design is not just for the designers**
  - JAD sessions require a clear statement of the purpose of the session and its goals.
  - JAD sessions are usually run by a facilitator who keeps the participants focused.
    - Can have observers, but observers must remain silent according to the rules of JAD.

EXAMPLE METHOD: JOINT APPLICATION DESIGN

- Pros:
  - Workshop-type feel facilitates participation
  - All stakeholders feel ownership and teamwork
  - All concerns laid out on the table
- Cons:
  - Requires a skilled facilitator
  - Social issues – some individuals may dominate
  - Consensus building can be difficult in a large group
  - You effectively combine steps – design and validation – so you may still need an external review



SUMMARY

- User-Centered Design (UCD)
  - Plan, Understand Context, Define Requirements, Prototype, Validate – **and Iterate**
- Prototyping –
  - Think many; at a level of fidelity appropriate to project at that time
  - Iterative refinement
- Storyboarding – use it to understand flow
  - Can help with measures of *effectiveness* and *efficiency*
- Observation and JAD are example techniques to more directly work with end users
  - We will identify and explore more methods in lab on Thursday
- ISO 13407 and ISO 9241-XXX are standards in Usability
  - 13407 (1999) defined the process but was superseded by 9241-210 (2008)
  - 9241-210 provides more detailed guidance on practices