Task-Centered System Design

Task-Centered System Design

Q: What are the three goals of the Task-Centered Design approach (the approach is systematically used for ..., ..., and ...)?
A:
• technique to analyze the user's tasks to inform the design of the user interface;
• systematically determine if an interface matches the needs of its end users;
• systematically discover usability issues.

For the requirement analysis, we should use the **end-user perspective**, which study that "exactly who would use the system to do exactly what?", and it means software (or application) should make user convenient. (Oppositely, **software perspective** study "what functions should the system have?", it implies that users should fit system).

Q: List the four phases of the task-centered design process (bulleted list of points).
A:

1. Identification (Identify Users + Tasks)
   Indentify **specific users** and articulate their **concrete tasks**.
2. Requirements
   Decide which of these tasks and users the design will support.
3. Design
   Base design representations & dialog sequences of these tasks.
4. Walk-Through Evaluations
   Using your design, walk through these tasks to test the interface.

Phase 1: Identify Users + Tasks

• get in touch with real people who will be potential users of the new system
– prototypical categories
– also consider extremes

• learn about their real tasks

- articulate specific, detailed examples of tasks they perform or want to perform that the system should support
  • routine tasks
  • infrequent but important tasks
  • infrequent and incidental tasks

• how to identify tasks?
– immersing oneself into a real person's environment
– observing people in their actual work context
– interviewing people as they do their work
– shadowing a person over the course of his or her day
– serving people's requests

Phase 2: Requirements

Decide which tasks & users the design will support:

• which user types will be addressed by the interface?
– most designs will not be able to handle everybody
– specify why particular users are included/excluded

• which tasks will be addressed by the interface?
– most designs will not be able to handle all tasks
– list requirements in terms of how they address tasks: absolutely must include/should include/could include/exclude
– specify why tasks are in these categories

Phase 3: Design as Scenarios

• develop designs to fit users and specific tasks
– ground interfaces in reality

• use task descriptions to
– get specific about possible designs
– consider real-world contexts of real users
– consider how design features work together: what would a user do and/or see for each step when performing this task

Phase 4: Walk-Through Evaluation

• debug the newly developed interface design]

• process:

1. select one of the task scenarios
2. for each user's step/action in the task
   a) can you build a believable story that motivates the user's action?
   b) can you rely on the user's expected knowledge and training about the system to be able to perform the task?
   c) if you cannot:
   – you have located a problem in the interface!
   – note the problem, including any comments
   – assume it has been fixed
   d) go to the next step in the task