Task-Centered System Design

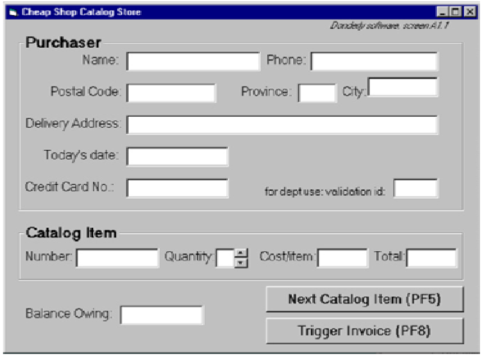
How to develop task examples

How to evaluate designs through a task-centered walk-through

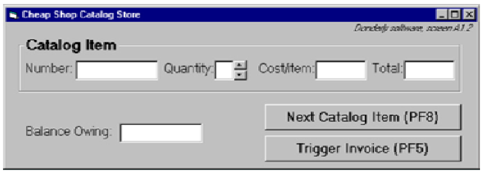
Exercise: The Cheap Shop interface

Cheap Shop

Screen 1



Screen 2



Requirements Analysis: Focusing On The Software

- Designing for a faceless user: A pretend person that will magically change his or her abilities to adapt to your system (elastic user)

Requirements Analysis: Focusing On The Person

- Determining *who* will be doing exactly *what* with your system

- Designing for Mary Hart: A real person with real constraints that is trying to get her job done (inelastic user)

The Task-Centered Process

Phase I: Identification

- Identify specific users (*who*)
- Articulate realistic example tasks (*what*)

Phase II: Requirements

- Narrow down the scope of the project: Decide which *user group's tasks* will be supported by the design in order to determine the requirements of the system

Phase III: Design

- Base design representations and dialog sequences on the supported tasks

Phase IV: Walkthrough Evaluations

- Convert your tasks to scenarios
- Use the scenarios to walk through the designs in order to test the proposed interface

Prior To Starting The Task-Centered Process (For Assignment 1)

Do a write up for "Phase 0".

- It is not part the traditional Task-Centered approach but is needed for your TA:
- Background environment

- What the system will be used for

- System constraints

Phase I: Identification

1. Get in touch with real people who will be potential users of your system

- Identify a range of actual end users

Example users: Accountants

Prototypical examples

Extreme examples

Phase I: Identification

Spend time with them discussing how the system might fit in

- Determine who would be willing to talk to you about this?
- If you can’t get them interested, who will actually buy/use your system?

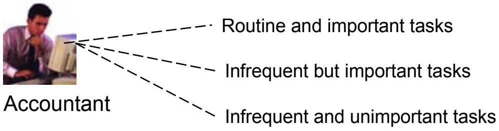
Learn about the user’s tasks

- Articulate concrete, detailed examples of tasks they currently complete or those that they want to complete (ones that they want to do but can’t do with the existing system)
- Categorize the tasks
  - Routine and important
  - Infrequent but important
  - Infrequent and unimportant

Phase I: Identification

Identify a broad coverage of users and types of tasks

- The typical ‘expected’ users

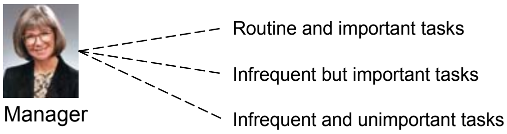

Routine and important tasks

Infrequent but important tasks

Infrequent and unimportant tasks

Accountant

- The occasional but important users

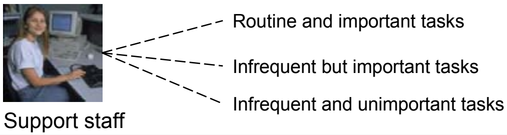

Routine and important tasks

Infrequent but important tasks

Infrequent and unimportant tasks

Manager

- The unusual users


Routine and important tasks

Infrequent but important tasks

Infrequent and unimportant tasks

Support staff

Phase I: Identification

Ways of getting information about users and their tasks

- Direct contact (ideal)

- Interview an intermediary (reasonable alternative)

OR

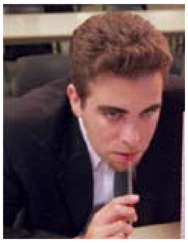
If all else fails..

- Describe your expected set of users and expected set of tasks
- These will become your “assumed users and tasks”
- **Be sure** that you verify this information and modify your assumptions accordingly

Phase I: Identification

2. Use the information about the users and their tasks to produce several task examples

Task Examples: Are stories that describe the actual usage of the system as well as providing a detailed description of the person who is using that system.



Fred Smith enters the first journal entry for January 1 which is a withdraw of $12.99 from petty cash..

Phase I: Identification

Characteristics of good a task

a) Says *what* the user wants to do but *not how* they would do it
- No assumptions are made about the interface (tasks are used to form the basis for designs so potential designs should not be constrained at this point)

b) Are very *specific*
- Says exactly what the user wants to do (e.g., specifies actual items the user would eventually want to input)

c) Describes a *complete* job
- Forces the designer to consider how interface features work together
- Contrasts how information input/output flows through the dialog

Do not:
- *Just create a simple list of things that the system should do*
- *Present a goal independent of other goals*

Phase I: Identification

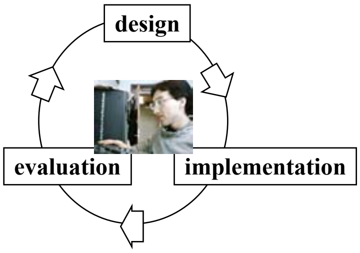
d) Says *who* the users are

- Describe what they know
- Name names, if possible
- Reflects the real interests of real users
- Find tasks that illustrate functionality in a person’s real work context

Phase I: Identification

3. Tasks are evaluated

- Circulate descriptions to users, and rewrite if needed.
- Ask users for:
  - omissions,
  - corrections,
  - clarifications,
  - suggestions.



Phase II: Requirements

Which user groups will be addressed by the interface?

- Designs can rarely handle everyone!

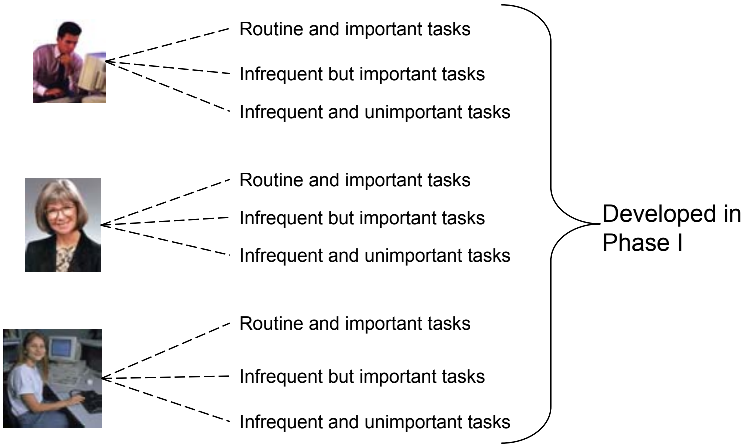
Coffee Cup Software

Dilbert © United Features Syndicate

Phase II: Requirements

Which user groups will be addressed by the interface?

- Designs can rarely handle everyone!



Routine and important tasks

Infrequent but important tasks

Infrequent and unimportant tasks

Routine and important tasks

Infrequent but important tasks

Infrequent and unimportant tasks

Routine and important tasks

Infrequent but important tasks

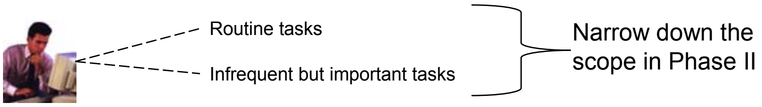
Infrequent and unimportant tasks

Developed in Phase I

Phase II: Requirements

Which user groups will be addressed by the interface?

- Designs can rarely handle everyone!
- Indicate why are particular users/tasks included or excluded?



Routine tasks

Infrequent but important tasks

Narrow down the scope in Phase II

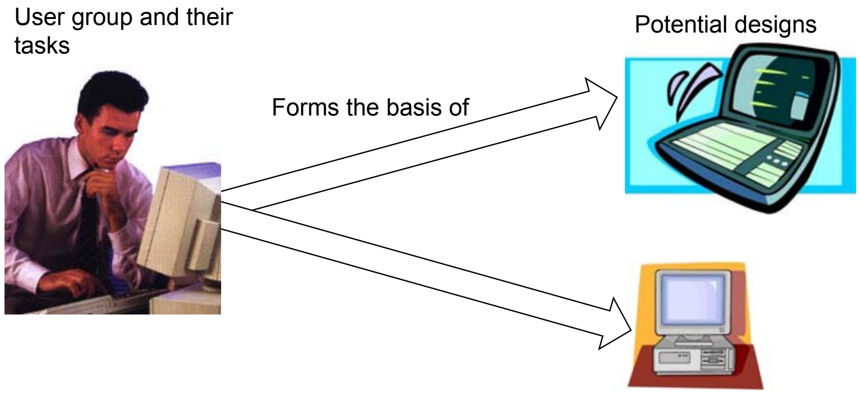
Phase II: Requirements

Which tasks will be addressed by the interface?

- Requirements are listed in terms of how they address tasks
  - Absolutely must include:
    …
  - Should include:
    …
  - Could include:
    …
  - Exclude:
    ...
- Discussion includes why each requirement belongs in a particular category

Phase III: Produce Designs Based On The Tasks

- **Develop prototype interfaces around the user group and their tasks**



User group and their tasks

Forms the basis of

Potential designs

Phase IV: Walk-Through Evaluation

Convert your tasks to scenarios

Tasks

- Design independent
- Allows different ideas to be tried out (written up in Phase I)

"*Millie Varunda is price-comparing the costs of a child's bedroom set, consisting of a wooden desk, a chair, a single bed, a mattress, a bedspread, and a pillow all made by Furnons Inc. She takes the description and total cost away with her to check against other stores.*"

Scenarios

- Design specific
- Used to evaluate the effectiveness of a particular design (converted from the tasks as the evaluation is performed in this phase).
- "*..Bob takes Millie's list and walks over to the computer. He selects the first screen and selects the number field under the catalog item heading. Bob then types in 323000111 into the input field and adjusts the spinner control to increase the number of items from default of 1 to the desired value of 3…*"

Phase IV: Walk-Through Evaluation

Scenarios are good for debugging an interface

- Usability debugging

Algorithm for a walk-through evaluation:

1. Select one of the scenarios

2. For each user's step/action in the scenario:

a) Can you build a believable story that **motivates** the user's actions?

b) Can you rely on the user's expected **knowledge and training** about system?

c) If the answer is 'no' then you've located a problem!

- Once a problem is identified, either jot down a quick solution or assume that it has been repaired and go to the next step in the scenario (complete the walk through or walk throughs prior to redesigning the system).

Example: The Cheap Shop Catalog Store

- In Cheap Shop, people shop by browsing the paper catalogs scattered around the store.

- When people see an item they want, they enter the item code and price from the catalog onto a form.

Item code	Amount
323066 697	1

Example: The Cheap Shop Catalog Store

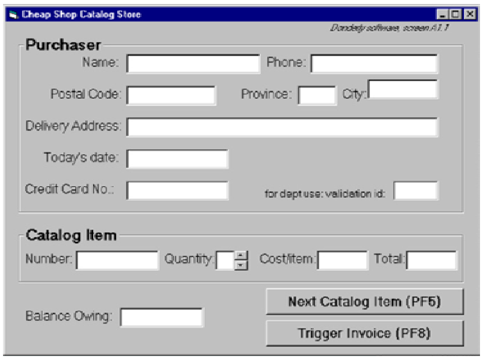
- People give this form to a clerk, who brings the item(s) from the back room to the front counter.

Example: The Cheap Shop Catalog Store

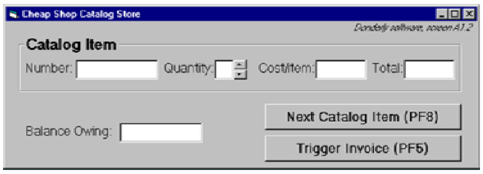
- People then pay for the items they want.

Cheap Shop

Screen 1



Screen 2



Specifications

- **To create an order**
  - On screen 1, shoppers enter their personal information and their first order.
  - Text is entered via the keyboard.
  - The tab or mouse is used to go between fields.
- **Further orders**
  - Shoppers go to the 2nd screen by pressing the 'Next Catalog Item' button.
- **Order completion**
  - Shoppers select 'Trigger Invoice'.
  - The system automatically tells shipping and billing about the order.
  - The system returns to a blank screen #1.
- **To cancel order**
  - Shoppers do not enter input for 30 seconds (as if they walk away).
  - The system will then clear all screens and return to the main screen.
- **Input checking**
  - All input fields are checked when either button is pressed.
  - Erroneous fields will blink for 3 seconds, and will then be cleared.
  - The shopper can then re-enter the correct values in those fields.

Developing Task Examples: Cheap Shop

Task example 1

- Fred Johnson, who is caring for his demanding toddler son, wants a good quality umbrella stroller (red is preferred, but blue is acceptable).
- He browses the catalog and chooses the "Roll 'em out" brand stroller (cost $99.95 item code 323 066 697).
- He pays for it in cash, and uses it immediately.
- Fred is a first-time customer to this store, has little computer experience, and he types very slowly with one finger. He lives nearby at #1109 Dalhousie Dr. N.W.

Roll 'em out stroller. This well made but affordable Canadian stroller fits children between 1-3 years old. Its wheels roll well in light snow and mud.
…**$99.95**

Red:	**323 066 697**
Blue:	**323 066 698**

Developing Task Examples: Cheap Shop

Discussion

- Fred has many properties of our typical expected user:
  - Many customers are first time shoppers.
  - A good number have no computer experience.
  - A good number are poor typists.
- The task type is routine and important.
  - Many people often purchase only one item.
  - A good number of those pay by cash.
  - As with Fred, people often have a general sense of what they want to buy, but decide on the actual product only after seeing what is available.

Developing Task Examples: Cheap Shop

Task example 2

- Millie Varunda is price-comparing the cost of a child's bedroom set:
  - A wooden desk, a chair, a single bed, a mattress, a bedspread, and a pillow all made by Furnons Inc.
- She takes the description and total cost away with her to check against other stores.
- Three hours later, she returns and decides to buy everything but the chair.
- She pays by credit card.
- She asks for the items to be delivered to her daughter's home at 47 Sun Valley Drive S.W., in the basement suite at the back of the house.
- Millie is elderly and arthritic.

Developing Task Examples: Cheap Shop

Discussion

- Like Millie,
  - A reasonable number of store customers are elderly, with infirmities that inhibit their physical abilities.
  - A modest number of them also enjoy comparison shopping, perhaps because they have more time on their hands or because they are on a fixed income.
- The task type is less frequent, but still important.
  - Although this would be considered a 'major' purchase in terms of the total cost, the number of items purchased is not unusual.
  - Delivery of large items is the norm.
  - Most customers pay by credit card for larger orders.

Developing Task Examples: Cheap Shop

Task example 3

- Jim Tam, Ace Salesguy ™, is the sole salesperson in the store and has been given a list of 10 items by a customer who does not want to use the computer.
- The items are:
  - 4 pine chairs, 1 pine table, 6 blue place mats, 6 “Lor” forks, 6 “Lor” table spoons, 6 “Lor” teaspoons, 6 “Lor” knives, 1 “Tot” tricycle, 1 red ball, 1 “Silva” croquet set
- After seeing the total, the customer tells Jim he will take all but the silverware.
- The customer then decides to add 1 blue ball to the list.
- The customer starts paying by credit card, but then decides to pay cash and tells Jim he wants the items delivered to his home the day after tomorrow. While this is occurring, 6 other customers are waiting in line.
- Jim is a new employee and this is the first time that he has worked the front counter alone.

Developing Task Examples: Cheap Shop

Discussion

- This task introduces the clerk as a system user.
  - Because the store has a high turnover in its staff, new employees such as Jim are also common.
  - Thus Jim reflects a ‘rare’ but important group of users.

- The task type is less frequent, but still important
  - The task, while complex, is fairly typical i.e., people making large numbers of purchases often ask the clerk to help them.
  - Similarly, clerks mention that customers often change their mind partway through a transaction i.e., by changing what they want to buy and/or by changing how they want to pay for it.
  - Customers, however, rarely give specific delivery dates, with most wanting delivery as soon as possible.
  - Lineups for clerks are common during busy times.

Walkthrough Template For Each Scenario

Scenario number: ____

No.	Description of the step	Does the user have the knowledge/training to do this?	Is it believable that they are they motivated to this?	Comment / solution

Recap: How Tasks Can Be Used

As a design tool

- Tasks should not only guide to the design of the interface but also they should be used as a criteria for ranking the functionality of the system.

As an evaluation mechanism

- A task walk-through is a fast and effective way of finding major usability problems

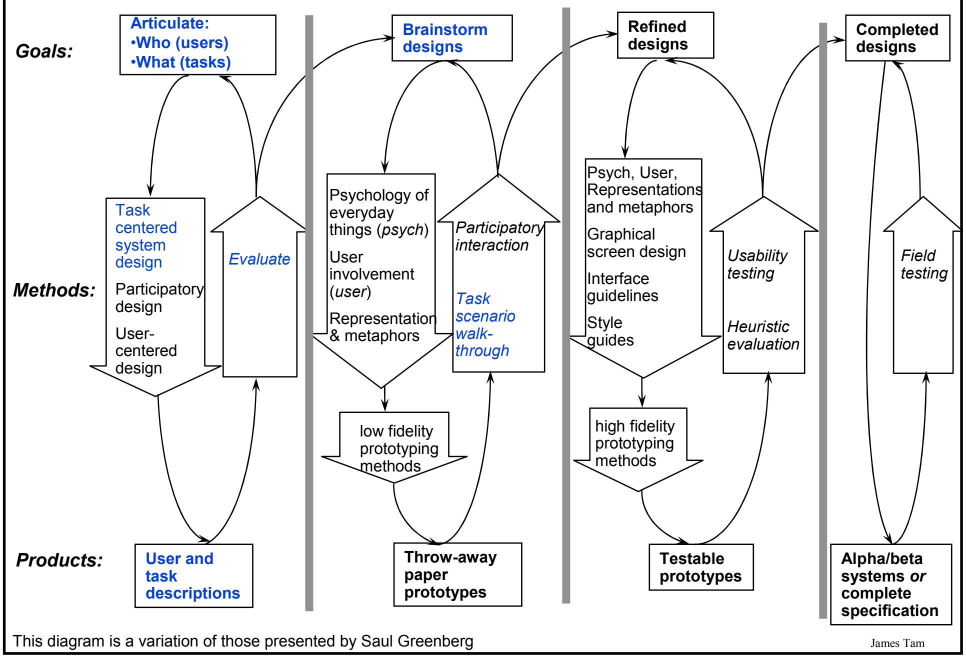
You Now Know

How to develop concrete task examples

How to use task examples to motivate your designs

How to evaluate designs through task-centered walkthroughs

Interface Design And Usability Engineering



Goals:

Articulate:
•Who (users)
•What (tasks)

Brainstorm designs

Refined designs

Completed designs

Methods:

Task centered system design

Participatory design

User-centered design

Evaluate

Psychology of everyday things (*psych*)

User involvement (*user*)

Representation & metaphors

Participatory interaction

Task scenario walk-through

Psych, User, Representations and metaphors

Graphical screen design

Interface guidelines

Style guides

Usability testing

Heuristic evaluation

Field testing

low fidelity prototyping methods

high fidelity prototyping methods

Products:

User and task descriptions

Throw-away paper prototypes

Testable prototypes

Alpha/beta systems *or* complete specification

This diagram is a variation of those presented by Saul Greenberg