Task-Centered System Design

How to develop task examples

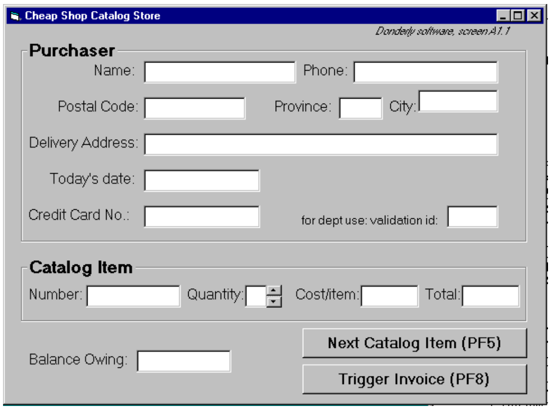
How to evaluate designs through a task-centered walk-through

Exercise: The Cheap Shop interface

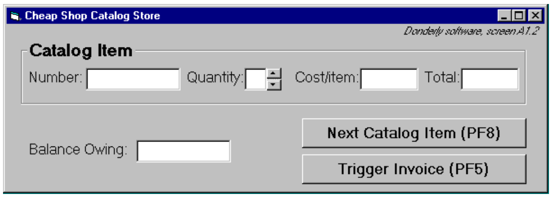
An alternative the task-centered approach

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)

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 who is trying to get her job done (inelastic)**

The Task-Centered Process

Phase I: Identification

- Identify specific users
- Articulate realistic example tasks

Phase II: Requirements

- Decide which of these tasks and users the design will support in order to determine the requirements of the system

Phase III: Design

- Base design representation and dialog sequences on the tasks

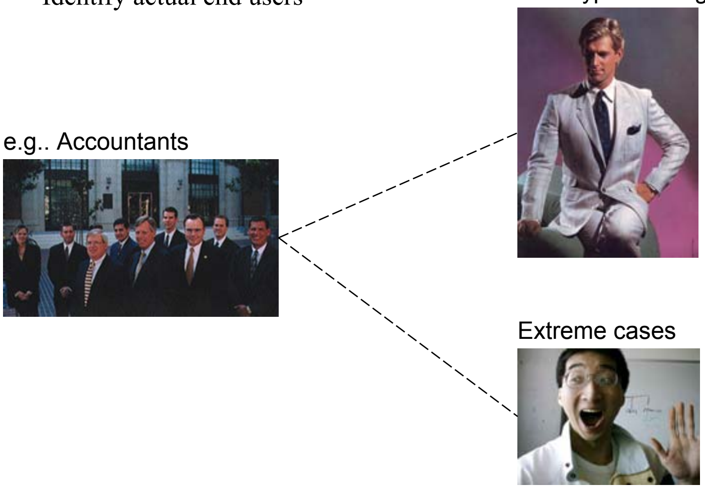
Phase IV: Walkthrough Evaluations

- Using your design, walk through your scenarios to test the proposed interface

Phase I: Identification

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

- Identify actual end users



e.g.. Accountants

Prototypical categories

Extreme cases

Phase I: Identification

Spend time with them discussing how the system might fit in

- Who would be willing to talk to you about this?
- If you can't get them interested, who will actually buy/use your system?
- If there are no real users or tasks…think again, there probably are!

Learn about the user's tasks

- Articulate concrete, detailed examples of tasks they perform or want to perform (ones that they currently can't do but want to do with your system)
  - Routine
  - Infrequent but important
  - Infrequent and unimportant

Phase I: Identification

Ways of getting information about users and their tasks

- Direct contact (ideal)
- Interview an intermediary (reasonable alternative)

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.

Phase 1: Identification

Characteristics of good a task

a) Says what the user wants to do but not how they would do it
  - No assumptions made about the interface
  - Can be used to compare different design alternatives in a fair way

b) Are very specific
  - Says exactly what the user wants to do
  - Specifies actual items the user would eventually want to input (in some form)

c) Describes a complete job
  - Forces 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

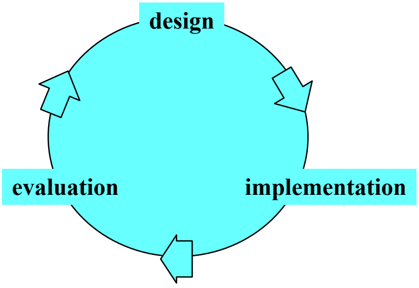
- Describe what they know
- Name names, if possible
- Reflects the real interests of real users
- Find tasks that illustrate functionality in a person's real work context

Phase I: Identification

3. Tasks are evaluated

- Circulate descriptions to users, and rewrite if needed
  - Ask users for:

    omissions
    corrections
    clarifications
    suggestions



Phase I: Identification

4. Identify a broad coverage of users and task types

- The typical ‘expected’ user, Typical routine tasks
- The occasional but important user, Infrequent but important tasks
- The unusual user Infrequent but unimportant tasks

Accountant

Manager

Support staff

Phase II: Requirements

Which user types will be addressed by the interface?

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

Which tasks will be addressed by the interface?

- Designs can rarely handle all tasks
- Requirements 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: Design As Scenarios

Develop prototype interfaces around the user group and their tasks

Convert the tasks to scenarios

Use the scenarios to

- Get specific about possible designs
- Consider how design features work together to help a person accomplish real work
- Consider the real world contexts of real users
- Consider how design features work together
  - What the user would see / do on a step-by-step basis when performing the task

Phase IV: Walk-Through Evaluation

Scenarios are good for developing an interface

- Usability debugging

Algorithm:

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 user's expected **knowledge and training** about system?

c) If you cannot rely on the above then you've located a problem!

- Once a problem is identified, either jot down a quick solution or assume that it has been repaired

d) Go to the next step in the scenario

Example: The Cheap Shop Catalog Store

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

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

Item code	Amount
323066 697	1

Example: The Cheap Shop Catalog Store

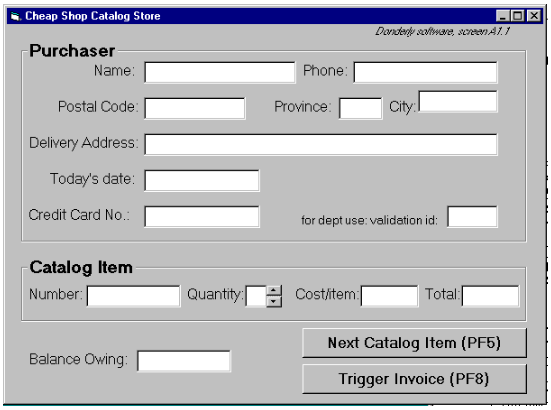
- People give this form to a clerk, who brings the item(s) from the back room to the front counter.

Example: The Cheap Shop Catalog Store

- People then pay for the items they want.

Cheap Shop

Screen 1



Cheap Shop Catalog Store

Donderly software, screen A1.1

Purchaser

Name: Phone:

Postal Code: Province: City:

Delivery Address:

Today's date:

Credit Card No.: for dept use: validation id:

Catalog Item

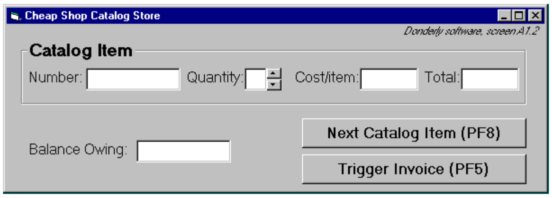
Number: Quantity: Cost/item: Total:

Balance Owing:

Next Catalog Item (PF5)

Trigger Invoice (PF8)

Screen 2



Cheap Shop Catalog Store

Donderly software, screen A1.2

Catalog Item

Number: Quantity: Cost/item: Total:

Balance Owing:

Next Catalog Item (PF8)

Trigger Invoice (PF5)

Specifications

- **To create an order**
  - On screen 1, shoppers enter their personal information and their first order
  - Text is entered via 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 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 says he types very slowly with one finger. He lives nearby on Deer Bottom Avenue 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 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.
- 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 31247 Lucinda 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 low 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 ™, the sole salesperson in the store, is 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. The customer tells Jim he wants the items delivered to his home the day after tomorrow. While this is occurring, 6 other customers are waiting for Jim.
- 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

***Task number:* ____**

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

Goal-Centered System Design

Goal

- Desired end condition
- Tend to be stable over time

Task

- The intermediary process that you go through to achieve your goal.
- May change as technology and work patterns change over time.

Goal-Centered System Design

Develop a Persona

- A precise and specific description of the user and what the person wishes to accomplish (goals)
- A pretend user developed from investigating the problem domain (based on actual users)

An alternative to the Task-centered approach

See Allan Cooper 'The inmates are running the asylum'

Goal-Centered System Design

Develop a cast of characters

- A set 3 – 12 personas (1 will be the primary persona)

Avoid elastic personas (be as specific and detailed as possible!)

Task-Based Vs. Goal-Based Approach

Task-based

- Can ask users for more info

Goal-based

- Avoids outlier cases

Both

- Based on real users
- Provide a focus for the design (resolve design conflicts)

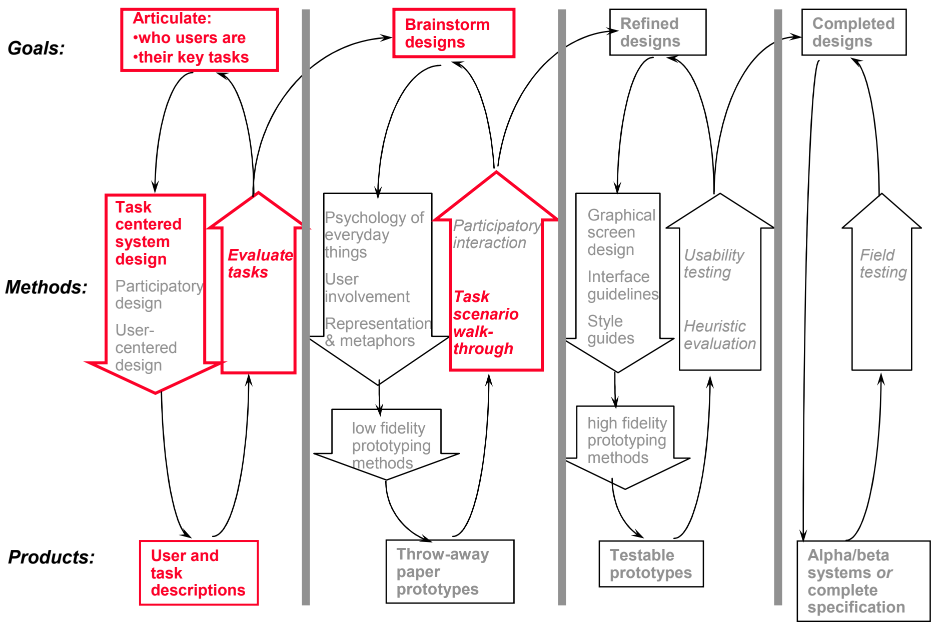
You Now Know

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 are
•their key tasks

Brainstorm designs

Refined designs

Completed designs

Methods:

Task centered system design

Participatory design

User-centered design

Evaluate tasks

Psychology of everyday things

User involvement

Representation & metaphors

Participatory interaction

Task scenario walk-through

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