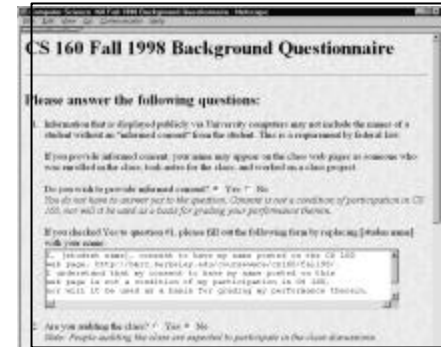


Design Discovery and Task Analysis

CS 160 Spring 2001

Lawrence A. Rowe
Computer Science Division - EECS
University of California at Berkeley
<http://bmr.c.berkeley.edu/~larry>

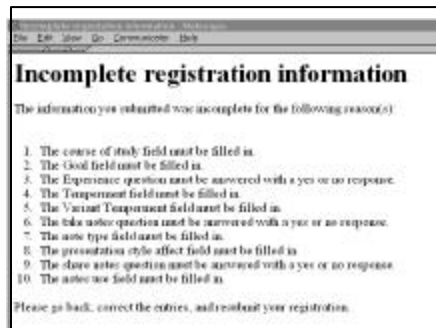
Interface Hall of Shame or Fame?



CS 160:1/26/01

2

Hall of Shame or Fame?



CS 160:1/26/01

3

Interface Hall of Shame



What is problem?

CS 160:1/26/01

4

Interface Hall of Shame



- What is problem?
- Requires "recall over recognition" rather than "recognition over recall"

CS 160:1/26/01

5

"One most unfortunate product is the type of engineer who does not realize that in order to apply the fruits of science for the benefit of mankind, he must not only grasp the principles of science, but must also know the needs and aspirations, the possibilities and the frailties, of those whom he would serve."

-- Vannevar Bush

Outline

- What is usability?
- Design models and processes
 - Waterfall –vs- user-centered iterative design
- Design process
- Task Analysis
 - Task –vs- scenario

CS 160:1/26/01

7

Usability

- Ease of learning
 - faster the second time and so on...
- Recall
 - remember how from one session to the next
- Productivity
 - perform tasks quickly and efficiently
- Minimal error rates
 - if they occur, good feedback so user can recover
- High user satisfaction
 - confident of success

CS 160:1/26/01

8

Design Processes

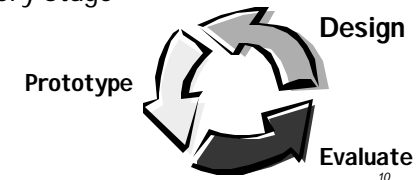
- Waterfall model
sequence of steps from software engineering
customer is **not** the user
- Spiral model
only continue next step if feedback is positive
- Prototyping
design by constructing executable models
should the prototype become the system?
- User-centered iterative design
include the user in the design process and iterate

CS 160:1/26/01

9

User-centered Iterative Design

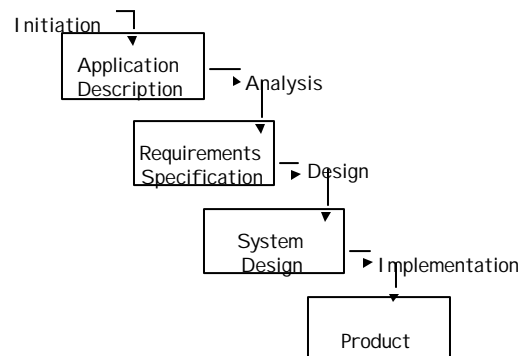
- Developers working **with** target users
- Think of the world in users terms
- Understanding work process
- Not technology-centered/feature driven
- I terate at every stage



CS 160:1/26/01

10

Waterfall Model (Soft. Eng.)



CS 160:1/26/01

11

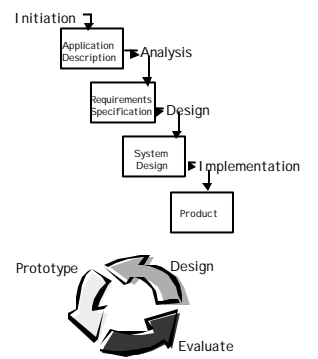
Why Do It?

- Nearly 25% of all applications projects fail
Over run budgets & management pulls the plug
Others complete, but are too hard to learn/use
- Solution is user-centered design
Easier to learn & use products
Helps keep development on or ahead of schedule
Reduced training costs
Improved management visibility

CS 160:1/26/01

12

Waterfall vs. Iterative User-Centered Design



The diagram illustrates the Waterfall model on the left and the Iterative User-Centered Design (IUCD) model on the right. The Waterfall model is a linear sequence of boxes: Initiation, Application Description, Analysis, Requirements Specification, Design, System Design, Implementation, and Product. The IUCD model is a circular process with three main stages: Prototype, Design, and Evaluate, connected by arrows in a clockwise cycle.

- Focus differs
 - IUCD focuses on user's perspective
 - WF focuses on developing working software
- WF has no feedback (?)
- Fixing errors is expensive
 - Increases by factor of 10 at each stage
 - IUCD finds errors earlier

CS 160:1/26/01 13

Design

- Design driven by requirements
 - Purpose** for artifact
 - Not **how** it is to be implemented
- Design represents the artifact
 - screen sketches or storyboards
 - flow diagrams/outline showing task structure
 - executable prototypes
- Design representation should reduce complexity

CS 160:1/26/01 14

Task Example


- Task: write an essay
- User scenario...
 - Write essay**
 - start word processor
 - write outline
 - fill out outline
 - Start word processor**
 - find word processor icon
 - double click on icon
 - Write outline**
 - write down high-level ideas
 - ...

CS 160:1/26/01 15

Web Design Representations

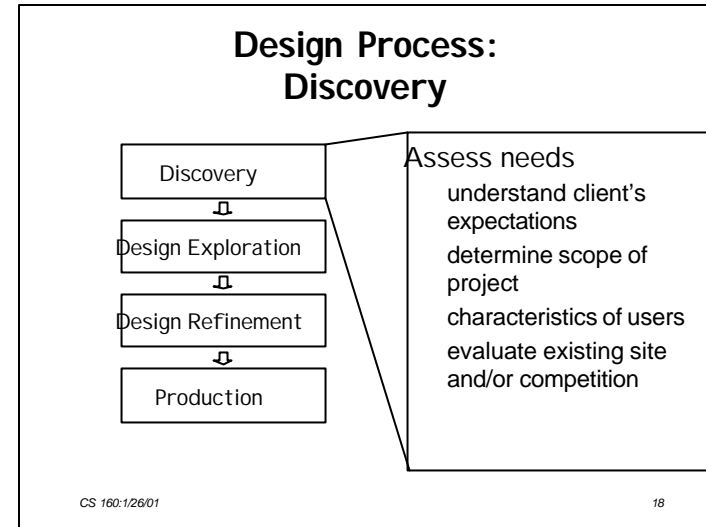
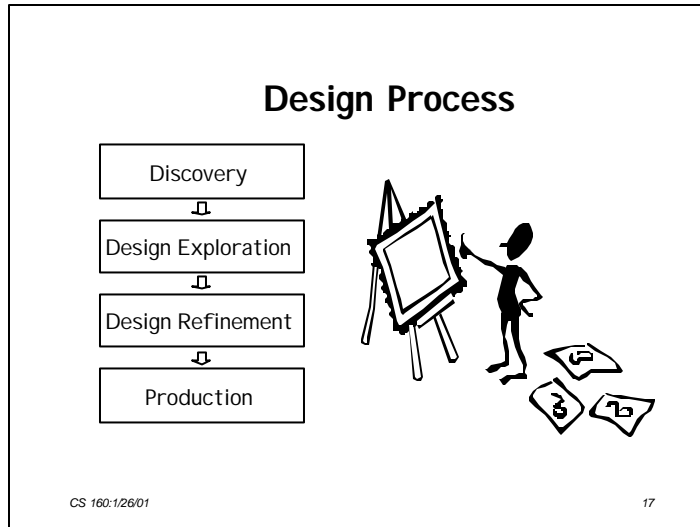
- Designers create representations of sites at *multiple levels of detail*
- Web sites are iteratively refined at all levels of detail

Site Maps Storyboards Schematics Mock-ups



The image shows four examples of web design representations: a Site Map (a hierarchical tree diagram), a Storyboard (a sequence of panels showing user interactions), a Schematic (a detailed layout of a page), and a Mock-up (a realistic visual representation of a web page).

CS 160:1/26/01 16



- ### Understanding the User
- How do your users work?
Task analysis, interviews, & observation
 - How do your users think?
Understand human cognition
Observe users performing tasks
 - How do your users interact with UIs?
Observe!
- CS 160:1/26/01 19

- ### Involving the User
- Users help designers learn
What their job involves
What tools they use
i.e, what they do
 - Developers reveal technical capabilities
Builds rapport & an idea of possibilities
 - Users try prototype & comment on it
developers make incremental changes & iterate
- CS 160:1/26/01 20

Observation Techniques

- In the work place
- Use recording technologies
 - Notebooks
 - Tape recorders
 - Video cameras
- Ask users to think out loud while working
 - Look for job-specific procedures / terminology
- Show users transcript & ask about it

CS 160:1/26/01

21

What Users Might Say

- "This is too difficult"
- "You don't have the steps in the order we do them"
- Do not take comments personally
 - You should not have a personal stake
- Goal is to make the system easy to use for your intended users

CS 160:1/26/01

22

Caveats

- Politics
 - "Change agents" can cause controversy
 - Must get a sense of the organization
 - Must have buy-in from all those involved
- Design forever without prototyping
 - Key idea: rapid prototyping, evaluation, & iteration
- System-level apps are poor candidates for this approach
 - Networking, etc.

CS 160:1/26/01

23

Example of Poor Design

- BART "Charge-a-Ticket" Machines
 - Allow riders to buy or add money to tickets
 - Takes ATM cards, credit cards, and cash
- Problems
 - One "path of operation"
 - ticket type -> payment type -> payment -> ticket
 - BART Plus has minimum of \$28, no indication of this until after inserting >= \$1
 - cannot switch to regular BART ticket
 - Large dismiss transaction button does nothing

CS 160:1/26/01

24

Why Task Analysis?

- System will fail if not done?
 - App may not do what the user needs
 - App might not solve user's problem
- "System must match users' tasks"
- Why not define set of "good" interfaces?
 - Infinite variety of tasks and users
 - Guidelines are usually too vague
 - e.g., "give adequate feedback"

CS 160:1/26/01

29

Questions



- Who is going to use system?
- What tasks do they now perform?
- What tasks are desired?
- How are the tasks learned?
- Where are the tasks performed?
- What's the relationship between user & data?

CS 160:1/26/01

30

Questions (cont.)

- What other tools does the user have?
- How do users communicate with each other?
- How often are the tasks performed?
- What are the time constraints on the tasks?
- What happens when things go wrong?

CS 160:1/26/01

31

Who?

- Identity of user?
 - In-house or specific customer is easy
 - Several typical users needed for mass market product
- Background
- Skills
- Work habits and preferences
- Physical characteristics

CS 160:1/26/01

32

Who (BART)?

- Identity?
People who ride BART
- Background
Have an ATM or credit card
Use BART fare machines
- Skills
Know how to put cards into ATM machines
Know how to buy BART tickets

CS 160:1/26/01

33

Who (BART cont.)?

- Work habits and preferences
Not applicable
- Physical characteristics
Varying heights
do not make it too high or too low!
Think about elevator control placement in Soda

CS 160:1/26/01

34

Elevator Controls in Soda



- How high off the ground?
- Affordances?

CS 160:1/26/01

35

Talk to Them

- Find real users
- Talk to them
Find out what they do
Ask them how your system might be used
- Are they too busy?
buy their time
t-shirts, coffee mugs, etc.

CS 160:1/26/01

36

What Tasks?

- Important for both automation & new functionality
- Relative importance of tasks?
- Observe users
 - On-line billing example
 - small dentists office had billing automated
 - assistants were unhappy with new system
 - old forms contained hand-written margin notes

CS 160:1/26/01

37

What Tasks (BART)?

- Old tasks?
 - Cash to buy new ticket
 - Cash to add fare to existing ticket
 - Cash or credit to buy a BART Plus at window
- New tasks?
 - Cash, credit, or ATM card
 - buy new ticket or add fare to existing ticket
 - buy a BART Plus ticket
 - new currency!
- Level of detail can vary

CS 160:1/26/01

38

How are Tasks Learned?

- What does the user need to know?
- Do they need training?
 - Academic
 - General knowledge / skills
 - Special instruction / training

CS 160:1/26/01

39

How are Tasks Learned (BART)?

- Walk up & use system
 - Cannot assume much background/training
- Training?
 - Too time consuming
- Must be simple & similar to existing systems
 - BART machines
 - ATM machines

CS 160:1/26/01

40

Where is the Task Performed?

- Office, laboratory, point of sale?
- Effects of environment on users?
- Users under stress?
- Confidentiality required?
- Do they have wet, dirty, or slippery hands?
- Soft drinks?
- Lighting?
- Noise?

CS 160:1/26/01

41

Where (BART)? Train Station

- Loud
 - Dependence on voice I/O not a good idea
- Others looking over your shoulder
 - Lose privacy
 - PIN input must be confidential
 - do not confirm with sound
- Dim lighting
 - Make sure messages are readable

CS 160:1/26/01

42

What is the Relationship Between Users & Data?

- Personal data
 - Always accessed at same machine?
 - Do users move between machines?
- Common data
 - Used concurrently?
 - Passed sequentially between users?
- Remote access required?
- Access to data restricted?

CS 160:1/26/01

43

Data Relationships (BART)

- Personal data
 - Users may use any machine -> store on card
- Common data
 - Fare rules (e.g., how much for BART Plus)
 - Used concurrently
- Access to data restricted?
 - Only you can use your ATM or credit card
- No need for remote access

CS 160:1/26/01

44

What Other Tools Does the User Have?

- More than just compatibility
- How user works with collection of tools
Example: automating lab data collection
 - how is data collected now?
 - by what instruments and manual procedures?
 - how is the information analyzed?
 - are the results transcribed for records or publication?
 - what media/forms are used and how are they handled?

CS 160:1/26/01

45

Other Tools (BART)

- Not relevant

CS 160:1/26/01

46

How do Users Communicate With Each Other?

- Who communicates with whom?
About what?
- Follow lines of organization?
For /against it?
- Example: assistant to manager
Installation of computers changes communication between them
People would rather change their computer usage than their relationship [Hersh82]

CS 160:1/26/01

47

How do Users Communicate (BART)?

- Not relevant

CS 160:1/26/01

48

How Often do Users Perform the Tasks?

- Frequent users remember more details
- Infrequent users may need more help
 - Even for simple operations
- Which function is performed
 - Most frequently?
 - By which users?
 - Optimizing frequent tasks will improve perception of good performance

CS 160:1/26/01

49

How Often (BART)?

- Varying frequency of users
 - Some take BART every day (most)
 - Some take it only occasionally
- Varying frequency of tasks
 - Can only do BART Plus every 2 weeks
 - not frequent → more instructions
 - Might do add fare or buy new ticket every day
 - probably more common
- Observe users to find out for sure

CS 160:1/26/01

50

What are the Time Constraints on the Task?

- What functions will users be in a hurry for?
- Which can wait?
- Is there a timing relationship between tasks?

CS 160:1/26/01

51

Time Constraints (BART)?

- Users will almost always be in a hurry
- Lines form
- Take less than 1 minute/transaction
- Be able to do any task in any order

CS 160:1/26/01

52

What Happens When Things Go Wrong?

- How do people deal with
 - Task-related errors?
 - Practical difficulties?
 - Catastrophes?
- Is there a backup strategy

CS 160:1/26/01

53

Things Go Wrong (BART)?

- Confusion on task
 - "Dismiss transaction" button
- Practical difficulty
 - Generated ticket has too much money on it
 - Cash-in policy?
- Catastrophe
 - Machine eats card → swipe instead of insert
- Backup strategy
 - Use cash in regular machines (use ATM)

CS 160:1/26/01

54

Selecting Tasks

- Real tasks users have faced
 - Collect any necessary materials
- Should provide reasonable coverage
 - Compare check list of functions to tasks
- Mixture of simple & complex tasks
- When you move to rough design stage
 - Discard features that don't support your tasks
 - Add a real task that exercises that feature

CS 160:1/26/01

55

What Should Tasks Look Like?

- Say what the user wants to do, but not how the user would do it
 - Allows comparing different design alternatives
- Tasks should be very specific
 - Forces us to fill out description with other details that become relevant
- Some tasks should describe a complete job
 - Forces us to consider how features work together

CS 160:1/26/01

56

Look of Tasks (cont.)

- See where inputs come from and where outputs go
 - Working with other tools
 - Saving / loading
- Tasks should say who the users are
 - Design can really differ depending on who
 - If possible, name names
 - allows getting more info. as it becomes relevant
 - Characteristics of the users
 - job, expertise, etc.

CS 160:1/26/01

57

Look of Tasks (cont.)

- Reflect interests of potential users
 - Illustrate proposed functionality in context of work
 - Users really want to do
- Users are not always right
 - Cannot anticipate new technology accurately
- Job is to build system users will want
 - Not system users say they want
 - Be very careful about this (you are outsider)
 - if you cannot get users interested in your hot idea, you are missing something

CS 160:1/26/01

58

Using Tasks in Design

- Write up a description of tasks
 - Formally (book) or informally (us)
 - Run by users and rest of the design team
 - Get more information where needed
- Rough out an interface design
 - Major screens & functions (not too detailed)
 - Hand sketched
- Produce scenarios for each task
 - What user has to do and what they would see
 - Step-by-step example of performing task

CS 160:1/26/01

59

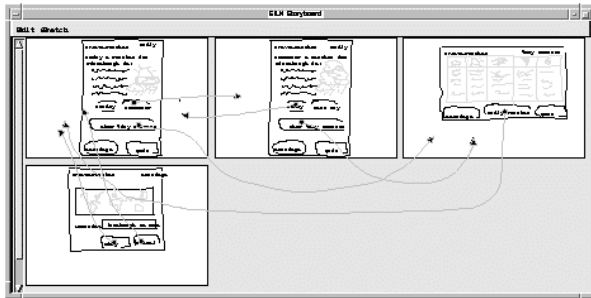
Scenarios (cont.)

- Scenarios are design specific
- Tasks are not design specific
- Scenarios force us to get specific
 - How various features will work together
 - Settle design arguments by seeing examples
- Show users storyboards
 - Sequences of sketches showing screens
 - Actions users can take

CS 160:1/26/01

60

Storyboards



CS 160:1/26/01

61

Other Examples of Storyboards

<http://staff9.cis.uoguelph.ca/planets.htm>
<http://animationartgallery.com/ddraconwash.html>
<http://www.sbdoc.com/>
http://www.cs.berkeley.edu/~landay/research/publications/CHI96/short_storyboard.html
http://theforce.net/troops/t_story.shtml
<http://thesims.ea.com/us/about/storyboards/>

CS 160:1/26/01

62

Review

- Answer questions before designing
 - Who, what, where, when, how often?
 - Relationship between users & data?
 - What other tools do users have?
 - What happens when things go wrong?
- Selecting tasks
 - Cover real tasks with reasonable functionality
- What should tasks look like?
 - Description of specific problem to be solved
 - Tasks can be abstract – “go to grocery store”
- What should scenarios look like?
 - Steps taken to solve a particular example of a task

CS 160:1/26/01

63

Your Assignment

- Complete task analysis for your app

CS 160:1/26/01

64