Pluralistic and Cognitive Walkthroughs
Overview

• The Nature of a Walkthrough
• Pluralistic Walkthroughs
• Cognitive Walkthroughs
• Cognitive Jogthrough
• Comparisons to Usability Testing
The Nature of a Walkthrough

• Walkthroughs are an *evaluation technique* rather than a planning/design technique
• What is essential for any walkthrough is that the system specs be detailed and complete
• Depending on the type of walkthrough, a prototype (a testable version) is either:
  – a requirement (pluralistic walkthrough)
  – nice to have but not a requirement (cognitive walkthrough)
Walkthrough & Prototypes

• Prototyping for website walkthroughs could be done on paper or in (X)HTML
  – Paper versions referred to as storyboards
• Aim for at least medium fidelity in the prototype
  – Fidelity refers to how closely the prototype matches the final design in look and functionality
  – In this case you don’t need to display the final design (in terms of colors/images), you primarily need the layout and labeling to be intact
Shared Aspects of Walkthroughs

• Walkthroughs attempt to model (to predict and understand) the behaviors of users
  – This includes both mental and physical behavior
• Rather than focus on a single user and his/her idiosyncrasies, a walkthrough generally focuses on the user group
  – Especially true for cognitive walkthroughs
• The focus is always on first-time users and how they will react to the system.
  – Experienced users are not considered
Introducing Pluralistic Walkthroughs

• This type of walkthrough involves multiple groups, including the users (thus the use of ‘pluralistic’ in the name)

• Representatives of at least three groups are present for the walkthrough:
  – Users (at least two, hopefully more)
  – User experience professionals (one or two; generally serve as moderator and recorder)
  – Programmers (one or two)

• Other relevant groups could also be present
Pluralistic Walkthrough Process

1. A task is chosen for testing
2. Storyboards are prepared for that task (e.g., registration, checkout) and the first storyboard is given to each person present
3. Approaching the task as a user would, each writes on the first storyboard (or on a piece of paper) the actions to take:
   • press the down arrow key twice to scroll the page, click this empty text box, type text in it, then click this button next to the text box
Pluralistic Walkthrough Process

4. Once everyone is finished with that first storyboard they compare notes and discussion begins
   • Users always give their input first

5. At the end of the discussion the facilitator shows the ‘correct’ sequence of actions (based on the specs/use cases) and then the next storyboard is distributed to participants
Pluralistic Walkthrough Process

6. This process of individual analysis, followed by group discussion and then a review of the ‘correct’ actions, is repeated for each new storyboard, one at a time.

7. A list of prioritized usability issues and their recommended solutions is the end result.
Facilitating the Walkthrough

• Moderator
  – Introduces the task
  – Distributes the storyboards
  – Keeps discussion on task
  – Informs participants of ‘correct’ actions

• Recorder
  – Takes notes during discussion
  – Gathers storyboards after each ‘round’
  – Prepares list of usability issues and solutions
Important Considerations

• Make sure that everyone understands the task to be accomplished before the first storyboard is analyzed

• Let the users offer their experiences first and allow them to *completely describe* their actions, then have everyone else join in
  – Otherwise a programmer/developer may cut off one of the users prematurely

• Allow yourself plenty of time for the walkthrough of a multi-step task
Value of Pluralistic Walkthroughs

- Group discussion is very valuable for identifying issues and solutions.
  - The diversity of perspectives is a definite strength of this method
- Can educate developers and other parties concerning what derails the user and how to fix those problems
- Only requires a paper prototype, so changes are made early (before major development has occurred)
Possible Issues

• Hostility from the programmers
  – The programmers may not be used to receiving this sort of feedback
  – Having the programmers do some code reviews in the preceding weeks can get them to be more open to receiving feedback

• Missing issues outside the defined task
  – Since the focus is on this one task and how it would be completed by a first-time user, issues outside the task are effectively ignored
Introducing Cognitive Walkthroughs

• In this type of walkthrough the users are not involved
  – Just the user experience experts are involved
  – A difference compared to pluralistic approach

• As a result, it is crucial to thoroughly understand the knowledge and skills that a first-time users would possess
  – If this is not well understood then it is possible to miss important issues or to misjudge the difficulty presented by the interface
Walking in the User’s Shoes

• User experience experts truly need to “walk in the user’s shoes” and try to experience the interface from that perspective

• This can be a more difficult process than one would expect, especially staying in that ‘new user’ mode
  – Personas are extremely helpful here

• During the walkthrough the goal is to tell a believable ‘story’ of how the user would behave (while also identifying issues)
Defining the Users / Personas

• Experience/exposure to that OS
  – Windows, Mac, Linux, Unix
• Experience/exposure to similar software
  – You are creating a new imaging program; are users likely to have used Photoshop previously?
• General level of computer experience
• Other considerations relevant to the specific task being analyzed
Comparisons to Pluralistic Method

- Both are conducted in a group
  - Groups tend to be smaller in cognitive method
  - Can be conducted alone (pluralistic cannot)
- Still involves a moderator and recorder
  - Moderator has less work (mostly keeping the discussion on track)
  - Recorder has more work (no storyboards with actions already written on them)
- Does not involve users or programmers
  - Pluralistic can identify individual user issues; cognitive is at the user group level
Comparisons to Pluralistic Method

• Still has the issue of limiting the scope of usability problems that can be found, due to the focus on a specific task or tasks
• Both are focused on a task or tasks and approach each task at the level of actions
• Can be conducted without a visual interface, just based on use cases and system specifications
  – Pluralistic requires something visual
Theoretical Basis

- Lewis and Polson's CE+ theory of exploratory learning
- Exploratory learning occurs in four steps:
  1) The user sets a goal to accomplish with the system
  2) The user searches the interface for any currently available controls to accomplish that goal
  3) The user selects the control and corresponding action that seems likely to make progress toward the goal
  4) The user performs the selected action and evaluates the system's feedback for evidence that progress is being made toward the current goal
Defining the Functionality

• In order to successfully conduct a cognitive walkthrough you need to understand the following for each task analyzed:
  – The actions that comprise the task
  – What precedes each action (e.g., a prompt)
  – What follows each action (e.g., a confirmation page or interface feedback)
Cognitive Walkthrough Process

1. A task/goal is chosen, with a defined starting condition and ending condition

2. This task/goal is accomplished through a series of actions, both physical and mental.
Cognitive Walkthrough Process

3. For each action these questions are asked:
   A. Do the users recognize that this action is necessary to achieve the desired goal?
   B. Can the users locate the control (link, button, etc.) that triggers the action?
   C. Once the control is activated, do users recognise that it relates to the desired effect?
   D. After the action is taken do the users understand the feedback they get, so they perceive progress is being made?
Cognitive Walkthrough Process

4. A list of prioritized usability issues and their recommended solutions is the end result.
A. Recognizing Action is Necessary

- How they recognize an action is needed:
  - System instructs them to do so
  - They know this from similar experiences
  - Because it is explicitly associated with the desired goal

- Example:
  - Task is to locate a book in a library database. Entering a search term is the first action. Entering queries is essential to search, which is known from similar previous experiences.
B. Locating the Control

- How they locate the control to which they apply the action:
  - Being able to see the link in a navbar
  - Having a button available with what looks to be an appropriate label
  - Knowing where to look from similar experience

- Example:
  - There is a text input box with a submit button next to it labeled ‘Search’.
C. Control Relates to Desired Effect

- Realizing the control was appropriate one:
  - System gives instruction that makes the relationship obvious or labeling is obvious
  - They recognize the relationship between control and effect from other experiences
  - No other controls look right

- Example:
  - There is a text input box with a submit button next to it labeled ‘Search’.
D. Perception of Progress

- Seeing that they are making progress:
  - System indicates progress via text output or some other response
  - They recognize progress from similar experiences

- Example:
  - Page changes to a list of search results following submission. The page change and the list of results indicates progress was made.
Stories of Success / Failure

• For each action (and a given task/goal could have many actions comprising it) a story of success is given if all four questions are satisfactorily addressed

• A failure story arises when one or more of the four questions are not satisfactorily addressed
  – In those cases it is time to start exploring solutions to the problems
Cognitive Walkthrough Data

• Use of a standardized form (structured around the four questions) is very beneficial for efficient data recording and later analysis
• Clearly note any failures/issues identified and begin to work out solutions to those issues
Cognitive Jogthrough

• Similar to a cognitive walkthrough, except the session is videotaped so the data recorder role is not necessary
• Discussion is more lively because the pace is picked up
• Allows more tasks to be considered because of the increased rate of speed and greater flexibility in the process
Comparisons to Usability Tests

- A walkthrough can identify more *task-level issues* than a usability test.
- Yet the usability test identifies a whole range of issues and is not limited to one task, so ultimately more issues are found.
  - Estimates are that a cognitive walkthrough finds less than half of what a usability test finds.
  - Pluralistic walkthroughs fare about as well.
  - Walkthroughs are no substitute for a usability test.
Comparisons to Usability Tests

- The usability test is also appropriate at any point in the development process
  - Walkthroughs are useful early in development
  - Cognitive walkthroughs just need a good understanding of users and the system specs
  - Do a walkthrough before the usability test in order to catch and fix major task-level issues

- Walkthroughs can often be done more quickly than a usability test